A METHOD FOR DECENTRALIZED BUSINESS PROCESS MODELING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

OKTAY TÜRETKEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

JUNE 2007

Prof. Dr. Semih BİLGEN	(METU, EEE)	
Assoc. Prof. Dr. Onur DEMİRÖRS	(METU, II)	
Assoc. Prof. Dr. Ali DOĞRU	(METU, CENG)	
Dr. Altan KOÇYİĞİT	(METU, II)	
Prof. Dr. A. Kadir VAROĞLU	(BAŞKENT, MAN)	

ABSTRACT

A METHOD FOR DECENTRALIZED BUSINESS PROCESS MODELING

Türetken, Oktay

Ph.D., Department of Information Systems

Supervisor: Assoc. Prof. Dr. Onur Demirörs

June 2007, 237 pages

This thesis study proposes a method for organizations to perform business process modeling in a decentralized and concurrent manner. The Plural method is based on the idea that organizations' processes can be modeled by individuals actually performing the processes. Instead of having a central and devoted group of people to understand, analyze, model and improve processes, individuals are held responsible to model and improve their own processes concurrently. These individual models are then integrated to form organization's process network. The method guides the application of this approach in organizations with the activities to be followed and the artifacts to be produced. To apply the method, the study also introduces a notation and a prototype toolset. A multiple-case study involving two cases is conducted in order to evaluate the applicability of the method for decentralized process modeling and validate the expected benefits.

Keywords: Business Process Modeling, Decentralized Process Modeling, Role-Based Process Modeling

ÖZ

ÖZEKSİZ İŞ SÜREÇLERİ MODELLEME İÇİN BİR METOT

Türetken, Oktay
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Doç. Dr. Onur Demirörs

Haziran 2007, 237 sayfa

Bu çalışma, organizasyonların özeksiz (merkezi olmayan) ve eşgüdümlü bir şekilde iş süreçlerini modellemelerini sağlayacak bir metot önermektedir. Plural metodu, süreç modellenmenin süreçleri gerçekleştiren bireylerce yapılabileceği fikrine dayanmaktadır. Merkezi bir grubun süreçleri anlaması, analiz etmesi, modellemesi ve iyileştirmesi yerine, süreç sahipleri kendi süreçlerini eşgüdümlü bir şekilde modelleme ve iyileştirmeden sorumlu tutulmaktadır. Birleştirilen bu bireysel modeller organizasyonun süreç ağını oluşturmaktadır. Metot, bu yaklaşımın takip edilecek faaliyetler ve üretilecek ürünler açısından organizasyonlar tarafından etkin bir şekilde uygulanması için yol göstericidir. Çalışma, bu yaklaşıma özel bir notasyon ve prototip bir araç seti ve metodun özeksiz iş süreçleri modelleme yaklaşımına uygunluğunun değerlendirilmesi ve beklenen yararların gerçekleştiğinin doğrulanması amacıyla gerçekleştirilen çoklu-örnek olay incelemesini içermektedir.

Anahtar Kelimeler: İş Süreçleri Modelleme, Özeksiz Süreç Modelleme, Rol-tabanlı Süreç Modelleme

To my parents, Emire & Hasan Türetken

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my supervisor Onur Demirörs. The ideas that stimulated this study were all his works. I am deeply grateful for his fate in me and letting me to grow these ideas. All this time, he has been supportive, generous and patient.

I wish to express my warm and sincere thanks to my committee members Semih Bilgen and Altan Koçyiğit for their insightful comments and suggestions over the last three years. I would also like to thank the committee members Ali Doğru and A. Kadir Varoğlu for their valuable comments.

My most sincere appreciation goes to my friends Çiğdem Gencel, Alpay Karagöz, Ali Yıldız and Ayça Tarhan. They never hesitated to provide support whenever I needed it. Our extensive discussions and their comments were invaluable. I would also like to thank Ali and Alpay for backing me up during my frequent absences.

Many thanks go to my friends Alpay Ertürkmen, Sibel Gülnar and Funda Akgür for participating in case studies and providing valuable comments. Thanks also go to Özge Biçer and Doruk Eker for applying the approach on their own and providing valuable feedbacks. My sincere thanks are due to my friends in Informatics Institute for their company. I have many good memories from these years.

I am also grateful to my family-in-law. They were always considerate and supportive.

Finally, I am deeply grateful to my parents and my brother for their endless support and encouragement during these years*. They never doubted me. I always knew that they were with me whenever I needed them. But my wife, Nermin, had the heaviest the load. She shouldered much of my hesitations and worries, and I am eternally grateful for her support and love. Without her belief, encouragement and understanding, I am not sure I could have done it.

** Benden her türlü desteği hiçbir zaman esirgemeyen ve bana herzaman inanan Anneme, Babama ve kardeşim Engin'e en içten teşekkürlerimi sunuyorum.*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In many fields of organizational life, such as creating process scripts for workers to follow, establishing quality manuals, assessing and identifying added value, establishing control mechanisms, automating workflow and identifying software requirements, we observe process modeling as a core activity. As a result, process modeling has become a major focus of attention in many fields. These include but not limited to business process management, workflow management, enterprise modeling, business process reengineering, and software process improvement. Although these fields generally exploit process models for different purposes, one way or another, process modeling becomes the center of their frameworks.

Majority of process modeling initiatives utilize central approaches for process definition in which group of experts, sometimes called 'process engineers' or 'process engineering group' ([7], [47], [86], [101]), work with the individuals -actually performing the activities- in order to understand, model and improve organization's processes. With a top down, centralized approach, unavoidably it takes considerable amount of time to model an organization's processes. Once processes are stable it is generally difficult and not desired to change them frequently [86]. However, to respond to the demand of the markets, organizations should be able to change their way of working rapidly. That is, any improvement opportunity raised in the business should be immediately captured and incorporated into the organization process-base. In other words, we need to reduce the cycle time for modeling and improvement in the order of days. Process infrastructure should also enable real users besides process engineers or process groups to perform such changes [117], [128].

An alternative approach to this centralized view is to delegate this responsibility to individuals or teams that actually perform the processes. This research proposes a method for process modeling to be performed concurrently and in a decentralized way by each agent in the organization. It demonstrates how an organization can perform process modeling in such a manner and discusses its advantages and limitations. A method that enables each agent to own its processes would bring the notion of process thinking in the organization and, in turn, would provide mechanism for improvement to be initiated at the bottom and concurrently spread over the organization. Process modeling in this manner would take less time to develop and would enable organization's process-base to be maintained more easily and efficiently.

1.1. The Context

The growth of information society increased the significance of knowledge which in turn increased the importance of process models. Information society has also increased a wider distribution of knowledge and expertise within the organization and society as a whole [49]. The results of wider distribution of knowledge also enables (and requires at the same time) organizations to change more frequently and to change much faster. The process model infrastructure of the organizations of information society therefore should enable frequent and rapid changes.

In the society, where knowledge is the primary resource for individuals and for the economy, many researchers on business management agree that the traditional structures of organizations are not appropriate for creating products and services that require knowledge work and its integration [49], [6], [127]. According to Senge [127], the unified premise of quality movement is 'to make continual learning a way of organizational life, especially improving the performance of the organization as a total system'. Senge acknowledges that this can be achievable if traditional authoritarian, command-and-control hierarchy -where the top thinks and the local acts- is broken. Merging thinking and acting at all levels is necessary. One of the first prerequisites of this achievement is removing impediments disempowering the workers, such as the quality control experts, unnecessary bureaucracy, and providing authority, environment and appropriate tools to perform their work.

With the assumption that knowledge is at the top and unskilled workers are at the bottom, in traditional vertical structures, skilled workers at the top first analyze and then optimize the information at the bottom, and return this information as task descriptions to lower levels. This might be an efficient and an appropriate method in conventional production organizations where the majority of tasks can be automated or do not require integration of knowledge work. In knowledge based environments, however, the greatest knowledge is at the bottom where it is created and should be analyzed. This type of organizations can be viewed as being consist of autonomous, interacting and collaborating units, which own their tasks, information and resources involved in the process. They perform their processes concurrently and interact when needed. Knowledge workers, in these organizations, have their own individual models of their processes and ways of tackling particular problems. The knowledge of 'how a knowledge worker carries out his process' belongs to the knowledge worker himself, so he is of little value if his job can be prescribed [50]. Hence, these professionals feel that they require some autonomy in planning, executing and controlling their work and applying their knowledge without close supervision [129]. As the studies by Sommerville [129] reveal, these professionals require some control over their work activities and strongly resent particular work practices imposed by the organization unless they participate in the design of that process.

In general, studies on business process modeling focus on notations, architectures and tools for process definitions and execution of these definitions [8], [25], [125], [126]. These studies generally assume that the definitions will somehow be developed and maintained by a group of people in the organizations, generally referred as process engineers or process developers. Studies on process improvement, on the other hand, commonly introduce two major roles that are associated with the definition and maintenance of processes [86], [91]. First one is the practitioner role (process performer or process owner [29]) that *provides input to* process definition and improvement and the second is the process group (process engineering group) that is responsible for *facilitating* and *managing* the process of definition and improvement. However, in many of these works, the roles that are responsible from the definition of processes are implicit or hidden. Some of these studies even avoid or hesitate to name the role that actually performs process definition and maintenance (e.g. [86]). Others admit that, due to several reasons, this responsibility is mostly left to the process engineering groups [7] and few of them refer to a another group 'process improvement project team' besides the process group, responsible to define and improve organization's processes [150].

One way or another, process modeling is generally performed by a group of experts (process group, process engineers, process improvement team, process model designers ([7], [29], [51], [94], [150]), who frequently work with the groups of individuals -actually performing the activities- one by one in order to understand, define and improve organization's processes. The degree of the involvement of these individuals is one of the most critical factors deriving the success of these projects. If an organization designs processes so that the process performers have no motivation to follow it and are not held accountable for their actions, they will probably not follow it as designed [88]. Similarly, there are clear examples of software systems that failed because it was not designed to meet its users' goals [106]. Alter [3] signifies the importance of employee involvement in improving their work practices.

Armour [7] identified many problems that might arise when process is developed by people who do not actually employ it. With the approach where process engineers develop the processes with little or no user involvement, the outcome is generally high-level, disconnected, vague, paper-intensive, and incomplete processes. In addition, the value of devised processes is presumably questionable and not proven. Since process engineers are usually not employing the process they are devising, they may lose the ability to validate that the process is actually helpful. The definitions are generally left at the description level, in book form, not the executable level, since there is a lot more knowledge the process engineers need to get to be able to include the process knowledge to the level where it actually executes. If this is made available, it would greatly increase the effort and time for process engineers to define the process. If defined in detail, the definitions tend to become constraining rather than value adding, since process engineers are rarely required to prove the value of the processes they define. The frequent outcome is a resistance against the processes that are defined this way.

Baddoo & Hall [10] performed an empirical study on 200 software engineers in different positions in 13 companies in order to investigate the perceptions that different staff groups have regarding their role in process improvement. They found out that these practitioners consider process improvement activities as a management responsibility. Developer groups do not perceive themselves accountable for process improvement and they do not consider themselves empowered for process improvement. In general, these practitioners do not see themselves assuming ownership of processes.

Having recognized the importance, in general, process modeling and improvement approaches urge organizations to motivate and empower their employees - process performers - in taking parts in process definition and improvement projects [10]. Model creation, which was a skill left to experts, becomes a necessity for the people in enterprises for better understanding and manipulation of the models and evaluation of process alternatives [93]. As highlighted by Barnett [13], "business processes belong to the 'business' and the responsibility for defining them should as well."

1.2. The Problem

Many researchers acknowledge the significance and necessity of having teams or individuals growing and maintaining their own processes rather than having a process engineering team to take responsibility for that ([7], [57], [88], [150]). However, none of these works provides a concrete and explicit method that would enable this to happen in that way. Because current approaches for process modeling implicitly or explicitly assume a central unit (e.g. process engineering group) that performs and controls the process definition as a whole ([7], [8], [91]). They lack necessary mechanisms to enable the process modeling to proceed in a decentralized way. This makes it more difficult for individuals to actually own the definitions since decentralized enactment of processes by this autonomous agents need a similar architecture in defining their processes as well.

There are very few attempts (such as [8 p151]) putting process owners - the individuals actually performing activities - in to the centre of process definition. However, these initiatives utilize methods and related notations where the centralized process definition is inherent. This time, the centralized way of process definition is shifted from process engineers to a number of enthusiastic end users. Hence, the benefit is limited and disadvantages and flaws of the centralized structure are innate.

In centralized approaches, process modeling is performed sequentially starting from a functional group and it requires a significant amount of effort and time for organizations to model their processes. Besides, due to several reasons including the bureaucracy in the change process, issues related to the rights and responsibilities, and the difficulties in impact analysis; frequent changes in stable processes are generally avoided in these centralized architectures [8] [86]. However, with the increasing concerns about the competitiveness in the market, organization's processes should be quickly adapted to the changes and opportunities. Besides, organizations' process infrastructure should enable these rapid changes to be incorporated in the process-base not only by process engineers or process groups but also by the process owners.

The idea of decentralized modeling by process owners is first proposed by Demirors [40], [39] as the Horizontal Change Approach (HOC-A) to manage change in software development organizations. HOC-A proposes process modeling and change to be performed in a decentralized manner concurrently by all the members of the organizations. An approach to be effectively applied in organizations and achieve its benefits needs a systematic way and mechanisms as guidance for organizations.

There is a need for a method that would enable agents to concurrently model their processes, help them to identify and resolve inconsistencies between other agents' definitions, enable them to easily integrate these partial models to form the organization's process network, and finally allow them to continuously maintain their own definitions.

1.3. The Solution Approach

Process owners' involvement in process definition and improvement is critical for knowledge based organizations ([7], [57]). As the degree of the involvement of the knowledge workers who perform the processes increases, the likelihood of the model to reflect the executed process as well as the likelihood of the performers to embrace the models increases. Hence, going beyond the participation in process definition is to fully delegate this responsibility to them by empowering these individuals to take the responsibility of defining and maintaining the way they perform their activities. Instead of having a group of process engineers to model, evaluate and improve the processes; let these knowledge workers to take responsibility for measuring their processes, identifying their problems, improving their processes and providing feedback for the processes they consume.

Each individual, as the owner of his activities, models and maintains his processes in coordination with the suppliers and consumers of these processes. In this scheme, process engineering group (or the process coordination team as its name in the method) are the coordinators and catalysts between individuals and ensures that the activities of process modeling is performed and maintained as planned. In that sense, they are only then enabled to facilitate and manage the definition process. The only process they own in the organization is the meta-process - the process of process modeling.

In such an approach, each agent, with an assumption that s/he has the greatest knowledge of her/his activities, is expected to model the activities it performs with all agents in the organization in concurrent and a decentralized way. However, an organization going for this approach is confronted with some limitations in utilizing the notations and tools used for centralized approaches. First, there would be considerable degree of inconsistencies between these individual models depicting a partial representation of the process. Secondly; -also related to the former- it would require significant effort to integrate these separately designed models due to the inconsistencies and overlaps. In addition, once integrated, individual models become outdated and leave their places to the integrated models on which individual maintenance is no longer valid. Any change in the way the business works should now be reflected on these integrated models and be performed centrally.

1.3.1. The Plural Method

In the light of the current context in process modeling, its difficulties and limitations, we propose a method that guides organizations to perform a concurrent and decentralized business process modeling.

The method is based on the idea that organizations' processes are carried out by roles that are played by agents. In an organization, there is a network of processes each of which is performed by one or more roles and a role may take part in many processes. Agents participate in these processes by taking over roles. Figure 1 depicts an abstract example demonstrating these relationships. For example, 'Process W' is carried out by 'Role C' and 'Role E' which are both played by 'Agent 3'.

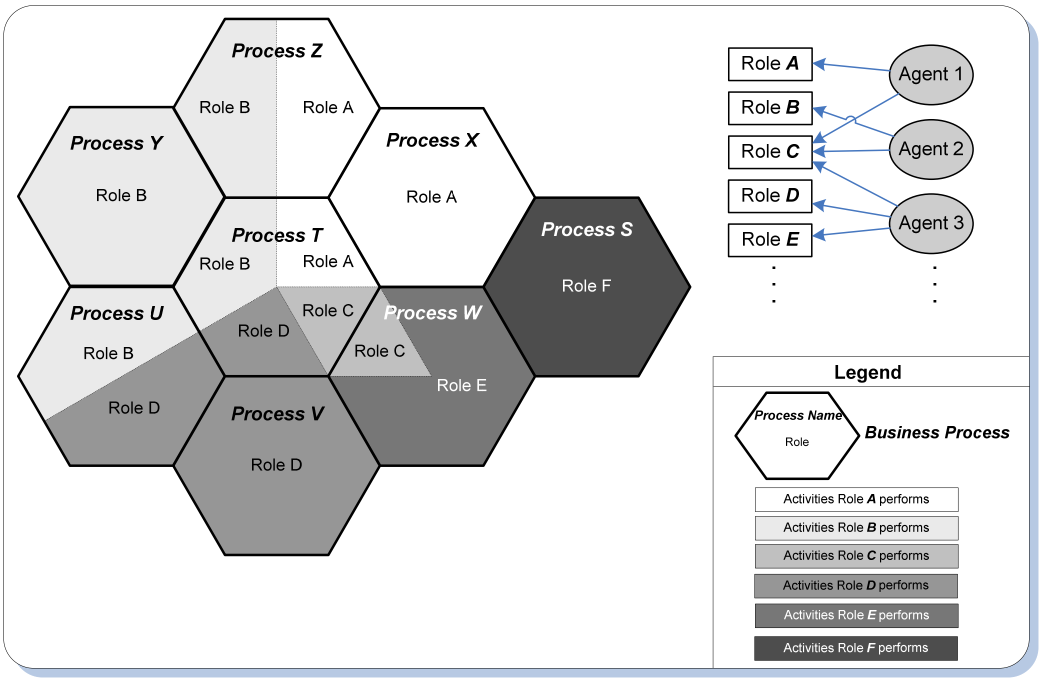


Figure 1. Business Processes, Roles and Agents

Each agent defines the way it performs its activities with respect to the roles it plays in that process. The definition includes the activities that role performs, the inputs and resources it requires and the outputs it produces. In addition to this knowledge, each agent is also expected to define the roles that they get that inputs from. Similarly, they also define the roles that they sent their outputs to. These set of expectations make up the interface points between roles. As definition continues, the expectations are eventually fulfilled and the interface between the roles became apparent.

The organization goes through three phases during process definition:

- The *context definition* phase defines the aim and scope of the modeling process by identifying the processes to be covered and the roles that participate in these processes. Each agent is assigned to one or more roles with respect to their responsibilities in the organization.
- In *description and conflict resolution* phase, development agents define the activities they perform and their expectations regarding to each role they are assigned to. During definition, agents consider other agent's expectations as well and identify inconsistencies between them. If these exists a conflict between agents, they negotiate and resolve it. Individual role-process diagrams are then verified by the coordinator and validated by the peer agent(s).
- Consistent role-process models are merged in *integration and change* phase. Also in this phase various diagrams, such as role and process dependencies, are generated and analyzed for improvement opportunities.

Figure 2 presents the phases and data flow among them. Any change is fed back to previous phases.

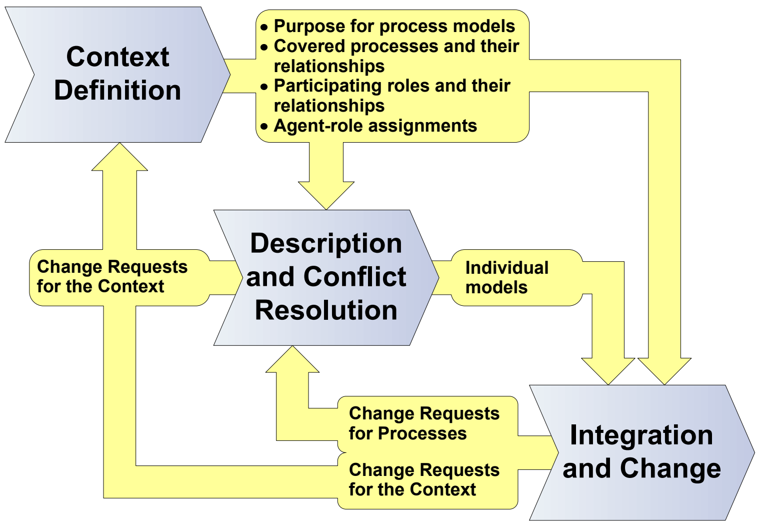

Context Definition
• Purpose for process models
• Covered processes and their relationships
• Participating roles and their relationships
• Agent-role assignments
Description and Conflict Resolution
Change Requests for the Context
Individual models
Change Requests for Processes
Change Requests for the Context
Integration and Change

Figure 2. Method Phases

The method is more suited for organizations where knowledge is distributed among workers. In these organizations the knowledge workers play the central role in performing activities and achieving organization's goals. They interact and collaborate among themselves and with computerized tools to increase their efficiency and effectiveness.

In Plural, knowledge workers are expected to define the processes with a modeling notation. It is also presumed that the knowledge workers have knowledge and organizational environment that provide empowerment and motivation to continuously improve their processes. Therefore, self-modeling is considered to be the very basic part of their responsibilities rather than an additional burden loaded on daily work activities. From agents' perspective, it is an enabler for them to think, understand, define and improve their own processes.

1.4. Research Strategy

The research objective of this study is to develop a method for knowledge based organizations to enable process modeling to be performed in a decentralized and concurrent manner. We evaluated the current approaches with respect to their applicability for decentralized business process modeling. Based on the findings discovered, a method is developed. In order to observe the applicability of the method and its components, and validate the expected benefits, a multiple-case study involving two case studies was conducted. The method gone through significant improvements in the first case and take its current status in the second case study.

The first study was performed in Informatics Institute, METU (Middle East Technical University). The objective of the study was to examine the drawbacks of the method and enhance its components including its phases and activities as well as the notation and the toolset. It mainly covered the processes of Information Systems Department such as 'student admissions', staff recruitments', 'instructing', and etc. Agents modeled their own processes with respect to the roles they play and their expectations from others and these models were then integrated into complete process models of the department.

The second case evaluated the applicability of the method and reviewed its benefits as well as its limitations. It included a set of processes of a small size software organization. Software engineers exploited the method and other components, and were interviewed to provide feedback on the approach followed and to elicit whether expected benefits are observed or not. Advantages as well as the limitations were examined.

1.5. Organization of the Thesis

The remainder of the thesis is structured into five chapters.

In Chapter 2, related researches on decentralized business process modeling in business process management, software engineering and other areas are surveyed. The advantages and limitations of related works with respect to the requirements of our method are analyzed and described.

Chapter 3 forms the hearth of the thesis and describes the proposed method in detail. The phases of the method, its activities and the artifacts to be produced are described. Assumptions of the method, related roles, their responsibilities as well as their skill requirements are discussed. Also in this chapter, the notation and the toolset components are described.

Chapter 4 presents the implementation of the approach in two case studies. The chapter gives the details of the implementations, their results as well as the lessons learned and discussions.

Chapter 5 presents the conclusions reached and summarizes the contribution of this research. New questions that are raised by our research and the subjects that require further investigations are also described in this chapter.

CHAPTER 2

RELATED RESEARCH

This chapter summarizes the literature related to decentralized business process modeling. First section of the chapter describes the Horizontal Change Approach as one of first studies suggesting processes to be owned and modeled by individuals actually performing the processes for managing change in software engineering organizations. Section 2.2 describes the viewpoints approach applied in software engineering and process elicitation fields and can be applied to the process modeling problems. In section 2.3, related work on process-centered software engineering environments are described. Section 2.4, 2.5 and 2.7 discusses the concepts related to business process management, enterprise modeling and modeling notations, respectively. Finally, section 2.8 discusses agent based approaches to business process management.

2.1. Horizontal Change Approach and Horizontal Change Notation

One of the most influential study on decentralized process modeling is the Horizontal Change Approach (HOC-A) and its notation (HOC-N) proposed by Demirors [40]. HOC-A introduces the idea of agents modeling their activities in a decentralized manner to manage change in software development organizations. In HOC-A, process modeling and change are performed in a decentralized manner concurrently by all the members of the organization. In this sense, it is analogous to neural networks in which the overall goal is achieved collectively without direction at any specific organizational level.

Demirors argue that [41] the methods exploited for software process improvement are based on the following implicit assumptions:

- 'The principles of quality management in manufacturing can readily be applied to software development process.' [72]
- 'The essence of software development process can be explicitly defined and the model can be enacted in real world.'
- 'Process modeling can be accomplished vertically, starting from the top and can be handled as an engineering problem. That is, the process modeling can be accomplished

vertically, starting from the top (the requirements) of the software development process and handled as an engineering problem to be implemented by a set of experts in the process improvement field.' Demirors call this approach *vertical process improvement*. The approaches in software quality movement assume a central specification of process models by process modeling experts ([101], [29], [85]). The activity is also called 'Software process engineering' [101]. However, this vertical approach might have some difficulties in knowledge intensive organizations [99]. Processes set by a group of people are enacted by knowledge workers whereas '… a knowledge worker has little value if her job can be prescribed [50].'

Interrelated with these assumptions, Demirors identified inherent difficulties of software quality movement and proposed the HOC-A, where process improvement is perceived in the context of *change* and thus in the context of creating new knowledge. The emphasis is on creating an environment for change that utilizes the expertise of all knowledge workers to achieve organizational goals. The principles of HOC-A can be summarized as follows [39]:

- Processes should enable the surfacing of assumptions and expectations in an organization: gaps between the inherent assumptions and expectations of individuals cause major process inefficiencies.
- Creation of new knowledge is an essential part of work for all knowledge workers: knowledge creation, including process knowledge, is embedded in every grain of the organization.
- 'How to measure' determines 'what to measure': the development of metrics and measurement criteria should be performed in collaboration with the owners and the consumers of the processes and the products it produces.
- 'The reward is modeling it, the punishment having done it': process description activity is more important than the process models it produces. The focus of process modeling activity is to surface problems and to generate solutions for process owners rather than to prescribe, monitor and control.
- Diverse communication should occur among individuals and spontaneous distribution of knowledge should occur irrespective of hierarchy.
- 'More control occurs from less control': knowledge workers should be empowered to take responsibility for measuring their processes, identifying their problems, improving their processes and providing feedback for the processes they consume.

From the viewpoint of process modeling and design, agents are generally viewed as humans, or machines that perform an activity [32], [2], [137]. In HOC-A, however, agents are responsible for three distinct roles; modeling, change and enactment [39]. Agents model their own process as they perceive. In that respect, modeling role is similar to those performed by process group, though it is a concurrent activity rather than being top-down or bottom-up. In change role, agents like managers, 'develop measurements of the processes; communicate their results by identifying conflicts and inefficiencies in the personal processes'. Agents, in enactment role, are knowledge workers executing their own processes. An important class of agents -*process agents*- is responsible from maintenance and visualization of the process network. They work as change agents of the whole process or meta-processes. They envisage the process as a whole and explain it in visual, verbal or other forms, develop relevant measures, identify problems and suggesting improvements over the processes.

Demirors distinguishes HOC-A from personal software process (PSP) [73] that focuses on improving personal processes as part of a larger process improvement effort. The focus in HOC-A, on the other hand, is on the whole process and models are primarily used to understand and improve the total process.

HOC-A requires a disciplined guideline, a notation and a tool to be applied in organizations in a systematic, efficient and effective way. For the notation aspect, Demirors proposes the Horizontal Change Notation (HOC-N) [40].

Horizontal Change Notation (HOC-N) enables modeling of individual behavior in a decentralized way, while supporting inter-agent communication and integration of these individual process models. It is based on an approach essentially designed for the reactive specification of multiprocessor computing. It is a unique notation that enables behavioral representation of agents' activities performed in parallel.

Agents are capable of performing some set of activities (*actions*) in an *environment.* They can store two types of state information in their memories: global and local. They can store a copy of the state of the environment, which is a set of global objects with their values, and can store local objects with their values which are not accessible from outside the agent. Actions are performed in the memories of the agents. Agents can also perform loading copies of a state of the environment into agent's memory, and discharging copies from memory to objects in the environment.

A reactive specification is represented by:

$$(P, Q) \rightarrow [\delta; s; \beta]$$

P is the condition under which system reacts by action [δ; s; β] and the outcome is discharged if Q is satisfied. δ represents the loading list; s represent the actions and β represents the discharging list. Let α be the current state of the environment and α` be the state after s is executed, then the process of the agent can be specified as follows:

```
While P(α) is not satisfied do nothing;
Load the listed part of α into the local memory;
Perform 's' in local memory;
If Q(α`) is satisfied
  Discharge β into the global memory;
```

Due to the inappropriateness of some of the assumptions of multiprocessor computing, specific extensions are devised in order above idea to be used in the context of HOC-A. Extensions include the notation to facilitate agent specifications, inter-agent communication, and differentiation between local and global parameters. A detailed description of the notation can be found in [40].

Following example is an abstract from [39]:

The example is based on the scenario in ISPW-6 [90], where two software engineers (SE1, SE2) a project manager (PM), a design engineer (DE), a quality assurance engineer (QE) and a configuration control board (CCB) are participating to implement a requirement change. Figure 3 depicts partial definitions of the roles defined by the agents CCB and SE1.

```
CCB:<

{received(request.Module, S1) Λ ~inuse(Module,_) Λ ~sent(Module, S1)
; ~received(request.Module.cancel,S1)}
→ [ send(Module, S1); sent(Module, S1) Λ inuse(Module, S1) ]

{received(request.Module, S1) Λ inuse(Module, S2) Λ ~sent("Module in use", S1)
; ~received(released.Module, S2)}
→ [ send("Module in use", S1); sent("Module in use", S1) Λ conflict=conflict
+1]

…

SE1:<

{received(SE1_task, PM) Λ specs_reviewed Λ received(Module.c, CCB)
; ~task canceled}
→ [receive(Module.c, CCB); received(Module.c, CCB)]

…
```

Figure 3: Partial Individual Process Models in HOC-N

The first activity specified by CCB, send (Module, S1), will be executed when CCB receives a request pertaining to Module from S1, Module should not be in use by any other agent, and Module has not already been sent. After the module is sent the variables sent(Module, S1) and inuse(Module, S1) will be true if the request is not canceled by S1. In the second activity, CCB specifies send("Module in use", S1), when Module is in use by another agent.

The activity specified by SE1, receive(Module.c, CCB), is executed when the task is received from PM and specifications reviewed about the task and Module.c has not already been received. After Module.c is received, received(Module.c, CCB) becomes true if the task is not canceled.

Algorithms are developed to automate the visualization of total process coded in horizontal change notation and to enable generation of communication flow diagrams, activity dependency diagrams and completeness checks. A partial communication flow diagram for the agent SE1 is given in Figure 4.

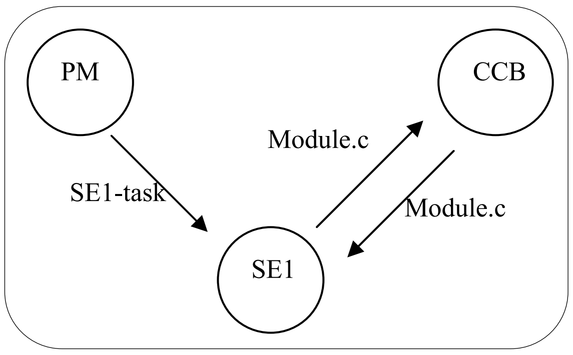


Figure 4: A partial communication flow diagram

HOC-N, as a modeling notation, answers some of the critical requirements of HOC-A. It is suitable for descriptive, decentralized and concurrent modeling. However, it is weak in representing the organizational and informational perspectives of the processes (as we discuss in section 3.6.1). Since it is a text based notation, it is also weak in visual representation and its formality poses difficulties for agents in learning notation's semantics and using it to express their process behavior. In addition, HOC-N requires agents to identify and cover all the states that can be encountered and to define unique actions for each state. Analyzing all combination of states and related actions is particularly a great challenge in the context of an organization, together with the assumption that an organization has finite and observable number of states.

In order for organizations to apply HOC-A and exploit its advantages, there are key enablers that should address unique requirements of the approach. Although HOC-N answers some of these key requirements, the lack of a method and the tool support forbid the approach to be followed by organizations in an efficient way.

2.2. View-Based Approaches

View-based approaches (or 'ViewPoints') [54], [111], [112] are employed in requirements engineering and process elicitation fields to describe a complex phenomena as a union of different perspectives (views) hold by different stakeholders. The construction of a complex model involves many agents (participants or actors) that have different views of the artifact or system they are trying to model (the domain of discourse). Since these agents are assigned different roles or responsibilities, their perspectives are generally partial or incomplete. The combination of the agent and the view that the agent holds is termed a viewpoint [56].

Although achieved centrally, the aim is to acquire portions of the processes and to merge them to form the complete model. In this sense, it shares common points particularly in terms of the notations and the tools that can be utilized for decentralized modeling. This section goes through the most influential view-based approaches, discusses and compares the methods, notations and tools utilized in these approaches and their applicability to decentralized modeling.

2.2.1. Controlled Requirements Expression (CORE)

CORE [108] method is one of the earliest requirements analysis and specification method that provides prescriptive guidelines on specifying and analyzing system requirements based on viewpoints. The phases of the method comprise the definition of the problem, viewpoint identification and gathering, and documenting information about viewpoints. Each viewpoint is described with a tabular collection diagram, where the sources of inputs and the destination of outputs to each action performed by each viewpoint are identified and inconsistencies are identified based on these interactions. The support of tabular collections diagrams of behavioral characteristics of the processes is limited.

2.2.2. Process Viewpoints

Sommerville et.al. [130], [131] propose a view-based approach that stresses on the utilization of 'views' on process elicitation and improvement. The method utilized for process elicitation involves the identification and definition of the viewpoints and questions to be used for their elicitation and potential process improvements. The research does not suggest and make use of a tool for the method they propose for process elicitation. Thus, activities are performed manually by process agents. A process engineer works with related participants to 'elicit' process descriptions. Then process agents identify overlapping parts and inconsistencies among these views manually. Since it is manual, they describe processes with any notation that the process engineer fell most suitable to reflect the perspective with. These separate views are not merged. Sommerville argue that even if it were possible to integrate all the separate views, the resulting single model would be so complex that it would be completely incomprehensible. The absence of tool support is not because the work has a narrow scope but it is related with the method they propose. The method essentially is too unrestrictive for viewpoint analysis that it can be supported by a tool. Using any type of notation for process descriptions of separate views and finding similarities and inconsistencies among them cannot be supported by a tool developed with today's technology.

2.2.3. Multi-View Process Modeling (MVP)

Verlage [145], [146] is one of the first that introduced a formalization of core requirements of view based process elicitation. MVP calls 'view' as the part of a software process model which corresponds to a role (i.e., supports the role's tasks). Views may overlap and have to be integrated to produce a comprehensive software process model. The MVP concept is analogous to HOC-A [40] in terms of how organizational processes should be developed and maintained. Verlage proposes the following three steps, as a method, to be applied for the development and use of comprehensive software process models:

1. In the first step, every participating role models its own view, which is more likely to represent the real world than a description made by other people.

2. In the second step, similarities and inconsistencies between views are detected and these separately developed views are integrated into a comprehensive software process model.
3. In the final step, every role uses its own view during project guidance. The comprehensive software process model is not presented as a whole, and unnecessary information is altered. The process is presented using the role's own terms defined in the corresponding view.

Processes are defined with MVP-L [23] (multi-view process modeling language), a formal domain language for representation of software development processes. In essence, MVP-L is used to capture process knowledge from a single viewpoint. It is rule-based, i.e. control flow among processes is expressed by using pre- and post-conditions (called entry and exit criteria). An excerpt of an MVP-L process model is shown in Figure 5 [146].

```
process_model E14_Write_Specifications () is
-- The Write Specification process is where the Specification team
   writes the first five volumes of the six volume Specifications. Those five
   volumes are: the Mission Volume, the User's Reference Manual Volume,
   the Functional Specification Volume, the Functional Specification
   Verification Volume, and the Usage Profil Volume.

process_interface
...

  exports
   effort : Process_effort := 0;

  product_flow
    consume
      externRe: External_References_Container;
      prorep: Project_Reports_Container;
      worpap: Working_Papers_Container;
      ...

    produce
    consume_produce
      mv: Mission_Volume;
      fs: Functional_Spec;
      ...

  entry_exit_criteria
    local_entry_criteria
      (index1.status = 'complete' and worpap.status = 'complete' and ...;
    local_exit_criteria
      (mv.status = 'complete' and fs.status = 'complete' ) and ...;

end process_interface
```

Figure 5: MVP-L Process Interface (incomplete)

Similarity analysis (identifying overlapping portions in process models) is based on the semantics in the MVP-L. Views are subject to a tentative set of consistency rules to detect inconsistencies between two views. The choice of the rules to be applied depends on the degree of overlap between the two views. The differences in the abstraction hierarchies are not resolved: each hierarchy is kept separately. MVP-L supports the representation of behavioral perspective of the processes, while the representation of functional and organizational perspectives is limited and the support for the informational perspective is weak. Due to its formal and textual structure, it is hard to be managed by the end users [16]. An environment, MVP-E [15] (multi-view process modeling environment) for a full implementation and validation of the approach is still pending.

2.2.4. V-Elicit Environment

Turgeon and Madhavji [142] have proposed a prototype tool called V-Elicit that helps to elicit process models from multiple sources or views. Process elicitation comprises; gathering process information from the agents participating in a process, from documents, and through observation; modeling this information; and verifying that the model built is consistent and complete [143]. The primary concern of process engineers is the crosschecking different representations of the process that are elicited from same source.

V-Elicit can elicit domain-specific process information that is defined by the role of an agent (requirement engineer, designer, and etc.) in the process and the aspects (data-flow, control-flow, and etc.) of interest associated with the role. Processes are described and maintained with a text-based notation and integrated models are visualized for analysis.

Works by Verlage and Turgeon & Madhavji are in parallel in many points. Both share common points in their method and the requirements they define for multi view process modeling. However, Turgeon & Madhavji's work on multi view process elicitation is one of the most complete approaches with an available tool support. V-elicit tool responses a larger set of requirements. However, there were still topics, such as common element identification that require further enhancements.

2.2.5. Discussions on View-Based Approaches

Although the methods, notations and tools utilized in view-based approaches answers many of the requirements of decentralized modeling (in particular; modeling with roles, inconsistency checking between individual descriptions, integration of definitions), there are critical attributes that forbid them to be utilized effectively and efficiently for decentralized and concurrent process modeling done by process owners. This is because these approaches implicitly or explicitly assume process modeling to be performed centrally by a group of people that controls the modeling process as a whole. This results innate characteristics that are critical for the applications of decentralized modeling by process owners. For example, views are outdated once they are merged into a complete model and are not maintained as separate entities from then on. Any change in the process is represented in the integrated model, which constrain view-based methods from being utilized as an effective technique for decentralized process maintenance.

One of the significant characteristics of individual definitions (views) that decentralized process modeling approach seeks is the mutual exclusiveness of the definitions. That is, once complete and consistent, each individual definition describes a separate portion of the process (in terms of the activities performed by a role) which does not overlap with other definitions. In view-based approaches, on the contrary, the aim is to describe the process from as many views as possible. Since there are theoretically infinite numbers of views, overlaps between these views are inevitable in practice; because views represent the same process from different perspectives. These overlaps should be detected and handled before integration occurs.

In order to come up with a complete process model, view-based approaches try to capture different perspectives with different representation schemes. The usual method is, first, to identify the overlapping portions of the descriptions, and then to resolve inconsistencies while merging the partial descriptions [111], [143]. In addition, these activities are even harder to perform if descriptions are done with different process modeling notations or by using different aspects. The techniques and tools for performing these activities (identifying common elements and inconsistencies) suggested by related studies commonly need further improvements [111], [143] or they are not implemented at all [131], [145]. In almost all methods, inconsistency checking between views is performed in batches once they are complete. This significantly degrades the modeling efficiency. A more effective way is to check inconsistency all the way through the process definition performed concurrently by all agents.

2.3. Process-Centered Software Engineering Environments and PMLs

Software process programming primarily evolved with Osterweil‘s [114] argument maintaining that “software processes are software too”. Starting from the notion that every organization and even a project requires its unique processes, Osterweil brought the idea that the organizations must define their processes and must tailor it for its specific project. The process model should take into account all the particularities of the organization and product being developed. He suggested that the notion of a ‘process program’ should become a key focus of software engineering research and practice and his perception has started research areas on process modeling languages (PMLs) and on the development of Process-Centered Software Engineering Environments (PSEEs).

PSEEs are systems that support large scale software development by providing mechanisms and notations (PMLs) for explicitly and precisely modeling the process of development and maintenance of software, and mechanisms for enacting the modeled process. PSEE provides a variety of services, such as assistance for software developers, automation of routine tasks, invocation and control of software development tools, and enforcement of mandatory rules and practices [4], [58]. By the end of ‘80s and in early ‘90s, software engineering community has developed a number of PSEEs such as; Adele [17], [18], ALF [27], APEL [37], EPOS [30], [31], HFSP [87], Marvel [84], Melmac [53], Merlin [83], PADM [24], Process Weaver [97], SPADE [12], [11].

The degree of enforcement and guidance that is provided to the user is an important issue in PSEEs. Based on the degree, the following four levels can be distinguished [48] [63]:

- Passive role, where the PSEE operates only on user's request. PSEEs of this type generally keep track of the actions being carried out for process monitoring.
- Active guidance, where the PSEE guides the process and prompts the users for certain activities when necessary. The system inspects and controls the outcomes of process enactment.
- Enforcement, where the user is forced to act as specified by the process model. The enforcement can be loose, where some of the activities are enforced; or strict where the system controls every activity being enacted.
- Automation, where the system executes the activities with no user intervention.

Many of the PSEEs adopt more than a single form of user support (eg. SPADE, Marvel). In general, for activities that do not require user intervention, PSEE automate the task; whereas, for other activities the enforcement approach is adopted.

PSEEs are generally characterized by their process modeling languages (PMLs). The software processes are represented with a PML and the process model is analyzed and enacted by the environment. Model analysis involves the investigation of the presence or absence of certain properties, such as circularity, consistency or redundancy. Enactment is the execution of the processes by humans or computerized tools. Process analysis and enactment being the focus, the PMLs for PSEEs are typically formal languages.

There are many attempts to classify PML paradigms [4], [36], [102]. Ambriola [4] classifies PMLs according to the phase of the process lifecycle they support and the level of abstraction they provide:

- Process Specification Languages (PSL) are used in the requirement specification and assessment phases.
- Process Design Languages (PDL) offer features that are useful in the design phase.
- Process Implementation Languages (PIL) are used to support the implementation and monitoring phases.

Curtis et.al. [36] classifies the approaches to represent process information within PSEEs into the following categories:

- Programming models
- Petri net models
- Functional models
- Plan-based models
- Quantitative models

In below paragraphs, we elaborate some of the most significant PSEEs and PMLs based on these categories.

a. Programming models (process programming language based)

Process programming languages [101] extend existing conventional programming languages to represent concepts related to software processes. APPL/A [133], [134], developed by Osterweil et.al. is one of the best known examples of PMLs belonging to this class. These languages are used for constructing process programs that can be enacted by a machine. Emphasis on the executability of process programs is a particularly important attribute of the approach. The purpose is on automating or otherwise automatically supporting the execution of the process. APPL/A is an extension to Ada, extended with persistency, relation management, transactions, and triggers. On the contrary of what is generally the case for process programming, APPL/A supports for multiple representational paradigms. It is procedural, but supports events and triggers and data modeling. JIL [135], [152] is a successor of APPL/A with enhancements for representing processes in higher levels of abstraction. Another programming model for process modeling is the specific language (PADM-PML) used in the Process Analysis and Design Methodology (PADM) PSEE environment [24], which is influenced by object-oriented languages, reflexive systems, and role-interaction theory.

b. Petri net models (graph/net)

With significant extensions, Petri nets are highly utilized in workflow management [147], [148] and in software engineering field (SPADE’s SLANG [11], [12], Melmac’s Funsoft Nets [53], and Process Weaver [97]).

SLANG, the language for SPADE environment, is an enactable formal language based on Petri nets and object orientation utilized for the definition of the static structure. ‘FUNSOFT’ nets are high level Petri nets which have been particularly developed to support software process modeling. Their semantics is defined by Predicate/Transition nets. Being a Petri net based formalism, FUNSOFT nets represent non-determinism and parallelism. Special emphasis is given to simulation and validation to support the analysis of properties of the modeled process. They do not support full enactment.

Process Weaver [97] is based on an internal language, hidden to the user, and usable by a number of views. The environment supports activity-oriented process modeling, i.e., process steps, their relationships, and attributes are explicitly modeled in the environment. In Petri net formalism, the places describe resources and the transitions describe process steps. Process weaver supports the idea of presenting a process model from multiple graphical viewpoints. For example, it presents a task-oriented view for process enforcement and enactment, and it can present a process flow view and a hierarchical breakdown of activities.

c. Functional models

In functional models, software process is described as a collection of activities which are characterized by their input and output relationship. The process is defined as mathematical functions representing relationships between inputs and outputs. If the relationship is not simple enough, activities are decomposed into sub-activities together with the definitions of their input and output unless the process steps can be mapped to external tool invocation or manual operation. HFSP (Hierarchical and Functional Software Process) [87] is an example of a functional model. The focus is mainly on representing functional and behavioral perspectives.

d. Plan-based models

Plan-based models emerged from the planning paradigm in artificial intelligence research [70]. GRAPPLE [70], [71] is an example of this type. It models the processes as set of goals, sub-goals, preconditions, constraints and effects. A 'plan' is a hierarchical and partial order of operators that achieves a goal. Processes are formally defined by operators and plans can be viewed as the data structures that represent instantiations of processes. Each operator has core-clauses: a precondition that defines the state that must realize in order for the action to be performed, and a set of effects (can be considered as post-conditions) that are the state changes resulting from performing the action. Goal clause that defines the principal effects of an action, and a constraints clause that defines restrictions on parameter values enhance the core-clauses. Below is an example of an operator:

```
OPERATOR   system-check-in
GOAL:      archived(System)
PRECONDS:  unit-tested(System)
CONSTR:    part-of(X,System)
SUBGOALS:  controlled(X) ITERATED
      UNTIL all-controlled(System)
EFFECTS:   ADD archived(System)
```

e. Quantitative models

There are only few examples of quantitative models in process modeling research. System dynamics approach [1] applies feedback and control system techniques to social and industrial domains. Abdel-Hamid et.al. [1] have used this technique to provide a quantitative representation of many behavioral observations of software projects. However, their reliability is very much dependent on the accuracy of the values of the constructs and parameters they observed on real processes [36].

f. Rule-based models

Rule-based models are analogous to plan based models in terms of the clauses utilized in specifying rules, which also have their origin in the artificial intelligence field [119]. Rules usually specify an event causing their activation, a guarding condition and an action to be taken in case the condition is true. Popular rule format is using pre and post-conditions that surrounds an action. Hence, rule-based models generally reflect behavioral view of the processes i.e. they try to reflect the activities to be performed under certain conditions.

Marvel [84] is an example that employs rule-based approach by its modeling language MLS (Marvel Strategy Language). In MARVEL, a collection of strategies are combined to define the structure and behavior of the programming environment. These strategies are written in advance by a super-user familiar with Marvel environment. Each strategy comprises an objectbase description, tool descriptions, and/or rules that model the software development process. The 'objectbase' description defines the structure of a project database (source code, documentation, and etc.) in terms of classes of objects. A tool description consists of a 'tool envelope', which defines the relationship tools for invocation. A rule consists of three parts. The first part is the *pre-condition* that must be true before a software activity can be performed. The second part is the *activity* or the task to be performed when precondition is satisfied. It comprises a tool name, an operation name, and any arguments to the operation. The last part is a set of *post-conditions*. One post-condition stands for a successful termination and the other post-conditions represent a failed termination due to several possible kinds of faults. Exactly one post-condition becomes true after the activity is terminated.

Another example of a PSEE that utilizes rule-based approach is ALF [27]. ALF's modeling language (MASP/DL - Model for Assisted Software Process Description Language [45]) consists of five complementary sub-languages. It comprises 'object model' to describe the data; an 'operator type' sub-language for tool specification; a 'rule based' sub-language for expressing the reaction to some predefined events; a 'constraint' sub-language to define operator invocation order with respect to precedence rules; and a first order logic language to define 'characteristics'. Characteristics are expression which has to be true and is used as invariant and/or objective. Below is a simple example represented by the rule model description language in MASP/DL [27]:

```
IF ON EXIT OPERATOR compile(_x)
EVALUATE successful_compilation(_x)
EVALUATED TRUE
THEN link( _x, /lib/maths.l)
```

2.3.2. Decentralization via PSEEs

Most of the existing PSEEs mentioned above assume that the processes are defined by central unit and they are executed by a central server which also monitor and records execution and invoke necessary tools. Usually, each of these PSEEs uses its peculiar process modeling language. With the increasing necessity of providing support for physically distributed teams or groups possibly belonging to different and geographically distributed organizations, there have been studies on distributed process enactment not only in the software engineering field [35] but also in workflow management field [109], [59], [60]. However, to support cooperation of different teams requires the adoption of new development paradigms rather than standard client-server architectures. These autonomous teams should be able to define their own procedures and use their own set of tools and data. For the development of products, they should collaborate to share tools, exchange data and decide on some common policies or procedures for the part of the work that involves collaboration. For this decentralized view (as opposed to 'distributed'), Ben-Shaul and Kaiser [20] developed the Oz environment.

Oz is developed as an enhancement to the Marvel [84] and adopted the view of 'international alliance' whereby independent countries sign 'treaties' that determine their collaboration but retain full control over their local laws. The sites that involves in the interaction explicitly specify in what ways they are willing to participate in a multi-site operation, and the specifications are loaded into each site's local PSEE to establish all that is needed to enable those interactions.

Oz's focus is on interconnecting environments that are independent and loosely coupled, both physically and logically [21]. This means that a SubEnv should be able to behave as a complete environment by itself when not collaborating with any other SubEnvs, and SubEnvs must be able to operate concurrently and independently, except when their processes explicitly collaborate. For the collaboration, a common sub-process should be defined. This is obtained by developing and signing one or more treaties that define a common sub-process, a common sub-schema for accessing data, and a set of access constraints. For the execution of these common processes, the summit model is utilized during which two SubEnvs cooperate by executing a common process previously defined by means of a treaty. The treaty definition by negotiation and summit model allow each SubEnv to collaborate together while behaving independently and autonomously.

In principal, Oz uses Marvel's rule-based PML (MSL) [19] for representing processes. A rule represents a process step consisting of an optional action (activity) with its pre-condition and post-condition. If the precondition is satisfied, the process step corresponding to a rule is enacted. Completion of the action leads to asserting the post-condition.

Oz's unique approach that addresses the problem of decentralized development between teams of developers has issues relating to its scalability in organizations for decentralization among individual participants. Since agents (and roles) are highly interdependent between each other in terms of the information they share and provide, Oz's applicability regarding to whether it can be scaled down to the agent or role level is questionable. Since SubEnvs are highly autonomous and independent and the aim is to maximize the locality of them, the combination of processes performed by all sub-environments does not necessarily constitute the entire process model of the organization. The focus is solely on activities performed in cooperation. Hence there is no motive for integrating the process models of sub-environments.

Another significant work on decentralization of software development processes is based on the viewpoints framework [54] (section 2.2). Leonhardt et al. [98] proposes an approach addressing the issues on decentralized and concurrent software engineering as an alternative to traditional centralized software development. They considered the use of decentralized process models to drive consistency checking and conflict resolution. Individual process models are represented locally by associating development participants with them. The process models use pattern matching on individual's development histories to determine the particular state of the development process, and utilize rules to trigger situation dependent assistance to the user.

The environment they developed for the enactment of the process models maintains an ordered work record of development actions for each process. This record also defines a particular sequence of process states. These states are used to identify suitable courses of actions. A specific course of actions is appropriate for a set of similar states, which is called a situation. Rules map situations to actions by:

```
<situation><course of action>
```

A <situation> is the pre-condition of the rule, and a <course of action> defines what should be done once a decision has been made.

During enactment, one of three kinds of responses is released. 'Informal guidance' may include help text, video clips, and etc. Specific recommendations may include a limited set of actions that a developer is advised to select. 'Automatic execution' includes running specific actions.

The critical characteristic of this approach is the decomposition of complex global models into simpler local ones for decentralized enactment. Hence, it is yet not clear how and by whom the local process models are developed [98]. This view is contradictory to our bottom-up approach for generating the individual definitions.

2.3.3. Discussion on PSEEs and PMLs

Despite some successful examples [33], PSEEs did not gained general acceptance or widespread use. This issue is mostly attributed to their prescriptive approach that pushes the automation for enforcement [34]. PSEEs have been developed and used as a mean to impose good practices and uniform behaviors, as a guide for directing people through the right path and preventing them from making mistakes. This view is one of the most important factors that brought about this prescriptive approach adopted by most PSEEs, which also become the main reason for their failures [34]. In general, the emphasis in PSEEs and their PMLs is on controlling, monitoring and enactment as opposed to understanding, learning and improvement. Hence, issues such as (continuous) process improvement for organizations is not addressed adequately [58].

In the overall, the goal of the processes in knowledge intensive organizations is to provide support for their creative tasks rather than constraining them to follow a predefined pattern of activities. The complexity and flexibility is intrinsic in such processes and these processes cannot be prescribed to knowledge workers, since the knowledge of what to do, how to do, and when to do certain activities belongs to the knowledge worker.

2.4. Business Process Management

The term 'business process' has been the common concept in many fields such as, business process management (BPM), business process reengineering (BPR), business process improvement (BPI), workflow management (WfM), enterprise modeling (EM), and process innovation. Much of the literature in these fields would suggest organizations to focus in business processes and implement process orientated structures in order to be more responsive to an increasingly changing competitive environment. Davenport [38] defines a (business) process as:

> A structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action. Taking a process approach implies adopting the customer's point of view. Processes are the structure by which an organization does what is necessary to produce value for its customers.

Davenport's definition views clear boundaries to the business processes where it has clear inputs and outputs; consists of activities, which are ordered in time and space; and should provide a value to the customer. Hammer & Champy [66] summarizes the definition of the business process as: "a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer." Another precise view to the processes defines it as: "a set of partially ordered steps or activities intended to reach a common goal' [36], [99]. Alter's [3] definition emphasizes on the customer: "a related group of steps or activities that use people, information, and other resources to create value for internal or external customers." It consists of steps related in time and place, have a beginning and end, and has inputs and outputs. Laudon et al. [96] points out the peculiarity of business processes and view processes as the unique ways in which organizations coordinate and organize work activities, information, and knowledge to produce a valuable product or service.

Keen [89] argues that the workflow view of processes – with clearly definable inputs and outputs and discrete tasks that follow and depend on one another in a clear succession – is limiting. In organizations there are many processes that have no clear inputs, flows, and outputs such as those governing management succession, manager-staff-relations, management development, and incentives and promotions. This restrictive view of processes can cause management to ignore these “soft” processes, which might need direct focus for improvement. Keen defines a process as “any work that meets the following four criteria: it is recurrent; it affects some aspect of organizational capabilities; it can be accomplished in different ways that make a difference to the contribution it generates in terms of cost, value, service, or quality; and it involves coordination.”

In 90s’, the automation of business processes with workflow management systems (WfMS) has attracted attentions of many organizations. In many aspects, WfMSs are analogous to PSEEs utilized in software engineering organizations in 80s’ (section 2.3). They extend the notion from software processes to a more generic view of business processes.

The Workflow Management Coalition (WfMC) defines workflow as: “The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.” [155]. Workflow is enacted with a workflow management systems, which is defined as: “A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications [155] ”. The emphasis on the enactment is evident in these definitions. In 2000’s, this traditional restrictive view on the business processes was criticized by many researchers and practitioners, which led to the terms such as ‘business process management (BPM)’, ‘enterprise modeling (EM)’ to emerge. BPM is using methods, techniques, and tools to support the design, enactment, control, and analysis of operational business processes that involves humans, organizations, applications, documents and other sources of information [149]. The term is, in effect, emerged as an extension to the classical workflow management systems and approaches. Other definitions for BPM stress on the improvement perspective and view it as an approach to improving organization's business processes [52]. This view considers BPM as a set of activities that seek to make business processes more effective, more efficient, and more capable of adapting to an ever-changing environment.

There are several industry initiatives, standardization bodies and organizations working on BPM concepts and standardization, including BPMI (Business Process Management Initiative), WfMC (Workflow Management Committee), OASIS (Organization for the Advancement of Structured Information Standards) Committees, Rossetta Net, W3C (World Wide Web Consortium), and OMG (Object Management Group), Microsoft’s BizTalk and etc. For example, BPMI is devoted to the development of open specifications for the management of e-Business processes and defines open specifications such as the Business Process Modeling Language (BPML) [167] and the Business Process Query Language (BPQL). Similarly, OASIS drives the development, convergence, and adoption of e-business standards including BPEL (Business Process Execution Language), ebXML (Electronic Business using eXtensible Markup Language).

2.5. Enterprise Modeling Frameworks

An approach that extends and covers the modeling from processes to other attributes of the organization is the *enterprise modeling* (EM). EM is representing the enterprise in term of its organization and operations, i.e. processes, behavior, activities, information, objects, and material flows, resources and organization units, system infrastructure and architecture [8]. The aim is to 'externalize' the enterprise knowledge to share it by business applications and users in order to improve the performance of the enterprise.

Explicit representation of business processes being common to BPM and EM approaches, they use analogous or common methods and notations for business process modeling. However, notations utilized for enterprise modeling usually allow building the model of an enterprise according to various points of view in addition to the process. Function, network, economic, architecture and etc. views are also represented in an integrated way.

Defining an enterprise requires following a methodological and systematic approaches, which are known as enterprise frameworks and architectures. A framework as a fundamental structure which allows defining the main sets of concepts to model and to build an enterprise [8]. Some of the frameworks are designed for integrating enterprise modeling (Zachman, CIMOSA, and etc.) and others for developing and integrating enterprise applications (ARIS, and etc.).

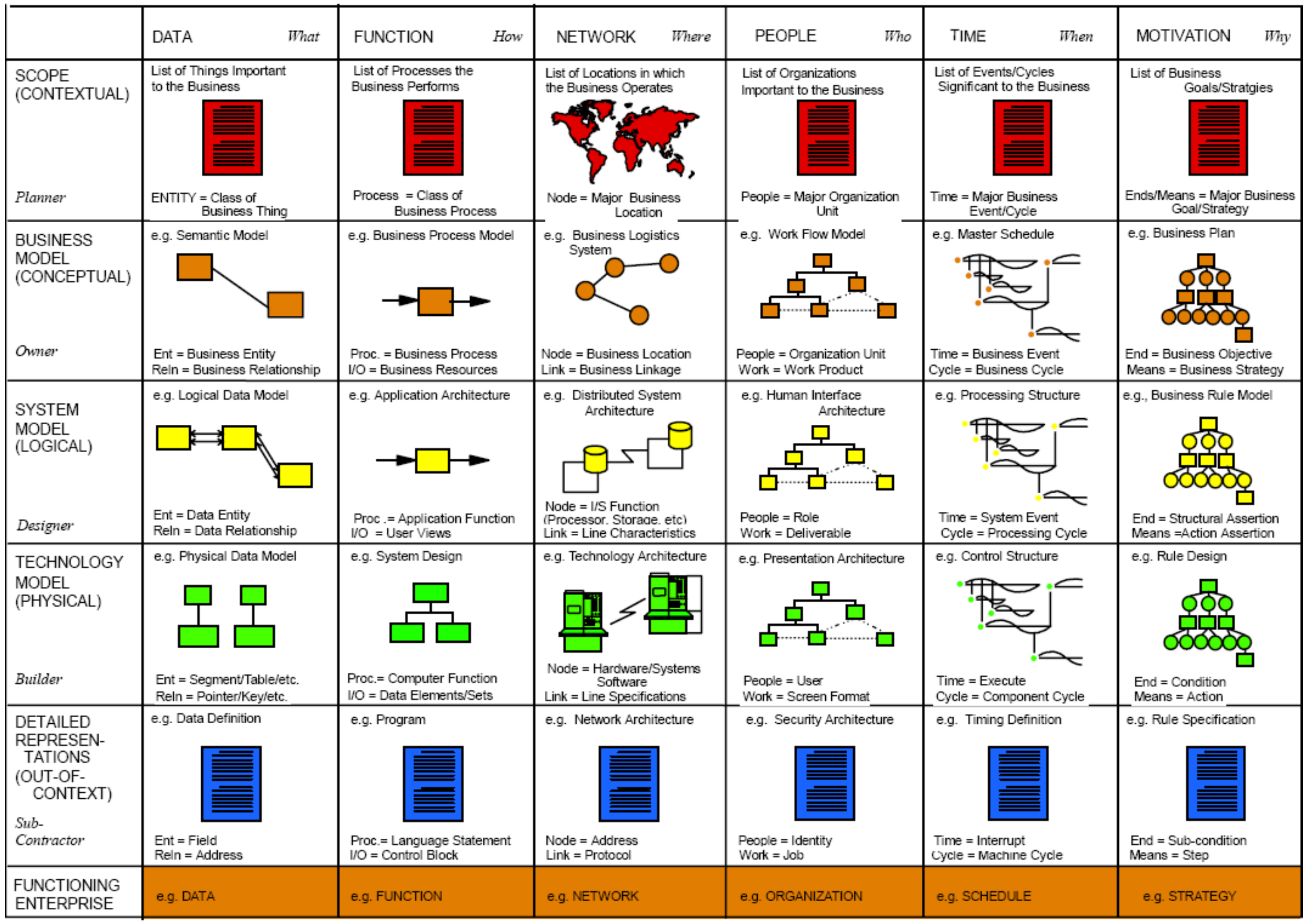
The following frameworks are some of the most significant and influential ones that are being pursued by related organizations and industry: The Zachman framework [161] developed and maintained by the Zachman Institute for Architecture has been used in industry and governmental organizations as EM approach and as a reference categorization structure for enterprise knowledge repositories. The GERAM (Generalized Enterprise Reference Architecture and Methodology) framework [74] from The University of Brisbane integrates different EM modeling aspects and domains. The GRAI (Graphs with Results and Actions Inter-related) framework [144] developed by the GRAI Lab and Graisoft is known by its strong support for performance indicator management and supporting decision making. ARIS (Architecture of Integrated Information Systems) by IDS Scheer [74] has strong top-down process modeling and integration capabilities. The DoDAF (U.S. Department of Defense Architecture Framework) (formerly C4ISR - Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) is a comprehensive framework and methodology targeted specifically at systems engineering in the military systems [46]. The CIMOSA (Computer Integrated Manufacturing Open System Architecture) framework [28] and TOGAF (The Open Group Architecture Framework) from the Open Group [136] are other important frameworks that are utilized for EM. There is also an ISO standard (ISO 15704 [75]) that places the concepts used in methodologies and reference architectures such as ARIS, CIMOSA, GRAI, and etc. within an encompassing conceptual framework that allows the coverage and completeness of any such approach to be assessed. The following paragraphs provide a brief outline to two of these frameworks.

2.5.1. The Zachman Framework for Enterprise Architecture

The Zachman framework [161] uses a matrix based around the six basic communication interrogatives and six model types which relate to stakeholder groups to give a holistic view of the enterprise which is being modeled. The horizontal dimension defines the enterprise in what, how, where, who, when, and why, which can also be defined respectively as data, function, network, people, time, and motivation. The vertical dimension is presented in six rows covering the roles played by different actors within the enterprise: the planner, owner, designer, builder, subcontractor, and worker. These actors have corresponding perspectives that are defined as scope, business model, system model, technology model, detailed representations, and functioning enterprise.

Each cell in the matrix comprises a specific enterprise artifact. Zachman suggests that a fully architected enterprise would have an explicit representation that describes the enterprise's current and future activities related to that cell. He also maintains that all models in adjacent horizontal and vertical cells should be consistent with the artifacts in the cell. Proposition is that a fully described enterprise with the framework would ideally have complete alignment of its business mission to its systems implementation, and would be completely efficient in its application of resources, priorities, and processes.

The Zachman framework has been applied in many corporations such as General Motors, Bank of America, and Health Canada, and etc. and also spawned a number of other similar frameworks for applying enterprise architecture in specific domains. These include The Open Group Architecture Framework (TOGAF), and the Department of Defense Architecture Framework (DoDAF).



	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business; ENTITY = Class of Business Thing	List of Processes the Business Performs; Process = Class of Business Process	List of Locations in which the Business Operates; Node = Major Business Location	List of Organizations Important to the Business; People = Major Organization Unit	List of Events/Cycles Significant to the Business; Time = Major Business Event/Cycle	List of Business Goals/Stratgies; Ends/Means = Major Business Goal/Strategy
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model; Ent = Business Entity, Reln = Business Relationship	e.g. Business Process Model; Proc. = Business Process, I/O = Business Resources	e.g. Business Logistics System; Node = Business Location, Link = Business Linkage	e.g. Work Flow Model; People = Organization Unit, Work = Work Product	e.g. Master Schedule; Time = Business Event, Cycle = Business Cycle	e.g. Business Plan; End = Business Objective, Means = Business Strategy
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model; Ent = Data Entity, Reln = Data Relationship	e.g. Application Architecture; Proc .= Application Function, I/O = User Views	e.g. Distributed System Architecture; Node = I/S Function (Processor, Storage, etc), Link = Line Characteristics	e.g. Human Interface Architecture; People = Role, Work = Deliverable	e.g. Processing Structure; Time = System Event, Cycle = Processing Cycle	e.g., Business Rule Model; End = Structural Assertion, Means =Action Assertion
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model; Ent = Segment/Table/etc., Reln = Pointer/Key/etc.	e.g. System Design; Proc.= Computer Function, I/O = Data Elements/Sets	e.g. Technology Architecture; Node = Hardware/Systems Software, Link = Line Specifications	e.g. Presentation Architecture; People = User, Work = Screen Format	e.g. Control Structure; Time = Execute, Cycle = Component Cycle	e.g. Rule Design; End = Condition, Means = Action
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition; Ent = Field, Reln = Address	e.g. Program; Proc.= Language Statement, I/O = Control Block	e.g. Network Architecture; Node = Address, Link = Protocol	e.g. Security Architecture; People = Identity, Work = Job	e.g. Timing Definition; Time = Interrupt, Cycle = Machine Cycle	e.g. Rule Specification; End = Sub-condition, Means = Step
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY

Figure 6. The Zachman Framework

2.5.2. ARIS Framework

Architecture of Integrated Information Systems (ARIS) utilizes views in representing the enterprise. The conceptual design of ARIS is a based on an integration concept which is derived from a holistic analysis of business processes [74]. The enterprise is perceived with five views; each is represented with one or more modeling notations [125]. Figure 7 gives ARIS House presenting these five views.

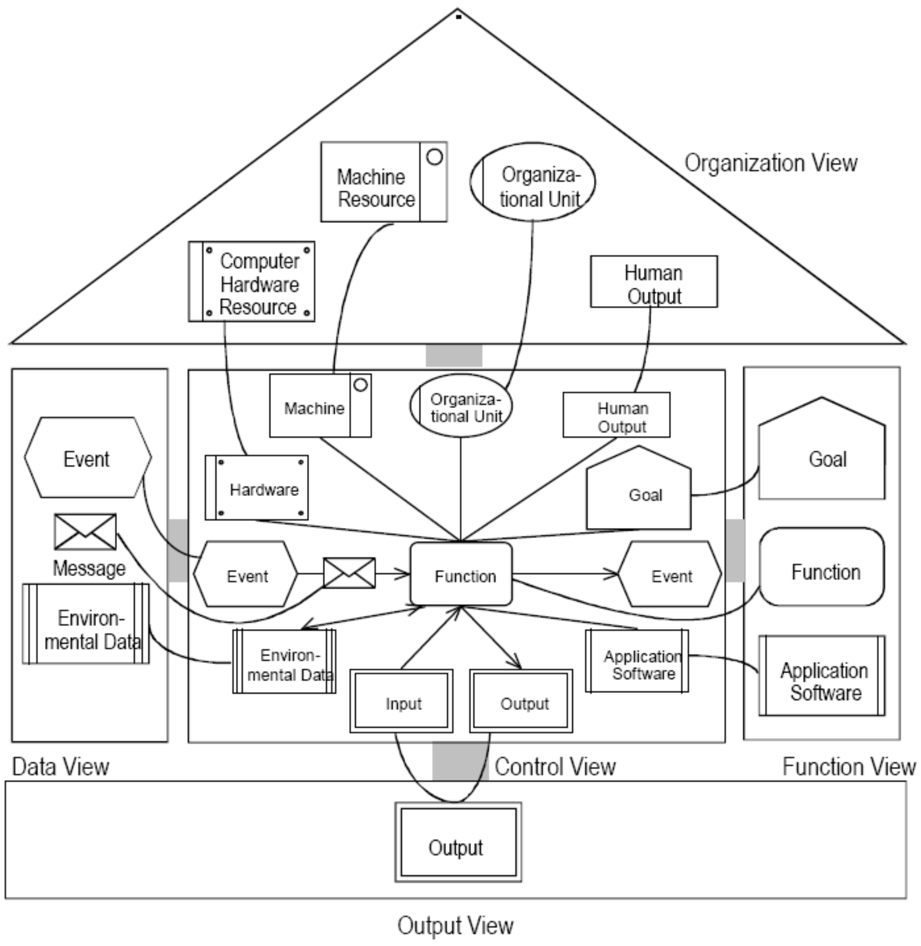


Figure 7. Views of the ARIS House

- *Function view* forms the functions to be performed (processes) and their interrelationships with each other. It contains the description of the function, the enumeration of the individual sub-functions that belong to the overall relationship and the positional relationships that exist between the functions. With different abstractions, the terms function, process and activity are used in the same meaning. Objectives are supported by functions and are controlled by them. In addition, applications software is closely aligned with functions, so, objectives and application software is also allocated to the function view.

- *Organization view* represents the relationships and structure of the entities executing the functions. Thus, a combination of the users and the organizational units as well as their relationships and structures are presented.
- *Data view* comprises relationships between data being processed as well as their states.
- *Output (Product/Service) View* represents the relationships between the products/services produced.
- *Control (Process) View* describes the relationship between other views via business processes.

ARIS is also structured in accordance with a lifecycle concept of an information system's descriptive levels. Lifecycle models in the form of level or phase concepts describe the lifecycle of an information system.

- Requirements definition
- Design specification
- Implementation

Thus, each view is mapped to the levels and is represented with appropriate models. Figure 8 shows the ARIS House concept with the descriptive levels.

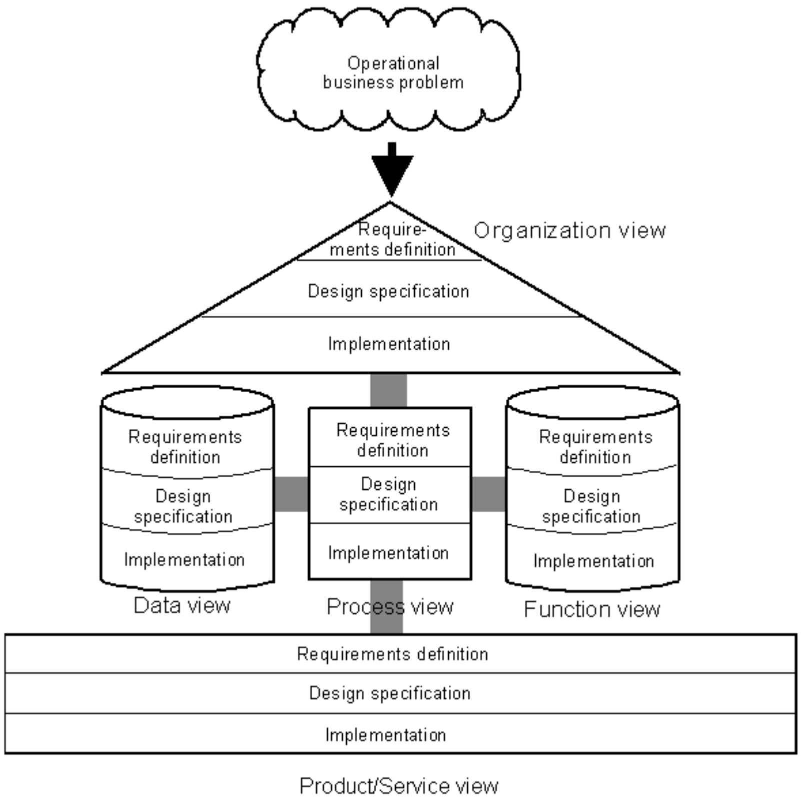


Figure 8. ARIS house with phase concept

Following the concept of a lifecycle model the various description methods for information systems ensures a description from business management-related problems all the way down to their technical implementation. In that respect, the ARIS is used as a framework for the development and optimization of integrated information systems as well as a description of their implementation.

There is an extensive tool support for the ARIS framework. The ARIS Collaborative Suite [126] is a commercial set composed of a variety of standalone tools supporting the framework. Its functionalities are later extended to provide support for many process modeling approaches with variety of process modeling notations [74], including BPMN (business process modeling notation) [25] and UML (unified modeling language) [113]. It also comprises extensions to support a number of other frameworks including Zachman and DoDAF.

2.6. Methods for Business Process Management and Enterprise Modeling

This section elaborates some of the related methods utilized for BPM and EM particularly focusing on the portions defining the procedures for business process modeling.

The view that stresses on the improvement perspective of BPM proposes a set of activities to be followed by the organizations. Goal setting for the enterprise is generally the first step towards BPM. The mission, vision and the goals of the enterprise are determined, critical success factors are formulated. Based on these factors, the processes of the enterprise are evaluated and a specific process is selected and described. Based on the description and its quantification, improvement opportunities are identified and implemented. The execution is monitored. For continuous improvement, the cycle of selection, description, quantification, improvement selection and implementation is repeated. The method is presented in Figure 9 [52].

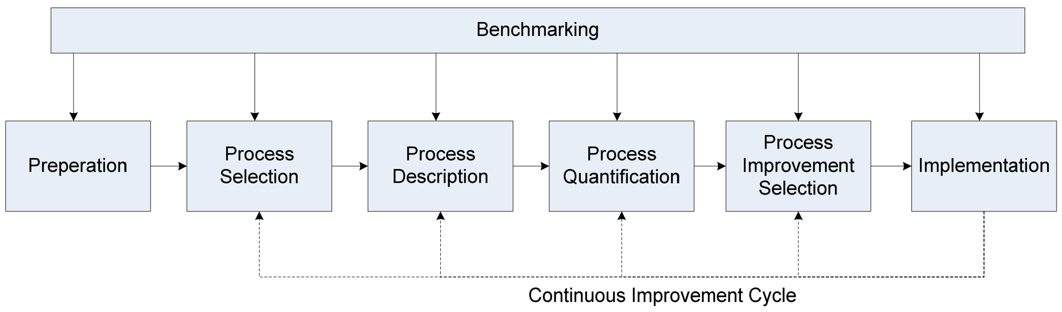


Figure 9. The Business Process Management Method

Process description is the phase where the business processes are generally modeled with a process modeling notation. Representing the business processes of an organization with process models is called *business process modeling*. A process model is an abstract description of a process that represents selected process elements that are considered important to the purpose of the model [36]. In usual, the selection of the process elements - the components of the process- depends on the process modeling notation utilized to define the model.

2.6.1. ARIS Modeling Procedure

Together with its framework, ARIS proposes a procedure to be followed for modeling, designing and implementing enterprise systems. ARIS procedural model follows the views and phases of the life cycle defined in ARIS framework. The procedure is depicted on Figure 10 [125] as an event-process chain (EPC) (section 2.7.2) diagram.

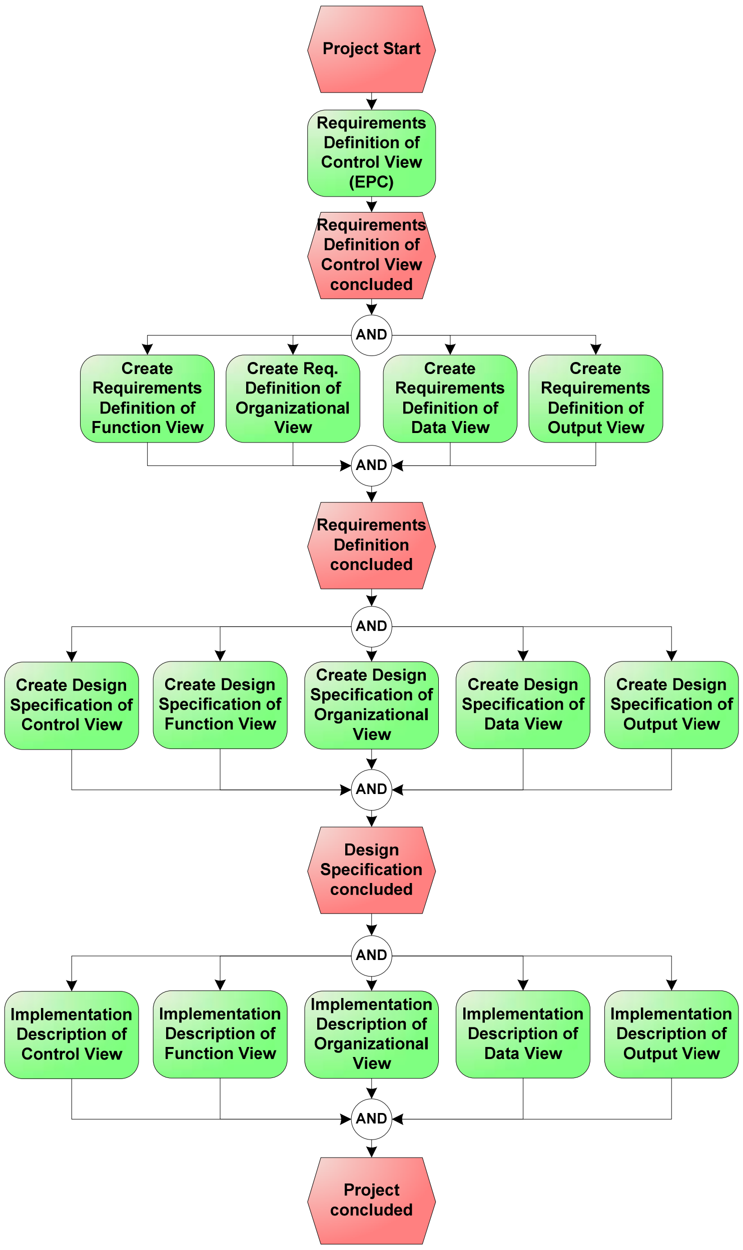

Project Start
Requirements Definition of Control View (EPC)
Requirements Definition of Control View concluded
AND
Create Requirements Definition of Function View
Create Req. Definition of Organizational View
Create Requirements Definition of Data View
Create Requirements Definition of Output View
AND
Requirements Definition concluded
AND
Create Design Specification of Control View
Create Design Specification of Function View
Create Design Specification of Organizational View
Create Design Specification of Data View
Create Design Specification of Output View
AND
Design Specification concluded
AND
Implementation Description of Control View
Implementation Description of Function View
Implementation Description of Organizational View
Implementation Description of Data View
Implementation Description of Output View
AND
Project concluded

Figure 10. ARIS Modeling Procedure

In particular, the activities in requirements phase of the procedure is associated with the business process modeling and related concepts. The requirements definition begins with the control view, i.e., a description of the business processes. In design and implementation phases, function, organization, data, output and control views are executed in parallel.

The method proposed by ARIS is a brief procedure that complements the framework; however, it assumes that each phase is developed by a specific central role, such as Requirements Definition Project Group for requirements definition, and pursues a complete centralization.

2.6.2. Riva Method for Business Process Modeling

Riva (formerly STRIM) is a method for the elicitation, modeling, analysis and design of organizational processes, proposed by Ould [118]. It utilizes role activity diagrams (RAD) (section 2.7.3) based on role modeling for representing processes. Activities are divided among roles; what an organization is trying to achieve with the process are the process goals; activities are designed to achieve these goals; activities need people within the groups to interact collaboratively. An important notion in Riva is the view that the organizational activity forms a network of related processes, rather than a hierarchy. The whole process is represented in a RAD rather than a number of models each representing a portion of the process. An example RAD is presented in Figure 15 (in section 2.7.3).

Role is the main structuring concept in Riva. The notion ranges from functional units through job titles to abstractions such as responsible engineer, customer, and approval of large claims. It can also include computer systems and applications, meetings, committees and teams, and rooms. It can be considered as an area of responsibility. Every activity should be performed within a role. The interactions between roles are any collaborative act involving two or more roles. Interactions generally take the form of approval, delegation, reporting, agreement, authorization, negotiation, questioning and informing.

Riva proposes the following key stages [61], particularly emphasizing the way to identify the process architecture of the organization:

- Brainstorm the subject matter of the organization or a part of the organization to identify its essential business entities. Essential business entities (EBEs) are things or entities that form the essence of the business that the organization is in. For example, in a school, student, lecturer, course, school, and etc. are EBEs.
- Identify those EBEs that have a lifetime, which the organization must handle. These are called units of work (UOWs). Some EBEs corresponds to units of work (UOWs) and some does not. For instance, student is a unit of work: it applies, proceeds, graduates and etc. A course is planned, prepared, lectured and etc. Depending on the scope (whether we are interested in the lifetime of the school), the school might or might not corresponds to UOW.

- Create UOW diagram that shows the (dynamic) relationships between UOWs that pertain when one UOW generates (or calls for or demands or activates or requires) another. For example, a project generates one or more artifacts.
- For each UOW, hypothesizing that, in the process architecture, there will be a case process that deals with a single instance of the UOW, a case management process that deals with the flow of instances, and a case strategy process that determines the future strategy for both the case and case management processes.
- Transforming the UOW diagram into a corresponding process architecture by turning the relationships between UOWs into relationships between the corresponding case processes and case management processes.
- Document each process (case and case management) in detail by a team deemed to be its owner [62].
- Review and validate descriptions and develop Role Activity Diagram for each process.

The approach focuses on the interaction between roles which ease the modeling of concurrent engineering processes. It provides a well-established method for identifying process architecture of the organization. However, the process of modeling and maintaining the organizational processes is yet centralized. In addition, the notation (RAD) lack necessary constructs for representing the information aspects of the processes, which leads to difficulties in encapsulating and isolating the roles and its interface in terms of the information it requires and produces.

The Riva method is not supported via a commercial tool. Although there is a simple graphical tool (RADModeller) for modeling role activity diagrams, the support in the sense provided with generic enterprise modeling or business process management tools is nonexistent.

2.6.3. User Enabled Business Process Modeling

One of the very few attempts that put user to the center of the modeling process is the 'User Enabled Business Process Modeling' methodology that are applied in a number of process modeling projects carried out as pilots in different organizations [8]. In the literature, there are not much resources and information available about the details of the method, procedures as well as the implications of implementing it in organizations. The activities to be carried out throughout modeling project are as follows:

1. Project organization and initialization: The project starts with setting up the project group (steering) and a technical group. The project group comprises consultants (process engineers) and members of the organization. The technical group includes members of each main organization unit. The first step into the project is the definition of an adapted usage of the model elements to establish a common understanding about the terms and wording within the enterprise specific domain and to have a mapping between application domain and methodology.

2. Training during the project: During a pilot project consultants train and motivate the members of the technical group over a set of tutorials that includes concepts related to the modeling notation, the method itself, interview techniques, model verification, and etc.

3. Motivation: The results of the first project including the models as well as the benefits to the end users, are used to motivate the members of the project and technical group.

4. Visualization and distribution of processes: The processes specific to the organization units are visualized by the related member of the technical group, who also bridges the members of the units to the project group.

5. Presentation of the pilot project: The outcome is presented to the organization together with pros and cons of the development.

6. Partial projects: The members of the technical group receive self standing small modeling tasks. The results are discussed together with the consultants.

7. Projects initialized by the members of the project group: The 'snow ball principle' is initialized and runs by inviting the process owner of the organization units to do small modeling tasks.

8. Derive different views according to the roles and competencies: A process assistant represents relevant business processes, organizations and systems with links to a graphical viewer run on browsers so that the enterprise stakeholder surf through the structure of the enterprise.

The method is applied in a number of pilot organizations and projects are monitored for one year period. The results indicate that the technical group (consisting representative of each organization unit) are able run their own modeling projects and the members of these organizations use modeling to support their daily work. More than 50% of the participants stated that modeling was beneficial to support their work. As a result of user involvement and empowerment, employee contribution increases and resistance to change decreases which in turn affects work satisfaction.

The method is supported by the MO^{2}GO tools developed by Fraunhofer Institute: Production Systems and Design Technology. It is originally designed for representation, analyzing and optimization of enterprise structures and business processes. The language utilized in the tool is based on event-driven process chains (EPCs) (section 2.7.2).

The approach pursued by the user-enabled business process modeling method shifts the responsibility of modeling and improving the processes from process engineers (consultants in their case) to a group of representative end users. However, this is achieved by utilizing the techniques, the notation and tools proposed by centralized approaches. Thus, decentralization is not accomplished and disadvantages and limitations of the centralized view are inherited. The method does not provide a rigorous method for the utilization in organizations. Method's prototypical version is yet in its early phases of the development.

2.7. Business Process Modeling Notations

In section 2.3, we discuss major process modeling languages that are used in the software engineering field both for systems specifications and process execution via PSEEs. This section goes over some of the business process modeling notations that are well known and established in research and practice. These notations are being utilized in a range of fields including business process management, business process reengineering, software engineering, and enterprise modeling.

2.7.1. Business Process Modeling Notation (BPMN)

The Business Process Modeling Notation (BPMN) [25] is a standardized graphical notation for expressing business processes. The objective of BPMN is to support both technical and business users. Thus, together with simple forms of descriptions with core element set, the notation is also able represent complex process semantics for execution. The BPMN specification also provides a mapping between the graphics of the notation to the underlying constructs of execution languages, particularly BPEL4WS (Business Process Execution Language for Web Services) [14]. BPMN carries out specifications of many other notations and languages such as UML Activity Diagram, IDEF, ebXML, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM, and Event-Process Chains (EPCs). It is maintained by Business Process Management Initiative (BPMI) and Object Management Group (OMG).

Using BPMN, the activities of the business process and the flow controls are represented on a Business Process Diagram (BPD). The set of core elements for business process diagram is given in Table 1 [25].

Table 1. Business Process Diagram Core Element Set

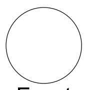
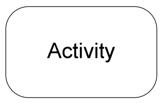
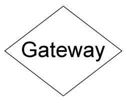
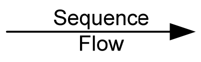
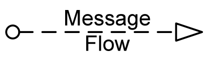
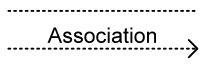
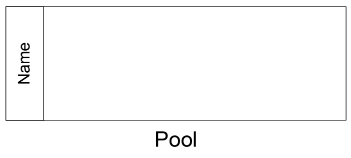
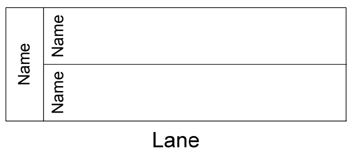
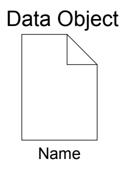
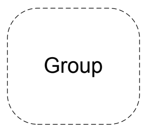
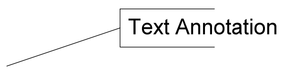
Symbol/Element	Description
 Event	An event is something that 'happens' during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result). There are three types of Events, based on when they affect the flow: Start, Intermediate, and End.
 Activity	An activity is a generic term for work that company performs. An activity can be atomic or non-atomic (compound).
 Gateway	A Gateway controls the divergence and convergence of Sequence Flow. It determines branching, forking, merging, and joining of paths. Internal markers (and, or, xor) indicates the type of behavior control.
 Sequence Flow	A Sequence Flow shows the order that activities is performed.
 Message Flow	A Message Flow shows the flow of messages between two participants that are prepared to send and receive them. Two separate Pools represents the two participants (e.g., business entities or business roles).

Table 1. Business Process Diagram Core Element Set (continued)

Symbol/Element	Description
	An Association is used to associate information with Flow Objects. Text and graphical non-Flow Objects can be associated with the Flow Objects.
	A Pool represents a Participant in a process. It also acts as a 'swimlane' and a graphical container for partitioning a set of activities from other Pools.
	A Lane is a sub-partition within a Pool and extends the entire length of the Pool. Lanes are used to organize and categorize activities.
	Data Objects represents artifacts - the input and output of activities.
	Represents a grouping of activities and that does not affect the Sequence Flow.
	Text Annotations represents additional information for the reader.

BPMN, originating from the process engineering field [100], used for models purposing not only the description of processes but also their execution. Thus it also has a more extensive list of elements that can be utilized for process execution and mapping for execution oriented languages. Figure 11 presents an example diagram for the travel booking process [151].

