



**YER SİSTEM MODELLEMESİ İÇİN İŞ AKIŞ  
TASARIM VE YÖNETİM SİSTEMİ**

**DOKTORA TEZİ**

**Ufuk Utku TURUNÇOĞLU**

**Anabilim Dalı : Hesaplamalı Bilim ve Mühendislik**

**Programı : Hesaplamalı Bilim ve Mühendislik**

**Tez Yöneticisi : Prof. Dr. Nüzhet DALFES**

**Aralık 2009**

## ÖNSÖZ

Bu tez konusunun şekillenmesine ve uygulanmasına bilgi ve tecrübeleri ile katkıda bulunan danışmanım Prof. Dr. Nüzhet Dalfes'e, çalışmamda bana yol gösteren başta Sylvia Murphy, Cecelia DeLuca, Fei Lui ve Ryan O'Kuinghttons olmak üzere tüm ESMF (Earth System Modeling Framework) projesi geliştiricilerine, CCSM konusundaki tecrübelerini benimle paylaşan ve bu konudaki sorunlarımı çözmek için zaman ayıran Mariana Vertenstein ve Brian Eaton'a, daha önce model kuple edilmesi konusunda yaptığı çalışmalara ait sonuçları benimle paylaşan Shaowu Bao'ya, WRF modeli ve ESMF-IO yapısı hakkında bilgi ve yardımları ile destek olan John Michalakes ve Qiu Xin'e ve CCSM'in POP bileşeni ve SCRIP uygulaması konusundaki tecrübelerini benimle paylaşan Jon Wolfe'ye yardımları için teşekkür ederim.

Ayrıca, bu tezin gerçekleşmesindeki sonsuz destekleri, sabırları ve anlayışları için başta sevgili eşim Melda Turunçoğlu olmak üzere tüm aileme teşekkür ederim.

Bu tez, İstanbul Teknik Üniversitesi "İleri Teknolojiler Projesi" tarafından desteklenmiştir. Bu çalışmanın bir bölümünde hesaplama kaynağı olarak NSF (National Science Foundation) ve NCAR (National Center for Atmospheric Research) tarafından sağlanan TeraGrid grid hesaplama kaynağı kullanılmıştır. TeraGrid, Indiana Üniversitesi, LONI, NCAR, NCSA, NICS, ORNL, PSC, Purdue University, SDSC, TACC ve UC/ANL merkezleri tarafından oluşturulmaktadır. Bu çalışmada kullanılan hesaplama zamanının bir kısmı NSF MRI Grant #CNS-0421498, NSF MRI Grant #CNS-0420873, NSF MRI Grant #CNS-0420985 projeleri tarafından NCAR, Colorado Üniversitesi ve IBM Shared University Research (SUR) programı tarafından sağlanmıştır.

Aralık 2009

Ufuk Utku TURUNÇOĞLU

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖNSÖZ</b> . . . . .	<b>ii</b>
<b>İÇİNDEKİLER</b> . . . . .	<b>iii</b>
<b>KISALTMALAR</b> . . . . .	<b>v</b>
<b>ŞEKİLLER LİSTESİ</b> . . . . .	<b>vii</b>
<b>ÖZET</b> . . . . .	<b>ix</b>
<b>SUMMARY</b> . . . . .	<b>1</b>
<b>1. GİRİŞ</b> . . . . .	<b>2</b>
1.1. Yer Sistem Biliminde Karşılaşılan Problemler ve Amaç . . . . .	2
1.2. Çözüm Yöntemleri ve Tarihsel Gelişim . . . . .	4
1.3. Ön Veri ve Köken Bilgisinin Yer Sistem Bilimlerindeki Önemi . . . . .	8
<b>2. TEMEL TANIMLAR VE METODOLOJİ</b> . . . . .	<b>11</b>
2.1. Köken Bilgisi ve Tipleri . . . . .	11
2.2. Ön Veri . . . . .	12
2.3. Ontoloji Kavramı ve Yer Sistem Bilimlerinde Kullanımı . . . . .	13
2.4. Önerilen Metodoloji . . . . .	16
<b>3. KULLANILAN ARAÇLAR</b> . . . . .	<b>21</b>
3.1. İş Akışı Uygulaması: Kepler . . . . .	21
3.2. Çerçeve Uygulaması: ESMF . . . . .	24
3.3. Hesaplama Kaynağı: TeraGrid ve Küme Sistemler . . . . .	28
<b>4. UYGULAMA ÖRNEĞİ - I</b> . . . . .	<b>32</b>
4.1. Küresel Dolaşım Modeli: CCSM4 . . . . .	32
4.2. Test Uygulaması ve Amaç . . . . .	33
4.3. Köken ve Ön Veri Bilgisinin CCSM Modelinden Toplanması . . . . .	34
4.4. CCSM Modeli Üzerinde Yapılan Değişiklikler . . . . .	39
4.5. Geliştirilen Aktörler ve Tasarlanan İş Akışı . . . . .	41
4.6. Uygulama Örneği Sonuçları . . . . .	49
<b>5. UYGULAMA ÖRNEĞİ - II</b> . . . . .	<b>50</b>
5.1. Kuple Edilmiş Model Sistemi: ROMS ve WRF . . . . .	50
5.2. WRF Sınırlı Alan Atmosfer Modeli . . . . .	51
5.3. ROMS Okyanus Modeli . . . . .	52
5.4. Test Uygulaması . . . . .	52
5.5. Geliştirilen Araçlar . . . . .	54
5.6. SCRIP Uygulaması ve Yapılan Değişiklikler . . . . .	56
5.7. Modellerin Kuple Edilmesi . . . . .	59
5.8. Geliştirilen Aktörler ve Tasarlanan İş Akışı . . . . .	62
5.9. Uygulama Örneği Sonuçları . . . . .	70
<b>6. SONUÇLAR</b> . . . . .	<b>72</b>