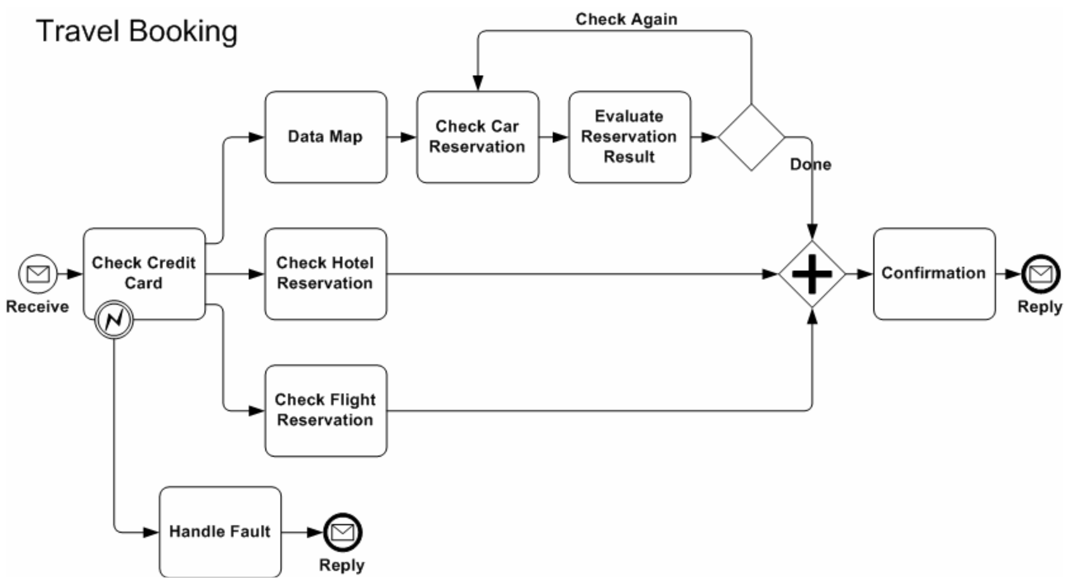


Figure 11. An example business process diagrams with BPMN

Support for execution makes BPMN strong in representing behavioral and functional aspects of the processes. However, its support for organization and information perspectives of the processes is weak (refer to section 3.6.1 for the discussions on perspectives in processes). In addition, the role concept, which plays a central part in decentralization of the modeling practice, is not implemented explicitly.

2.7.2. extended Event Driven Process Chain (eEPC)

extended Event Driven Process Chains (eEPCs) are semi-formal and widely accepted in practice [125]. Its formalism enables to integrate organizational and informational views of the processes into the behavioral view. eEPC is based on event-driven process chains (EPC) introduced by Scheer [125]. EPC is a business process modeling notation mainly focusing on representing the control view of business processes. It is a part of the ARIS method [74] and mainly used for business process management, business process reengineering, workflow definition, software development and activity based costing (ABC).

The main constructs of the EPC are *functions* (*activity in Plural*) and *events*. An event can trigger a function or a function can produce an event so combinations of events and functions in a sequence produce EPCs. In order to represent control relationship between triggering events and functions, logical operators are used. An EPC illustrates the chronological course of a business process [74]. A process model can be hierarchically structured across any number of levels by assigning more detailed EPC models to every function within an EPC.

EPCs are extended to capture data and organization view of business processes as eEPCs. In extended event-driven process chain (eEPC) diagrams, the information objects of data view and organizational element of organizational view can be represented together with the functions of the function view and the events of the data view.

Events define the state or condition that starts and ends an activity, so events are always the start or the end nodes of the eEPC diagrams. The process starts with a stimulus message, that is, an external event that triggers one or more activities in the process.

In order to illustrate the splits/joins and processing loops in eEPC, logical operators (rules) are used. Figure 12 presents three examples of operator usage.

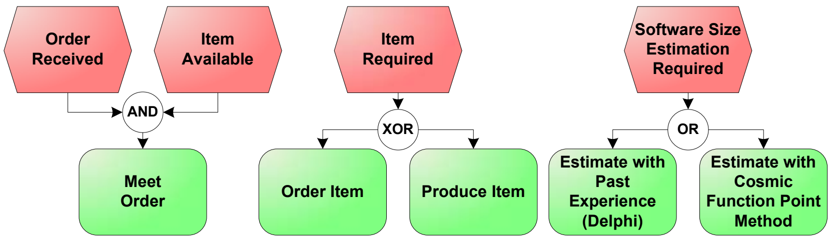

Order Received
Item Available
AND
Meet Order
Item Required
XOR
Order Item
Produce Item
Software Size Estimation Required
OR
Estimate with Past Experience (Delphi)
Estimate with Cosmic Function Point Method

Figure 12. Logical operators (rules) in eEPC

Figure 13 presents an example of an eEPC diagram [103].

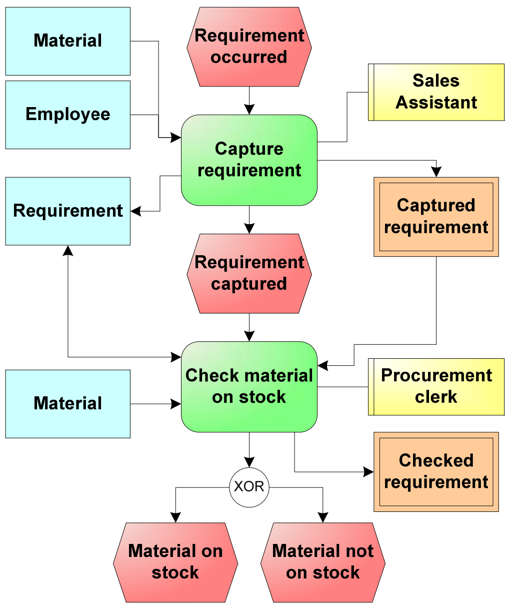


Figure 13. An example eEPC diagram

eEPC have a rich set of process elements and their connectors. The set of process elements, connectors and attributes can be limited depending on the modeling purpose. For each project a specific set of process constructs can be determined and set.

As eEPC is the center of ARIS framework (section 2.5) in control view that integrates function, data, organization, and output views, the notation is strong in representing the functional, behavioral, information and organizational aspects of the processes.

2.7.3. Role Activity Diagram

Role Activity Diagram (RAD) is based on role modeling. A process is modeled as a number of roles that interact with each other. In this context, a role is viewed as an area of responsibilities that can be carried out by a person, a group of people, or a computer system. There are a number of activities taking place within each role in a certain order.

A RAD consists of a set of activities, decisions and transactions. Decisions and transactions essentially form the interactions between two roles. The basic concepts of RAD were first introduced by Holt et.al. [68], and later enriched by Ould [115]. Figure 14 shows the basic process elements used for RADs. [116]

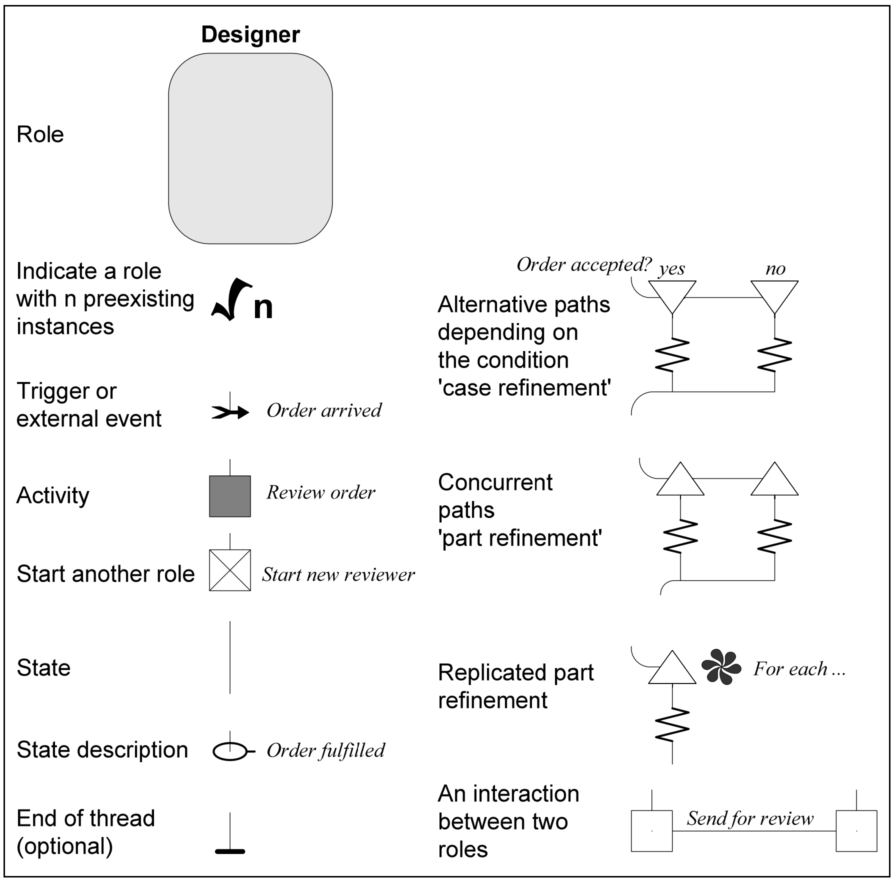

Designer
Role
Indicate a role with n preexisting instances
n
Trigger or external event
Order arrived
Activity
Review order
Start another role
Start new reviewer
State
State description
Order fulfilled
End of thread (optional)
Order accepted?
yes
no
Alternative paths depending on the condition 'case refinement'
Concurrent paths 'part refinement'
Replicated part refinement
For each ...
An interaction between two roles
Send for review

Figure 14. Process elements for RAD

The activities carried out by a role are depicted in a grey box labeled with its name. At the top of the box, in one or more of the roles, a small arrow with a phrase describes the event that triggers the process. The RAD is read by tracing the lines from that event. A black box represents an activity carried out by a role. Decisions are represented with a question and the alternative courses of action follow, each headed by a triangle pointing downwards. Where a role starts a number of concurrent threads of action, those threads are shown hanging from a connected set of upward-pointing triangles. Role collaboration (passing information, agreeing something, approving something and etc.) is represented with a horizontal line joining a white box in each of the interacting roles. A circle at the end of the thread indicates the ending event - the process goal. Figure 15 shows a portion of an example RAD.

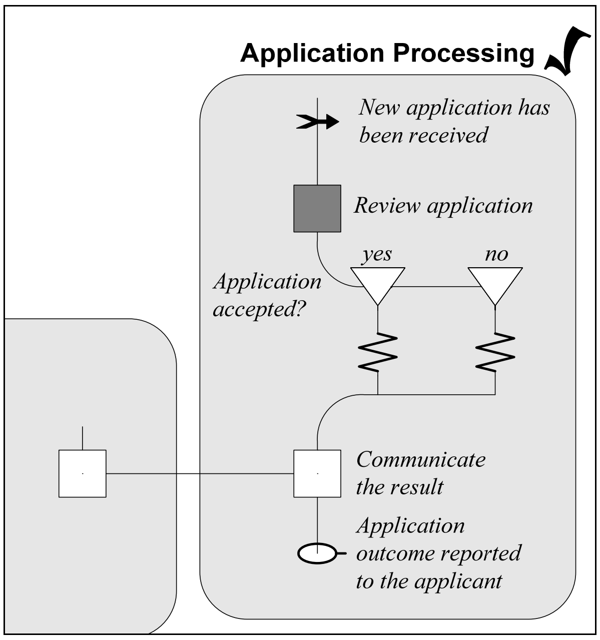


Figure 15. A portion of a RAD

RAD has strong capabilities in representing systems in terms of interacting roles and their relationships. However, it is weak in representing informational perspective of the processes as well as inheritance, composition and reuse of the specification.

One of the important distinctions between RAD and other process modeling notations is the view that processes are connected in a network rather than being contained in one another. Thus, processes are not decomposed into successively smaller sub-processes but rather represented as a RAD comprising all what it has. This representation may have limitations and disadvantages for processes that are larger in scope and complexity. It also has problems in presenting the whole picture to the user.

2.7.4. Integrated DEFinition (IDEF)

The Integrated DEFinition (IDEF) is a suite of definition languages that have become standard modeling techniques [67]. Although IDEF was originally intended for use in systems engineering, the suite evolved and currently contains the necessary notations to support software development. Currently, it covers a range of uses from function modeling to information, simulation, object-oriented analysis and design and knowledge acquisition.

In mid 70's, in response to the identification of the need to improve manufacturing operations, U.S. Air Force established the Integrated Computer-Aided Manufacturing (ICAM) program. To model functions (processes), data, and dynamic (behavioral) elements of the manufacturing operations, the program developed the IDEF which is largely based on Structured Analysis and Design Technique (SADT) [105]. The aim was to use IDEF as a regimented approach to analyzing an enterprise, capturing current process models, and for modeling activities within an enterprise. Thus, an enterprise could develop a basis for process improvement planning and have a foundation to define information requirements. IDEF is currently maintained by Knowledge Based Systems, Inc.

The IDEF methods that are currently pursued are as follows:

IDEF0: Function Modeling	IDEF7: Information System Audit Method
IDEF1: Information Modeling	IDEF8: User Interface Modeling
IDEF1X: Data Modeling	IDEF9: Scenario-Driven IS Design
IDEF2: Simulation Model Design	IDEF10: Implementation Architecture Modeling
IDEF3: Process Description Capture	IDEF11: Information Artifact Modeling
IDEF4: Object-Oriented Design	IDEF12: Organization Modeling
IDEF5: Ontology Description Capture	IDEF13: 3-Schema Mapping Design
IDEF6: Design Rationale Capture	IDEF14: Network Design

IDEF0 through IDEF4 (including IDEF1X) are the ones that are most commonly used.

The IDEF0 is the most representative construct of IDEF methodology. It shows the high-level activities of a process. It allows the user to depict a view of the process including the inputs, outputs, controls and mechanisms (refereed as ICOMs) (Figure 16).

- Inputs are resources consumed or transformed (refined) by the process.
- Outputs are the elements created through the consumption/transformation of the inputs by the process.
- Controls are the objects guiding the process: policies, guidelines, standards, laws.
- Mechanisms are the agents that accomplish the actions (activities) contained by the process.

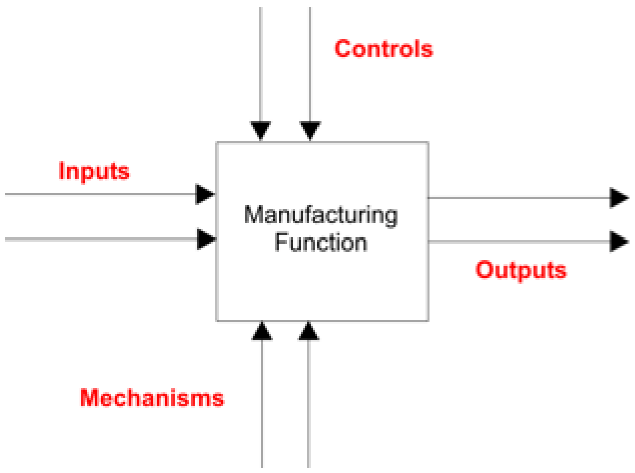


Figure 16. IDEF0 Box and Arrow Graphics

The processes can be further decomposed to show lower-level activities and ICOMs.

IDEF1 is used to identify what information is currently managed in the organization and to capture conceptual views of the enterprise’s information. IDEF1X is used for data modeling, which captures the logical view of the enterprise's data and is based on an entity relationship model. It is a design method for logical database design once the information system requirements are known.

IDEF2 Simulation Model Design method is used to represent time varying behavior of resources in a manufacturing system. It has been replaced by various commercial products and notations.

The IDEF3 Process Description Capture provides a mechanism for collecting and documenting processes. IDEF3 comprises two description modes, process flow and object state transition network. A process flow description is used to describe what happens to a part as it flows through a sequence of processes (Figure 17). The object state transition network description summarizes the allowable transitions an object may undergo throughout a particular process (Figure 18) (adapted from [92]).

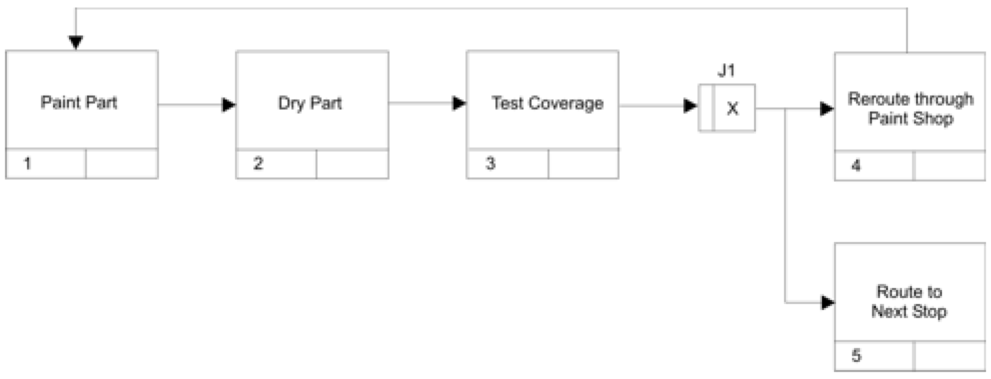


Figure 17. An Example IDEF3 Process Description Diagram

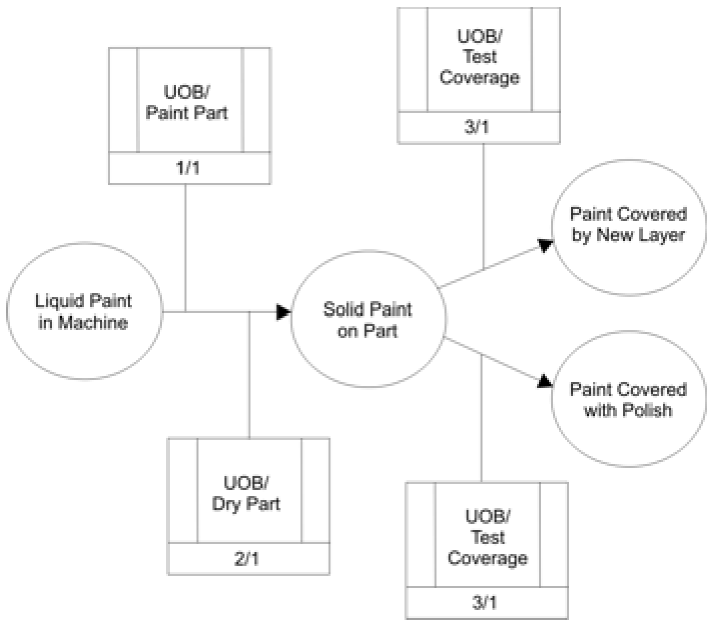


Figure 18. An Example IDEF3 Object State Transition Network Diagram

The IDEF4 object-oriented design method is designed to support the object-oriented paradigm. The basic organization of an IDEF4 model is given in Figure 19.

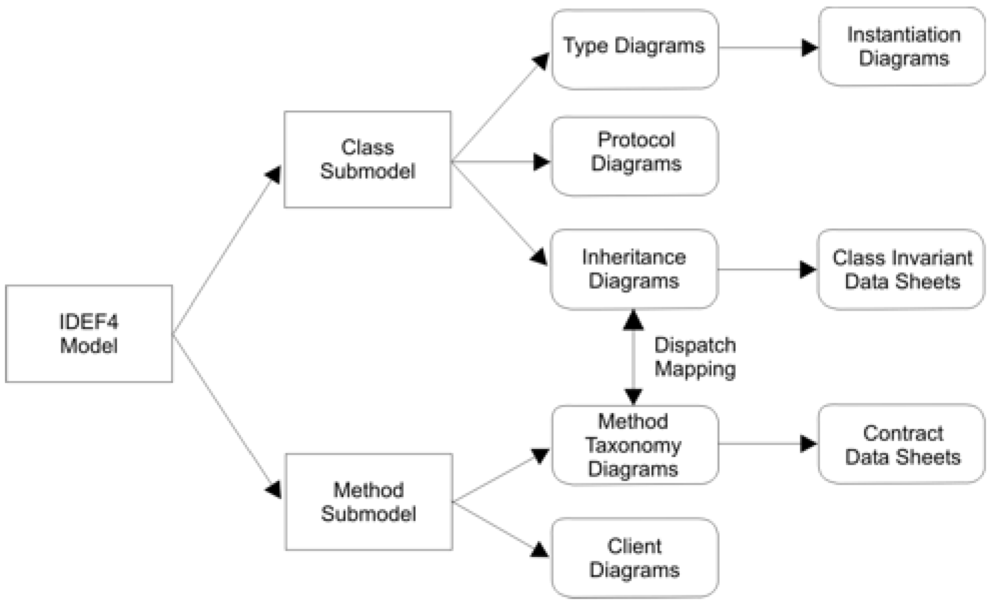


Figure 19. Organization of the IDEF4 Model

IDEF4 comprises two conceptual sub-models; class and method. The class sub-model is composed of; the inheritance diagrams that specify class inheritance relations; type diagrams that specify class composition; protocol diagrams that specify method invocation protocols; and instantiation diagrams that describe object instantiation scenarios that assist the designer in validating the design. The method sub=model comprises; method taxonomy diagrams which classify method types by behavior similarity and client diagrams which illustrate clients and suppliers of methods, to specify functional decomposition.

IDEF4 is specifically targeted towards software development and as such has similarities with the class, object, activity and state diagrams in the Unified Modeling Language (UML) [113], which is one of the leading languages utilized in object oriented system analysis and design.

The IDEF5 method is specifically designed to assist in creating, modifying, and maintaining ontologies. IDEF6 through 14 exist today in various stages and are intended to provide the capability to describe additional views.

Although IDEF has strong capabilities to view organizational knowledge from multiple perspectives, its support for the separation of the concern based on roles (thus agents) is weak. The emphasis in IDEF3 diagrams is on representing a sequence of activities performed over an organizational entity, which is a method frequently utilized for workflow systems via process enactment.

There are several standalone tools supporting one or more IDEF notations. For example, Knowledge Based Systems, Inc. provides AI0 Win 6.0 for IDEF0, SmartER 5.0 supporting IDEF1 and IDEF1X, ProSim 6.0 for IDEF3 and SmartClass supporting IDEF4. Similarly, IDEFine maintains Workflow Modeler, Simulator and Generator tools for related IDEF notations.

2.8. Agent-based Approaches to Business Process Management

In the last decade, agent-based approaches in artificial intelligence and business process management fields have received great interest for engineering complex distributed systems. Increasingly, many computer systems are being viewed in terms of autonomous agents. Proposed solutions have already been developed for many different domains and software engineering [80], [122] and business process management (BPM) [78], [81] are no exceptions.

Jennings [82] and Wooldridge et.al. [154] argue that analyzing, designing, and implementing 'complex software systems' as a collection of interacting, autonomous agents (i.e. multi-agent system) offer a number of significant advantages over contemporary methods. In [153], Wooldridge summarizes why agents are perceived to be a significant movement in software engineering and then reviews several techniques and formalisms that have been developed for engineering agent-based systems. Both studies emphasize that the agent-based methods are promising approaches to developing a range of complex and distributed systems.

In agent-based approaches, an *agent* is an autonomous, problem solving computer program that interacts with other agents when it has interdependencies [80]. It is a computer system, situated in some environment that is capable of flexible autonomous action in order to meet its design objectives [79]. Following are the basic attributes of agents:

- Agents are *autonomous* in that they are not controlled directly by humans or others and they have control over their own actions and internal state;
- they are *responsive* and perceive their environment and react to it;
- they are *proactive* in that they agents do not simply act in response to their environment, they exhibit opportunistic, goal-directed behavior and take the initiative where appropriate; and
- they are *social* in that they interact, when appropriate, with other artificial agents and humans in order to complete their own problem solving.

One of the most influential study on applying agent-based approaches to business process management is the ADEPT [78].

2.8.1. ADEPT

The ADEPT (Advanced Decision Environment for Process Tasks) project is presented as a method to conceptualize business process management as a collection of intelligent agents. Jennings and his colleagues [78] argue that there are several common characteristics of business processes which make an organization suitable to be viewed as a multi-agent system. Some of these common characteristic which are directly relevant to our focus can be summarized as:

- Within organizations, there is a decentralized ownership of the tasks, information and resources involved in the business process.
- Different groups are relatively autonomous - they control how their resources are consumed, by whom, at what cost, and in what time frame. They also have their own information systems, with their own idiosyncratic representations, for managing their resources.
- There is a high degree of natural concurrency in process execution.
- Business processes are highly dynamic and hardly predictable - it is difficult to give a complete a priori specification of all the activities that need to be performed and how they should be ordered. Any detailed time plans which are produced are often disrupted by unavoidable delays or unanticipated events (e.g. people are ill or tasks take longer than expected).

ADEPT's approach is to delegate responsibility for '*enacting*' specific business process activities to the constituent components – autonomous agents, rather than maintain it centrally. Thus, each of the business processes' main activities are assigned to a particular problem solving entity, and that entity is responsible for ensuring the activity is fulfilled within the specified constraints. How or by which means a goal is accomplished is left to the responsible entity to determine. An agent may need services of others to achieve specific sub-activities, thus responsibility delegation continues can continue through many levels of nesting. As a consequence, many decisions that are traditionally made in the process description at design time are moved to the execution system and determined at run time.

Agents are autonomous, thus, there is no control dependencies between them. Unlike the case in object-oriented systems, if an agent requires a service that is managed by another agent, it cannot simply instruct it to start the service (methods in OO). Rather, the agents should come to a mutually acceptable agreement about the terms and conditions under which the desired service will be performed. In ADEPT, these contracts are called *service level agreements* (SLAs), which is analogous to 'treaties' in Oz environment [19]. In order to define SLAs, agents go through an inter-agent negotiation process in which parties express conflicting demands and then they either make concessions and agree or search for new alternatives. Details of the negotiation protocol are given in [78]. The conceptual architecture of an example ADEPT system is given in Figure 20 (adapted from [81]).

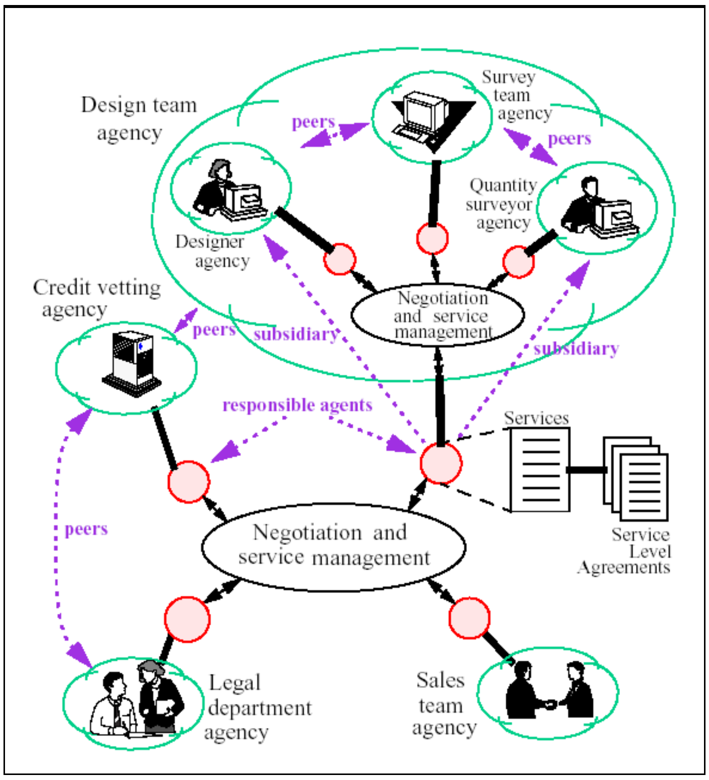


Figure 20: The conceptual architecture of an example ADEPT system

In ADEPT, the term 'service' denotes manual or automated activities that an agent can manage. Designing an artifact, providing an insurance quote for a customer, or reviewing a paper for a scientific journal are some example for services. A task is the simplest service and represents an atomic unit of problem solving in the ADEPT system. A service is described by *service description language* (SDL) which corresponds to ADEPT's process definition language. Figure 21 [81] gives an exemplar service definition with SDL. The definition comprises the service name, a set of inputs, a set of outputs, a guard, and a body. The *inputs* (optional or mandatory) specify the information used by the service. ADEPT also uses information sharing language that performs a number of semantic mappings and transformations when agents are interacting. The inputs and outputs are defined in terms of the information model of the agent that is responsible for the service.

When the service is invoked, the *guard* condition is evaluated. If it evaluates to false, the service fails without the body of the service being processed.

```
(service
 name     Prepare_Table
 inputs   (Home_Guests guests cli man
     Home_TableAndChairs accommodation ser man
     Home_Cutlery cutlery any man
     Home_Crokery crockery any man
     Home_Glasses glasses any opt)
 outputs  (Home_Seat_Allocation)
 guard    "( <= guests.number accommodation.number )"
 body     ( ..... ))
```

Figure 21: ADEPT Service Description for 'Prepare_Table'

The *body* specifies how the service is to be executed. It comprises the restrictions on the order of its component services, the conditions under which the service will be considered successful, and how information flows between those component services. The body is composed of a single block and a block can be decomposed to further nested sub-blocks. Block's syntax is the following:

```
<block-type> `:' <block-identifier> `{' <execution-list> `} ->'
<completion-expression>
```

The <block-type> represents the sequence. A service can/must be performed in parallel (can-para/must-para), or iterates until some condition holds (loop). Figure 22 gives an example of the usage of block types. The <block-identifier> refers to the completion state of that block. The <execution-list> is a comma separated list of services, conditionals and blocks.

Services are called by referring to them by their names together with their parameters to be executed.

```
sequence:meal{
must-para:organise {
  cond:have_friends "(not (empty-set service::friends))",
  Plan_Menu(restrictions = service::friends),
  Plan_GuestList( candidates = service::friends),
  } -> (and have_friends Plan_Menu Plan_GuestList)
can-para:prepare {
  Prepare_Food(food = Plan_Menu::menu),
  Prepare_Table(guests = Plan_GuestList::choice)
  } -> (and Prepare_Food Prepare_Table),
Eat_Meal(<unspecified>),
can-para:clean_up {
  Wash_Up(<unspecified>), Dry_Up(<unspecified>)
  } -> (and Wash_Up Dry_Up)
} -> (and organise prepare Eat_Meal clean_up)
```

Figure 22: ADEPT Service Description for 'Meal'

Agents communicate via an *agent communication language* (ACL). ACL consists of messages containing a limited number of primitive message types, which includes the identity of the sender, recipient and thread of communication, the service concerned, and the information model with reference to which the contents of the message should be understood. Table 2 (adapted form [78]) presents the message types. Ten of these messages are used during negotiation and three are used during service execution.

Table 2: ADEPT Sample SLA (service level agreement)

Slot Name	**Instantiated Values**
SERVICE_NAME:	cost_&_design_network
SLA_ID:	a1001
SERVER_AGENT:	NDD
CLIENT_AGENT:	CHL
SLA_DELIVERY_TYPE:	on-demand
DURATION (minutes):	320
START_TIME:	9:00
END_TIME:	18:00
VOLUME:	35
PRICE (per costing)	
PENALTY:	30
CLIENT_INFO:	customer_profile
REPORTING_POLICY:	customer_quote

2.8.2. Discussions on Agent Based Approaches

The way the agent-based approaches view organizations and the notion of agent is analogous to the view we have for modeling and enactment of business processes in a decentralized manner. With this perspective, every computerized agent can be coupled with a human agent (or its role) to support or perform related activities and collaborate with others. However, the emphasis in agent-based approaches is yet on the enactment of the agents' service definitions, which are presumed to be defined already. Thus, to meet the requirements posed by decentralization in modeling, these approaches need to be extended with a mechanism, a graphical notation and a tool for human agents to define the set of services they provide and their dependencies to others. In addition, we believe that the approaches need also elaborate the role concept as well as the representation and handling possible inheritance and composition relationships both between the agents (roles) and information items.

CHAPTER 3

THE PLURAL METHOD

For an approach to achieve its benefits and be useful, it is essential to provide a systematic way to implement it in an organization. Plural is a method for business process modeling. It provides a disciplined guidance for organizations to perform modeling in a decentralized and concurrent way.

This chapter presents a method we proposed for organizations to perform business process modeling. First section of the chapter discusses the modeling approach pursued by Plural. Sections 3.2 through 3.5 present the Plural method and its phases in detail. Section 3.6 describes the Plural notation. Finally, section 3.7 defines a brief set of requirements for a tool that can be utilized for the method and presents a prototypical toolset developed for this purpose.

3.1. The Modeling Approach

Since the systems we are dealing with are generally large and complex, we need ways to separate concerns [77]. Separation of concerns is a process of decomposing a complex system into more simple units according to certain criteria. We divide the system into simpler units in order to isolate related parts from other concerns that might be out of interest. For example, the waterfall lifecycle in software development is based on the temporal separation where the development process is separated into time slots during which different activities such as requirements, coding, etc. are performed.

Plural perceives an organization from two perspectives. Figure 1 (pg.6) depicts this structure of the organization in the way it is considered in Plural. First, there is a network of roles carrying out activities. They share (demand and supply) information between each other in order to carry out the activities they are responsible for. These activities collectively form a part of or a complete process. Second, there is network of processes interconnected between each other similar to the way the roles are interconnected. In general, the process modeling approaches concentrate on the second network during process definition. That is, the separation of concern is across processes. Plural focuses on processes as well as roles, and utilizes them as the primary means for structuring the modeling process. Thus, it tries to achieve a separation of concern across both entities - roles and processes.

Plural’s approach is based on allowing each participant in the organization to come up with the partial model of the processes s/he participates in terms of the roles s/he plays. These partial models are then integrated to form the complete picture of the process and organization’s process network. The idea from this perspective implies a complete grassroots approach to modeling. That is, the complete representation is derived by joining what is represented by all. However, the process definition executes more efficiently when, at the beginning, the goal, the objectives and more importantly the scope of what is to be modeled is communicated and accepted by all parties. Simply, a backbone is necessary, which sets the goal and boundaries without intervening or even directing what is to be done inside by each participant (agent). Then, each agent is given full responsibility to define what it does in the slot it owns and how it interacts with others by communication and negotiating with others.

The concept used in linking performers and processes is *role.* That is, the relationship between agents and processes was created via roles, where each agent, while participating in the processes, is playing one or more roles. Figure 23 depicts these relationships.

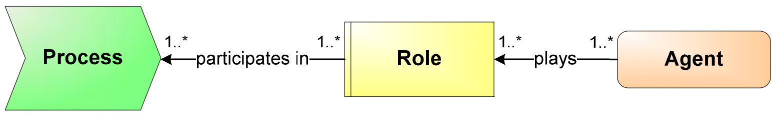


Figure 23. Agent, Process and Role

There is an analogy between the way the object-oriented methods deal with systems and the way Plural perceives organizations. In the object-oriented concept, the problem is divided into number of objects that interact with each other to solve the problem. Objects without knowing the details of how other objects provide their services, call them to solve the problem they are responsible to solve. In the case of software systems, the problem is to enable an actor to satisfy a goal with the software s/he uses. Each object is designed to be self-contained module with a clear responsibility and the tools (attributes and actions) necessary to carry out its role (encapsulation). In addition to knowing how to perform its responsibilities, each object has the requisite information (attributes) and knows exactly what information it needs to obtain from its collaborators. In that respect, notion of roles is analogous to objects in the problem space. They interact with each other in order to perform the activities they are responsible from. When they require, they make use of the services (operations) provided by other roles. Similar to objects, roles are not interested with inner activities of how others provide their services unless the service is given on time with appropriate quality (information hiding). Accordingly, a role can be described by an interface containing the set of services that it offers to its environment, called its input interface, and by an interface containing the set of services that it may request from its environment, called its output interface [5]. Each role (or associated agent) will strive to increase the quality of the service it provides. It will negotiate with suppliers not only to increase the quality of the services it uses but also to demand new ones.

Role concept enabled agents to represent the information they know best about the processes, i.e., the activities they perform via the roles they act for. In addition to the list and sequence of activities, more challenging information to be extracted from the agents was the interaction between her roles and the other's roles in the system. The interface was considered to be the messages it sends and receives and the activities it performs together with other roles. This information became the gluing points of the individual models. Hence, the idea is mainly based on roles defining their dependency as a set of expectations from other roles. Dependencies are the logistic, financial, informational, or managerial relationships that members of a process establish with other people or software systems in the process to achieve their goals [107], [104]. The approach presumed that a dependency forms a precondition for an activity to be performed. In that respect, a role (thus, the associated agent) should somehow be notified or communicated about whether that precondition is satisfied or not. The message can include a notification about the state of an object in the environment or the object itself. For example, an agent can receive a message indicating that a report is ready or it can receive the report itself. The communication is via a message it receives from the environment originating from other roles played by people or other systems such as applications.

In the real world, ideally, expectations are fulfilled. That is, an agent, playing a role in the process, are generally provided necessary resources and inputs in order to perform its activities and produce related outputs. We capture and articulate this knowledge by expecting each agent to express its necessary inputs and resources together with the roles that they think they get that inputs and resources from. They are also asked to define the outputs and the roles that they think they sent to. As these expectations were defined and eventually fulfilled or renounced, the interface points between the roles became apparent. Figure 24 presents an individual process description of a role (role A). The figure displays how a role represents the activities it performs and its expectations from others.

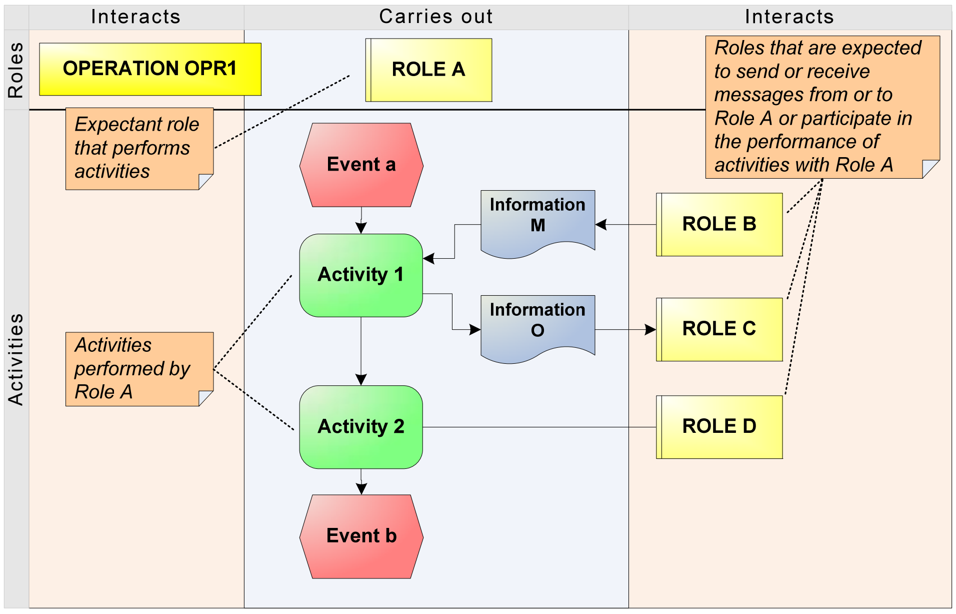

Interacts
Carries out
Interacts
Roles
Activities
OPERATION OPR1
ROLE A
Roles that are expected to send or receive messages from or to Role A or participate in the performance of activities with Role A
Expectant role that performs activities
Event a
Information M
ROLE B
Activity 1
Information O
ROLE C
Activities performed by Role A
Activity 2
ROLE D
Event b

Figure 24. Roles and Expectations

Each individual models and maintains his processes in coordination with the suppliers and consumers of these processes. In this scheme, the coordination team (analogous to process engineering group) are the facilitators and catalyses between individuals and ensures that the process of process modeling is performed and maintained as planned.

Plural's approach is mainly for organizations where knowledge workers are integrated and collaborate for production. In this type of organizations -which are sometimes called human oriented systems [29]- human have central and active role in performing the activities needed to accomplish their goals. They interact and collaborate among themselves and with computerized tools. The purpose of these tools is to assist their work and increase their efficiency and effectiveness. Software engineering organizations are good examples of this type. Others may include an engineering center, a symphony orchestra [73], a bank [29], and a headquarter, where commander and officers plan and monitor the maneuvers. The idea is also applicable in organizations where a part of the business is performed by knowledge workers e.g. engineering departments in a manufacturing company.

3.2. Method Phases

Plural method can be executed as a process in organizations. Thus, it can be performed throughout the life of the organization to define and improve its processes and to maintain its process-base. The organization goes through three phases in order to establish its process-base and necessary infrastructure. Figure 25 presents these three phases and the way each can proceed.

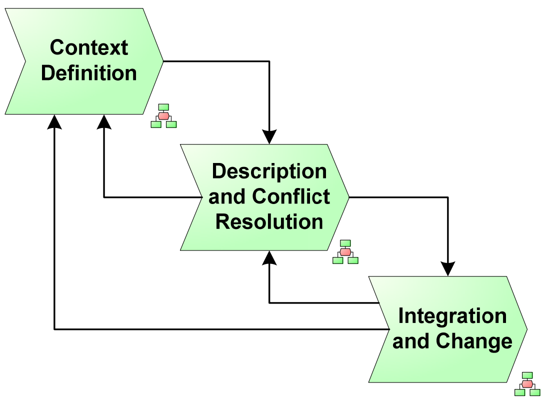


Figure 25. Phases of the method and feedbacks

In the *context definition* phase, all process owners collectively define the aim and scope of the modeling process. Based on the roles each agent plays in the organization, the development agents start defining the activities they perform in the processes in the *description and conflict resolution* phase. The development agents identify and resolve inconsistencies and conflicts between their definitions and others. This role based definition is considered complete after they are validated by associated peer agents and verified by the coordination team. In the integration and change phase, complete and consistent models are merged; new models are generated and analyzed. Change requests related to processes are proposed.

The process definition continues as a never ending cycle, rather than being a project undertaken by a team of people and completed once process models are defined and stabilized. After the organization establishes its process-base, the cycle of phases repeats itself for number of times for any change request until the change is incorporated and the processes reach to another consistent and complete state. Therefore, the third phase may last longer where the organization's process-base settles in a complete and consistent state. This state is no longer valid once a change is proposed regarding to the context or the processes. Based on its type, the request triggers the first or second phases. Sections 3.3, 3.4, and 3.5 presents each phase in detail, describes activities carried out by certain process roles, and provides guidelines and examples.

Being a set of activities to be carried out in the organization, the process of process modeling can be represented with the notation utilized in representing organizations' business processes. Therefore, the study describes each phase of the method with a process diagram depicting the activities performed, the roles that participate and the artifacts produced. The process diagram comprises columns (swimlanes) each representing the activities performed by a specific role. The notation for the process diagrams is described in section 3.6.4.

3.3. Context Definition (Phase I)

This phase sets up the organization for concurrent and decentralized process modeling. The primary goal is to achieve a structural frame of the organization in terms of a high level process network, participating roles and agents and their structural relationships. Figure 26 illustrates the process diagram for the context definition phase. First, the group determines and states the aim and objectives for modeling processes. The coordination team is established. The processes that will be modeled and the roles that participate in those processes are determined and depicted on related diagrams. Roles are mapped to the agents and the execution plan is documented, communicated to all stakeholders and approved by all process owners. As in any other similar undertakings in an organization, the support from upper level management is critical. All stakeholders should have a clear understanding and consensus about the aim and scope of this initiative, their role and responsibilities.

Following subsections describe the activities of this phase in detail.

3.3.1. The (Kickoff) Meeting and Determining the Purpose for Modeling

The organization initiates the process with a kickoff meeting that brings all related process owners and stakeholders together. Process owners include the individuals that participate in the execution of the processes that are likely to be covered. Stakeholders may include all individuals that are affected by execution of the processes such as internal or external customers of the artifacts produced by these processes and upper level management which sponsors and supports the modeling process and ensures that processes promote the vision and mission of the organization.

The moderator or a participant familiar with Plural concepts introduces the Plural method and present a brief overview of the path that will be followed by the organization. The moderator ensures that all participants have a clear understanding of the process and have their questions answered. It is also a good practice to prepare a number of sample process diagrams of organization’s processes before the meeting, which might provide insight into the notation used and definition process.

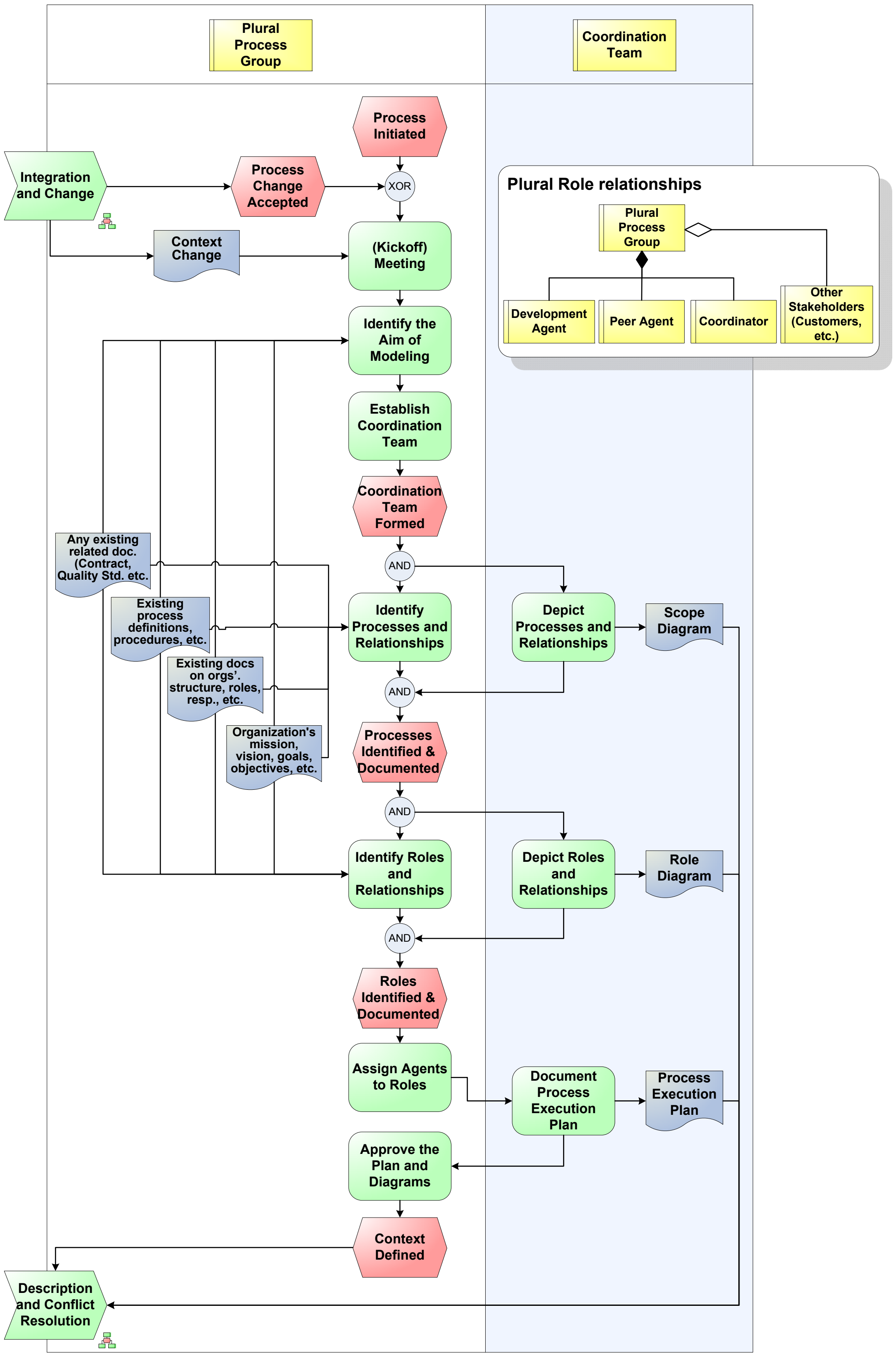

Plural Process Group
Coordination Team
Process Initiated
Integration and Change
Process Change Accepted
XOR
Context Change
(Kickoff) Meeting
Identify the Aim of Modeling
Establish Coordination Team
Coordination Team Formed
AND
Any existing related doc. (Contract, Quality Std. etc.
Existing process definitions, procedures, etc.
Existing docs on orgs'. structure, roles, resp., etc.
Organization's mission, vision, goals, objectives, etc.
Identify Processes and Relationships
Depict Processes and Relationships
Scope Diagram
AND
Processes Identified & Documented
AND
Identify Roles and Relationships
Depict Roles and Relationships
Role Diagram
AND
Roles Identified & Documented
Assign Agents to Roles
Document Process Execution Plan
Process Execution Plan
Approve the Plan and Diagrams
Context Defined
Description and Conflict Resolution
Plural Role relationships
Plural Process Group
Development Agent
Peer Agent
Coordinator
Other Stakeholders (Customers, etc.)

Figure 26. Context Definition Phase

The participants in the meeting first identify organization's intentions to model its processes. It is important at the beginning to know what the process model will be used for and what the expectations of the people involved in the processes are. As discussed in section 3.6.1, the organization may utilize process models for several purposes. Models might be used to support the execution, monitoring and control of the processes, to measure and improve them or to better understand and learn the actual execution. Models can also be used to elicit functional requirements of a software system that will support the execution of the processes [42].

The objectives of the organization in modeling its processes does not, however, change the way the Plural method is executed. The objective might influence the way the individuals perceive and model their processes. It might change how related notation and the tool can be tailored for specific purposes. For example, granularity level of the models and the level of formality applied might differ if the purpose is to support process understanding and improvement as opposed to monitoring and control.

3.3.2. Establishing the Coordination Team

At this stage, the coordination team is established. The role of the coordination team is significant in Plural. They are the primary participants of the Plural method. The dedicated team of coordinators facilitates the execution of the modeling process.

Coordinators should be familiar with process modeling concepts and Plural method. They involve in one or more of the following tasks:

- Facilitating and monitoring the definition process
- Providing guidance to the development agents (individuals actively modeling their processes) mainly in process modeling and maintaining the process network
- Representing the voice of the external roles (internal/external customers) and roles played by computerized systems.
- Verifying individual role-process diagrams
- Integrating and generating models for process analysis
- Envisaging the top view of processes as a whole, explaining and analyzing it, and identifying problems and capturing high-level improvements over the processes
- Guiding and directing agents towards improving their processes with respect to higher level organizational change requests and acting as the primary means of catalysts among agents
- Facilitating the maintenance of individual process definitions
- Improving the Plural process

- Validating that the resulting individual definitions of a process is all that should be performed to serve the goal of that process and ensuring that all activities to be performed in that process are present in the integrated models.

Depending on the scope, once the aim is determined and coordination team is established, the kickoff meeting is closed. The group may also arrange subsequent meeting with relevant group members, stakeholders and the coordinators in order to perform succeeding activities of the method including the determination of the scope, the processes to be included and the roles that participate in these processes. For these succeeding meetings, the coordinators organize valuable inputs by analyzing current process definitions, procedures and standards.

3.3.3. Identifying and Presenting Processes and Relationships

Identifying processes that will be modeled through the Plural method determines the scope of the modeling effort. The coordination team depicts the coverage on a scope diagram which represents the processes and their relationships as well as the roles that participate in these processes. Section 3.6.5 describes the scope diagram and gives examples (Figure 41 p98).

By identifying the processes that will be modeled, the process group sets the frame for the execution of the Plural method. At this stage, the group can benefit from variety of resources including existing process definitions and procedures; documents representing the organizational structure, roles and responsibilities; resources representing organization's mission, vision, goals and objectives; or any related documents such as quality standards, handbooks, and etc.

The following subsections clarify important terms concisely by going through the related literature and provide brief guidelines for organizations in identifying their processes, roles as well as their relationships with each other.

a. Business Processes and Goals

We investigated some of the views for the definitions of the term ‘business process’ in section 2.4. Yet, Plural’s definition of the term is concise: ‘a set of activities that supports one or more business goals’. Each process in an organization should be performed with a specific rationale which helps organization in achieving its one or more goals and collectively its mission and vision – reason for its existence. Goals are reasons for actions, thus a business goal is the reason or the business justification for the organization to perform that process. They are desirable and measurable states that the organization intends to achieve. The overall goal of a business process can be decomposed into sub-goals and achieving these sub-goals can be achieved by assigning them to roles [160]. Since processes are enacted by a set of interacting agents, agents take over the sub-goals by acting for these roles.

In Plural, a process model is represented with *activities* that are logical steps within a process; *events* that activates or is created by activities; *roles* that performs the activities, *information items* that are input to or output from activities; and *information sources* where information items reside for future use. Roles are also sources for information items; they interact by sending or receiving information in the forms of documents, messages, emails, and etc. *Business goals* are also a part of the representation if it ought to present goal achieving aspect of the processes. In that respect, a process can be further elaborated as one or more agents acting in defined roles to enact the process steps (activities) that collectively accomplish the goals for which the process was designed [36]. Goals are derived from organization's vision and aligned with its mission; the reason for its existence.

Process identification and scoping has a profound effect on the success of upcoming phases and the application of the method in whole. A top-driven and a collaborative approach, where a group of in identifying and judging processes is generally necessary. This is because different groups of people in the organization are likely to identify and judge the processes and their salience differently. The product of the first phase of Plural reflects this top-driven view as an important input to bottom-up modeling in the next stage. The question here is: 'Whose word counts most?'. Looking at processes from the customer's point of view is the general rule for most process movements [89]. In turn, business process definitions are formed around terms of the customer. Many definitions of process entail the creation of outputs that is of value to the customer. Keen highlights the limitations of this customer focus as it may overlook mandated processes such as tax reporting and may ignore some background processes. Investors' focus for the identification of the processes and their valuation is more valid in the long run. In this context, Keen refers investors as the stockholders, financial experts and investment houses that set the market price of a firm's stock and the agencies that fund a public –sector organization's budgets. Investors also include the customers and employees as well. 'The questions as to whether a process defines the company to investors (as well as to customers and employees) and whether it is critical to business performance help clarify the link between process and investor value.'

b. Roles, Agents and Organizational Units

Identification of the roles that participates in processes is critical in Plural, because the role concept is one of the building blocks in the Plural method. Discovery of the correct roles and their relationships early at this stage have significant effects on the efficiency and success of the definition process.

West defines role as 'a brief, summary description of a person's function in relationship to a particular aspect of the business' [150]. In that respect, roles are relative to both the hierarchical structure of the organization and a particular aspect of business. Plural adapts West's view of roles, agents and organizational units. *Agent* Ali, as a system engineer (*occupying* the *position* of system engineer) may *play* the *role* of 'producer' in relationship to engineering (process) and could have the role of 'sponsor' in the organization's process improvement effort [150]. Typically a position *covers* many roles. Some examples of roles in a software organization may include; process improvement sponsor, project stakeholder, system quality assurance, requirements analyst, software developer, purchasing approval authority, facilities manager, and etc. Figure 27 gives the Plural perspective of the relationships between these three terms.

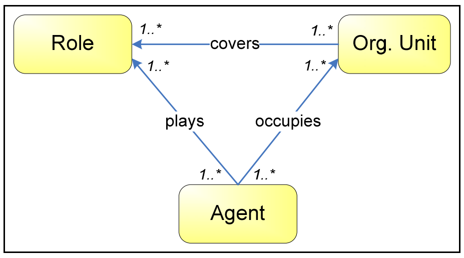


Figure 27. Agents, roles and organizational units

Organizational units are associated with the structure of the organization. The organizational entities, such as positions, departments, teams, or groups, are synthesized as organizational units. Regarding these definitions discussed above, Plural's refinement for these terms are as follows:

- *Agents* represent actors such as persons, group of persons, hardware, software or any combinations of those. They might occupy an organizational unit such as a position, department and might play one or more roles.
- *Roles* are the brief summary of a set of functional responsibilities in relationship to a particular aspect of the business. Agents perform tasks with respect to the roles they are playing in that particular aspect. They might own that particular process or a portion of it.
- *Organizational units* are structural entities in organizations which are occupied by one or more agents, and covers one or more roles. In Plural, organizational units are not represented.

Figure 28 gives an example of agents, roles and organizational units as well as their relationships. The figure depicts an example demonstrating how agents can be associated with organizational units and roles. For example, *Agent X* occupies two organizational units: the *professor* and the *department head*. Being a professor, Agent X plays the *instructor* and the *advisor* roles. As a department head, she acts for the *dept. head* role.

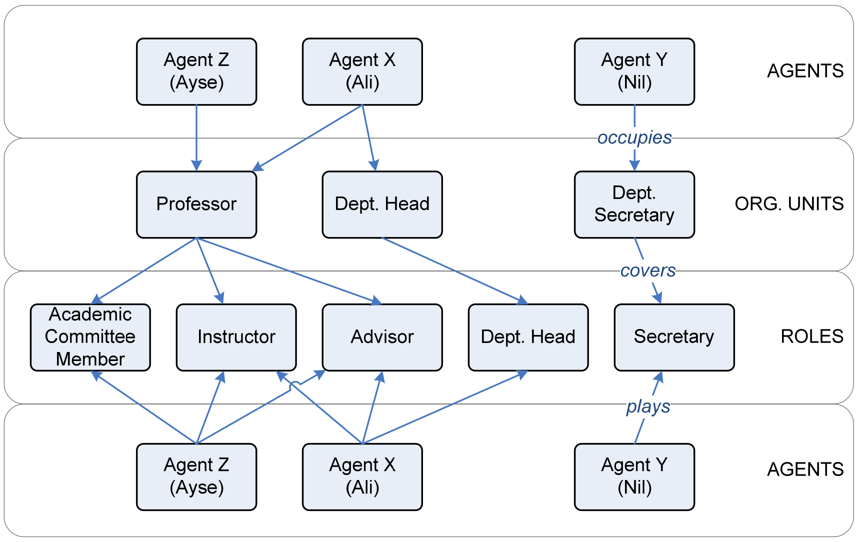


Figure 28. An example for agents, roles, org. units and their relationships

Defining the roles and responsibilities in an organization is difficult but critical in all aspects of business, particularly for successful process improvement [150]. Besides, there is no unique way or procedure that can be used for identifying and documenting these structures in organizations. Plural proposes modeling of these entities to be explicit and essential.

3.3.4. Identifying and Presenting Roles and Relationships

In this phase, the group identifies the structural relationships between roles identified in the previous activity. Since Plural utilizes a role-based process modeling approach, it is utmost important to identify the roles and their structural relationships clearly in this phase. They are important in establishing the interface between the roles and communicate this to the agents.

The static relationships between roles are depicted on a role diagram described in section 3.6.6. Figure 43 gives an example of a role diagram (pg.99). These relationships might be association, aggregation (composition) or generalization type. For example, *configuration manager* and *project manager* roles are a type of *project team member* role, that is, they have (inherit) all of the responsibilities of the project team member, which is a more general role. *Project team* is an example of an aggregate role, which might comprise other roles such as *developer*, *designer*, and etc. Plural is mainly concerned with the aggregation (composition) and generalization relationship between roles and association is secondary and informational. This is because; Plural is not explicitly interested in or requires tracing such relationships between roles and traces the relationships that directly affect the services (operations) the role provides. In that respect, representing association relationships is informative while defining aggregation (composition) and generalization relationships between roles is mandatory.

a. Active vs. inactive roles

With respect to the scope of process description, roles identified and depicted on role diagrams are of two types:

- *Active* roles are the ones whose activities are modeled by an associated agent. Each active role has its own individual role-process model describing the activities it performs in the context of a specific process. Each active role- in turn- is assigned to one or more agent in the organization.
- *Inactive* roles, on the other hand, are the ones that are external to the organization (e.g. customer) or taken out of the scope with respect to the processes covered (e.g. internal customer). They interact with active roles but they do not have process descriptions.

Active and inactive role designations can be performed on process level, that is, an active role in a process might be defined as inactive in other.

3.3.5. Assigning Agents to Roles

Having established the consensus on the scope and role definitions, each agent takes over the (active) roles with respect to their actual responsibilities in the organization. Agents can be assigned to multiple roles and roles can be taken over by multiple agents. The coordination team ensures that no active role is left unassigned. The assignments can be represented in the role diagram (section 3.6.6)

Two Plural roles related to process definition are the *development* and *peer* roles:

- *Development* (agents); are the ones that are assigned to active roles and responsible for concurrently modeling the portion of the processes their active roles participate. They are knowledge workers executing their own processes. They model their processes as they perceive their execution. They develop measurements of the processes; communicate their results by identifying conflicts and inefficiencies with the peers and coordinators.
- *Peer* (agents); are also assigned to active roles to validate the process definitions of the development agents for these roles. They are proxies for development agents. They are responsible for providing a valid process description that reflects the way the processes are carried out and the way they should be.

There are some issues that the process group may face in assigning agents to roles. The following subsections provide heuristics for assignments.

1. *Roles played by multiple agents*: It is quite typical that a role is played by many agents in the organization. For example, several agents might be working as a developer or a test engineer in a software company. Similarly, there might be number of individuals working as instructors in a university. In general, there are two alternative assignment schemes for

these circumstances. The organization might select a representative agent for the role and assign the other agents as peers. The other alternative is to let each of the agents to be assigned for the role and have models for the same role with a number equal to the number of agents assigned to it. This would enable these agents to represent and maintain their own way of performing the processes and handling their problems. Plural allows both alternatives and the process group should decide whether it is worth having multiple definitions for the same role or coming up with a unified definition. In process integration with multiple definitions, one may include all or a number of role's definitions or may include just one representative definition in the activity level process diagrams.

2. *Representing generalized roles*: In essence, the situation of representing generalized roles is almost identical to the situation of having multiple agents acting for one role. For both cases, we have number of agents acting for a role. Therefore, the scenarios mentioned above are also applicable for this case. In addition to those, a sound deviation is first to let all agents define their processes for the same role, then analyze definitions and decide whether it is applicable to generalize the definitions into a one that is accepted by all roles before going into the integration phase. In addition, during process descriptions, it is a good practice for the coordinators to be monitoring the development not only for verification purposes but also for observing any reusable process components that can be generalized. In these cases the change request, if accepted, is feed back to this phase where it changes the scope and role diagrams and assignments.
3. *Representing aggregate roles*: Aggregate roles represent the ones that are played with groups of people such as teams, departments, boards, and committees. In a role diagram, they might be represented as the aggregation of the roles that make them up, but this representation is optional which might depend on the scope. For example, it might not be necessary to represent the department head, department member, or other roles that compose the department role, if in a scope all we are interested in is how that department role acts as a whole regardless of what its subparts perform inside. For other cases, it might be necessary to represent all members to represent their responsibilities for a specific process. Regardless of how it is handled and represented in role diagrams, the aggregate role is assigned to two representative agents, one as the development agent who will model the activities that aggregate role performs in a process and the other as the peer that will validate the definition.

3.3.6. Planning the Execution

Once roles are taken over by agents, the coordination team plans the description and conflict resolution phases. The scope, Plural roles (development, peer, and coordinator) and agent assignments to process roles, the schedule and other concerns such as, risk and configuration management, tool management including setup and maintenance, are documented on a Process

Execution Plan. The plan is approved by all participating agents and stakeholders and baselined before description and conflict resolution phase commences. Plan and its approval ensure that there are no crucial disagreements between customers, investors, internal managers, and the staff about the processes, and any ambiguity related with the scope.

As will be discussed in later phases, the outputs of these steps are subject to changes as the group proceeds to process description and analysis. Change requests for the context are communicated with the coordination team and if accepted by all participants, are reflected on related models and plan by the coordinators. The plan and other diagrams are baselined and the change is communicated to all parties.

3.4. Description and Conflict Resolution (Phase II)

Once the execution plan is approved, based on the schedule, the process description and conflict resolution phase may commence. The primary goal is to come up with a complete and consistent set of individual role-process models. Modeling at this stage carries role-based modeling to a further step where the individuals that play those roles model their processes. If deemed necessary, the coordination team performs orientation sessions to agents for the procedure to be followed, the notation, the tool and the techniques to be used, before modeling initiates.

Figure 29 illustrates the process diagram for the context definition phase. As the first activity of the phase, each development agent determines the operations for each role for the processes they participate. Then, for each operation, they develop a description with an individual role-process diagram. During process description, development agents identify inconsistencies between definitions based on the expectations of other roles. Once inconsistencies and conflicts are resolved, the definitions are ready for validation by peers and verification by the coordination team. The phase ends after all diagrams are verified and validated.

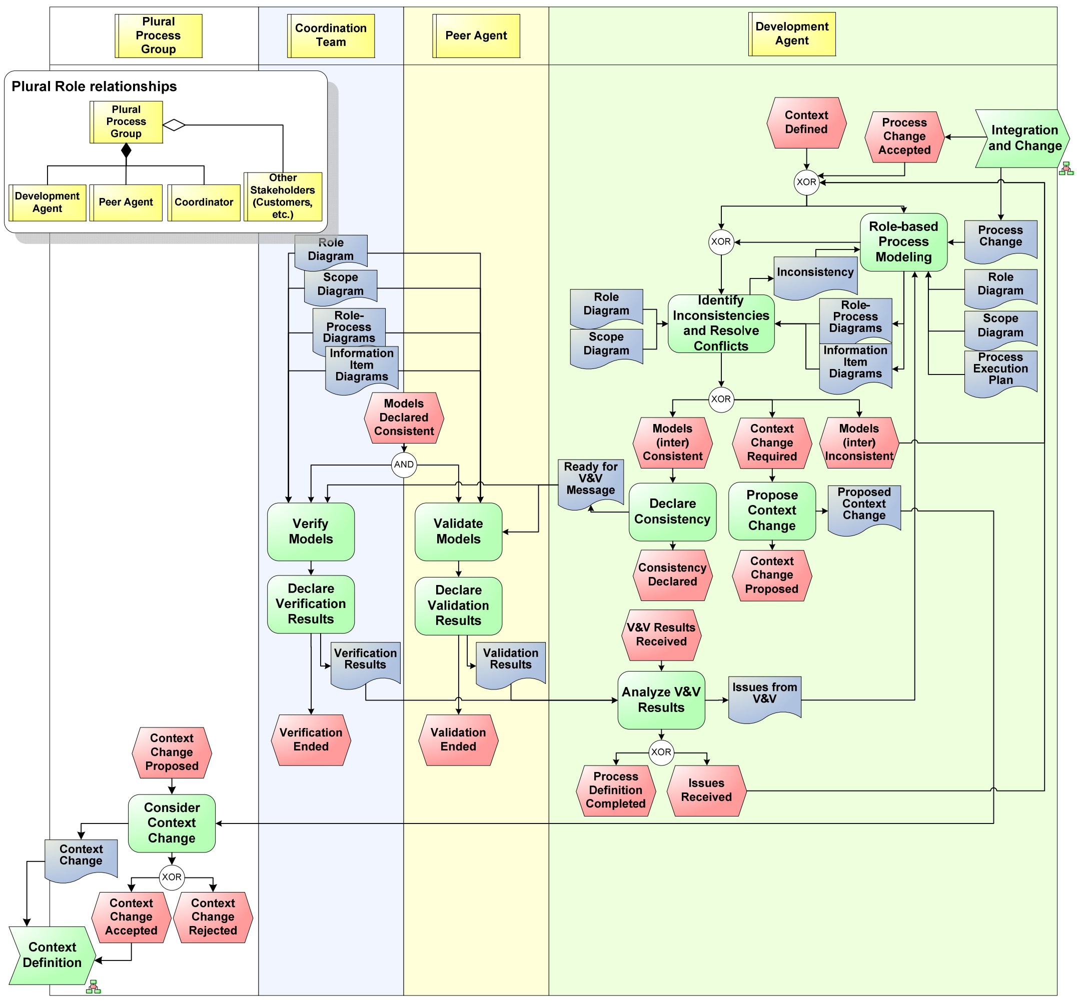


Figure 29. Description and Conflict Resolution Phase

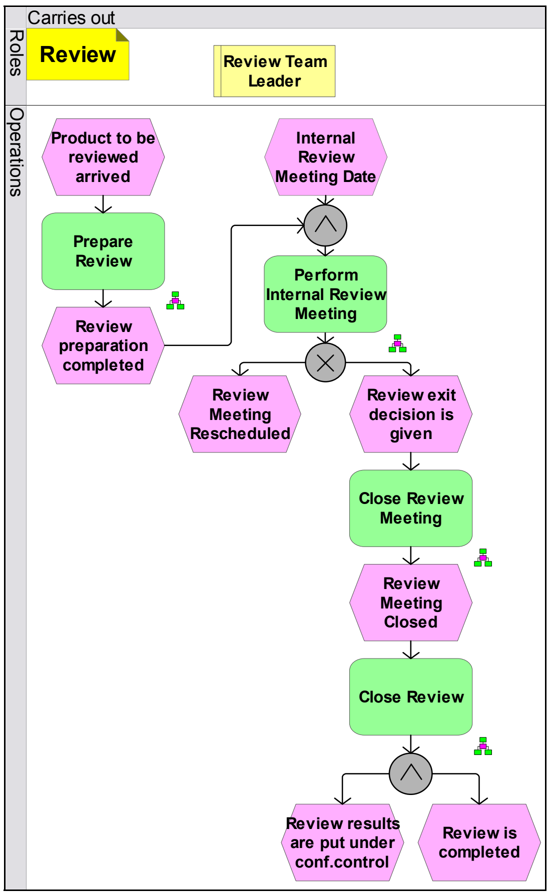
The primary output of this phase is a set of verified and validated individual role-process diagrams. Consistency should be established within (intra-consistency) and among (inter-consistency) role-process models. However, besides these concrete outputs, individual process modeling by each agent in the organization is itself an important and a rewarding artifact of this phase (and the Plural method). It enlightens agents about, the roles they are playing; the activities they are performing; the information they are producing and consuming; and their interaction with other roles. Successful completion of this stage implies that, many of the implicit assumptions and conflicts related with above items are uncovered, shared and solved and a common understanding for activities and artifacts among agents is established.

Section 3.4.1 elaborates the role-based process modeling approach adapted in Plural. Section 3.4.2, describes how inconsistencies among individual models are identified and resolved. In section 3.4.3, validation and verification of process models are discussed.

3.4.1. Role-Based Process Modeling

All development agents, as assigned to one or more active roles, start the description by first identifying the operations of the roles for each process they participate. That is, on an individual role-process diagram, they depict the services these roles provide to others as a set of operations. Modeling can be synchronous or asynchronous and agents can start modeling from any of the roles and processes.

Conceptually, there is a hierarchy between processes, operations and activities each representing a different abstraction level in process's context. A process comprises a set of role operations and an operation has activities in atomic level describing the details of how that role carries out the operation.



Carries out
Roles
Review
Review Team Leader
Operations
Product to be reviewed arrived
Internal Review Meeting Date
Prepare Review
Perform Internal Review Meeting
Review preparation completed
Review Meeting Rescheduled
Review exit decision is given
Close Review Meeting
Review Meeting Closed
Close Review
Review results are put under conf.control
Review is completed

Figure 30. List of operations (process: review, role: review team leader)

Figure 30 gives an example diagram for the operations of the review team leader for the review process. In the example, for the review process the review team leader role defines four operations together with the triggering and ending events for those operations. Figure 44 (pg.99) gives an example of an extended role diagram which represents the operations of the roles for specific processes.

For each of its operations, the role has one or more individual role-process diagrams depicting the behavior of the service it provides. Section 3.6.4 describes the individual role-process diagrams utilized in the phase. In an individual role-process diagram, an agent describes the activities its role performs in that operation, the information items it requires while performing these activities and the outputs it produces. This information forms *role's context*. In addition to that, agents represent the sources of the inputs and the destination of the outputs, if any, to and from its role's activities. That is, the information items in and out of role's context. The sources might represent other roles or items such as project repositories, folders, and software tools or other operations of the same role. Agents also represent the activities their roles perform together with other roles. All these representations of the interaction form the expectations of that role from other roles.

Figure 31 gives the diagram for *prepare review* operation of *review team leader* role in the review process. The diagram has swimlanes (columns) and the role has a primary swimlane in which the activities performed by that role are described.

An individual role-process diagram is consistent in terms of role's expectations if, in the models of the other roles, they answer to these expectations by modeling the expected interface. For example, for the case given in Figure 31, review team leader expected to receive the review request, partly filled review record and the product to be reviewed from the author role to start its activities. That is, these are the three information items that it requires in order to perform that operation. This expectation is considered to be satisfied, if the author role (or its equivalent), in any of its operation defines that it sends these items to the review team leader. Otherwise there is an inconsistency between the expectations of these two roles in terms of their interface. Satisfied expectations is a form of treaty [21] that explicitly determines how and by which means each role wants to interact with other roles.

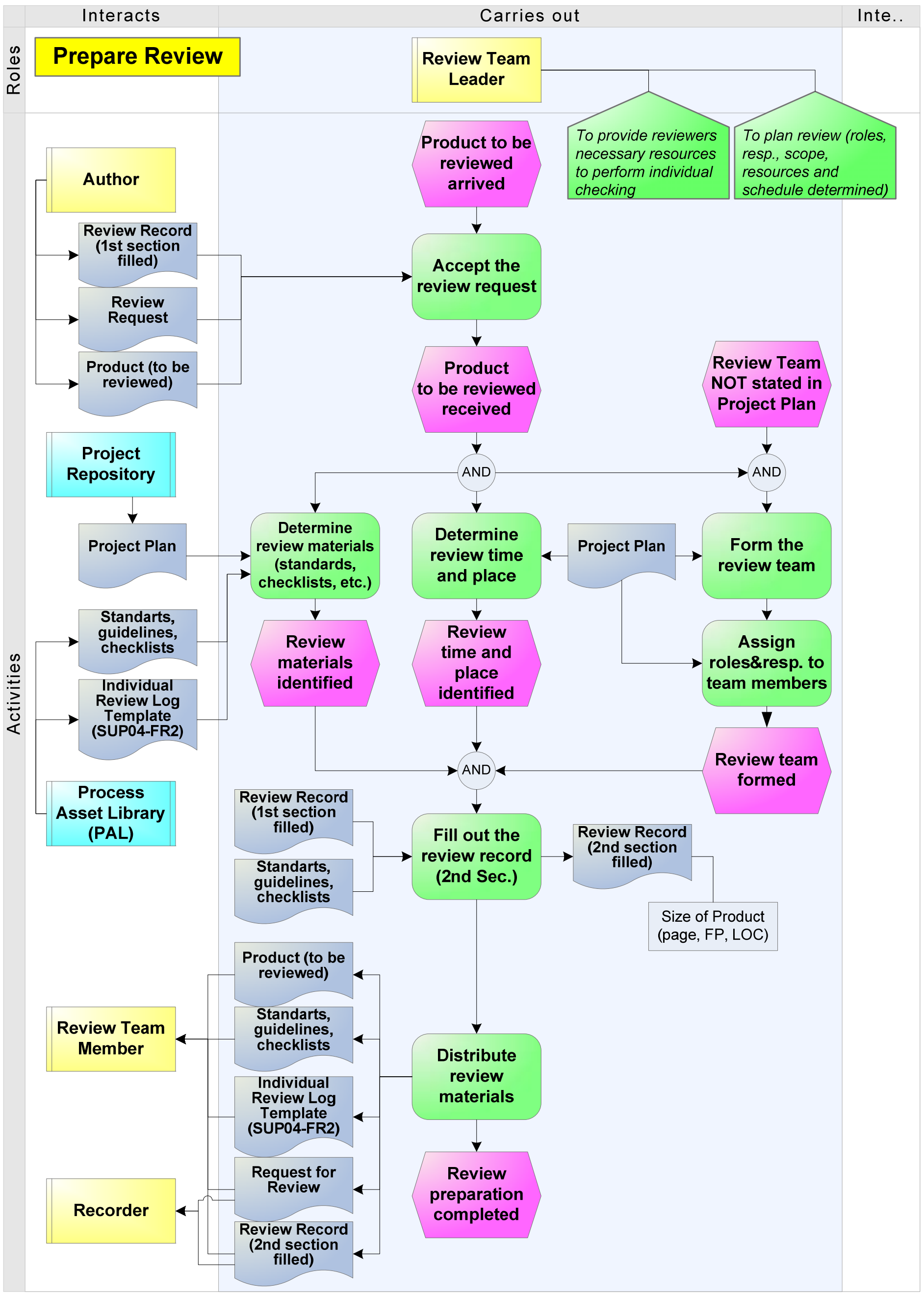

Interacts
Carries out
Inte..
Roles
Prepare Review
Review Team Leader
To provide reviewers necessary resources to perform individual checking
To plan review (roles, resp., scope, resources and schedule determined)
Activities
Author
Review Record (1st section filled)
Review Request
Product (to be reviewed)
Product to be reviewed arrived
Accept the review request
Product to be reviewed received
Review Team NOT stated in Project Plan
AND
AND
Project Repository
Project Plan
Determine review materials (standards, guidelines, checklists, etc.)
Determine review time and place
Project Plan
Form the review team
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Review materials identified
Review time and place identified
Assign roles&resp. to team members
Review team formed
AND
Process Asset Library (PAL)
Review Record (1st section filled)
Standarts, guidelines, checklists
Fill out the review record (2nd Sec.)
Review Record (2nd section filled)
Size of Product (page, FP, LOC)
Product (to be reviewed)
Review Team Member
Standarts, guidelines, checklists
Distribute review materials
Individual Review Log Template (SUP04-FR2)
Request for Review
Review preparation completed
Recorder
Review Record (2nd section filled)

Figure 31. An individual role-process diagram (role: review team leader, oper.: prepare review)

3.4.2. Identifying Inconsistencies and Resolving Conflicts

Consistency identification and resolution is concurrently performed during process description. Rather than performing it in batches or on specific intervals, it can be carried out while development agents modeling their processes. Furthermore, development agents can check expectations of other roles from their roles even before they start modeling processes.

An inconsistency in Plural represents a contradictory expectation present on individual role-process diagrams. The individual models are inter-consistent if all interacting roles fulfill their expectations from each other. Thus, an inconsistency becomes an unsatisfied expectation. In that respect, we presume that there is an inconsistency between two individual process models if the interaction modeled by one of the role is different than the one modeled by the other. The interaction refers to the information (message) received or sent between two roles or an activity performed collaboratively by two or more roles.

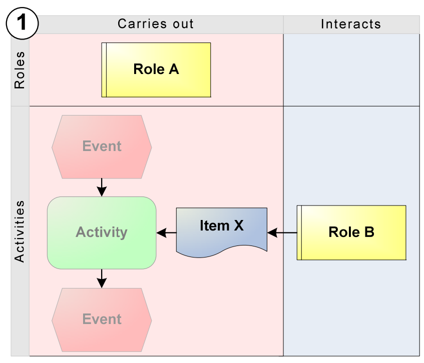
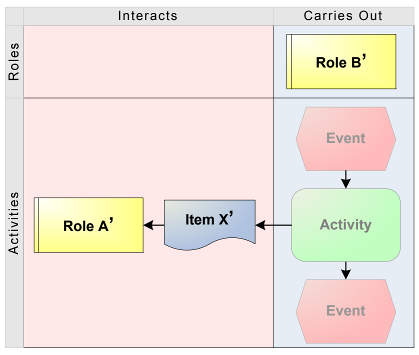
Inconsistency (unsatisfied expectation) identification is an important enabler for the Plural. During process modeling agent playing a specific role checks the interface of other roles in order to validate that other role's expectations are hold by its description. This practice of inconsistency identification is unlike most of the approaches utilized in related works and tools in literature [55], [111], [143], [146]. Most approaches (particularly researches on view-based approaches and multi-view software development) utilize "batch" inconsistency checking [64]. That is, checking inter-consistency after all individual process models are elicited. So, resolving inconsistencies (either manually [146] or tool-guided [55], [143]) is completely a separate activity performed by a process engineer (or a group of process engineers) after all partial process models are complete. Plural's approach to consistency check, however, is proactive. The aim is towards avoiding inconsistencies before they are created by the agents. To increase modeling efficiency, checks should be performed concurrently during modeling, so that agents are made aware of the expectations of other roles in the environment and inconsistencies that would be created due to certain modeling practice. With this approach, considerable time can be saved from rework and consistency resolution.

A tool is necessary to support the identification of inconsistencies as they occur. The method does not offer a protocol for inter-agent communication and negotiation, but, a tool, first, can provide the necessary functions for identifying inconsistencies between individual process models; second can provide functionalities (messaging, chat, and etc.) that facilitate the negotiation to resolve conflicts. The basic requirements and a prototype of such a tool are described in Section 3.7.

In case of conflicts, agents communicate and share related issues. Issues in such interactions primarily comprise the underlying reasons for defining a set of expectations or not satisfying them. In general, agents try to uncover the primary rationale of the others in defining a specific expectation from him or why his expectation is not being fulfilled by others. These interactions to solve conflicts are one of the primary means for organizations that help agents to uncover hidden assumptions and share common understanding.

If both sides insist on their position, they negotiate and resolve conflicts. With an agreement, agents change (re-model) their activities in order to reflect the solution and establish the consistency among the expectation declared in individual models. Automated conflict resolution is not applicable in these cases.

Table 3. Type 1 expectation and its fulfillment

Expectation Type 1:	**Fulfilled by:**
(Role A ← Item X ← Role B) (i.e. Role A receives Information item X from Role B)	**(Role B′ → Item X′ → Role A′)** (i.e. Role B′ sends Inf. item X′ to Role A′) WHERE (Role B′ is Role B OR Role B′ is generalized by* Role B OR Role B′ is composed of Role B) AND (Item X′ is Item X OR Item X′ is generalized by Item X OR Item X′ comprises Item X OR Item X′ is composed of Item X**) AND (Role A′ is Role A OR Role A′ is generalized by Role A OR Role A′ is composed of Role A)
 1 Carries out Interacts Roles Role A Activities Event Activity Item X Role B Event	 Interacts Carries Out Roles Role B' Activities Event Role A' Item X' Activity Event

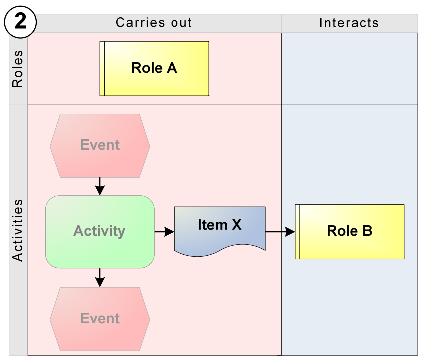
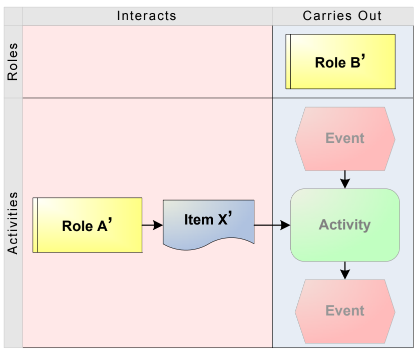
* Generalization and composition relationships are valid for multiple levels, e.g.1 (role A generalizes role B generalizes role C) implies (role A generalizes role C), e.g. 2 (role A generalizes role B comprises role C) implies (role A comprises role C)

** To satisfy the expectation all items comprising Item X including Item X′ should also be delivered to the related role. For example, if developer expects the project plan to be received from the project manager, and if the plan is composed of sections, developer's expectation is fulfilled if all of the sections that make up the plan are delivered to the developer (or equivalent) by the project manager (or equivalent).

Table 3, Table 4 and Table 5 present the three types of expectations and the way they can be satisfied in Plural. For example, the first type of expectation supposes that, role A expects to receive information item X from role B. The process-base would be consistent if, role B, in one of its individual diagrams, states that it sends item X to role A. This is the simplest form of fulfillment of this expectation. Other cases which involve roles' and items' equivalents (in terms of inheritance or composition) also fulfill this expectation. Table 3 describes the conditions that satisfy this expectation.

Type 2 expectations are, in essence, reciprocals of type 1 expectations, because, when a role satisfies a type 1 expectation, it implicitly states a type 2 expectation to the system. Table 4 describes the conditions that satisfy a type 2 expectation.

Table 4. Type 2 expectation and its fulfillment

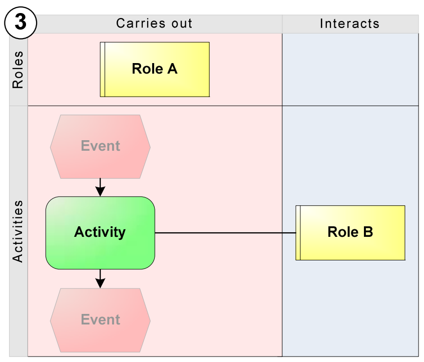
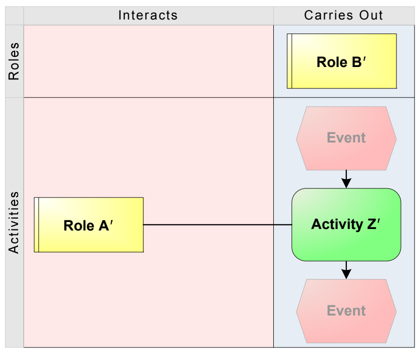
Expectation Type 2:	**Fulfilled by:**
(Role A → Item X → Role B) (i.e. Role A sends Information item Y to Role B)	**(Role B′ ← Item X′ ← Role A′)** (i.e. Role B′ receives Inf. item X′ from Role A′) WHERE (Role B′ is Role B OR Role B′ is generalized by* Role B OR Role B′ is composed of Role B) AND (Item X′ is Item X OR Item X′ is generalized by Item X OR Item X′ comprises Item X OR Item X′ is composed of Item X**) AND (Role A′ is Role A OR Role A′ is generalized by Role A OR Role A′ is composed of Role A)
 2 Carries out Interacts Roles Activities Role A Event Activity Item X Role B Event	 Interacts Carries Out Roles Activities Role B' Event Role A' Item X' Activity Event

* Generalization and composition relationships are valid for multiple levels, e.g.1 (role A generalizes role B generalizes role C) implies (role A generalizes role C), e.g. 2 (role A generalizes role B comprises role C) implies (role A comprises role C)

** To satisfy the expectation all items comprising Item X including Item X′ should also be received by the related role. For example, if project manager expects the project plan to be sent to the developer, and if the plan is composed of sections, project manager's expectation is fulfilled if all of the sections that make up the plan are received by the developer (or equivalent) from the project manager (or equivalent).

Type 3 expectations are related with the activities that roles perform together. Table 5 describes how a type 3 expectation can be fulfilled.

Table 5. Type 3 expectation and its fulfillment

Expectation Type 3: **(Role A → Activity Z ← Role B)** (i.e. Role A performs Activity Z together with Role B)	**Fulfilled by:** **(Role B′ → Activity Z′ ← Role A′)** (i.e. Role B′ Activity Z′ together with Role A′) WHERE (Role B′ is Role B OR Role B′ is generalized by***** Role B OR Role B′ is composed of Role B) AND (Activity Z′ is Activity Z) AND (Role A′ is Role A OR Role A′ is generalized by Role A OR Role A′ is composed of Role A)
 3 Carries out Interacts Roles Activities Role A Event Activity Role B Event	 Interacts Carries Out Roles Activities Role B′ Event Role A′ Activity Z′ Event

* Generalization and composition relationships are valid for multiple levels, e.g.1 (role A generalizes role B generalizes role C) implies (role A generalizes role C), e.g. 2 (role A generalizes role B comprises role C) implies (role A comprises role C)

Appendix A presents the three types of expectations and all possible cases that satisfy these expectations. Since innate aggregation and generalization (inheritance) relationships between roles as well as information items are represented in Plural, consistency checking with respect to the expectations also requires considering such relationships. These relationships between roles are depicted in role diagrams (section 3.6.6) in context definition phase (section 3.3.4). On the other hand, the aggregation and generalization relationships between information items are depicted on information item diagrams by the role that generalizes or aggregates information items. The notation for information item diagrams and examples are given in section 3.6.7 (pg.99). Existence of individual role-process diagrams is mandatory in Plural, but information item diagrams are optional based on type of the relationship that exists between items and whether agents want to utilize and represent this relationship. If an agent feel that it is necessary to use and represent generalization relationship between items, then it is necessary for that agent to represent this relationship on an information item diagram.

An example case for information item and role generalization is as follows: In a software organization, many roles may initiate the review process for several information items. They represent this expectation by sending the item to be reviewed to the review team leader. One way to satisfy these expectations is to expect review team leader representing each of these items in its models as incoming items. Another way for the leader is to generalize these items into a more generic one. In review team leader's context, all those items are just *kinds of* products to be reviewed. Therefore, the team leader can generalize these items and consider that incoming set as the products to be reviewed sent by the project team members and represent it on an information item diagram.

An example scenario to role aggregation can be as follows: Suppose that, on its model, the *project manager* role defines that it sends the *project plan* to the *project team* at some point in project management process. It is defined that the project team is *composed of* team members, such as configuration manager, quality representative, trainer, and etc. who *inherits* project team member's responsibilities. Hence, in checking whether expectations are fulfilled or not, item delivery to the project team should be considered as a delivery to the roles it is composed of or it generalizes; as it is actually happening in real life.

Section 3.7 describes the add-on tool developed for agents to analyze the expectations and possible inconsistencies at anytime during process description. Agents can check whether other roles' expectations from his/her roles are satisfied or not. This is also true for his/her role's expectations from others. At a consistent state where all expectations are fulfilled, both list become identical.

a. Inconsistency and Conflict

There are various reasons for an inconsistency to happen. In case of simple *mistakes* (naming, representation, and etc. mistakes), an inconsistency can be handled without communication taking place between agents. However, if both sides insist on their position, which results a conceptual disagreement, then the inconsistency becomes a *conflict*. In this case, they should communicate and negotiate in order to solve the issue. The coordination team is facilitator to this interaction and can moderate the negotiation unless the conflict is resolved. Agents can bring issues for a discussion that requires an organization wide consensus. Figure 32 displays the amount of effort required to solve the inconsistencies between individual process models (based on [111]).

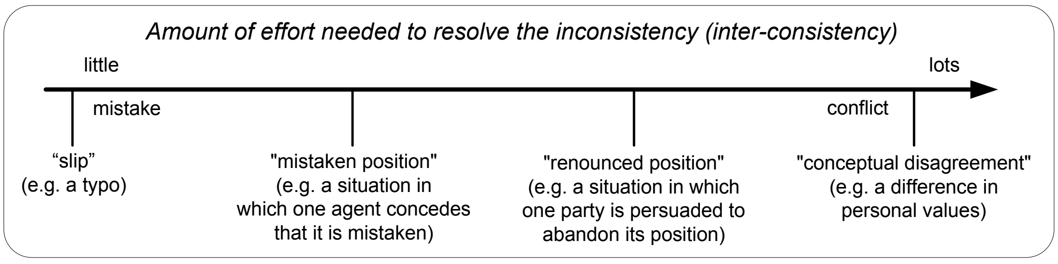

Amount of effort needed to resolve the inconsistency (inter-consistency)
little
lots
mistake
conflict
"slip" (e.g. a typo)
"mistaken position" (e.g. a situation in which one agent concedes that it is mistaken)
"renounced position" (e.g. a situation in which one party is persuaded to abandon its position)
"conceptual disagreement" (e.g. a difference in personal values)

Figure 32. Effort needed to resolve inconsistencies

b. Different granularity levels presented in individual definitions

There is a high probability that the agents may adapt practices that results individual definitions to represent items with different granularity levels. For example, an analyst may expect to send the whole requirements document to the designer, whereas the designer may expect to receive particular portions of that item depending on specific events or conditions (functional requirements, user interface definitions, non-functional requirements, design constraints, and etc.). In essence, these different abstraction level practices adapted by agents manifest themselves as inconsistencies between definitions or unsatisfied expectations of both roles. For these specific cases, agents can pursue either of the following strategies. First, one of the roles can either increase or decrease the granularity level of the representations in order to match other's expectations. For example for the above case, the analyst may decrease its definition's abstraction level and may represent each detail part in the individual definition. The second alternative is to retain the granularity level of the individual representations and depict the aggregation relationship between the item and its parts in an information item diagram. If, for instance, the analyst depicts these relationships between requirements document and all of its parts in a diagram, then these relationships can be taken into consideration in checking whether expectations are fulfilled or not. The choice among these alternative solutions is made by related agents and depends on their insistence to keep the abstraction level of their own definitions.

3.4.3. Verifying and Validating Models

Inter-consistent individual role-process models are validated by peer agents. Peers ensure that role's process descriptions represent role's responsibilities correctly and completely and satisfy role's goals for performing those processes.

Verification of models involves checking process models against the semantic rules specified for individual role-process diagrams. Semantic rules for process diagrams are given in Table 8 (pg.92). The development agents and peers can also perform verification during process description and validation.

In section 3.3.4.a, we described active and inactive roles indicating the ones (active) whose activities are modeled by associated agents and the others (inactive) that are left out of the scope of modeling but interact with active roles by sending or receiving information items or performing activities together. Hence, active roles also have expectations from the inactive roles as well as application systems and information stores. The list of the expectations actually forms the interface between the processes and these entities. For example, analysis of an information store reveals what information items are stored in and retrieved from that information store. Since the expectation-fulfillment mechanism does not exist in these interactions, it is the peer agent's and coordination team's responsibility to ensure that these expectations are valid and consistent.

3.5. Integration and Change (Phase III)

Once the models are correct, complete and consistent, i.e., all the expectations of roles are satisfied within and all individual models are verified and validated, the organization had a set of models that implicitly or explicitly convey a great extent of knowledge, such as; what processes is carried out; which roles participates in these processes and what they perform; what information a role needs; when it needs this information as well as how it acquires it. Therefore, integration and model generation is a matter of querying and questioning the right answers to this process knowledge base. Model generation can be fully or partially automated by a tool which needs presentational abilities to extract this information and present it with various ways to the organization.

Figure 33 depicts the process diagram for this phase. Based on the individual role-process diagrams, process, operation and activity level process diagrams as well as diagrams representing role or process dependencies can be generated. Although the coordination team is responsible for this activity, other agents may also join model integration and generation particularly if the activity is supported by a tool that automates the generation.

The organization at this phase achieves a process-base that comprises diagrams representing the structural frame (scope and role diagrams), individual role-process diagrams that are maintained by agents, and variety of generated models that visualize the way the organization works from different perspectives. Diagrams help agents to acquire the ability to understand the way processes execute, pinpoint problems and inefficiencies, identify improvement opportunities, and recommend changes and improvements.

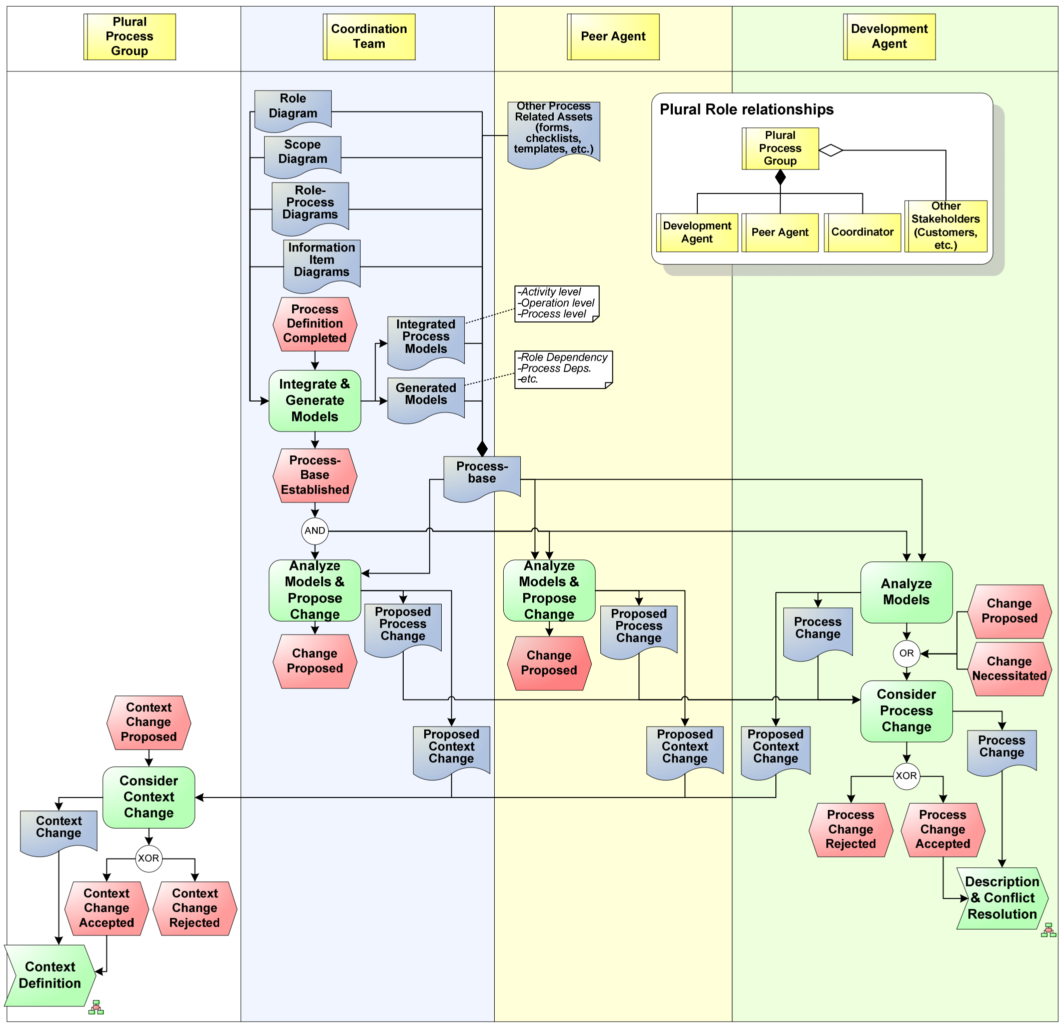


Figure 33. Integration and Change Phase

3.5.1. Generating and Analyzing Diagrams

The underlying base for all generated models comprises consistent and complete individual role-process diagrams, role and scope diagrams and information item diagrams (if exist). Each generated model is a query to the process base that visualizes a portion of the processes from a specific perspective. In that respect, model generation is taking a snapshot of that perspective of the organization's process-base at any point in time. The generated model is valid until a change is performed to the models that form the base for the generation. In this case, the model should be re-generated in order to reflect the change.

Generated process diagrams visualize the process from different abstraction levels and provide a big picture of the activities performed by related roles. The first type of process diagram that can be generated is the activity level process diagrams that integrates individual role-process models into a process model that depicts all the activities performed in that process at the lowest level of detail. The integration for the these diagrams is simply putting each individual model of that process side by side and joining them with the messages the roles sent to each other and the activities they perform together. Operation level process diagrams are synthesized versions of their activity level diagrams depicting the roles' operations and their data dependency between them in a specific process. Process level diagrams depict the data transfer between a specific process and other roles, application systems and information stores as well as processes that extend the process or are included into it. The generation of these diagrams is described in section 3.6.10.

Dependency diagrams address the interactions. Role dependency diagrams present a set of roles and their relationship, in terms of the information items, between each other and application systems/information stores. The process dependency diagram depicts the message exchange between processes as well as the interface to external roles, information stores and application systems.

Generated models provide a macro view of the processes performed by the organization. Agents analyze the way the organization operates, identify problems and drawbacks of current execution and investigate improvement opportunities. Role dependency diagrams are conceptually similar to Yu's actor relationship models [159] and Katzenstein's dependency diagrams [88]. Together with process dependencies they help the organization to understand the interactions and interdependencies existing between roles and the implications of changing these relationships as well as to identify and compare alternative executions.

3.5.2. Proposing and Handling Context Changes

Particularly during process description and inconsistency identification and resolution, agents can recognize deficiencies or problems related to the context. An example can be given for a case where all associated roles for a process define the activities they perform in that process but the agents (particularly coordinators) recognize that process goal is not fully addressed by related individual definitions. That is, the definitions for each role does not fully cover all the activities that are expected to be performed to fulfill the overall process goal. This can be due to a missing role that should have been defined in context definition phase or an incomplete or incorrect process definition. It can also be an indication of a problem in the way the current process executes.

If a change is related with the roles participates in processes and their relationships or the processes covered and their relationships with each other and with roles – macro level changes-, then agents can propose change requests for the context. Process group decides on the acceptance or rejection of these changes in context. If accepted, the change is implemented by the coordination team. Dependent on the scope of the change and its impact, the process description can be paused and the process group returns back to the context definition phase and the execution restarts.

3.5.3. Proposing and Handling Process Changes

Process changes involve the modifications on the way roles perform their tasks and on the artifacts they own – micro level changes. These changes are reflected on individual role-process diagrams by related development agents. Changes can be raised by any agent in the organization and can be related with other roles' processes or roles they own. It can be raised through direct communication with agents or indirectly by modifying or adding any expectation from that role owned by agents. For example, a change request on role A's processes can be raised by other roles that interacts with role A. They can incorporate an additional expectation from role A and an unfulfilled expectation results an inconsistency in the system which should be handled immediately.

Changes related to individual role definitions are directly performed by the related agents that are playing that role. The change is then reviewed by its peer agent and coordinators. With respect to the principle of information hiding, if a change did not affect role's interface, then it is an alteration in role's context which does not affect other roles' expectations and the way they perform their tasks. It is role's responsibility to perform its activities and produce its artifacts in one way or another to fulfill other's expectations. However, if an update modifies role's interface (thus expectations) with neighbor roles, then that change should be incorporated in all related models or it should be revoked after a negotiation between parties. These cases manifest themselves as inconsistencies between expectations which should be resolved in related models. A change request is a starting spark that triggers other changes and this wave of change does not settle until all effected individual models are updated and in turn they reach to another consistent state again.

3.6. The Notation for Plural

A business process modeling method requires a notation to describe the processes and related structures. This section describes the notation that satisfies the requirements of the Plural method. Section 3.6.1 describes basic requirements of the notation. Sections 3.6.3 through 3.6.9 present the conceptual framework.

3.6.1. Basic Criteria for the Notation

In section 2, we surveyed several process modeling notations/languages utilized in diverse fields including software engineering, business process management, and etc. Based on the method proposed in this study, this section identifies the key characteristics of a notation that can be utilized for Plural.

Process models in general try to answer the following questions: *what* is going to be done, *who* is going to do it, *when* and *where* will it be done, *how* and *why* will it be done [36], [132]. If process elements, i.e, the constructs of the notation, are able to answer all of these questions, the model is assumed to be complete, consistent and integrated [36]. From a perspective, we can differentiate process modeling notations with respect to the extent to which their constructs emphasize the information that answers these questions. Process modeling notations generally try to answer one or more of these questions. Curtis et.al. [36], describes four commonly represented information perspectives as:

- *Functional* represents what process elements are being implemented and what information entity flows are related to these process elements. (i.e. represents the set of activities performed, their decomposition into sub-activities, and the artifacts used or produced by these activities. e.g. Data Flow Diagrams (DFD) [77])
- *Behavioral* represents when or under which conditions the process elements are performed. According to Curtis, this perspective also deals with the aspects of how they are performed through feedback loops, iteration, complex decision-making conditions, entry and exit criteria, and etc.
- *Organizational* represents where and by whom (which actors) in the organization process elements are implemented.
- *Informational* represents the information entities created or manipulated by a process, including their structure and relationships.

In addition, we can differentiate between the questions the behavioral representation scheme tries to answer. As specified above, behavioral perspective focuses on ‘when/under what condition will the process be done’ and ‘how will it be done’. Demirors [40] calls the later approach *processual* representation and gives an example to differentiate between these two. The section ‘how to install a software’ in its manual is a processual explanation whereas the troubleshooting section that explains ‘what to do in case of certain problems’ is a behavioral explanation.

Two different attitudes to process modeling are *descriptive* and *prescriptive* approaches. Descriptive methods study existing processes to answer the question ‘how the product is (or has been) actually developed or produced’ [101]. Prescriptive methods, on the other hand, define desired processes to answer the question ‘how the product should be developed or produced’. The goal is to define the required or recommended means of executing the process. Examples for prescriptive approaches include the manual approaches such as RUP (Rational Unified Process) [94] and ISO/IEC 12207 [76] or automated approaches like MARVEL [84]. The process modeling notation for Plural should be suitable for descriptive modeling, because, we want agents to model the actual execution of the activities. We believed that understanding what is actually being done is the first step to be able to improve it [16]. It should support describing behaviors instead of prescribing step by step processes as in most other process modeling notations [40].

In general, prescriptive approaches are implemented with processual representations (APPL/A [133, 134]). Similarly, descriptive models are generally represented with behavioral approaches (HOC-N [40]).

Plural requires agents to develop models of their conduct with respect to the roles they act for. In that respect, representing roles' behavior under certain conditions is vital. Thus, the notation should possess behavioral and organizational representations capability. In addition, each agent is required to represent its role's expectations as a set of information entities it requires. In that scheme, the structural relationships between these information entities become crucial. This requires the notation to represent the informational perspective of the processes as well.

In general, process models serve for the following purposes:

- Supporting *execution* of the process; where a process model is a program to enact the activities in an environment.
- Support for *monitoring* the execution of the model.
- Support for *controlling* the execution.
- Support for *measuring* the actual execution with respect to the baseline process model.
- Supporting *improvement* in the process.
- Support for better *understanding* and *learning* the actual execution.

A process model may serve one or more of these purposes. A process model for execution support may well serve for monitoring and control, and measurement can be useful for monitoring as well as understanding and improving the process.

The granularity of process models is highly correlated with the purpose they serve for [40]. Notations that emphasize execution, monitoring and control usually yield detailed models, while notations that give emphasis to understanding and learning usually target focused models, which try to capture the essential elements of the processes.

Although the Plural method does not constrain or indicate what purposes the process models defined by the Plural will serve, mainly due to its structure and requirements, its emphasis is implicitly more on models that will serve to facilitate human understanding and communication in organizations and support learning and improvement. Particularly when process improvement is the main concern, the method should enable organizations to surface implicit assumptions and expectations of the agents and make them explicit. The support for execution and control is weak. Therefore, the notation for Plural should give emphasis to understanding, learning and improving (vs. measuring, enactment, control) of processes.

The centralized development of the models, where the development is performed by a group or an individual, is predominant in process modeling methods and related notations [7], [29], [91],

[150]. On the other hand, the decentralized process modeling in Plural requires the description of separate and partial models of the behavior for each role and integrating these models to form the complete description of the process. The integration points are mainly the information (messages) these roles interchange during process execution. Thus, the notation for Plural should support the development of the separate models and enables each agent, with respect to the roles it plays, to represent the set of expectations from other roles mainly in terms of the information and resources it requires.

As a summary, the notation for Plural should possess the following properties:

1. It should be suitable for descriptive modeling
2. It should be able to represent behavioral, organizational and informational perspective of the processes
3. Its emphasis should be on understanding, learning and improving
4. It should be suitable for decentralized and concurrent modeling. This implies that it should enable agents to represent their processes and expectations separately in terms of the information they require and with respect to the roles they play.

3.6.2. Applicability of the Available Notations for Plural

Many of the notations we surveyed in section 2, satisfy the first three requirements with little or no extensions. However, all these notations we surveyed assume that related models will be created by a central unit, thus they lack the support for decentralized and concurrent development.

Among the process modeling languages we reviewed in section 2.3 (which are generally utilized in PSEEs), the ones that are based on rule-based formalism, such as MSL (Marvel Strategy Language) utilized in Marvel and Oz systems are better candidates than the ones based on programming languages and Petri-net formalism. The representation of rules in the form of condition-actions is suitable for the representation of behavioral aspects as well as modeling descriptive specifications. However, these languages should enable visual constructs in order to be efficiently utilized by end users.

Event based notations such as eEPCs are analogous to rule-based models. Notations such as BPMN, eEPC, and RADs provide extensive support for the representation of behavioral aspects and are suitable for the descriptive modeling. In addition, they can be used to create models that can be used to foster understanding and improvement in organizations.

However, the notations as well as related methods we evaluated require a considerable degree of refinement and extension to be used for business process modeling where role behaviors are created and maintained individually and independently.

In the next section we describe the notation we propose for Plural. For representing the behavior of the organization, we utilized process diagrams that are in large based on the eEPCs. eEPCs provide a better fit to above requirements mainly due to their strong support for the representation of multiple aspects of the processes, behavioral, organizational and information in particular. In addition, rule-based structure for representing pre- and post conditions for activities was also suitable for representing partial and independent representations of role behavior. We also utilized object oriented concepts for representing such relationships as inheritance and composition between roles as well as information items.

3.6.3. The Structure and Elements

The Plural notation extends concepts including eEPCs (extended Event Driven Process Chains) in ARIS (Architecture of Integrated Information Systems) method [74], and use case and class diagrams in UML (Unified Modeling Language) [113] with respect to the unique requirements of the approach.

It comprises a set of diagrams and associated process elements for representing the organization and its processes from different perspectives. This section presents the details of the process elements and their relationships and introduces the diagrams. Each diagram is then described in a separate section.

Each diagram expresses different aspects of the organization's process knowledge:

- *Scope Diagram* represents the high level processes covered, their relationships and the roles participates in these processes.
- *Role Diagram* represents the structural relationships between roles.
- *Process Diagram* represents the behavior of the processes.
- *Information Item Diagram* represents the static relationships between information items.
- *Role Dependency Diagram* illustrates the information items exchanged between roles.
- *Process Dependency Diagram* shows the interactions of processes in terms of the information items used and produced.

The role diagrams and information item diagrams are used to model the structure while the others are used for representing the behavior of the organization.

Any component of a process is a process element [36]. It is the fundamental unit of a process. Each diagram has a specific set of process elements and their relationships with each other. Table 6 presents the elements, their semantics and symbols in Plural. The details of how these elements are associated in each diagram are given in subsequent sections. In addition, some of these elements are elaborated in section 3.3.3 when describing how they are perceived in Plural.

Table 6. Process Elements

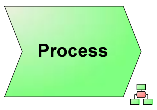
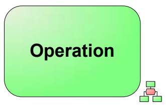
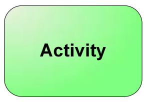
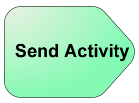
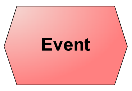
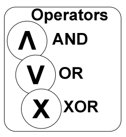
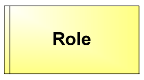
Symbol/Element	Description	Example	Appears in
	(Business) Process is a set of activities that supports one or more business goals. The term is discussed in more detail in section 3.3.3.	Project management, Employee recruitment, Process modeling, Process Assessment	- Process Diagram - Scope Diagram - Process Dependency Diagram
	An operation is a set of activities that represents the services that a role provides with respect to a particular process.	Initiate Review, Perform Individual Checking, Pre-review Applicants, Perform Interview	- Process Diagram
	An activity is a logical step within a process. It takes an object in a particular business context and changes its initial status to another distinguishable one to support one or more business objectives.	Review a Document, Write a message, Create a Record, Update a Report, Archive a Document,	- Process Diagram
	Send activity is a special kind of activity where the inputs are transferred (carried to) (rather than transformed or used to create outputs) from a designated location to another. These activities form the interface points between the role and the rest of the world in terms of the information it sends. All information items go through these activities if they are to be delivered outside the context boundary of related role.	Send a message/ document, Hand over a form, Store a document to a Folder	- Process Diagram
	An event represents a state of an object that is relevant in terms of business context, which controls or influences the further procedure of the business process. Events trigger activities and are the results of activities. An activity is a time consuming occurrence while an event is related to one point in time	Document reviewed, Message sent, Customer arrived, Message received, Application accepted	- Process Diagram
	Logical operators (rules) link events and activities in process chains		- Process Diagram
	Roles are the brief summary of a set of functional responsibilities in relationship to a particular aspect of the business. Agents perform tasks with respect to the roles they are playing in that particular aspect. It behaves under its bound roles. They participate in a particular process by performing all or a number of its activities. Roles can be played by an agent of an individual, a team, a group of people or by a software system. The term is discussed in more detail in section 3.3.3.	Configuration Manager, Secretary, Student, PhD. Student, Instructor, Customer, Design Department, Company X, Project Team, Development Team, Configuration Control Board, Finance system, Antivirus software	- Process Diagram - Role Diagram - Scope Diagram - Role Dependency Diagram - Process Dependency Diagram - Information Item Diagram

Table 6. Process Elements (continued)

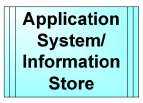
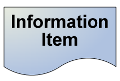
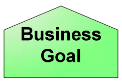
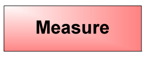
Symbol/Element	Description	Example	Appears in
	The information store represents the places where information is stored and retrieved. This includes not only physical places like libraries, shelves or books, but also hardware, application systems, web sites or any other software that agents interact with. Application systems, in that sense, are a kind of information store. They can also be used during process execution to support the activity that is being performed by the agent. In addition, in case of software systems, an information source may well be treated as a role that is programmed to perform certain activities or participate in a process on the same level any other role does (such as backing up the database, performing virus scans on specific intervals, and etc.).	Finance system, Human resource management system, CASE tool, Project management tool, Web site, Library, Process repository, Project bookshelf, Operating system directories, Antivirus software	- Process Diagram - Role Diagram - Scope Diagram - Role Dependency Diagram - Process Dependency Diagram - Information Item Diagram
	Information item represents a means to store and transmit information. It can be in the form of a document, an email, a fax, a CD, or a *verbal* message that is produced out of an activity, input to be processed by an activity or a resource retrieved from an information store. An information item can be: - an *input* to a process and consumed during the processing; or - an *output* as a final object produced or resulted out of that process; or - a *resource* that is fed into a process and used as a part of the transformation process. Unlike inputs, information resources are not consumed. Templates or standards are examples of this kind. A software requirements specification document can be 'input' for 'software design' process, while a specific design template can be 'information resource' for that activity.	Purchase order, Message of acceptance, Software Requirements Spec. (SRS), Review form, List of issues, Notification email, Checklist, Application form.	- Process Diagram - Information Item Diagram - Role Dependency Diagram - Process Dependency Diagram
	Goals are reasons for actions. They are desirable and measurable states that either the overall process or individuals in the process intend to achieve. The overall goal of a business process can be decomposed into sub-goals and achieving these sub-goals can be achieved by assigning them to roles. By playing roles, agents take over goals to achieve. the model represented each person's issues individually.	To increase sales, To increase product quality, To reduce number of bugs, To decrease time to market,	- Process Diagram - Scope Diagram
	Measures represent standard units, specified by a scale, used to express size, amount, or degree. They are normally quantitative in nature capturing numbers, duration, percentages, and etc.	Number of defects by severity, Number of person hours used, Number of pages reviewed, Mean time to failure, Lines of code	- Process Diagram

Table 6. Process Elements (continued)

Symbol/Element	Description	Example	Appears in
Agent	Agents represent named individuals, teams, departments or other software systems in the organization. Human agents occupy one or more organizational units like positions, or represent teams or groups of people, and playing one or more roles. The term is discussed in more detail in section 3.3.3.	Ali Yilmaz (In project manager position, taking roles as project manager, acquisition manager and leader of the directing board in related processes), Selma Kalfa (in software engineer position, taking roles as developer in one project and test leader in another), Finance department (acting for the finance dept. role in a process)	- Role Diagram

Table 7 presents the complete set of relationships between process elements.

Table 7. Element Relationships

Rel.No	Relationship	To	From	Description	Appears in
Rel.01	is composed of	Process	Operation, Activity	A process may comprise operations which may also be composed of activities that are atomic	- Process Diagram (not shown explicitly)
		Operation	Activity		
Rel.02	is predecessor of	Activity (Operation / Process)	Activity (Operation /Process)	Activity A is predecessor of activity B implies that B is performed only after A is completed	- Process Diagram
Rel.03	is input for	Information Item	Activity (Operation /Process)	An information item that is input to an activity represents that it is consumed or used as a resource (e.g. guideline, checklist, and etc.) during process execution	- Process Diagram - Role Dependency Diagram - Process Dependency Diagram
Rel.04	creates output to	Activity (Operation /Process)	Information Item	The execution of the activity outputs the information item as the final product of that activity	- Process Diagram - Role Dependency Diagram - Process Dependency Diagram
Rel.05	sends	Send Activity	Information Item	Represents the transfer of information	- Process Diagram
Rel.06	carries out (participates)	Role	Activity (Operation /Process)	Role is responsible from the execution of the activity or participates in the performance of a process	- Process Diagram - Scope Diagram
Rel.07	provides	Role	Information item	Provides relationship represents a transfer of the information item from the role. The role is active in the transfer activity, that is, the transfer is performed by the role.	- Process Diagram - Role Dependency Diagram - Process Dependency Diagram

Table 7. Element Relationships (continued)

Rel.No	Relationship	To	From	Description	Appears in
Rel.08	provides	Application System / Information Store	Information item	Similar to the transfer of the item from a role, the item can also be provided by an application system or information store. However, unlike roles, they are passive in the transfer activity. The transfer is performed by another role that is outside the context of the relationship.	- Process Diagram - Role Dependency Diagram - Process Dependency Diagram
Rel.09	is used by	Information item	Role	The relationship represents the transfer of the information item to a role, which uses it to perform a specific activity.	- Process Diagram - Role Dependency Diagram - Process Dependency Diagram
Rel.10	is stored in	Information item	Application System / Information Store	The information items are stored in an application system or an information store to be retrieved for further needs.	- Process Diagram - Role Dependency Diagram - Process Dependency Diagram
Rel.11	has	Role	Goal	Role pursues a specific goal which is the reason to perform specific activities	- Process Diagram
Rel.12	measures	Role	Measure	Role determines and quantifies measures during the execution of a process	- Process Diagram
Rel.13	is captured in	Measure	Information item	Measures can be captured in an information item such as a document, form, and etc.	- Process Diagram
Rel.14	supports	Application system	Activity (Operation /Process)	An application system can be used by a role to perform an activity	- Process Diagram
Rel.15	activates	Event	Activity (Operation /Process)	An event triggers an activity	- Process Diagram
Rel.16	creates	Activity (Operation /Process)	Event	Execution of an activity results an event	- Process Diagram
Rel.17	leads to	Activity (Operation /Process)	Logical operators	An activity can be connected to a logical operator if it results multiple events.	- Process Diagram
Rel.18	activates	Logical operators	Activity (Operation /Process)	A logical operator triggers an activity (where in that case it is a point that connects a number of incoming events to an outgoing activity)	- Process Diagram
Rel.19	is evaluated by	Event	Logical operators	An event can be connected to logical operator which will eventually trigger an activity	- Process Diagram
Rel.20	leads to	Logical operators	Event	A logical operator can lead to an event (where in that case it is a point that connects an incoming activity to a number of outgoing events)	- Process Diagram
Rel.21	supports	Activity (Operation /Process)	Goal	Activity supports a specific goal to be realized or the goal is the reason for the activity to be performed	- Process Diagram - Scope Diagram

Table 7. Element Relationships (continued)

Rel.No	Relationship	To	From	Description	Appears in
Rel.22	results	Measure	Activity (Operation /Process)	Measures are the results of an execution of a activity	- Scope Diagram
Rel.23	includes	Process	Process	When process includes another one, the included one is unconditionally incorporated into the execution of the one that includes it.	- Scope Diagram
Rel.24	extends	Process	Process	The process that is extended by another one conditionally incorporates that process in it. That condition is dependent on the runtime execution of the extended process.	- Scope Diagram
Rel.25	generalizes	Process	Process	When a process generalizes another one, the specific process that generalizes the other inherits the behavior of the generalized process where it can alter it by including other activities, roles, and etc.	- Scope Diagram
Rel.26	association	Role	Role	An association relationship is a simple structural connection between roles. For example; manager role 'leads' the project team, the instructor 'teaches' the student, and etc.	- Role Diagram
Rel.27	comprises	Role	Role	Comprises (or is composed of) is a special kind of an association relationship denoting the whole/part within which one or more roles are parts of a larger whole. Comprises can be of two types; aggregation and composition. Composition is a stronger than aggregation in the meaning that the whole and the parts in composition have coincident lifetimes. This means that the part cannot exist without the whole. For example, review team leader role might not exist without the review team role (composition); whereas, this might not be the case for the developer role and the development team (aggregation)	- Role Diagram
Rel.28	generalizes	Role	Role	Generalization between roles refers to a relationship between a general role and a more specific version of that role, in which specific role inherits the operations of the general role, yet it has responsibilities specific to it. Specific role can be considered as being a 'kind of' the general role. For example, PhD Student and MS Student roles inherit the responsibilities of the Student, which is a more general role. Similarly, project team member is a general role to the developer and tester roles	- Role Diagram
Rel.29	plays	Agent	Role	An agent plays one or more roles while participating in a process and performing an activity	- Role Diagram

Table 7. Element Relationships (continued)

Rel.No	Relationship	To	From	Description	Appears in
Rel.30	is peer for	Agent	Agent	An agent can be a proxy for another agent.	- Role Diagram
Rel.31	association	Information Item	Information Item	An association relationship is a simple structural connection between information items. For example; project plan 'is associated with' the review record, the software design document 'is associated with' software requirements specification document and the software design document template, and etc.	- Information Item Diagram - Role Dependency Diagram - Process Dependency Diagram,
Rel.32	comprises	Information Item	Information Item	Comprise represents the whole/part relationship where one or more information items are parts of a larger whole. Comprise relationship can be in *aggregation* or *composition* type. Aggregation denotes optional existence of the part in the whole whereas the existence of the part in composition is mandatory. For example, the thesis manuscript must contain the introduction section in it while the appendix section is optional.	- Information Item Diagram - Role Dependency Diagram - Process Dependency Diagram - Process Diagram
Rel.33	generalizes	Information Item	Information Item	The generalization can be considered as a 'is a' or 'a kind of' relationship between information items. It is a relationship between a general information item and a more specific type of that item.	- Information Item Diagram - Role Dependency Diagram - Process Dependency Diagram - Process Diagram

3.6.4. Process Diagram

Process diagrams play a significant role in Plural. Each role has one or more individual role-process diagrams depicting the activities it performs in relation to a process. Complete and consistent individual role-process diagrams are merged to form the integrated process diagrams representing the whole picture for a specific process with different abstraction levels. With respect to the abstraction levels and usage, four types of process diagrams are as follows:

1. Individual role-process diagram
2. Activity level process diagram
3. Operation level process diagram
4. Process level process diagram

Process diagrams are primarily based on eEPCs (extended Event Driven Process Chains) described in section 2.7.2. In number of business process modeling projects we observed that users found it not very difficult to understand and use eECPs and got quickly adapted to its semantics [42]. These characteristics of eEPCs made them suitable candidates for satisfying the unique requirements of Plural.

A subset of eEPC process elements and their relationships is adapted in Plural. In addition to this subset, two process elements (measure and sent activity) and their connection properties are introduced as an extension. Besides, the approach needed to use eEPC diagrams in a different way than the way they are generally utilized in practice. Similar to the structure in tabular collection diagrams used in CORE [108], [111], the sources of the inputs and the destination of the outputs to and from activities are also represented on eEPCs. The sources are either other roles or information stores such as software systems, libraries and etc. For process models, we use eEPCs with column display (columnar eEPCs) where each column or swim-lane represents the portion of a process performed by a specific role.

Figure 34 displays the process elements and their relationships that exist in process diagrams. The numbers near to the relationships represent the unique numbers of the relationship which are presented in Table 7. The direction of arrows represents the way the relationship name should be read (e.g. role → carries out → activity).

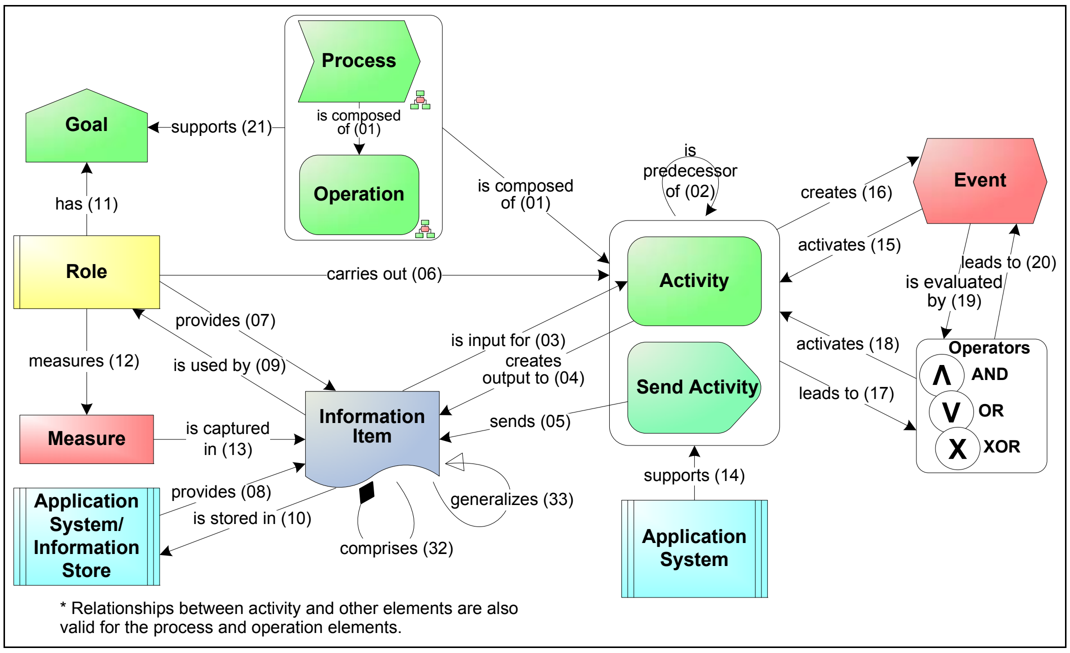


Figure 34. Elements and relationships in a process diagram

A process diagram can be verified with respect to a set of semantic rules that are listed in Table 8.

Table 8. Semantic Rules for Process Diagrams

Rule No	Rule	Description
SR.01	Each path in the diagram must begin with an event and end either with an event or a send activity.	This rule implies that all paths of a selected model (event-activity-event chains) must begin with an object occurrence of the event type and end with an object occurrence of either the event or 'send activity' type.
SR.02	All activities and events have only one incoming/outgoing connection	This rule implies that all functions and events of a diagram must have a maximum of one incoming/outgoing connection. (e.g. two events triggering an activity can not be connected to that activity directly but must be connected via a logical operator of type and, or, xor.)
SR.03	Order at the operator must be preserved	For a logical operator, when all incoming connections come from activities, then all outgoing connections may only lead to events.
SR.04	No cycles may exist between an element and an activity	There should not be two connections (one outgoing and one incoming) between a specific event and an activity.
SR.05	Triggering and ending events for processes and operations must be preserved in lower level diagrams assigned to them	Events that triggers or ends a process or an operation must also exist in the lower level diagrams that depict them, and vice versa. For example, 'order arrived' event triggering the 'initiate order processing' operation in a process diagram should also exist as a triggering event for an activity in the lower level diagram assigned to the 'initiate order processing' operation. Vice versa is also true, i.e. an ending event in the lower level diagram that describes initiation of the order should also appear as an ending event for that operation in higher level process diagrams.
SR.06	An information item cannot be both input and output from an activity.	An information item that is input for a specific activity cannot be connected to the same activity as an output.
SR.07*	Sources and destinations of the information items must be represented.	On an individual role-process diagram, on the first occurrence of an information item input to an activity, the source of the item that it is gathered from (another role, an application systems/information store, or another process) must be represented. On subsequent occurrences of that item, the representation of the source is optional. Similarly, an information item output from an activity must be input to a subsequent activity (or process) on the diagram, send to a role, or stored in an application system/information store.
SR.08*	Only one active swimlane may exist on an individual role-process diagram	On an individual role-process diagram, activities and events may be placed only on the active swimlane that belongs to the role that owns that diagram.
SR.09*	An activity connected to two or more roles should be preceded and followed by an event.	On an individual role-process diagram, an activity that is performed by two or more roles should be preceded by an event and similarly should be followed by an event. This is a requirement related to the integration of the individual role-process diagrams.

** Applicable only to individual role-process diagrams*

a. Individual role-process diagram

Individual role-process models are used by agents in description and conflict resolution phase (section 3.4), in which each agent playing a specific role in the process, models its own activities. Each individual process diagram has a primary swim-lane for representing the activities of the role that owns the diagram. There should not be any activity on other swim-lanes except the primary one. In integrated process diagrams, on the other hand, there is one swim-lane for each role participating in the process. A swimlane owned by multiple roles indicates that the activities in that swimlane are performed by these roles together concurrently. Figure 39 gives an abstract example of an individual role-process diagram utilized in Plural.

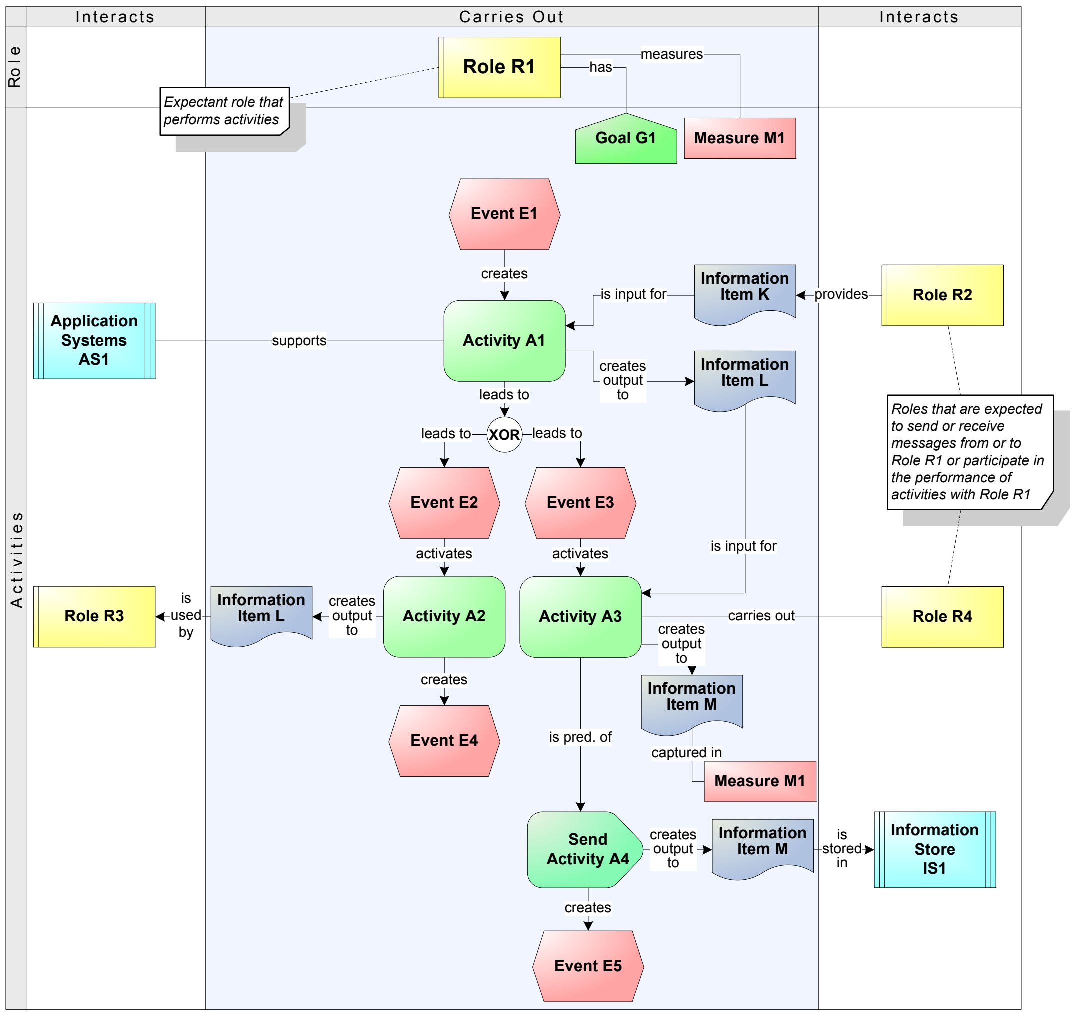
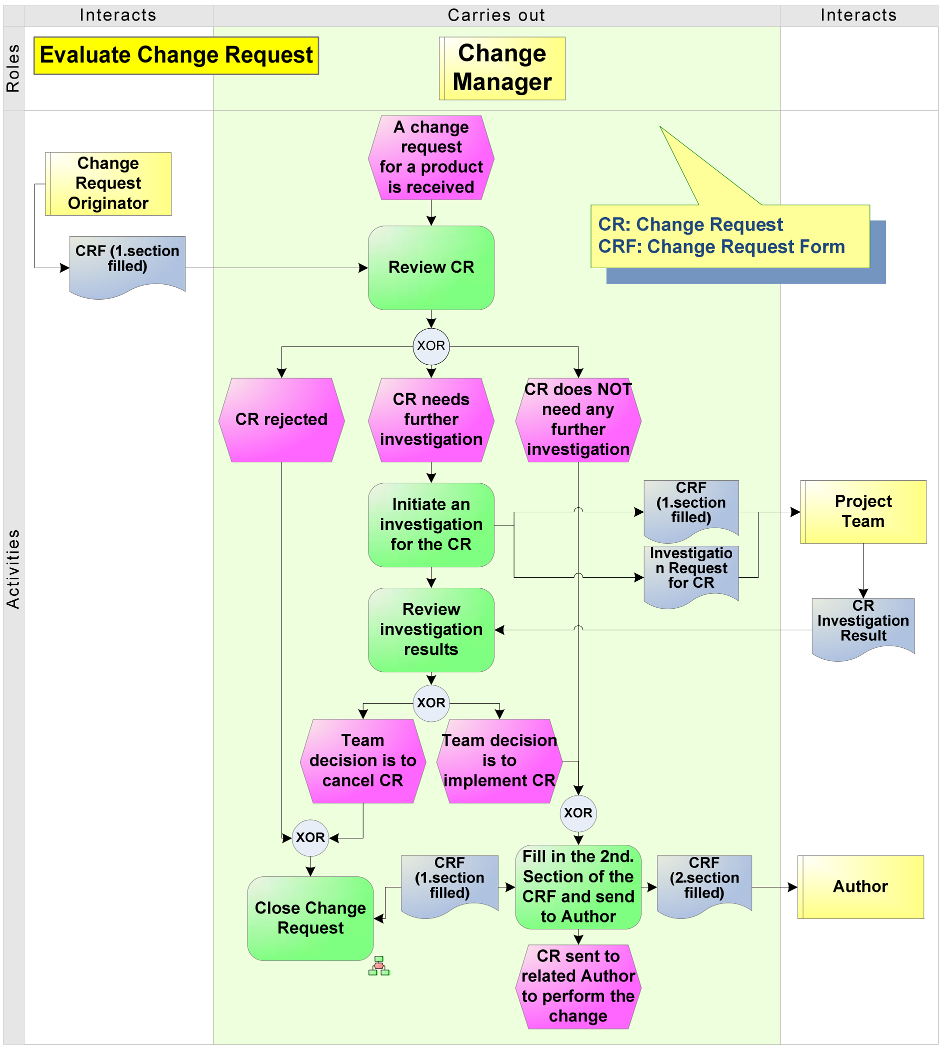


Figure 35. An abstract individual role-process diagram

Figure 36 displays an example of a role-process diagram for *change manager* role for *evaluate change request* operation in the *change management* process. In the diagram, change manager presumes that it receives (acquires) an information item (change request form - 1. section filled) from *change request originator* role and provides back necessary documents it produces out of its activities. There is alike individual role-process diagrams for each role that actively participates in the change management process. These participating roles are determined in a role diagram in context definition phase of the Plural method (section 3.3)



Interacts
Carries out
Interacts
Roles
Evaluate Change Request
Change Manager
Activities
A change request for a product is received
Change Request Originator
CRF (1.section filled)
Review CR
CR: Change Request
CRF: Change Request Form
XOR
CR rejected
CR needs further investigation
CR does NOT need any further investigation
Initiate an investigation for the CR
CRF (1.section filled)
Investigation Request for CR
Project Team
CR Investigation Result
Review investigation results
XOR
Team decision is to cancel CR
Team decision is to implement CR
XOR
XOR
Close Change Request
CRF (1.section filled)
Fill in the 2nd. Section of the CRF and send to Author
CRF (2.section filled)
Author
CR sent to related Author to perform the change

Figure 36. An individual role-process diagram (role: change manager operation: evaluate change request)

b. Activity level process diagram

Integrated process diagrams can be in activity, operation or process abstractions levels. Activity level process diagrams integrate all related individual role-process diagrams and represent the process in the lowest detail. Figure 37 depicts an activity level integrated process diagram for the change management process.

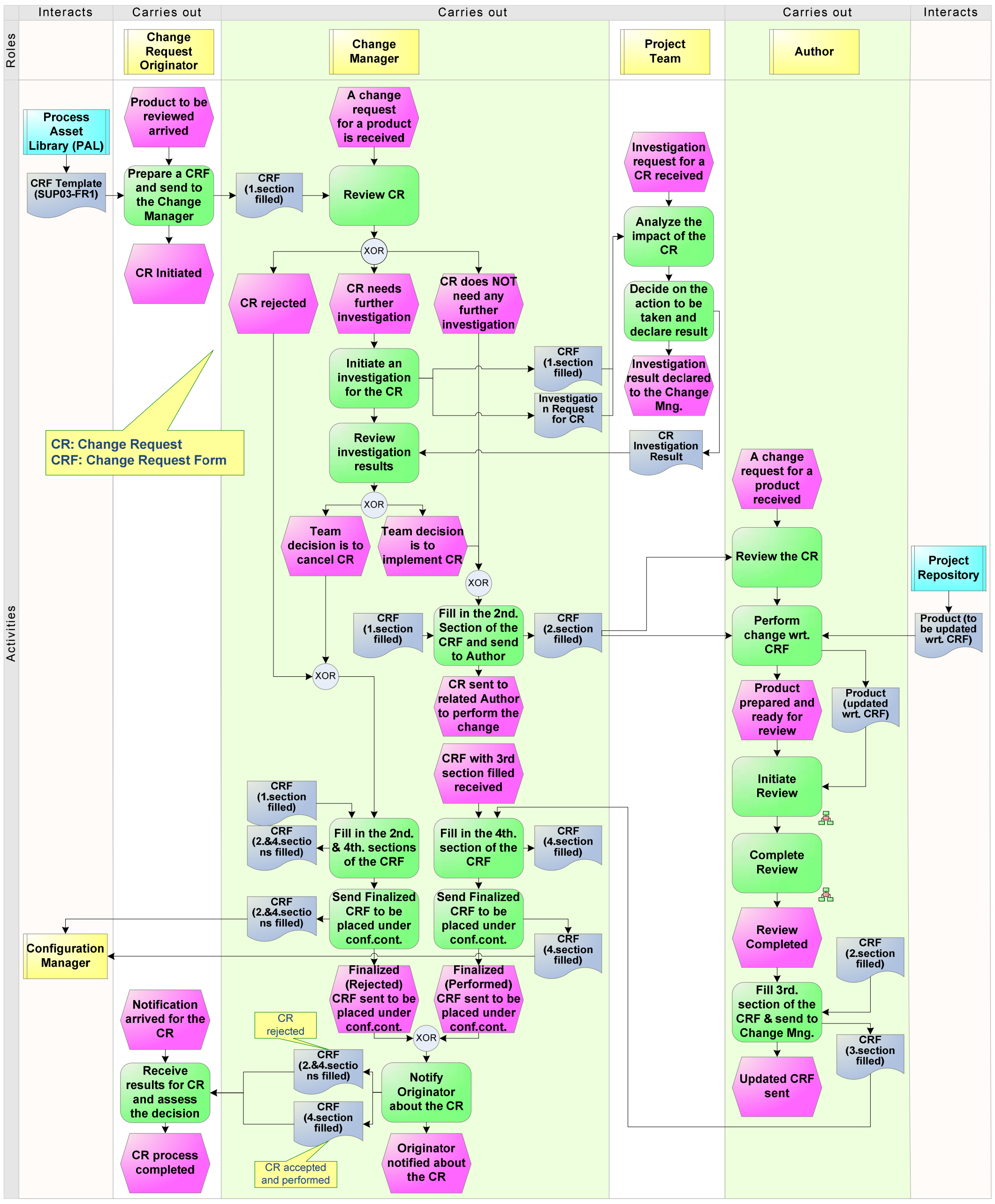

Interacts
Carries out
Carries out
Carries out
Interacts
Roles
Activities
Change Request Originator
Change Manager
Project Team
Author
Process Asset Library (PAL)
CRF Template (SUP03-FR1)
Product to be reviewed arrived
Prepare a CRF and send to the Change Manager
CR Initiated
CRF (1.section filled)
A change request for a product is received
Review CR
XOR
CR rejected
CR needs further investigation
CR does NOT need any further investigation
Investigation request for a CR received
Analyze the impact of the CR
Decide on the action to be taken and declare result
Investigation result declared to the Change Mng.
Initiate an investigation for the CR
CRF (1.section filled)
Investigation Request for CR
CR Investigation Result
Review investigation results
CR: Change Request
CRF: Change Request Form
XOR
Team decision is to cancel CR
Team decision is to implement CR
XOR
A change request for a product received
Review the CR
Project Repository
Product (to be updated wrt. CRF)
CRF (1.section filled)
Fill in the 2nd. Section of the CRF and send to Author
CRF (2.section filled)
Perform change wrt. CRF
XOR
CR sent to related Author to perform the change
Product prepared and ready for review
Product (updated wrt. CRF)
CRF with 3rd section filled received
Initiate Review
CRF (1.section filled)
CRF (2.&4.sections filled)
Fill in the 2nd. & 4th. sections of the CRF
Fill in the 4th. section of the CRF
CRF (4.section filled)
Complete Review
CRF (2.&4.sections filled)
Send Finalized CRF to be placed under conf.cont.
Send Finalized CRF to be placed under conf.cont.
CRF (4.section filled)
Review Completed
Configuration Manager
CRF (2.section filled)
Finalized (Rejected) CRF sent to be placed under conf.cont.
Finalized (Performed) CRF sent to be placed under conf.cont.
Fill 3rd. section of the CRF & send to Change Mng.
Notification arrived for the CR
CR rejected
XOR
CRF (3.section filled)
CRF (2.&4.sections filled)
Notify Originator about the CR
Updated CRF sent
Receive results for CR and assess the decision
CRF (4.section filled)
CR process completed
CR accepted and performed
Originator notified about the CR

Figure 37. An activity level process diagram (process: manage change)

c. Operation level process diagram

Operation level process diagrams depict the roles' operations and their data dependency between them in a specific process. Figure 38 presents an example of the diagram for the change management process.

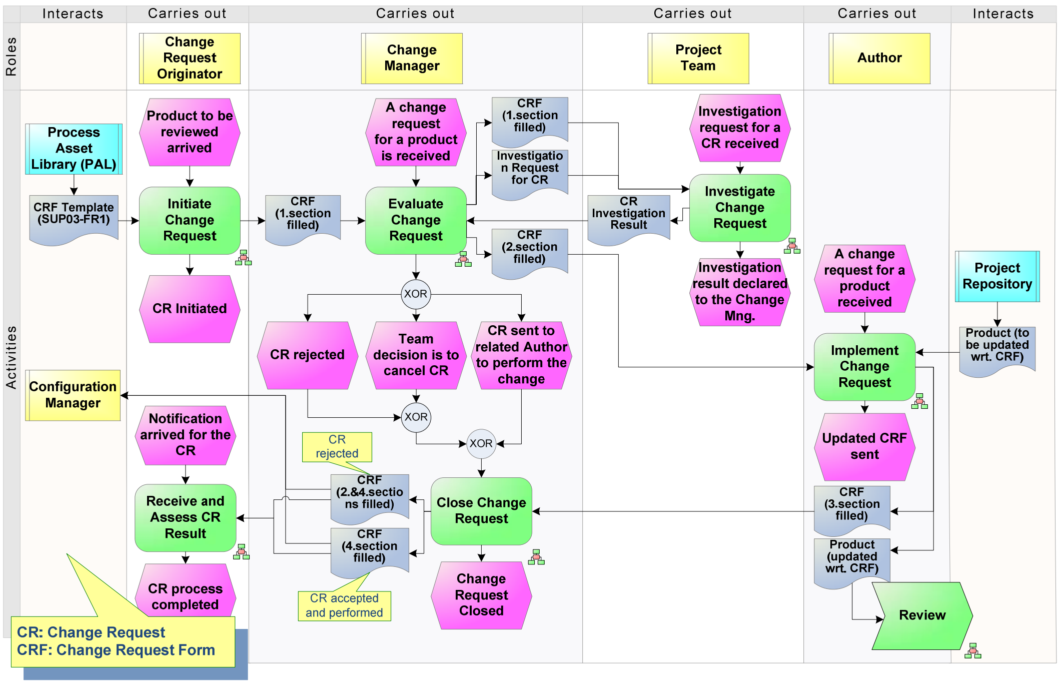

Interacts
Carries out
Carries out
Carries out
Carries out
Interacts
Roles
Activities
Change Request Originator
Change Manager
Project Team
Author
Process Asset Library (PAL)
CRF Template (SUP03-FR1)
Product to be reviewed arrived
Initiate Change Request
CR Initiated
CRF (1.section filled)
A change request for a product is received
Evaluate Change Request
CRF (1.section filled)
Investigation Request for CR
CRF (2.section filled)
XOR
CR rejected
Team decision is to cancel CR
CR sent to related Author to perform the change
XOR
XOR
CR rejected
CRF (2.&4.sections filled)
CRF (4.section filled)
CR accepted and performed
Close Change Request
Change Request Closed
Investigation request for a CR received
Investigate Change Request
CR Investigation Result
Investigation result declared to the Change Mng.
A change request for a product received
Implement Change Request
Updated CRF sent
CRF (3.section filled)
Product (updated wrt. CRF)
Review
Project Repository
Product (to be updated wrt. CRF)
Configuration Manager
Notification arrived for the CR
Receive and Assess CR Result
CR process completed
CR: Change Request
CRF: Change Request Form

Figure 38. An operation level process diagram (process: manage change)

d. Process level process diagram

Process Level process diagram depicts the process and its data relationship between other roles or other entities (application systems, information stores). It also displays the roles that participate in the process. Figure 39 gives an example of a process level process diagram for the change management process.

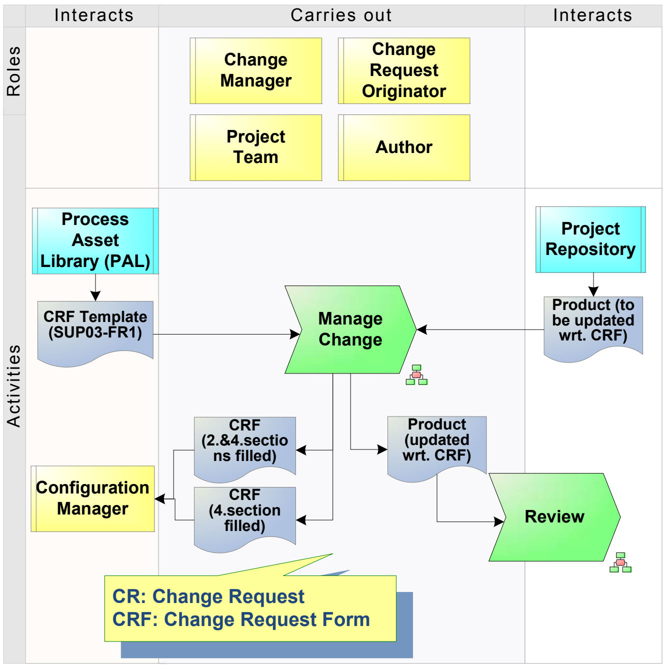


Figure 39. A process level process diagram (process: manage change)

Section 3.6.10 describes how process, operation and activity level process diagrams are generated based on individual definitions.

3.6.5. Scope Diagram

Scope diagram defines the span of the entire work that the Plural method will be applied to. In the context definition phase, the scope of the project is determined and accordingly a list of processes to be covered is determined (section 3.3.3). This list of processes, their relationships and the roles that participate in these processes are depicted in a scope diagram. For each process, the diagram optionally represents the process goals, measures and a brief description. Semantic of a process in the scope diagram is similar to the concept of use cases in UML [113]. They can be included, extended and generalized. Figure 40 gives the elements and their relationships in a scope diagram.

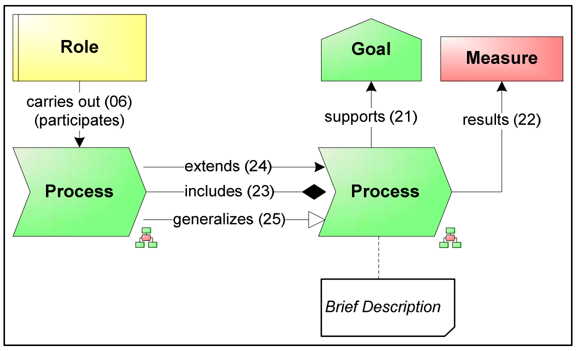


Figure 40. Elements and relationships in a scope diagram

The connection between roles and processes are 'participates' relationships and implies that the role is participating in the process by performing one or more activities to serve the process goal (Table 7 relationship 06). Relationships among processes can be of 'includes', 'extends' and 'generalizes' type (Table 7 relationships 23, 24 and 25 respectively). When a process includes the other, it brings that reusable process into its execution (the arrow head is in diamond-shaped and points to the process that includes the other). The inclusion in extends relationship is conditional which depends on the runtime execution of the extended process (arrow head points to the process that is extended). Generalization is between a generalized process and a specific process where the specific process is similar to the generalized one but it has some differences that should be noted (hollow arrow head points to the generalized process). The relationships of the process between goal and measure elements are described in Table 7 relationship 21 and 22 respectively. Figure 41 gives an example of a scope diagram.

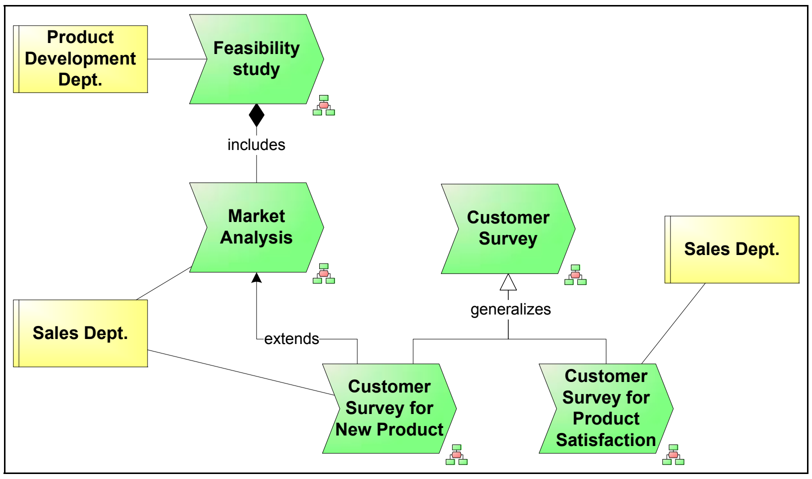

Product Development Dept.
Feasibility study
includes
Market Analysis
Customer Survey
Sales Dept.
generalizes
Sales Dept.
extends
Customer Survey for New Product
Customer Survey for Product Satisfaction

Figure 41. A scope diagram

3.6.6. Role Diagram

There are inherent static relationships between the roles participating in the processes and these relationships are depicted in a role diagram. Similar to the class concept in UML [113], relationships are in association, aggregation (composition) and generalization type. Role diagram is similar to the class diagram in UML in terms of its semantics in which roles are considered as classes. Figure 42 displays the relationships between roles in a role diagram.

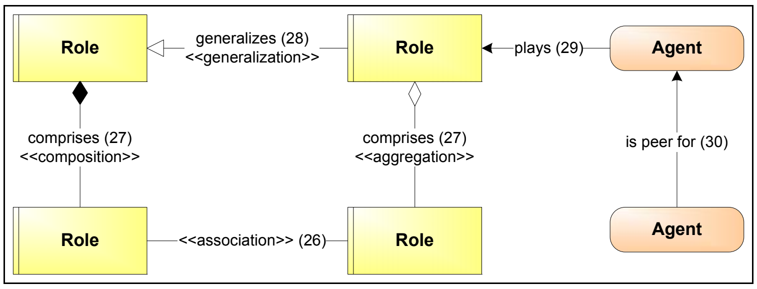


Figure 42. Elements and relationships in a role diagram

Relationships between elements in a role diagram are described in Table 7 as relationships 26 through 30. The arrow head in comprises relationship is in diamond-shaped and points to the role that comprises the other. The hollow arrow head in generalizes points to the generalized role. The mapping between agents and roles is related with organizational settings and its representation is optional. Figure 43 presents a portion of an example role organization diagram.

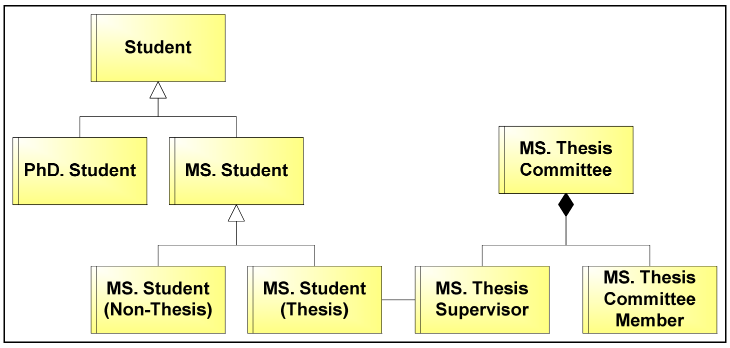


Figure 43. A role diagram

Extended Role Diagram: Role diagrams may also represent the operations of each role together with the process the operation is belonging to. Figure 44 gives an example of an extended role diagram.

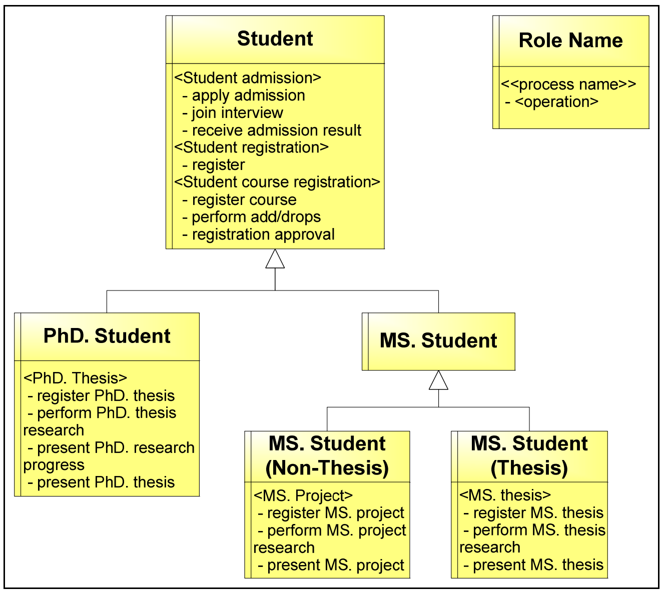


Figure 44. An extended role diagram

3.6.7. Information Item Diagram

Information item diagram presents the static relationships between information items. Similar to the role diagrams, the relationships in information item diagrams are in association, aggregation (composition) and generalization type between information items. Figure 45 displays the elements and their relationships. These relationships are described in Table 7 as relationships 31 through 33. The arrow head in comprises relationship is in diamond-shaped and points to the information item that comprises the other. The hollow arrow head in generalizes points to the generalized information item.

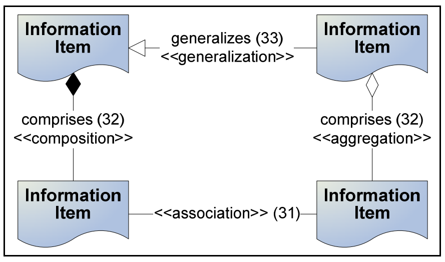


Figure 45. Elements and relationships in an information item diagram

Figure 46 gives three examples of information item diagrams.

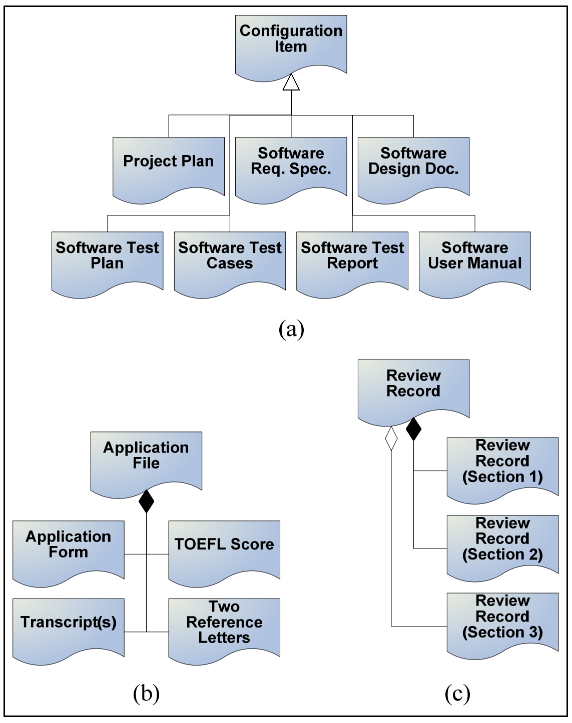


Figure 46. Information item diagrams

3.6.8. Role Dependency Diagram

Role dependency diagrams depict the roles and their message exchange between each other and other entities like application systems and information stores in terms of the information items. Figure 47 displays the elements and their relationships in a role dependency diagram.

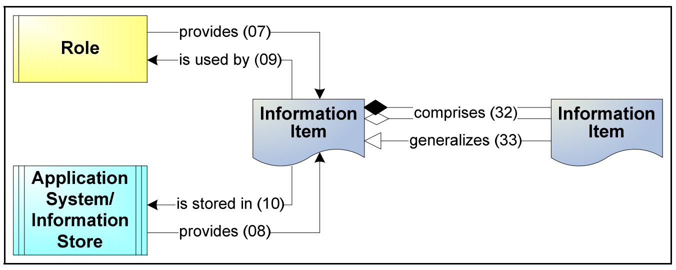


Figure 47. Elements and relationships in a role dependency diagram

‘Provides’ or ‘is used by’ relationships do not imply any ownership between roles and information items. In a lifecycle of a work product, it might be produced or updated by multiple roles. Therefore, with specific states, it is used or provided by many roles one of which only owns it. In Plural, ownership of work products is not explicitly represented.

Role dependency diagrams can be filtered for a set of roles or processes, that is, it may include the roles in a specific set of processes or may represent the dependency between a numbers of roles regardless of the processes they participate. Conceptually, the aggregation and generalization relationships between roles as well as information items can be considered in a role dependency diagram. For example, the dependency of the review team to a specific role was nothing but the union of the dependency of the roles that constituted that team. Figure 48 gives an example of a role dependency diagram representing the exchange of information items in a review process. Figure 49 gives an example of a dependency between two roles where generalization relationship between information items is utilized.

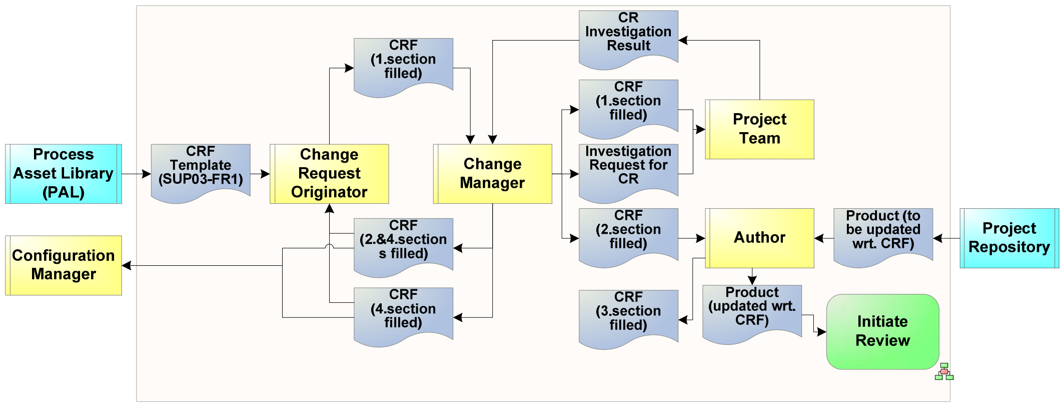


Figure 48. A role dependency diagram (process: manage change)

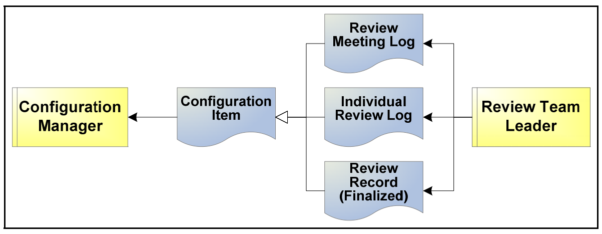


Figure 49. A role dependency diagram (configuration manager and review team leader)

Section 3.6.10.d describes how the role dependency diagrams can be generated based on individual role-process diagrams, role diagrams and information item diagrams.

3.6.9. Process Dependency Diagram

The process dependency diagram presents the information item exchange between processes as well as the interface to external roles, information stores and application systems. Figure 50 displays the elements and their relationships in a process dependency diagram.

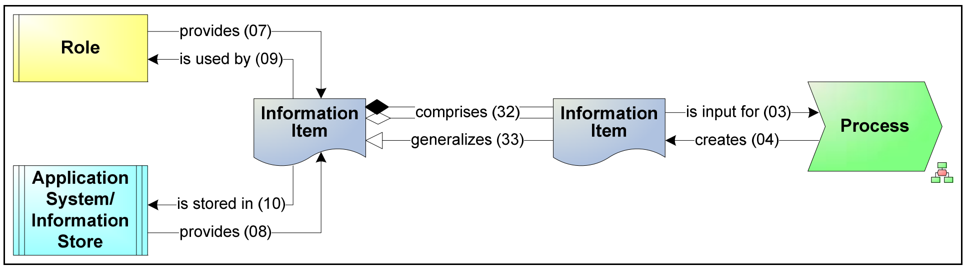
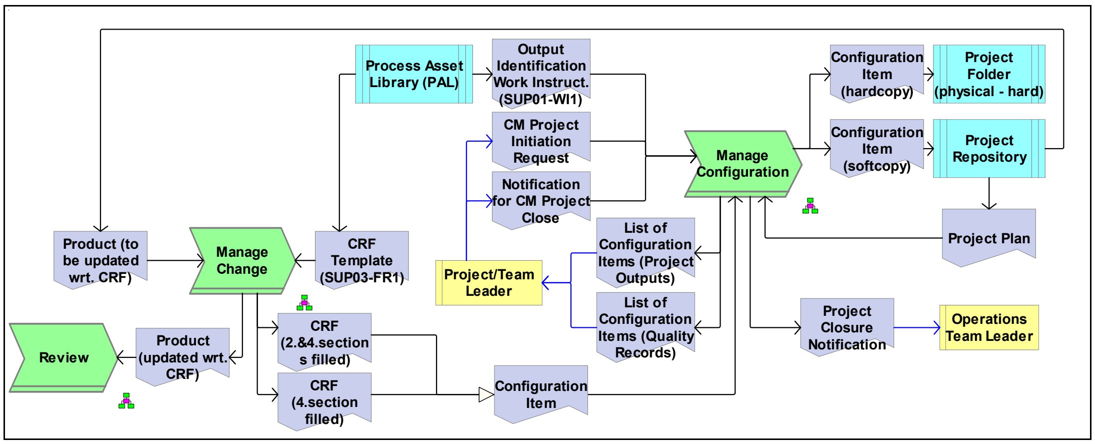


Figure 50. Elements and relationships in a process dependency diagram

Process diagrams can be filtered for a specific set of processes. Similar to the case in role dependency diagrams, the aggregation and generalization relationships between information items can be taken into account in the generation of process dependency diagrams. Figure 51 gives an example of a process dependency diagram between change management and configuration management which also presents how the 'configuration item' has been generalized.



Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request
Notification for CM Project Close
Configuration Item (hardcopy)
Project Folder (physical - hard)
Configuration Item (softcopy)
Project Repository
Manage Configuration
Project Plan
Product (to be updated wrt. CRF)
Manage Change
CRF Template (SUP03-FR1)
List of Configuration Items (Project Outputs)
Project/Team Leader
List of Configuration Items (Quality Records)
Project Closure Notification
Operations Team Leader
Review
Product (updated wrt. CRF)
CRF (2.&4.sections filled)
CRF (4.section filled)
Configuration Item

Figure 51. A process dependency diagram

Process dependency diagrams can be generated from other diagrams present in the process base. Section 3.6.10.e describes the way this generation can be performed.

3.6.10. Generating Diagrams

In Plural some of the diagrams can be generated from individual role-process diagrams, role diagrams and information item diagrams. As mentioned in 3.4.2, existence of individual role-process diagrams and role diagrams is mandatory in Plural, but the information item diagrams are optional since it depends on whether agents feel necessary to use and represent any generalization and composition relationship between items. The following diagrams can be generated in Plural:

1. Activity level process diagrams
2. Operation level process diagrams
3. Process level process diagrams
4. Role dependency diagrams
5. Process dependency diagrams

Subsequent sections describe how each of the above diagrams can be generated.

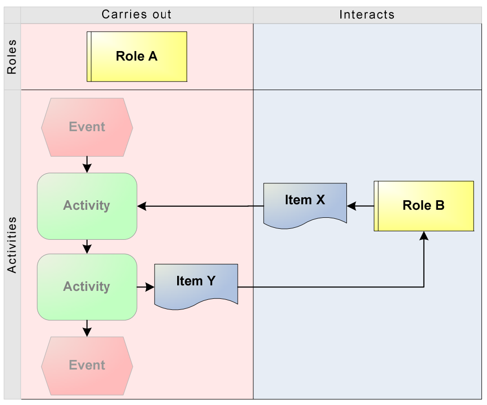
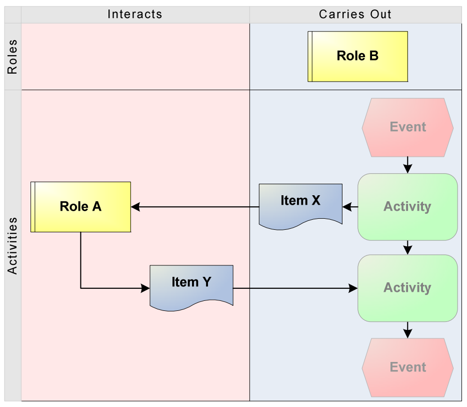
a. Generating Activity Level Process Diagrams

We describe the notation for process diagrams in section 3.6.4 and give an example in Figure 37 (pg.95). Activity level process diagrams are integrated individual role-process diagrams representing a process in the lowest level of functional abstraction - the activity. For an activity level process diagram, each individual role-process diagram that belongs to a process is merged into each other side by side via the information items the roles sent to each other and the activities they perform together. Figure 37 (on page 95) depicted diagram for the manage change process.

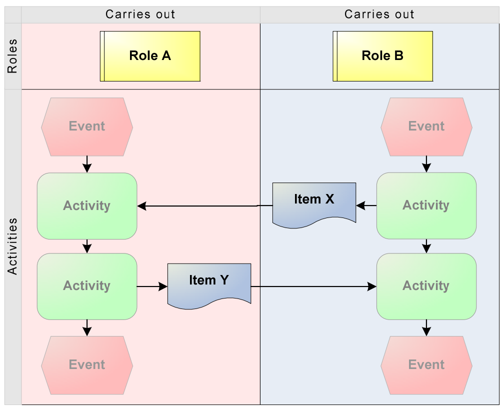
The decision of which individual role-process diagrams to include in the integration process is based on agents' declaration of the operations and the primary processes they belong to. The declaration is provided in a process diagram such as the one given in Figure 30 (pg.68). In Figure 30, for example, depicts the operations of the review team leader in the review process.

Table 9, Table 10 and Table 11 present three cases of consistent individual role-process diagrams and the way they can be integrated to form activity level process diagrams. Table 9 presents the simplest case for the generation.

Table 9. Integration of individual role-process diagrams (Case1)

Individual role-process diagram **Operation O in Process P** **Expectations:** 1 - Role A ← Item X ← Role B 2 - Role A → Item Y → Role B  Carries out Interacts Roles Activities Role A Event Activity Item X Role B Activity Item Y Event	**Individual role-process diagram** **Operation K in Process P** **Fulfilled by (Case 1)** 1 - Role B → Item X → Role A 2 - Role B ← Item Y ← Role A  Interacts Carries Out Roles Activities Role B Event Role A Item X Activity Item Y Activity Event

Integrated Activity level process diagram for Process P (Case1)



Since there are generalization relationships between roles, a role can inherit the operations of a more general role. These relationships, actually, occur when a process in which the specific role participates includes another reusable one in which a more general role involves in its activities. For example, in a software organization, the project management process may include the review process where the project manager role performing ‘initiate review’ operation inherits this operation from the author role of the review process. The operation belongs to the author role and the review process and it is performed by the project manager role in the project management process. This inheritance and include relationships may also be present for other specific role and process couples such as; providing training process and trainer role, development process and developer role. In integrations, the details of such operations are presented in the integration of the reusable process and left as operations in the integrated process which includes it. That is, for the above case, in the project management process, the activity level process diagram will not display the details of the ‘initiate review’ operation which is elaborated in the activity level process diagram of the review process. Table 10 gives an example of such a case and the way the integration should occur.

Table 10. Integration of individual role-process diagrams (Case2)

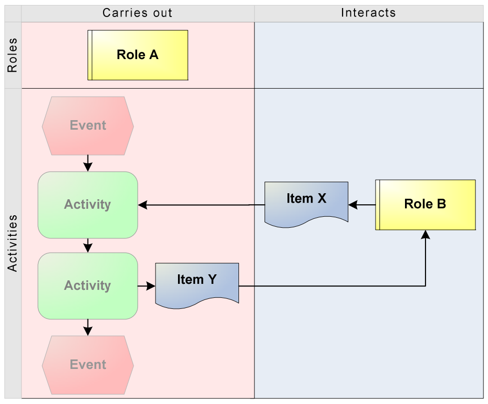
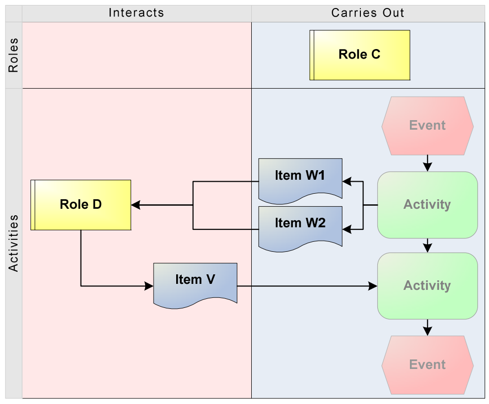
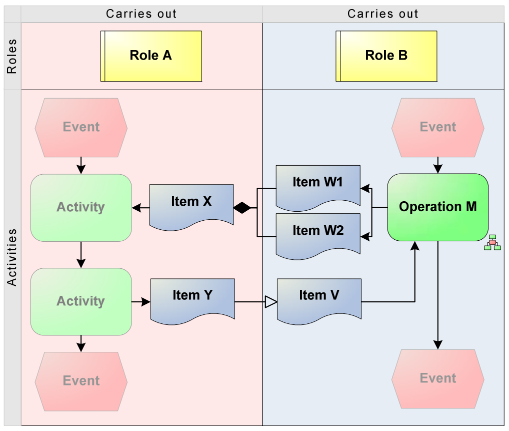
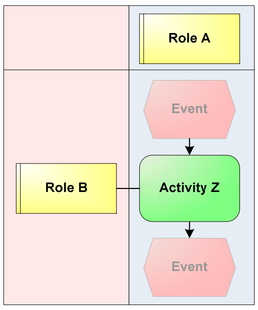
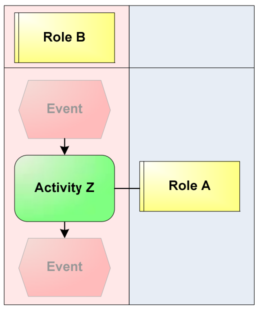
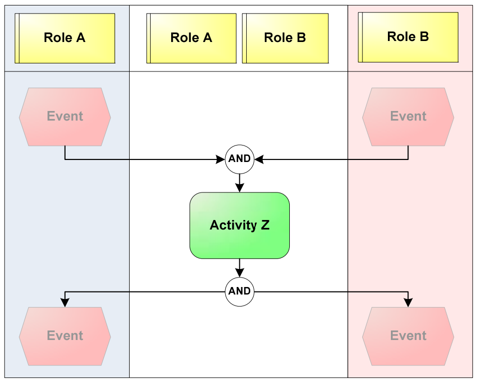
Individual role-process diagram **Operation O in Process P** **Expectations:** 1 - Role A 🡠 Item X 🡠 Role B 2 - Role A 🡢 Item Y 🡢 Role B  Carries out Interacts Roles Activities Role A Event Activity Item X Role B Activity Item Y Event	**Individual role-process diagram** **Operation M in Process S** **Fulfilled by (Case 2)** 1 - Role C 🡢 Item W1 🡢 Role D 2 - Role C 🡢 Item W2 🡢 Role D 3 - Role C 🡠 Item V 🡠 Role D WHERE Role B generalizes Role C AND Role A composes Role D AND Item X is composed of Item W1 AND Item X is composed of Item W2 AND Item Y generalizes Item V  Interacts Carries Out Roles Activities Role C Event Item W1 Role D Activity Item W2 Item V Activity Event
Integrated Activity level process diagram for Process P (Case2)  Carries out Carries out Roles Activities Role A Role B Event Event Activity Item X Item W1 Item W2 Operation M Activity Item Y Item V Event Event	

Table 11 gives the case when the expectation involves the activities that roles perform together.

Table 11. Integration of individual role-process diagrams (Case3)

Individual role-process diagram **Operation Q in Process P** **Expectations:** 1 - Role A → Activity Z ← Role B 	**Individual role-process diagram** **Operation L in Process P** **Fulfilled by (Case 3)** 1 - Role B → Activity Z → Role A 
Integrated Activity level process diagram for Process P (Case3)  Role A Role A Role B Role B Event Event AND Activity Z AND Event Event	

b. Generating Operation Level Process Diagrams

Operation level process diagrams carry activity level diagrams to a higher abstraction level where roles' operations and their data dependency between them in a specific process are presented. The notation and examples for these diagrams are given in section 3.6.4 . Each operation has a primary process and a role that own that operation. For example, in the review process, the author role has initiate review and complete review operations. As noted above, these declarations are provided by a role in a process diagram such as the one given in Figure 30 (pg.68). Each operation has an individual role-process diagram depicting the activities that role performs within. The Plural pursues a common rule about the conservation of inputs and outputs between activities, operations and processes which represent the functions that are performed in different abstraction levels. For the operations, all inputs to the role's context defined in an individual role-process diagram originating from another role, application system, information store, or role's other operation are considered inputs to the operation. Similarly, outputs from role's context delivered to other roles, operations or stored in an application system or information store are also considered outputs from the operation. Thus, the interface of an operation can be represented as the input and output information to and from its environment. For example, the operation depicted in Figure 36 (on page 94) can be represented in an operation level process diagram given in Figure 52.

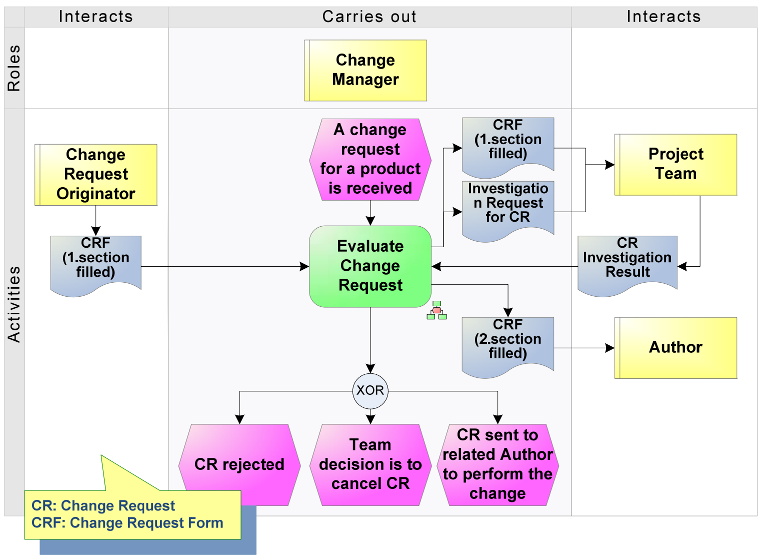
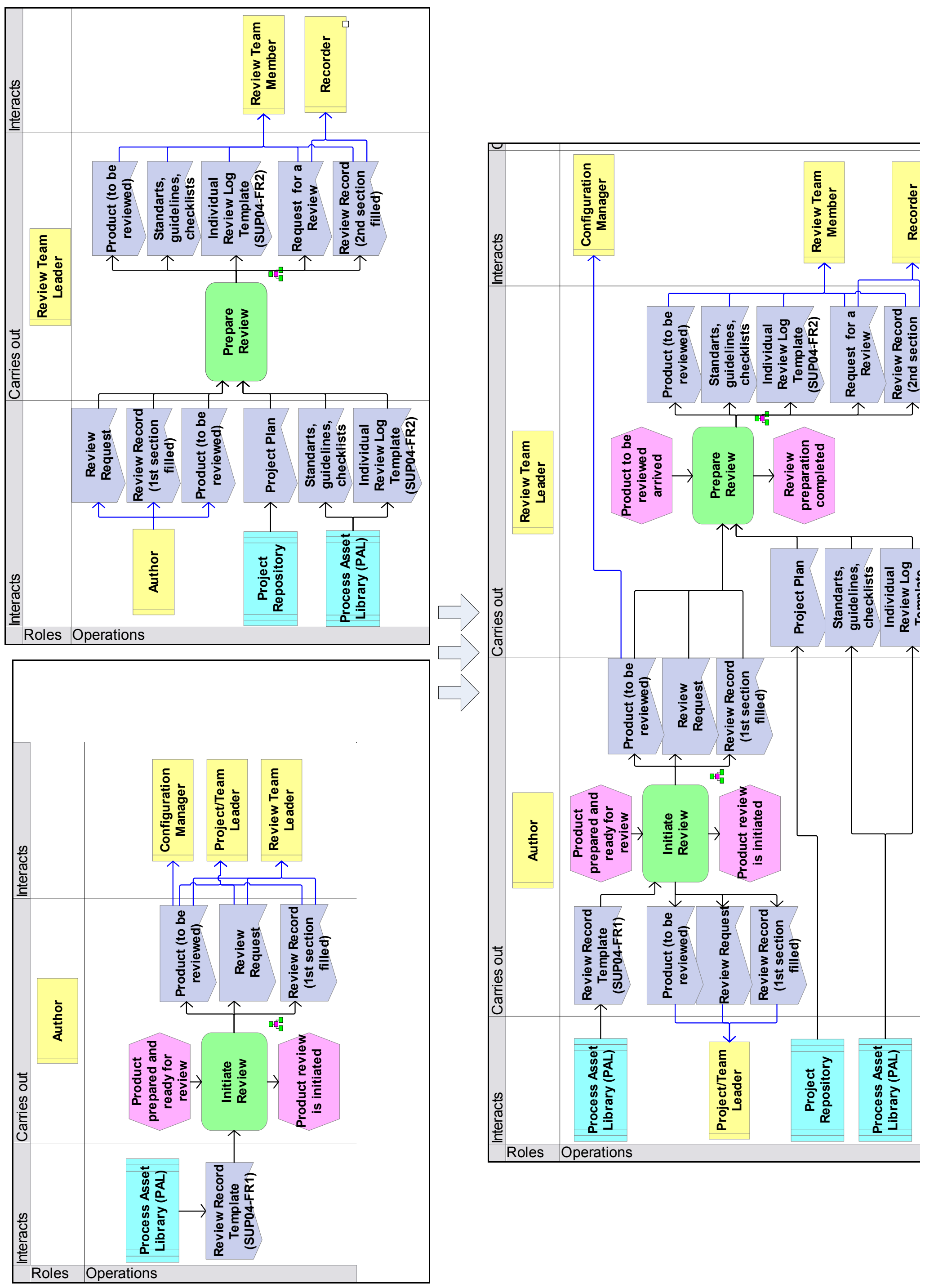


Figure 52. An operational level process diagram for the evaluate change request operation

Similar to the integration process performed in the generation of activity level process diagrams, in operation level diagrams the operations belonging to that primary process are put side by side for each role. If an input to an operation is originated from a role that has an operation in that process and produces it as an output, then that item is represented as an item delivered from the producer operation to the receiving operation. Figure 53 depicts this situation for two operations in the review process; the initiate review operation of the author and prepare review operation of the review team leader.



Roles
Operations
Interacts
Carries out
Interacts
Process Asset Library (PAL)
Review Record Template (SUP04-FR1)
Product prepared and ready for review
Initiate Review
Product review is initiated
Author
Product (to be reviewed)
Review Request
Review Record (1st section filled)
Configuration Manager
Project/Team Leader
Review Team Leader
Project Repository
Project Plan
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Prepare Review
Request for a Review
Review Record (2nd section filled)
Review Team Member
Recorder
Product to be reviewed arrived
Review preparation completed

Figure 53. An integration of operations

Figure 54 presents a complete example of the generated diagram for the review process.

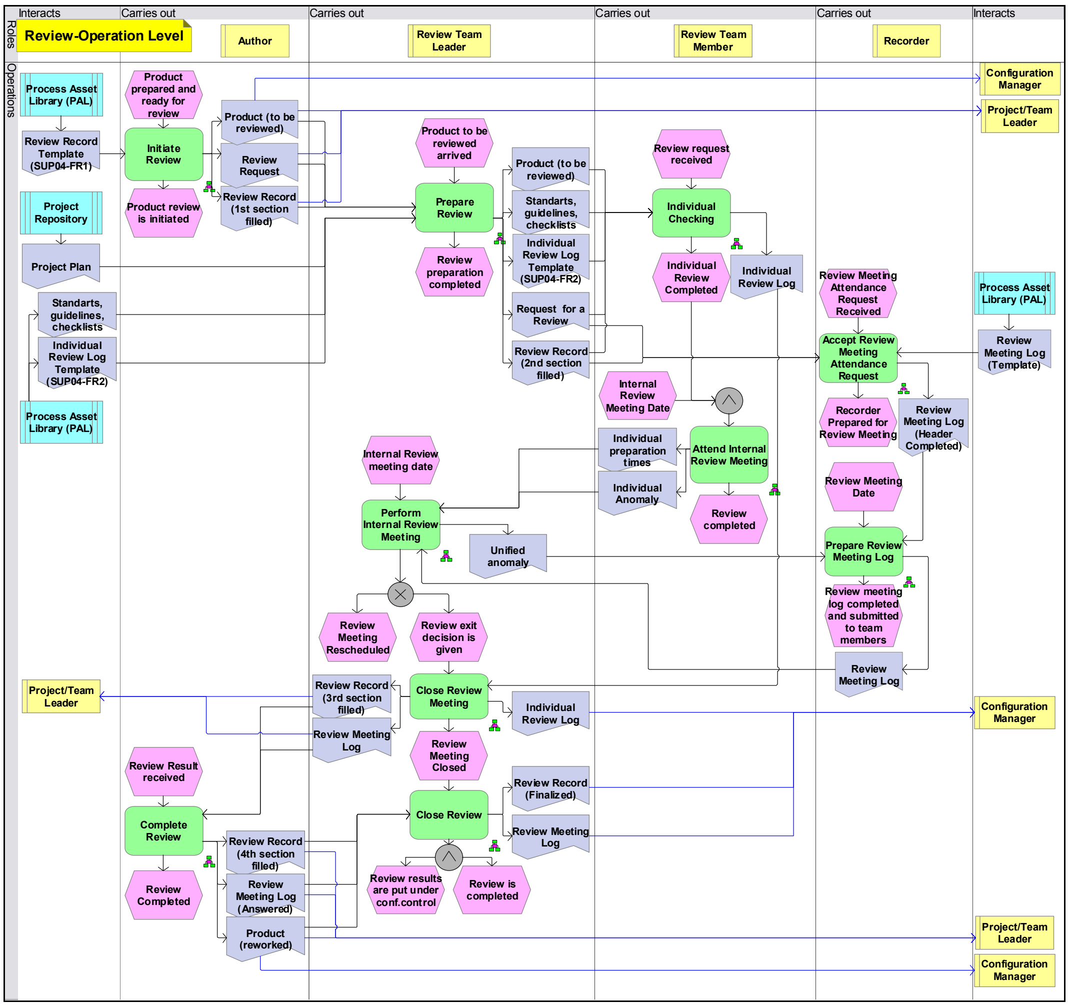
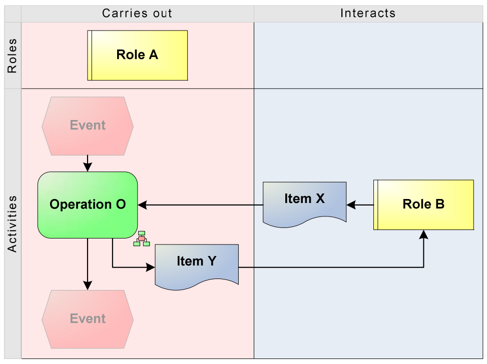
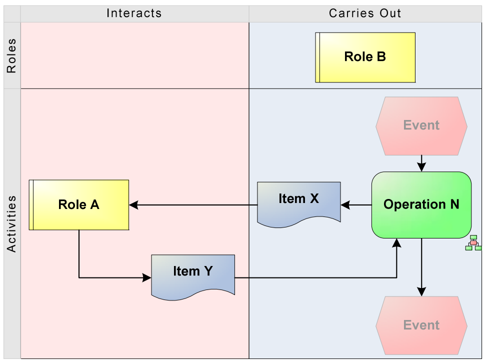
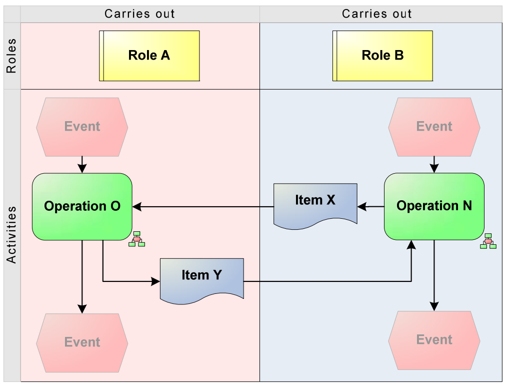

Interacts
Carries out
Carries out
Carries out
Carries out
Interacts
Roles
Operations
Review-Operation Level
Author
Review Team Leader
Review Team Member
Recorder
Process Asset Library (PAL)
Review Record Template (SUP04-FR1)
Project Repository
Project Plan
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Process Asset Library (PAL)
Product prepared and ready for review
Initiate Review
Product review is initiated
Product (to be reviewed)
Review Request
Review Record (1st section filled)
Product to be reviewed arrived
Prepare Review
Review preparation completed
Product (to be reviewed)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Request for a Review
Review Record (2nd section filled)
Review request received
Individual Checking
Individual Review Completed
Individual Review Log
Internal Review Meeting Date
Individual preparation times
Individual Anomaly
Attend Internal Review Meeting
Review completed
Internal Review meeting date
Perform Internal Review Meeting
Unified anomaly
Review Meeting Rescheduled
Review exit decision is given
Review Meeting Attendance Request Received
Accept Review Meeting Attendance Request
Recorder Prepared for Review Meeting
Review Meeting Log (Header Completed)
Review Meeting Date
Prepare Review Meeting Log
Review meeting log completed and submitted to team members
Review Meeting Log
Configuration Manager
Project/Team Leader
Process Asset Library (PAL)
Review Meeting Log (Template)
Project/Team Leader
Review Record (3rd section filled)
Review Meeting Log
Close Review Meeting
Individual Review Log
Review Meeting Closed
Review Result received
Complete Review
Review Completed
Review Record (4th section filled)
Review Meeting Log (Answered)
Product (reworked)
Close Review
Review Record (Finalized)
Review Meeting Log
Review results are put under conf.control
Review is completed
Configuration Manager
Project/Team Leader
Configuration Manager

Figure 54. An operation level process diagram for the review process

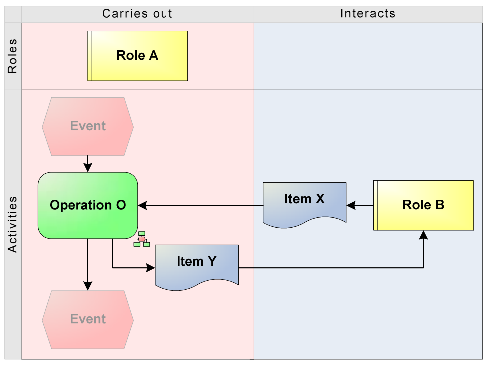
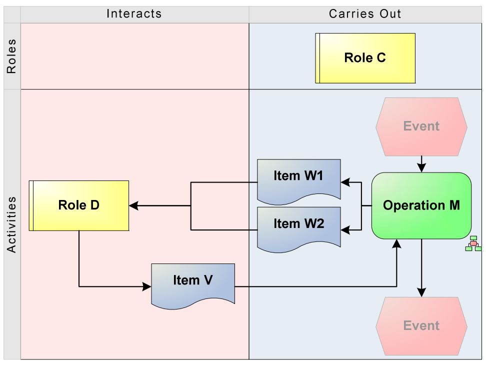
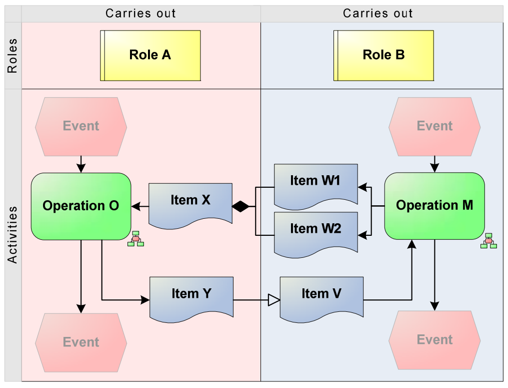
Generalization and composition relationships between information items are also represented in operation level process diagrams similar to the way they are handled in activity level process diagrams. Table 12 presents a case for the integration of two operation level process diagrams.

Table 12. Integration of operations (Case 1)

Operation level-process diagram Operation O in Process P	**Operation level-process diagram Operation N in Process S**
 Carries out Interacts Roles Activities Role A Event Operation O Item X Role B Item Y Event	 Interacts Carries Out Roles Activities Role B Event Role A Item X Operation N Item Y Event
Integrated Operation level process diagram for Process P	
 Carries out Carries out Roles Activities Role A Role B Event Event Operation O Item X Operation N Item Y Event Event	

Based on the case depicted on Table 10 (pg.105), Table 13 presents the same case with diagrams that are in higher abstraction levels.

Table 13. Integration of operations (Case 2)

Operation level-process diagram **Operation O in Process P** 	**Operation level-process diagram** **Operation M in Process S** WHERE Role B generalizes Role C AND Role A composes Role D AND Item X is composed of Item W1 AND Item X is composed of Item W2 AND Item Y generalizes Item V 
Integrated Operation level process diagram for Process P 	

c. Generating Process Level Process Diagrams

We describe the notation and the syntax in process level process diagrams in section 3.6.4. This section describes how process level process diagrams can be generated from their operation level diagrams.

Process level diagrams are synthesis of their operation level diagrams where all operations are unified into the process they belong to and the interface becomes the data transfer between the process and the outer context consisting applications systems, information stores and roles that are not participating in the process but interacting via messages they receive and provide to the roles that perform operations within the process. Thus, the interface of process level diagrams comprises the union of the information items the participator roles send to or receive from other entities in the environment. Figure 55 presents the generated process level diagram for the review process from its operations level diagram given in Figure 54.

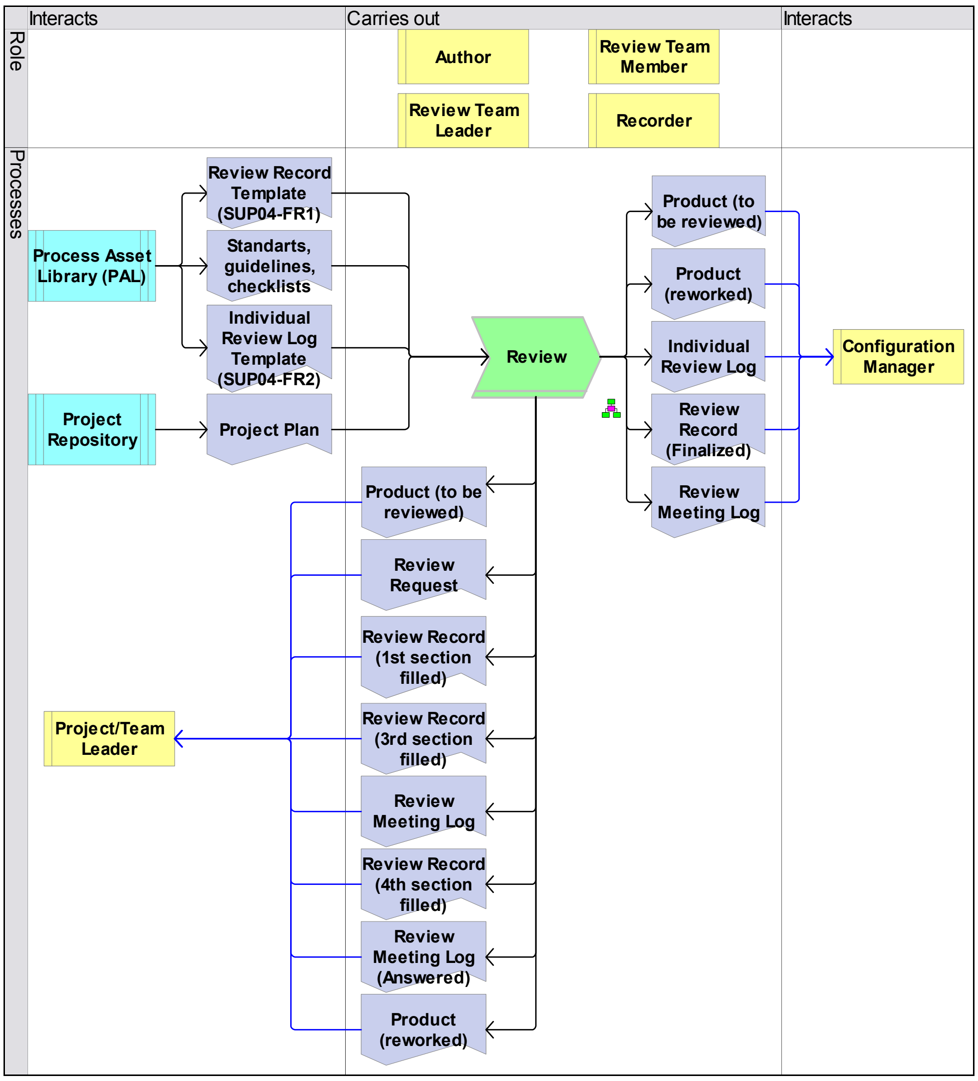


Figure 55. A process level process diagram (review process)

d. Generating Role-Dependency Diagrams

Role dependency diagrams present the roles and what information items are is exchanged between each other, application systems and information stores. Its syntax and examples are given in section 3.6.8. Role dependencies can be generated from individual role-process diagrams. Table 14 and Table 15 present two cases for these types of generations.

Table 14. Generation of role dependencies (Case1)

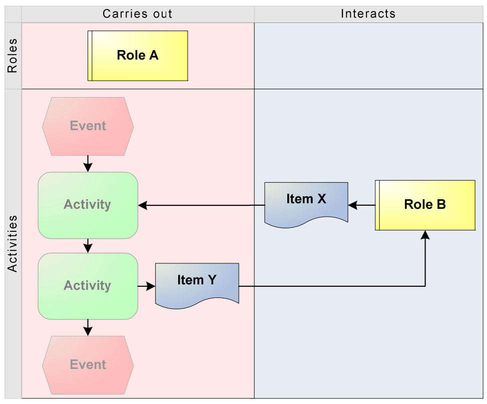
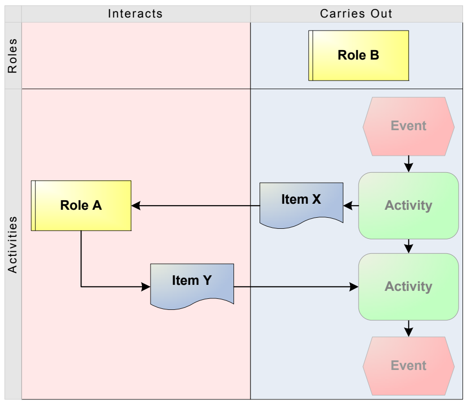
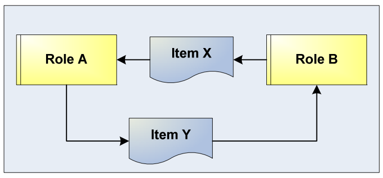
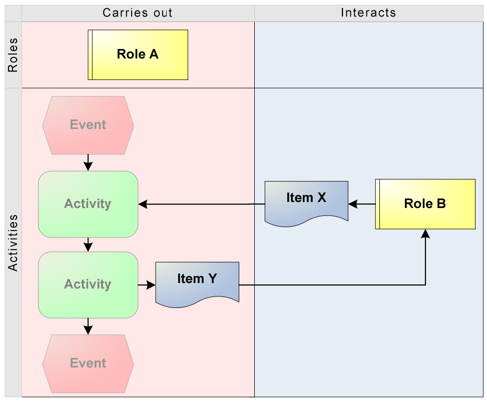
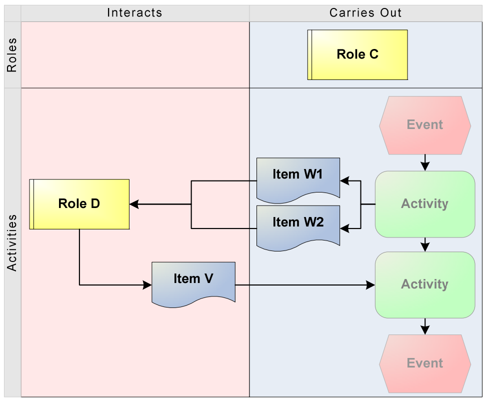
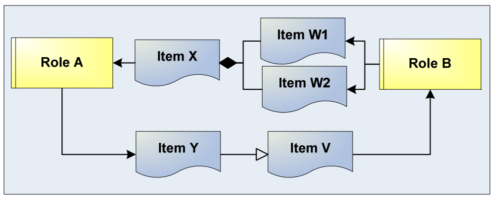
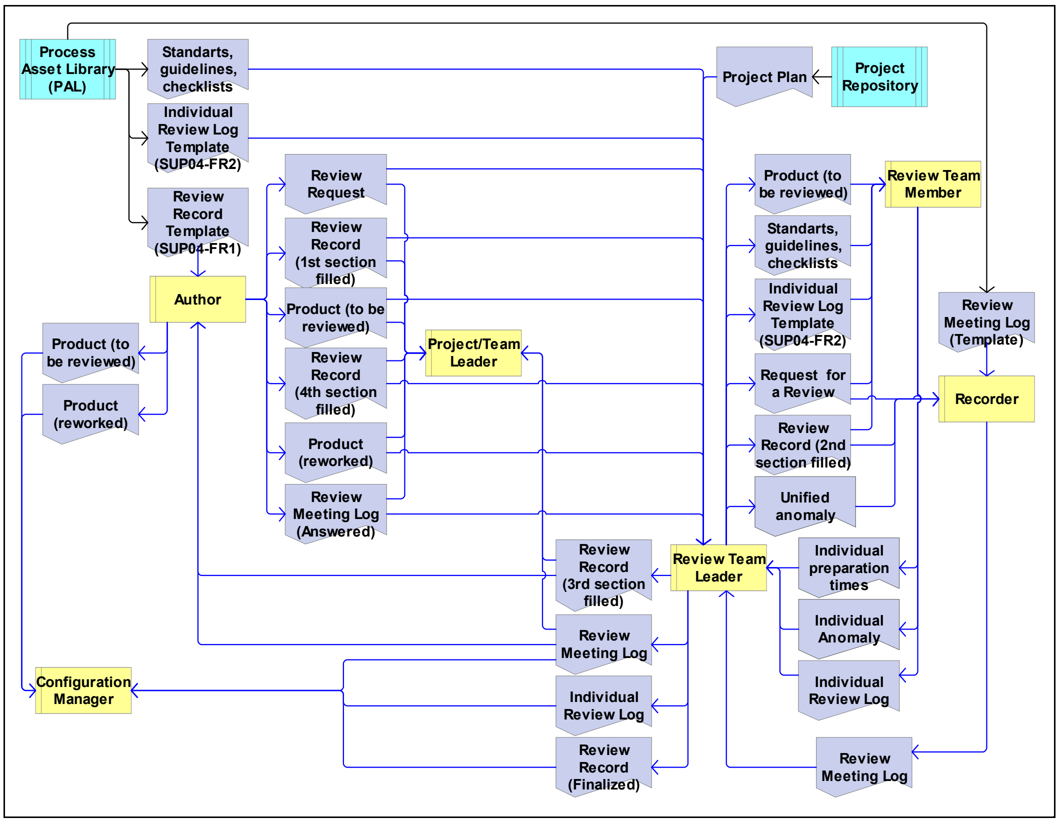
Individual role-process diagram **Operation O in Process P**	**Individual role-process diagram** **Operation K in Process P**
 Carries out Interacts Roles Activities Role A Event Activity Item X Role B Activity Item Y Event	 Interacts Carries Out Roles Activities Role B Event Role A Item X Activity Item Y Activity Event
Role dependency diagram for Role A & Role B (Case1) 	

Table 15 presents a case where inheritance between roles and information items are considered.

Table 15. Generation of role dependencies (Case2)

Individual role-process diagram **Operation O in Process P**	**Individual role-process diagram** **Operation M in Process S** WHERE Role B generalizes Role C AND Role A composes Role D AND Item X is composed of Item W1 AND Item X is composed of Item W2 AND Item Y generalizes Item V
 Carries out Interacts Roles Activities Role A Event Activity Item X Role B Activity Item Y Event	 Interacts Carries Out Roles Activities Role C Event Item W1 Role D Item W2 Activity Item V Activity Event
Role dependency diagram for Role A & Role B (Case2)  Item W1 Role A Item X Role B Item W2 Item Y Item V	

With respect to the scope they intent to represent, role dependency diagrams can be filtered for specific roles and processes. Figure 56 depicts a role dependency diagram for the roles that participates or interacts in a review process. Thus the diagram is generated from the individual role-process diagrams of the roles that participates in that process.



Process Asset Library (PAL)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Review Record Template (SUP04-FR1)
Author
Product (to be reviewed)
Product (reworked)
Review Request
Review Record (1st section filled)
Product (to be reviewed)
Review Record (4th section filled)
Product (reworked)
Review Meeting Log (Answered)
Project/Team Leader
Project Plan
Project Repository
Product (to be reviewed)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Request for a Review
Review Record (2nd section filled)
Unified anomaly
Review Team Member
Review Meeting Log (Template)
Recorder
Review Record (3rd section filled)
Review Meeting Log
Individual Review Log
Review Record (Finalized)
Review Team Leader
Individual preparation times
Individual Anomaly
Individual Review Log
Review Meeting Log
Configuration Manager

Figure 56. Role dependencies in a review process

Role dependencies can also utilize the composition relationships between roles. For example, for the above diagram, considering a review team role composed of review team leader, review team member and the recorder roles, the diagram can be re-generated as given in Figure 57.

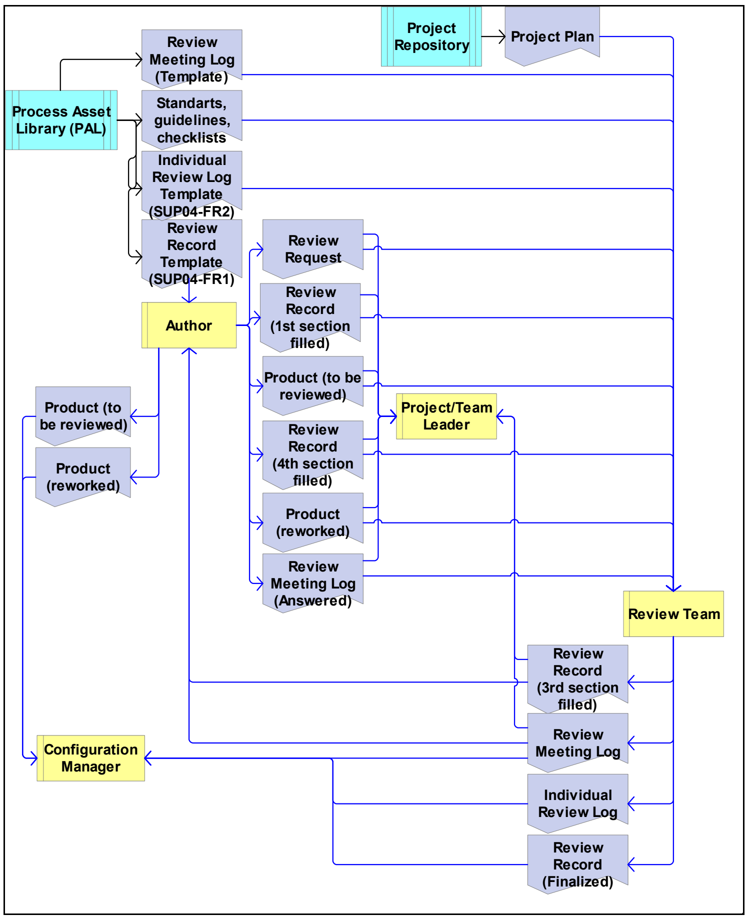
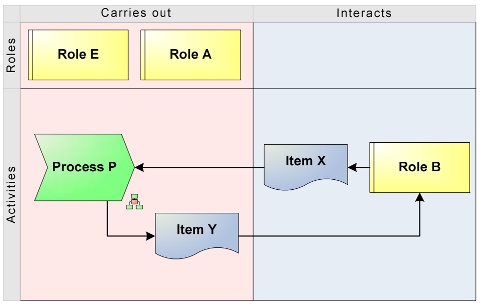
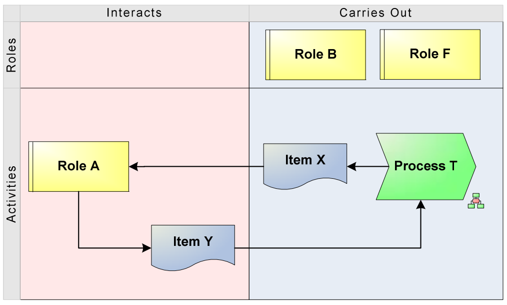
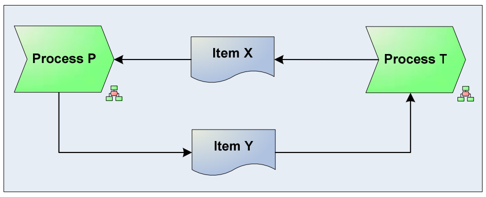

Project Repository
Project Plan
Review Meeting Log (Template)
Process Asset Library (PAL)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Review Record Template (SUP04-FR1)
Review Request
Review Record (1st section filled)
Author
Product (to be reviewed)
Product (to be reviewed)
Project/Team Leader
Product (reworked)
Review Record (4th section filled)
Product (reworked)
Review Meeting Log (Answered)
Review Team
Review Record (3rd section filled)
Review Meeting Log
Configuration Manager
Individual Review Log
Review Record (Finalized)

Figure 57. Role dependencies in a review process (with composite role)

e. Generating Process-Dependency Diagrams

The process dependency diagrams present a set of processes and their interface to each other in terms of the information items they exchange with each other and with external roles, information stores and application systems. The syntax in process dependency diagrams and examples are given in section 3.6.9. Process dependencies can be generated by integrating two or more process level process diagrams and merging them with respect to the information items exchanged between these processes. Generation should also make use of the inheritance and composition relationships between information items -if any-, which are depicted in information item diagrams. Table 16 and Table 17 give examples of two cases for two process level diagrams and the way they can be merged to form process dependency diagrams.

Table 16. Integration of processes (Case 1)

Process level-process diagram **Process P**  Carries out Interacts Roles Activities Role E Role A Process P Item X Role B Item Y	**Process level-process diagram** **Process T**  Interacts Carries Out Roles Activities Role B Role F Role A Item X Process T Item Y
Process dependency diagram for Process P & T 	

The case depicted in Table 17 makes use of the inheritance and composition relationships between roles and information items.

Table 17. Integration of processes (Case 2)

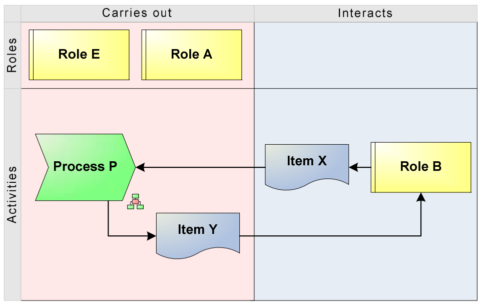
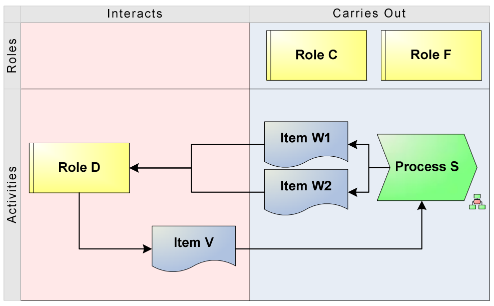
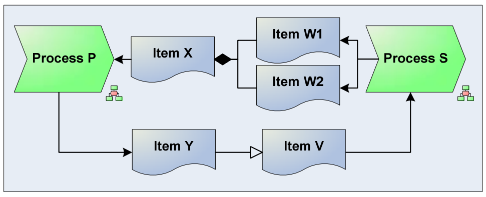
Process level-process diagram **Process P**  Carries out Interacts Roles Activities Role E Role A Process P Item X Role B Item Y	**Process level-process diagram** **Process S** WHERE Role B generalizes Role C AND Role A composes Role D AND Item X is composed of Item W1 AND Item X is composed of Item W2 AND Item Y generalizes Item V  Interacts Carries Out Roles Activities Role C Role F Role D Item W1 Item W2 Process S Item V
Process dependency diagram for Process P & S  Process P Item X Item W1 Item W2 Process S Item Y Item V	

Figure 58 presents how dependencies between two processes (manage change and manage configuration) can be generated from two process level diagrams for each process. The dependency is established via configuration items the configuration manager receives from project team members. The configuration manager generalizes the items it receives from project team members such as a change request forms received from the change manager.

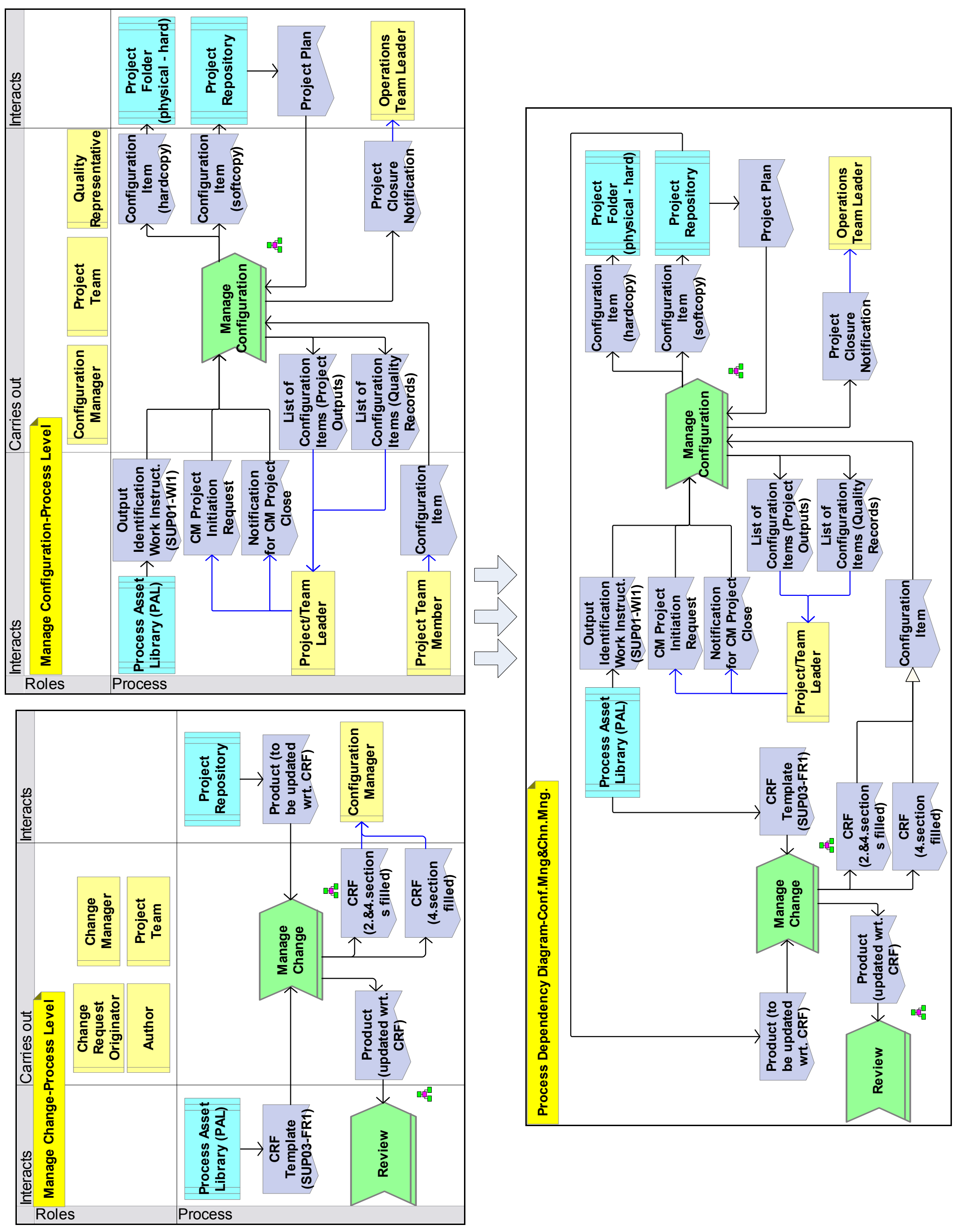

Manage Configuration-Process Level
Roles
Process
Interacts
Carries out
Configuration Manager
Project Team
Quality Representative
Project Folder (physical - hard)
Project Repository
Project Plan
Operations Team Leader
Configuration Item (hardcopy)
Configuration Item (softcopy)
Project Closure Notification
Manage Configuration
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request
Notification for CM Project Close
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
Configuration Item
Process Asset Library (PAL)
Project/Team Leader
Project Team Member
Manage Change-Process Level
Change Request Originator
Author
Change Manager
Project Team
Project Repository
Product (to be updated wrt. CRF)
Configuration Manager
Manage Change
CRF (2.&4.sections filled)
CRF (4.section filled)
Product (updated wrt. CRF)
CRF Template (SUP03-FR1)
Review
Process Dependency Diagram-Conf Mng&Chn.Mng.

Figure 58. A generation of process dependency diagram for manage change and configuration processess

3.7. The Tool for Plural

To be effective, Plural method should be supported by adequate tools that are used to create and maintain the process-base. Creation and maintenance of diagrams should be supported by the tools that also automate some of the activities - model generation in particular. This section, first, presents high level functional requirements of such a tool. Then it describes the toolset we customized and extended for Plural.

3.7.1. High Level Functional Requirements

This section presents the high level functional requirements of a tool that facilitate the Plural method we described in sections 3.2 through 3.5. In particular, the tool should mainly provide functionalities that would;

- enable agents to model their processes using the modeling notation (described in section 3.6);
- identify and highlight inconsistencies between individual role-process models; and,
- provide agents with a number of process analysis instruments (model generation, model analysis) to visualize the process base mainly to help identify improvement opportunities.

Table 18 lists these functional requirements and their priorities. The word *shall* in requirement statement identifies mandatory requirements, while the words *should* and *might* represent recommended ones. Mandatory requirements denote the ones without which an applicable and acceptable application of the method on real life case studies is possible in practice. Other requirements (recommended requirements) can be handled with practical workarounds and manual operations without critically affecting the application of the method, but sacrificing the real benefit that can be achieved if full support was available. Thus, based on their criticality, the requirements are prioritized.

Table 18. High level functional requirements of a tool for Plural

Requirement	Priority	Description
Rq01 - The tool shall enable the definitions of the context by related diagrams	1	The tool shall enable agents to define and maintain scope and role diagrams (sections 3.6.5 and 3.6.6 respectively).
Rq02 - Agents shall be able to enter their process descriptions separately and concurrently, and maintain them as separate entities	1	Agent shall be able to describe the processes for the roles they are assigned to with individual role-process diagrams. During process descriptions, the tools shall also enable information item diagrams representing the relationships between information items be defined by agents. It shall enable the descriptions to be performed concurrently, i.e. with more than one agent at a time.

Table 18. High level functional requirements of a tool for Plural (continued)

Requirement	Priority	Description
Rq03 - The tool should help agents to verify process diagrams with respect to their semantic rules	2	Agents can be provided analysis functionalities that check individual role-process diagrams with respect to the semantic rules set for them in order to verify their correctness. For the generated models to be correct and consistent, inconsistencies and issues *within* individual descriptions should be resolved. Verification of individual role-process diagrams (section 3.4.3) with respect to their semantic rules (Table 8 p92) can be partially or fully automated. Diagrams can be checked in batches or at any point in the definition process and a report can be generated presenting the rules applied to check for conformance, the results as well as suggestions for resolutions.
Rq04 - The tool shall detect and display inconsistencies between individual role-process diagrams to the agents as they occur (inter-consistency checking)	1	Individual process models, which might be consistent within, might not be consistent among themselves. Thus, these inconsistencies should be located and highlighted by the tool. In contrast to many current approaches to consistency checking, we require tool to identify inconsistencies as they occur, i.e., while agents are modeling their activities. Resolving inconsistencies is necessary in order to generate coherent models. The inconsistencies between individual diagrams and the way they can be resolved are described in section 3.4.2. Automatic inconsistency resolution by the tool is not applicable for theses cases. Therefore, the tool shall identify inconsistencies and supply appropriate assistance in resolution of them.
Rq05 - The tool might provide functions (messaging/chat) for agents to communicate to resolve conflicts	3	Agents negotiate to resolve conflicts. The tool might facilitate the communication by means of both synchronous and asynchronous communication instruments.
Rq06 - The tool should generate models based on the individual role-process diagrams, role diagrams, scope diagrams and information item diagrams	2	The generation of complete models of the process, such as process, operation and activity level process diagrams, as well as global diagrams such as role and process dependencies can be automated by the tool. Agents should be provided functionalities that enable them to select the type of models to be generated as well as filter or limit the generation for specific scope such as a set of roles or processes.
Rq07 - The tool should provide a security scheme for users based on their access rights	2	The user should be able to access only to diagrams and process elements that they are given right to. For example, development and peer agents can be given a workspace where they create and edit their models while they can be provided only read rights on other's workspaces. Coordinator, on the other hand, can be given all rights to the elements in the process-base.
Rq08 - The tool should provide process-base to be baselined at specific points in time, as well as provide functionalities for backup and maintenance of the base	2	The tools should enable to baseline the process-base on specific time intervals, such as end of the phases. Similarly the tool should enable users to backup their workspace (related diagrams) as well as the whole base at any point in the description for recover.

The next section discusses the degree of the support provided by available tools.

3.7.2. Applicability of the Available Tools for Plural

There are several tools used for process modeling in different fields and in different contexts. In Chapter 2, we investigated process centered software engineering environments (PSEEs), tools utilized for Viewpoints approaches, such as V-Elicit, MVP-E, and etc., tools that support enterprise modeling and business process management such as ARIS Collaborative Suite, MO^{2}GO, and etc.; and finally we surveyed tools used in agent-oriented approaches. However, none of the tools we surveyed in the literature fulfills all the requirements of the Plural method listed in Table 18. This is mainly due to the unique approach the Plural tries to address, which led to special requirements to be adopted. Many of the tools provide support for the first two requirements (Rq01 and 02) with no or little extensions or modifications. However, there is a lack of support for requirements related to inter-consistency identification and model generation (04 and 06, respectively). These requirements (particularly 04) can be considered to be critical and unique to Plural method and the lack of support for them forbids any available tool to be directly utilized for the method.

The next section describes the tool and the add-on we developed to support the Plural method.

3.7.3. ARIS Toolset and the Plural Add-on

Based on the requirements stated above, the tool for Plural can either be developed or an available tool can be extended to fulfill method's unique requirements. We decided that a prototypical solution that satisfies the high priority requirements would be sufficient to utilize the Plural method in real life cases. With respect to the effort and resources required for the development of a new software, we decided to develop an add-on to an available tool and make use of its rich features it already has. Critical requirements that are not supported by the tool are addressed by an add-on.

Based on the degree of support on the requirements it provides and its features we chose the ARIS Toolset and Web Designer tools from the ARIS Collaborative Suite [126]. It mainly supports the ARIS framework (section 2.5.2), but it is extended to cover many other modeling languages and frameworks.

The Toolset and Web Designer are two modeling environments which are client-server based and web-based respectively. Enterprise related models and all elements are stored in and retrieved from a local or a central database. In our case studies, we generally used the Web Designer via web browsers for modeling processes, which connects to a central database and stores pertinent data on. The add-on is developed as a web-based tool that connects to this central database and analyze process repository to detect and present inconsistencies between individual role-process diagrams.

ARIS tools supports variety of software engineering, enterprise and process modeling notations and related diagrams, including the eEPCs, UML, and BPMN. It has features to define method filters to restrict the users to use a specific set of diagrams and process elements as well as the relationships between these elements. Thus, we defined a specific method filter for agents to use only the diagrams and elements of the Plural notation. Plural's diagrams and process elements, which are based on the ones currently exist in the Toolset, were defined.

Each process element in ARIS is an entity residing in the database. It may or may not occur in diagrams. Elements with the same type and name is allowed but not recommended. When the user creates an element having the same name and type with an element already present in the database, it asks user whether the newly created is the same as the one already present or whether user want to create a new element with the same name and type. This choice is also asked when the user changes the name of an element to a same name of another that is already present.

The ARIS has also security features based on the user access rights. User groups and users can be defined and specific access rights can be given to them for the directories created (workspaces). For Plural, each role defined in the scope and role diagrams in the context definition phase is defined as a user group with a specific workspace where related individual role-process diagrams reside. Agents, then, based on the roles they are assigned to, are associated with these user groups. A user has 'create' and 'edit' rights on the diagrams in the workspaces of the user groups it belongs to and has read rights on the workspaces of the other user groups. These definitions for the user groups and rights are performed and maintained by the tool administrator who is also a member of the coordination team. Each user connects to the process-base with a username and password with related rights based on the user groups they belong to.

The focus of the add-on was on the functions with higher priorities and to develop the parts that are mandatory for utilizations of the method on case studies. Table 19 displays the requirements and how they are fulfilled by ARIS tools and the add-on.

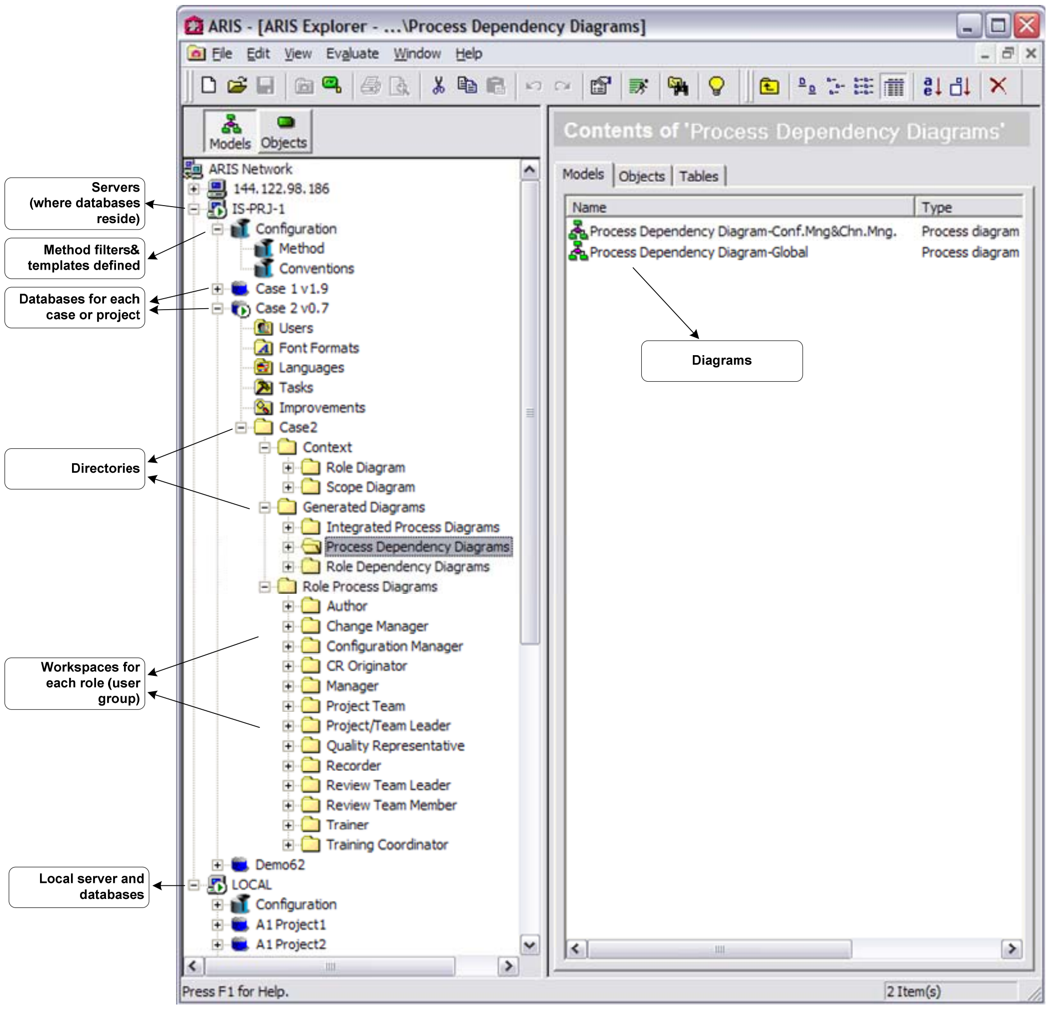
Table 19. Degree of support by the toolset

Requirement	Priority	Degree of Support	Description
Rq01 - The tool shall enable the definitions of the context by related diagrams	1	Largely by ARIS Toolset &Web Designer	The notation for scope and role diagrams is supported with workarounds.
Rq02 - Agents shall be able to enter their process descriptions separately and concurrently, and maintain them as separate entities	1	Fully by ARIS Toolset &Web Designer	ARIS tools enable each agent to define its processes with individual role-process diagrams concurrently via web browsers.
Rq03 - The tool should help agents to verify process diagrams with respect to their semantic rules	2	Fully by ARIS Toolset &Web Designer	ARIS tools enable the user to define semantic rules for the diagrams with an easy and drag & drop type interface. Then users verify diagrams with respect to these rules. The tool generates a report indicating the result for each rule and diagram.
Rq04 - The tool shall detect and display inconsistencies between individual role-process diagrams to the agents as they occur (inter-consistency checking)	1	Fully by The Plural add-on	The primary functionality of the add-on is to identify and present inconsistencies to agents and provide them the ways the inconsistencies can be resolved.
Rq05 - The tool might provide functions (messaging/chat) for agents to communicate to resolve conflicts	3	None	This feature is left as a future work.

Table 19. Degree of support by the toolset (continued)

Requirement	Priority	Degree of Support	Description
Rq06 - The tool should generate models based on the individual role-process diagrams, role diagrams, scope diagrams and information item diagrams	2	Partially by ARIS Toolset, Web Designer & The Plural add-on	ARIS Toolset has functions to generate process diagrams semi-automatically based on the individual role-process diagrams. However, the generated diagrams need manual layout to present the content in a clearer way to the user. The add-on generates role and process dependencies in the text format as a list of role-item-role and process-item-process triples. The generation of diagrams based on these lists is left as future work.
Rq07 - The tool should provide a security scheme for users based on their access rights	2	Fully by ARIS Toolset &Web Designer	This requirement is fully supported by the tools
Rq08 - The tool should provide process-base to be baselined at specific points in time, as well as provide functionalities for backup and maintenance of the base	2	Fully by ARIS Toolset	The tools provide features to baseline and backup all or a part of the process-base.

Figure 59 and Figure 60 displays the primary windows of the ARIS Toolset.



Name	Type
Process Dependency Diagram-Conf.Mng&Chn.Mng.	Process diagram
Process Dependency Diagram-Global	Process diagram

Figure 59. ARIS Toolset: Main Window

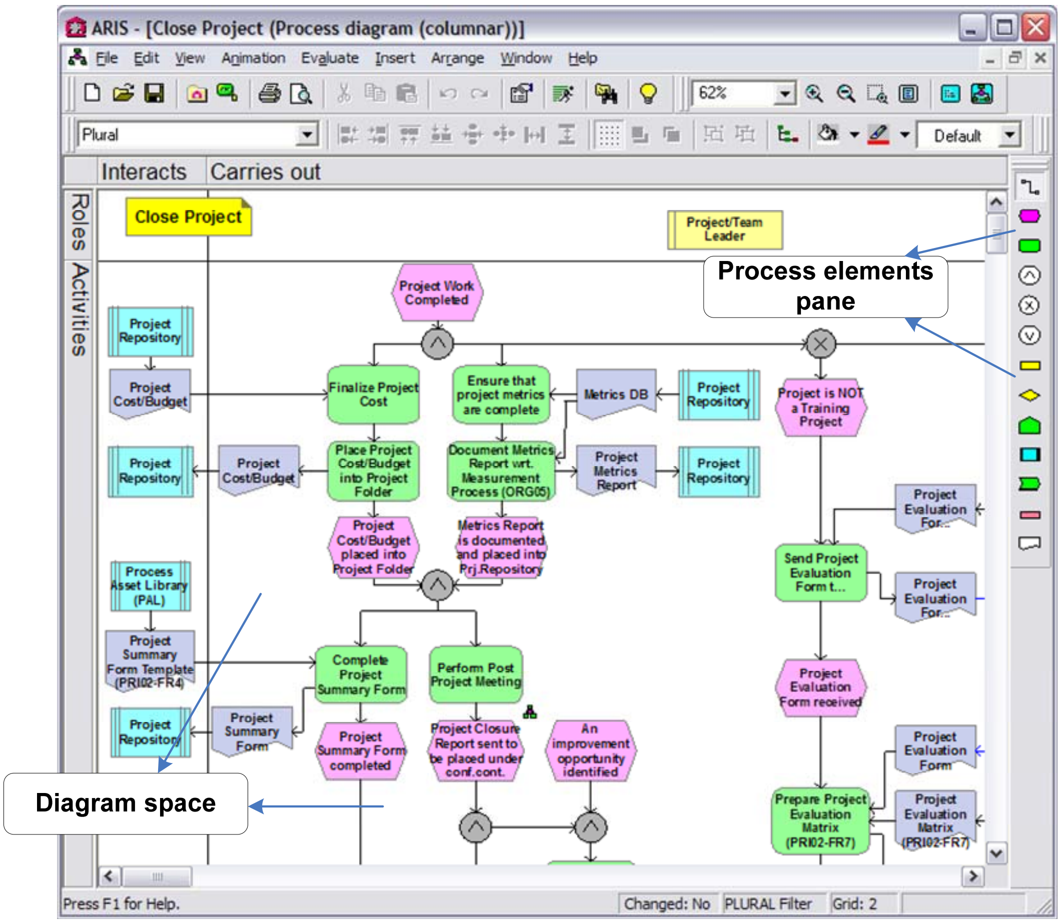
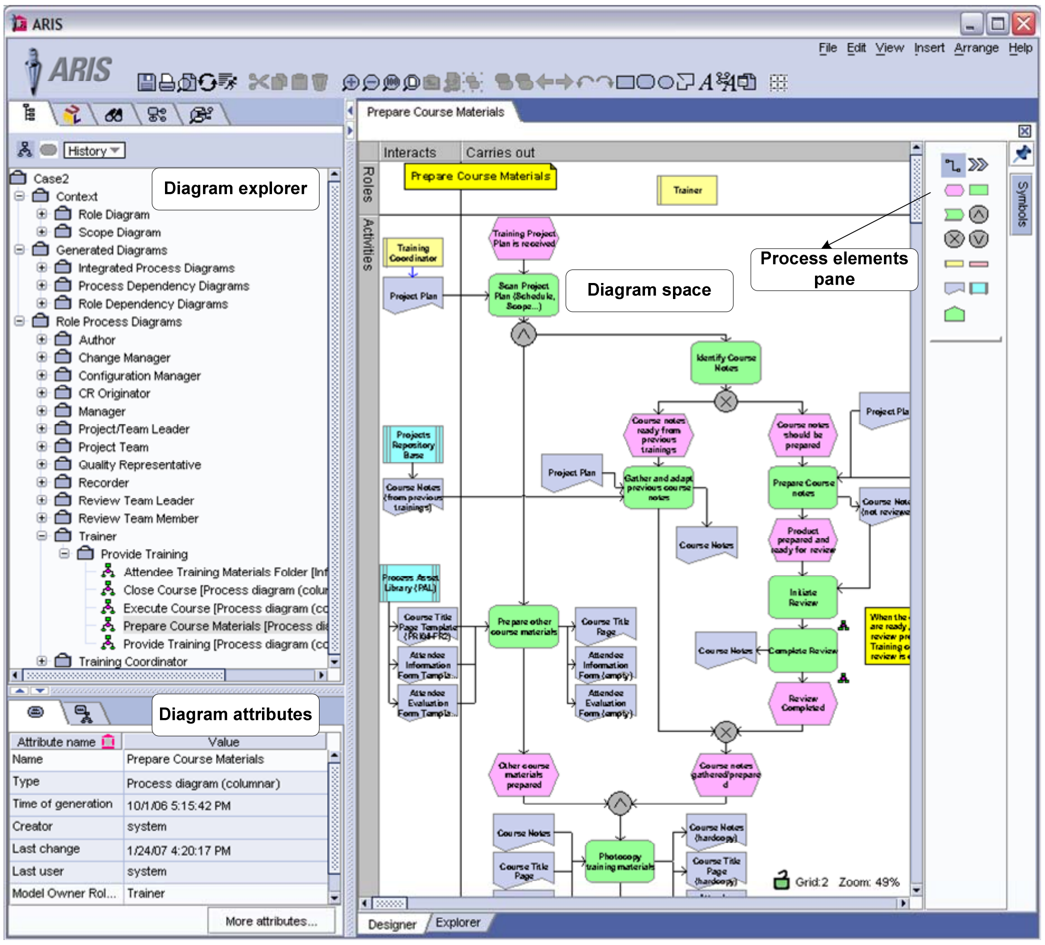

ARIS - [Close Project (Process diagram (columnar))]
File Edit View Animation Evaluate Insert Arrange Window Help
Plural
Default
Interacts
Carries out
Roles
Activities
Close Project
Project/Team Leader
Process elements pane
Diagram space
Project Work Completed
Project Repository
Project Cost/Budget
Finalize Project Cost
Ensure that project metrics are complete
Metrics DB
Project is NOT a Training Project
Place Project Cost/Budget into Project Folder
Document Metrics Report wrt. Measurement Process (ORG05)
Project Metrics Report
Project Evaluation For...
Project Cost/Budget placed into Project Folder
Metrics Report is documented and placed into Prj.Repository
Send Project Evaluation Form t...
Process Asset Library (PAL)
Project Summary Form Template (PRI02-FR4)
Complete Project Summary Form
Perform Post Project Meeting
Project Evaluation Form received
Project Summary Form
Project Summary Form completed
Project Closure Report sent to be placed under conf.cont.
An improvement opportunity identified
Project Evaluation Form
Prepare Project Evaluation Matrix (PRI02-FR7)
Project Evaluation Matrix (PRI02-FR7)
Press F1 for Help.
Changed: No
PLURAL Filter
Grid: 2

Figure 60. ARIS Toolset: Diagram Definition Window

Figure 61 gives the user interface for ARIS Web Designer.

The add-on tool is a web-based tool that connects to the process-base and analyzes models to identify the expectations and inconsistencies between them and provide role dependencies. It enables agents to analyze the expectations and possible inconsistencies at anytime during process description. Agents can check whether other roles' expectations from his role(s) are satisfied or not. This is also true for his role's expectations from others. At a consistent state when there is no inconsistency between expectations, both list become identical.



Attribute name	Value
Name	Prepare Course Materials
Type	Process diagram (columnar)
Time of generation	10/1/06 5:15:42 PM
Creator	system
Last change	1/24/07 4:20:17 PM
Last user	system
Model Owner Rol...	Trainer

Figure 61. ARIS Web Designer: Diagram Definition Window

In terms of the source of the expectations, the analysis of the expectations is performed based on the following forms:

A. With respect to the expectations of other roles from Role A:

A.1. The information items that other roles expect to be output from (delivered from) Role A to them (Role O ← Item X ← Role A)

A.2. The information items that other roles expect to be input to (sent to) Role A from them (Role O → Item X → Role A)

A.3. The activities other roles expect to perform together with Role A (Role O → Activity Z ← Role A)

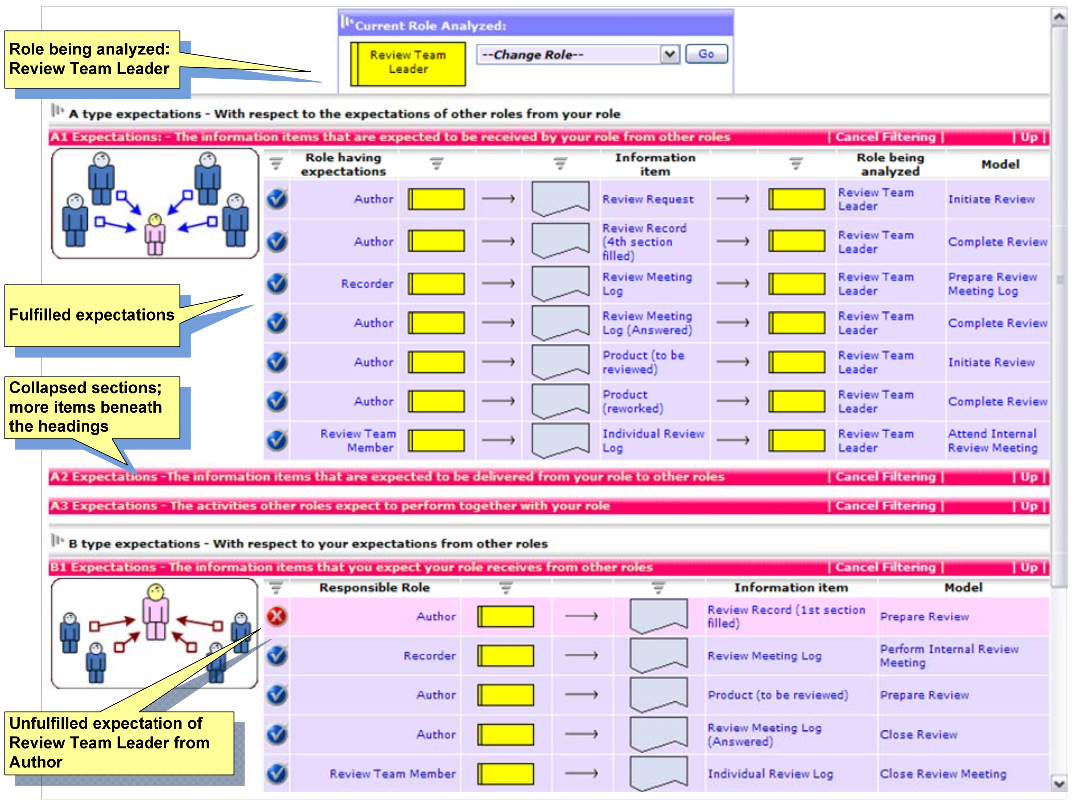
B. With respect to the expectations of Roles A from other roles:

B.1. The information items that Role A expects it sends to other roles (Role A → Item X → Role O)

B.2. The information items that Role A expects it receives from other roles (Role A ← Item X ← Role O)

B.3. The activities that Role A expects it performs together with other roles (Role A → Activity Z ← Role O)

Figure 62 gives a portion of the list of expectations from review team leader role.


Role being analyzed: Review Team Leader
Fulfilled expectations
Collapsed sections; more items beneath the headings
Unfulfilled expectation of Review Team Leader from Author
Current Role Analyzed:
Review Team Leader
--Change Role--
Go

A type expectations - With respect to the expectations of other roles from your role

A1 Expectations: - The information items that are expected to be received by your role from other roles | Cancel Filtering | | Up |

	Role having expectations				Information item			Role being analyzed	Model
✓	Author	🟨	⟶		Review Request	⟶	🟨	Review Team Leader	Initiate Review
✓	Author	🟨	⟶		Review Record (4th section filled)	⟶	🟨	Review Team Leader	Complete Review
✓	Recorder	🟨	⟶		Review Meeting Log	⟶	🟨	Review Team Leader	Prepare Review Meeting Log
✓	Author	🟨	⟶		Review Meeting Log (Answered)	⟶	🟨	Review Team Leader	Complete Review
✓	Author	🟨	⟶		Product (to be reviewed)	⟶	🟨	Review Team Leader	Initiate Review
✓	Author	🟨	⟶		Product (reworked)	⟶	🟨	Review Team Leader	Complete Review
✓	Review Team Member	🟨	⟶		Individual Review Log	⟶	🟨	Review Team Leader	Attend Internal Review Meeting

A2 Expectations -The information items that are expected to be delivered from your role to other roles | Cancel Filtering | | Up |

A3 Expectations - The activities other roles expect to perform together with your role | Cancel Filtering | | Up |

B type expectations - With respect to your expectations from other roles

B1 Expectations - The information items that you expect your role receives from other roles | Cancel Filtering | | Up |

	Responsible Role				Information item	Model
✗	Author	🟨	⟶		Review Record (1st section filled)	Prepare Review
✓	Recorder	🟨	⟶		Review Meeting Log	Perform Internal Review Meeting
✓	Author	🟨	⟶		Product (to be reviewed)	Prepare Review
✓	Author	🟨	⟶		Review Meeting Log (Answered)	Close Review
✓	Review Team Member	🟨	⟶		Individual Review Log	Close Review Meeting

Figure 62. The Add-on: Listing inconsistencies

The expectations that are fulfilled with respect to the rules (section 3.4.2) are highlighted as being satisfied while the others are marked as an inconsistency. All agents are required to analyze their models and expectation for their each role with respect to the items given above and resolve the issues. Resolution is performed in the form of model changes. Model changes are instantly reflected on Plural.

In case of an inconsistency, such as the one given in Figure 62, the agent either changed its description in order to match other's expectations or it insisted on its position and communicated with other agents in order to solve the conflict. As described in section 3.4.2, resolution is under agents' responsibility and automated conflict resolution is not intended in these cases. During process description, coordinator moderated such interactions where necessary, until all inconsistencies were resolved and inter-consistency is established among these models.

The Plural add-on also enables the agents to check the interaction between active roles and the other entities including the inactive roles, application systems and information stores. The list provides the interface of the processes from its external world. Agents check the integrity and consistency of this interaction during phase 2 (description and conflict resolution).

The details about the source code and installation procedures of the Plural add-on are provided in Appendix G.

CHAPTER 4

APPLICATION OF THE METHOD

This chapter presents the application of the Plural method on a multiple-case study involving two case studies; first in a university department and second in a software development and consultancy company. Section 4.1 elaborates the design of our multiple-case study. It describes the questions of the study, the propositions, and data collection and analysis strategies. In section 4.2 and 4.3, we briefly describe the conduct of the two cases. Section 4.4 discusses our findings and lessons learned.

4.1. Multiple Case Study Design

Case studies have been a common research strategy in many fields, including psychology, sociology, political science, business, social work, information systems [65] , [110], [156], [158]. This can be attributed to its strength in helping investigators to understand complex social inter-relationships [69]. They rely on multiple sources of evidence and benefit from the prior development of theoretical propositions. They can be based on qualitative as well as quantitative evidence.

In order to explore the applicability of the method for decentralized process modeling and to uncover improvement opportunities for the method, we conducted a multiple-case study involving two cases. First case study was conducted in a university department and the second case study was performed in a small software development and consultancy company and included a set of its engineering processes.

We considered the case study as an appropriate research strategy to investigate the application of the Plural method in real-life organizational settings and to examine its implications on individuals and on case organizations. This is because case studies are preferred research methods in examining contemporary events when the behaviors cannot be manipulated by the investigator [157]. During our examinations in case studies, we had no control over the behavioral events.

Case studies can be designed in many ways; but the primary distinction in designing case studies is between *single-* and *multiple-case* designs [157]. Single case designs are generally for instances when there are no other cases available for replication. Multiple case designs include single cases, but design must follow a replication rather than sampling logic for each single case. In this thesis study, we used multiple-case design strategy, since, a multiple case often considered more compelling as it strengthens the results by replicating the pattern-matching and increasing confidence in the robustness of the theory. The replication approach we followed in our multiple-case study design is depicted in Figure 63 (based on [157]).

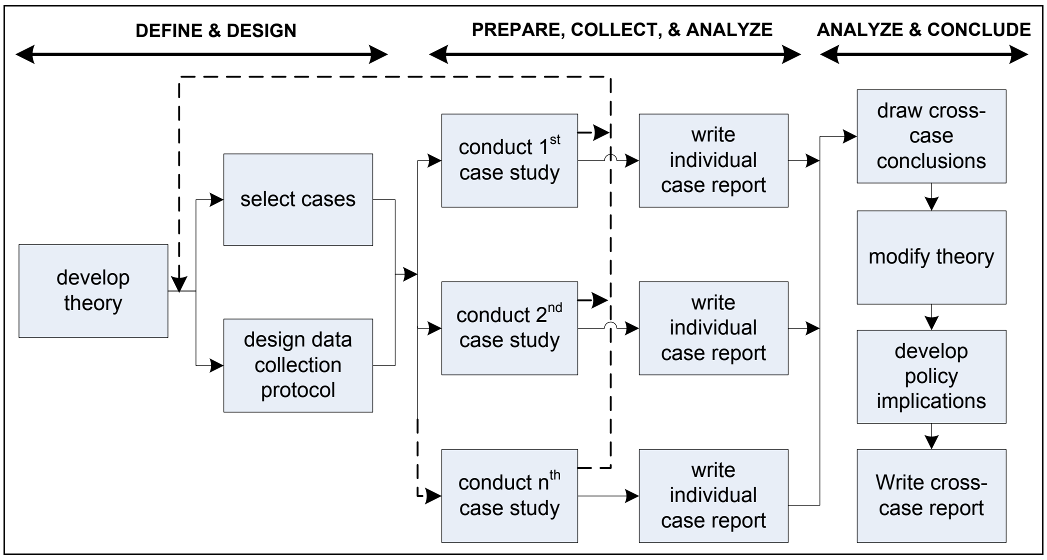

DEFINE & DESIGN
PREPARE, COLLECT, & ANALYZE
ANALYZE & CONCLUDE
develop theory
select cases
design data collection protocol
conduct 1st case study
conduct 2nd case study
conduct nth case study
write individual case report
write individual case report
write individual case report
draw cross-case conclusions
modify theory
develop policy implications
Write cross-case report

Figure 63. Case Study Method

4.1.1. Case study questions and propositions

Our multiple-case study has the following primary research question:

"How applicable is the Plural method for decentralized business process modeling?"

We were also interested in identifying improvement points and enhancing the method and other components, particularly in this first case study. Thus, the following question was also particularly relevant for the first case:

"What improvement opportunities can be identified for the method?"

The focus of the first case was on exploring the application of the method; examine its drawbacks and enhancing its structure and components. The requirements for the notation and the toolset were also elucidated and solidified. Based on the findings in this case, the method and other components of the Plural were improved. Hence, this case utilized the earlier versions of the method and the notation proposed in the study. The differences are highlighted when necessary. The emphasis in the second case was on exploring and evaluating the applicability of the method and observing expected benefits.

We propose that the method is applicable for modeling organization's processes in a decentralized way and benefits of decentralizing the modeling process can be gained by the individuals and in turn by the organization. In particular, we claimed that:

1. An organization following the Plural method in modeling its processes can perform process modeling in less time (due to concurrency in individual modeling)

2. Considering all participants' viewpoints and expectation serves model completeness

3. The method eases incorporating process change (due to decentralization and consistency tracking based on role expectations)

4. The method helps discovering the interaction points between roles, expectations as well as conflicts and makes them visible to all participating agents

For the first proposition, if the method enables concurrent process modeling in an effective manner, then due to the concurrency in modeling, the total time for building the organization's process-base decreases. In other words, as the number of the agents participating in concurrent process modeling increases, the probability of having a decrease in the total time increases, since the effort for modeling is shared among these participants. This proposition is certainly valid with the assumption that the method provides appropriate ways and mechanisms that enables the total effort for modeling processes to remain as it is (if not decreases) or to increase to a value which still results a lower load per participant. However, it is difficult (if possible) to test the validity of the assumption with real organizational settings. Thus we focus on the total time it takes to model organization's processes and compare it with the values for similar previous projects and experts' judgments. The effort utilized for the study for each phase was tracked by each agent.

As discussed in the related literature, researches in process modeling and requirements engineering literature reveal that the approaches considering process information to be captured from multiple perspectives (viewpoints) help model completeness [54], [56], [112], [145]. The Plural's support for the second proposition (method facilitating model completeness) is inherent, since the processes are modeled by the process participants and it is ensured that all other interacting agents define their expectations explicitly from the participating agents and implicitly from the process itself. In order to explore whether this proposition is maintained by the case studies or not, we sought for two sources of evidence: First we examined the differences between existing prior process definitions (if exist) and the ones developed with the Plural method and noted the captured and missed elements; Secondly, we interviewed with the agents and their peers for their outlook for the degree the method helped them to uncover incompleteness and ambiguities in prior definitions and its ability to surface any implicit and incorrect assumptions hold by the participants. In interviews, we also tried to investigate the extent the models reflect the actual execution of the processes. We had a chance to have on-site observations during and after the case study conduct and seized any signs of supporting evidence.

The third proposition is facilitated by the structure of the Plural method and the notation utilized for this purpose. In Plural, since the individual models are maintained, process change is also pursued in a decentralized way. If a change does not have an effect on role's expectation, then it is reflected as a simple alteration on role's individual model. If, however, a change alters role's expectations, then one or more inconsistencies (in terms of role's expectations) are raised in the process-base, which should be solved by related agents. The agent may revoke the change or persuade others to change their definitions to reach to another consistent state. Macro level process changes (establishing new processes, process decay, changes in roles) are handled by the Plural process group and reflected by coordinators in scope and role diagrams. These changes may also include micro level changes that should be handled by development agents.

The Plural method is mainly based on agents unveiling their expectations from others (in terms of the information they receive/produce and the activities they perform together with others) and negotiating with others to ensure that these expectations are fulfilled. In this respect, the successful application of the method enables agents to discover the interaction points between roles, expectations as well as conflicts and makes them visible to all participating agents (the fourth proposition). In the integration phase, agents develop role-dependency models that explicitly present the information dependencies between roles. During the case study conduct we observed and noted conflicts between agents and the way they were handled. During interviews we tried to capture agents' views about method's ability to surface any implicit and incorrect assumptions hold by the participants. We prepared a set of questions for the participants to be used as a guideline in interviews. After the first case study, the questions were organized into a questionnaire. The questionnaire and answers to the questions for the second case study are given in Appendix F.

During the case study conduct and in interviews we also tried to identify improvement opportunities for the method as well as other components – notation and the toolset.

4.2. Case Study 1

4.2.1. Background

The case study is performed in the Department of Information Systems in Informatics Institute, METU (Middle East Technical University). The processes covered in the case study mainly included 'student admissions and enrollments', 'staff recruitments' and 'instructing' processes (complete list of processes is given in section 4.2.2.a in Figure 64). These processes are usually regulated by university rules and specific official legislations and are usually written as specific guidelines or procedures to be followed by the participants. The development agents mostly resorted to these short prescriptions during process definitions.

The Information Systems Department provides graduate degrees including Master's (M.S.) degrees in Information Systems and Software Management and Doctor of Philosophy (Ph.D.)

degree in Information Systems for its 194 M.S. and 57 Ph.D. students. Department also provides a service course to all university undergraduate students (around 1100 students each semester). It has 9 full-time faculty members and 28 research assistants.

The current definitions were considered partial (representing activities from a single viewpoint), incomplete and ambiguous by the department members and the organization was seeking ways and methods to come up with a complete model of the processes that would represent the way the organization works from multiple perspectives and would enable an appropriate guidance particularly for future performers. This motivation provided a critical provision for the case study to be performed in the organization and eased the way the method is accepted and embraced by the participants.

The study group comprised six agents two of which also participated as coordinators. Five of the agents have a graduate degree in computer science or similar subjects while one agent has a degree from a school of higher education. Only one of the coordinators was familiar with the Plural notation and toolset, whereas others were not directly acquainted with the method and related notation and tools utilized in the study.

The next section describes the details the conduct of the first case study.

4.2.2. Overview of the Case Study 1 Conduct

a. Phase I - Context Definition

We started the study with a kick off meeting with all participating agents and the top level managers of the department. After a short presentation about the framework, the study group determined the aim of the project as to model the processes of the organization in order to facilitate human understanding and communication and guide current and future performers. The study group then started identifying the processes to be covered in the study. A preliminary work done before the meeting about the current high level processes performed in the organization was helpful in identifying the process to be included.

The initial scope included 22 high level processes performed in the organization. However, during the description and the conflict resolution phase, study group decided to narrow the scope to 12 high level processes that they rank the highest priority to the organization in terms of their extent, the frequency of performance and criticality. During subsequent meetings the coordinators and development agents determined the active roles that participates in these processes and completed the scope diagram as depicted in Figure 64.

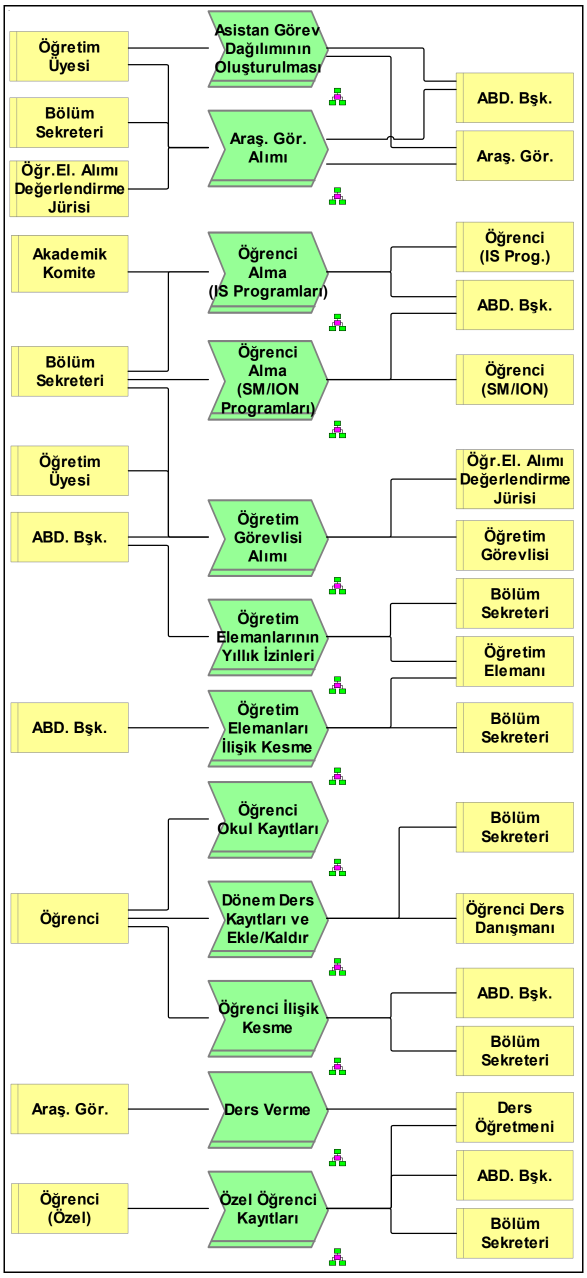


Figure 64. The Scope Diagram: Case Study 1

Having decided on the processes to be modeled and related active roles, the study group then identified the relationships between the roles and depicted the role diagram as given in the Figure 65. Initial versions of the role diagram depicted the relationships between active roles and lacked some of the inactive (external) roles that surfaced as the group proceeded into the description phase. Thus, the role diagram was subjected to the changes in subsequent phases; however, these changes were particularly related to inactive roles which did not affect the scope or the execution plan significantly.

Based on their responsibilities in the organization, each participating agent took over the active role depicted on the diagrams. The group made sure that each active role is assigned to at least one development agent. The coordinators, then, prepared the execution plan and distributed to the group and ensured that each participant had a consensus over the information depicted on the diagrams and the included in the plan. The final version of the plan is given in Appendix B.

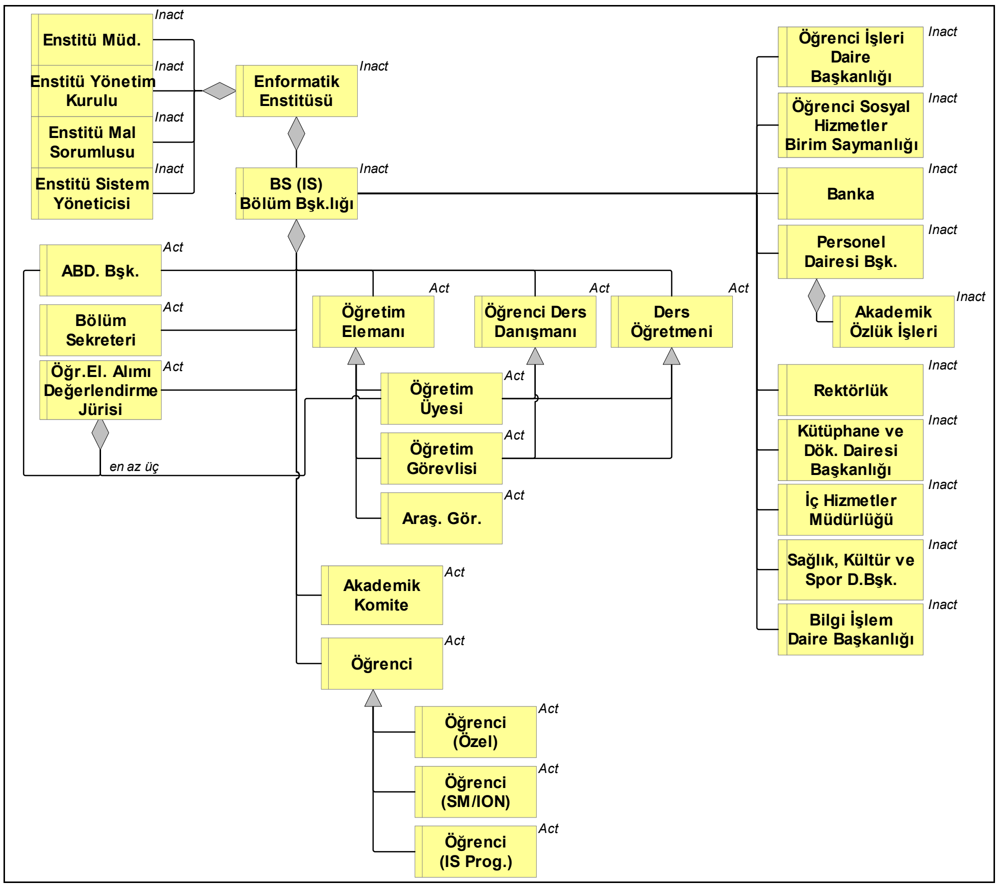


Figure 65. The Role Diagram: Case Study 1

b. Phase II - Description and Conflict Resolution

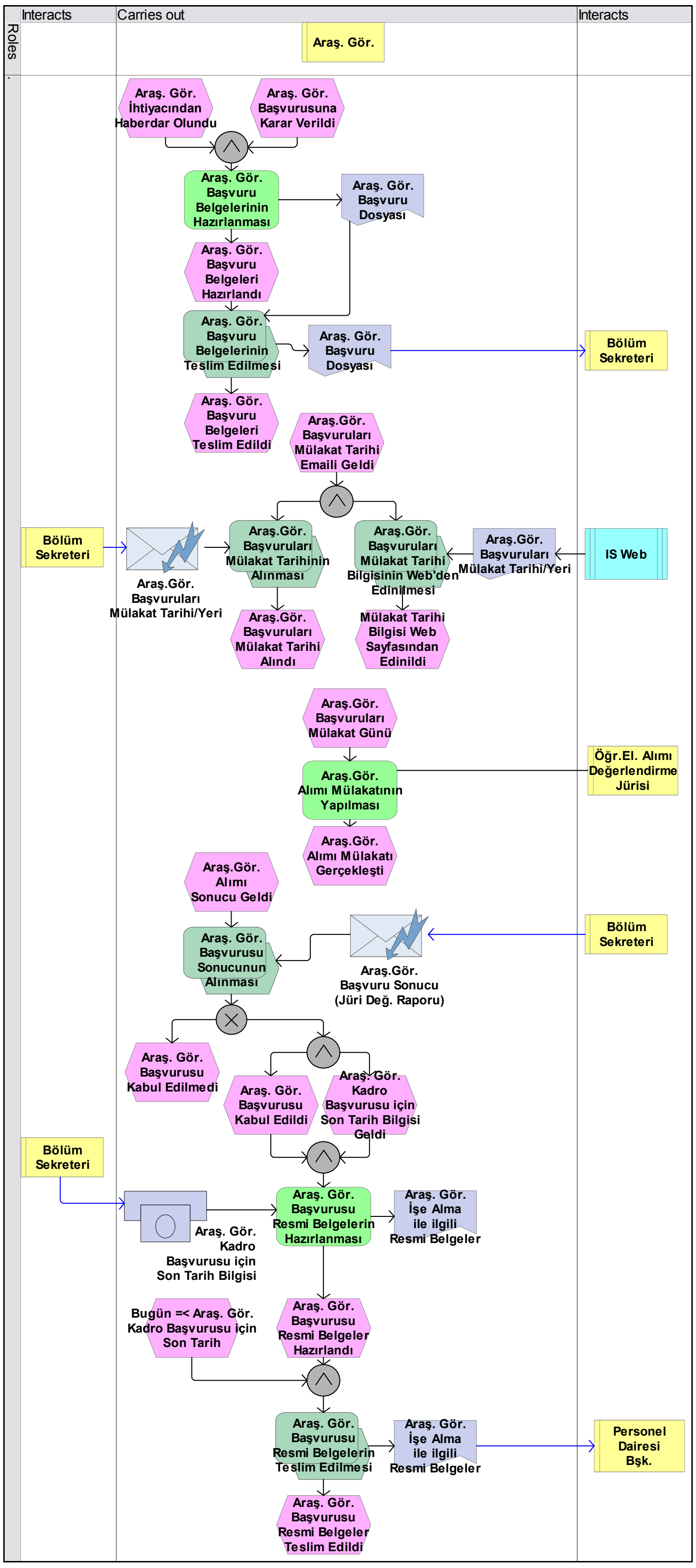
The phase commenced when each development agent started modeling the activities of the active roles they were associated with. In an individual role-process diagram, each agent represented the activities s/he performs with respect to the process and the role s/he plays. For example, the Agent A, being associated with role 'research assistant' started modeling the activities he performs in 'research assistant recruitment' process. For some of the processes, modeling was performed asynchronously.

Before a development agent started individual process modeling, the coordination team performed an orientation session to the agent. Sessions ranged between 0.5 to 1.5 hours for each agent depending on his/her familiarity to related concepts. In these sessions, issues related with the method, the notation and the tools are communicated. For some of the agents, individual process modeling continued in pairs - with a coordinator. This helped the agents to adapt to the process rapidly thus increased their effectiveness and efficiency for modeling the rest of the processes. It also eased the verification of related models done by the coordinators.

As described in the Plural method (see section 3.4.1), the method first requires each agent to identify the operations they service in a particular process and then represent the details of each operation in an individual role-process diagram. However, this first case utilized the earlier versions of the method and the notation, the method was utilized in a slightly different manner. Instead of literally identifying the operations, the development agents represented the activities they perform in a process in one individual role-process diagram and conceptually list all operations on that diagram. This led to complexity and decreased the readability and understandability of diagrams. Thus, at the end of the case, based on this and other related finding, we introduced the 'operation' concept into the model (refer to section 3.4.1 for the description of the concept).

Figure 66 presents an individual role-process diagram listing all activities of a role with respect to a specific process. In such a representation, the role presents all its operations - the services- without explicitly naming the operation. It represents the details including the activities it performs, the information items it requires and produces. Sources of the inputs and destination of the outputs are also represented. Sources and destinations are other roles, information stores such as bookshelves or application systems where information resides. Earlier versions of the notation included specific icons for representing the type of information items such as email, fax, verbal information, telephone, and etc. Based on our observations in this case study, later versions excluded these icons and unified the representation of the information items to simplify the representation.

Agents represented their expectations as the exchange of information items from other roles. During process descriptions they checked whether they satisfied others expectations and whether their expectations are fulfilled by other roles. For process descriptions and checking expectations they used ARIS Web Designer and the Plural add-on (section 3.7.3).



Interacts
Carries out
Interacts
Roles
Araş. Gör.
Araş. Gör. İhtiyacından Haberdar Olundu
Araş. Gör. Başvurusuna Karar Verildi
Araş. Gör. Başvuru Belgelerinin Hazırlanması
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Belgeleri Hazırlandı
Araş. Gör. Başvuru Belgelerinin Teslim Edilmesi
Araş. Gör. Başvuru Dosyası
Bölüm Sekreteri
Araş. Gör. Başvuru Belgeleri Teslim Edildi
Araş.Gör. Başvuruları Mülakat Tarihi Emaili Geldi
Bölüm Sekreteri
Araş.Gör. Başvuruları Mülakat Tarihi/Yeri
Araş.Gör. Başvuruları Mülakat Tarihinin Alınması
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisinin Web'den Edinilmesi
Araş.Gör. Başvuruları Mülakat Tarihi/Yeri
IS Web
Araş.Gör. Başvuruları Mülakat Tarihi Alındı
Mülakat Tarihi Bilgisi Web Sayfasından Edinildi
Araş.Gör. Başvuruları Mülakat Günü
Araş.Gör. Alımı Mülakatının Yapılması
Öğr.El. Alımı Değerlendirme Jürisi
Araş.Gör. Alımı Mülakatı Gerçekleşti
Araş.Gör. Alımı Sonucu Geldi
Araş. Gör. Başvurusu Sonucunun Alınması
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
Bölüm Sekreteri
Araş. Gör. Başvurusu Kabul Edilmedi
Araş. Gör. Başvurusu Kabul Edildi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi Geldi
Bölüm Sekreteri
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. Başvurusu Resmi Belgelerin Hazırlanması
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Bugün =< Araş. Gör. Kadro Başvurusu için Son Tarih
Araş. Gör. Başvurusu Resmi Belgeler Hazırlandı
Araş. Gör. Başvurusu Resmi Belgelerin Teslim Edilmesi
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Personel Dairesi Bşk.
Araş. Gör. Başvurusu Resmi Belgeler Teslim Edildi

Figure 66. The activities of role ‘Araş. Gör.’ (Research Assistant) in ‘Research Assistant Recruitment’ process

Unlike the current version of the method, model validation was not performed explicitly with the assumption that models can be validated by the individuals that perform the processes and for our case they were the individuals that actually modeled their own processes. Thus, we assumed that the models developed by these individuals are valid as they are completed. However, we later found out that for few cases (in particular, student role in student quits process, academic staff role in academic staff quits process), models convey incomplete information in terms of process behavior, although the models are consistent in terms of roles' expectations and verified with respect to notation's semantics. Based on this finding, we introduced 'peer agent' role (described in 3.3.5) into the method that is responsible from the validation of the individual models to ensure that individual models describe correct information before they are integrated.

c. Phases III - Integration and Change

The integration and change phase commenced once the individual process description is complete, that is, individual models are validated and verified. Coordinators merged these diagrams into activity level process diagrams to visualize the process as a whole. The full representation of the process was necessary in order to provide a top-view to all process participants and other stakeholders. The add-on tool we used had features to perform that integration automatically; however, the integrated model needed manual layout to present the diagram in a clearer way to the user. Figure 67 presents a partial diagram for the 'research assistant recruitment' process (the complete diagram is given in Appendix C Figure 89).

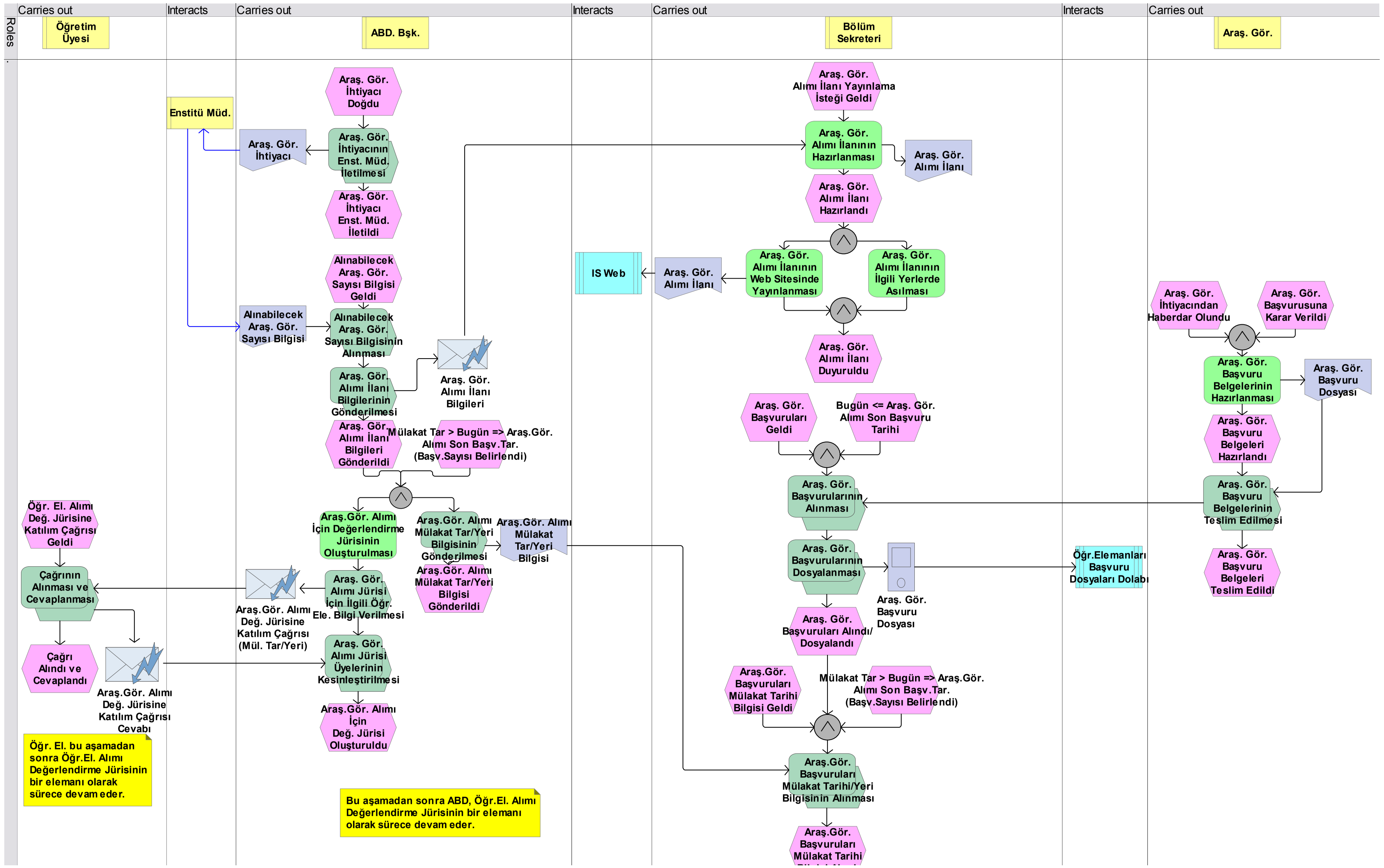

Roles
Carries out
Interacts
Carries out
Interacts
Carries out
Interacts
Carries out
Öğretim Üyesi
ABD. Bşk.
Bölüm Sekreteri
Araş. Gör.
Enstitü Müd.
IS Web
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. İhtiyacı Doğdu
Araş. Gör. İhtiyacının Enst. Müd. İletilmesi
Araş. Gör. İhtiyacı
Araş. Gör. İhtiyacı Enst. Müd. İletildi
Alınabilecek Araş. Gör. Sayısı Bilgisi Geldi
Alınabilecek Araş. Gör. Sayısı Bilgisi
Alınabilecek Araş. Gör. Sayısı Bilgisinin Alınması
Araş. Gör. Alımı İlanı Bilgilerinin Gönderilmesi
Araş. Gör. Alımı İlanı Bilgileri
Araş. Gör. Alımı İlanı Bilgileri Gönderildi
Mülakat Tar > Bugün => Araş.Gör. Alımı Son Başv.Tar. (Başv.Sayısı Belirlendi)
Araş.Gör. Alımı İçin Değerlendirme Jürisinin Oluşturulması
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisinin Gönderilmesi
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi Gönderildi
Araş. Gör. Alımı Jürisi İçin İlgili Öğr. Ele. Bilgi Verilmesi
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı (Mül. Tar/Yeri)
Araş. Gör. Alımı Jürisi Üyelerinin Kesinleştirilmesi
Araş.Gör. Alımı İçin Değ. Jürisi Oluşturuldu
Bu aşamadan sonra ABD, Öğr.El. Alımı Değerlendirme Jürisinin bir elemanı olarak sürece devam eder.
Öğr. El. Alımı Değ. Jürisine Katılım Çağrısı Geldi
Çağrının Alınması ve Cevaplanması
Çağrı Alındı ve Cevaplandı
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı Cevabı
Öğr. El. bu aşamadan sonra Öğr.El. Alımı Değerlendirme Jürisinin bir elemanı olarak sürece devam eder.
Araş. Gör. Alımı İlanı Yayınlama İsteği Geldi
Araş. Gör. Alımı İlanının Hazırlanması
Araş. Gör. Alımı İlanı
Araş. Gör. Alımı İlanı Hazırlandı
Araş. Gör. Alımı İlanının Web Sitesinde Yayınlanması
Araş. Gör. Alımı İlanının İlgili Yerlerde Asılması
Araş. Gör. Alımı İlanı
Araş. Gör. Alımı İlanı Duyuruldu
Araş. Gör. Başvuruları Geldi
Bugün <= Araş. Gör. Alımı Son Başvuru Tarihi
Araş. Gör. Başvurularının Alınması
Araş. Gör. Başvurularının Dosyalanması
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuruları Alındı/ Dosyalandı
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisi Geldi
Mülakat Tar > Bugün => Araş.Gör. Alımı Son Başv.Tar. (Başv.Sayısı Belirlendi)
Araş.Gör. Başvuruları Mülakat Tarihi/Yeri Bilgisinin Alınması
Araş.Gör. Başvuruları Mülakat Tarihi
Araş. Gör. İhtiyacından Haberdar Olundu
Araş. Gör. Başvurusuna Karar Verildi
Araş. Gör. Başvuru Belgelerinin Hazırlanması
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Belgeleri Hazırlandı
Araş. Gör. Başvuru Belgelerinin Teslim Edilmesi
Araş. Gör. Başvuru Belgeleri Teslim Edildi

Figure 67. A partial activity level process diagram for ‘Research Assistant Recruitment’ process

Process level diagrams were also generated based on the individual models. An example for the 'research assistant recruitment' process is presented in Figure 68. Process level diagrams helped agents to visualize the overall interface of the process from other roles (external or internal) as well as information stores. Agents had a chance to examine any interacting parties or information items that could have been neglected. In this case study, these overlooked items were spotted on individual diagrams during process descriptions before these aggregate diagrams were generated.

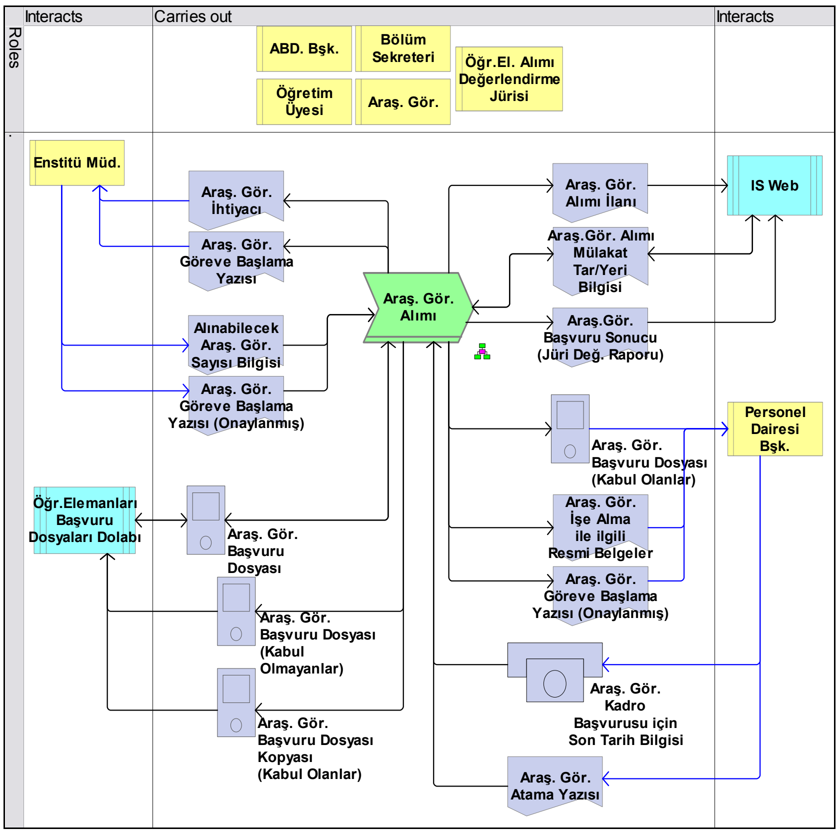

Interacts
Carries out
Interacts
Roles
ABD. Bşk.
Bölüm Sekreteri
Öğr.El. Alımı Değerlendirme Jürisi
Öğretim Üyesi
Araş. Gör.
Enstitü Müd.
Araş. Gör. İhtiyacı
Araş. Gör. Göreve Başlama Yazısı
Alınabilecek Araş. Gör. Sayısı Bilgisi
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Araş. Gör. Alımı
Araş. Gör. Alımı İlanı
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
IS Web
Personel Dairesi Bşk.
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Dosyası (Kabul Olmayanlar)
Araş. Gör. Başvuru Dosyası Kopyası (Kabul Olanlar)
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. Atama Yazısı

Figure 68. A process level process diagram for 'Research Asst. Recruitment' process

To visualize the role relationships, agents also generated role dependency diagrams such as the ones depicting the information item exchange between roles within 'instructor recruitments' process. Figure 69 presents the item exchange between the 'Information Systems Department' (BS Bölüm Başkanlığı) role (which is composed by all active roles scoped in the study) and all other inactive roles.

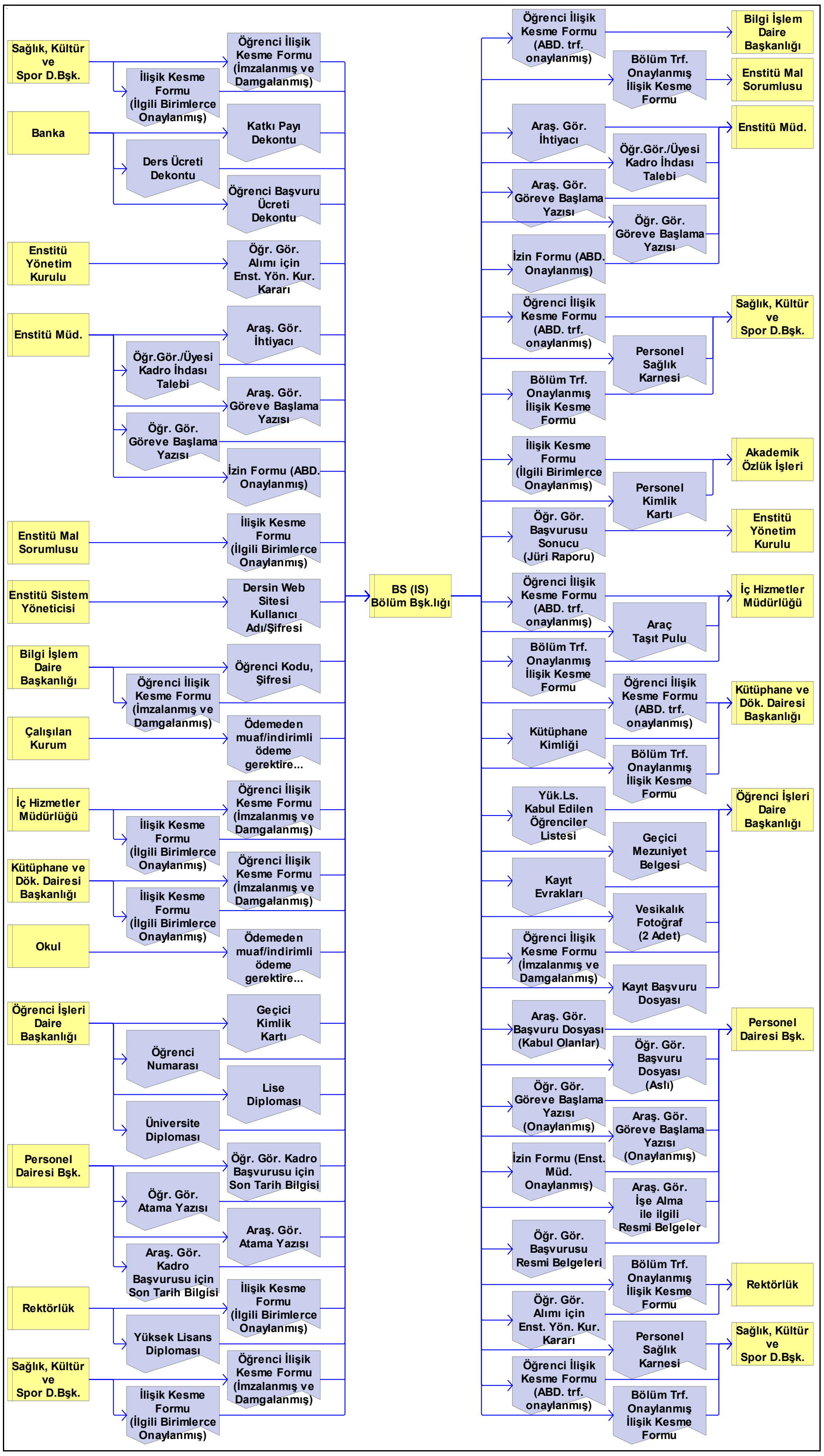

Sağlık, Kültür ve Spor D.Bşk.
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Banka
Katkı Payı Dekontu
Ders Ücreti Dekontu
Öğrenci Başvuru Ücreti Dekontu
Enstitü Yönetim Kurulu
Öğr. Gör. Alımı için Enst. Yön. Kur. Kararı
Enstitü Müd.
Araş. Gör. İhtiyacı
Öğr.Gör./Üyesi Kadro İhdası Talebi
Araş. Gör. Göreve Başlama Yazısı
Öğr. Gör. Göreve Başlama Yazısı
İzin Formu (ABD. Onaylanmış)
Enstitü Mal Sorumlusu
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Enstitü Sistem Yöneticisi
Dersin Web Sitesi Kullanıcı Adı/Şifresi
Bilgi İşlem Daire Başkanlığı
Öğrenci Kodu, Şifresi
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
Çalışılan Kurum
Ödemeden muaf/indirimli ödeme gerektire...
İç Hizmetler Müdürlüğü
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Kütüphane ve Dök. Dairesi Başkanlığı
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Okul
Ödemeden muaf/indirimli ödeme gerektire...
Öğrenci İşleri Daire Başkanlığı
Geçici Kimlik Kartı
Öğrenci Numarası
Lise Diploması
Üniversite Diploması
Personel Dairesi Bşk.
Öğr. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Öğr. Gör. Atama Yazısı
Araş. Gör. Atama Yazısı
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Rektörlük
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Yüksek Lisans Diploması
Sağlık, Kültür ve Spor D.Bşk.
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
BS (IS) Bölüm Bşk.lığı
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Bilgi İşlem Daire Başkanlığı
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Enstitü Mal Sorumlusu
Araş. Gör. İhtiyacı
Enstitü Müd.
Öğr.Gör./Üyesi Kadro İhdası Talebi
Araş. Gör. Göreve Başlama Yazısı
Öğr. Gör. Göreve Başlama Yazısı
İzin Formu (ABD. Onaylanmış)
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Sağlık, Kültür ve Spor D.Bşk.
Personel Sağlık Karnesi
Bölüm Trf. Onaylanmış İlişik Kesme Formu
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Akademik Özlük İşleri
Personel Kimlik Kartı
Öğr. Gör. Başvurusu Sonucu (Jüri Raporu)
Enstitü Yönetim Kurulu
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
İç Hizmetler Müdürlüğü
Araç Taşıt Pulu
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Kütüphane ve Dök. Dairesi Başkanlığı
Kütüphane Kimliği
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Yük.Ls. Kabul Edilen Öğrenciler Listesi
Öğrenci İşleri Daire Başkanlığı
Geçici Mezuniyet Belgesi
Kayıt Evrakları
Vesikalık Fotoğraf (2 Adet)
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
Kayıt Başvuru Dosyası
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Personel Dairesi Bşk.
Öğr. Gör. Başvuru Dosyası (Aslı)
Öğr. Gör. Göreve Başlama Yazısı (Onaylanmış)
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
İzin Formu (Enst. Müd. Onaylanmış)
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Öğr. Gör. Başvurusu Resmi Belgeleri
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Rektörlük
Öğr. Gör. Alımı için Enst. Yön. Kur. Kararı
Personel Sağlık Karnesi
Sağlık, Kültür ve Spor D.Bşk.
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Bölüm Trf. Onaylanmış İlişik Kesme Formu

Figure 69. Dependency with Inactive Roles

Once all agents considered that the process models are complete, consistent and stable, the application of the method was suspended in the integration and change phase. At this point, the first modeling cycle was considered complete. We, then, interviewed with the agents about the method followed, the notation and toolset utilized, and the way they consider the method helped them to express their knowledge about their processes as well as the way the method can be improved. Our findings and lessons learned both in the first and second case studies are discussed in Section 4.4.

The individual report for this case study is documented as a technical report [140] which also includes all diagrams defined and generated throughout the case study. The list of these diagrams and a subset of the diagrams developed is given in Appendix C. Next section describes the second case study performed in regard to this thesis work.

4.3. Case Study 2

4.3.1. Background

The decentralized process modeling approach is more suitable for knowledge oriented organizations since knowledge in these organizations is highly distributed among workers [50]. Each knowledge worker has his/her own specialty on which s/he masters and continuously improves himself/herself. Software engineering organizations are good examples of this type [138]. Furthermore software engineers are familiar with process modeling and related practices. Therefore, we decided to conduct the second case study in a software organization. The purpose was to explore and evaluate the applicability of the method and observe expected benefits. The participants followed the Plural method in modeling their own processes and were interviewed to provide feedback on the method and other components to further enhance the method and to identify difficulties and limitations as well as advantages and benefits.

The organization that we performed the case study was a small software development and consultancy organization, established in 1999. It currently has 12 knowledge workers where 9 of them work full time. Their established quality system has certified with ISO 9000 in 2000. Workers have graduate degrees in computer science or similar subjects and 4 to 12 years of field experience in software development and related practices. Organization's main fields of business are software development, training on software engineering, independent validation and verification and consultancy on software quality management and process improvement for software development organizations.

The study group consisted of four members all participated as development agents. One of the participants also acted as the coordinator. Development agents were responsible from role-based process description and validation of other individual models as peer agents where necessary. All agents were familiar with process modeling and related concepts. Yet, except the coordinator they were not directly acquainted with the notation and the toolset they utilized for the study.

The organization already had procedures and guidelines for process execution written in natural language, which eased both the initiation and execution of the study during process and roles identifications and individual modeling. Participants, going through the three phases of the method, modeled the activities they perform within the processes covered in the case study.

Next section elaborates each phase and gives the details about their execution in the case study.

4.3.2. Overview of the Case Study 2 Conduct

a. Phase I - Context Definition

The study started with a kick off meeting with all agents and stakeholders. After a brief orientation to the approach, the goal and objectives of the study were determined. The focus was on defining process models that facilitates human understanding and communication, particularly as a means to communicate and guide possible current and future performers.

During the kickoff meeting, the study group also determined the high level processes that will be covered during the study. The group identified the processes performed in at least one type of project carried out in the organization. For that purpose, the training life cycle is selected in which the company provides a set of specific courses to the customers. The organization executes training, project management, review, configuration management and change management processes in training projects. Each process has a key behavior of the organization and expresses a goal that the organization wants to achieve. For example, organization performs review process in order to decrease rework cost and increase the quality of products produced.

During the next meeting the team identified the primary roles that participate in these processes and their relationships. The results provided a framework for all participants about the responsibilities and the scope of the individual process modeling activity. It also provided an important input to the plan for the study. Coordinator agent depicted this coverage on a 'scope diagram' (Figure 70) which defines the span of the entire study. Later in the study, agents defined the activities for each role with an individual role-process diagram.

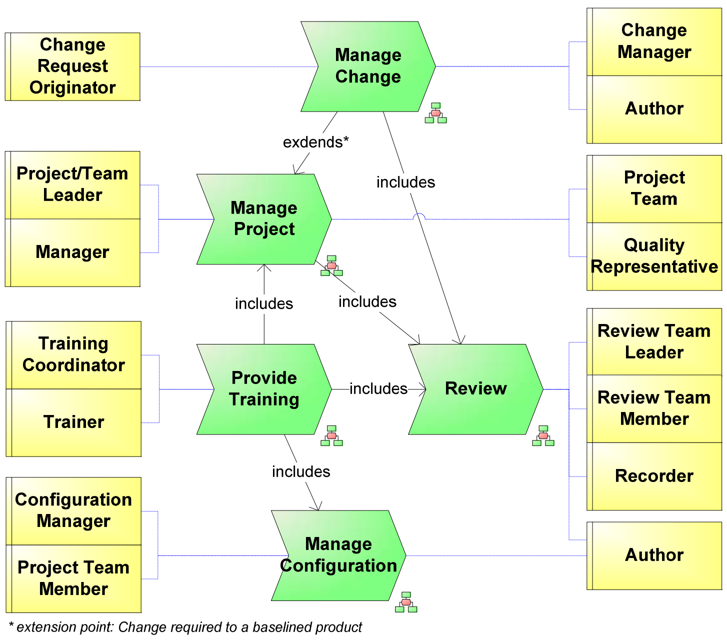


Figure 70. The Scope Diagram: Case Study 2

In identifying and associating the roles in the organization, the inherent static relationship between these roles are revealed. These relationships were in association, aggregation (composition) or generalization type. For example, *trainer* role has (inherits) all of the responsibilities of the *project team member*, which is a more general role. Project team role is composed of other roles like, quality representative, project/team leader, and etc. The study group determined these relationships with respect to the processes covered in the project and the roles that participate in these processes. The coordinator then depicted this scheme on a role diagram as given in Figure 71.

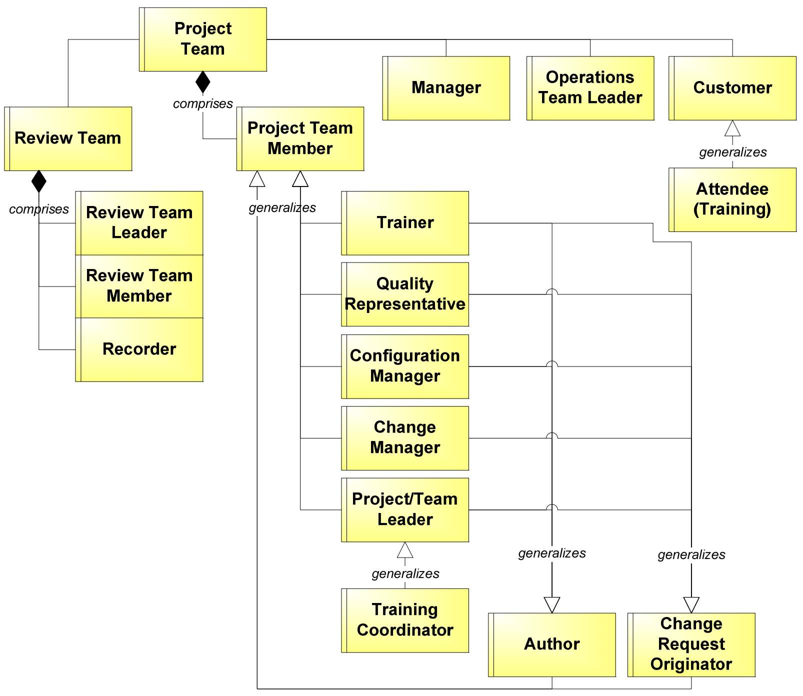


Figure 71. The Role Diagram: Case Study 2

As depicted in the diagram, there was a significant degree of generalization in the environment. For example, configuration manager role inherited all responsibilities of the project team member, author and change request originator roles.

The role diagram, at this phase, included all roles that interact in some way in the execution of the processes. That is active and inactive roles are all depicted on the diagram. Each active role had its own individual role-process model describing the activities it performs in the context of a specific process. Inactive roles did not have such process descriptions. Each active role- in turn- was assigned to an agent in the organization.

The role diagram was not only useful in communicating role relationships to the organization in a structured way but also necessary for consistency checking in terms of role expectations. Later in the project, consistencies checking among individual role process models utilized these relationships (in particular the generalization and aggregation relationships) in order to check whether roles' expectations are fulfilled or not. For example, if project team member role expects a specific information item from the review team role, then based on the aggregation relationships, this expectation can be considered to be satisfied if the review team leader (as a part of the review team role) defines that it sends this item to the project team member. Since there is an aggregation relationship between review team and review team leader role, expectation from the review team can be satisfied by any of the role that constitutes it.

Both diagrams (scope and role) were subjected to changes during the execution of the study, specifically during process definition by agents. However, these changes did not affect the scope significantly.

Having established the consensus on the scope and role definitions, each agent took over the active roles with respect to their actual responsibilities in the organization. The coordinator made sure that no active role is left unassigned. With respect to the scope and related assignments, coordinator documented and distributed a plan for the execution of the study. The final version of the plan is given in Appendix D.

b. Phase II - Process Description and Conflict Resolution

For each development agent, before individual process modeling started, coordinator performed orientation sessions for the procedure to be followed, the notation, the tool and the techniques to be used. These sessions ranged between 0.5 to 1 hours of co-work. All development agents, as playing one or more active roles, started modeling the activities of the roles they play, with respect to the processes covered. For example, a development agent A, who is assigned the 'review team leader' and 'quality representative' roles began modeling the activities he perform as a review team leader in the review process. Modeling started concurrently but for some of the processes, it continued asynchronously.

Firstly, agents identified the operations of each role for the processes that role participates. That is, they identified the services that role provides and present the information in a process diagram. Figure 72 gives an example of such a representation. As depicted on the diagram, *trainer* role defined the following three operations with respect to the *training* process; 'prepare course materials', 'execute course' and 'close course'.

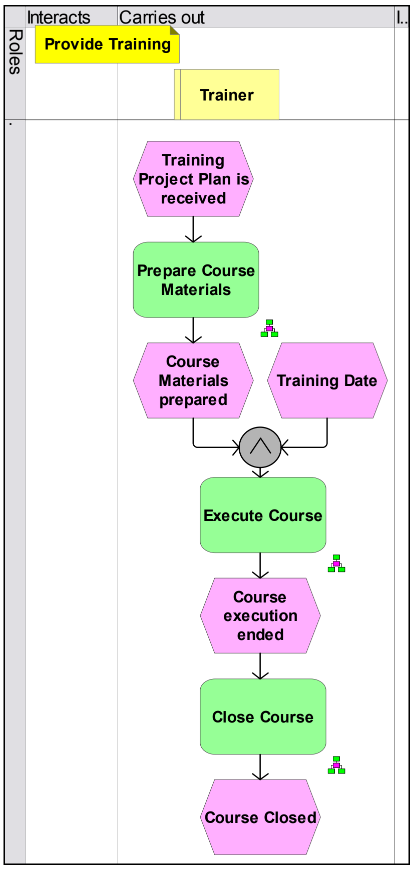


Figure 72. The Operations of the Trainer role in the Provide Training process

The generalization relationships between roles later shaped the extended role diagram. Figure 73 gives the operations of *author* role and two other roles (trainer and change manager) that inherit its operations. For example, based on this definition, the change manager role, in any of its processes, may perform 'initiate review' operation, which is inherited from the author role.

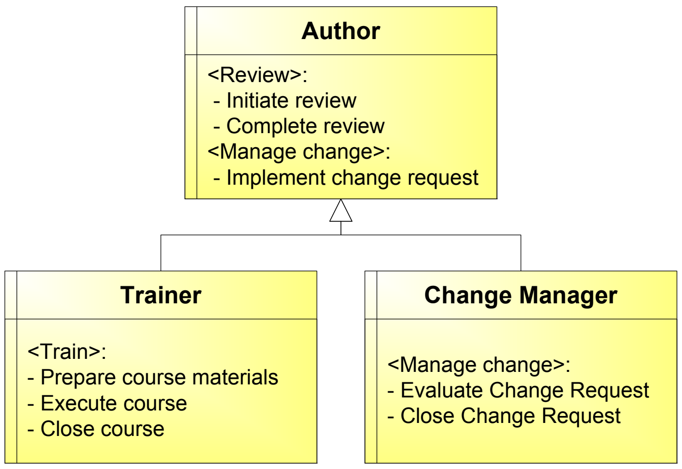


Figure 73. Role Operations

For each of its operations, the role had one or more individual role-process models depicting the behavior of the service it presents. It described the activities it performs in that operation, the information items it requires while performing these activities and the outputs it produces. In addition to that, development agents were asked to represent the sources of the inputs and the destination of the outputs, if any, to and from its role's activities. The sources might represent other roles or items such as project repositories, folders, and software tools. Figure 74 gives an example of an individual role-process diagram for 'initiate project' operation of the 'configuration manager' role in the 'manage configuration' process. The definition of entities providing or gathering information to or from the role's activities formed the expectations of that role from other roles. For example, on the Figure 74, in order configuration manager role to start its operation, it expects 'project initiation request' (from the 'project/team leader role') and gathers 'output identification work instruction' from the 'process asset library'.

Since there are aggregation and inheritance relationship between roles as well as the information items, consistency checking with respect to the expectations also required considering such relationships. These relationships between roles were depicted in the role diagram (see Section 4.3.2.a). On the other hand, the aggregation and inheritance relationships between information items were depicted by the role that generalizes a set of information items. For example, the configuration manager (CM) receives many artifacts from project team members that will be put under configuration control. On an information item diagram, the CM generalized all these items as the 'configuration items' received from team members.

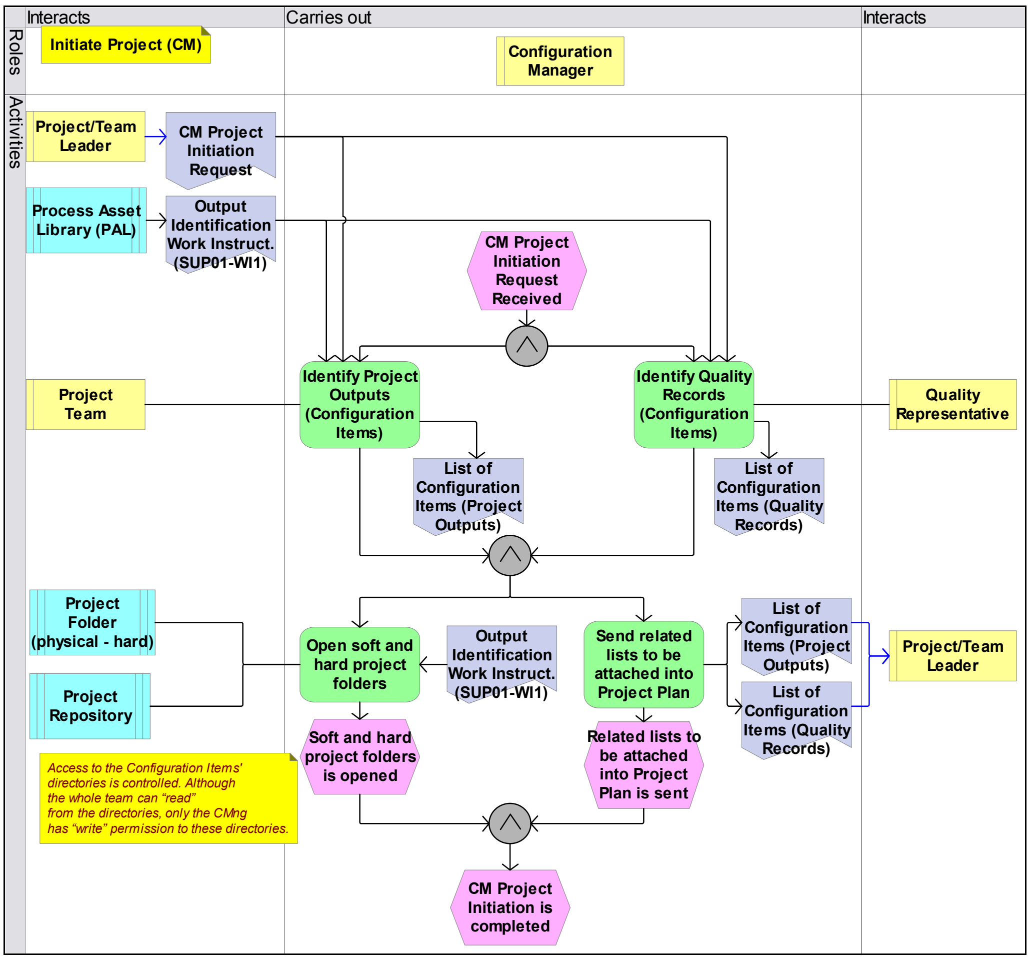

Interacts
Carries out
Interacts
Roles
Initiate Project (CM)
Configuration Manager
Activities
Project/Team Leader
CM Project Initiation Request
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request Received
Project Team
Identify Project Outputs (Configuration Items)
Identify Quality Records (Configuration Items)
Quality Representative
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
Project Folder (physical - hard)
Project Repository
Open soft and hard project folders
Output Identification Work Instruct. (SUP01-WI1)
Send related lists to be attached into Project Plan
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
Project/Team Leader
Soft and hard project folders is opened
Related lists to be attached into Project Plan is sent
Access to the Configuration Items' directories is controlled. Although the whole team can "read" from the directories, only the CMng has "write" permission to these directories.
CM Project Initiation is completed

Figure 74. An individual role-process diagram (Role: Configuration Manager, Process: Manage Configuration, Operation: Initiate Project (CM))

Agents used the add-on tool (described in section 3.7.3) to analyze the expectations and possible inconsistencies during process description. For most of the cases, agents, before they start modeling their operations, checked expectations of others', if any, and defined their activities based on these expectations. This ensured that their models satisfy other's expectations and avoided inconsistencies at the very beginning. If an expectation was not acknowledged by the agent, he/she started a discussion session with the expectant about the underlying rationale for the expectation and why he/she should not fulfill. The coordinator facilitated and moderated such interactions until all inconsistencies were resolved. Resolution was under agents' responsibility. Automated conflict resolution was not intended in these cases. We did not encounter any conflict that could not been resolved by participating agents and required a mediator to decide on a final judgment. Significant conflicts that were debated by all agents occurred for two of the cases which were both related to the review process:

- First conflict was related to the part the project/team leader role plays in the review process. The author expected that (the agent who was playing the author role expected that) once an item to be reviewed is complete, author sends it to the project/team leader who would initiate the review process. Project/team leader agreed that he receives the item to be reviewed from author but not for review initiation but for monitoring the progress. He expected the author to initiate the review.

- Second conflict was about author's attendance to review meeting. The author expected to attend the review meeting to justify any case and understand reviewers' comments. However, review team leader and review team members refused this expectation and insisted that the review meeting should be performed internal to the review team.

For both cases, related agents communicated and discussed the argument and raised the issue to other agents in the organization. To justify their cases, they explored sources (books, technical reports, and documents on best practices, and etc.) and shared their findings with others.

For some of the inconsistencies caused by typos or different naming practices, agents simply updated their definitions. This was also the case for the items that are overlooked by agents and identified in inconsistency analysis.

A notable observation was on agents' enthusiasm about checking their definitions against other's expectations, analyzing inconsistencies, and rapidly acting for its solution. As noted by one of the agents, they considered consistency checking as a compilation of their definitions. They also avoided being the one that was responsible for any inconsistency in the process-base. Achievement of such a motivation was important since the success of the method was highly dependent on active involvement of the participants in all phases.

Once individual diagrams are declared complete by the development agents, they were first validated by the peer agent and then subjected to verification by the coordinator. Both coordinators and peers were present during model development and acted on-site for any issue that might result an incorrect or inconsistent (with respect to notation's rules and semantics) definitions. This not only reduced the effort required for verification and validation but also decreased rework.

a. Phases III - Integration and Change

Once individual models were verified and validated, based on these models several diagrams visualizing the process-base from different perspectives were generated. As agents only worked on the parts of the processes that are performed by them, they required a representation that covers all activities of each of the processes. Thus, the coordinator first integrated individual role-process diagrams into activity level process diagrams to depict the activities within the process at the lowest level of detail and visualize the process as a whole. Figure 75 gives a part of the generated diagram for the manage configuration process.

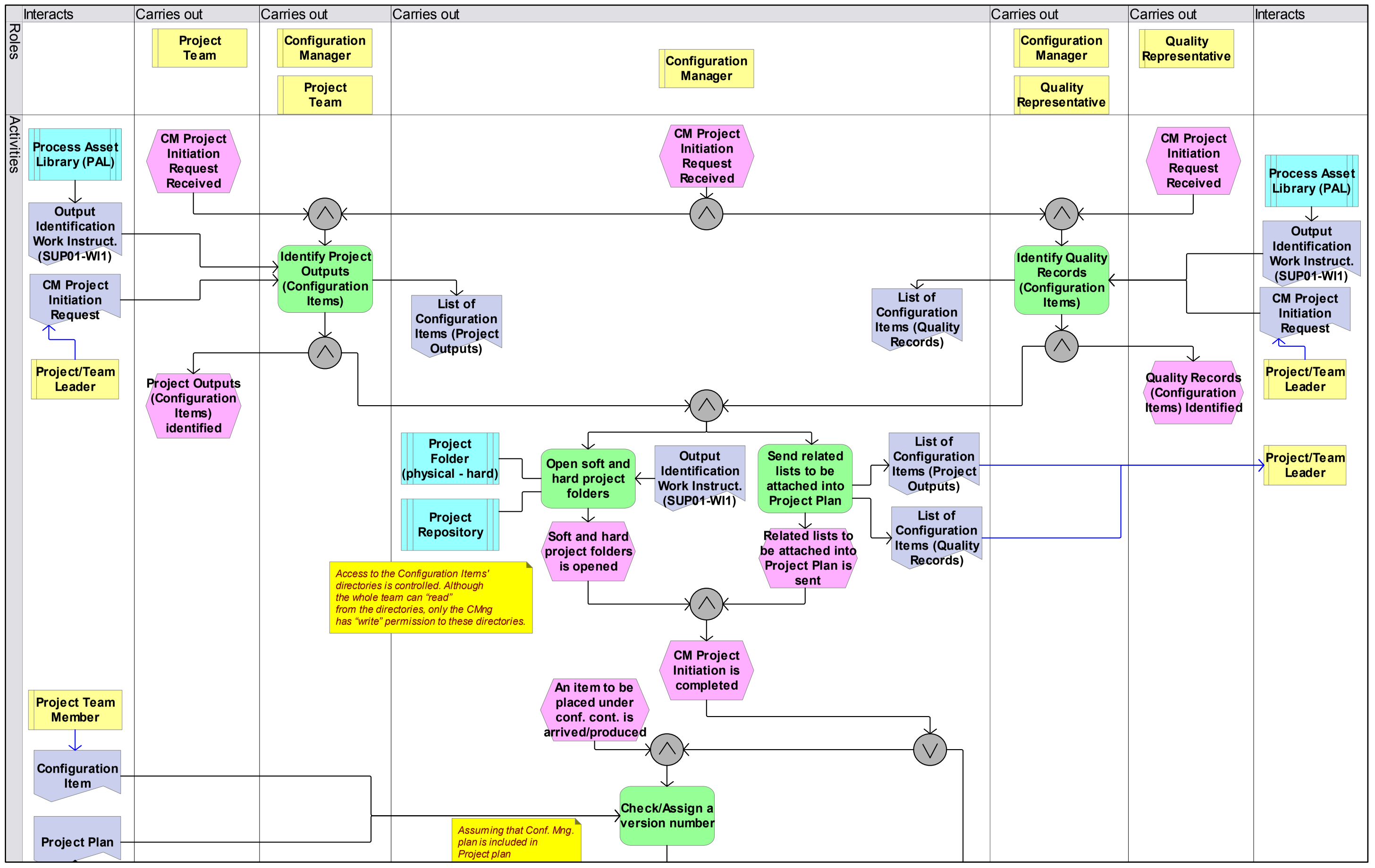

Interacts
Carries out
Carries out
Carries out
Carries out
Carries out
Interacts
Roles
Activities
Project Team
Configuration Manager
Project Team
Configuration Manager
Configuration Manager
Quality Representative
Quality Representative
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request
Project/Team Leader
CM Project Initiation Request Received
Identify Project Outputs (Configuration Items)
List of Configuration Items (Project Outputs)
Project Outputs (Configuration Items) identified
CM Project Initiation Request Received
Project Folder (physical - hard)
Project Repository
Open soft and hard project folders
Output Identification Work Instruct. (SUP01-WI1)
Send related lists to be attached into Project Plan
Soft and hard project folders is opened
Related lists to be attached into Project Plan is sent
Access to the Configuration Items' directories is controlled. Although the whole team can "read" from the directories, only the CMng has "write" permission to these directories.
CM Project Initiation is completed
An item to be placed under conf. cont. is arrived/produced
Check/Assign a version number
Assuming that Conf. Mng. plan is included in Project plan
List of Configuration Items (Quality Records)
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
Identify Quality Records (Configuration Items)
CM Project Initiation Request Received
Quality Records (Configuration Items) Identified
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request
Project/Team Leader
Project/Team Leader
Project Team Member
Configuration Item
Project Plan

Figure 75. The partial activity level process diagram for Manage Configuration

The add-on tool we used had features to perform that integration automatically; however, the integrated model needed manual layout to present the diagram in a clearer way to the user.

In interviews, these diagrams were found useful for the full representation of the processes. However, agents also considered some of the integrated models to be too complex to follow and comprehend and requested models that present the process information in an aggregated way. Operation-level process diagrams served this purpose. They represent the process with role's operations and information exchange between these operations. Figure 76 gives an example of an operation-level process diagram for the configuration management process.

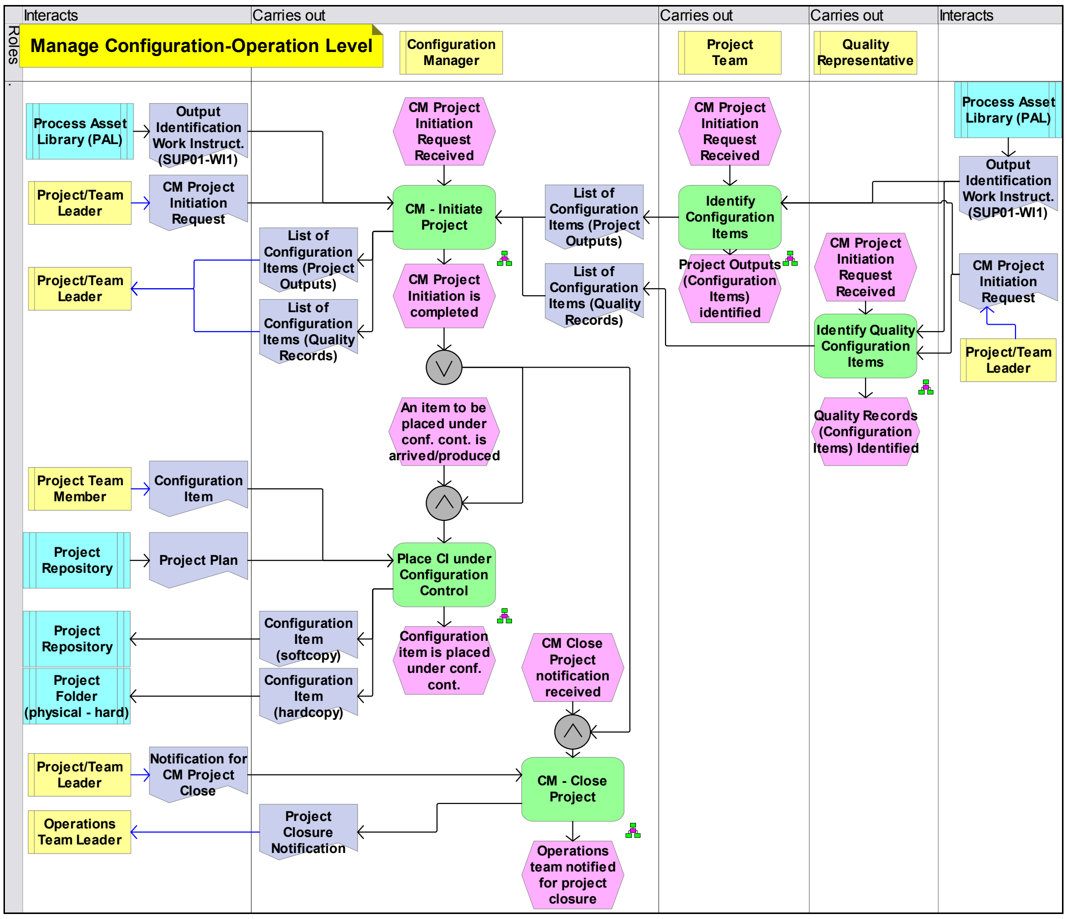

Interacts
Carries out
Carries out
Carries out
Interacts
Roles
Manage Configuration-Operation Level
Configuration Manager
Project Team
Quality Representative
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
Project/Team Leader
CM Project Initiation Request
CM Project Initiation Request Received
CM - Initiate Project
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
CM Project Initiation is completed
Identify Configuration Items
Project Outputs (Configuration Items) identified
Identify Quality Configuration Items
Quality Records (Configuration Items) Identified
An item to be placed under conf. cont. is arrived/produced
Project Team Member
Configuration Item
Project Repository
Project Plan
Place CI under Configuration Control
Configuration Item (softcopy)
Configuration Item (hardcopy)
Project Folder (physical - hard)
Configuration item is placed under conf. cont.
CM Close Project notification received
Notification for CM Project Close
CM - Close Project
Operations Team Leader
Project Closure Notification
Operations team notified for project closure

Figure 76. The operation level process diagram for Manage Configuration

In order to visualize the process and its interface and aggregate process information to the highest level, we generated process-level process diagrams. As noted in the first case study, these diagrams visualized the process and its overall interface to the outer world. They revealed key relationships between the process and other entities and helped agents to examine any implication of changing them. This also enabled agents to spot any interacting parties or information items that could have been neglected. Similar to the first case study, these overlooked items were already caught on individual diagrams during process descriptions before these aggregate diagrams were generated. A generated diagram for the configuration management process is given in Figure 77.

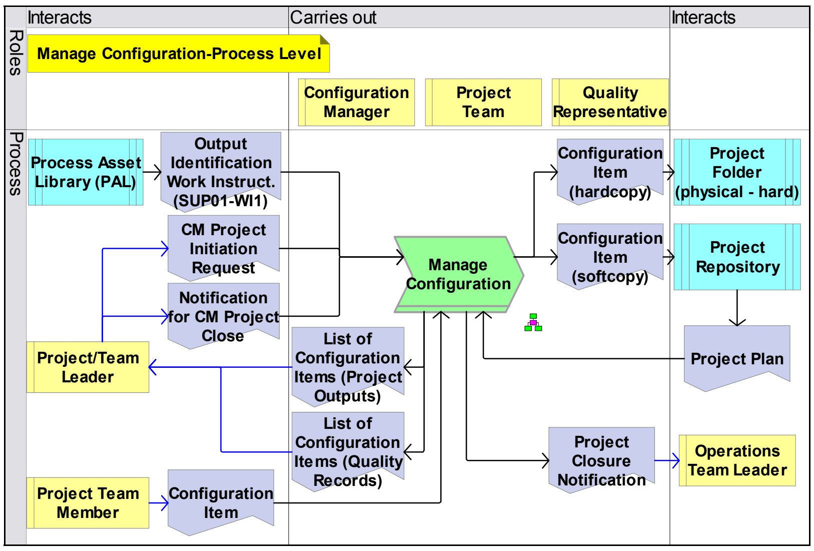

Interacts
Carries out
Interacts
Roles
Manage Configuration-Process Level
Configuration Manager
Project Team
Quality Representative
Process
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request
Notification for CM Project Close
Project/Team Leader
Project Team Member
Configuration Item
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
Manage Configuration
Configuration Item (hardcopy)
Configuration Item (softcopy)
Project Folder (physical - hard)
Project Repository
Project Plan
Project Closure Notification
Operations Team Leader

Figure 77. The process level process diagram for Manage Configuration process

Other types of diagrams were also generated to depict process dependencies as well as role dependencies. Figure 78 and Figure 79 gives two examples of the generated role and process dependency diagrams. Figure 125 in Appendix E gives the process dependency diagram for all processes covered in the case study. These diagrams aggregated the information into more focused and filtered forms and helped agents to envisage and understand the processes from multiple perspectives, in particular the exchange of information through the processes and roles.

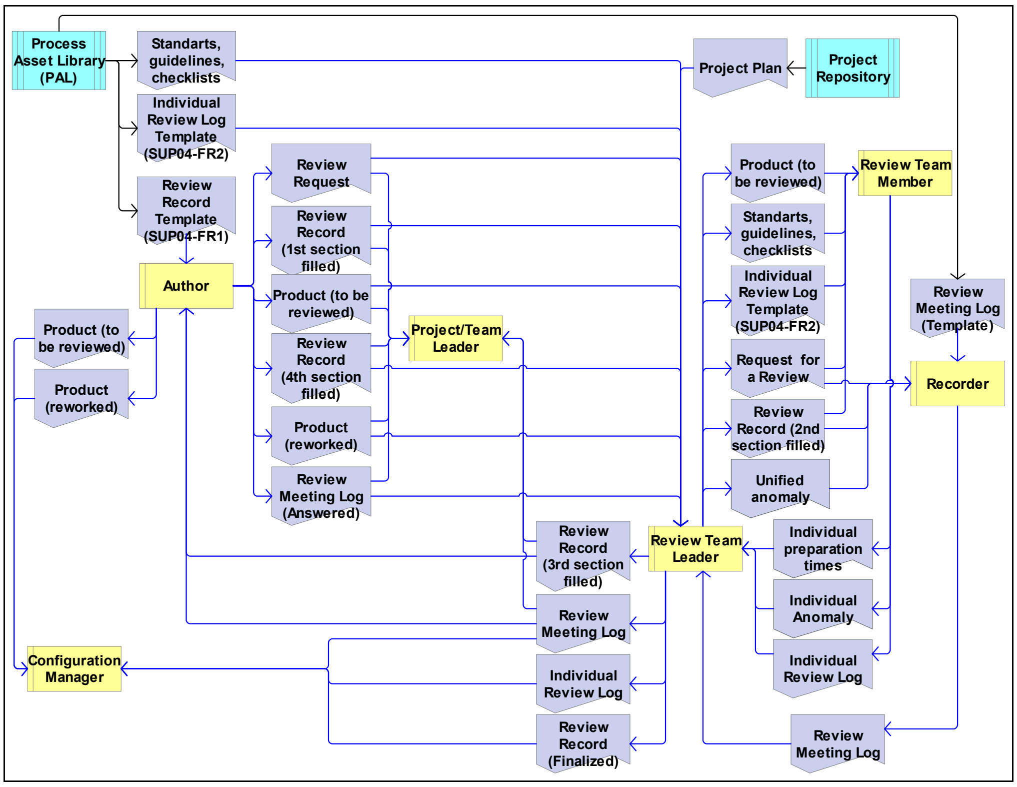

Process Asset Library (PAL)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Review Record Template (SUP04-FR1)
Author
Product (to be reviewed)
Product (reworked)
Configuration Manager
Review Request
Review Record (1st section filled)
Product (to be reviewed)
Review Record (4th section filled)
Product (reworked)
Review Meeting Log (Answered)
Project/Team Leader
Review Record (3rd section filled)
Review Meeting Log
Individual Review Log
Review Record (Finalized)
Review Team Leader
Project Plan
Project Repository
Product (to be reviewed)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Request for a Review
Review Record (2nd section filled)
Unified anomaly
Review Team Member
Review Meeting Log (Template)
Recorder
Individual preparation times
Individual Anomaly
Individual Review Log
Review Meeting Log

Figure 78. The role dependency diagram for the review process

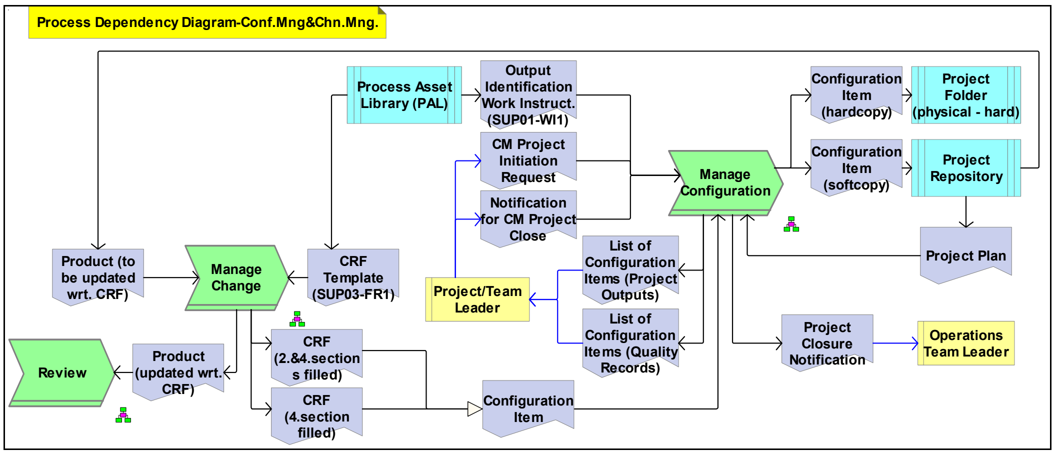

Process Dependency Diagram-Conf.Mng&Chn.Mng.
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request
Notification for CM Project Close
Manage Configuration
Configuration Item (hardcopy)
Project Folder (physical - hard)
Configuration Item (softcopy)
Project Repository
Project Plan
Product (to be updated wrt. CRF)
Manage Change
CRF Template (SUP03-FR1)
Project/Team Leader
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
Project Closure Notification
Operations Team Leader
Review
Product (updated wrt. CRF)
CRF (2.&4.sections filled)
CRF (4.section filled)
Configuration Item

Figure 79. The process dependency diagram for configuration and change management processes

The application of the method was suspended in the integration and change phase, and the first modeling cycle was considered complete, once all agents believed that the process models are complete, consistent and stable. In a short meeting with all participants, the completion of the first cycle was announced to the organization and participants' experiences and comments are shared. At this point, we interviewed with participants and asked them to fill a questionnaire related to their observations on the application of the method, and its implications. During interviews, we tried to gain further insight into their answers with open-ended questions and sought for their underlying rationale. We also asked further questions related to the notation and toolset utilized, and the way they consider the method helped them to express their knowledge about their processes. The questionnaire and participants' answers are given in Appendix F.

As in the first case study, the individual case study report for the seconds study is documented as a technical report [141]. The diagrams defined and generated throughout the case study are given in this technical report. Appendix E provides a list of these diagrams and a subset of the diagrams that was developed.

4.4. Findings and Discussions

In the previous sections, we described practices and instances we observed during the conduct of the case studies. The first case study also revealed some of the deficiencies of the method and the way it was enhanced for future applications (introduction of the 'operations' concept as an abstraction between processes and activities, need for peer agents, and etc.). This section discusses our cross-case findings based on our observations, interviews and analysis, and how they can relate to each of the propositions we maintained during the study.

In the first case study, we had a chance to observe the application of the method on processes which are more independent of each other, mechanic and involve less knowledge work. Due to these characteristics, the execution of the first case was easier and went more smoothly. In the second case study, the processes were more interrelated and involved mostly knowledge work. There was a significant degree of inheritance between roles as well as information items. Each knowledge worker had its own goals in performing certain activities and, when necessary, requested services provided by others. Thus, the agents in the second case were acting as goal-achieving autonomous entities unlike the agents in the first case, who were surrounded and constrained by legislative procedures. Although, the second case was conceptually harder to conduct, we believe that it revealed more benefits and advantages to the organization.

In next subsections, we summarize our findings regarding to our propositions.

4.4.1. Decrease in the total time required for process modeling

Table 20 summarizes the extent and the effort utilized in the case studies. First case study covered 12 high level processes which required 128 person-hours of effort to model with the Plural method. The context definition phase comprised meetings and work by the coordinators for modeling and documentation, which totaled to 18 person-hours. 90 person-hours of effort are utilized by agents for process descriptions, conflict resolutions and model verifications. Each agent defined different portion of the processes, thus the effort required to model their sections was unequal. Agent 3's section covered a larger scope which, in turn, resulted the highest value of effort committed to definition (as given in Table 20). The support by the toolset for the integration and model generation was limited. Hence, the last phase required 20 person-hours by the coordinators to integrate individual role process diagrams and generate others.

The second case study took 40 person-hours of effort. The context definition phase comprised two meetings and work by the coordinator for modeling and documentation, which totaled to 10 person-hours. 25 person-hours of effort are utilized for process definition and conflict resolution by the agents. As given in Table 20, Agent 1 committed the highest value of effort to definition. With the partial support of the toolset, the integration took considerable amount of time, which can be significantly reduced as it can be automated to a large extent. The integration phase required 5 person-hours by the coordinator to integrate individual role process diagrams and generate others.

Table 20. The extent of the Case Studies

# of high level processes	12	5
# of development agents	6	4
# of roles	30 (13 active / 17 inactive)	18 (15 active / 3 inactive)
# of distinct role operations	N/A	48
# of individual role-process diagrams	55	68
# of atomic activities	288	178
Effort (person-hour)		
Context Definition	18	10
Definition and Conflict Resolution	90	25
Agent 1	*9.0*	*9.0*
Agent 2	*5.0*	*5.0*
Agent 3	*20.5*	*2.5*
Agent 4	*13.0*	*2.5*
Agent 5	*11.5*	-
Agent 6	*17.0*	-
Coordinator	*12.0*	*6.0*
Coordinator 2	*2.0*	-
Integration	20	5
Total	**128**	**40**
Productivity (# of atomic activities/total effort**)**	**2.25** (288/128)	**4.45** (178/40)
Duration (hour)		
Context Definition	6	4
Definition and Conflict Resolution	20.5	9
Integration	14	5
Total	**40.5**	**18**

Based on the settings and effort utilized, for the first case study it took 6 hours to complete the context definition phase, 20.5 hours for the description and conflict resolution phase and 14 hours for the integration, which totaled to 40.5 hours for the entire project to complete its first cycle (the modeling cycle is complete when the process models are considered complete, consistent and stable). Total duration for the second case was 18 hours; of which 4 hours were spent for context definition phase, 9 hours for the description and conflict resolution and 5 hours for the integration and change phase.

Due to the organizational settings and certain constraints, the individual modeling in the description and conflict resolution was partially performed asynchronously in both cases. That is, for some of the processes, development agents performed individual modeling at different times when they were available. However, as we discussed in section 4.1 in case study questions and propositions, based on the processes that are modeled concurrently by all agents, we assumed that the parts performed asynchronously could have been performed synchronously (concurrently) by all agents without having a significant degree of increase (if it does not decrease) in the effort committed by any agent. Thus, the total time for the description and conflict resolution phase was considered to be the time devoted by an agent that utilized the highest amount of effort for that phase. In the first case, it was Agents 3 and in the second case it was Agent 1 that utilized the highest amount of effort (20.5 and 9 hours respectively) for that phase which became the duration committed for the overall phase. If modeling had been performed by a central group which would have worked with these individuals sequentially, the total duration would have been 96 hours for the first case which is significantly more than 40.5 hours. For the second case it would have been 28 hours instead of 18 hours. This is certainly true with a further assumption that the productivity of the group would remain (if not decrease) at equal or near values. Figure 80 depicts this scenario for the two cases. Assuming a central team working with the individuals one-by-one, the total duration for the description and conflict resolution phase would have been significantly more. Therefore, as we achieved concurrency in process modeling, we achieved a decrease in the total time required to model the organization's processes. In fact, the method enabled the time required to model the organizations' processes to be measured in hours instead of weeks or months.

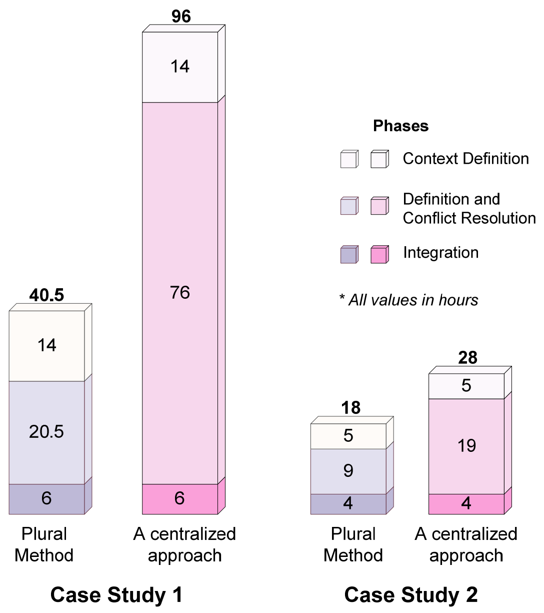


Figure 80. The duration with the Plural method vs. a centralized approach

We also noticed a decrease not only for the duration but also for the modeling effort utilized by the participants. It is difficult (if possible) to set up a laboratory setting to experiment and compare values for the total effort required for the projects and the efficiency of the modeling practice with the Plural method against other conventional methods. However, to have an indication of the achievement, we compared the values realized in similar projects ([42], [43], [44]) as described in the following paragraphs.

The projects were performed in the military organizations, with three of the individuals who also participated in our case studies; and utilized the same toolset and a similar notation. The method utilized a conventional approach where a central team of people worked with individuals that are actually performing activities one by one and developed organization's process models. Details of the projects are given in ([43] and [44]).

As a metric for the average productivity of the individuals participated in the process modeling activity in all projects, we calculated the total effort put forth by these individuals divided by the total number of atomic activities (lowest-level activities in a process). The average value for these projects came out to be 1.48 activities per person-hour (with the number of distinct project modules equal to 7 and standard deviation being 0.36) [44]. In addition, the total effort value does not include the effort utilized by the individuals that participated in the project from the user side (customer), which might significantly reduce this value. For our case studies, the productivity based on the same values turned out to be 2.25 (activity/person-hour) for the first and 4.45 for the second case study (higher value for the second case study may be attributed to the relatively high degree of experience of its participants). Particularly the value for the second case is significantly higher than the average productivity value realized in the previous projects.

We attribute this difference mainly due to the fact that the central group in these previous projects needed a considerable amount of time to work with individuals from the field to understand and analyze the way they perform their work and represent this information onto the process models. Considerable amount of effort was also committed for the validation of the defined models by the users and changes due to misunderstandings or parts that are overlooked [43].

4.4.2. Facilitating model completeness

As we discussed in section 4.1, we asserted that the Plural method facilitates model completeness by taking into account the perspectives of all involving parties. In order to support this claim, we first analyzed the current process descriptions and noted the differences between them and the developed ones. Secondly, we analyzed our observations noted during the conduct and the results of the interviews to examine how the definition is evolved and how explicit declaration of expectations contributed to model completeness.

During process description, development agents started modeling the 'as-is' processes, i.e., the activities they were currently doing at that time. They also referred to written process definitions. However, as they proceeded to inconsistency analysis and resolution, they started identifying the problems related with the current execution and the existing process definitions. Explicit representation of unfulfilled expectations clearly represented which key dependencies and individual and process-level goals are not being achieved satisfactorily. This provided information that aids efforts to effectively analyze problems and generate new solutions. Development agents identified several problems particularly related with incompleteness and ambiguities in procedures as well as implicit assumptions they hold during executions. To recall an example, for the review process, the author, project leader and review team leader all had different expectations and understandings related to the review initiation activity. Existing process descriptions also lacked the necessary level of detail that would actually guide the execution. The author expected to send the product to be reviewed to the project leader whom he thought would initiate the review. Whereas, project leader expected the author to initiate the review by sending related product to him and to the review team leader. Review team leader, on the other hand, just wanted to receive the product to be reviewed from someone as the initiation, but attached with the product; he also expected the review record with related sections filled.

The real execution was performed in one way or another since they had that tacit knowledge to handle any ambiguity or fill any gap between the execution and the definition. Together with the modeling study, they reflected on how they should actually perform their responsibilities. As they were modeling and observing the inconsistencies to handle them, they started solving many of the ambiguity and incompleteness problems of the current execution and the definition. They started adapting their processes with respect to other's situations and expectations. All that turned the definitions into a 'to-be' model of the organization.

4.4.3. Facilitating process change

The individuals that perform process modeling are apparent in Plural so as the ones that perform process change. The models that are maintained throughout the life of the organization are (1) context models (scope and role diagrams) that are owned by all individuals collectively and maintained by the coordinators; and (2) individual role process models that are developed and maintained by the individuals that are enacting these models. Thus, a change is either a micro level affecting the individual diagrams or a macro level that affects the structure – the context diagrams- that should first be approved by all individuals in relation to the processes that are affected and applied by the coordinators on these models.

For some cases, the development agents realized that, the execution was no longer adhering to the definition, which is in essence, a common problem in the organizations. For a reason, they changed the way they perform the process but found it difficult and time consuming to change the definition. Besides, there was generally an ambiguity for the role that will be responsible to perform that change. In the software organization, changes to the process definitions and related artifacts were performed by a specific agent and managed via the change management process. However, that current structure of the process, as agents noted in interviews, put a degree of bureaucracy on implementing the change itself. A necessity for change or an improvement opportunity was identified by the agents themselves but they somehow hesitated to initiate the change process, in which, they were expected to fill a change request form and hand it to the process engineer, wait for an approval and if approved observe it as reflected. Instead, they simply started altering their execution and began departing from the definition. For example, the review record that some of the agents were using incorporated useful information about metrics to be collected, which was absent in the standard form in their process assets library. So, this improvement chance was hindered or postponed.

The study gave them the responsibility to reflect any change they consider necessary not only on the way they perform their tasks but also on the artifacts they own. Being the supplier of these artifacts, they were responsible to maintain them and communicate any change with the customers of these artifacts. This might also involve negotiations with them on the content and structure. The agent playing the review team leader role restructured the review record mentioned above; reflected necessary changes on it, highlighted the portions each role is expected to fill in and communicated these changes to related agents.

With the approach adapted, the responsibilities were inherently more lucid. Changes to the scope including the changes on role definitions, were discussed and approved by the study group including all members and reflected on the models by the coordinator. Changes related to individual role definitions, on the other hand, were directly performed by the related agent that were playing that role and reviewed by its peer agent. If a change did not affect role's interface, then it was a simple alteration in role's context. For example, configuration manager's alteration in the operation of 'placing under configuration control' did not affect the way other roles perform their processes. Because the change does not have an impact on other role's expectations, but only affected the way the configuration manager performs its activities and produce its artifacts. However, if an update modified its interface with the neighbor roles, then that change was incorporated in related models or it was revoked after a negotiation between parties. Review team member, for instance, wanted all review materials (standards, checklists, and etc.) to be delivered to him by the review team leader, whereas, at that time the review team leader handed only the product to be reviewed. These cases manifested themselves as inconsistencies between expectations which should be resolved in related models. For the above case, after a negotiation, parties agreed on a resolution where the team leader updated its model. Hence, the inconsistency was eliminated and team member's expectation was fulfilled.

An informative metric for our case studies would be the number of omitted or overlooked items. However, any number for such type of inconsistencies would be misleading, since for most of the cases, agents started modeling their activities based on others' expectations that were already defined. Thus, when they started modeling their activities they immediately incorporated such information into their models, as if that information would in any case be incorporated into their models.

4.4.4. Facilitating the discovery of interaction points, expectations and conflicts

The Plural method is based on agents' revealing their expectations and model their own activities based on the expectations of others; if required, negotiating to ensure that expectations are fulfilled. Thus, the method ensures that each role's interaction points are visible to all organization.

The interaction points are also explicitly represented in generated role-dependency and process-dependency diagrams. These diagrams aggregated the information into more focused and filtered forms and helped agents to envisage and understand the processes from multiple perspectives, in particular the exchange of information through the processes and roles. First versions of the generated diagrams also helped agents to identify missing or incomplete parts in individual descriptions. In these cases, these changes are feed back to prior phase where individual diagrams are updated by related agents. For example, in the first case study, one of these changes in research assistant requisition process resulted an inconsistency between the individual descriptions of the roles that take part in that process. Specifically, in the research assistant requisition process research assistant and secretary role overlooked an information item that should have been sent from secretary to the accepted assistant (last day to apply for the position). The first agent that realized this omission was the research assistant that changed his definition and caused an inconsistency in the process-base. So, in order to resolve the inconsistency, secretary needed to change her definition in order to fulfill the new expectation and adapt to the new situation.

The examples we discussed in above paragraphs indicated cases where agents surfaced their assumptions, unveiled their expectations and shared them with the organization. They had chances to identify and resolve controversies among each other on the way the organization is working.

We observed that participants adapted to the method more rapidly than expected. In both case studies, the planned effort was more than the actual values (planned 60.5 hrs vs. actual 40 hrs). The participants in the second case were given a questionnaire and interviewed to elicit their attitude to the way the method executes; its benefits as well as limitations. The questionnaire and the results are given in Appendix F. According to the questionnaire and interviews, they did not encounter any significant difficulties in following the method and found the notation relatively easy to learn and use. They also found it useful to isolate their roles and responsibilities clearly from others and maintain them separately.

Agents strongly agreed that modeling gave them a better understanding of the processes they perform and explicit modeling of their interface between other roles provided useful and important information about the process otherwise would have been omitted. They all agree that having the responsibility to model the processes they participate prompted them to think of these processes more thoroughly and in detail. Despite agents commented that it required additional effort; they found it beneficial and essential to explicitly model the transfer of information items between roles with respect to value it served for models' completeness and consistency. Likewise, agents noted that models - in particular the integrated activity level process diagrams - were generally complex and hard to follow, but were accurate representations of the processes and represented essential information about them. Particularly for process guidance, role-based modeling was regarded as valuable and helpful since it clearly represented the responsibilities and the interface for each role. This would also help the organization to be ready for outsourcing roles more easily, since it is generally the roles that are outsourced for a specific process.

4.4.5. Generalizing process components

During process descriptions, we observed that it is a good practice for the coordinator agents to be monitoring the development not only for verification purposes but also for observing any reusable process components that can be generalized. For example, in the software organization, each agent had a particular structure of initiating a change request in the organization. For this case, the coordinator agent, based on the definitions of the development agents, proposed a generalized change request (CR) initiation procedure to be performed by a particular role; the CR originator. The study group accepted this modification and each role initiating a change request inherited this responsibility from the CR originator role. This example also points to a dilemma on whether to let each agent to maintain its way of doing that set of activities or to establish a standard generalized way of reaching that process' goal. Based on the experience on the case studies, it is important that the Plural process group with all its members and other stakeholder would decide on this matter. For some cases, it would be beneficial to maintain separate ways of doing a particular work but for some other cases, like the one mentioned above, it might not worth keeping multiple descriptions and related agents might decide on a specific procedure such as a best practice to be maintained in a unified way.

4.4.6. Process goals and metrics

Role-based modeling by agents eased each individual to define role-based metrics and integrate the information into their process definitions. For the review process, for all participating roles, agents defined individual goals of the role for that process and the related metrics they would like to collect during the execution. They were able to define when and by which means the metrics will be collected and stored so that they could also track their individual performance. For example, review team leader stated that 'prepare review' operation supports the following goals; 'to provide reviewers necessary resources to perform individual checking' and 'to plan the review process'. During its execution, the role measures the 'size of the product to be reviewed' base metric on the review record (Figure 31, pg.70). Analysis of these role-based metrics would then highlight the overall process performance. Capturing each role's objectives for processes was important in understanding why the process operates as it does and what the key implications of a process change are. This may help organization to assess process goals and the goals of its participants and help them to understand process' complexity before altering key relationships during the process change [88], [159].

Most of the current process modeling approaches and languages aim to express what steps a process consists of or how they are to be performed. Yu et.al. [159] stress on understanding '*why'* of processes behind the 'whats' and 'hows' in order to improve a process. In general, process improvement approaches deal explicitly with the whys, which reflect the complex social relationships among process participants.

CHAPTER 5

CONCLUSIONS

Majority of the process modeling and improvement approaches assumes a centralized structure for modeling and improving organizations' processes. In general, a central group is responsible to work with process owners to define processes and maintain them and control the process modeling as a whole. However, this central framework poses difficulties particularly in knowledge based organizations where knowledge is highly distributed. Such a central unit limits the degree of the involvement of the process owners in the modeling practice, which is one of the most critical factors contributing to the success of such imperatives. This centralized structure makes it more difficult for process owners to own the definition and maintain it. This also poses difficulties for the improvement to be actually owned by these individuals. Besides it requires considerable amount of time and effort for this central team to work with these process owners one by one to understand and model their work and improve it.

A decentralized approach that empowers process owners to model their own processes concurrently can enable modeling to be performed more rapidly. Moreover, this grassroots approach to process modeling would let these knowledge workers to reveal their expectations, communicate and share their process knowledge, discover implicit assumptions and misunderstandings, and start 'thinking' about the way they perform their activities and the way it can be improved.

In order this approach to be applied in organizations and rip the benefits of decentralization and concurrency, we need an important enabler - a methodical way that would guide the organization. Our survey of the related literature revealed a lack of mechanisms and methods that would enable individuals to concurrently model their processes; help them to identify and resolve inconsistencies among individual definitions; enable them to easily integrate these partial models to form the organization's process network; and finally allow them to continuously maintain their own definitions.

This thesis has addressed a range of issues arising from decentralized and concurrent business process modeling. In particular, it has focused on a method that guides organizations to perform process modeling in that manner. The method is a novel contribution to the field of process modeling and improvement where centralized frameworks are prevailing.

This chapter examines the contributions in more detail, highlights the benefits and suggests future research directions based on the findings discovered during the thesis study.

5.1. Contributions

5.1.1. The Plural method and the case study findings

The major contribution of this research is the Plural method that can be utilized for decentralized and concurrent business process modeling. The objective is to provide agents a systematic way, and necessary mechanisms as well as infrastructure to define their processes and maintain these definitions separately and integrate these definitions to form organization's process network. Individual definitions form the organization's process-base which let the visualization of processes in different abstraction levels and from different perspectives. The method facilitates the empowerment of individuals. It encourages them to start thinking and improving their processes, to uncover implicit assumptions and identify problems and issues related with the current execution.

In order to explore the applicability of the Plural method for decentralized business process modeling, a multiple-case study was selected as a research strategy and two cases were conducted. First case study was conducted in a university department where we had a chance to observe the utility and drawbacks of the method and other enablers. Based on the improvement opportunities identified in the first case study, we enhanced the method as well as the notation and the toolset. In the second case, the method was utilized in a small size software engineering organization in order to examine its applicability and to observe whether projected benefits of the approach are realized or not. We discuss our findings based on the study questions and related propositions. The discussions on these findings are contributions of this thesis study.

The results discovered in the case studies and investigations during the research study revealed that the proposed method can be successfully utilized for decentralized business process modeling and despite its limitations, expected benefits can be achieved. Our findings based on the analysis of the case study results suggest early evidences that knowledge workers (consequently the organization) benefit from taking the responsibility of understanding, modeling and improving processes by themselves.

The method enables the organizations to capture the perspectives of multiple agents. These agents hold a partial knowledge of their organization's processes ([108], [56]), which mainly comprises the activities they perform and artifacts they consume/produce with respect to their expectations and objectives which differ for each stakeholder [95]. Representing all these viewpoints served models' completeness and ensured that explicitly defined expectations are shared among agents.

The interaction points between roles (consequently agents) are one of the most fragile points in processes performed by knowledge workers [22] and are potential locations in identifying implicit assumptions of participating agents. Based on our findings in our case studies, we infer that, the Plural method helps discovering these communication points, expectations as well as conflicts and makes them visible to all participating agents. For process improvement and redesign, it is important to make sure that participants have a roadmap of others' positions, which is critical to productive negotiation [88]. Shared perspectives of agents including its goals and interdependencies help organization to succeed in process definition and improvement.

Another advantage we observed in our case studies was the total time devoted for modeling organization's processes. In current approaches, generally there is a central group that performs modeling sequentially starting from a functional area. The group devotes significant time to understand and model current execution of processes [150]. In the approach where process performers can model their own processes in concurrent way, modeling effort is shared among these agents. Thus, the total time for process definition and conflict resolution becomes the time committed by a single agent that performed its portion the longest. As this agent completes its section, there is an opportunity for the others to have completed their sections already. The completion is the last synchronization point for all agents, where they check the consistency among all models for the last time and finish the definition altogether. Having benefitted from the concurrency in process modeling, our case studies completed in 40.5 and 18 hours for the first and second cases respectively, instead of 96 and 28 hours if concurrency have not been achieved.

Our finding in the case studies also revealed that, the method helps organizations to incorporate changes in processes more easily and rapidly compared to the methods where a central group develops and maintains the definitions. Because, similar to the process definition, maintenance is also performed by related agents concurrently and in a decentralized manner and the impact of any change related to the interaction of two or more roles is directly visible as it occurs. If a change alters one's expectations from others, the implications of that change are immediately visible to all participants. The impact is revealed as an inconsistency between models that should be handled by the interacting roles. Affected agents alter their definitions until all definitions reach to another consistent and stable state. This visible impact analysis and consistency preservation view helps organization to maintain its process-base in a more efficient and effective way.

Empowering knowledge workers to take responsibility to model and improve their processes is central in Plural. Based on our observations and interviews in case studies, we concurred that, role specific modeling increases personalization and in turn fosters a sense of ownership and empowerment that also motivates active participation. According to the researches on software engineering organizations, empowerment is one of the most important factors to get software engineers involved in process improvement or similar imperatives [9]. Paulk et.al. describe the formation of process ownership teams on a software development project at IBM-Houston, where the team comprised of "…the people who perform the processes and are therefore in the best position to know what improvements should be made." [120] Although this concept of process ownership to improve practitioner support and involvement in process improvement is appreciated in process improvement initiatives, in general organizations rarely had this idea pursued in their organizations [9]. We believe that the lack of mechanisms, appropriate process infrastructure and related enablers is one of the important reasons that forbid organizations to pursue such an approach.

Overall we can conclude that the study provided initial evidence of the approach's value and showed how an organization might exploit its strengths using the enablers proposed in the thesis. The method helped participants to clearly define their expectations and goals and negotiate with other agents to achieve them. It provided an environment and a mechanism to define their expectations, unveil and discuss problems and establish a common jargon between participants while letting them to represent and keep theirs. All these hands on experiences, in turn, facilitated communication and knowledge sharing between participants.

5.1.2. Other contributions to the field of business process modeling

Survey on related literature on process modeling is another contribution of this study. We reviewed approaches in diverse fields, including software engineering, business process management, business process re-engineering, enterprise modeling and agent based approaches; summarized their key properties and evaluated their applicability with respect to the unique requirements of decentralization and concurrency in process modeling. Our review of the related literature unveiled that the majority of the process modeling approaches utilize centralized view in process modeling. Not only the related methods but also the notations and tools utilized in process modeling inherit and maintain this centralized view [139]. Very few attempts trying to shift process modeling responsibility to all process owners suffer from the lack of a systematic method and other enablers. Our review on related work also illustrated that, although there are several notations and tools utilized for process modeling in diverse fields, there are very few methods or guidelines that clearly state how organization starts from the very beginning and proceeds to a state where it has a complete, consistent and validated process-base. In general, many of the current approaches focus on the underlying representational formalisms, their implementation and enactment on related tools. Plural provides a guideline ensuring that process modeling and maintenance can proceed in a decentralized manner.

As one of the key enablers of a decentralized business process modeling approach, we identified fundamental characteristics and requirements of a notation that supports the method. Based on the requirements, we developed a conceptual framework for a notation including a number of diagrams each capturing different aspects of the process. The Plural notation is another contribution of this study.

Contributions are also made by developing the techniques and rules to generate a set of diagrams representing organizational knowledge from diverse viewpoints. Proposed rules enable generation of higher level process diagrams, role and process dependencies, based on individual descriptions.

Another important enabler for the decentralized approach is a tool integrated with the notation that supports the modeling practice. As a contribution of the study, basic requirements for such a tool are elicited and a prototype is developed.

5.2. Limitations and Future Work

Our case studies also revealed some limitations and success factors of the method as well as the notation and the toolset utilized in the study. We discussed some of these limitations and the way they are mitigated in sections 4.2 and 4.3. This section provides a broader view into the method's and the study's limitations and the future work.

Modeling processes that are new to the organization

The case studies showed that the expected benefits from the approach are not fully realized if the processes being defined are not performed or not effectively established in the organization. For the change management process, for instance, there existed a definition but, the process itself was never enacted. Therefore, the agents had some difficulties and needed more guidance particularly about their responsibilities and their goals in performing that process. For this case, the coordinator and the study group provided additional information to set the basis for the new process by identifying primary inputs and outputs, and a brief description of what is expected from each participating role. In doing this, they utilized the existing definition and process standards to identify the backbone of the process. During new process definition, pair modeling -development agents modeling with peer agents and/or the coordinator- was beneficial.

As a future work, the method can be extended to cover processes that are new to the individuals and to the organization. It can provide direct guidance for the organization not only for modeling these processes but also helping them to effectively establish it in the organization.

Process change and process improvement

Although process improvement can be perceived in the context of process change, the current structure of the Plural method does not indicate or assume that process changes would lead an improvement in processes in the local (individual) or in the global (organizational) levels. That is, we believe that the method provides efficient and effective mechanisms to capture improvement opportunities at the bottom where it is initiated, but we have no data about the degree of these changes contribute to improvement of individual processes and concurrently organizational processes as a whole. Nonetheless, our claim is that the decentralization in modeling is one of the fundamental steps towards the decentralized process improvement and the method can be extended in the direction of including related practices and strategies to enable this to happen in that way.

Maturity of the organization

The organization in the second case study can be considered to have a relatively higher maturity in terms of several factors including its operational environment, its process stability as well as the way it considers process improvement. Having its quality system certified with ISO 9000 might also have contributed its process stability. The interviews and the questionnaire revealed that the organization pursue a culture that provides a degree of empowerment and motivation for individuals to continuously improve themselves. Modeling and improving processes considered as a part of their responsibilities rather than an additional burden loaded on daily work activities. We believe that, this eased the way the approach is adapted by the organization, since it fostered agent's motivation and commitment on to the study, which is utmost important for successful implementation of the approach in the organization.

Limitation of the tool utilized and its criticality

Tool support for the approach is important and there are significant works that can be done on that side. A tool that is specifically developed (or adapted) to answer the unique requirements of the method is necessary in order to rip the most benefit of applying the Plural method. This would increase the modeling efficiency and effectiveness, while decreasing the effort and ease the application of the method in organizations in different domains.

In this study we developed a set of basic requirements of a tool and prioritized these requirements as being mandatory or recommended for the application of the method. Mandatory requirements denoted the ones which are critical for the application of the method on real life case studies. We addressed these mandatory requirements with a commercial tool and with its extension that we developed. We enhanced the usability of the add-on tool based on the user feedbacks. However, the recommended functionalities were partially fulfilled or were not been fulfilled at all. Particularly, the current toolset lack efficient diagram generation features. Thus the toolset used for the study did not provide the desired degree of support to the method and the add-on had its own limitations. The real benefits would be gained with a tool answering the unique requirements of the approach.

Cognitive limitations of the diagrams

Besides managerial and organizational issues mentioned above, there were also cognitive limitations of the diagrams utilized for the approach. In essence, the trade-off between information and cognitive complexity is inherent in most visual representations. Additional information can be represented with the sacrifice in the ease of perception. For some of the processes, an individual role model for an operation (such as 'student admission' of the department secretary role or 'initiate project' of project/team leader role) was too large to fit into a regular sheet of paper which increased its complexity. The situation was worse for the some of the integrated and generated models. For activity-level process diagrams for example, models with higher abstraction levels like operational-level process models were generated to ease understanding. Similarly, a complete role dependency diagram for the organizations was too complex that needed other representational techniques. For those cases, separate diagrams depicting a portion of the data were generated such as the ones depicting role dependencies only for one specific role or a process. As a future work, new types of diagrams representing processes with different abstraction levels and perspectives can be included in the Plural notation. The method and the notation can also be extended to cover new types of diagrams that address different aspects of the process knowledge. For example, role-state diagrams -that represent the transitions a role undergoes throughout a particular process- can be included. New diagram generation techniques can be developed for these purposes.

Scale of the organization

The case studies were performed in relatively small size organizations (the second case study in particular) with a limited number of participants and scope. Therefore, we have currently no data if the approach will scale-up to be used for larger organizations with hundreds of knowledge workers and spanning a larger extent of processes across the organization. However, in case studies, we applied the method in processes with diverse complexities and sizes and there are indications that the scalability of the method is related more with the nature of the process that is being modeled - its complexity and stability, the number of roles and external parties that participates in the processes and the complexity of their interaction and collaboration, and etc. - than the number of the processes to be modeled and the number of knowledge workers that participates in these processes. Each process is handled with a set of participating agents and the whole process of modeling is monitored by a coordination team. In large organizations, there would be a need for more number of coordinators that should monitor the execution of the method, concurrently maintain the bird-view of the organization's processes and collaborate for capturing large scale improvement opportunities – changes related to process architecture such as generalized process components.

The case study methodology adapted for the research also poses some limitations for this study. The general validity of the conclusions was limited with the limited number of case studies performed. The representativeness of the organizations also poses limitations for the generalizability of the findings within a broader context.

In order to gain further evidence about the validity of the approach - its assumptions and assertions, we can go through additional case studies on diverse organizations and processes. These future case studies can also enable the Plural method to be refined and improved based on the results. Future case studies should also include larger organizations and environments that we mentioned above in order to gather more data on method's applicability and scalability.

The case organizations should also be monitored for longer periods in order to observe ongoing implications of the utilization of the method and validate that the benefits and improvements are substantial and continuous.

REFERENCES

1. Abdel-Hamid, T.K. & Madnick, S.E. (1989). Lessons learned from modeling the dynamics of software development. *Communications of the ACM, Vol.32 Iss.12*.

2. Acuna, S. T. & Ferre, X. (2001). Software Process Modelling. *Proc. Of World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001)*. Orlando (Florida), USA.

3. Alter, S. (1996). *Information Systems: A Management Perspective*. 2nd Ed. The Benjamin/Cummings Publishing Company, Inc.

4. Ambriola, V., Conradi, R. & Fuggetta, A. (1997). Assessing Process-Centered Software Engineering Environments. *ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 6, Iss.3*, 283 - 328.

5. Andersen, E.P. (1997). *Conceptual Modeling of Objects: A Role Modeling Approach*. PhD. Thesis. Department of Informatics, University of Oslo.

6. Argyris, C. (1994). Good Communication That Blocks Learning. *Harvard Business Review, July-August 1994*.

7. Armour, P.G. (2004). *The Laws of Software Process: A New Model for the Production and Management of Software.* Auerbach Publications.

8. ATHENA (Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application) Programme Team. (2004). *State of the Art in Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability*. Ver.1.2. Project Name: Enterprise Modelling in the Context of Collaborative Enterprises. WP.A1.1.

9. Baddoo, N., Hall, T. & Wilson, D. (2000). Implementing A People Focused SPI Programme. In Maxwell, K., Kusters, R., van Veenendaal, E. and Cowderoy, A. (eds): *Project Control: The Human Factor. Proceedings Of The 11th European Software Control And Metrics Conference*, (pp. 373-381). Munich, Germany.

10. Baddoo, N. & Hall, T. (2002). Practitioner Roles in Software Process Improvement: An Analysis using Grid Technique. *Software Process Improvement and Practice. Vol.7.* 17-31.

11. Bandinelli, S., Fuggetta, A., Ghezzi, C. & Lavazza, L. (1994). SPADE: An Environment for Software Process Analysis, Design, and Enactment. In A.Finkelstein, J.Kramer, and B.Nuseibeh (Ed.), *Software Process Modelling and Technology* (pp.223–247). Research Studies Press Ltd., Taunton, Somerset, U.K.

12. Bandinelli, S., Braga, M., Fuggetta, A. & Lavazza, L. (1994). The architecture of the SPADE-1 Process-Centered SEE". B.C. Warboys (Ed.), *Proceeding of the Third European Workshop on Software Process Technology (EWSPT'94)*, Villard-De-Lans, France, Springer-Verlag.

13. Barnett, G. (2006). *Four BPM Lessons Learned*. BPMInstitute.org. Web article last viewed in 14.July.2006 <http://www.bpminstitute.org/articles/article/article/four-bpm-lessons-learned.html>

14. BEA Systems, IBM, Microsoft, SAP AG, Siebel Systems (2003). *Business Process Execution Language for Web Services*. Ver.1.1.

15. Becker, U., Hamann, D., Münch, J. & Verlage, M., (1997). MVP-E: A Process Modeling Environment. In Khalet El Emam (Ed.), *IEEE TCSE Software Process Newsletter, Vol. 10*.

16. Becker, U., Hamann, D. & Verlage, M., (1997). Descriptive Modeling of Software Processes. *In Proceedings of the Third Conference on Software Process Improvement*. Barcelona, Spain.

17. Belkhatir, N., Estublier, J. & Melo, W. (1991). Adele 2 - A Support to Large Software Development Process. *In Proceedings of First International Conference on the Software Process*. 159 - 170.

18. Belkhatir, N., Estublier, J. & Melo, W. (1993). Software process model and work space control in the Adele system. In L.Osterweil (Ed.), *Proceedings of the 2nd International Conference on the Software Process*, (pp. 2-11). IEEE Computer Society Press, Los Alamitos, Ca.

19. Ben-Shaul, Israel Z. (1993). *Oz: A decentralized process centered environment*. (Technical Report CUCS-011-93). Columbia University, Department of Computer Science.

20. Ben-Shaul, Israel Z. & Kaiser, G.E. (1994). A paradigm for decentralized process modeling and its realization in the Oz environment. *In Proceedings of 16th International Conference on Software Engineering* (pp. 179–190). Sorrento, Italia.

21. Ben-Shaul, Israel Z. & Kaiser, G.E. (1995). *An Interoperability Model for Process-Centered Software Engineering Environments and its Implementation in Oz*. (Technical Report CUCS-03495). Columbia University, Department of Computer Science.

22. Brady, S. & DeMarco, T. (1994). Management-aided software engineering. IEEE Software. Vol. 11. 6-25

23. Bröckers, A., Lott, C., Rombach, H.D., Verlage, M. (1995). *MVP-L Language* (Report Version 2, Internal Report Nr. 265/95), Department of Computer Science, University of Kaiserslautern, Germany.

24. Bruynooghe, R.F., Greenwood, R.M., Robertson, I., Sa, J., Snowdon, R.A. & Warboys, B.C. (1994). PADM: Towards a Total Process Modelling System. In A. Finkelstein, J. Kramer, and B. Nuseibeh (Ed.). *Software Process Modelling and Technology*. (pp. 293-334). Research Studies Press Ltd., Taunton, Somerset, U.K.

25. Business Process Management Initiative-BPMI. (2004). *Business Process Modeling Notation (BPMN)*. Ver.1.0.

26. Business Process Management Initiative-BPMI. (2003). *Business Process Modeling Language (BPML)*.

27. Canals, G., Boudjlida, N., Derniame, J-C, Godart, C. & Lonchamp, J. (1994). ALF: A Framework for Building Process-Centred Software Engineering Environments. In A.Finkelstein, J.Kramer, and B.Nuseibeh (Ed.). *Software Process Modeling and Technology*. (pp. 153-185). Research Studies Press Ltd., Taunton, Somerset, U.K.

28. CIMOSA Association (1995). *CIMOSA - Open System Architecture for CIM, Technical Baseline*. Ver.3.2. Springer-Verlag.

29. Conradi, R., Fernström, C., Fuggetta, A., & Snowdon. R. (1992). Towards a Reference Framework for Process Concepts. J.-C. Derniame (Ed.), *In 2nd European Workshop on Software Process Technology (EWSPT'92)*, (pp. 3-17). Trondheim, Norway. Springer LNCS 635. Also As EPOS TR 158-1992.

30. Conradi, R., Jaccheri, M. L., et.al. (1994). EPOS: Object-Oriented and Cooperative Process Modeling". In A.Finkelstein, J.Kramer and B.A. Nuseibeh (Ed.). *Software Process Modelling and Technology*. (pp. 33-70). PROMOTER, Advanced Software Development Series, Research Studies Press Ltd. (John Wiley), Also as EPOS TR 198-1993, SU-report 6/94.

31. Conradi, R., Hagaseth, M. & Liu, C. (1994). Planning support for cooperating transactions in EPOS. In G. Wijers, S. Brinkkemper, T.Wasserman (Ed.), *Advanced Information Systems Engineering - 6th International Conference, CAiSE'94*, (pp. 2-13). Utrecht, The Netherlands. Also as EPOS TR 209. SU-report 11/94.

32. Conradi, R. & Liu, C. (1995). Process Modelling Languages: One or Many?. *Proceedings of the Fourth European Workshop on Software Process Technology*, Noordwijkerhout, The Netherlands. Springer LNCS 913.

33. Conradi, R. (1997). Process Modelling Languages: Needs, coverage, alternatives, evaluation. *PROMOTER/RENOIR Summer School on Software Process Technology*. Grenoble, France. 1997.

34. Cugola, G. & Ghezzi, C. (1998). Software Processes: a Retrospective and a Path to the Future. *Software Process - Improvement and Practice, 4(2)*. 101-123.

35. Cugola, G., Di Nitto, E. & Fuggetta, A. (1998). Exploiting an event-based infrastructure to develop complex distributed systems. *Proceedings of the 20th International Conference on Software Engineering, (ICSE98)*, Kyoto, Japan.

36. Curtis, B., Kellner, M.I. & Over J. (1992). Process Modeling. *Communications of the ACM, Vol. 35, No.9*.

37. Dami, S., Estublier, J. & Amiour, M. (1998). APEL: a Graphical Yet Executable Formalism for Process Modeling. *Automated Software Engineering.* 60-96. Kluwer Academic Publisher, Boston.

38. Davenport, T.H. (1993). *Process Innovation: Reengineering work through information technology*, Harvard Business School Press, Boston.

39. Demirors, O., & Frailey, D.J. (1995). A horizontal approach for software process improvement, *Proceedings of Nineteenth Annual International Computer Software and Applications Conference*, COMPSAC-95. Dallas, TX USA. 326 - 331

40. Demirors, O. (1995). *A Horizontal Reflective Process Modeling Approach for Managing Change in Software Development Organizations*. Ph.D. Thesis. School of Engineering and Applied Science, Southern Methodist University.

41. Demirors, O. (1997). Assumptions and difficulties of software quality movement. *Proceedings of the 23rd EUROMICRO Conference - New Frontiers of Information Technology*. 115 - 122.

42. Demirors, O., Gencel, C. & Tarhan, A. (2003). Utilizing business process models for requirements elicitation. *Proceedings of 29th Euromicro Conference*. IEEE Press. 409 - 412.

43. Demirors, O., Gencel, C & Tarhan, A. (2006). Challenges of Acquisition Planning: Two Large System Acquisition Experiences. *Proceedings of the 32nd Euromicro Conference* (EUROMICRO'06), Cavtat/Dubrovnik, Croatia. 256-263.

44. Demirors, O., Gencel, C & Tarhan, A. (2006). Pre-Contract Challenges: Two Large System Acquisition Experiences. In *Enterprise Integration*, Idea Group Inc., Hershey-PA.

45. Derniame, J-C. et.al. (1992). *Reference Manual for the MASP Definition Language*. (Document Ref: ALF/NCY-JCD/WP-4/4-1-D2.) France.

46. DoD Architecture Framework Working Group. (2004). *DoD Architecture Framework*. Ver.1.0.

47. Donaldson, S.E. & Siegel, S.G. (2000). *Successful Software Development*, 2nd Edition. Prentice Hall PTR.

48. Dowson, M & Fernström, C. (1994). Towards Requirements for Enactment Mechanisms. *Proceedings of the Third European Workshop on Software Process Technology. LCNS, Vol.772.* 90 - 106. Springer-Verlag.

49. Drucker, P.F. (1992). The New Society of Organizations, *Harvard Business Review*, Sept-Oct, 95-104.

50. Drucker, P.F. (1993). Port-capitalist Society. HarperCollins Publishers, Inc. NY, USA.

51. Dyba, T., Dingsoyr, T. & Moe, N.B. (2004). Process Improvement In Practice: A Handbook for IT Companies. Kluwer Academic Publishers.

52. Elzinga, D.J., Horak, T., Lee, C. & Bruner, C. (1995). Business Process Management: Survey and Methodology. *IEEE Transactions On Engineering Management, Vol.42, No.2.*

53. Emmerich, W. & Gruhn, V. (1991). FUNSOFT Nets: A Petri-Net based Software Process Modeling Language. *In Proceedings of the 6th International Workshop on Software Specification and Design*, Como, Italy. 175-184. IEEE Computer Society Press.

54. Finkelstein, A., Kramer, J., Nuseibeh, B, Finkelstein, L. & Goedicke, M. (1992). Viewpoints: A Framework for Integrating Multiple Perspectives in System Development. *International Journal of Software Engineering and Knowledge Engineering, Vol. 2, No. 1.* World Scientific Publishing Co. 31-58.

55. Finkelstein, A.C.W., Gabbay, D., Hunter, A., Kramer, J. & Nuseibeh, B. (1994). Inconsistency handling in multiperspective specifications. *IEEE Transactions on Software Engineering, Vol.20, Iss.8,* 569 - 578.

56. Finkelstein, A. & Sommerville, I. (1996). The Viewpoints FAQ. *Software Engineering Journal, Vol. 11.* 2 - 4.

57. Fowler, M. & Scott, K. (1999). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 2nd Ed. Addison Wesley.

58. Garg, P., & Jazayeri, M. (1996). Process-Centered Software Engineering Environments: A Grand Tour. *Trends in Software, Vol. 5.* John-Wiley and Sons.

59. Gillmann, M., Weissenfels, J., Shegalov, G., Wonner, W. & Weikum, G. (2000). A Goal-driven Auto-Configuration Tool for the Distributed Workflow Management System Mentor-lite. *In Proceeding of ACM SIGMOD Conference on Management of Data (SIGMOD)*, Dallas, Texas.

60. Graw, G., Gruhn V. & Krumm H. (1996). Support of cooperating and distributed business processes. *Proceedings of the International Conference on Parallel and Distributed Systems*, 22 -31.

61. Green, S. & Ould, M. (2004). The Primacy of Process Architecture. *Fifth Workshop on Business Process Modelling, Development, and Support, in conjunction with the Conference on Advanced Information Systems Engineering (CAiSE04)*. Riga, Latvia.

62. Green, S. & Ould, M. (2005). A framework for classifying and evaluating process architecture methods. *Software Process: Improvement and Practice*. Vol.10, Iss.4. 415 - 425.

63. Gruhn, V. (1994). Interpersonal Process Support Systems. *Season Report 1/94*, Lion GmbH, Bochum, Germany.

64. Grundy, J., Hosking, J., & Mugridge, W.B. (1998). Inconsistency management for multiple-view software development environments, *IEEE Transactions on Software Engineering, Vol.24, Iss.11*, 960 - 981.

65. Hamel, J., Dufour, S., & Fortin, D. (1993). *Case Study Methods*. Newbury Park, CA: Sage Publications, Inc.

66. Hammer, M. & Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*, Harper Business. NY.

67. Hanrahan, R.P. (1995). The IDEF Process Modeling Methodology. *Crosstalk*. 1995-06.

68. Holt, A.W., Ramsey, H.R. & Grimes, J.D. (1983). Coordination System Technology as the basis for a programming environment. *Electrical Communication*. Vol.57-4. 308-314.

69. Hodkinson, P. & Hodkinson, H. (2001). The strengths and limitations of case study research. *Proceedings of the Annual Learning and Skills Development Agency Conference*.

70. Huff, K.E. & Lesser, V.R. (1988). A plan-based intelligent assistant that supports the software development process. *In Proceedings of the 3rd ACM Symposium on Software Development Environments*, (pp. 97–106). Boston, Massachusetts.

71. Huff, K.E. (1989). GRAPPLE Example: Processes As Plans. *Proceedings of the 5th International Software Process Workshop - Experience with Software Process Models*, 156 - 158.

72. Humphrey, W.S. (1989). *Managing the Software Process*. Addison-Wesley Pub. Co., Inc.

73. Humphrey, W.S. (1998). *A Discipline for Software Engineering*. Addison Wesley, Longman Inc.

74. IDS Scheer AG. (2004). *ARIS Method*. ARIS 6 Collaborative Suite 6.2.3.

75. International Standards Organization - ISO. (1999). ISO 15704: Requirements for Enterprise Reference Architecture and Methodologies, ISO TC 184/SC5/WG1.

76. International Standards Organization - ISO. (2004). *ISO/IEC TR 12207. Information technology - Software life cycle processes.* ISO/IEC 12207 & AMD1/2.

77. Jackson, M. (1995). *Software Requirements & Specifications: A lexicon of practice, principles and prejudices*. Addison Wesley.

78. Jennings, N.R. et .al. (1996). ADEPT: Managing Business Processes using Intelligent Agents. *Proceedings of the 16th Annual Conference of the British Computer Society: Specialist Group on Expert Systems (ISIP Track).*

79. Jennings,N.R., Sycara, K. & Wooldridge, M. (1998). A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems, 1,* 7-38, Kluwer Academic Publishers.

80. Jennings, N.R. & Wooldridge, M. (2000). Agent-Oriented Software Engineering. *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99).*

81. Jennings, N.R., Norman, T.J., Faratin, P., O'Brien, P. & Odgers, B. (2000). Autonomous agents for Business Process Management. *International Journal of Applied AI 14(2),* 145-189.

82. Jennings, N.R. (2001). An Agent-Based Approach for Building Complex Software Systems. *Communications of The ACM, Vol. 44, No. 4.*

83. Junkermann, G., Peuschel, B., Schfer, W., & Wolf, S. (1994). MERLIN: Supporting Cooperation in Software Development Through a Knowledge-Based Environment. In A. Finkelstein, J. Kramer, and B. Nuseibeh (Ed.), *Software Process Modelling and Technology: Chapter 5*, (pp. 103-129). Research Studies Press Limited (John Wiley).

84. Kaiser, G. E. (1988). Rule-based modelling of the software development process. *Proceedings of the 4th international software process workshop on Representing and enacting the software process*, (pp. 84 - 86.). Devon, United Kingdom.

85. Kan, S.H. (2002). *Metrics and Models in Software Quality Engineering*, 2nd Ed. Addison Wesley.

86. Kasse, T. (2004). *Practical insight into CMMI.* Artech House.

87. Katayama, T. (1989). A Hierarchical and Functional Software Process Description and its Enaction. *In Proceedings of the 11th International Conference on Software Engineering,* (pp. 343–352). Pittsburgh, PA.

88. Katzenstein, G. & Lerch, F.J. (2000). Beneath the Surface of Organizational Processes: A Social Representation Framework for Business Process Redesign. *ACM Transactions on Information Systems, Vol. 18, No. 4*, 383–422.

89. Keen, G.W.P. (1997). *The Process Edge: Creating Value Where It Counts*. Harward Business School Press, Boston, Massachusetts.

90. Kellner, M.I. et.al. (1991). ISPW-6 Software Process Example. *Proceedings of the ISPW-6.* IEEE.

91. Kenneth, R.S. & Baker, E.R. (1999). *Software Process Quality: Management and Control.* Marcel Dekker, Inc., NY, USA.

92. Knowledge Based Systems, Inc. (2007). IDEF Integrated DEFinition Methods. Last viewed in 31.March.2007. http://www.idef.com/.

93. Kosanke, K., Jochem, R., Nell, J.G. & Ortiz Bas, A. (2002). Summary of the Proceedings of Enterprise Inter- and Intra-Organizational Integration Workshops. *In IFIP International Federation for Information Processing, Vol. 108.* Springer.

94. Kruchten, P. (2003). *The Rational Unified Process: An Introduction*, 3rd Ed. Addison Wesley.

95. Land, F.F. (1985). Is an information theory enough?, *Computer Journal 28, 3*, 211-215.

96. Laudon, K.C. & Laudon, J.P. (1998). *Management Information System: New Approaches to Organization & Technology*. Printice Hall International, Inc.

97. Le Brasseur, M. & Perdreau, G. (1995). PROCESS WEAVER: from CASE to Workflow Applications. *IEE Colloquium on CSCW (Computer Supported Co-operative Working) and the Software Process (Digest No. 1995/036),* (pp. 7/1 -7/5).

98. Leonhardt, U., Kramer, J., Nuseibeh, B. & Finkelstein, A. (1995). Decentralised Process Enactment in a Multi-Perspective Development Environment. *Proceedings of the 17th international conference on Software engineering*, (pp. 255-264). Seattle, Washington, USA.

99. Lindsay, A., Downs, D. & Lunn, K. (2003). Business processes—attempts to find a definition. *Information and Software Technology, 45.* 1015–1019.

100. List, B. & Korherr, B. (2006). An Evaluation of Conceptual Business Process Modelling languages. Proceedings of the 2006 ACM Symposium on Applied Computing. ACM Press. New York, NY, USA.

101. Lonchamp, J. (1993). A structured conceptual and terminological framework for software process engineering. *Proceedings of the Second International Conference on Software Process- Continuous Software Process Improvement*, (pp. 41-53). Berlin, Germany. IEEE Computer Society Press, Los Alamitos, CA.

102. Lonchamp, J. (1994). An assessment exercise. *Software Process Modeling and Technology*, In A. Finkelstein, J. Kramer, and B. Nuseibeh, (Ed). Research Studies Press, London, U.K.

103. Loos, P. & Allweyer, T. (1998). Process Orientation and Object-Orientation - An Approach for Integrating UML and Event-Driven Process Chains (EPC). *Publication of the Institut für Wirtschaftsinformatik*. Saarbrücken, Germany.

104. Malone, T.W., Crowston, K. & Herman, G.A. (Eds). (2003). *Organizing Business Knowledge: The MIT Process Handbook*. The MIT Press.

105. Marca, D.A. & McGowan, C.L. (1987). *SADT: Structured Analysis and Design Technique*. McGraw-Hill, Inc. New York, NY, USA.

106. Markus, L. & Keil, M. (1994). If we build it they will come: Designing information systems that users want to use. *Sloan Management Review 35, 4*, 11 - 25.

107. McCann, J. E. & Galbraith, J. R. (1981). Interdepartmental relations. In P. C. Nystrom and W. H. Starbuck (Ed.), *Handbook of Organization Design, Vol. 2*, Oxford University Press.

108. Mullery, G. (1979). CORE - a method for controlled requirements expression. *Proceedings of 4th International Conference on Software Engineering (ICSE-4)*, (pp. 126-135), IEEE Computer Society Press.

109. Muth, P., et.al. (1998). From Centralized Workflow Specification to Distributed Workflow Execution. *Journal of Intelligent Information Systems, 10*, 159–184. Kluwer Academic Publishers.

110. Myers, M.D. (1997). Qualitative Research in Information Systems. *MIS Quarterly, 21:2*, (pp. 241-242).

111. Nuseibeh, B. (1994). *A Multi-Perspective framework for Method Integration*. PhD Thesis, Department of Computing, Imperial College, London.

112. Nuseibeh, B., Kramer, J., Finkelstein, A. (2003). ViewPoints: meaningful relationships are difficult!, *Proceedings of the 25th international conference on Software engineering*.

113. OMG (2005). *Unified Modeling Language: Superstructure*, Ver.2.0, Formal/05-07-04, Object Management Group.

114. Osterweil, L. (1987). Software processes are software too. *Proceedings of the 9th international conference on Software Engineering,* (pp. 2-13). Monterey, California, USA.

115. Ould, M. (1995). *Business Processes - modelling and analysis for re-engineering and improvement*. John Wiley & Sons. Chichester, UK.

116. Ould, M. (1997). Designing a re-engineering proof process architecture. *Business Process Management Journal, Vol. 3 No. 3*, 232-247, MCB University Press.

117. Ould, M. (2003). Preconditions for putting processes back in the hands of their actors. *Information and Software Technology, Vol. 45* Elsevier B.V. 1071-1074.

118. Ould, M. (2005). *Business Process Management: A Rigorous Approach*. BCS Publishing, London, UK.

119. Partridge, D. (1998). *Artificial Intelligence and Software Engineering - Understanding the Promise of the Future*. Glenlake Publishing Company, Ltd. and AMACOM.

120. Paulk, M., Chrissis, M., Curtis, B. & Weber, C. (1994). The Capability Maturity Model: Guidelines For Improving The Software Process. Addison Wesley.

121. Pender, J. (2003). *UML Bible*. John Wiley & Sons.

122. Petrie, C. (2001). Agent-Based Software Engineering. *Lecture Notes in AI 1957*, Springer-Verlag.

123. Pressman, R. S. (1992). *Software Engineering: A Practitioner's Approach*. 3rd Ed. McGraw-Hill Inc.

124. Rational Univ. (2000). *Rational Unified Process Fundamentals - Instructor Manual*. Version 2000.02.10.

125. Scheer, W.A. (1999). *ARIS- Business Process Frameworks.* 3rd Ed., Springer-Verlag, Berlin

126. Scheer, W.A. (2000). *ARIS- Business Process Modeling*. Springer-Verlag, Berlin

127. Senge, P. (1999). It's the Learning: The Real Lesson of the Quality Movement. *Journal for Quality & Participation, Vol.22, Iss.6*.

128. Smith, H. & Fingar, P. (2002). *Business Process Management: The Third Wave*, Wiley, Chichester.

129. Sommerville, I. & Rodden, T. (1995). *Human, Social and Organisational Influences on the Software Process*, (TR: CSEG/2/1995), CSEG, Computing Dept., Lancaster University.

130. Sommerville, I., Kotonya, G., Viller, S. & Sawyer, P. (1995). Process Viewpoints. *Proceedings of the 4th European Workshop on Software Process Technology*, (pp. 2 - 8). Springer-Verlag.

131. Sommerville, I., Sawyer, P. & Viller, S. (1999). Managing Process Inconsistency using Viewpoints, *IEEE Transactions on Software Engineering, 25 (6)*, 784-99.

132. Söderström, E., Andersson, B, Johannesson, P., Perjons, E, & Wangler, B. (2002). Towards a Framework for Comparing Process Modelling Languages. CAISE 2002. In A. Banks Pidduck et.al. (Ed.). *LNCS 2348,* (pp. 600-611). Springer-Verlag, Berlin, Heidelberg.

133. Sutton Jr., S.M., Heimbigner, D. & Osterweil, L.J. (1990). Language Constructs for Managing Change in Process-Centered Environments. *Proceedings of the 4th ACM SIGSOFT symposium on Software development environments*, (pp.206-217). Irvine, California, USA.

134. Sutton Jr., S.M., Heimbigner, D. & Osterweil, L.J. (1995). APPL/A: A Language for Software Process Programming". *ACM Transactions on Software Engineering and Methodology (TOSEM), Vol.4, Iss.3.* 221-286.

135. Sutton Jr., S.M., Staudt Lerner, B. & Osterweil, L.J. (1997). *Experience Using the JIL Process Programming Language to Specify Design Processes.* (Technical Report 97-68). Computer Science Department, University of Massachusetts at Amherst.

136. The Open Group. (2006). *The Open Group Architecture Framework (TOGAF)* 8.1.1 2006 Edition.

137. Totland, T. & Conradi, R. (1995). A Survey and Classification of Some Research Areas Relevant to Software Process Modeling. *EWSPT'95*, Leiden.

138. Turetken, O. & Demirors, O. (2005). Concurrent Software Process Modeling. *Proceedings of European Software Process Improvement and Innovation Conference (EuroSPI-2005)*, Budapest, Hungary.

139. Turetken, O. & Demirors, O. (2007). An Approach for Decentralized Process Modeling. In Q. Wang, D. Pfahl, and D.M. Raffo (Eds.): ICSP 2007, *Lecture Notes in Computer Science (LNCS) 4470*, pp. 195–207. Springer.

140. Turetken, O. (2007). *Decentralized Business Process Modeling: A Case Study in the Department of Information Systems, METU II.* Technical Report METU/II-TR-2007-10. Middle East Technical University, Informatics Institute.

141. Turetken, O. (2007). *Decentralized Business Process Modeling: A Case Study in a Small Software Organization*. Technical Report METU/II-TR-2007-11. Middle East Technical University, Informatics Institute.

142. Turgeon, J. & Madhavji, N.H. (1996). A Systematic, View-Based Approach to Eliciting Process Models. *Proceedings of the 5th European Workshop on Software Process Technology*, (pp. 276-282).

143. Turgeon, J. (1999). *A View-Based System for Eliciting Software Process Models.* Ph.D. thesis, McGill University.

144. Vallespir, B., Ducq, Y. & Doumeingts, G. (1999). Enterprise modelling and performance – Part 1: Implementation of performance indicators. *International Journal of Business Performance Management*. Vol.1, No.2. 134 - 153.

145. Verlage, M. (1994). Multi-View Modeling of Software Processes. *In Proceedings of EWSPT3*, (pp. 123-127). Springer-Verlag, Grenoble, France.

146. Verlage, M. (1996). About views for modeling software processes in a role-specific manner. *Joint proceedings of the second international software architecture workshop (ISAW-2) and*

international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops, (pp: 280-284). San Francisco, California, USA.

147. van der Aalst, W.M.P. (1998). The Application of Petri Nets to Workflow Management". *The Journal of Circuits, Systems and Computers, 8(1)*:21-66.

148. van der Aalst, W.M.P. & ter Hofstede, A.H.M. (2002). *YAWL: Yet Another Workflow Language.* (TR FIT-TR-2002-06), Queensland University of Technology, Brisbane.

149. van der Aalst, W.M.P., ter Hofstede, A.H.M. & Weske, M. (2003). Business Process Management: A Survey. *International Conference on Business Process Management (BPM 2003)*, LCNS Vol.2678. 1-12. Springer-Verlag, Berlin.

150. West, M. (2004). *Real Process Improvement Using the CMMI.* Auerbach Publications.

151. White, S.A. (2005). An Example of Using BPMN to Model a BPEL Process. *The Workflow Handbook 2005*. Edt. Fischer, L. Future Strategies Inc., Lighthouse Point, FL, USA.

152. Wise, A., Lerner, B.S., McCall, E.K., Osterweil, L.J. & Sutton Jr. S.M. (1999). *Specifying Coordination in Processes Using Little-JIL.* (TR UM-CS-1999-071). University of Massachusetts, Amherst.

153. Wooldridge, M. (1998). Agents and Software Engineering. *AI*IA Notizie*.

154. Wooldridge, M. & Ciancarini, P. (1999). Agent-Oriented Software Engineering. *Handbook of Software Engineering and Knowledge Engineering Vol.0, No.0.* World Scientific Publishing Company.

155. WM Coalition. (1999). *Workflow Management Coalition Terminology & Glossary*. (WFMC-TC-1011, Issue 3.0).

156. Yin, R.K. (1983). *The Case Study Method: An Annotated Bibliography* (1983-1984 ed.). Washington, DC: COSMOS Corporation.

157. Yin, R.K. (1994). *Case Study Research: Design and Methods*. Applied Social Research Methods Series, Vol.5, 2nd Ed., Sage Publications, Inc.

158. Yin, R.K. (2003). *Applications of Case Study Research*. Applied Social Research Methods Series, Vol.34, 2nd Ed., Sage Publications, Inc.

159. Yu, E. & Mylopoulos, J. (1994). Understanding "Why" in Software Process Modelling, Analysis, and Design. *Proceedings of the 16th ICSE*, (pp. 159-168). Sorrento, Italy.

160. Yu, L. (2000). *Agent Oriented and Role Based Business Process Management for Computational Media*. PhD Thesis. Fakultät Der Universität Zürich.

161. Zachman, A.J. (1987). A Framework for Information Systems Architecture. *IBM Systems Journal*. Vol.26, No.3. 276-292.

APPENDICES

APPENDIX A: TYPES OF ROLE EXPECTATIONS AND THEIR FULFILLMENT

Table 21, Table 22 and Table 23 present all possible cases for satisfying role expectations.

Table 21. Type 1 expectation and all possible cases for fulfillment

EXPECTATION TYPE 1: (Role A ← Item X ← Role B) (i.e. Role A receives Inf. item X from Role B) **Satisfied by: (Role B′ → Item X′ → Role A′) WHERE:**			
	Sender (Diagram Owner)	**Item sent**	**Receiver**
Case	**Role B′**	**Item X′**	**Role A′**
1-1	Role B	Item X	Role A
1-2	Role B	Item X	is generalized by Role A
1-3	Role B	Item X	comprises Role A
1-4	Role B	is generalized by Item X	Role A
1-5	Role B	is generalized by Item X	is generalized by Role A
1-6	Role B	is generalized by Item X	comprise Role A
1-7	Role B	comprises Item X	Role A
1-8	Role B	comprises Item X	is generalized by Role A
1-9	Role B	comprises Item X	comprises Role A
1-10	is generalized by Role B	Item X	Role A
1-11	is generalized by Role B	Item X	is generalized by Role A
1-12	is generalized by Role B	Item X	comprises Role A
1-13	is generalized by Role B	is generalized by Item X	Role A
1-14	is generalized by Role B	is generalized by Item X	is generalized by Role A
1-15	is generalized by Role B	is generalized by Item X	comprise Role A
1-16	is generalized by Role B	comprises Item X	Role A
1-17	is generalized by Role B	comprises Item X	is generalized by Role A
1-18	is generalized by Role B	comprises Item X	comprises Role A
1-19	comprises Role B	Item X	Role A
1-20	comprises Role B	Item X	is generalized by Role A
1-21	comprises Role B	Item X	comprises Role A
1-22	comprises Role B	is generalized by Item X	Role A
1-23	comprises Role B	is generalized by Item X	is generalized by Role A
1-24	comprises Role B	is generalized by Item X	comprise Role A
1-25	comprises Role B	comprises Item X	Role A
1-26	comprises Role B	comprises Item X	is generalized by Role A
1-27	comprises Role B	comprises Item X	comprises Role A
1-*28**	Role B	is included by Item X (i.e. item X comprises item X')	Role A
1-*29**	Role B	is included by Item X	is generalized by Role A
1-*30**	Role B	is included by Item X	comprises Role A
1-*31**	is generalized by Role B	is included by Item X	Role A
1-*32**	is generalized by Role B	is included by Item X	is generalized by Role A
1-*33**	is generalized by Role B	is included by Item X	comprises Role A
1-*34**	comprises Role B	is included by Item X	Role A
1-*35**	comprises Role B	is included by Item X	is generalized by Role A
1-*36**	comprises Role B	is included by Item X	comprises Role A

Table 22. Type 2 expectation and all possible cases for fulfillment

EXPECTATION TYPE 2: (Role A →Item X → Role B) (i.e. Role A sends Inf. item X to Role B) **Satisfied by: (Role B′ ← Item X′ ← Role A′) WHERE:**			
	Receiver (Diagram Owner)	**Item received**	**Sender**
	Role B′	**Item X′**	**Role A′**
2-1	Role B	Item X	Role A
2-2	Role B	Item X	is generalized by Role A
2-3	Role B	Item X	comprises Role A
2-4	Role B	is generalized by Item X	Role A
2-5	Role B	is generalized by Item X	is generalized by Role A
2-6	Role B	is generalized by Item X	comprise Role A
2-7	Role B	comprises Item X	Role A
2-8	Role B	comprises Item X	is generalized by Role A
2-9	Role B	comprises Item X	comprises Role A
2-10	is generalized by Role B	Item X	Role A
2-11	is generalized by Role B	Item X	is generalized by Role A
2-12	is generalized by Role B	Item X	comprises Role A
2-13	is generalized by Role B	is generalized by Item X	Role A
2-14	is generalized by Role B	is generalized by Item X	is generalized by Role A
2-15	is generalized by Role B	is generalized by Item X	comprise Role A
2-16	is generalized by Role B	comprises Item X	Role A
2-17	is generalized by Role B	comprises Item X	is generalized by Role A
2-18	is generalized by Role B	comprises Item X	comprises Role A
2-19	comprises Role B	Item X	Role A
2-20	comprises Role B	Item X	is generalized by Role A
2-21	comprises Role B	Item X	comprises Role A
2-22	comprises Role B	is generalized by Item X	Role A
2-23	comprises Role B	is generalized by Item X	is generalized by Role A
2-24	comprises Role B	is generalized by Item X	comprise Role A
2-25	comprises Role B	comprises Item X	Role A
2-26	comprises Role B	comprises Item X	is generalized by Role A
2-27	comprises Role B	comprises Item X	comprises Role A
2-*28**	Role B	is included by Item X (i.e. item X comprises item X')	Role A
2-*29**	Role B	is included by Item X	is generalized by Role A
2-*30**	Role B	is included by Item X	comprises Role A
2-*31**	is generalized by Role B	is included by Item X	Role A
2-*32**	is generalized by Role B	is included by Item X	is generalized by Role A
2-*33**	is generalized by Role B	is included by Item X	comprises Role A
2-*34**	comprises Role B	is included by Item X	Role A
2-*35**	comprises Role B	is included by Item X	is generalized by Role A
2-*36**	comprises Role B	is included by Item X	comprises Role A

Table 23. Type 3 expectation and all possible cases for fulfillment

EXPECTATION TYPE 3: (Role A → Activity Z ← Role B) *(i.e.* Role A performs Activity Z together with Role B*)* **Satisfied by: (Role B′ → Activity Z′ ← Role A′) WHERE:**			
	Performer (Diagram Owner)	**Activity performed**	**Participator**
	Role B′	**Activity Z′**	**Role A′**
3-1	Role B	Activity Z	Role A
3-2	Role B	Activity Z	is generalized by Role A
3-3	Role B	Activity Z	comprises Role A
3-4	is generalized by Role B	Activity Z	Role A
3-5	is generalized by Role B	Activity Z	is generalized by Role A
3-6	is generalized by Role B	Activity Z	comprises Role A
3-7	comprises Role B	Activity Z	Role A
3-8	comprises Role B	Activity Z	is generalized by Role A
3-9	comprises Role B	Activity Z	comprises Role A

APPENDIX B: THE EXECUTION PLAN FOR CASE STUDY 1

1. Overview

This document presents the plan for the execution of the Plural method in the Department of Information Systems, METU II. The objective is to model specific set of department's processes via the Plural method. The method and related activities are described in Section 5 of this plan.

The processes that will be covered are given in the scope diagram in Figure 64 (p. 131). The roles that are identified and their relationships are given in Figure 65 (p. 132).

The models to be developed include:

1. Individual role process diagrams for each role and for each process
2. Integrated process diagrams
3. Generated role- dependency and process-dependency diagrams

The project will mainly comprise three phases. Table 24 gives these phases and their schedule.

Table 24. Schedule Summary (Case Study 1)

ID	Phase	Start	Finish	Duration	Effort
1	**The Case Study 1**	**26-Oct-05**	**10-Nov-05**	**90 hrs**	**206 hrs**
1.1	Context Definition	26-Oct-05	28-Oct-05	12 hrs	26 hrs
1.2	Description and Conflict Resolution	28-Oct-05	2-Nov-05	30 hrs	132 hrs
1.3	Integration and Change	3-Nov-05	10-Nov-05	48 hrs	48 hrs

2. References

1. Turetken, Oktay (2005). *A Framework for Agent-Based Concurrent Business Process Modeling*, 2nd PhD Thesis Progress Report. Department of Information Systems, METU II (unpublished report).

3. Project Organization

The project roles and their relationships is as follows:

The Plural Process Group: Responsible for context definition and handling macro-level process changes. It comprises all participating agents.

The Coordination Team: Facilitates the execution of the method, verifies models, generates new ones, performs training and orientation for the agents, and facilitates meetings.

Development Agents: Perform individual process modeling and conflict resolution. They are also responsible for the maintenance of their models.

The resources and their assignment to the roles are given in Table 25.

Table 25. Project Resources and Their Assignments (Case Study 1)

Resource	Project Role	Mapped Process Role
Agent 1	Development Agent 1	ABD Bşk.
Agent 2	Development Agent 2, Coordinator 1	Akademik Komite
		Öğr.El. Alımı Değerlendirme Jürisi
		Öğrenci Ders Danışmanı
		Öğretim Üyesi
Agent 3	Development Agent 3	Bölüm Sekreteri
Agent 4	Development Agent 4	Öğrenci
		Öğrenci (Özel)
		Öğrenci (SM/ION)
Agent 5	Development Agent 5	Öğretim Görevlisi
		Ders Öğretmeni
Agent 6	Development Agent 6, Coordinator 2	Ögretim Elemanı
		Ögrenci (IS Prog.)
		Araş. Gör.

4. Work Plan

The schedule for the project and resource assignments is given in Figure 81.

ID	WBS	Task Name	Duration	Start	Finish	Work	Pred	Resource Names
1	1	**Decentralized Process Modeling Case Study 1**	**90 hrs**	**10/26/05**	**11/10/05**	**206 hrs**		
2	**1.1**	**Context Definition**	**12 hrs**	**10/26/05**	**10/28/05**	**26 hrs**		
3	1.1.1	Kickoff Meeting	2 hrs	10/26/05	10/26/05	12 hrs		Dev.Agent 1,Dev.Agent 2,Dev.Agent 3,Dev.Agent 4,Dev.Agent 5,Dev.Agent 6
4	1.1.2	Documenting scope and role diagrams	8 hrs	10/27/05	10/27/05	12 hrs	3	Coordinator 1,Coordinator 2[50%
5	1.1.3	Documenting execution plan	2 hrs	10/28/05	10/28/05	2 hrs	4	Coordinator 1
6	**1.2**	**Description and Conflict Resolution**	**30 hrs**	**10/28/05**	**11/2/05**	**132 hrs**	**2**	
7	1.2.1	Agent 1 - ABD Bşk.	4 hrs	10/28/05	10/28/05	6 hrs		Dev.Agent 1,Coordinator 1[50%
8	1.2.2	Agent 2 - Akademik Komite	2 hrs	10/31/05	10/31/05	3 hrs		Dev.Agent 2,Coordinator 2[50%
9	1.2.3	Agent 2 - Öğr.El. Alımı Değerlendirme Jürisi	2 hrs	10/31/05	10/31/05	3 hrs		Dev.Agent 2,Coordinator 2[50%
10	1.2.4	Agent 2 - Öğrenci Ders Danışmanı	2 hrs	10/31/05	10/31/05	3 hrs		Dev.Agent 2,Coordinator 2[50%
11	1.2.5	Agent 2 - Öğretim Üyesi	2 hrs	10/31/05	10/31/05	3 hrs		Dev.Agent 2,Coordinator 2[50%
12	1.2.6	Agent 3 - Bölüm Sekreteri	24 hrs	10/31/05	11/2/05	36 hrs		Dev.Agent 3,Coordinator 1[50%
13	1.2.7	Agent 4 - Öğrenci	8 hrs	10/31/05	10/31/05	12 hrs		Dev.Agent 4,Coordinator 1[50%
14	1.2.8	Agent 4 - Öğrenci (Özel)	2 hrs	11/1/05	11/1/05	3 hrs		Dev.Agent 4,Coordinator 1[50%
15	1.2.9	Agent 4 - Öğrenci (SM/ION)	2 hrs	11/1/05	11/1/05	3 hrs		Dev.Agent 4,Coordinator 1[50%
16	1.2.10	Agent 5 - Öğretim Görevlisi	4 hrs	10/31/05	10/31/05	6 hrs		Dev.Agent 5,Coordinator 2[50%
17	1.2.11	Agent 5 - Ders Öğretmeni	12 hrs	10/31/05	11/1/05	18 hrs		Dev.Agent 5,Coordinator 2[50%
18	1.2.12	Agent 6 - Ögretim Elemanı	4 hrs	10/31/05	10/31/05	6 hrs		Dev.Agent 6,Coordinator 1[50%
19	1.2.13	Agent 6 - Ögrenci (IS Prog.)	4 hrs	10/31/05	10/31/05	6 hrs		Dev.Agent 6,Coordinator 1[50%
20	1.2.14	Agent 6 - Araş. Gör.	16 hrs	11/1/05	11/2/05	24 hrs		Dev.Agent 6,Coordinator 1[50%
21	1.3	Integration and Change	48 hrs	11/3/05	11/10/05	48 hrs	6	Coordinator 1,Coordinator 2[50%

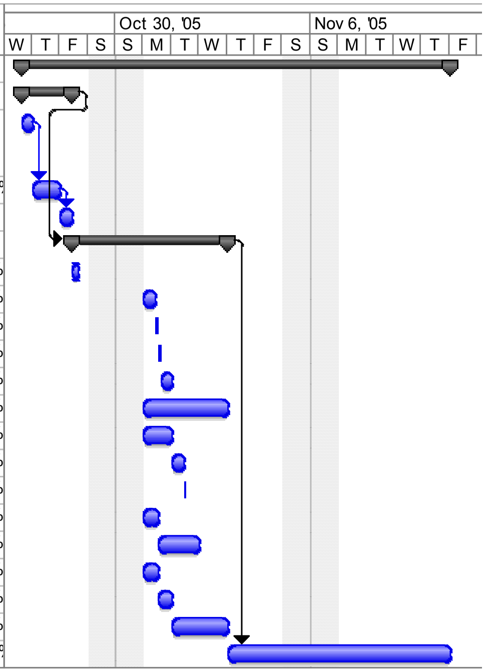


Figure 81. Project Schedule (Case Study 1)

5. Technical Process

The technical process to be followed is governed by the Plural method and described in the progress report [1] given as a reference in Section 2 of this plan.

6. Supporting Plans

6.1 Modeling platform

The modeling will be performed in agents' client PCs via a modeling toolset. The toolset is accessed via the internet browser (MS Internet Explorer 6 or above) through the following IP address:

http://144.122.98.186/webdesigner

6.2 Backup

The configuration manager (Coordinator 1) is responsible to backup the process-base every working day at 18:00 with the following naming convention:

CaseStudy1_vXX.mdb XX representing the version number incremented each day.

Backup files are stored in a separate server which can be accessed through the following IP address: ftp://144.122.98.186/casestudy1

6.3 Directory Structure and Naming Convention for Diagrams

Directory structure for the models is given below:

- *Main Directory*
  - *Context*: Scope and Role diagrams
  - *Role Process Diagrams*
    - *<Role Name>*: Individual role process diagrams for each role and for each process
  - *Generated Diagrams*
    - *Integrated Process Diagrams:* Activity and process level process diagrams
    - *Process Dependency Diagrams*
    - *Role Dependency Diagrams*

Naming Conventions:

Diagram Type	→	Naming
Scope Diagram	→	scope diagram
Role Diagram	→	role diagram
Individual Role Process Diagram	→	<role name>-<process name>
Activity Level Process Diagram	→	<process name>-activity level
Process Level Process Diagram	→	<process name>-process level
Role Dependency Diagram	→	Role dependency - <process names>/<role names>/global
Process Dependency Diagram	→	Process dependency - <process names>/global

6.4 Training/ Orientation

Coordinators may organize training or orientation sessions for development agents if required. Sessions can be scheduled just prior to the modeling so that the first modeling practice can be performed in pairs (development agent and coordinator).

6.5 Tracking Effort

Each agent will track individual time for each work activity depicted in the work plan. Values can be entered into the web-based effort tracking tool accesses through the following IP address:

http://144.122.98.186/case1/

APPENDIX C: DIAGRAMS IN CASE STUDY 1

This section of the thesis presents the list of the diagrams developed in the first case study and gives a subset of these diagrams as samples. All the diagrams listed in Table 26 are presented in the case study report [140].

Table 26. List of Diagrams in Case Study 1

ID	Diagram Name	Fig. No.	Diagram
1	**CONTEXT**		
1.1	Scope diagram	**Figure 82**	Scope Diagram
1.2	Role Diagram	**Figure 83**	Role Diagram
2	**INDIVIDUAL DIAGRAMS**		
2.1	**Araştırma Görevlisi (Research Assistant)**		
2.1.1	Araş. Gör. Alımı - Araş. Gör. (Research Assistant Recruitment - Research Asst.)	**Figure 84**	Individual role-process
2.1.2	Asistan Görev Dağılımının Oluşturulması - Araş. Gör. (Determining Research Asst. Work Plan - Research Asst.)		Individual role-process
2.1.3	Ders Verme - Araş. Gör. (Lecturing - Research Asst.)		Individual role-process
2.1.3.1	Ders Web Sayfasının Oluşturulması/Yayınlanması - Araş. Gör. (Preparing & Publishing Course's Web Page - Research Asst.)		Individual role-process
2.1.3.2	Ders Web Sayfasının Güncellenmesi - Araş. Gör. (Updating Course Web Page - Research Asst.)		Individual role-process
2.1.3.3	Ders Ödevlerinin Sonuçlandırılması - Araş. Gör. (Course Homeworks - Research Asst.)		Individual role-process
2.1.3.4	Ders Projesinin Sonuçlandırılması - Araş. Gör. (Course Projects - Research Asst.)		Individual role-process
2.1.3.5	Ders Sınavlarının Yapılması - Araş. Gör. (Course Exams - Research Asst.)		Individual role-process
2.1.3.6	Not Çizelgesinin Hazırlanması - Araş. Gör. (Preparing Course Grades Table - Research Asst.)		Individual role-process
2.2	**Ana Bilim Dalı (ABD) Bşk. (Program Director)**		
2.2.1	Araş. Gör. Alımı - Ana Bilim Dalı (ABD) Bşk. (Research Assistant Recruitment - Program Director)	**Figure 85**	Individual role-process
2.2.2	Asistan Görev Dağılımının Oluşturulması - Ana Bilim Dalı (ABD) Bşk. (Determining Research Asst. Work Plan - Program Director)		Individual role-process
2.2.3	Öğrenci Alma --Bilişim Sist. Programları - ABD Bşk. (Student Admissions -- Inf. Sys. Programs - Program Director)		Individual role-process
2.2.4	Öğrenci Alma -- Yaz. Yön & Bil. Sist. Online Programları - ABD Bşk. (Student Admissions - Soft. Mng. & Inf. Sys. Online Programs - Program Director)		Individual role-process
2.2.5	Öğrenci İlişik Kesme - ABD Bşk. (Student Quits - Program Director)		Individual role-process
2.2.6	Öğretim Elemanı İlişik Kesme - ABD Bşk. (Academic Staff Quits - Program Director)		Individual role-process
2.2.7	Öğretim Elemanlarının Yıllık İzinleri - ABD Bşk. (Staff Annual Vacations - Program Director)		Individual role-process
2.2.8	Öğretim Görevlisi Alımı - ABD Bşk. (Instructor Recruitments - Program Director)		Individual role-process
2.2.9	Özel Öğrenci Kayıtları - ABD. Bşk. (Special Student Registrations - Program Director)		Individual role-process

Table 26. List of Diagrams in Case Study 1(continued)

ID	Diagram Name	Fig. No.	Diagram
2.3	**Akademik Komite (Academic Committee)**		
2.3.1	Öğrenci Alma -- Bilişim Sist. Programları - Akademik Komite (Student Admissions -- Inf. Sys. Programs - Academic Committee)		Individual role-process
2.4	**Bölüm Sekreteri (Department Secretary)**		
2.4.1	Araş. Gör. Alımı - Böl. Sekreteri (Research Assistant Recruitment - Dept. Secretary)	**Figure 86**	Individual role-process
2.4.2	Dönem Ders Kayıtları ve Ekle/Kaldır - Böl. Sekreteri (Course Registration and Add/Drops - Dept. Secretary)		Individual role-process
2.4.3	Öğrenci Alma -- Bilişim Sist. Programları - Böl. Sekreteri (Student Admissions -- Inf. Sys. Programs - Dept. Secretary)		Individual role-process
2.4.4	Öğrenci Alma -- Yaz. Yön & Bil. Sist. Online Programları- Böl. Sekreteri (Student Admissions - Soft. Mng. & Inf. Sys. Online Programs - Dept. Secretary)		Individual role-process
2.4.5	Öğrenci İlişik Kesme - Böl. Sekreteri (Student Quits - Dept. Secretary)		Individual role-process
2.4.6	Öğretim Elemanı İlişik Kesme - Böl. Sekreteri (Academic Staff Quits- Dept. Secretary)		Individual role-process
2.4.7	Öğretim Elemanlarının Yıllık İzinleri - Böl. Sekreteri (Staff Annual Vacations - Dept. Secretary)		Individual role-process
2.4.8	Öğretim Görevlisi Alımı - Böl. Sekreteri (Instructor Recruitments - Dept. Secretary)		Individual role-process
2.4.9	Özel Öğrenci Kayıtları - Böl. Sekreteri (Special Student Registrations - Dept. Secretary)		Individual role-process
2.5	**Ders Öğretmeni (Lecturer)**		
2.5.1	Ders Verme - Ders Öğretmeni (Lecturing - Lecturer)		Individual role-process
2.5.1.1	Dersin Sunulması - Ders Öğretmeni (Presenting Course - Lecturer)		Individual role-process
2.5.1.2	Ders Ödevlerinin Verilmesi - Ders Öğretmeni (Course Homeworks - Lecturer)		Individual role-process
2.5.1.3	Ders Projesi Verilmesi - Ders Öğretmeni (Course Projects - Lecturer)		Individual role-process
2.5.1.4	Ders Duyurusunun Yapılması - Ders Öğretmeni (Course Announcements - Lecturer)		Individual role-process
2.5.1.5	Ders Sınavlarının Yapılması - Ders Öğretmeni (Course Exams - Lecturer)		Individual role-process
2.5.1.6	Öğrencilerin Notlandırılması - Ders Öğretmeni (Grading Students - Lecturer)		Individual role-process
2.5.2	Özel Öğrenci Kayıtları - Ders Öğretmeni (Grading Students - Lecturer)		Individual role-process
2.6	**Öğr.El. Alımı Değerlendirme Jürisi (Staff Recruitment Evaluation Committee)**		
2.6.1	Araş. Gör. Alımı - Öğr.El. Alımı Değerlendirme Jürisi (Research Asst. Recruitment - Staff Recruitment Evaluation Committee)	**Figure 87**	Individual role-process
2.6.2	Öğretim Görevlisi Alımı - Öğr.El. Alımı Değerlendirme Jürisi (Lecturer Recruitment - Staff Recruitment Evaluation Committee)		Individual role-process
2.7	**Öğrenci (Student)**		
2.7.1	Ders Kayıtları ve Ekle/Kaldır - Öğrenci (Course Registrations & Add/Drops - Student)		Individual role-process
2.7.2	Öğrenci Okul Kayıtları - Öğrenci (Student School Registrations - Student)		Individual role-process
2.7.3	Öğrenci İlişik Kesme - Öğrenci (Student Quits - Student)		Individual role-process
2.8	**Özel Öğrenci (Special Student)**		
2.8.1	Özel Öğrenci Kayıtları - Özel Öğrenci (Special Student Registrations - Special Student)		Individual role-process
2.9	**Öğrenci YY/BSO (Student SM/ION)**		

Table 26. List of Diagrams in Case Study 1(continued)

ID	Diagram Name	Fig. No.	Diagram
2.9.1	Öğrenci Alma -- Yaz. Yön & Bil. Sist. Online Programları - Öğrenci YY/BSO (Student Admissions - Soft. Mng. & Inf. Sys. Online Programs - Student SM/ION)		Individual role-process
2.10	**Öğrenci BS (Student IS)**		
2.10.1	Öğrenci Alma -- Bil. Sist. Programı - Öğrenci BS (Student Admissions - Inf. Sys. Program - Student IS)		Individual role-process
2.11	**Öğrenci Ders Danışmanı (Student Course Advisor)**		
2.11.1	Dönem Ders Kayıtları ve Ekle/Kaldır - Öğrenci Ders Danışmanı (Course Registration & Add/Drops - Student Course Advisor)		Individual role-process
2.12	**Öğretim Görevlisi (Instructor)**		
2.12.1	Öğretim Görevlisi Alımı - Öğretim Görevlisi (Instructor Recruitments - Instructor)		Individual role-process
2.13	**Öğretim Üyesi (Faculty Member)**		
2.13.1	Araş. Gör. Alımı - Öğretim Üyesi (Research Asst. Recruitment - Faculty Member)	**Figure 88**	Individual role-process
2.13.2	Asistan Görev Dağılımının Oluşturulması - Öğretim Üyesi (Determining Research Asst. Work Plan - Faculty Member)		Individual role-process
2.13.3	Öğretim Görevlisi Alımı - Öğretim Üyesi (Instructor Recruitments - Faculty Member)		Individual role-process
2.14	**Öğretim Elemanı (Academic Staff)**		
2.14.1	Öğretim Elemanı İlişik Kesme -Öğretim Elemanı (Academic Staff Quits - Academic Staff)		Individual role-process
2.14.2	Öğretim Elemanlarının Yıllık İzinleri - Öğretim Elemanı (Academic Staff Annual Vacations - Academic Staff)		Individual role-process
3	**GENERATED DIAGRAMS**		
3.1	**Generated Process Diagrams**		
3.1.1	**Araş. Gör. Alımı - Öğretim Üyesi**		
3.1.1.1	Araş. Gör. Alımı (Research Asst. Recruitment)- Acticity Level	**Figure 89**	Activity level process
3.1.1.2	Araş. Gör. Alımı (Research Asst. Recruitment) - Process Level	**Figure 90**	Process level process
3.1.2	**Asistan Görev Dağılımının Oluşturulması (Determining Research Asst. Work Plan)**		
3.1.2.1	Asistan Görev Dağılımının Oluşturulması (Determining Research Asst. Work Plan) - Activity Level		Activity level process
3.1.3	**Ders Verme (Lecturing)**		
3.1.3.1	Dersin Sunulması (Presenting Course) - Activity Level		Activity level process
3.1.3.2	Ders Ödevlerinin Verilmesi (Course Homeworks) - Activity Level		Activity level process
3.1.3.3	Ders Projesi Verilmesi (Course Projects) - Activity Level		Activity level process
3.1.3.4	Ders Duyurusunun Yapılması (Course Announcements) - Activity Level		Activity level process
3.1.3.5	Ders Sınavlarının Yapılması (Course Exams) - Activity Level		Activity level process
3.1.3.6	Öğrencilerin Notlandırılması (Grading Students) - Activity Level		Activity level process
3.1.3.7	Ders Verme (Lecturing) - Process Level		Process level process
3.1.4	**Dönem Ders Kayıtları ve Ekle/Kaldır - (Course Registration & Add/Drops)**		
3.1.4.1	Ders Kayıtları ve Ekle/Kaldır (Course Registration & Add/Drops)- Activity Level		Activity level process
3.1.4.2	Ders Kayıtları ve Ekle/Kaldır (Course Registration & Add/Drops) - Process Level		Process level process
3.1.5	**Öğrenci Alma -- Bilişim Sist. Programları (Student Admissions -- Inf. Sys. Programs)**		
3.1.5.1	Öğrenci Alma -- Bilişim Sist. Programları (Student Admissions -- Inf. Sys. Programs) - Acticity Level		Activity level process

Table 26. List of Diagrams in Case Study 1(continued)

ID	Diagram Name	Fig. No.	Diagram
3.1.5.2	Öğrenci Alma -- Bilişim Sist. Programları (Student Admissions -- Inf. Sys. Programs) - Process Level		Process level process
3.1.6	**Öğrenci Alma -- Yaz. Yön & Bil. Sist. Online Programları (Student Admissions - Soft. Mng. & Inf. Sys. Online Programs)**		
3.1.6.1	Öğrenci Alma -- Yaz. Yön & Bil. Sist. Online Programları (Student Admissions - Soft. Mng. & Inf. Sys. Online Programs) - Activity Level		Activity level process
3.1.6.2	Öğrenci Alma -- Yaz. Yön & Bil. Sist. Online Programları (Student Admissions - Soft. Mng. & Inf. Sys. Online Programs) - Process Level		Process level process
3.1.7	**Öğrenci Okul Kayıtları (Student School Registrations)**		
3.1.7.1	Öğrenci Okul Kayıtları (Student School Registrations) -Actvity Level		Activity level process
3.1.7.2	Öğrenci Okul Kayıtları (Student School Registrations) -Process Level		Process level process
3.1.8	**Öğrenci İlişik Kesme (Student Quits)**		
3.1.8.1	Öğrenci İlişik Kesme (Student Quits) - Activity Level		Activity level process
3.1.8.2	Öğrenci İlişik Kesme (Student Quits) - Process Level		Process level process
3.1.9	**Öğretim Elemanı İlişik Kesme (Staff Quits)**		
3.1.9.1	Öğretim Elemanı İlişik Kesme (Staff Quits) - Activity Level		Activity level process
3.1.9.2	Öğretim Elemanı İlişik Kesme (Staff Quits) - Process Level		Process level process
3.1.10	**Öğretim Elemanlarının Yıllık İzinleri (Staff Annual Vacations)**		
3.1.10.1	Öğretim Elemanlarının Yıllık İzinleri (Staff Annual Vacations) - Activity Level		Activity level process
3.1.10.2	Öğretim Elemanlarının Yıllık İzinleri (Staff Annual Vacations) - Process Level		Process level process
3.1.11	**Öğretim Görevlisi Alımı (Instructor Recruitments)**		
3.1.11.1	Öğretim Görevlisi Alımı (Instructor Recruitments) - Acticity Level		Activity level process
3.1.11.2	Öğretim Görevlisi Alımı (Instructor Recruitments) - Process Level		Process level process
3.1.12	**Özel Öğrenci Kayıtları (Special Student Registrations)**		
3.1.12.1	Özel Öğrenci Kayıtları (Special Student Registrations) - Activity Level		Activity level process
3.1.12.2	Özel Öğrenci Kayıtları (Special Student Registrations) - Process Level		Process level process
3.2	**Role Dependency Diagrams**		
3.2.1	Role Dependency in Student Admissions -- Inf. Sys. Programs	**Figure 91**	Role Dependency
3.2.2	Role Dependency in Instructor Recruitments	**Figure 92**	Role Dependency
3.2.3	Dependency with Inactive Roles	**Figure 93**	Role Dependency

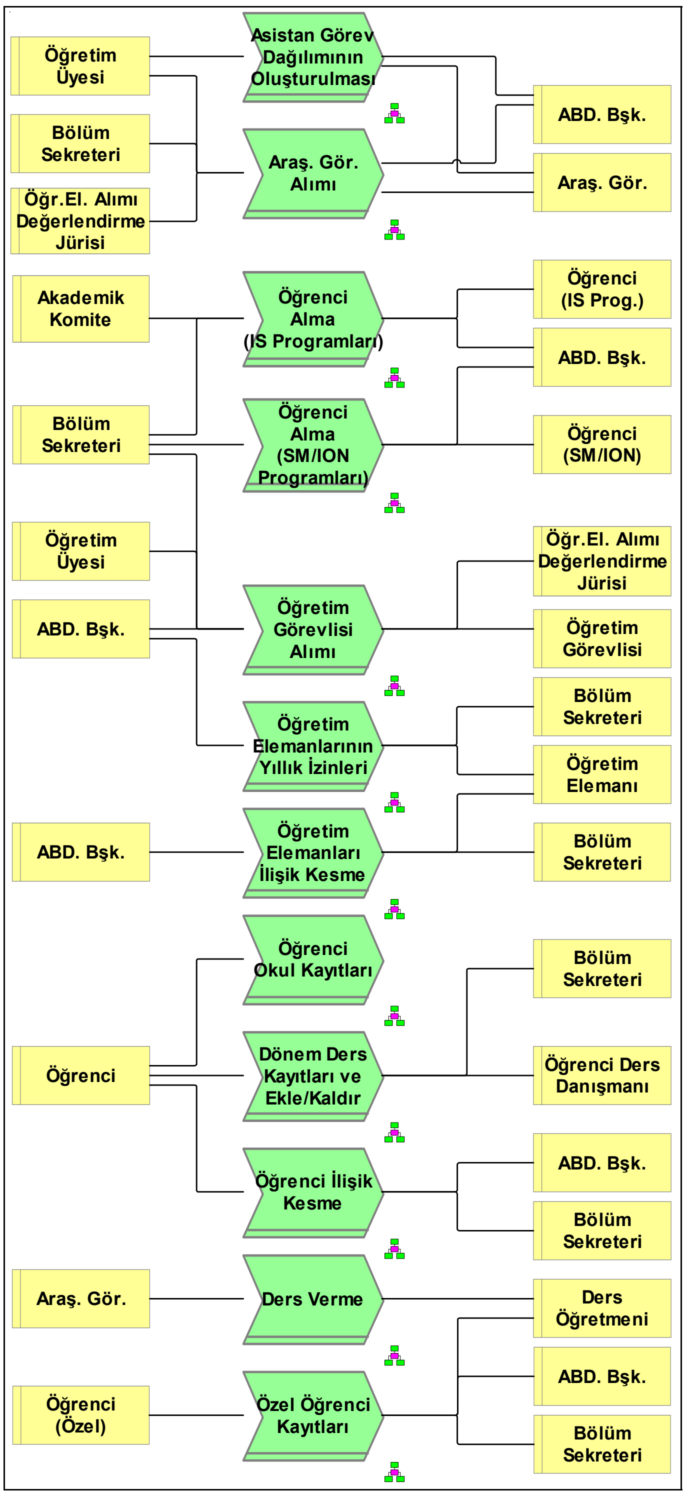


Figure 82. Scope diagram

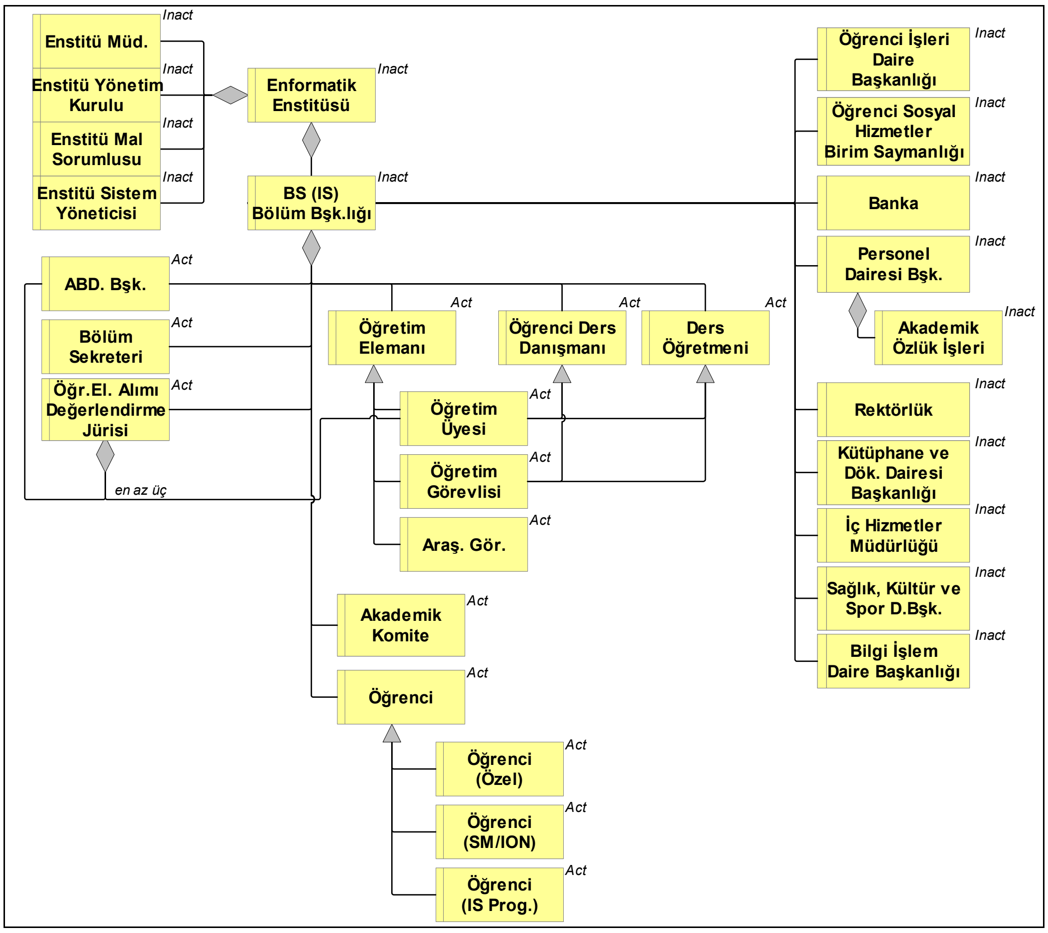


Figure 83. Role Diagram

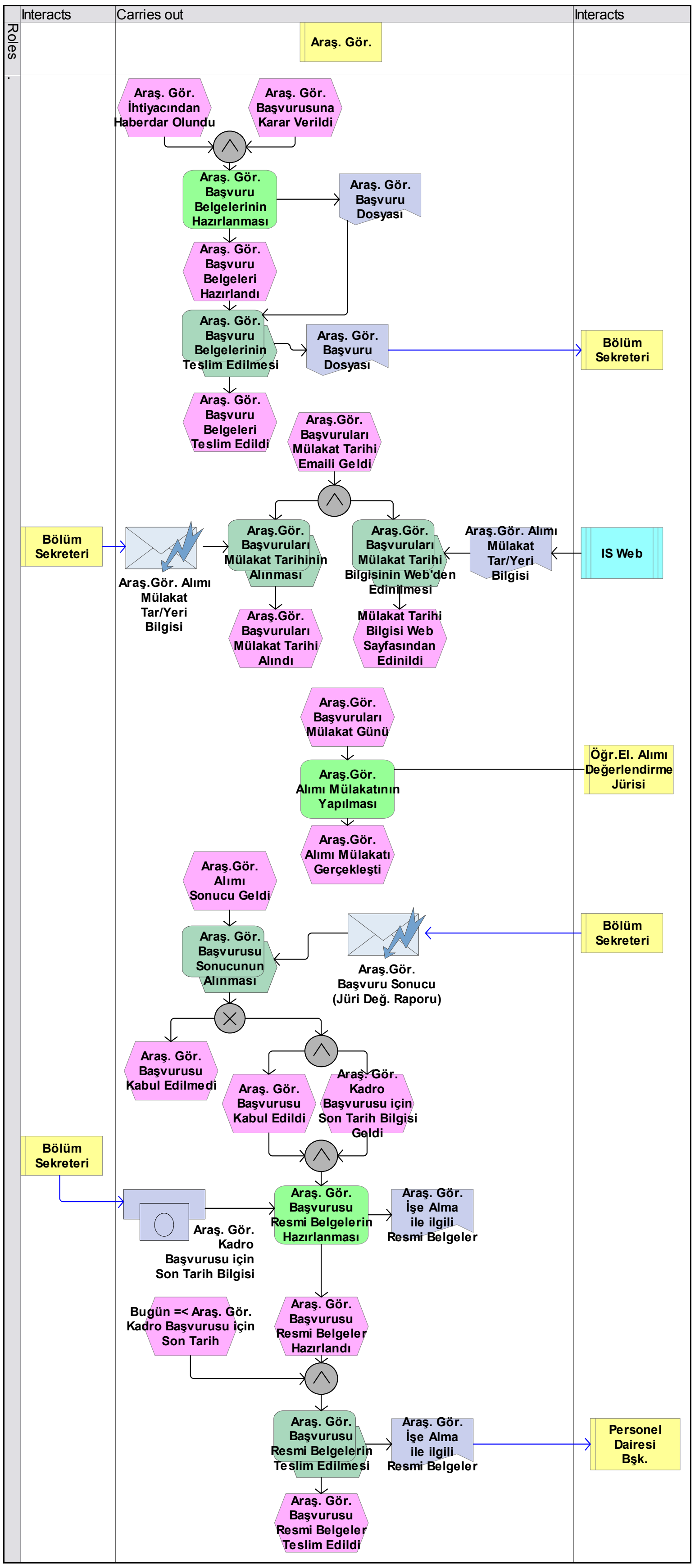

Interacts
Carries out
Interacts
Roles
Araş. Gör.
Araş. Gör. İhtiyacından Haberdar Olundu
Araş. Gör. Başvurusuna Karar Verildi
Araş. Gör. Başvuru Belgelerinin Hazırlanması
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Belgeleri Hazırlandı
Araş. Gör. Başvuru Belgelerinin Teslim Edilmesi
Araş. Gör. Başvuru Dosyası
Bölüm Sekreteri
Araş. Gör. Başvuru Belgeleri Teslim Edildi
Araş.Gör. Başvuruları Mülakat Tarihi Emaili Geldi
Bölüm Sekreteri
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Başvuruları Mülakat Tarihinin Alınması
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisinin Web'den Edinilmesi
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
IS Web
Araş.Gör. Başvuruları Mülakat Tarihi Alındı
Mülakat Tarihi Bilgisi Web Sayfasından Edinildi
Araş.Gör. Başvuruları Mülakat Günü
Araş.Gör. Alımı Mülakatının Yapılması
Öğr.El. Alımı Değerlendirme Jürisi
Araş.Gör. Alımı Mülakatı Gerçekleşti
Araş.Gör. Alımı Sonucu Geldi
Araş. Gör. Başvurusu Sonucunun Alınması
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
Bölüm Sekreteri
Araş. Gör. Başvurusu Kabul Edilmedi
Araş. Gör. Başvurusu Kabul Edildi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi Geldi
Bölüm Sekreteri
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. Başvurusu Resmi Belgelerin Hazırlanması
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Bugün =< Araş. Gör. Kadro Başvurusu için Son Tarih
Araş. Gör. Başvurusu Resmi Belgeler Hazırlandı
Araş. Gör. Başvurusu Resmi Belgelerin Teslim Edilmesi
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Personel Dairesi Bşk.
Araş. Gör. Başvurusu Resmi Belgeler Teslim Edildi

Figure 84. Araş. Gör. Alımı - Araş. Gör. (Research Assistant Recruitment - Research Asst.)

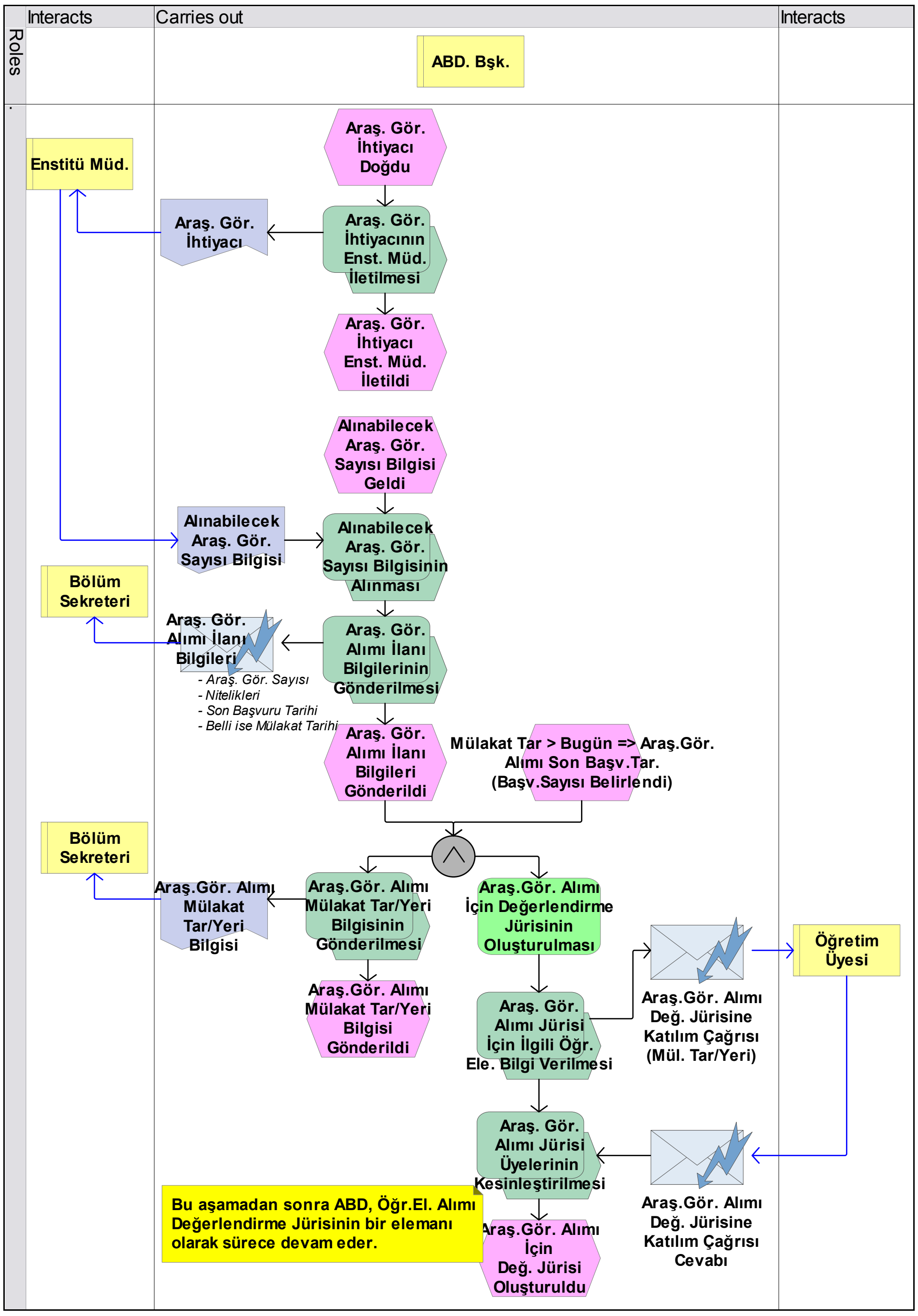

Interacts
Carries out
Interacts
Roles
ABD. Bşk.
Araş. Gör. İhtiyacı Doğdu
Enstitü Müd.
Araş. Gör. İhtiyacı
Araş. Gör. İhtiyacının Enst. Müd. İletilmesi
Araş. Gör. İhtiyacı Enst. Müd. İletildi
Alınabilecek Araş. Gör. Sayısı Bilgisi Geldi
Alınabilecek Araş. Gör. Sayısı Bilgisi
Alınabilecek Araş. Gör. Sayısı Bilgisinin Alınması
Bölüm Sekreteri
Araş. Gör. Alımı İlanı Bilgileri
Araş. Gör. Alımı İlanı Bilgilerinin Gönderilmesi
- Araş. Gör. Sayısı
- Nitelikleri
- Son Başvuru Tarihi
- Belli ise Mülakat Tarihi
Araş. Gör. Alımı İlanı Bilgileri Gönderildi
Mülakat Tar > Bugün => Araş.Gör. Alımı Son Başv.Tar. (Başv.Sayısı Belirlendi)
Bölüm Sekreteri
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisinin Gönderilmesi
Araş.Gör. Alımı İçin Değerlendirme Jürisinin Oluşturulması
Öğretim Üyesi
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi Gönderildi
Araş. Gör. Alımı Jürisi İçin İlgili Öğr. Ele. Bilgi Verilmesi
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı (Mül. Tar/Yeri)
Araş. Gör. Alımı Jürisi Üyelerinin Kesinleştirilmesi
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı Cevabı
Bu aşamadan sonra ABD, Öğr.El. Alımı Değerlendirme Jürisinin bir elemanı olarak sürece devam eder.
Araş.Gör. Alımı İçin Değ. Jürisi Oluşturuldu

Figure 85. Araş. Gör. Alımı - Ana Bilim Dalı (ABD) Bşk. (Research Assistant Recruitment - Program Director)

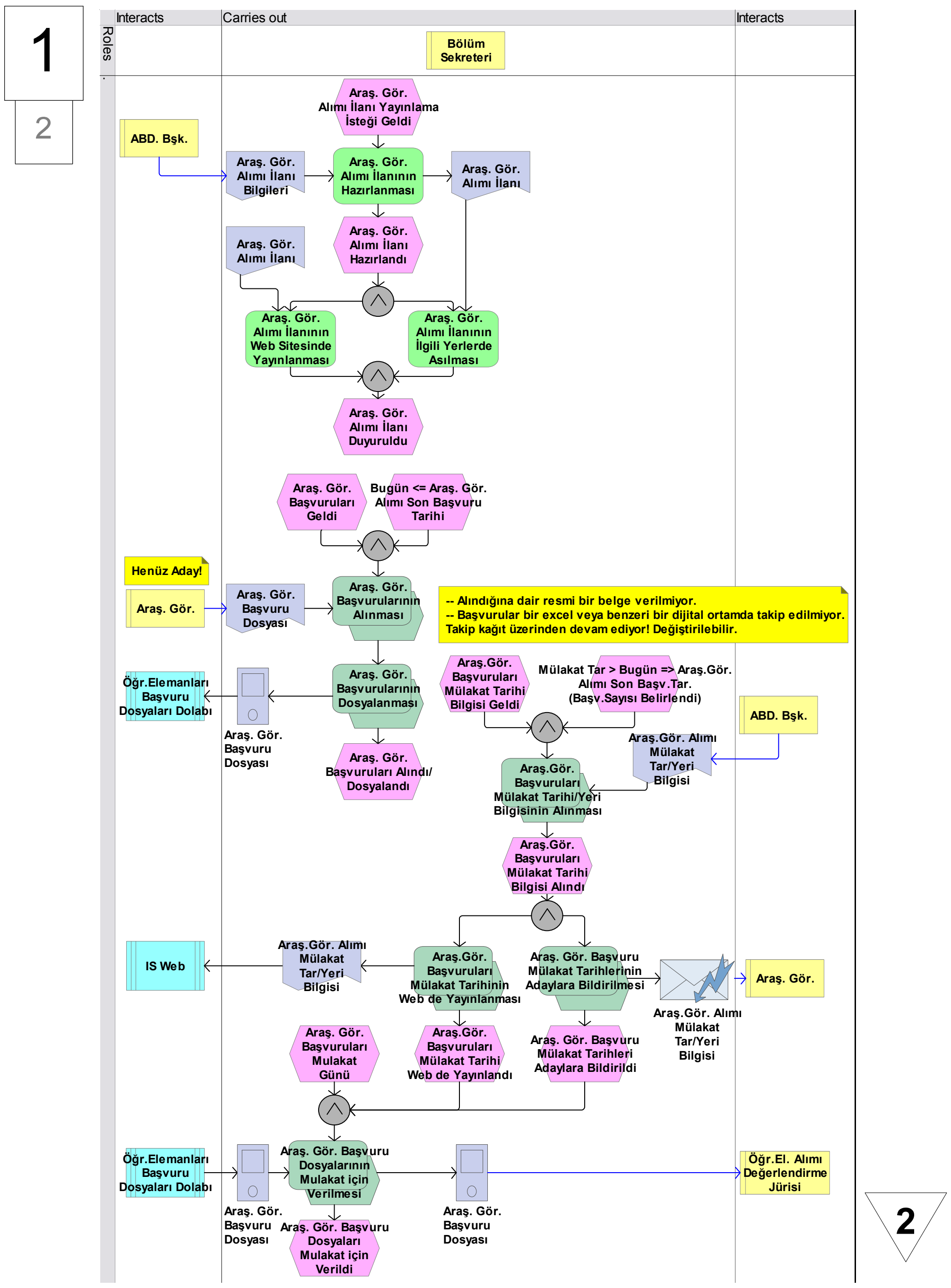

1
2
Interacts
Carries out
Interacts
Roles
Bölüm Sekreteri
Araş. Gör. Alımı İlanı Yayınlama İsteği Geldi
ABD. Bşk.
Araş. Gör. Alımı İlanı Bilgileri
Araş. Gör. Alımı İlanının Hazırlanması
Araş. Gör. Alımı İlanı
Araş. Gör. Alımı İlanı
Araş. Gör. Alımı İlanı Hazırlandı
Araş. Gör. Alımı İlanının Web Sitesinde Yayınlanması
Araş. Gör. Alımı İlanının İlgili Yerlerde Asılması
Araş. Gör. Alımı İlanı Duyuruldu
Araş. Gör. Başvuruları Geldi
Bugün <= Araş. Gör. Alımı Son Başvuru Tarihi
Henüz Aday!
Araş. Gör.
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvurularının Alınması
-- Alındığına dair resmi bir belge verilmiyor.
-- Başvurular bir excel veya benzeri bir dijital ortamda takip edilmiyor. Takip kağıt üzerinden devam ediyor! Değiştirilebilir.
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvurularının Dosyalanması
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisi Geldi
Mülakat Tar > Bugün => Araş.Gör. Alımı Son Başv.Tar. (Başv.Sayısı Belirlendi)
ABD. Bşk.
Araş. Gör. Başvuruları Alındı/ Dosyalandı
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Başvuruları Mülakat Tarihi/Yeri Bilgisinin Alınması
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisi Alındı
IS Web
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Başvuruları Mülakat Tarihinin Web de Yayınlanması
Araş. Gör. Başvuru Mülakat Tarihlerinin Adaylara Bildirilmesi
Araş. Gör.
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş. Gör. Başvuruları Mulakat Günü
Araş.Gör. Başvuruları Mülakat Tarihi Web de Yayınlandı
Araş. Gör. Başvuru Mülakat Tarihleri Adaylara Bildirildi
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Dosyalarının Mulakat için Verilmesi
Araş. Gör. Başvuru Dosyası
Öğr.El. Alımı Değerlendirme Jürisi
Araş. Gör. Başvuru Dosyaları Mulakat için Verildi
2

Figure 86. Araş. Gör. Alımı - Böl. Sekreteri (RA Recruitment - Dept. Secretary)

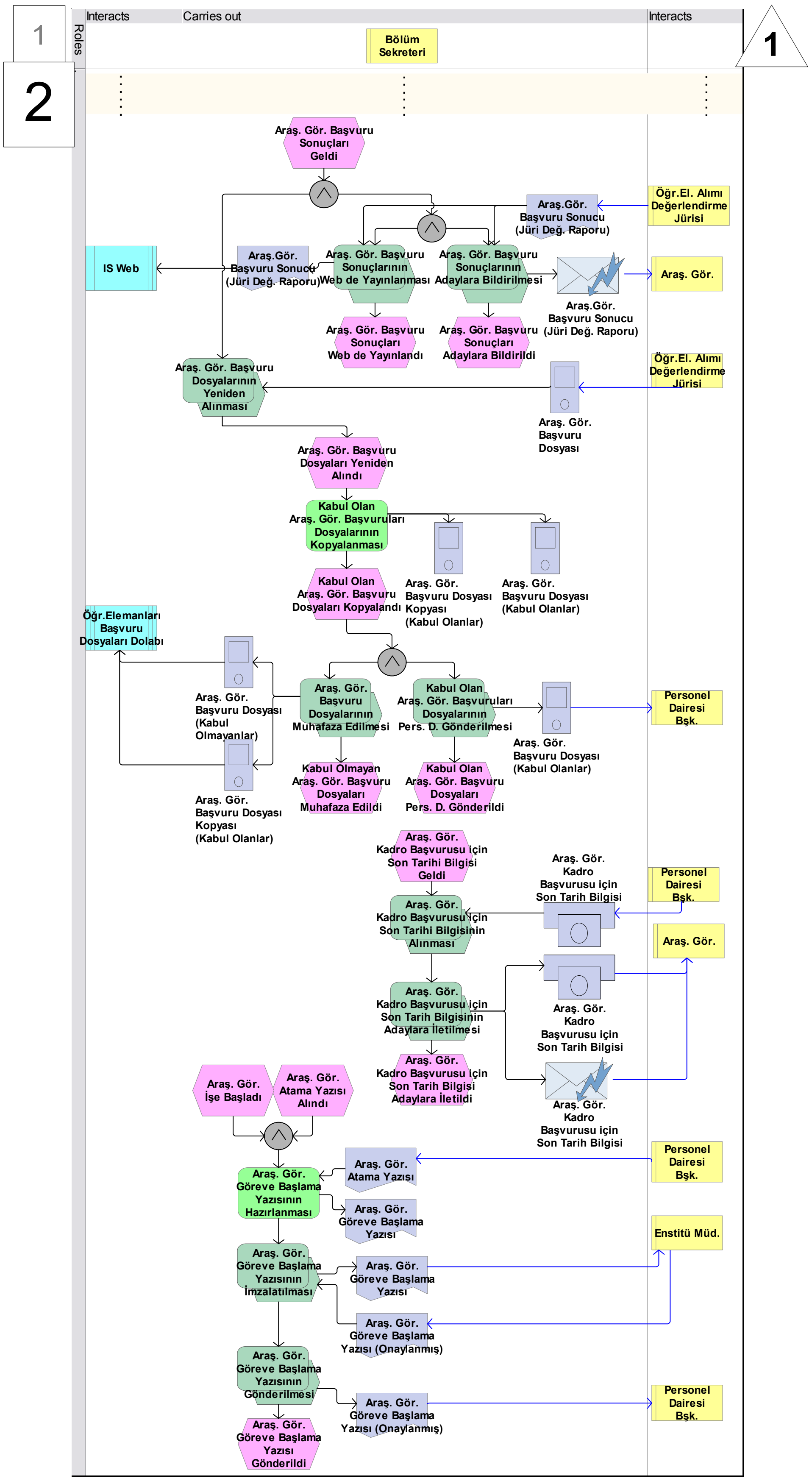
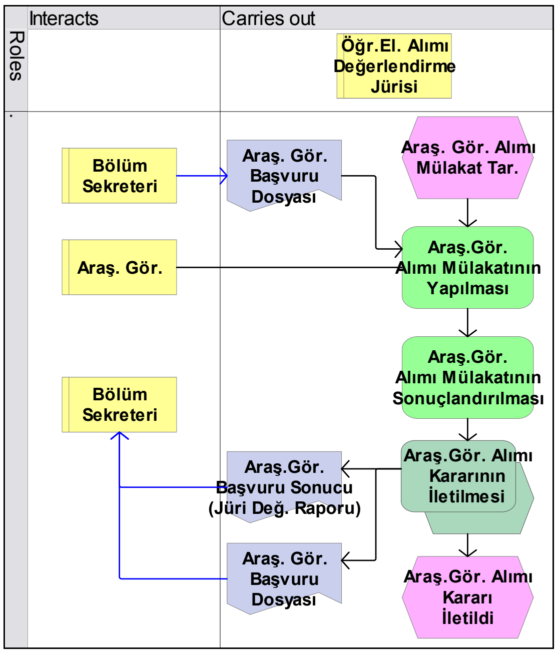
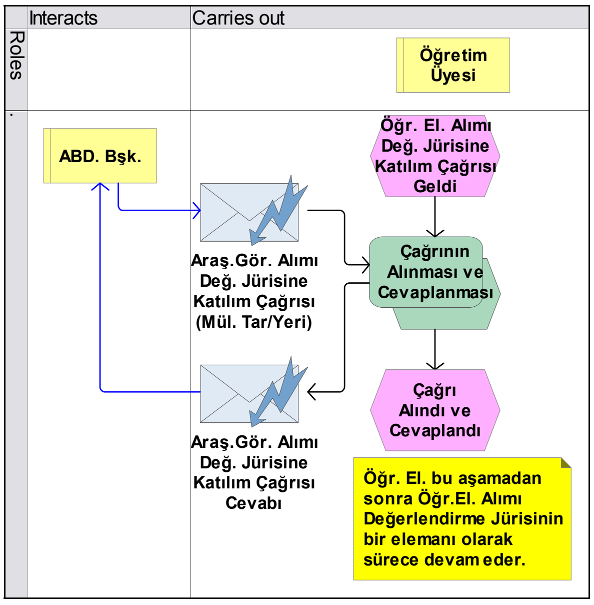

1
2
1
Roles
Interacts
Carries out
Interacts
Bölüm Sekreteri
Araş. Gör. Başvuru Sonuçları Geldi
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
Öğr.El. Alımı Değerlendirme Jürisi
IS Web
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
Araş. Gör. Başvuru Sonuçlarının Web de Yayınlanması
Araş. Gör. Başvuru Sonuçlarının Adaylara Bildirilmesi
Araş. Gör.
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
Araş. Gör. Başvuru Sonuçları Web de Yayınlandı
Araş. Gör. Başvuru Sonuçları Adaylara Bildirildi
Öğr.El. Alımı Değerlendirme Jürisi
Araş. Gör. Başvuru Dosyalarının Yeniden Alınması
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Dosyaları Yeniden Alındı
Kabul Olan Araş. Gör. Başvuruları Dosyalarının Kopyalanması
Kabul Olan Araş. Gör. Başvuru Dosyaları Kopyalandı
Araş. Gör. Başvuru Dosyası Kopyası (Kabul Olanlar)
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. Başvuru Dosyası (Kabul Olmayanlar)
Araş. Gör. Başvuru Dosyalarının Muhafaza Edilmesi
Kabul Olan Araş. Gör. Başvuruları Dosyalarının Pers. D. Gönderilmesi
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Personel Dairesi Bşk.
Araş. Gör. Başvuru Dosyası Kopyası (Kabul Olanlar)
Kabul Olmayan Araş. Gör. Başvuru Dosyaları Muhafaza Edildi
Kabul Olan Araş. Gör. Başvuru Dosyaları Pers. D. Gönderildi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi Geldi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Personel Dairesi Bşk.
Araş. Gör. Kadro Başvurusu için Son Tarihi Bilgisinin Alınması
Araş. Gör.
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisinin Adaylara İletilmesi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi Adaylara İletildi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. İşe Başladı
Araş. Gör. Atama Yazısı Alındı
Araş. Gör. Atama Yazısı
Personel Dairesi Bşk.
Araş. Gör. Göreve Başlama Yazısının Hazırlanması
Araş. Gör. Göreve Başlama Yazısı
Enstitü Müd.
Araş. Gör. Göreve Başlama Yazısının İmzalatılması
Araş. Gör. Göreve Başlama Yazısı
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Araş. Gör. Göreve Başlama Yazısının Gönderilmesi
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Personel Dairesi Bşk.
Araş. Gör. Göreve Başlama Yazısı Gönderildi

Figure 86. Araş. Gör. Alımı - Böl. Sekreteri (RA Recruitment - Dept. Secretary) (continued)



Interacts
Carries out
Roles
Öğr.El. Alımı Değerlendirme Jürisi
Bölüm Sekreteri
Araş. Gör. Başvuru Dosyası
Araş. Gör. Alımı Mülakat Tar.
Araş. Gör.
Araş.Gör. Alımı Mülakatının Yapılması
Araş.Gör. Alımı Mülakatının Sonuçlandırılması
Bölüm Sekreteri
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
Araş.Gör. Alımı Kararının İletilmesi
Araş. Gör. Başvuru Dosyası
Araş.Gör. Alımı Kararı İletildi

Figure 87. Araş. Gör. Alımı - Öğr.El. Alımı Değerlendirme Jürisi (Research Asst. Recruitment - Staff Recruitment Evaluation Committee)



Interacts
Carries out
Roles
Öğretim Üyesi
ABD. Bşk.
Öğr. El. Alımı Değ. Jürisine Katılım Çağrısı Geldi
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı (Mül. Tar/Yeri)
Çağrının Alınması ve Cevaplanması
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı Cevabı
Çağrı Alındı ve Cevaplandı
Öğr. El. bu aşamadan sonra Öğr.El. Alımı Değerlendirme Jürisinin bir elemanı olarak sürece devam eder.

Figure 88. Araş. Gör. Alımı - Öğretim Üyesi (Research Asst. Recruitment - Faculty Member)

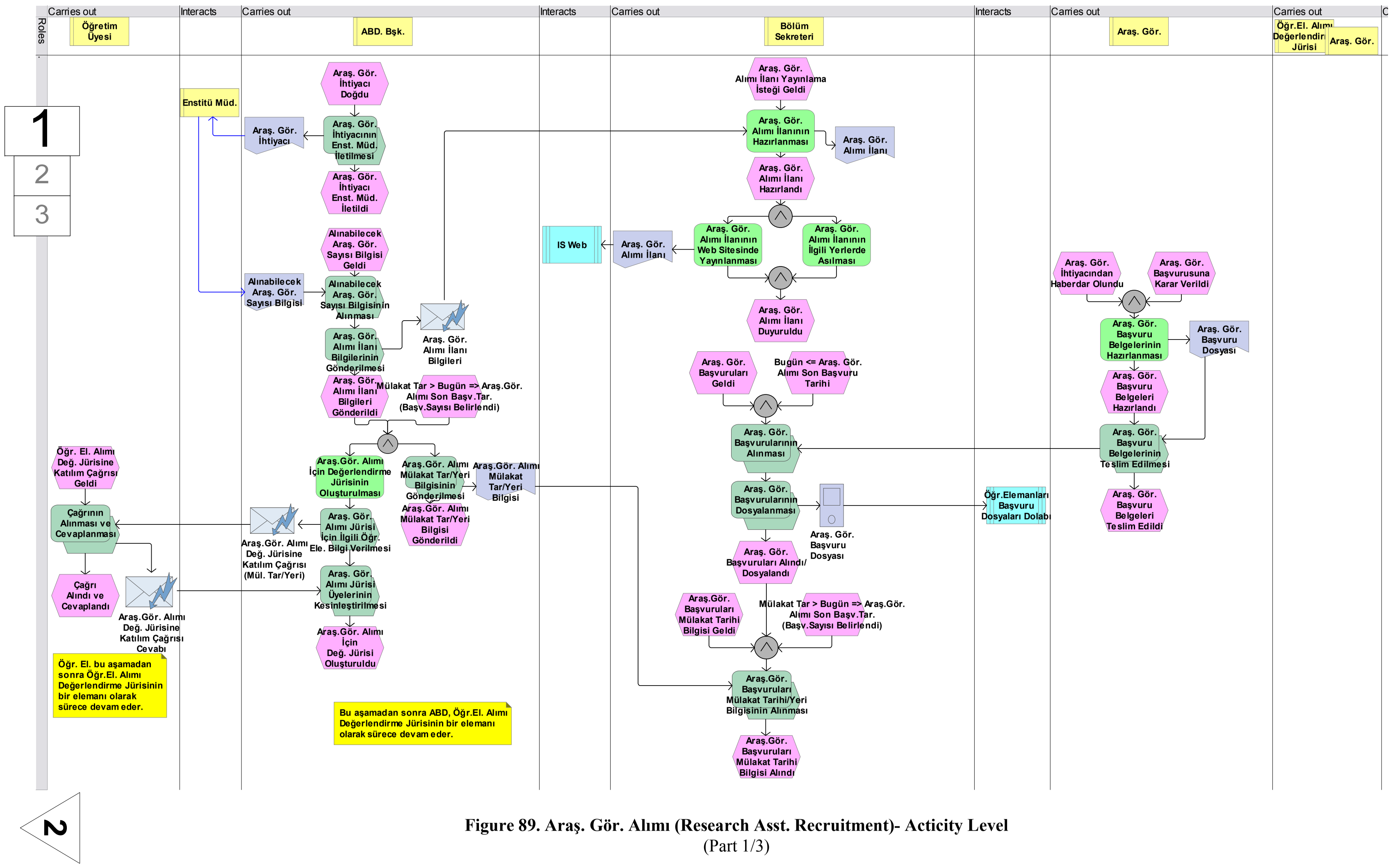
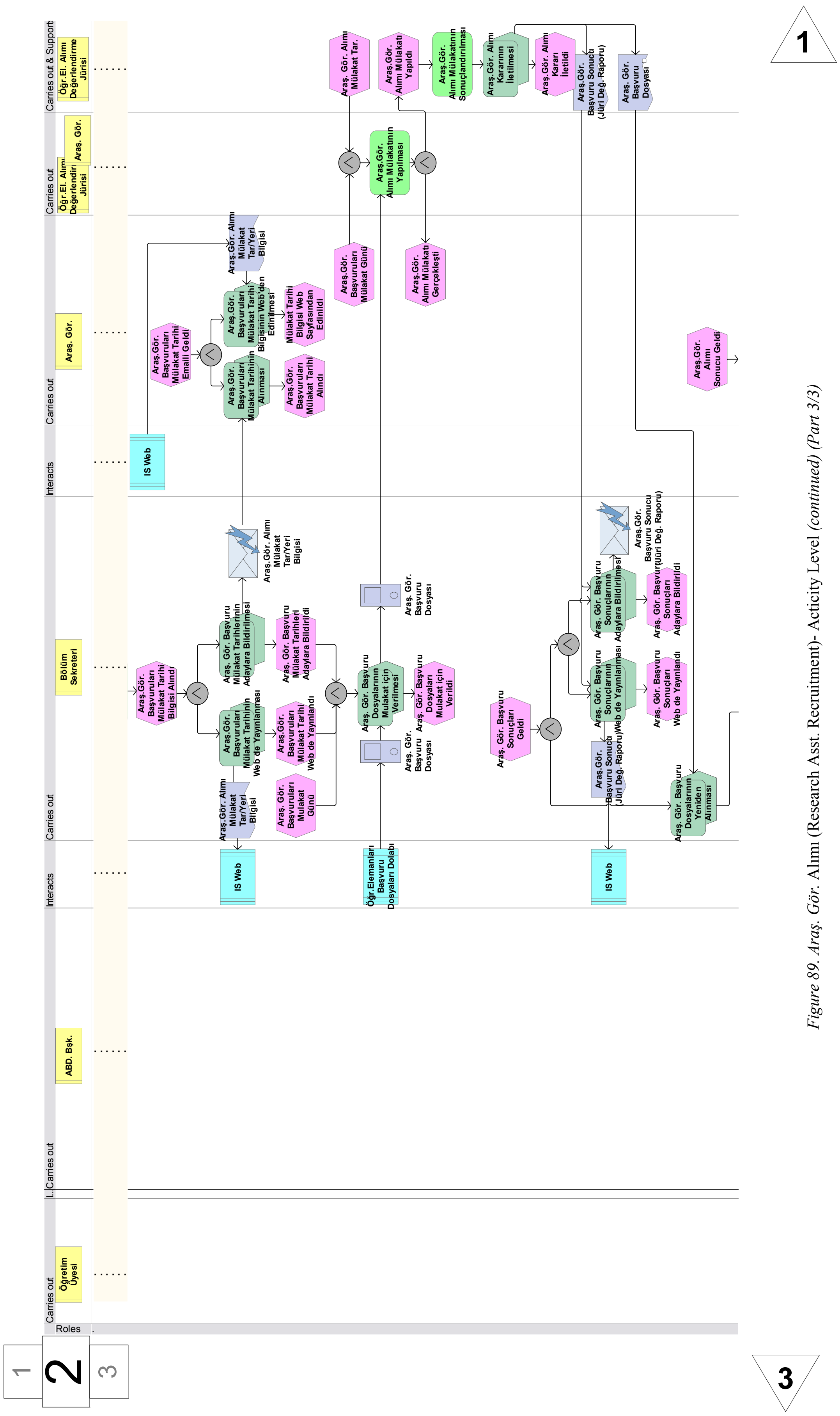

Roles
Carries out
Öğretim Üyesi
Interacts
Enstitü Müd.
Carries out
ABD. Bşk.
Interacts
Carries out
Bölüm Sekreteri
Interacts
Carries out
Araş. Gör.
Carries out
Öğr.El. Alımı Değerlendirme Jürisi
Araş. Gör.
1
2
3
Araş. Gör. İhtiyacı Doğdu
Araş. Gör. İhtiyacının Enst. Müd. İletilmesi
Araş. Gör. İhtiyacı
Araş. Gör. İhtiyacı Enst. Müd. İletildi
Alınabilecek Araş. Gör. Sayısı Bilgisi Geldi
Alınabilecek Araş. Gör. Sayısı Bilgisinin Alınması
Alınabilecek Araş. Gör. Sayısı Bilgisi
Araş. Gör. Alımı İlanı Bilgilerinin Gönderilmesi
Araş. Gör. Alımı İlanı Bilgileri
Araş. Gör. Alımı İlanı Bilgileri Gönderildi
Mülakat Tar > Bugün => Araş.Gör. Alımı Son Başv.Tar. (Başv.Sayısı Belirlendi)
Araş.Gör. Alımı İçin Değerlendirme Jürisinin Oluşturulması
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisinin Gönderilmesi
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi Gönderildi
Araş. Gör. Alımı Jürisi İçin İlgili Öğr. Ele. Bilgi Verilmesi
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı (Mül. Tar/Yeri)
Araş. Gör. Alımı Jürisi Üyelerinin Kesinleştirilmesi
Araş.Gör. Alımı İçin Değ. Jürisi Oluşturuldu
Bu aşamadan sonra ABD, Öğr.El. Alımı Değerlendirme Jürisinin bir elemanı olarak sürece devam eder.
Öğr. El. Alımı Değ. Jürisine Katılım Çağrısı Geldi
Çağrının Alınması ve Cevaplanması
Çağrı Alındı ve Cevaplandı
Araş.Gör. Alımı Değ. Jürisine Katılım Çağrısı Cevabı
Öğr. El. bu aşamadan sonra Öğr.El. Alımı Değerlendirme Jürisinin bir elemanı olarak sürece devam eder.
IS Web
Araş. Gör. Alımı İlanı Yayınlama İsteği Geldi
Araş. Gör. Alımı İlanının Hazırlanması
Araş. Gör. Alımı İlanı
Araş. Gör. Alımı İlanı Hazırlandı
Araş. Gör. Alımı İlanının Web Sitesinde Yayınlanması
Araş. Gör. Alımı İlanının İlgili Yerlerde Asılması
Araş. Gör. Alımı İlanı
Araş. Gör. Alımı İlanı Duyuruldu
Araş. Gör. Başvuruları Geldi
Bugün <= Araş. Gör. Alımı Son Başvuru Tarihi
Araş. Gör. Başvurularının Alınması
Araş. Gör. Başvurularının Dosyalanması
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuruları Alındı/ Dosyalandı
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisi Geldi
Mülakat Tar > Bugün => Araş.Gör. Alımı Son Başv.Tar. (Başv.Sayısı Belirlendi)
Araş.Gör. Başvuruları Mülakat Tarihi/Yeri Bilgisinin Alınması
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisi Alındı
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. İhtiyacından Haberdar Olundu
Araş. Gör. Başvurusuna Karar Verildi
Araş. Gör. Başvuru Belgelerinin Hazırlanması
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Belgeleri Hazırlandı
Araş. Gör. Başvuru Belgelerinin Teslim Edilmesi
Araş. Gör. Başvuru Belgeleri Teslim Edildi
2

Figure 89. Araş. Gör. Alımı (Research Asst. Recruitment)- Acticity Level
(Part 1/3)



Roles
Carries out
Öğretim Üyesi
Carries out
ABD. Bşk.
Interacts
Carries out
Bölüm Sekreteri
Interacts
Carries out
Araş. Gör.
Carries out
Öğr.El. Alımı Değerlendirme Jürisi
Araş. Gör.
Carries out & Supports
Öğr.El. Alımı Değerlendirme Jürisi
IS Web
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisi Alındı
Araş.Gör. Başvuruları Mülakat Tarihinin Web'de Yayınlanması
Araş. Gör. Başvuru Mülakat Tarihlerinin Adaylara Bildirilmesi
Araş.Gör. Başvuruları Mülakat Tarihi Web'de Yayınlandı
Araş. Gör. Başvuruları Mülakat Tarihleri Adaylara Bildirildi
Araş. Gör. Başvuruları Mulakat Günü
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş. Gör. Başvuru Dosyalarının Mulakat için Verilmesi
Araş. Gör. Başvuru Dosyaları Mulakat için Verildi
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Sonuçları Geldi
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
Araş. Gör. Başvuru Sonuçlarının Web de Yayınlanması
Araş. Gör. Başvuru Sonuçlarının Adaylara Bildirilmesi
Araş. Gör. Başvuru Sonuçları Web de Yayınlandı
Araş. Gör. Başvuru Sonuçları Adaylara Bildirildi
Araş. Gör. Başvuru Dosyalarının Yeniden Alınması
Araş.Gör. Başvuruları Mülakat Tarihi Emaili Geldi
Araş.Gör. Başvuruları Mülakat Tarihinin Alınması
Araş.Gör. Başvuruları Mülakat Tarihi Bilgisinin Web'den Edinilmesi
Araş.Gör. Başvuruları Mülakat Tarihi Alındı
Mülakat Tarihi Bilgisi Web Sayfasından Edinildi
Araş.Gör. Başvuruları Mülakat Günü
Araş.Gör. Alımı Mülakatı Gerçekleşti
Araş.Gör. Alımı Sonucu Geldi
Araş.Gör. Alımı Mülakatının Yapılması
Araş. Gör. Alımı Mülakat Tar.
Araş.Gör. Alımı Mülakatı Yapıldı
Araş.Gör. Alımı Mülakatının Sonuçlandırılması
Araş.Gör. Alımı Kararının İletilmesi
Araş.Gör. Alımı Kararı İletildi
Araş. Gör. Başvuru Dosyası
1
2
3

Figure 89. Araş. Gör. Alımı (Research Asst. Recruitment)- Acticity Level *(continued) (Part 3/3)*

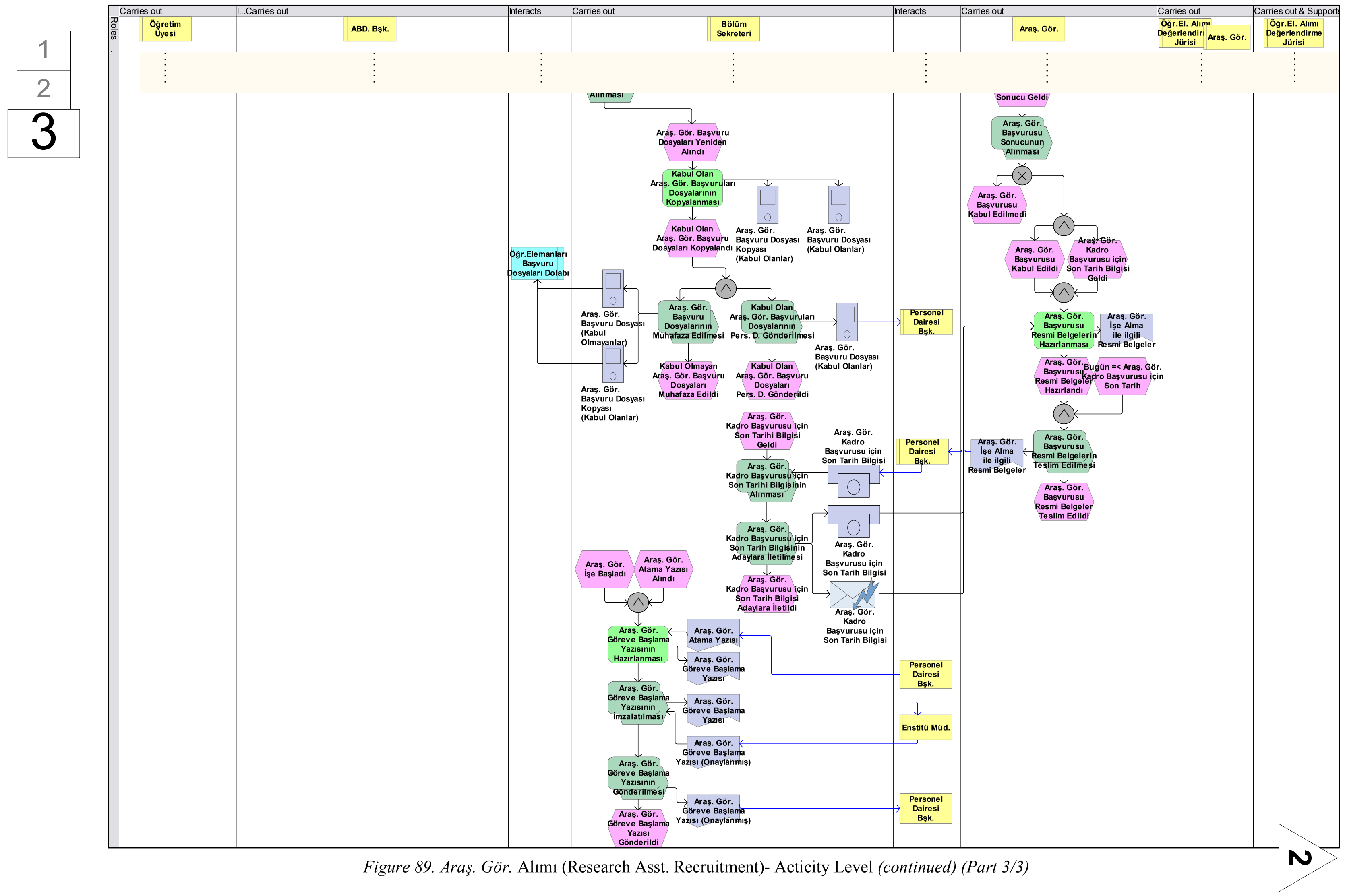

1
2
3
Roles
Carries out
Öğretim Üyesi
Carries out
ABD. Bşk.
Interacts
Carries out
Bölüm Sekreteri
Interacts
Carries out
Araş. Gör.
Carries out
Öğr.El. Alımı Değerlendirme Jürisi
Araş. Gör.
Carries out & Supports
Öğr.El. Alımı Değerlendirme Jürisi
Alınması
Araş. Gör. Başvuru Dosyaları Yeniden Alındı
Kabul Olan Araş. Gör. Başvuruları Dosyalarının Kopyalanması
Kabul Olan Araş. Gör. Başvuru Dosyaları Kopyalandı
Araş. Gör. Başvuru Dosyası Kopyası (Kabul Olanlar)
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. Başvuru Dosyası (Kabul Olmayanlar)
Araş. Gör. Başvuru Dosyası Kopyası (Kabul Olanlar)
Araş. Gör. Başvuru Dosyalarının Muhafaza Edilmesi
Kabul Olmayan Araş. Gör. Başvuru Dosyaları Muhafaza Edildi
Kabul Olan Araş. Gör. Başvuruları Dosyalarının Pers. D. Gönderilmesi
Kabul Olan Araş. Gör. Başvuru Dosyaları Pers. D. Gönderildi
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Personel Dairesi Bşk.
Sonucu Geldi
Araş. Gör. Başvurusu Sonucunun Alınması
Araş. Gör. Başvurusu Kabul Edilmedi
Araş. Gör. Başvurusu Kabul Edildi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi Geldi
Araş. Gör. Başvurusu Resmi Belgelerin Hazırlanması
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Araş. Gör. Başvurusu Resmi Belgeler Hazırlandı
Bugün =< Araş. Gör. Kadro Başvurusu için Son Tarih
Araş. Gör. Başvurusu Resmi Belgelerin Teslim Edilmesi
Araş. Gör. Başvurusu Resmi Belgeler Teslim Edildi
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Personel Dairesi Bşk.
Araş. Gör. Kadro Başvurusu için Son Tarihi Bilgisi Geldi
Araş. Gör. Kadro Başvurusu için Son Tarihi Bilgisinin Alınması
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisinin Adaylara İletilmesi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi Adaylara İletildi
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. İşe Başladı
Araş. Gör. Atama Yazısı Alındı
Araş. Gör. Göreve Başlama Yazısının Hazırlanması
Araş. Gör. Atama Yazısı
Araş. Gör. Göreve Başlama Yazısı
Personel Dairesi Bşk.
Araş. Gör. Göreve Başlama Yazısının İmzalatılması
Araş. Gör. Göreve Başlama Yazısı
Enstitü Müd.
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Araş. Gör. Göreve Başlama Yazısının Gönderilmesi
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Personel Dairesi Bşk.
Araş. Gör. Göreve Başlama Yazısı Gönderildi
2

Figure 89. Araş. Gör. Alımı (Research Asst. Recruitment)- Acticity Level *(continued) (Part 3/3)*

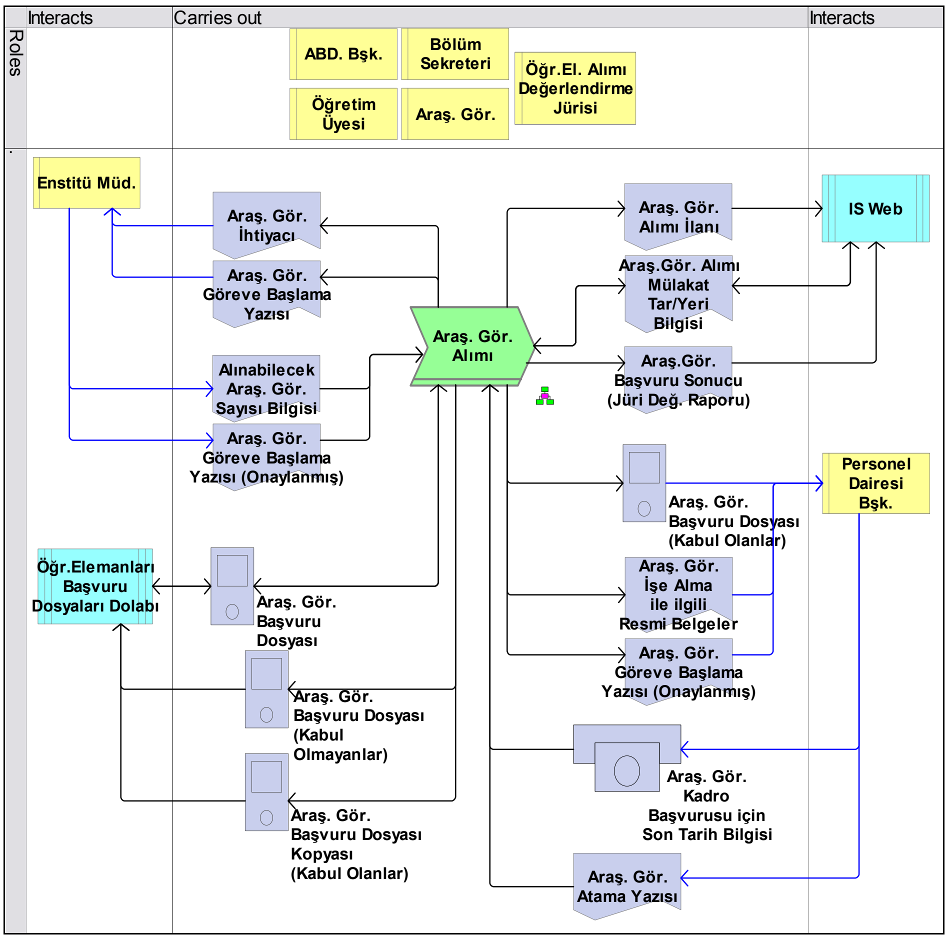

Roles
Interacts
Carries out
Interacts
ABD. Bşk.
Bölüm Sekreteri
Öğretim Üyesi
Araş. Gör.
Öğr.El. Alımı Değerlendirme Jürisi
Enstitü Müd.
Araş. Gör. İhtiyacı
Araş. Gör. Göreve Başlama Yazısı
Alınabilecek Araş. Gör. Sayısı Bilgisi
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Araş. Gör. Alımı
Araş. Gör. Alımı İlanı
Araş.Gör. Alımı Mülakat Tar/Yeri Bilgisi
Araş.Gör. Başvuru Sonucu (Jüri Değ. Raporu)
IS Web
Personel Dairesi Bşk.
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
Öğr.Elemanları Başvuru Dosyaları Dolabı
Araş. Gör. Başvuru Dosyası
Araş. Gör. Başvuru Dosyası (Kabul Olmayanlar)
Araş. Gör. Başvuru Dosyası Kopyası (Kabul Olanlar)
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Araş. Gör. Atama Yazısı

Figure 90. Araş. Gör. Alımı (Research Asst. Recruitment) - Process Level

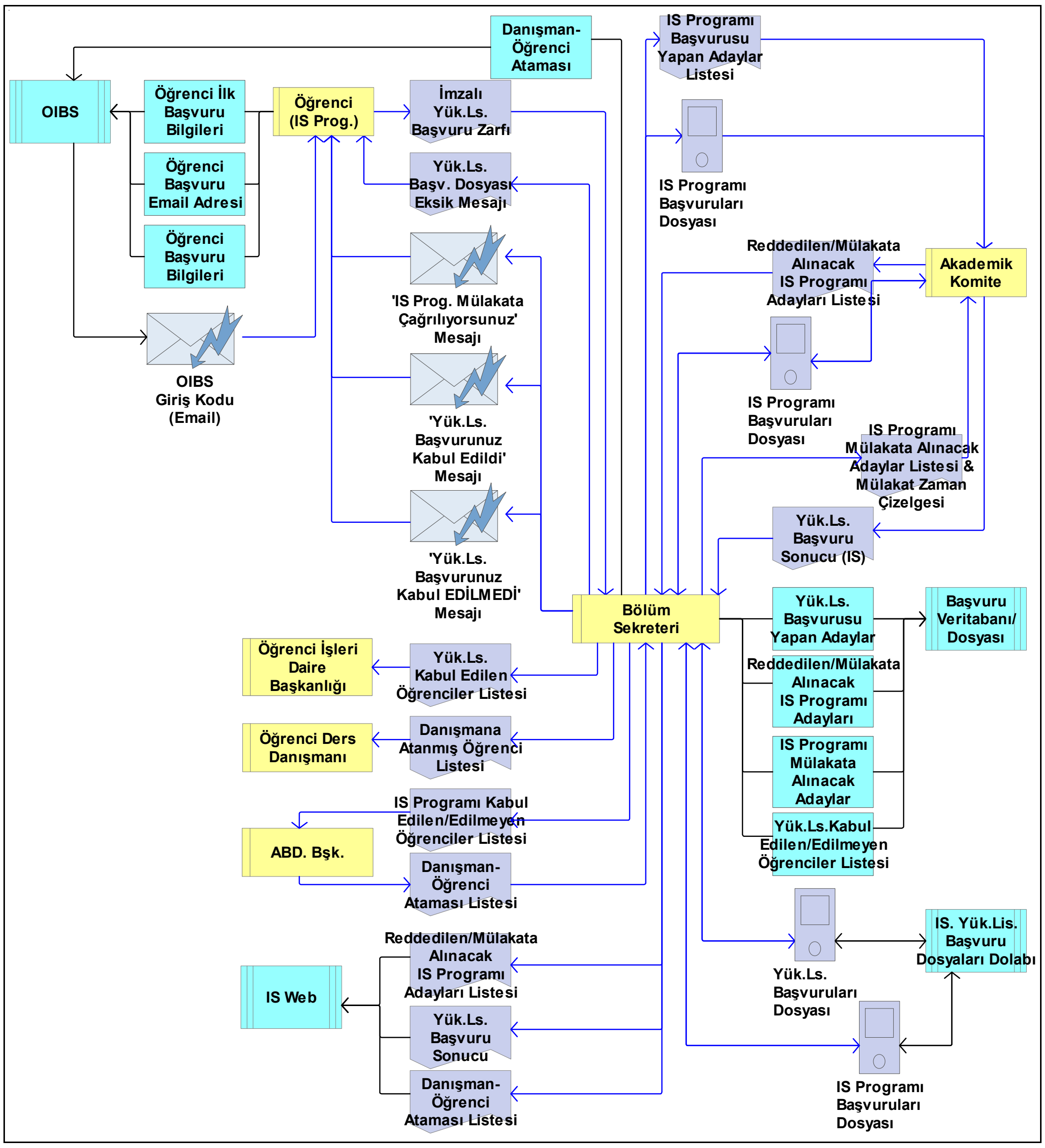

Danışman-Öğrenci Ataması
IS Programı Başvurusu Yapan Adaylar Listesi
OIBS
Öğrenci İlk Başvuru Bilgileri
Öğrenci (IS Prog.)
İmzalı Yük.Ls. Başvuru Zarfı
IS Programı Başvuruları Dosyası
Öğrenci Başvuru Email Adresi
Yük.Ls. Başv. Dosyası Eksik Mesajı
Öğrenci Başvuru Bilgileri
Reddedilen/Mülakata Alınacak IS Programı Adayları Listesi
Akademik Komite
'IS Prog. Mülakata Çağrılıyorsunuz' Mesajı
OIBS Giriş Kodu (Email)
IS Programı Başvuruları Dosyası
'Yük.Ls. Başvurunuz Kabul Edildi' Mesajı
IS Programı Mülakata Alınacak Adaylar Listesi & Mülakat Zaman Çizelgesi
'Yük.Ls. Başvurunuz Kabul EDİLMEDİ' Mesajı
Yük.Ls. Başvuru Sonucu (IS)
Bölüm Sekreteri
Yük.Ls. Başvurusu Yapan Adaylar
Başvuru Veritabanı/ Dosyası
Öğrenci İşleri Daire Başkanlığı
Yük.Ls. Kabul Edilen Öğrenciler Listesi
Reddedilen/Mülakata Alınacak IS Programı Adayları
Öğrenci Ders Danışmanı
Danışmana Atanmış Öğrenci Listesi
IS Programı Mülakata Alınacak Adaylar
IS Programı Kabul Edilen/Edilmeyen Öğrenciler Listesi
Yük.Ls.Kabul Edilen/Edilmeyen Öğrenciler Listesi
ABD. Bşk.
Danışman-Öğrenci Ataması Listesi
IS. Yük.Lis. Başvuru Dosyaları Dolabı
Reddedilen/Mülakata Alınacak IS Programı Adayları Listesi
Yük.Ls. Başvuruları Dosyası
IS Web
Yük.Ls. Başvuru Sonucu
IS Programı Başvuruları Dosyası
Danışman-Öğrenci Ataması Listesi

Figure 91. Role Dependency in Student Admissions -- Inf. Sys. Programs

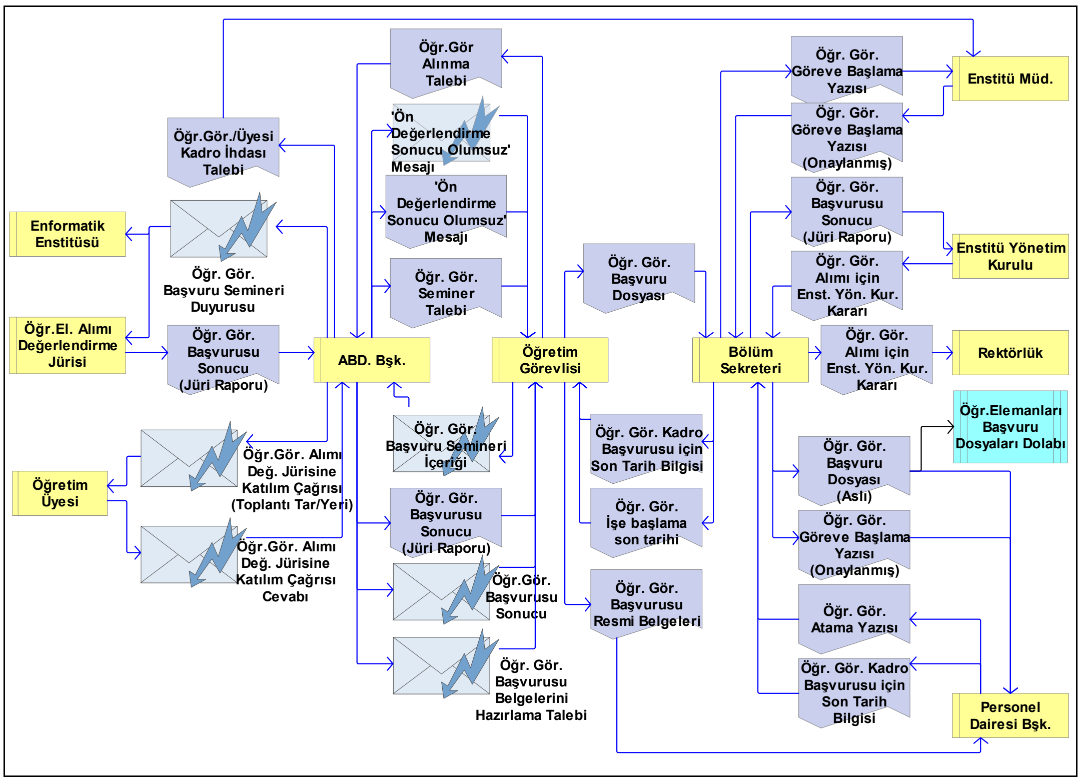
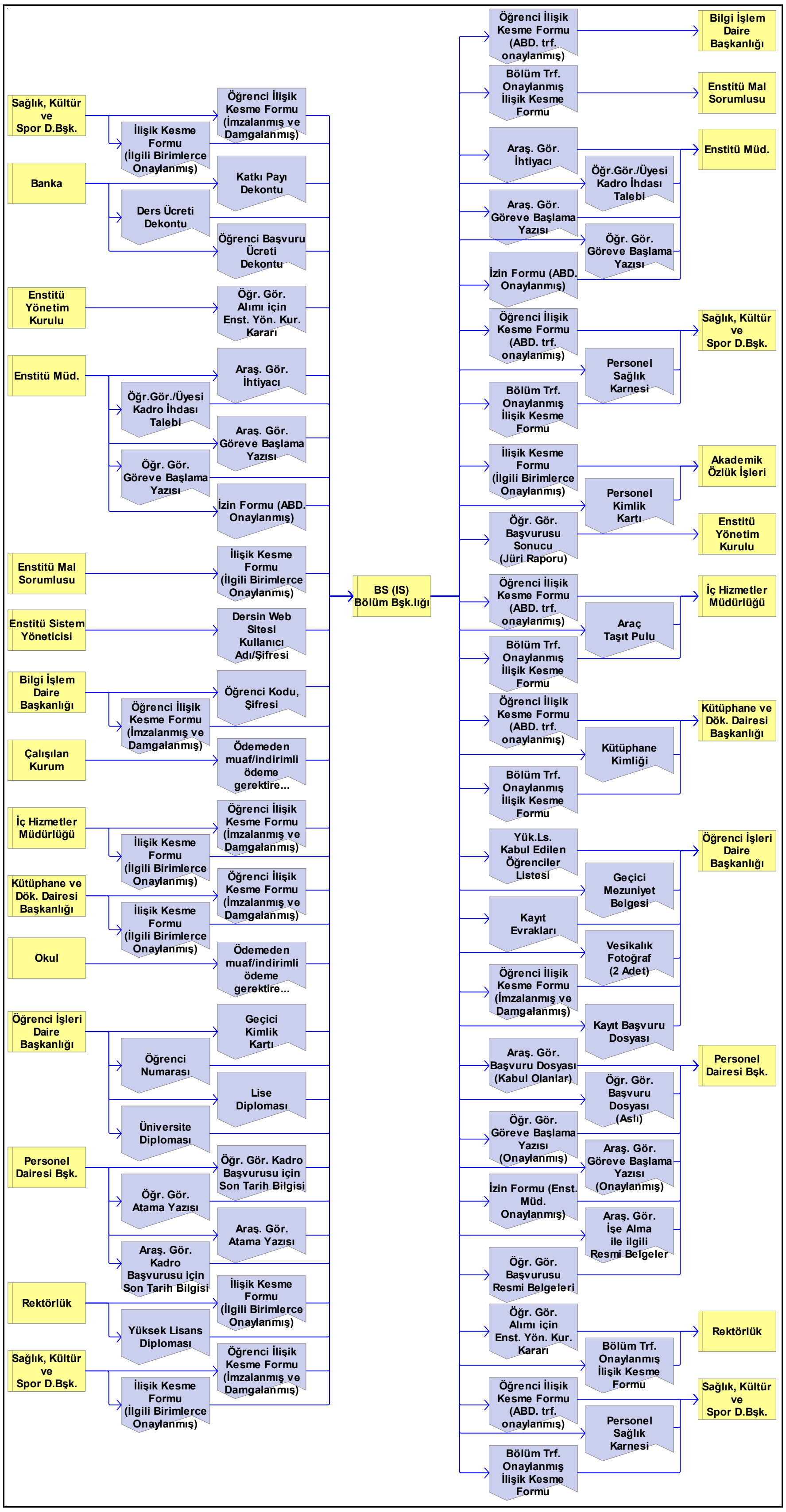


Figure 92. Role Dependency in Instructor Recruitments



Sağlık, Kültür ve Spor D.Bşk.
Banka
Enstitü Yönetim Kurulu
Enstitü Müd.
Enstitü Mal Sorumlusu
Enstitü Sistem Yöneticisi
Bilgi İşlem Daire Başkanlığı
Çalışılan Kurum
İç Hizmetler Müdürlüğü
Kütüphane ve Dök. Dairesi Başkanlığı
Okul
Öğrenci İşleri Daire Başkanlığı
Personel Dairesi Bşk.
Rektörlük
Sağlık, Kültür ve Spor D.Bşk.
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Katkı Payı Dekontu
Ders Ücreti Dekontu
Öğrenci Başvuru Ücreti Dekontu
Öğr. Gör. Alımı için Enst. Yön. Kur. Kararı
Araş. Gör. İhtiyacı
Öğr.Gör./Üyesi Kadro İhdası Talebi
Araş. Gör. Göreve Başlama Yazısı
Öğr. Gör. Göreve Başlama Yazısı
İzin Formu (ABD. Onaylanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Dersin Web Sitesi Kullanıcı Adı/Şifresi
Öğrenci Kodu, Şifresi
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
Ödemeden muaf/indirimli ödeme gerektire...
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Ödemeden muaf/indirimli ödeme gerektire...
Geçici Kimlik Kartı
Öğrenci Numarası
Lise Diploması
Üniversite Diploması
Öğr. Gör. Kadro Başvurusu için Son Tarih Bilgisi
Öğr. Gör. Atama Yazısı
Araş. Gör. Atama Yazısı
Araş. Gör. Kadro Başvurusu için Son Tarih Bilgisi
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Yüksek Lisans Diploması
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
BS (IS) Bölüm Bşk.lığı
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Araş. Gör. İhtiyacı
Öğr.Gör./Üyesi Kadro İhdası Talebi
Araş. Gör. Göreve Başlama Yazısı
Öğr. Gör. Göreve Başlama Yazısı
İzin Formu (ABD. Onaylanmış)
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Personel Sağlık Karnesi
Bölüm Trf. Onaylanmış İlişik Kesme Formu
İlişik Kesme Formu (İlgili Birimlerce Onaylanmış)
Personel Kimlik Kartı
Öğr. Gör. Başvurusu Sonucu (Jüri Raporu)
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Araç Taşıt Pulu
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Kütüphane Kimliği
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Yük.Ls. Kabul Edilen Öğrenciler Listesi
Geçici Mezuniyet Belgesi
Kayıt Evrakları
Vesikalık Fotoğraf (2 Adet)
Öğrenci İlişik Kesme Formu (İmzalanmış ve Damgalanmış)
Kayıt Başvuru Dosyası
Araş. Gör. Başvuru Dosyası (Kabul Olanlar)
Öğr. Gör. Başvuru Dosyası (Aslı)
Öğr. Gör. Göreve Başlama Yazısı (Onaylanmış)
Araş. Gör. Göreve Başlama Yazısı (Onaylanmış)
İzin Formu (Enst. Müd. Onaylanmış)
Araş. Gör. İşe Alma ile ilgili Resmi Belgeler
Öğr. Gör. Başvurusu Resmi Belgeleri
Öğr. Gör. Alımı için Enst. Yön. Kur. Kararı
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Öğrenci İlişik Kesme Formu (ABD. trf. onaylanmış)
Personel Sağlık Karnesi
Bölüm Trf. Onaylanmış İlişik Kesme Formu
Bilgi İşlem Daire Başkanlığı
Enstitü Mal Sorumlusu
Enstitü Müd.
Sağlık, Kültür ve Spor D.Bşk.
Akademik Özlük İşleri
Enstitü Yönetim Kurulu
İç Hizmetler Müdürlüğü
Kütüphane ve Dök. Dairesi Başkanlığı
Öğrenci İşleri Daire Başkanlığı
Personel Dairesi Bşk.
Rektörlük
Sağlık, Kültür ve Spor D.Bşk.

Figure 93. Dependency with Inactive Roles

APPENDIX D: THE EXECUTION PLAN FOR CASE STUDY 2

1. Overview

This document presents the plan for the execution of the Plural method in the Company. The objective is to model specific set of organization's processes via the Plural method. The method and related activities are described in Section 5 of this plan.

The processes that will be covered are given in the scope diagram in Figure 95 (p. 208). The roles that are identified and their relationships are given in Figure 96 (p. 208).

The models to be developed include:

1. Individual role process diagrams for each role and for each process
2. Integrated process diagrams
3. Generated role- dependency and process-dependency diagrams

The project will mainly comprise three phases. Table 24 gives these phases and their schedule.

Table 27. Schedule Summary (Case Study 2)

ID	Phase	Start	Finish	Duration	Effort
1	**The Case Study 2**	29-May-06	31-May-06	24 hrs	60.5 hrs
1.1	Context Definition	29-May-06	29-May-06	6 hrs	12 hrs
1.2	Description and Conflict Resolution	29-May-06	30-May-06	10 hrs	40.5 hrs
1.3	Integration and Change	31-May-06	31-May-06	8 hrs	8 hrs

2. References

1. Turetken, Oktay & Demirors, Onur (2005). Concurrent Software Process Modeling. *Proceedings of European Software Process Improvement and Innovation Conference* (EuroSPI-2005), Budapest, Hungary.

3. Project Organization

The project roles and their relationships is as follows:

The Plural Process Group: Responsible for context definition and handling macro-level process changes. It comprises all participating agents.

The Coordination Team: Facilitates the execution of the method, verifies models, generates new ones, performs training and orientation for the agents, and facilitates meetings.

Development Agents: Perform individual process modeling and conflict resolution. They are also responsible for the maintenance of their models.

Peer Agents: Proxies for development agents and they are responsible to validate individual models they are assigned to.

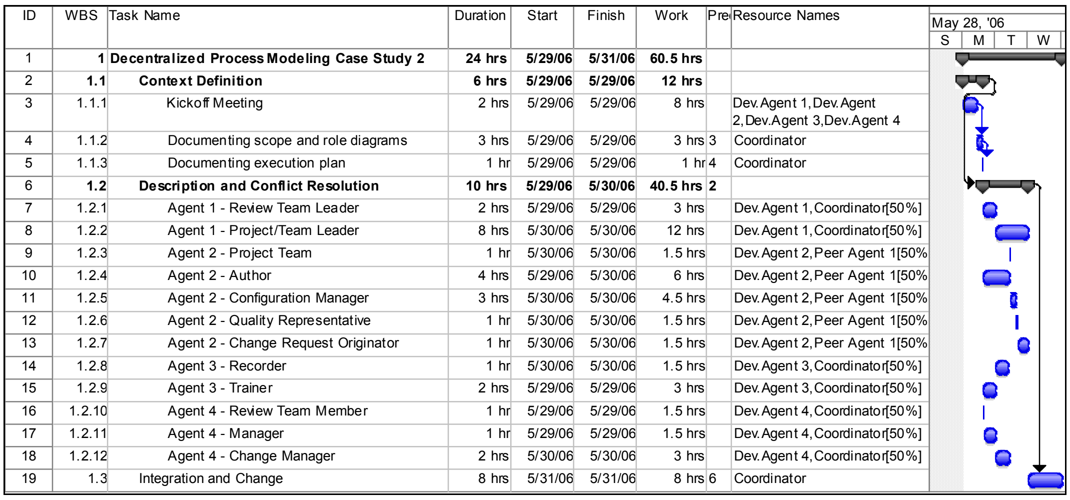
The resources and their assignment to the Plural and process roles are given in Table 25.

Table 28. Project Resources and Their Assignments (Case Study 2)

Resource	**Project Role**	**Mapped Process Role**
Agent 1	Development Agent 1, Peer 1	Review Team Leader
		Project/Team Leader
Agent 2	Development Agent 2, Coordinator 1	Project Team
		Author
		Configuration Manager
		Quality Representative
		Change Request Originator
Agent 3	Development Agent 3	Recorder
		Trainer
Agent 4	Development Agent 4	Review Team Member
		Manager
		Change Manager

4. Work Plan

The schedule for the project and resource assignments is given in Figure 81.



ID	WBS	Task Name	Duration	Start	Finish	Work	Pre	Resource Names
1	1	**Decentralized Process Modeling Case Study 2**	**24 hrs**	**5/29/06**	**5/31/06**	**60.5 hrs**		
2	**1.1**	**Context Definition**	**6 hrs**	**5/29/06**	**5/29/06**	**12 hrs**		
3	1.1.1	Kickoff Meeting	2 hrs	5/29/06	5/29/06	8 hrs		Dev.Agent 1,Dev.Agent 2,Dev.Agent 3,Dev.Agent 4
4	1.1.2	Documenting scope and role diagrams	3 hrs	5/29/06	5/29/06	3 hrs	3	Coordinator
5	1.1.3	Documenting execution plan	1 hr	5/29/06	5/29/06	1 hr	4	Coordinator
6	**1.2**	**Description and Conflict Resolution**	**10 hrs**	**5/29/06**	**5/30/06**	**40.5 hrs**	**2**	
7	1.2.1	Agent 1 - Review Team Leader	2 hrs	5/29/06	5/29/06	3 hrs		Dev.Agent 1,Coordinator[50%]
8	1.2.2	Agent 1 - Project/Team Leader	8 hrs	5/30/06	5/30/06	12 hrs		Dev.Agent 1,Coordinator[50%]
9	1.2.3	Agent 2 - Project Team	1 hr	5/30/06	5/30/06	1.5 hrs		Dev.Agent 2,Peer Agent 1[50%
10	1.2.4	Agent 2 - Author	4 hrs	5/29/06	5/30/06	6 hrs		Dev.Agent 2,Peer Agent 1[50%
11	1.2.5	Agent 2 - Configuration Manager	3 hrs	5/30/06	5/30/06	4.5 hrs		Dev.Agent 2,Peer Agent 1[50%
12	1.2.6	Agent 2 - Quality Representative	1 hr	5/30/06	5/30/06	1.5 hrs		Dev.Agent 2,Peer Agent 1[50%
13	1.2.7	Agent 2 - Change Request Originator	1 hr	5/30/06	5/30/06	1.5 hrs		Dev.Agent 2,Peer Agent 1[50%
14	1.2.8	Agent 3 - Recorder	1 hr	5/30/06	5/30/06	1.5 hrs		Dev.Agent 3,Coordinator[50%]
15	1.2.9	Agent 3 - Trainer	2 hrs	5/29/06	5/29/06	3 hrs		Dev.Agent 3,Coordinator[50%]
16	1.2.10	Agent 4 - Review Team Member	1 hr	5/29/06	5/29/06	1.5 hrs		Dev.Agent 4,Coordinator[50%]
17	1.2.11	Agent 4 - Manager	1 hr	5/29/06	5/29/06	1.5 hrs		Dev.Agent 4,Coordinator[50%]
18	1.2.12	Agent 4 - Change Manager	2 hrs	5/30/06	5/30/06	3 hrs		Dev.Agent 4,Coordinator[50%]
19	1.3	Integration and Change	8 hrs	5/31/06	5/31/06	8 hrs	6	Coordinator

Figure 94. Project Schedule (Case Study 2)

5. Technical Process

The technical process to be followed is governed by the Plural method and described in the study [1] given as a reference in Section 2 of this plan.

6. Supporting Plans

6.1 Modeling platform

The modeling will be performed in agents' client PCs via a modeling toolset. The toolset is accessed via the internet browser (MS Internet Explorer 6 or above) through the following IP address:

http://144.122.98.186/webdesigner

6.2 Backup

The configuration manager (Coordinator 1) is responsible to backup the process-base every working day at 18:00 with the following naming convention:

CaseStudy1_vXX.mdb - XX representing the version number incremented each day.

Backup files are stored in a separate server which can be accessed through the following IP address: ftp://144.122.98.186/casestudy1

6.3 Directory Structure and Naming Convention for Diagrams

Directory structure for the models is given below:

- *Main Directory*
  - *Context*: Scope and Role diagrams
  - *Role Process Diagrams*
    - *<Role Name>*: Individual role process diagrams for each role and for each process
  - *Generated Diagrams*
    - *Integrated Process Diagrams:* Activity and process level process diagrams
    - *Process Dependency Diagrams*
    - *Role Dependency Diagrams*

Naming Conventions:

Diagram Type	→	Naming
Scope Diagram	→	scope diagram
Role Diagram	→	role diagram
Individual Role Process Diagram	→	<role name>-<process name>
Activity Level Process Diagram	→	<process name>-activity level
Process Level Process Diagram	→	<process name>-process level
Role Dependency Diagram	→	Role dependency - <process names>/<role names>/global
Process Dependency Diagram	→	Process dependency - <process names>/global

6.4 Training / Orientation

Coordinators may organize training or orientation sessions for development agents if required. Sessions can be scheduled just prior to the modeling so that the first modeling practice can be performed in pairs (development agent and coordinator).

6.5 Tracking Effort

Each agent will track individual time for each work activity depicted in the work plan. Values can be entered into the web-based effort tracking tool accesses through the following IP address:

http://144.122.98.186/case2/

APPENDIX E: DIAGRAMS IN CASE STUDY 2

This section of the thesis presents the diagrams developed in the second case study. Table 29 presents the list of all diagrams that are developed or generated.

Table 29. List of Diagrams in Case Study 2

ID	Diagram Name	Fig. No.	Diagram Type
1	**CONTEXT**		
1.1	Scope Diagram	Figure 95	Scope Diagram
1.2	Role Diagram	Figure 96	Role Diagram
2	**INDIVIDUAL DIAGRAMS**		
2.1	**Author**		
2.1.1	Manage Change	Figure 97	Individual role-process
2.1.1.1	Implement Change Request	Figure 98	Individual role-process
2.1.2	Review □	Figure 99	Individual role-process
2.1.2.1	Initiate Review	Figure 100	Individual role-process
2.1.2.2	Complete Review	Figure 101	Individual role-process
2.2	**Change Manager**		
2.2.1	Manage Change	Figure 102	Individual role-process
2.2.1.2	Evaluate Change Request	Figure 103	Individual role-process
2.2.1.3	Close Change Request	Figure 104	Individual role-process
2.3	**Configuration Manager**		
2.3.1	Manage Configuration (Configuration Manager)		Individual role-process
2.3.1.1	Initiate Project (Configuration Manager)		Individual role-process
2.3.1.2	Place CI under Configuration Control		Individual role-process
2.3.1.3	Close Project (Configuration Manager)		Individual role-process
2.3.1.4	Configuration Items		Information item
2.3.2	Manage Project (Configuration Manager)		Individual role-process
2.3.2.1	Perform Progress Meeting (Conf. Mng.)		Individual role-process
2.3.2.2	Perform Post-Project Meeting (Conf. Mng.)		Individual role-process
2.4	**Change Request Originator**		
2.4.1	Manage Change (Change Request Originator)		Individual role-process
2.4.1.1	Initiate Change Request		Individual role-process
2.4.1.2	Receive and Assess CR Result		Individual role-process
2.5	**Manager**		
2.5.1	Manage Project (Manager)		Individual role-process
2.5.1.1	Allocate Resources and Form Project Team		Individual role-process
2.5.1.2	Approve Project Plan and Project Cost-Budget		Individual role-process
2.5.1.3	Perform Post-Project Meeting (Manager)		Individual role-process
2.6	**Project Team**		
2.6.1	Manager Change (Project Team)		Individual role-process
2.6.1.1	Investigate Change Request		Individual role-process
2.6.2	Manage Configuration (Project Team)		Individual role-process
2.6.2.1	Identify Configuration Items		Individual role-process
2.6.3	Manage Project (Project Team)		Individual role-process

Table 29. List of Diagrams in Case Study 2 (continued)

ID	Diagram Name	Fig. No.	Diagram Type
2.6.3.1	Define the quality objectives for the project		Individual role-process
2.6.3.2	Perform Progress Meeting (Project Team)		Individual role-process
2.6.3.3	Perform post-project meeting (Project Team)		Individual role-process
2.7	**Project/Team Leader**		
2.7.1	Manage Project (Project/Team Leader)		Individual role-process
2.7.1.1	Start-up Project		Individual role-process
2.7.1.2	Execute Project		Individual role-process
2.7.1.3	Perform Project Progress Meeting		Individual role-process
2.7.1.4	Initiate Resource Purchase		Individual role-process
2.7.1.5	Update Plan		Individual role-process
2.7.1.6	Control Customer Supplied Products		Individual role-process
2.7.1.7	Handle Deliverables		Individual role-process
2.7.1.8	Control Nonconforming Products		Individual role-process
2.7.1.9	Monitor Review Initiation		Individual role-process
2.7.1.10	Monitor Review Results		Individual role-process
2.7.1.11	Monitor Rework after Review		Individual role-process
2.7.1.12	Close Project		Individual role-process
2.7.1.13	Perform Post Project Meeting		Individual role-process
2.8	**Quality Representative**		
2.8.1	Manage Configuration (Quality Representative)		Individual role-process
2.8.1.1	Identify Quality Configuration Items		Individual role-process
2.8.2.	Manage Project (Quality Representative)		Individual role-process
2.8.2.1	Plan Quality		Individual role-process
2.8.2.2	Perform Progress Meeting (Quality Rep.)		Individual role-process
2.8.2.3	Initiate Internal Quality Audit		Individual role-process
2.8.2.4	Perform post-project meeting (Quality Rep.)		Individual role-process
2.9	**Recorder**		
2.9.1	Review (Recorder)	Figure 105	Individual role-process
2.9.1.1	Accept Review Meeting Attendance Request	Figure 106	Individual role-process
2.9.1.2	Prepare Review Meeting Log	Figure 107	Individual role-process
2.10	**Review Team Leader**		
2.10.1	Review (Review Team Leader)	Figure 108	Individual role-process
2.10.1.1	Prepare Review	Figure 109	Individual role-process
2.10.1.2	Perform Internal Review Meeting	Figure 110	Individual role-process
2.10.1.3	Close Review Meeting	Figure 111	Individual role-process
2.10.1.2	Close Review	Figure 112	Individual role-process
2.11	**Review Team Member**		
2.11.1	Review (Review Team Member)	Figure 113	Individual role-process
2.11.1.1	Individual Checking	Figure 114	Individual role-process
2.11.1.2	Attend Internal Review Meeting	Figure 115	Individual role-process
2.12	**Trainer**		
2.12.1	Provide Training (Trainer)		Individual role-process
2.12.1.1	Prepare Course Materials		Individual role-process
2.12.1.2	Execute Course		Individual role-process
2.12.1.3	Close Course		Individual role-process
2.12.1.4	Attendee Training Materials Folder		Information item
2.13	Training Coordinator		
2.13.1	Provide Training (Training Coordinator)		Individual role-process

Table 29. List of Diagrams in Case Study 2 (continued)

ID	Diagram Name	Fig. No.	Diagram Type
3	**GENERATED DIAGRAMS**		
3.1	**Generated Process Diagrams**		
3.1.1	Manage Change		
3.1.1.1	Manage Change-Activity Level		Activity level process
3.1.1.2	Manage Change-Operation Level		Operation level process
3.1.1.3	Manage Change-Process Level		Process level process
3.1.2	Manage Configuration		
3.1.2.1	Manage Configuration-Activity Level		Activity level process
3.1.2.2	Manage Configuration-Operation Level		Operation level process
3.1.2.3	Manage Configuration-Process Level		Process level process
3.1.3	Manage Project		
3.1.3.1	Start-up Project-Activity Level		Activity level process
3.1.3.2	Execute Project-Activity Level		Activity level process
3.1.3.3	Close Project-Activity Level		Activity level process
3.1.3.4	Manage Project-Operation Level		Operation level process
3.1.3.5	Manage Project-Process Level		Process level process
3.1.4	Provide Training		
3.1.4.1	Provide Training - Activity Level		Activity level process
3.1.4.2	Provide Training - Operation Level		Operation level process
3.1.4.3	Provide Training - Process Level		Process level process
3.1.5	Review		
3.1.5.1	Review-Activity Level	Figure 116	Activity level process
3.1.5.2	Review-Operation Level	Figure 117	Operation level process
3.1.5.3	Review-Process Level	Figure 118	Process level process
3.2	**Role Dependency Diagrams**		
3.2.1	Role Dependency - Change Manager & Conf. Manager	Figure 119	Role Dependency
3.2.2	Role Dependency - Review Team Leader & Conf. Manager	Figure 120	Role Dependency
3.2.3	Role Dependency - Manage Change	Figure 121	Role Dependency
3.2.4	Role Dependency - Review	Figure 122	Role Dependency
3.2.5	Role Dependency-Review (Aggregated on Review Team)	Figure 123	Role Dependency
3.3	**Process Dependency Diagrams**		
3.3.1	Process Dependency Diagram- Conf. Manager & Change Mng.	Figure 124	Process Dependency
3.3.2	Process Dependency Diagram-Global	Figure 125	Process Dependency

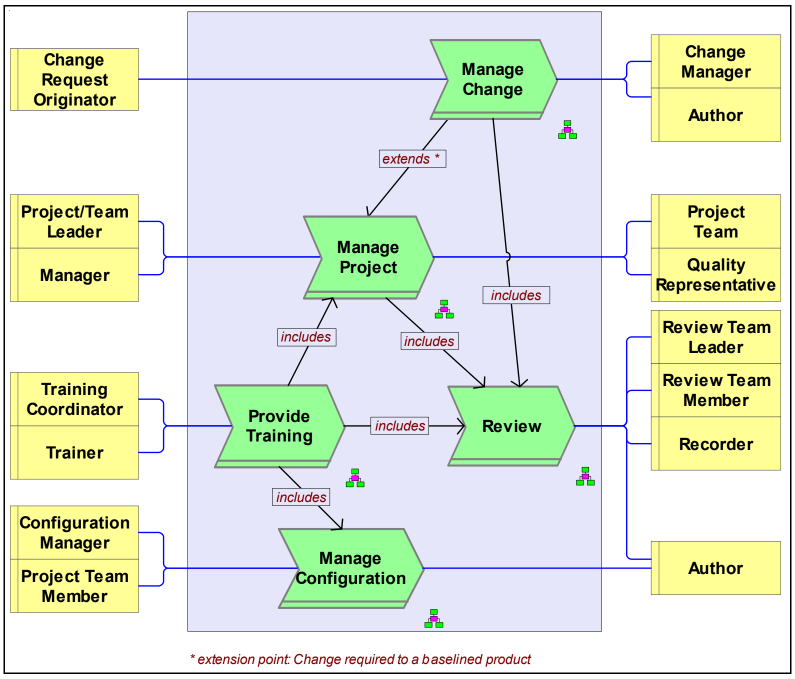


Figure 95. Scope Diagram

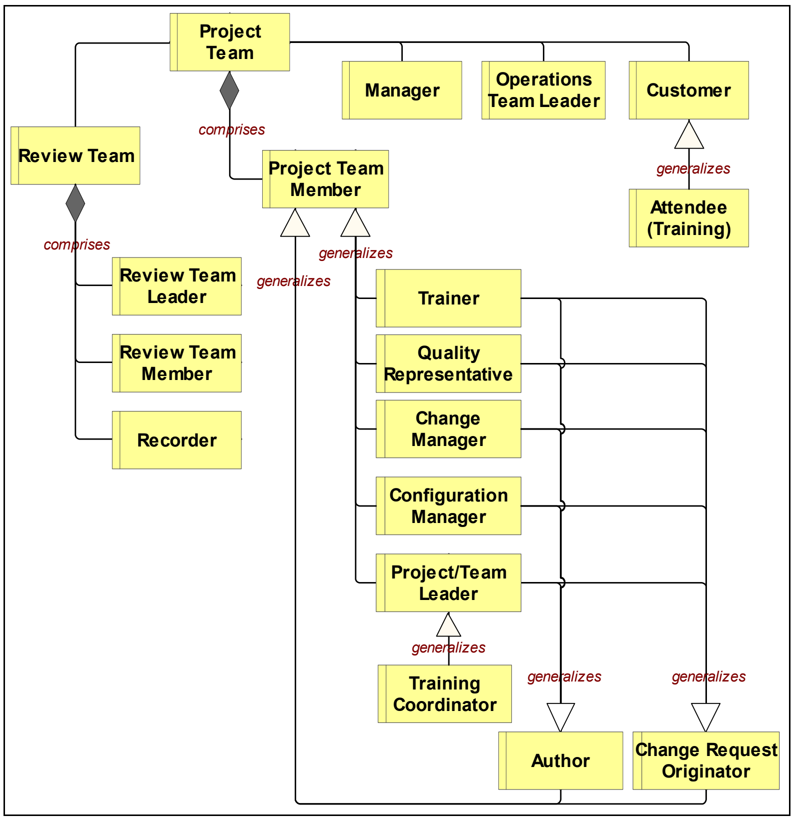


Figure 96. Role Diagram

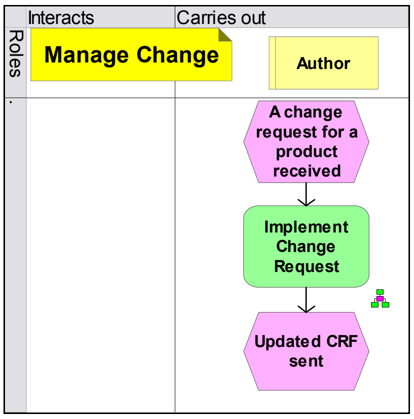


Figure 97. Manage Change

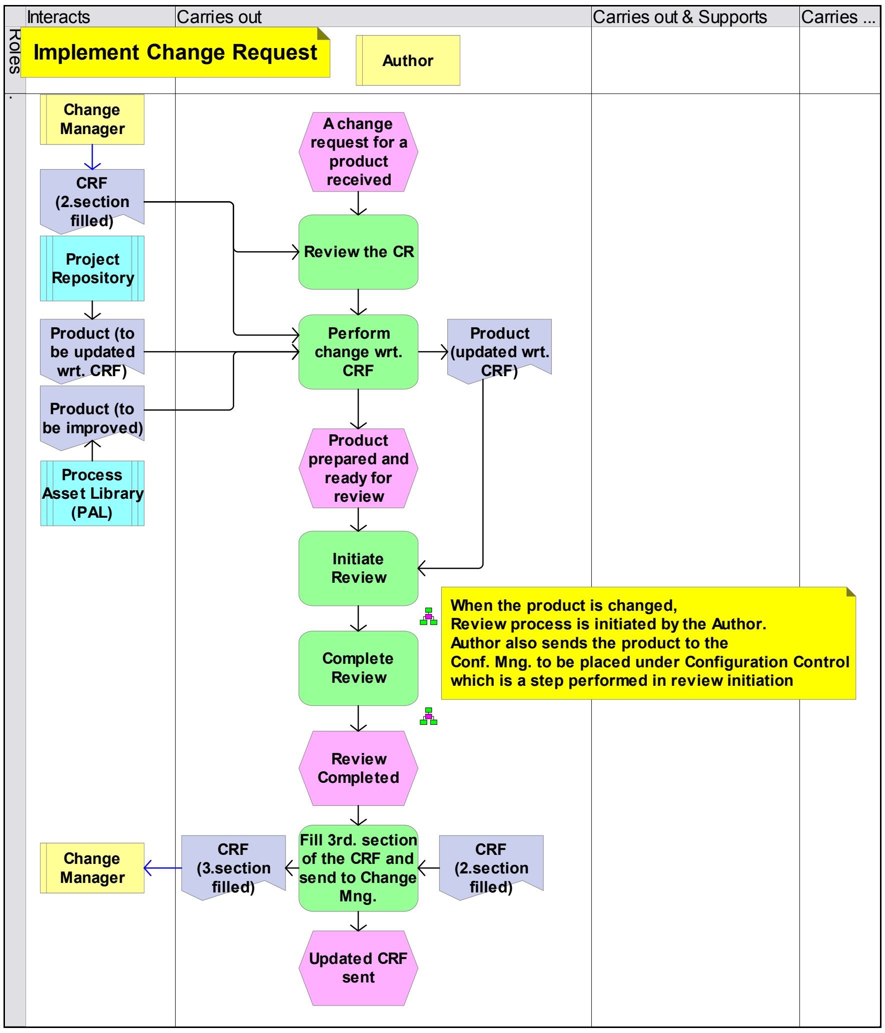

Interacts
Carries out
Carries out & Supports
Carries ...
Roles
Implement Change Request
Author
Change Manager
CRF (2.section filled)
Project Repository
Product (to be updated wrt. CRF)
Product (to be improved)
Process Asset Library (PAL)
A change request for a product received
Review the CR
Perform change wrt. CRF
Product (updated wrt. CRF)
Product prepared and ready for review
Initiate Review
Complete Review
When the product is changed, Review process is initiated by the Author. Author also sends the product to the Conf. Mng. to be placed under Configuration Control which is a step performed in review initiation
Review Completed
Change Manager
CRF (3.section filled)
Fill 3rd. section of the CRF and send to Change Mng.
CRF (2.section filled)
Updated CRF sent

Figure 98. Implement Change Request

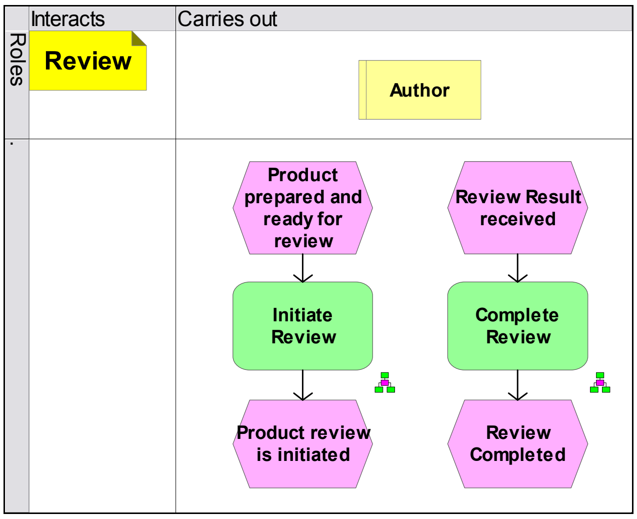


Figure 99. Review

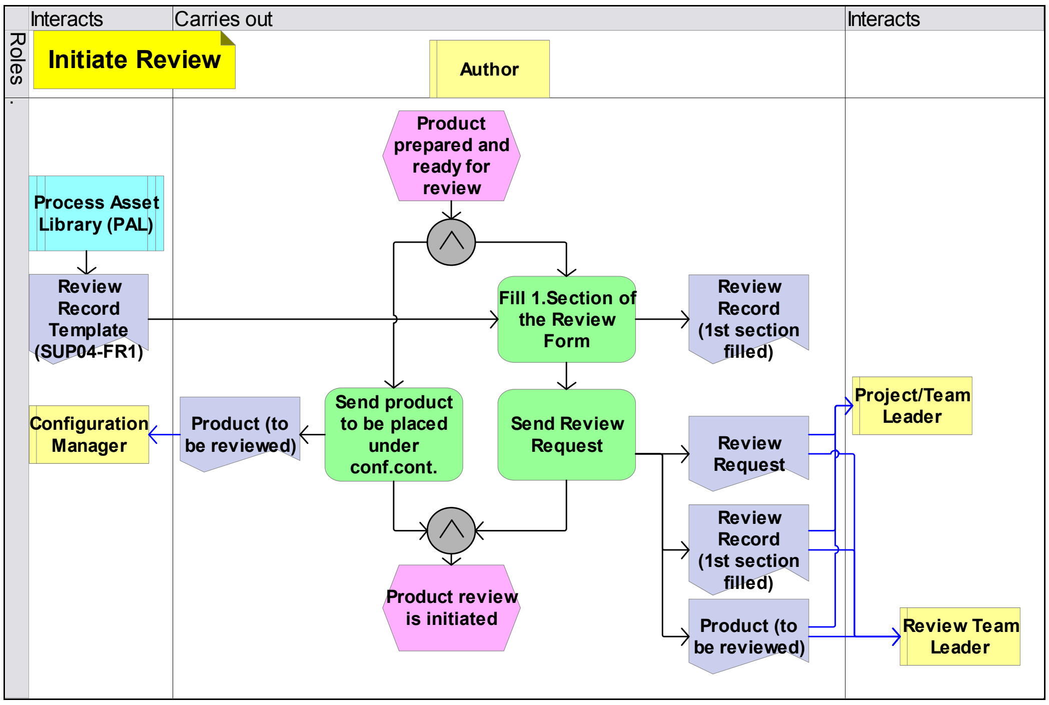
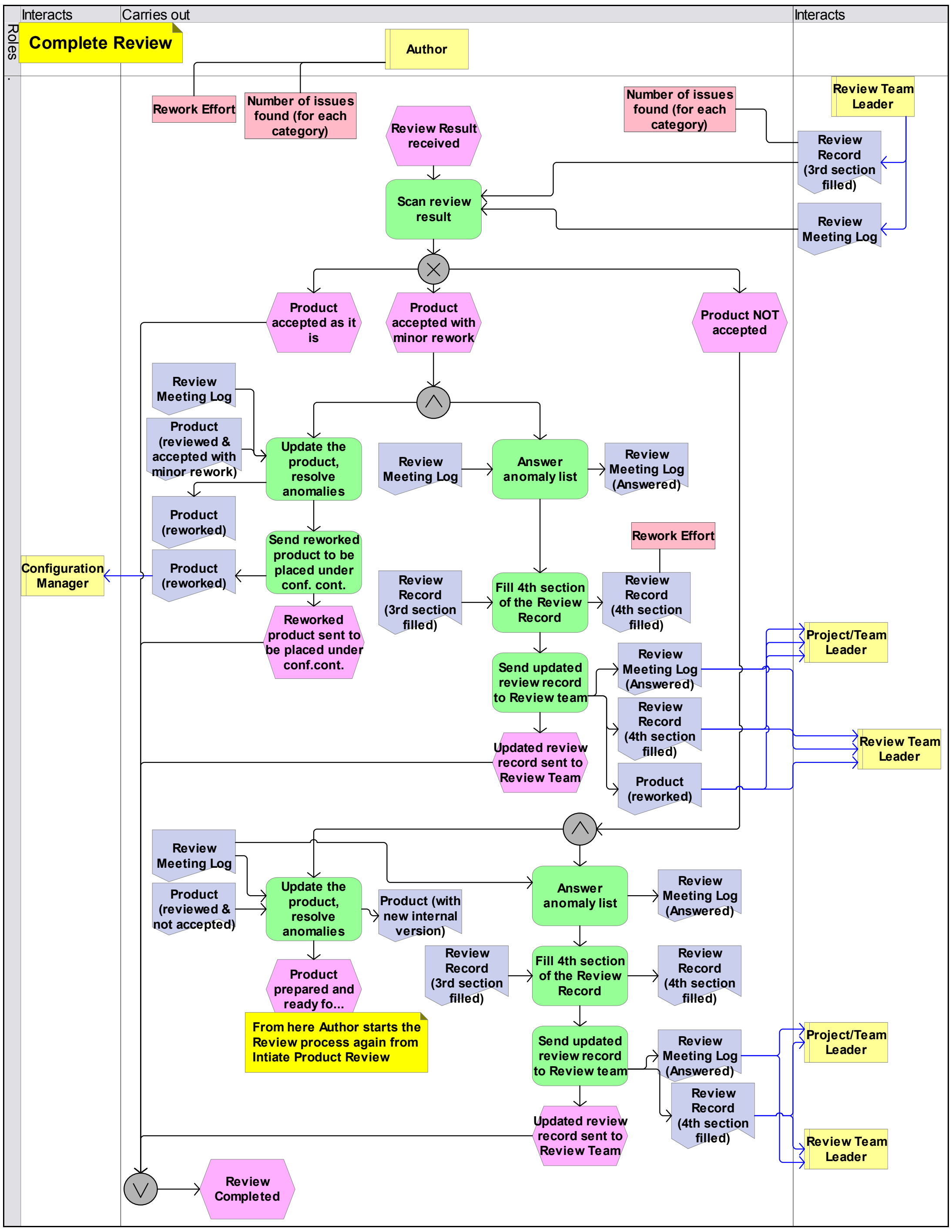

Interacts
Carries out
Interacts
Roles
Initiate Review
Author
Product prepared and ready for review
Process Asset Library (PAL)
Review Record Template (SUP04-FR1)
Fill 1.Section of the Review Form
Review Record (1st section filled)
Configuration Manager
Product (to be reviewed)
Send product to be placed under conf.cont.
Send Review Request
Project/Team Leader
Review Request
Review Record (1st section filled)
Product review is initiated
Product (to be reviewed)
Review Team Leader

Figure 100. Initiate Review



Interacts
Carries out
Interacts
Roles
Complete Review
Author
Rework Effort
Number of issues found (for each category)
Review Result received
Number of issues found (for each category)
Review Team Leader
Review Record (3rd section filled)
Scan review result
Review Meeting Log
Product accepted as it is
Product accepted with minor rework
Product NOT accepted
Review Meeting Log
Product (reviewed & accepted with minor rework)
Update the product, resolve anomalies
Review Meeting Log
Answer anomaly list
Review Meeting Log (Answered)
Product (reworked)
Rework Effort
Configuration Manager
Product (reworked)
Send reworked product to be placed under conf. cont.
Review Record (3rd section filled)
Fill 4th section of the Review Record
Review Record (4th section filled)
Reworked product sent to be placed under conf.cont.
Project/Team Leader
Send updated review record to Review team
Review Meeting Log (Answered)
Review Record (4th section filled)
Review Team Leader
Updated review record sent to Review Team
Product (reworked)
Review Meeting Log
Product (reviewed & not accepted)
Update the product, resolve anomalies
Product (with new internal version)
Answer anomaly list
Review Meeting Log (Answered)
Review Record (3rd section filled)
Fill 4th section of the Review Record
Review Record (4th section filled)
Product prepared and ready fo...
From here Author starts the Review process again from Intiate Product Review
Send updated review record to Review team
Review Meeting Log (Answered)
Project/Team Leader
Review Record (4th section filled)
Updated review record sent to Review Team
Review Team Leader
Review Completed

Figure 101. Complete Review

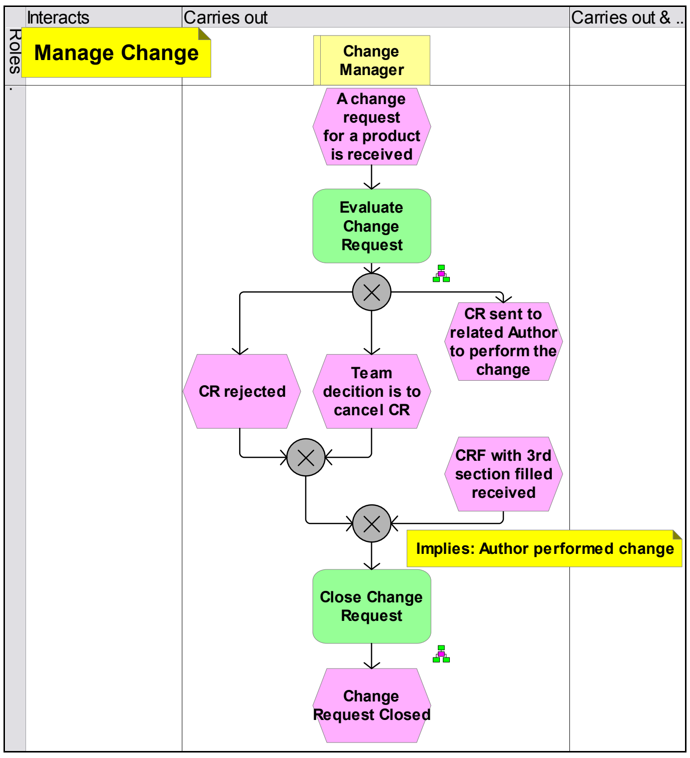

Interacts
Carries out
Carries out & ..
Roles
Manage Change
Change Manager
A change request for a product is received
Evaluate Change Request
CR sent to related Author to perform the change
CR rejected
Team decition is to cancel CR
CRF with 3rd section filled received
Implies: Author performed change
Close Change Request
Change Request Closed

Figure 102. Manage Change

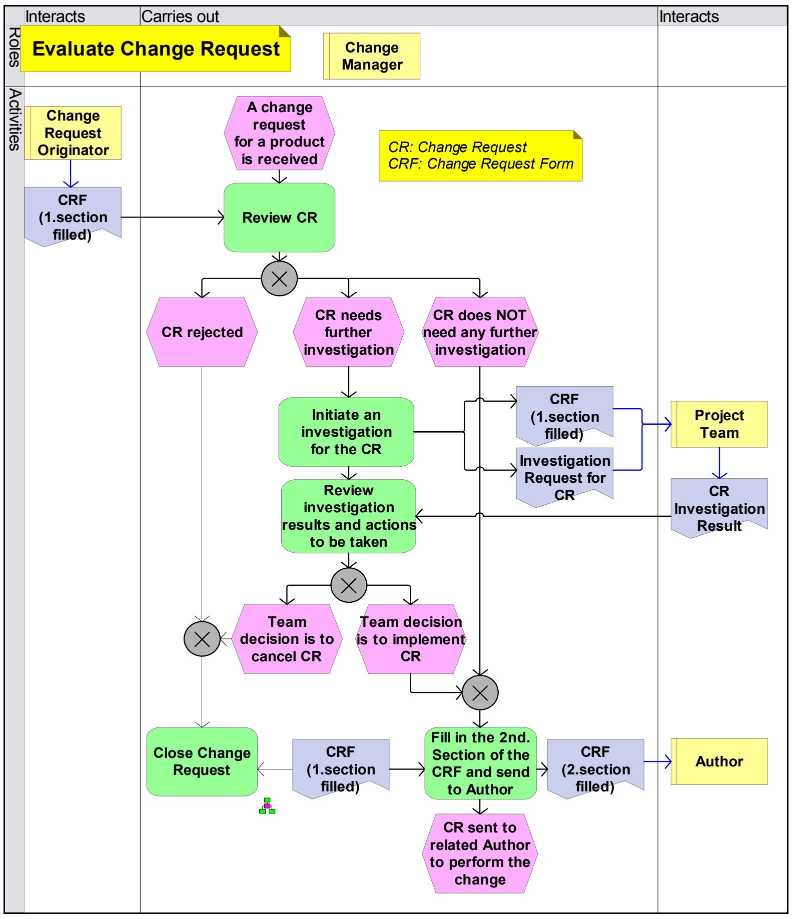

Interacts
Carries out
Interacts
Roles
Activities
Evaluate Change Request
Change Manager
Change Request Originator
CRF (1.section filled)
A change request for a product is received
CR: Change Request
CRF: Change Request Form
Review CR
CR rejected
CR needs further investigation
CR does NOT need any further investigation
Initiate an investigation for the CR
CRF (1.section filled)
Investigation Request for CR
Project Team
CR Investigation Result
Review investigation results and actions to be taken
Team decision is to cancel CR
Team decision is to implement CR
Close Change Request
CRF (1.section filled)
Fill in the 2nd. Section of the CRF and send to Author
CRF (2.section filled)
Author
CR sent to related Author to perform the change

Figure 103. Evaluate Change Request

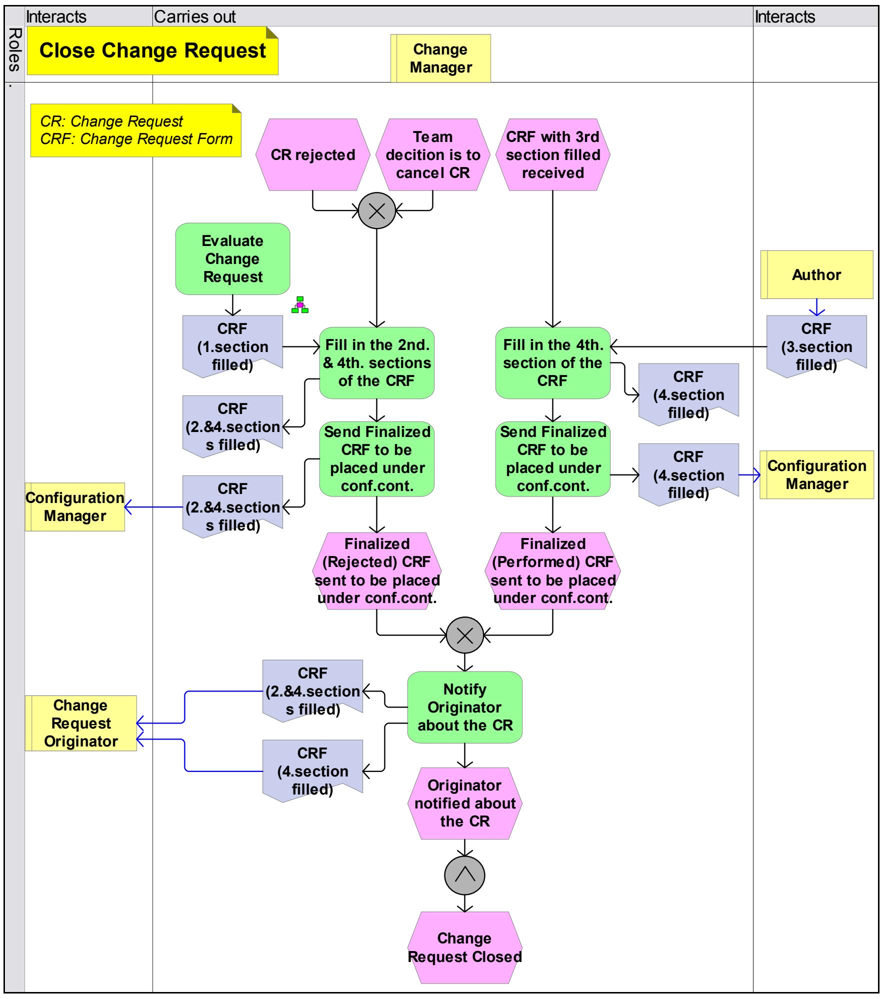

Interacts
Carries out
Interacts
Roles
Close Change Request
Change Manager
CR: Change Request
CRF: Change Request Form
CR rejected
Team decition is to cancel CR
CRF with 3rd section filled received
Evaluate Change Request
Author
CRF (1.section filled)
Fill in the 2nd. & 4th. sections of the CRF
Fill in the 4th. section of the CRF
CRF (3.section filled)
CRF (4.section filled)
CRF (2.&4.sections filled)
Send Finalized CRF to be placed under conf.cont.
Send Finalized CRF to be placed under conf.cont.
CRF (4.section filled)
Configuration Manager
Configuration Manager
CRF (2.&4.sections filled)
Finalized (Rejected) CRF sent to be placed under conf.cont.
Finalized (Performed) CRF sent to be placed under conf.cont.
CRF (2.&4.sections filled)
Notify Originator about the CR
Change Request Originator
CRF (4.section filled)
Originator notified about the CR
Change Request Closed

Figure 104. Close Change Request

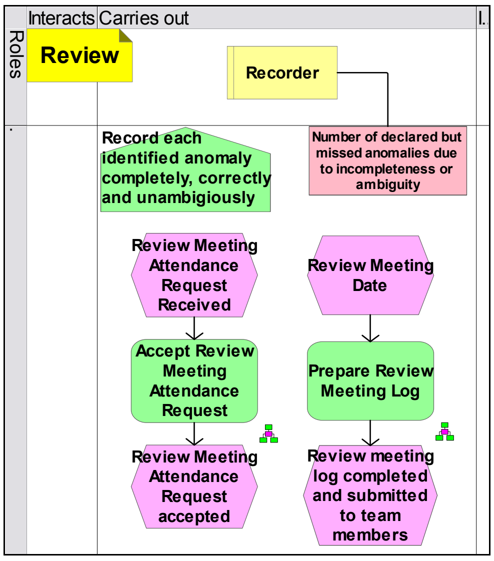

Interacts
Carries out
I.
Roles
Review
Recorder
Record each identified anomaly completely, correctly and unambigiously
Number of declared but missed anomalies due to incompleteness or ambiguity
Review Meeting Attendance Request Received
Review Meeting Date
Accept Review Meeting Attendance Request
Prepare Review Meeting Log
Review Meeting Attendance Request accepted
Review meeting log completed and submitted to team members

Figure 105. Review (Recorder)

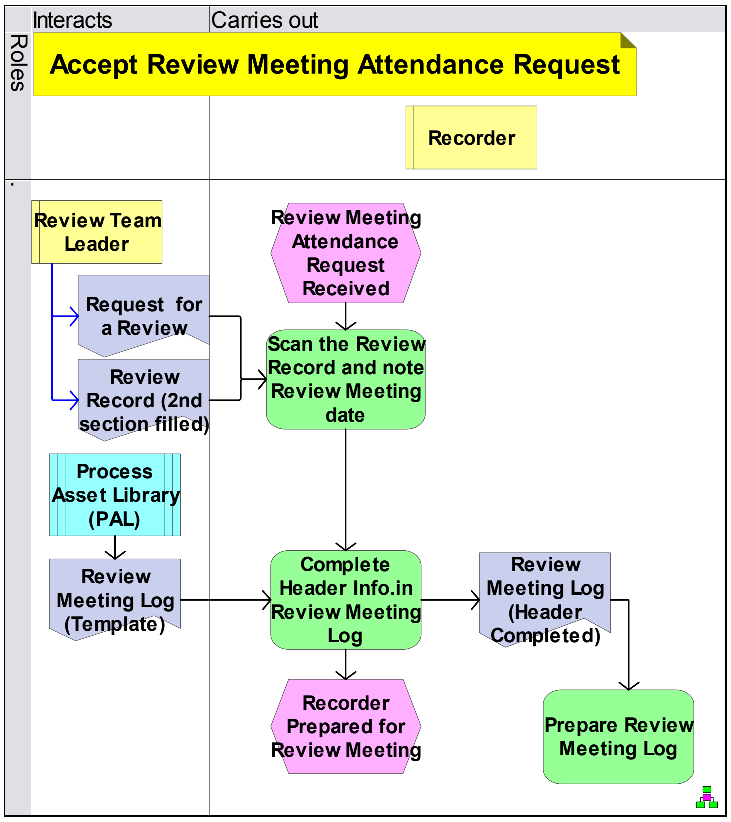
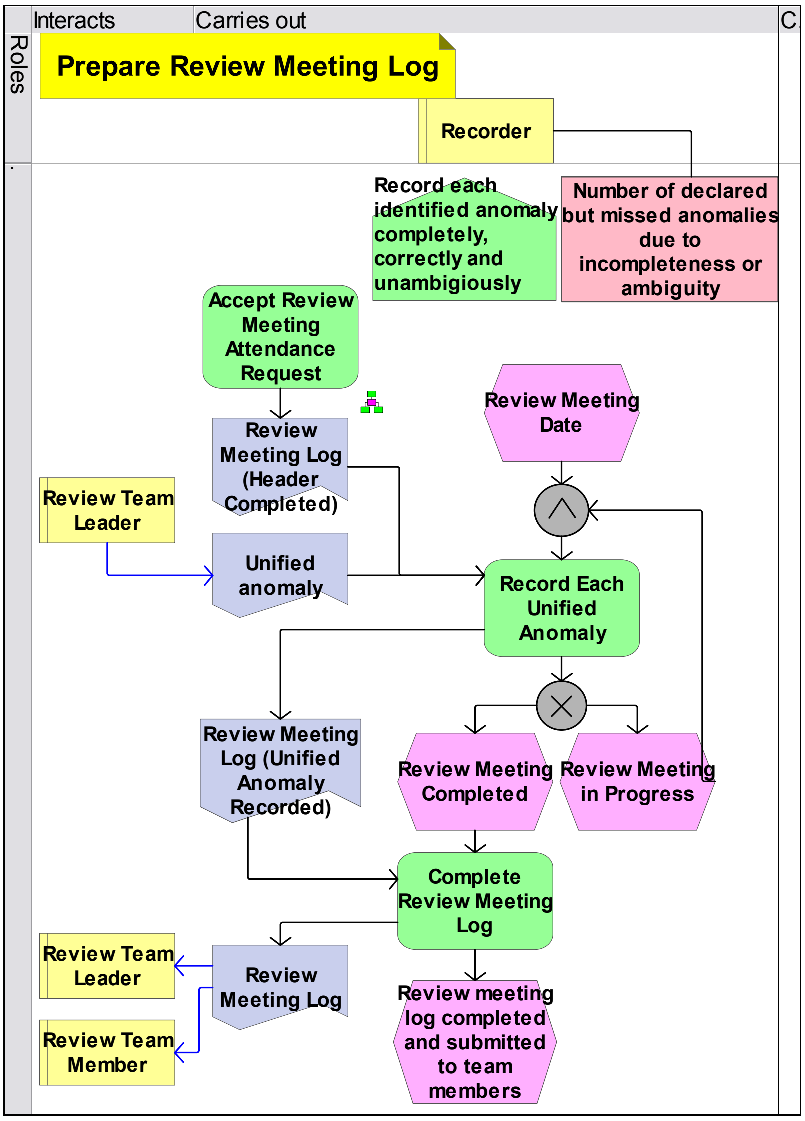

Interacts
Carries out
Roles
Accept Review Meeting Attendance Request
Recorder
Review Team Leader
Request for a Review
Review Record (2nd section filled)
Process Asset Library (PAL)
Review Meeting Log (Template)
Review Meeting Attendance Request Received
Scan the Review Record and note Review Meeting date
Complete Header Info.in Review Meeting Log
Review Meeting Log (Header Completed)
Recorder Prepared for Review Meeting
Prepare Review Meeting Log

Figure 106. Accept Review Meeting Attendance Request



Interacts
Carries out
C
Roles
Prepare Review Meeting Log
Recorder
Record each identified anomaly completely, correctly and unambigiously
Number of declared but missed anomalies due to incompleteness or ambiguity
Accept Review Meeting Attendance Request
Review Meeting Log (Header Completed)
Review Meeting Date
Review Team Leader
Unified anomaly
Record Each Unified Anomaly
Review Meeting Log (Unified Anomaly Recorded)
Review Meeting Completed
Review Meeting in Progress
Complete Review Meeting Log
Review Team Leader
Review Meeting Log
Review Team Member
Review meeting log completed and submitted to team members

Figure 107. Prepare Review Meeting Log

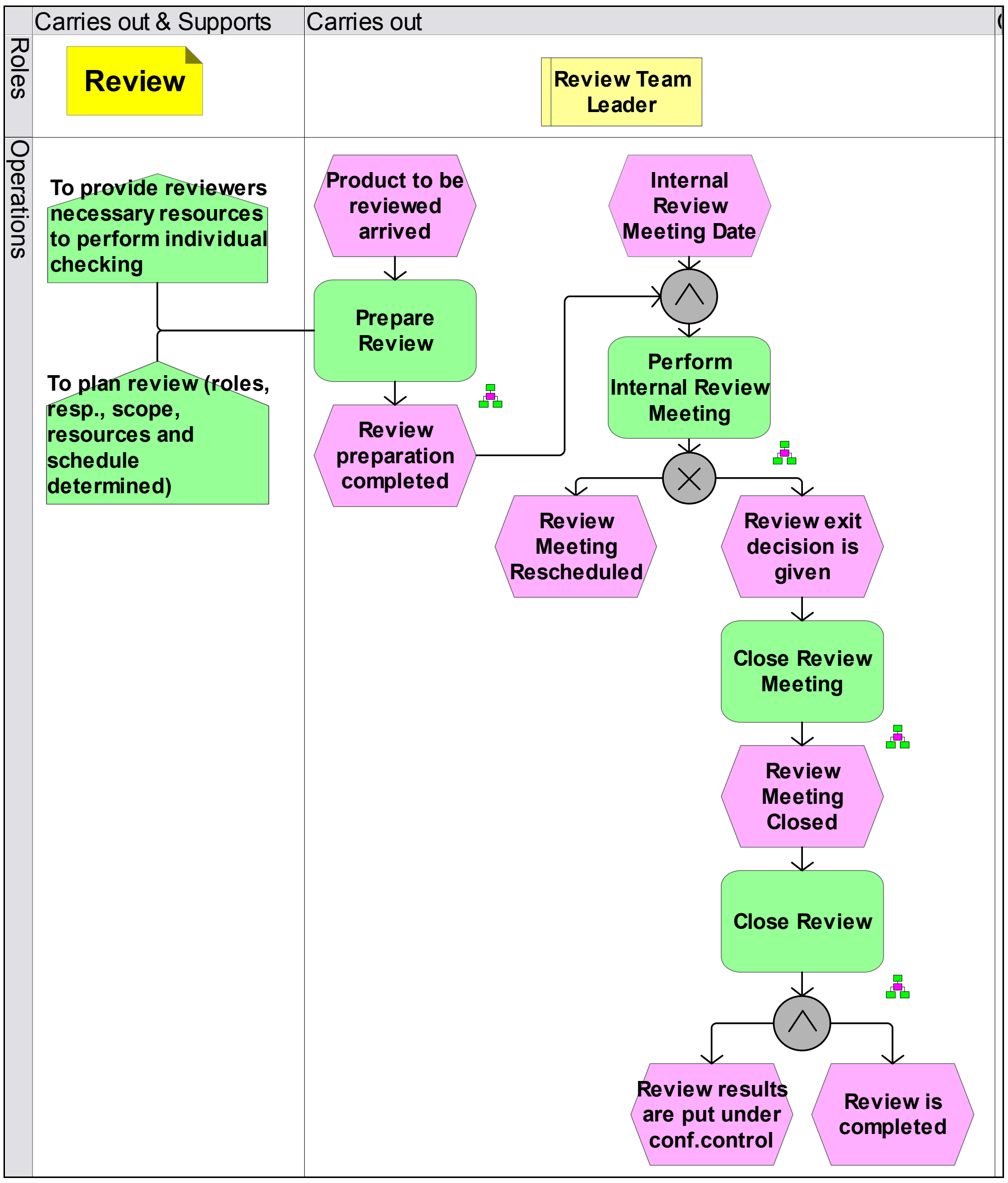

Carries out & Supports
Carries out
Roles
Operations
Review
Review Team Leader
To provide reviewers necessary resources to perform individual checking
To plan review (roles, resp., scope, resources and schedule determined)
Product to be reviewed arrived
Prepare Review
Review preparation completed
Internal Review Meeting Date
Perform Internal Review Meeting
Review Meeting Rescheduled
Review exit decision is given
Close Review Meeting
Review Meeting Closed
Close Review
Review results are put under conf.control
Review is completed

Figure 108. Review (Review Team Leader)

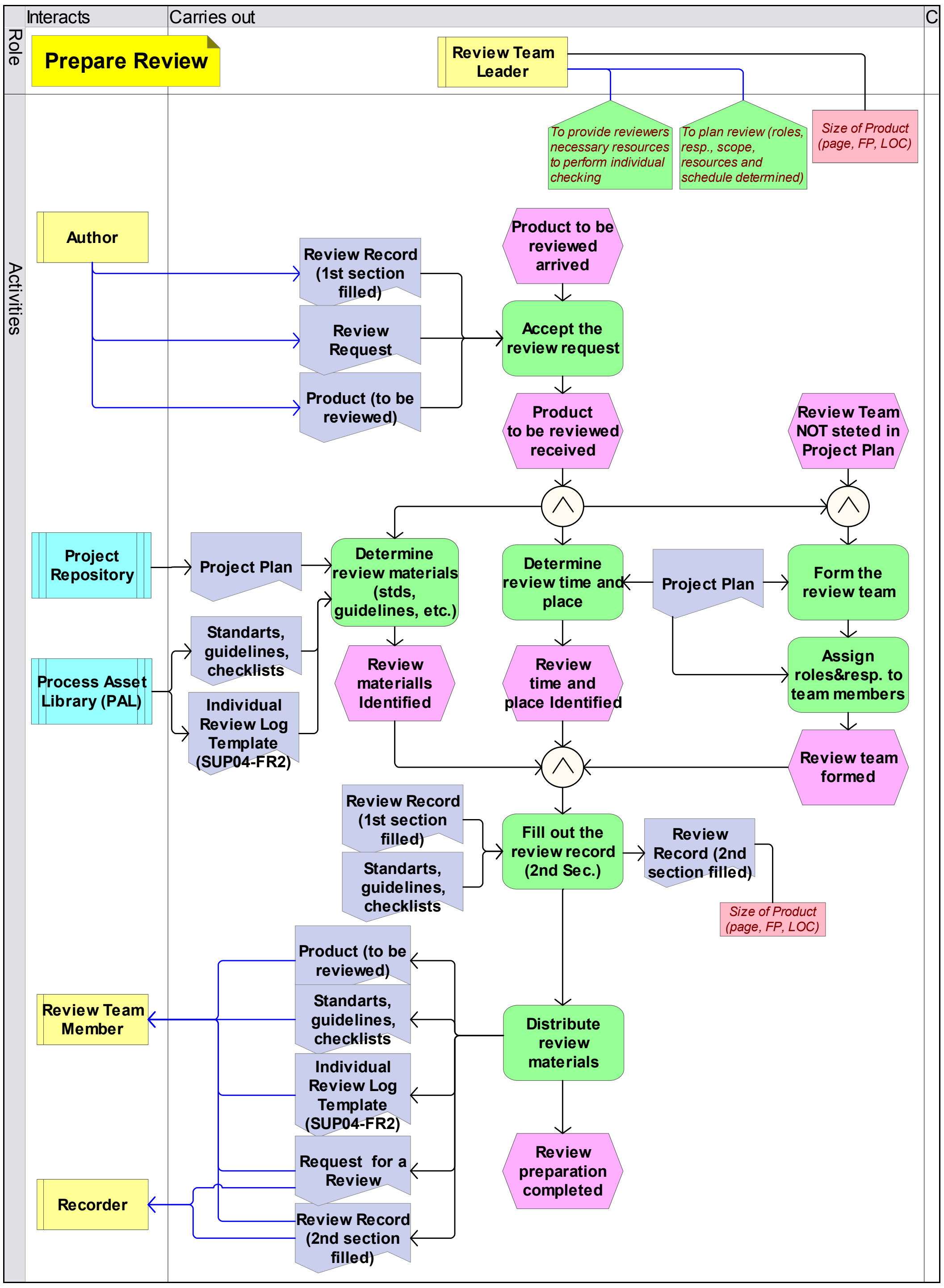

Role
Interacts
Carries out
C
Prepare Review
Review Team Leader
To provide reviewers necessary resources to perform individual checking
To plan review (roles, resp., scope, resources and schedule determined)
Size of Product (page, FP, LOC)
Activities
Author
Review Record (1st section filled)
Review Request
Product (to be reviewed)
Product to be reviewed arrived
Accept the review request
Product to be reviewed received
Review Team NOT steted in Project Plan
Project Repository
Project Plan
Determine review materials (stds, guidelines, etc.)
Determine review time and place
Project Plan
Form the review team
Process Asset Library (PAL)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Review materialls Identified
Review time and place Identified
Assign roles&resp. to team members
Review team formed
Review Record (1st section filled)
Standarts, guidelines, checklists
Fill out the review record (2nd Sec.)
Review Record (2nd section filled)
Size of Product (page, FP, LOC)
Product (to be reviewed)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Request for a Review
Review Record (2nd section filled)
Review Team Member
Recorder
Distribute review materials
Review preparation completed

Figure 109. Prepare Review

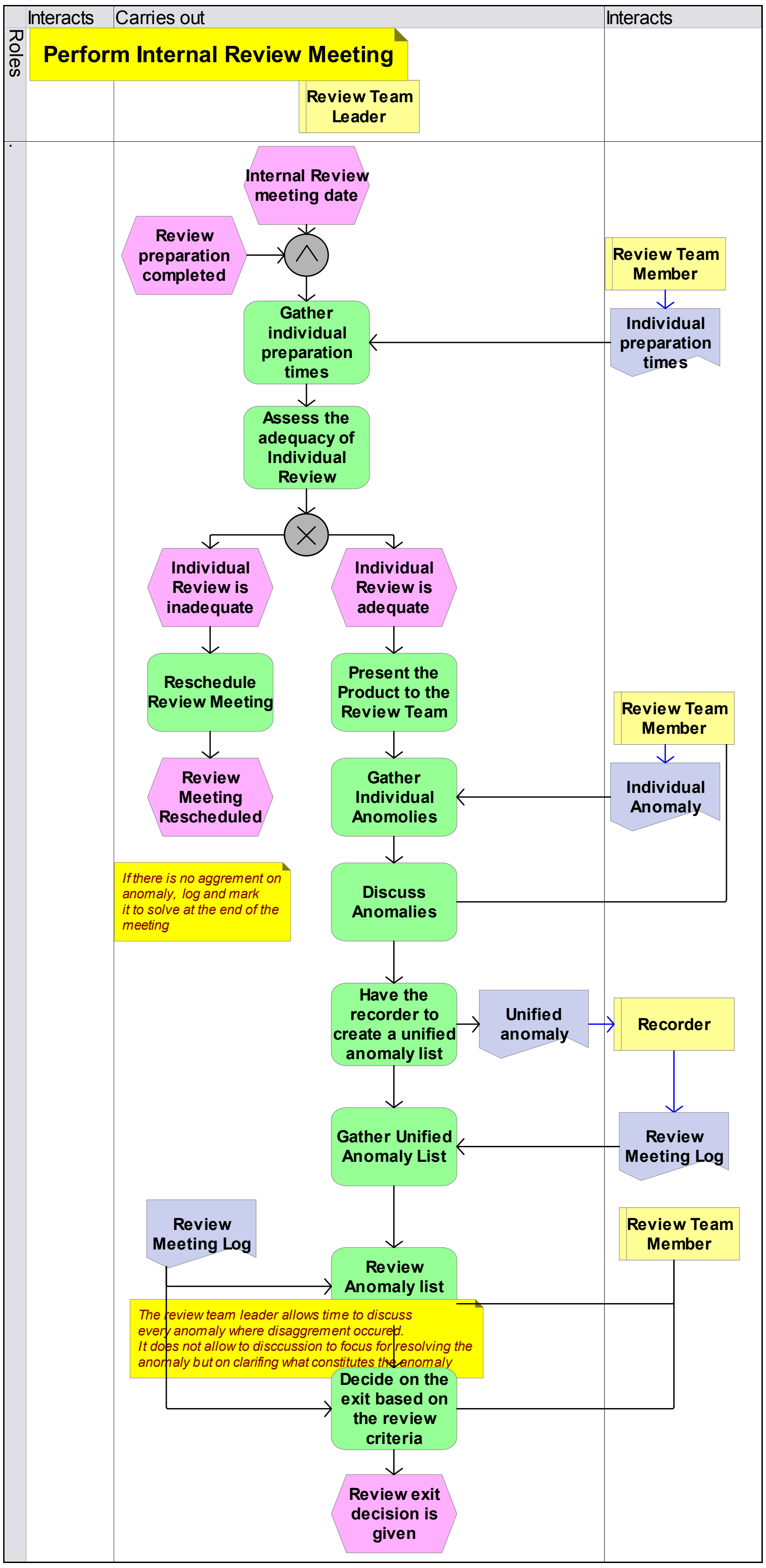

Interacts
Carries out
Interacts
Roles
Perform Internal Review Meeting
Review Team Leader
Internal Review meeting date
Review preparation completed
Review Team Member
Gather individual preparation times
Individual preparation times
Assess the adequacy of Individual Review
Individual Review is inadequate
Individual Review is adequate
Reschedule Review Meeting
Present the Product to the Review Team
Review Team Member
Review Meeting Rescheduled
Gather Individual Anomolies
Individual Anomaly
If there is no aggrement on anomaly, log and mark it to solve at the end of the meeting
Discuss Anomalies
Have the recorder to create a unified anomaly list
Unified anomaly
Recorder
Gather Unified Anomaly List
Review Meeting Log
Review Meeting Log
Review Team Member
Review Anomaly list
The review team leader allows time to discuss every anomaly where disaggrement occured. It does not allow to disccussion to focus for resolving the anomaly but on clarifing what constitutes the anomaly
Decide on the exit based on the review criteria
Review exit decision is given

Figure 110. Perform Internal Review Meeting

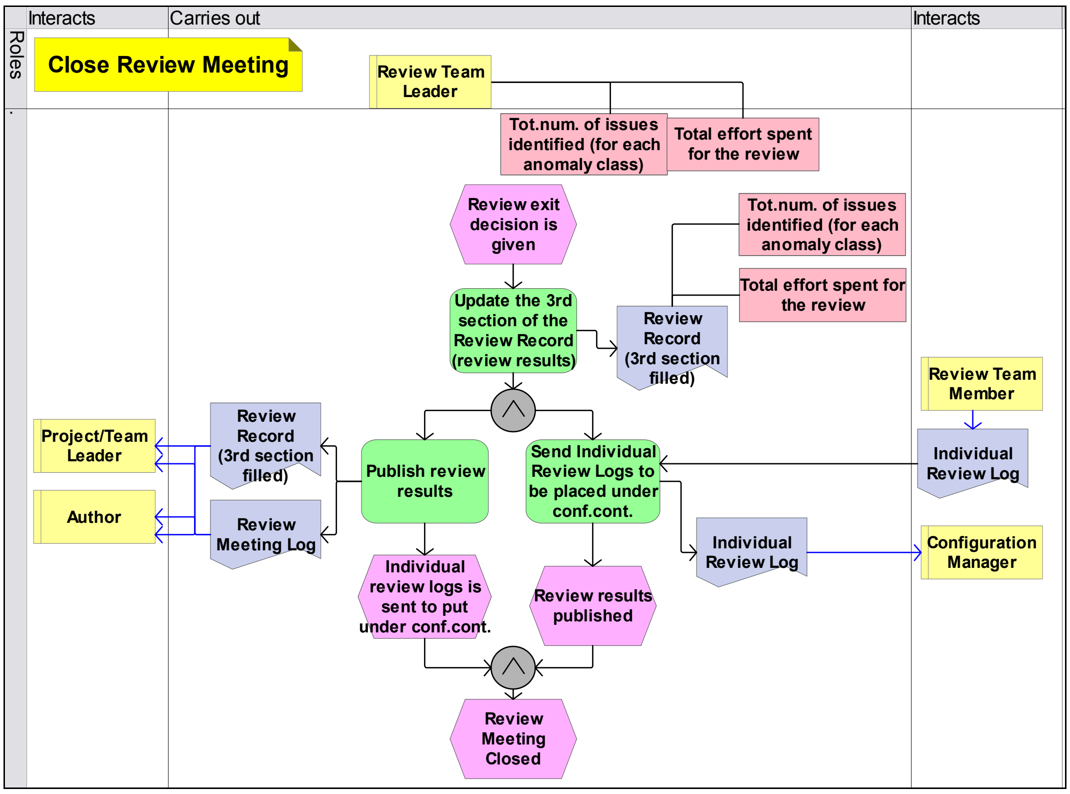

Interacts
Carries out
Interacts
Roles
Close Review Meeting
Review Team Leader
Tot.num. of issues identified (for each anomaly class)
Total effort spent for the review
Review exit decision is given
Tot.num. of issues identified (for each anomaly class)
Total effort spent for the review
Update the 3rd section of the Review Record (review results)
Review Record (3rd section filled)
Review Team Member
Project/Team Leader
Review Record (3rd section filled)
Publish review results
Send Individual Review Logs to be placed under conf.cont.
Individual Review Log
Author
Review Meeting Log
Individual Review Log
Configuration Manager
Individual review logs is sent to put under conf.cont.
Review results published
Review Meeting Closed

Figure 111. Close Review Meeting

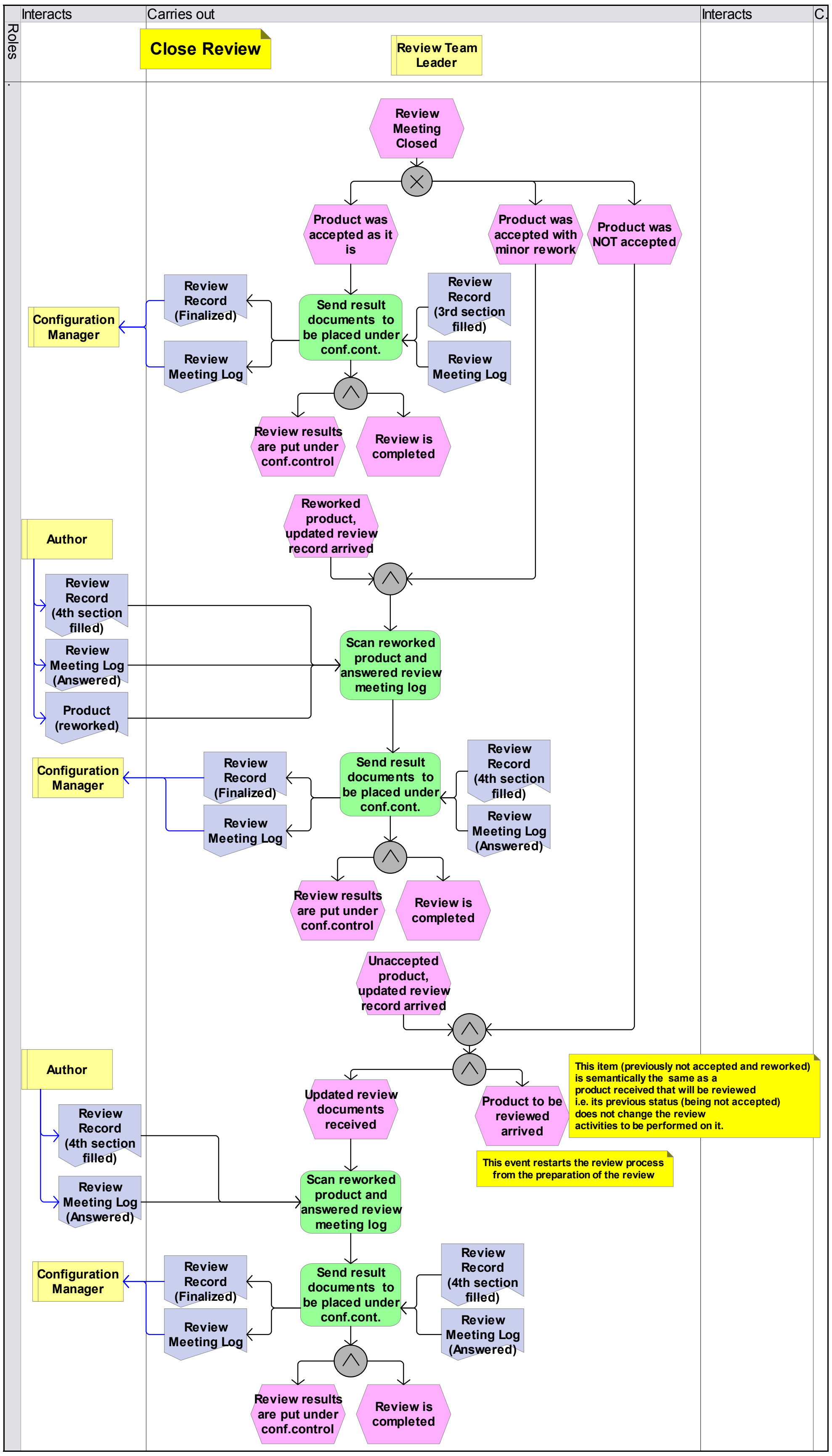
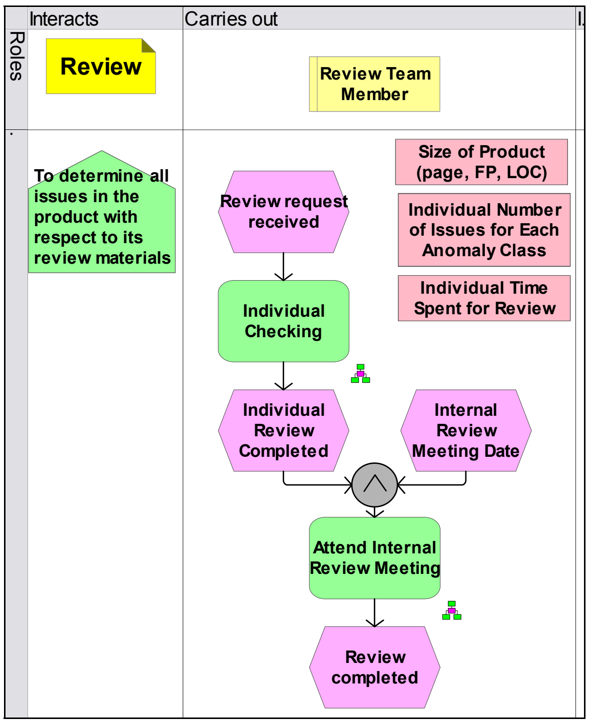
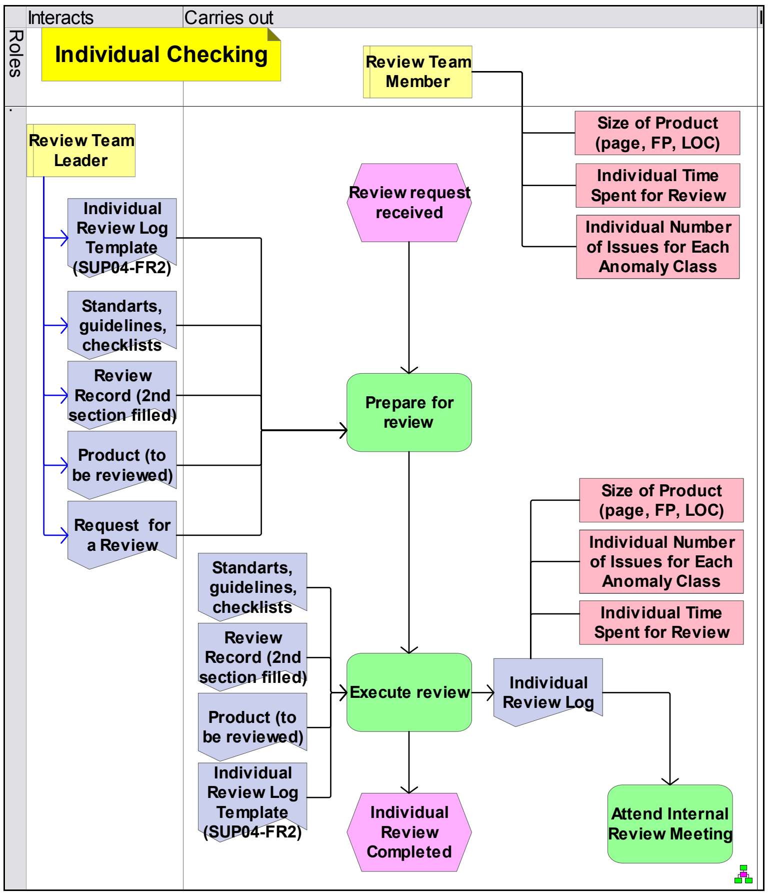

Interacts
Carries out
Interacts
C.
Roles
Close Review
Review Team Leader
Review Meeting Closed
Product was accepted as it is
Product was accepted with minor rework
Product was NOT accepted
Configuration Manager
Review Record (Finalized)
Review Meeting Log
Send result documents to be placed under conf.cont.
Review Record (3rd section filled)
Review Meeting Log
Review results are put under conf.control
Review is completed
Reworked product, updated review record arrived
Author
Review Record (4th section filled)
Review Meeting Log (Answered)
Product (reworked)
Scan reworked product and answered review meeting log
Configuration Manager
Review Record (Finalized)
Review Meeting Log
Send result documents to be placed under conf.cont.
Review Record (4th section filled)
Review Meeting Log (Answered)
Review results are put under conf.control
Review is completed
Unaccepted product, updated review record arrived
Author
Updated review documents received
Product to be reviewed arrived
This item (previously not accepted and reworked) is semantically the same as a product received that will be reviewed i.e. its previous status (being not accepted) does not change the review activities to be performed on it.
This event restarts the review process from the preparation of the review
Review Record (4th section filled)
Review Meeting Log (Answered)
Scan reworked product and answered review meeting log
Configuration Manager
Review Record (Finalized)
Review Meeting Log
Send result documents to be placed under conf.cont.
Review Record (4th section filled)
Review Meeting Log (Answered)
Review results are put under conf.control
Review is completed

Figure 112. Close Review



Interacts
Carries out
Roles
Review
Review Team Member
To determine all issues in the product with respect to its review materials
Size of Product (page, FP, LOC)
Individual Number of Issues for Each Anomaly Class
Individual Time Spent for Review
Review request received
Individual Checking
Individual Review Completed
Internal Review Meeting Date
Attend Internal Review Meeting
Review completed

Figure 113. Review (Review Team Member)



Interacts
Carries out
Roles
Individual Checking
Review Team Member
Review Team Leader
Size of Product (page, FP, LOC)
Individual Time Spent for Review
Individual Number of Issues for Each Anomaly Class
Review request received
Individual Review Log Template (SUP04-FR2)
Standarts, guidelines, checklists
Review Record (2nd section filled)
Product (to be reviewed)
Request for a Review
Prepare for review
Size of Product (page, FP, LOC)
Individual Number of Issues for Each Anomaly Class
Individual Time Spent for Review
Standarts, guidelines, checklists
Review Record (2nd section filled)
Product (to be reviewed)
Individual Review Log Template (SUP04-FR2)
Execute review
Individual Review Log
Individual Review Completed
Attend Internal Review Meeting

Figure 114. Individual Checking

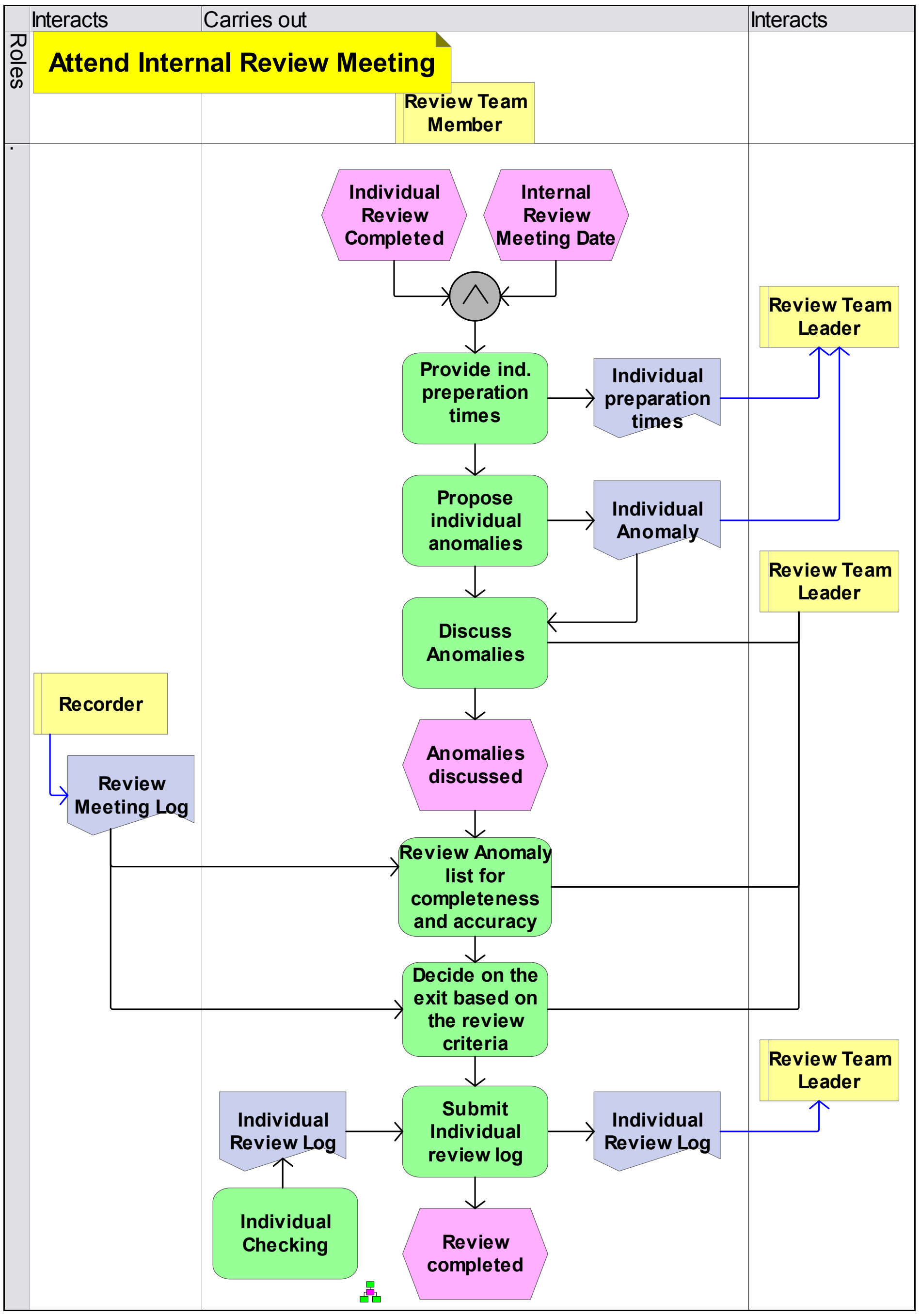
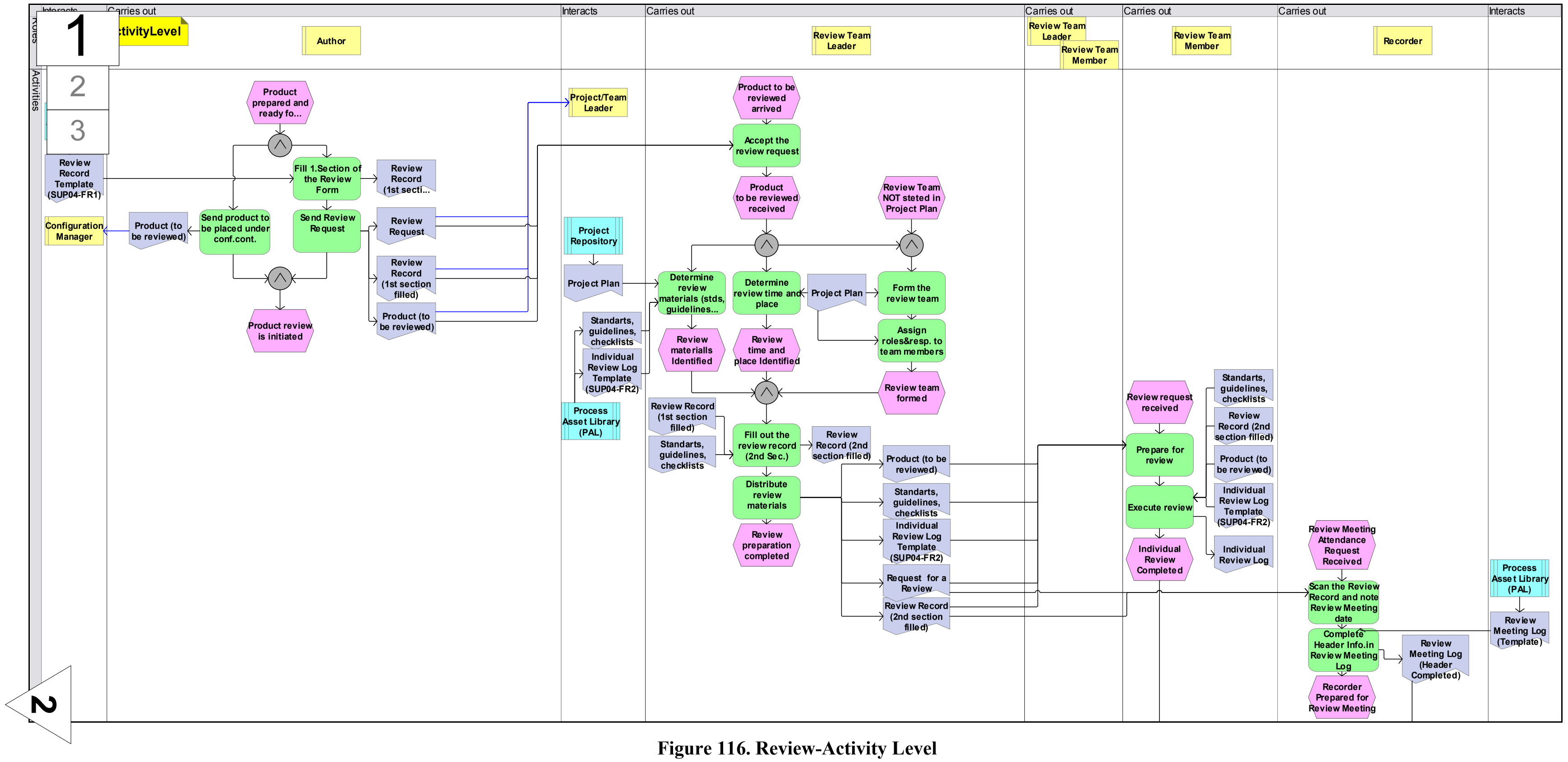
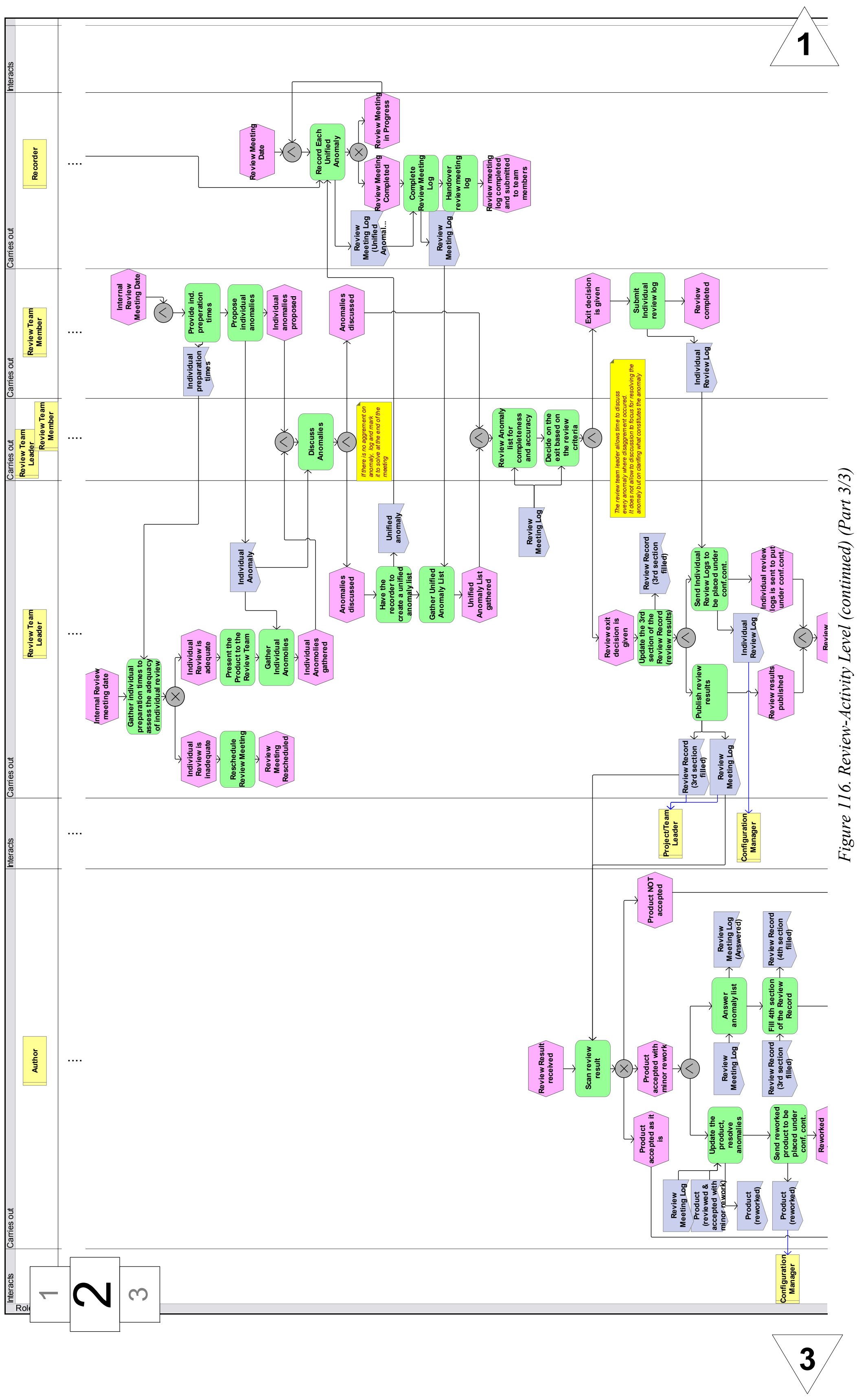

Interacts
Carries out
Interacts
Roles
Attend Internal Review Meeting
Review Team Member
Individual Review Completed
Internal Review Meeting Date
Provide ind. preperation times
Individual preparation times
Review Team Leader
Propose individual anomalies
Individual Anomaly
Review Team Leader
Discuss Anomalies
Recorder
Review Meeting Log
Anomalies discussed
Review Anomaly list for completeness and accuracy
Decide on the exit based on the review criteria
Review Team Leader
Individual Review Log
Submit Individual review log
Individual Review Log
Individual Checking
Review completed

Figure 115. Attend Internal Review Meeting



Interacts
Carries out
Interacts
Carries out
Carries out
Carries out
Carries out
Interacts
Roles
Activities
1
2
3
ActivityLevel
Author
Review Team Leader
Review Team Leader
Review Team Member
Review Team Member
Recorder
Review Record Template (SUP04-FR1)
Configuration Manager
Product prepared and ready fo...
Fill 1.Section of the Review Form
Review Record (1st secti...
Send product to be placed under conf.cont.
Product (to be reviewed)
Send Review Request
Review Request
Review Record (1st section filled)
Product (to be reviewed)
Product review is initiated
Project/Team Leader
Project Repository
Project Plan
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Process Asset Library (PAL)
Product to be reviewed arrived
Accept the review request
Product to be reviewed received
Review Team NOT steted in Project Plan
Determine review materials (stds, guidelines...
Determine review time and place
Project Plan
Form the review team
Assign roles&resp. to team members
Review materialls Identified
Review time and place Identified
Review team formed
Review Record (1st section filled)
Standarts, guidelines, checklists
Fill out the review record (2nd Sec.)
Review Record (2nd section filled)
Distribute review materials
Review preparation completed
Product (to be reviewed)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Request for a Review
Review Record (2nd section filled)
Review request received
Prepare for review
Execute review
Individual Review Completed
Standarts, guidelines, checklists
Review Record (2nd section filled)
Product (to be reviewed)
Individual Review Log Template (SUP04-FR2)
Individual Review Log
Review Meeting Attendance Request Received
Scan the Review Record and note Review Meeting date
Complete Header Info.in Review Meeting Log
Review Meeting Log (Header Completed)
Recorder Prepared for Review Meeting
Process Asset Library (PAL)
Review Meeting Log (Template)
2

Figure 116. Review-Activity Level
(Part 1/3)



Interacts
Carries out
Role
Recorder
Review Team Member
Review Team Leader
Author
Project/Team Leader
Configuration Manager
Review Meeting Date
Record Each Unified Anomaly
Review Meeting in Progress
Review Meeting Completed
Complete Review Meeting Log
Handover review meeting log
Review meeting log completed and submitted to team members
Review Meeting Log (Unified Anomal...
Review Meeting Log
Internal Review Meeting Date
Provide ind. preparation times
Individual preparation times
Propose Individual anomalies
Individual anomalies proposed
Anomalies discussed
Exit decision is given
Submit Individual review log
Review completed
Individual Review Log
Discuss Anomalies
If there is no aggreement on anomaly, log and mark it to solve at the end of the meeting
Review Anomaly list for completeness and accuracy
Decide on the exit based on the review criteria
The review team leader allows time to discuss every anomaly where disaggreement occured. It does not allow to discussion to focus for resolving the anomaly but on clarifing what constitutes the anomaly
Internal Review meeting date
Gather individual preparation times to assess the adequacy of individual review
Individual Review is adequate
Individual Review is inadequate
Reschedule Review Meeting
Review Meeting Rescheduled
Present the Product to the Review Team
Gather Individual Anomolies
Individual Anomaly
Individual Anomolies gathered
Have the recorder to create a unified anomaly list
Unified anomaly
Gather Unified Anomaly List
Unified Anomaly List gathered
Review exit decision is given
Review Record (3rd section filled)
Update the 3rd section of the Review Record (review results)
Send Individual Review Logs to be placed under conf. cont.
Individual review logs is sent to put under conf.cont.
Publish review results
Review results published
Review
Review Result received
Scan review result
Product accepted as it is
Product accepted with minor rework
Product NOT accepted
Update the product, resolve anomalies
Review Meeting Log (Answered)
Answer anomaly list
Fill 4th section of the Review Record
Review Record (4th section filled)
Product (reviewed & accepted with minor rework)
Product (reworked)
Send reworked product to be placed under conf. cont.
Reworked
1
2
3

Figure 116. Review-Activity Level (continued) (Part 3/3)

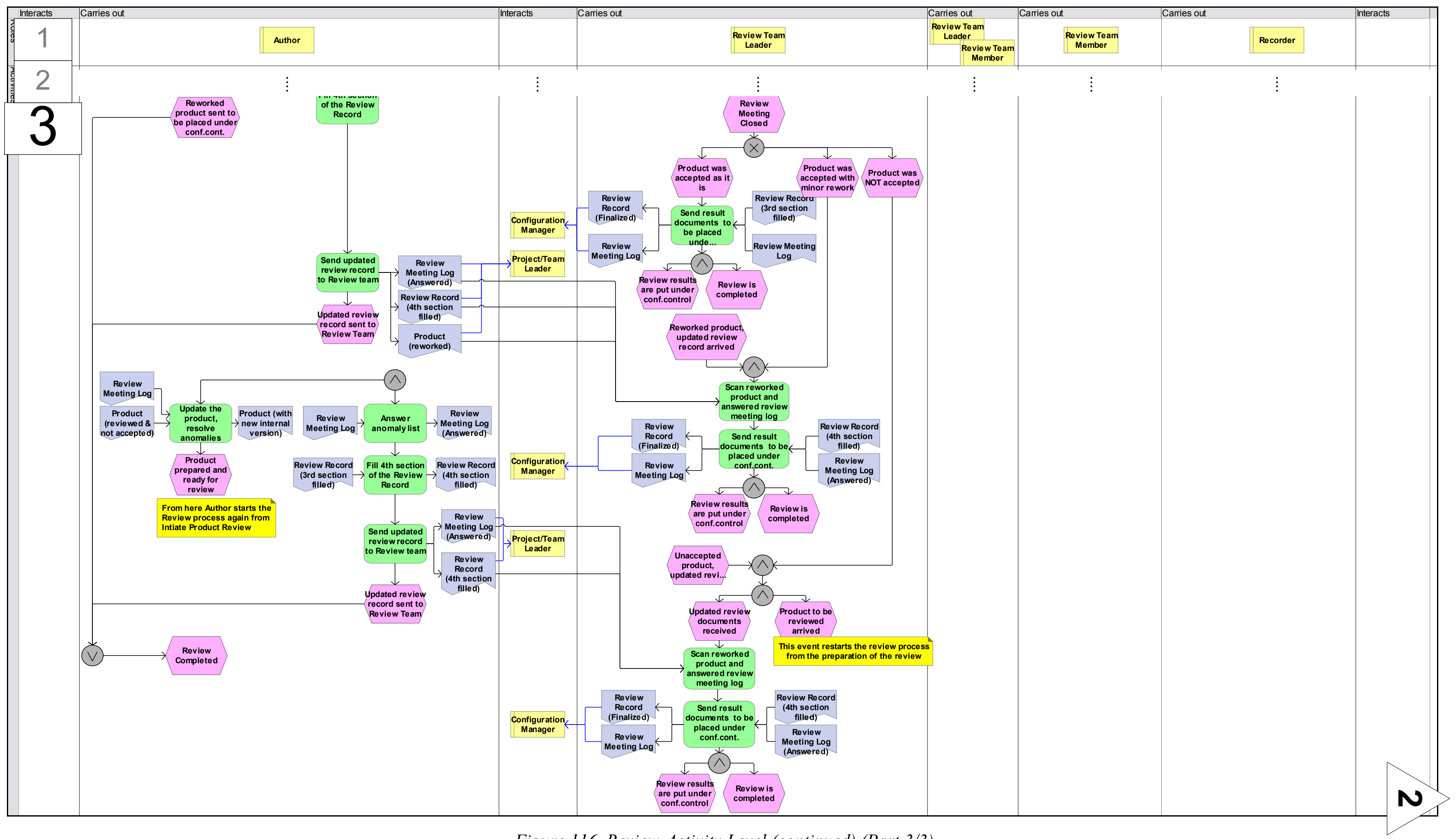

Interacts
Carries out
Interacts
Carries out
Carries out
Carries out
Carries out
Interacts
1
2
3
Roles
Activities
Author
Review Team Leader
Review Team Leader
Review Team Member
Review Team Member
Recorder
Reworked product sent to be placed under conf.cont.
Fill 4th section of the Review Record
Send updated review record to Review team
Updated review record sent to Review Team
Review Meeting Log (Answered)
Review Record (4th section filled)
Product (reworked)
Configuration Manager
Project/Team Leader
Review Meeting Closed
Product was accepted as it is
Product was accepted with minor rework
Product was NOT accepted
Review Record (Finalized)
Review Meeting Log
Send result documents to be placed unde...
Review Record (3rd section filled)
Review Meeting Log
Review results are put under conf.control
Review is completed
Reworked product, updated review record arrived
Scan reworked product and answered review meeting log
Review Meeting Log
Product (reviewed & not accepted)
Update the product, resolve anomalies
Product (with new internal version)
Review Meeting Log
Answer anomaly list
Review Meeting Log (Answered)
Product prepared and ready for review
Review Record (3rd section filled)
Fill 4th section of the Review Record
Review Record (4th section filled)
From here Author starts the Review process again from Intiate Product Review
Send updated review record to Review team
Review Meeting Log (Answered)
Review Record (4th section filled)
Updated review record sent to Review Team
Review Completed
Send result documents to be placed under conf.cont.
Review Record (Finalized)
Review Meeting Log
Review Record (4th section filled)
Review Meeting Log (Answered)
Review results are put under conf.control
Review is completed
Unaccepted product, updated revi...
Updated review documents received
Product to be reviewed arrived
This event restarts the review process from the preparation of the review
Scan reworked product and answered review meeting log
Send result documents to be placed under conf.cont.
Review Record (Finalized)
Review Meeting Log
Review Record (4th section filled)
Review Meeting Log (Answered)
Review results are put under conf.control
Review is completed
2

Figure 116. Review-Activity Level (continued) (Part 3/3)

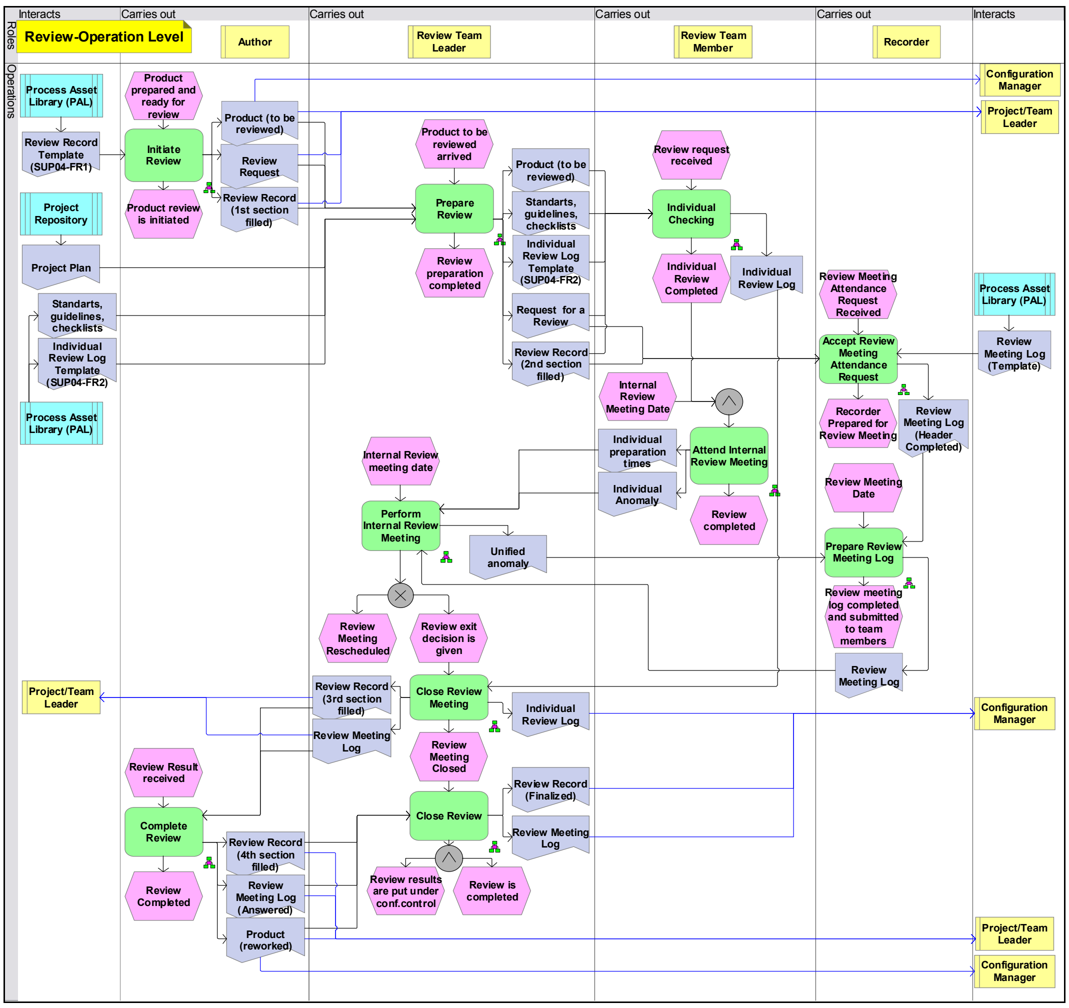

Interacts
Carries out
Carries out
Carries out
Carries out
Interacts
Roles
Review-Operation Level
Author
Review Team Leader
Review Team Member
Recorder
Operations
Process Asset Library (PAL)
Review Record Template (SUP04-FR1)
Project Repository
Project Plan
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Process Asset Library (PAL)
Product prepared and ready for review
Initiate Review
Product review is initiated
Product (to be reviewed)
Review Request
Review Record (1st section filled)
Product to be reviewed arrived
Prepare Review
Review preparation completed
Product (to be reviewed)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Request for a Review
Review Record (2nd section filled)
Review request received
Individual Checking
Individual Review Completed
Individual Review Log
Internal Review Meeting Date
Attend Internal Review Meeting
Individual preparation times
Individual Anomaly
Review completed
Configuration Manager
Project/Team Leader
Review Meeting Attendance Request Received
Accept Review Meeting Attendance Request
Recorder Prepared for Review Meeting
Review Meeting Log (Header Completed)
Process Asset Library (PAL)
Review Meeting Log (Template)
Review Meeting Date
Prepare Review Meeting Log
Review meeting log completed and submitted to team members
Review Meeting Log
Internal Review meeting date
Perform Internal Review Meeting
Unified anomaly
Review Meeting Rescheduled
Review exit decision is given
Project/Team Leader
Review Record (3rd section filled)
Review Meeting Log
Close Review Meeting
Individual Review Log
Review Meeting Closed
Configuration Manager
Review Result received
Complete Review
Review Completed
Review Record (4th section filled)
Review Meeting Log (Answered)
Product (reworked)
Close Review
Review results are put under conf.control
Review is completed
Review Record (Finalized)
Review Meeting Log
Project/Team Leader
Configuration Manager

Figure 117. Review-Operation Level

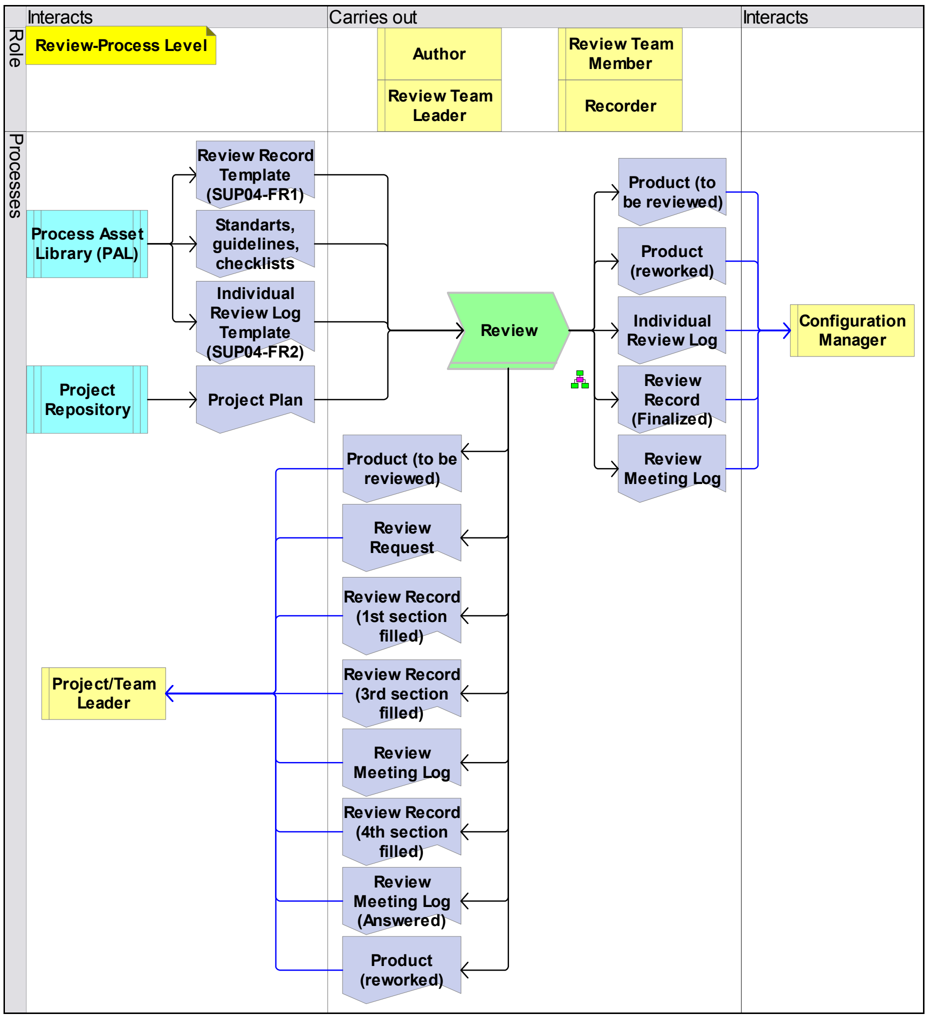


Figure 118. Review-Process Level

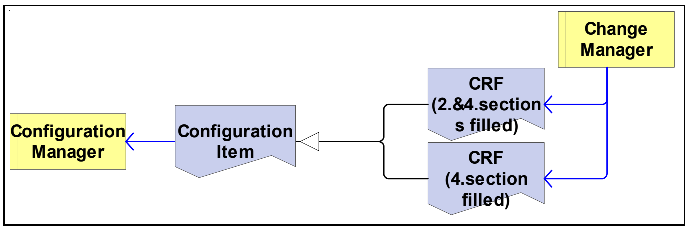


Figure 119. Role Dependency - Change Manager & Conf. Manager

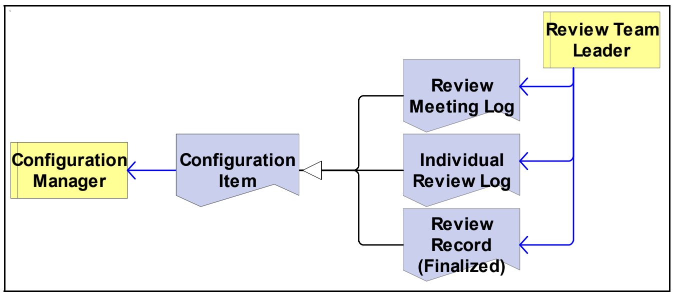


Figure 120. Role Dependency - Review Team Leader & Conf. Manager

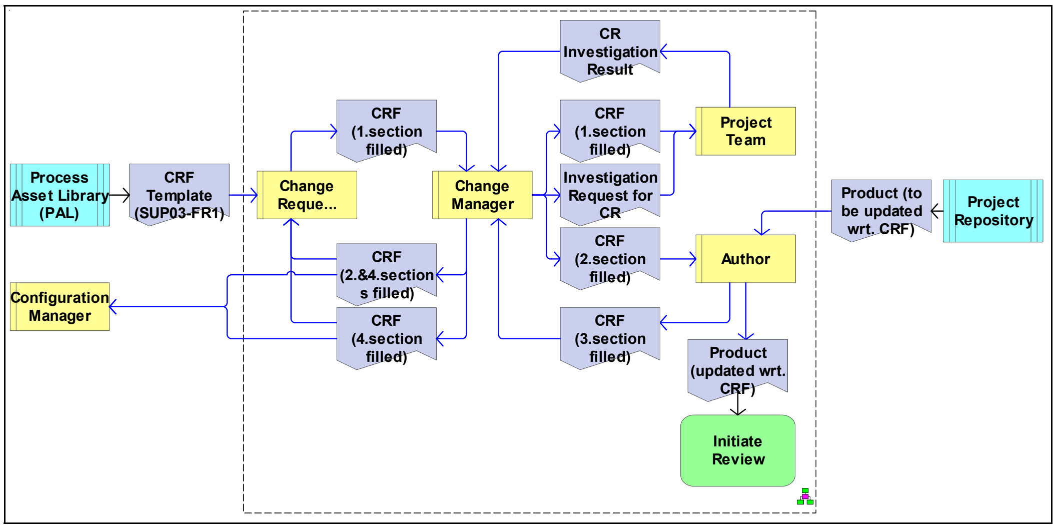
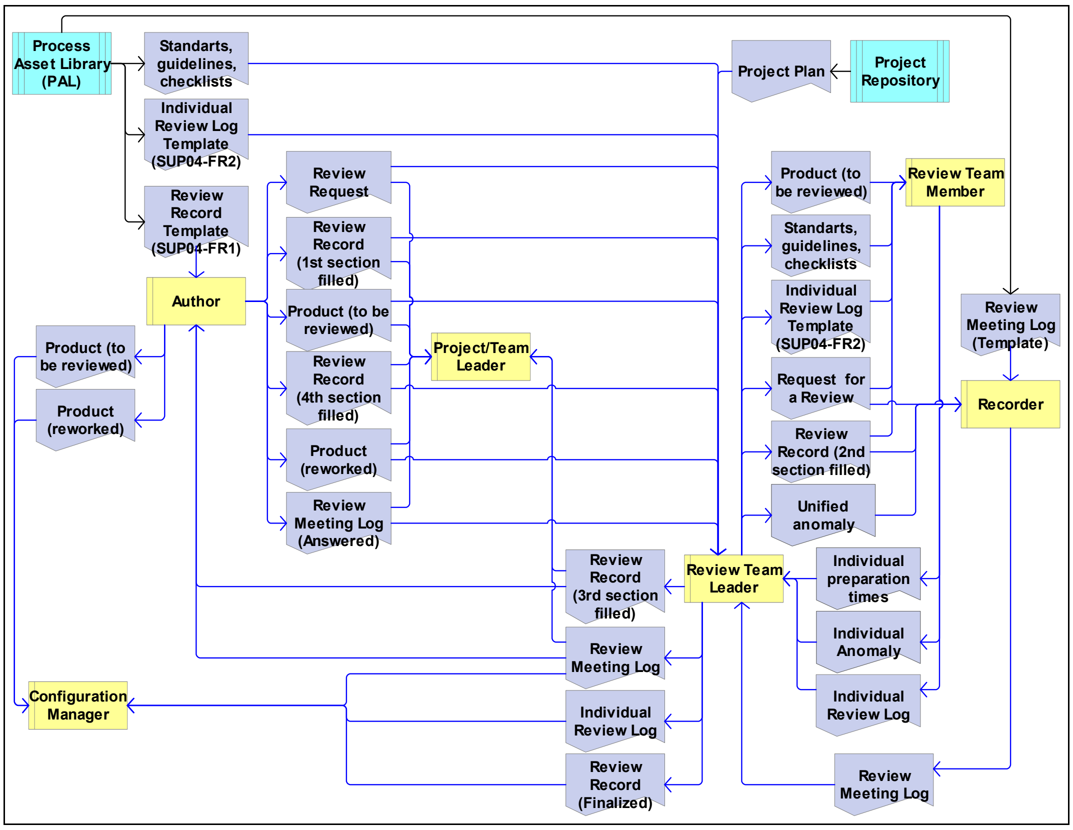


Figure 121. Role Dependency - Manage Change



Process Asset Library (PAL)
Standarts, guidelines, checklists
Project Plan
Project Repository
Individual Review Log Template (SUP04-FR2)
Review Request
Product (to be reviewed)
Review Team Member
Review Record Template (SUP04-FR1)
Review Record (1st section filled)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Review Meeting Log (Template)
Author
Product (to be reviewed)
Product (to be reviewed)
Project/Team Leader
Request for a Review
Recorder
Review Record (4th section filled)
Product (reworked)
Product (reworked)
Review Record (2nd section filled)
Review Meeting Log (Answered)
Unified anomaly
Review Record (3rd section filled)
Review Team Leader
Individual preparation times
Review Meeting Log
Individual Anomaly
Configuration Manager
Individual Review Log
Individual Review Log
Review Record (Finalized)
Review Meeting Log

Figure 122. Role Dependency - Review

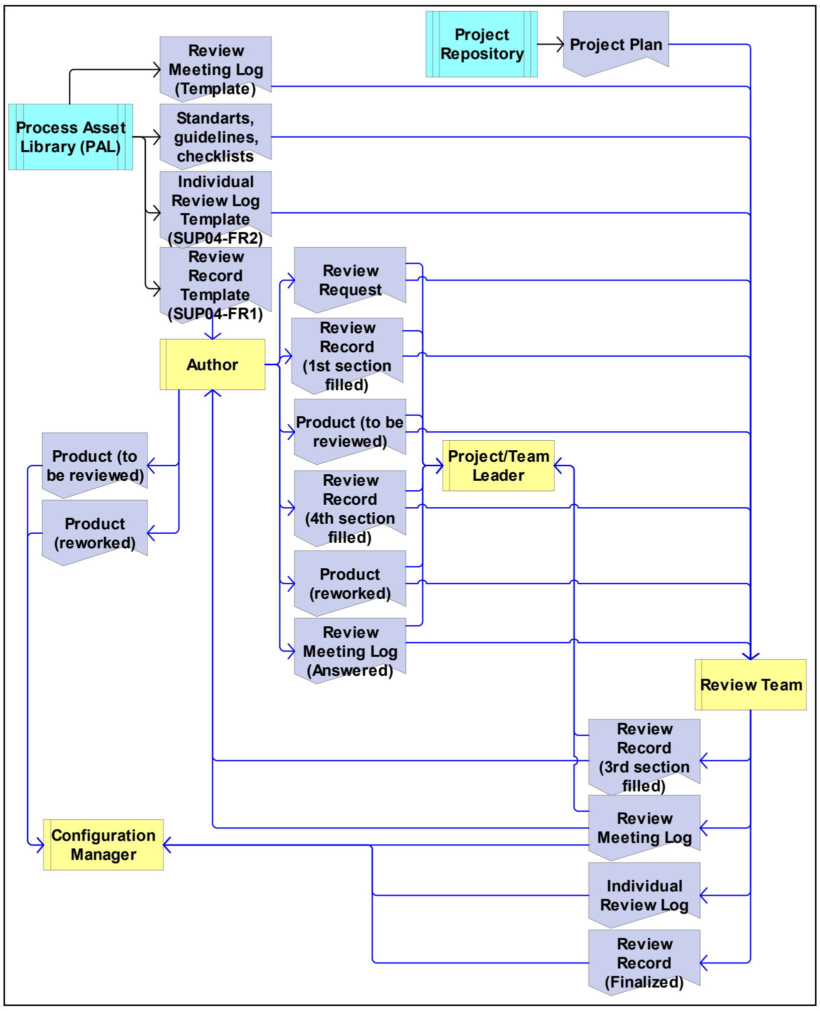

Project Repository
Project Plan
Review Meeting Log (Template)
Process Asset Library (PAL)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Review Record Template (SUP04-FR1)
Review Request
Review Record (1st section filled)
Author
Product (to be reviewed)
Project/Team Leader
Product (to be reviewed)
Review Record (4th section filled)
Product (reworked)
Product (reworked)
Review Meeting Log (Answered)
Review Team
Review Record (3rd section filled)
Review Meeting Log
Configuration Manager
Individual Review Log
Review Record (Finalized)

Figure 123. Role Dependency-Review (Aggregated on Review Team)

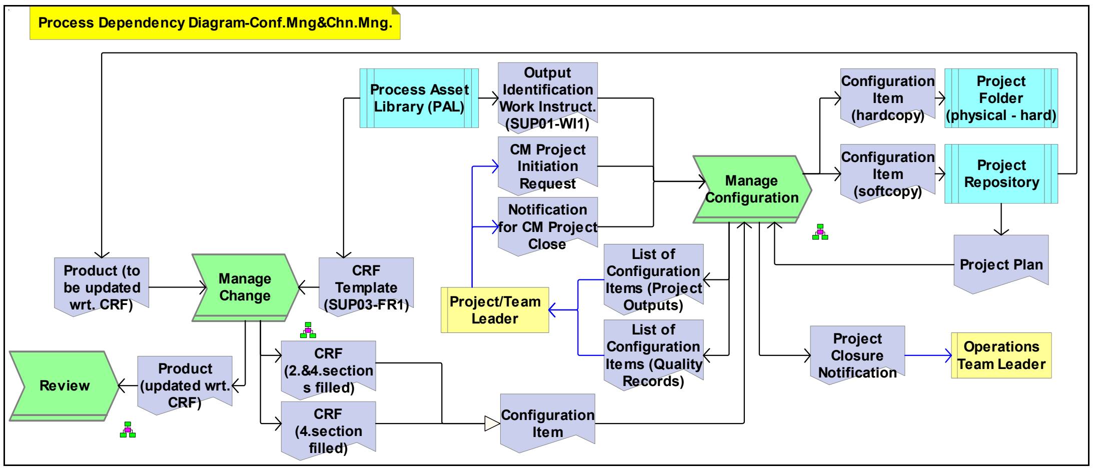

Process Dependency Diagram-Conf.Mng&Chn.Mng.
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
CM Project Initiation Request
Notification for CM Project Close
Manage Configuration
Configuration Item (hardcopy)
Project Folder (physical - hard)
Configuration Item (softcopy)
Project Repository
Project Plan
Product (to be updated wrt. CRF)
Manage Change
CRF Template (SUP03-FR1)
Project/Team Leader
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
Project Closure Notification
Operations Team Leader
Review
Product (updated wrt. CRF)
CRF (2.&4.sections filled)
CRF (4.section filled)
Configuration Item

Figure 124. Process Dependency Diagram- Conf. Manager & Change Mng.

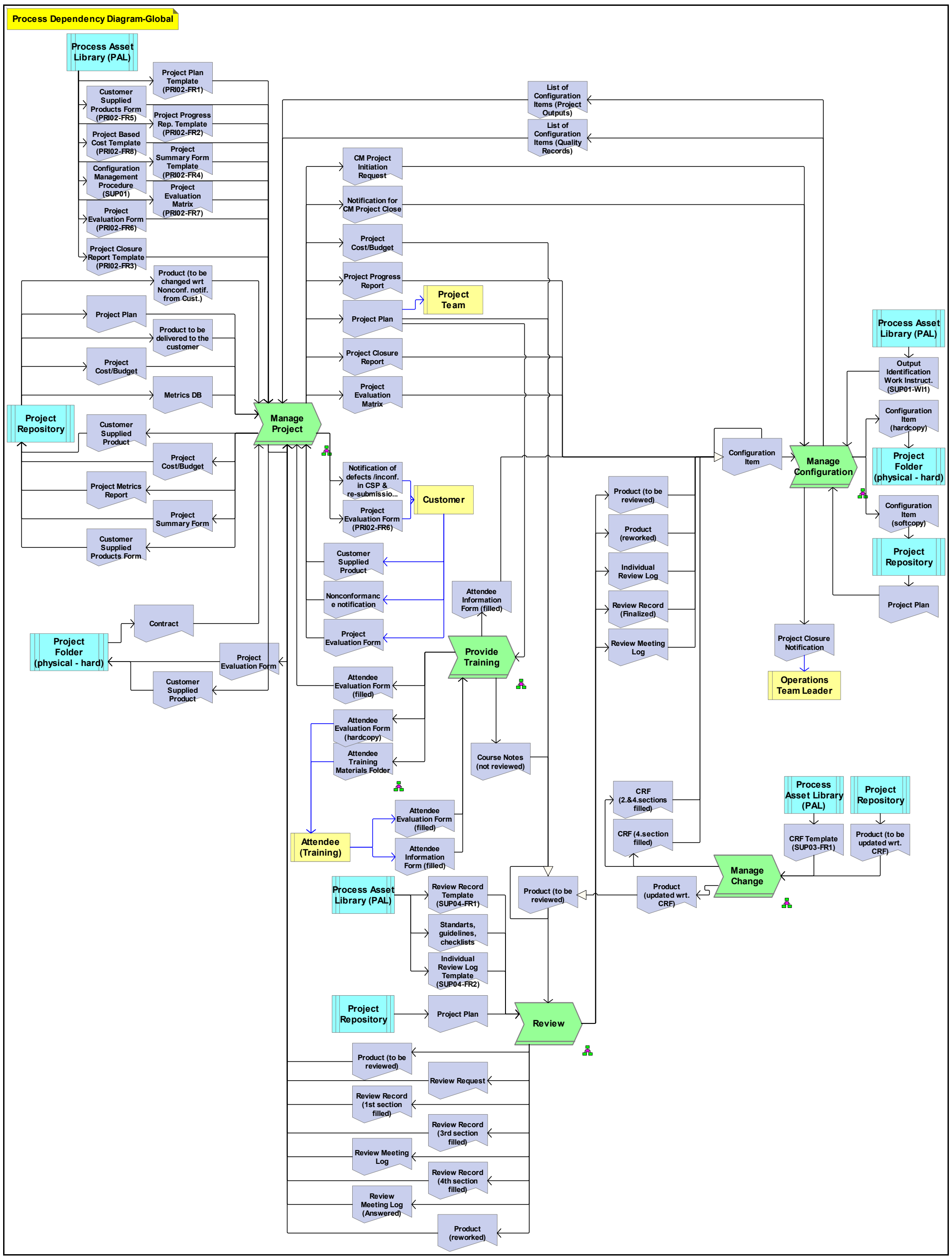

Process Dependency Diagram-Global
Process Asset Library (PAL)
Project Plan Template (PRI02-FR1)
Customer Supplied Products Form (PRI02-FR5)
Project Progress Rep. Template (PRI02-FR2)
Project Based Cost Template (PRI02-FR8)
Project Summary Form Template (PRI02-FR4)
Configuration Management Procedure (SUP01)
Project Evaluation Matrix (PRI02-FR7)
Project Evaluation Form (PRI02-FR6)
Project Closure Report Template (PRI02-FR3)
Product (to be changed wrt Nonconf. notif. from Cust.)
Project Plan
Product to be delivered to the customer
Project Cost/Budget
Metrics DB
Project Repository
Customer Supplied Product
Manage Project
Project Cost/Budget
Project Metrics Report
Project Summary Form
Customer Supplied Products Form
Contract
Project Folder (physical - hard)
Project Evaluation Form
Customer Supplied Product
List of Configuration Items (Project Outputs)
List of Configuration Items (Quality Records)
CM Project Initiation Request
Notification for CM Project Close
Project Cost/Budget
Project Progress Report
Project Team
Project Plan
Project Closure Report
Project Evaluation Matrix
Notification of defects /inconf. in CSP & re-submissio...
Customer
Project Evaluation Form (PRI02-FR6)
Customer Supplied Product
Nonconformance notification
Project Evaluation Form
Attendee Information Form (filled)
Provide Training
Attendee Evaluation Form (filled)
Attendee Evaluation Form (hardcopy)
Attendee Training Materials Folder
Course Notes (not reviewed)
Attendee Evaluation Form (filled)
Attendee (Training)
Attendee Information Form (filled)
Configuration Item
Manage Configuration
Process Asset Library (PAL)
Output Identification Work Instruct. (SUP01-WI1)
Configuration Item (hardcopy)
Project Folder (physical - hard)
Configuration Item (softcopy)
Project Repository
Project Plan
Product (to be reviewed)
Product (reworked)
Individual Review Log
Review Record (Finalized)
Review Meeting Log
Project Closure Notification
Operations Team Leader
CRF (2.&4.sections filled)
CRF (4.section filled)
Process Asset Library (PAL)
Project Repository
CRF Template (SUP03-FR1)
Product (to be updated wrt. CRF)
Manage Change
Product (updated wrt. CRF)
Product (to be reviewed)
Process Asset Library (PAL)
Review Record Template (SUP04-FR1)
Standarts, guidelines, checklists
Individual Review Log Template (SUP04-FR2)
Project Repository
Project Plan
Review
Product (to be reviewed)
Review Request
Review Record (1st section filled)
Review Record (3rd section filled)
Review Meeting Log
Review Record (4th section filled)
Review Meeting Log (Answered)
Product (reworked)

Figure 125. Process Dependency Diagram-Global

APPENDIX F: THE QUESTIONAIRE AND ANSWERS

The answer to the questions and their frequency for the second case study is given next to the shaded boxes.

Questionnaire

1. I have previously involved in a process modeling project, workshop, and etc.

a) Many times	1
b) Couple of times	3
c) Once	
d) Never	

2. How can you describe your degree of experience and competency about your responsibilities?

a) I am highly experienced	2
b) I am skilled and I rarely request help from my colleagues	2
c) I am not a beginner but I usually request help from colleagues and refer to quality, guideline or user manuals.	
d) I am a novice	

3. Did your organization go through a process improvement or similar undertaking in the last 8 years?

a) Yes	4
b) No	

4. How can you describe your degree of experience and competency about process modeling?

a) Highly experienced	
b) Skilled	3
c) Moderate	1
d) Novice	

5. Does your organization have predefined processes?

a) Yes	4
b) Partially	
c) No	

IF the answer is ‘a’ or ‘b’:

6. How accurately is the predefined process being followed?

a) Always	
b) Usually	3
c) Sometimes	1
d) Seldom	
e) Never	

7. To what extent did you get involved in the definition process?

a) Participated actively in the definition as a member of the process engineering team	2
b) Participated in the definition for the parts related to my responsibilities	
c) Provided input and feedbacks to the definition	
d) Never been asked or given responsibility	
d) Not an employee at that time	2

IF ‘No’:

8. Do you feel like you could have supplied valuable feedback for the activities that affects you?

a) Yes	2
b) No	

9. Do you believe that you have responsibility and authority to model and improve processes that affect you?

a) Yes	3
b) Not always	1
e) No	

10. Do you believe that you should have a responsibility and authority to model and improve processes that affect you?

a) Yes	4
b) Not always	
e) No	

11. Are you encouraged to make suggestions for changes in the processes in your organization?

a) Yes	3
b) Not always	1
e) No	

		Strongly Agree	Agree	Fairly	Disagree	Strongly Disagree
	General Effectiveness and Improvement					
12	Modeling the processes increased my awareness and gave me a better understanding of my activities and dependencies to others.	3	1			
13	Explicitly modeling the exchange of information between roles provided useful information.	4				
14	I can utilize another commonly used modeling method that would provide me similar or more degree of understanding of the processes.			4		
15	Explicitly modeling the transfer of information items between roles increased the modeling effort.	1		3		
16	Explicitly modeling the transfer of information items between roles did not add much value to the completeness of the model.				4	
17	Role dependency models reflected useful information about role's expectations and their dependencies. If disagree; what other dependencies can be represented?	2	2			
18	I feel I am more aware of the way I perform my activities.	3	1			
19	Analysis models (integrated models, dependency models) gave useful information about how our organization performs its processes. What other views (models) do you think can also be generated that would give insight into the way the organization works?		4			
20	Visible impact analysis on the changes related to roles' expectations helps modifying the processes.	2	2			
	The Method					
21	I found it difficult to scope and explicitly identify the processes to be covered			1	3	
22	I found it difficult to identify the roles participating in each process			3	1	
23	I found it hard to follow the procedure for modeling our activities.				4	
24	I found it difficult to identify and isolate the operations my role service to the others.			1	3	
25	I found it difficult to identify and describe the details of how I perform my operations.				4	
26	I found it difficult to identify the required information items (documents, messages, emails, and etc.) I use and produce.			3	1	
27	I found it difficult to identify the roles or other sources that I interact with.		2	2		

		Strongly Agree	Agree	Fairly	Disagree	Strongly Disagree
28	I wasted quite much time for understanding other roles' expectations from my role.		2	2		
29	When checking others expectations from my role, I realized a couple of items are missing in my definitions.		3	1		
30	There were several information items that were named differently by other agents which were in fact conceptually the same.		2	2		
31	There were a couple of cases where conflicts between my expectations and others arisen.		1	1	2	
32	For many times, I needed to change my process definition based on the expectations of others.		4			
33	Integrating individual models are necessary. Why?		4			
34	Generating role and process dependencies are necessary. Why?		4			
	The Modeling Notation					
35	The process modeling notation was rather difficult to *learn*.		1	2	1	
36	The process modeling notation was rather difficult to *use*.		3	1		
37	The notation was adequate to represent essential information I have about my activities.		4			
38	Individual models represented too many details about the processes.			2	2	
39	The models were accurate representations of the processes.	1	3			
40	Integrated process models (the activity level) were complex and hard to follow.		3	1		
41	Integrated process models (the process and operation level) were complex and hard to follow.			1	3	
42	Role and process dependency models were hard to follow.			1	3	
43	There were other properties in the process that you wanted to represent but were not captured by the model. If there were, what were they?			1	3	

APPENDIX G: THE PLURAL ADD-ON: SOURCE CODE AND INSTALLATION

This appendix provides the installation procedure of the Plural add-on developed as an extension to the ARIS Collaborative Suite. The add-on comprises a set of database components residing in the ARIS database and a web-site that connects to the database, analyzes and depicts related content based on user preferences and inquiry.

The extensions and the source code are provided in an accompanying CD submitted with this thesis manuscript.

The following list of activities gives the details of the procedure to be followed in order to set up the necessary toolset to be used for the Plural method.

1. ARIS Collaborative Suite and Database Installation

The versions of the related software programs that are installed and tested are as follows (on Microsoft Windows XP, Service Pack 2):

Extended Software	**Version**
ARIS Collaborative Suite (Client and Server Components)	6.23, 6.1
Oracle Database	10.1.0.2.0

Refer to related software program’s installation guides for the details of the settings, hardware and software requirements and other related details.

2. Creating a Database with ARIS Toolset

With ARIS Toolset, a database (database schema) that will store all model information should be created. Refer to ARIS Toolset Help for creating databases.

3. Installing Database Components

Related components that should be created in the database are listed below. These components can be created by executing the SQL (structure query language) script given in the accompanying CD labeled as “Source Code for Database Components.pdf”. The script can be executed in any tool that connects to the database with a user that has appropriate privileges (e.g. ARISADMIN62 - arisadmin).

4. Placing Website Components

A web server software (e.g.) that can run 'VBScript' files, should be installed in a server where related Plural add-on components will reside. The plural files and components are given in the accompanying CD as a compressed file labeled as 'plural_web_site.zip'. Uncompress the file and place all files into an appropriate web server folder and start the web service.

After the installation of related components, the Plural add-on can be reached via a web browser with the following address:

http://<the web address of the server>/plural/index.asp

5. Extending ARIS Collaborative Suite

In order to provide support for the Plural notation and its diagrams, and maintain a standard modeling practice, ARIS Collaborative Suite is extended with a method filter and a template that are utilized for the diagrams. Method filters restrict the users to use a specific set of diagrams and process elements as well as the relationships between these elements. Templates help maintaining a convention among the visualization of diagrams – the figures, icons, related colors, and etc.

The filter and template is provided in the accompanying CD under the root folder APPENDIX G'\ with the following file names:

- Plural Filter.amc
- Plural Template.act

Refer to ARIS Toolset Help for installing and applying method filters and templates.

VITA

Oktay Türetken was born in Antakya, Turkey in 1975. He received his bachelor degree in Industrial Engineering in Middle East Technical University (METU) in 1998. In 2001, he had his M.S. degree in Information Systems in the Informatics Institute, METU.

During 1998 and 2005, he worked as a research assistant in Industrial Engineering Department and Informatics Institute. During that period he participated in number of research projects and assisted many courses related to information systems and software engineering. He participated in ERP (enterprise resource planning) system implementation projects as a consultant, worked in software intensive system specification/acquisition projects for C4ISR systems for Turkish Land Forces Command and projects on the development of a conceptual modeling tool for modeling and simulation of C4ISR Systems for Undersecretariat for Defence Industries. Since 2005, he is working for Bilgi Grubu Ltd. as a researcher and consultant.

His research interests include; process improvement, business process modeling and requirements engineering.