<b>KAYNAKLAR</b> . . . . .	<b>75</b>
<b>EKLER</b> . . . . .	<b>80</b>
<b>ÖZGEÇMİŞ</b> . . . . .	<b>90</b>

## KISALTMALAR

<b>AR4</b>	: IPCC Forth Assessment Report
<b>AR5</b>	: IPCC Fifth Assessment Report
<b>CAM</b>	: Community Atmosphere Model
<b>CCSM</b>	: Community Climate System Model
<b>CIM</b>	: Common Information Model
<b>CMIP</b>	: Coupled Model Intercomparison Project
<b>CLM</b>	: Community Land Model
<b>CTSS</b>	: Coordinated TeraGrid Software and Services
<b>DyCore</b>	: Colloquium on Numerical Techniques for Global Atmospheric Models
<b>FMS</b>	: Flexible Modeling System
<b>ENES</b>	: European Network for Earth System Modeling
<b>ESC</b>	: Earth System Curator
<b>ESG</b>	: Earth System Grid
<b>ESMF</b>	: Earth System Modeling Framework
<b>FOL</b>	: First Order Logic
<b>GRIB</b>	: GRIdded Binary
<b>HDF</b>	: Hierarchical Data Format
<b>IPCC</b>	: Intergovernmental Panel on Climate Change
<b>KIF</b>	: Knowledge Interchange Format
<b>LEAD</b>	: Linked Environments for Atmospheric Discovery
<b>MCT</b>	: Model Coupling Toolbox
<b>MCEL</b>	: Model Coupling Environmental Library
<b>METAFOR</b>	: Common Metadata for Climate Modeling Digital Repositories
<b>MoC</b>	: Model of Computation
<b>MPI</b>	: Message Passing Interface
<b>MPMD</b>	: Multiple Program and Multiple Data
<b>NetCDF</b>	: Network Common Data Form
<b>OASIS</b>	: Ocean Atmosphere Sea Ice Soil
<b>OWL</b>	: Web Ontology Language
<b>PET</b>	: Persistent Execution Thread
<b>POP</b>	: Parallel Ocean Program
<b>PRISM</b>	: Program for Integrated Earth System Modeling
<b>RDF</b>	: Resource Description Framework
<b>RDFS</b>	: Resource Description Framework Schema
<b>ROMS</b>	: Regional Ocean Modeling System
<b>SCE</b>	: Standard Compile Environment
<b>SCRIP</b>	: Spherical Coordinate Remapping and Interpolation Package
<b>SPMD</b>	: Single Program and Multiple Data
<b>SRE</b>	: Standard Running Environment

- W3C** : World Wide Web Consortium
- WRF** : Weather Research and Forecasting
- WSS** : Web Services System

## ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 1.1 : İklim sisteminde karakteristik zaman ölçekleri (www.dkrz.de) . . .	3
Şekil 2.1 : İş akış sisteminin genel yapısı ve bileşenleri . . . . .	17
Şekil 2.2 : Yeni iş akışının otomatik olarak tetiklenmesi için gereken adımlar	20
Şekil 3.1 : Kepler uygulamasının genel görünümü ve içerdiği bileşenler . . .	22
Şekil 3.2 : ESMF uygulamasının sandviç yapısının şematik gösterimi . . . .	26
Şekil 3.3 : ESMF’i oluşturan temel sınıflar . . . . .	28
Şekil 4.1 : Köken bilgisinin toplanması için geliştirilen hiyerarşik yapı . . .	35
Şekil 4.2 : Örnek sistem köken bilgisi çıktısı (NCAR Frost, IBM BG/L) . .	36
Şekil 4.3 : CCSM modelinin atmosferik bileşeninden (CAM) toplanmış örnek sistem köken bilgisi çıktısı (NCAR Frost, BG/L) . . . . .	38
Şekil 4.4 : Örnek veri köken bilgisi çıktısı (CCSM Kuple bileşeni, atmosferik model değişkenleri) . . . . .	41
Şekil 4.5 : CCSM modeli mantıksal iş akışı tasarımı . . . . .	42
Şekil 4.6 : CCSM iş akışı için tasarlanmış aktörler . . . . .	42
Şekil 4.7 : CDO uygulaması için geliştirilmiş OWL bilgi tabanı genel sınıf yapısı . . . . .	44
Şekil 4.8 : CCSM modeli için geliştirilmiş Kepler üst seviye iş akışı . . . . .	45
Şekil 4.9 : Şekil 4.8’te görülen birleşik aktörün iç yapısı . . . . .	46
Şekil 4.10 : CCSM iş akışının zaman çizelgesi üzerine aktarılmış hali . . . . .	47
Şekil 4.11 : TeraGrid CCSM alt iş akışının Bluefire küme sistemine aktarılmış hali . . . . .	47
Şekil 4.12 : CCSM iş akışı tarafından üretilen ve e-posta olarak gönderilen XML dosyası . . . . .	48
Şekil 4.13 : CCSM iş akışı tarafından otomatik olarak tetiklenen iş akışı . . .	48
Şekil 5.1 : WRF modeli alanı ve yükselti verisi . . . . .	53
Şekil 5.2 : ROMS modeli alanı ve derinlik verisi . . . . .	54
Şekil 5.3 : Test uygulamasının genel yapısı . . . . .	55
Şekil 5.4 : SCRIP uygulaması tarafından desteklenen interpolasyon tipleri .	57
Şekil 5.5 : SCRIP uygulaması test sonuçları: (a) ROMS gridindeki test verisi, (b) WRF gridine interpolasyon ile aktarılmış veri, (c) hata. Test fonksiyonu $f = 2 + \sin^{16}(2\theta) \cos(16\phi)$ küresel harmonik, $l =$ $32, m = 16$ . . . . .	58
Şekil 5.6 : Kuple modelleme sisteminin yapısı . . . . .	61
Şekil 5.7 : Kuple (WRF ve ROMS) edilmiş model sistemine ait mantıksal iş akışı tasarımı . . . . .	63
Şekil 5.8 : WRF iş akış tasarımı . . . . .	64
Şekil 5.9 : WRF iş akış tasarımı (CCSM2WRF kompozit aktörü) . . . . .	65
Şekil 5.10 : WRF iş akışının zaman çizelgesi üzerine aktarılmış hali . . . . .	65
Şekil 5.11 : ROMS iş akış tasarımı . . . . .	66
Şekil 5.12 : Kuple model sistemi için iş akış tasarımı . . . . .	67



<b>Şekil 5.13</b> : ROMS modeli örnek sistem köken bilgisi çıktısı (NCAR Bluefire, IBM Küme) . . . . .	68
<b>Şekil 5.14</b> : ROMS örnek veri köken bilgisi çıktısı . . . . .	69
<b>Şekil 5.15</b> : WRF örnek veri köken bilgisi çıktısı . . . . .	70
<b>Şekil B.1</b> : Sistem köken bilgisini toplayan betiğin yapısı . . . . .	85
<b>Şekil D.1</b> : Kuple ve tek olarak çalıştırılmış modellerin deniz yüzeyi sıcaklığı karşılaştırmaları (01-07-1999 00:00). ROMS modeli (a ve b) için değerler en üst sigma seviyesinden alınmıştır. . . . .	87
<b>Şekil D.2</b> : Kuple ve tek olarak çalıştırılmış modellerin deniz yüzeyi sıcaklığı karşılaştırmaları (15-07-1999 00:00). ROMS modeli (a ve b) için değerler en üst sigma seviyesinden alınmıştır. . . . .	87
<b>Şekil D.3</b> : Kuple ve tek olarak çalıştırılmış okyanus modelinin yüzey akıntıları karşılaştırması (15-07-1999 00:00). . . . .	88
<b>Şekil D.4</b> : Kuple ve tek olarak çalıştırılmış atmosfer modelinin 2 metre yüzey sıcaklığı ve karışım oranı karşılaştırmaları (15-07-1999 00:00). . .	88
<b>Şekil D.5</b> : Kuple ve tek olarak çalıştırılmış atmosfer modelinin yağış ve gizli ısı karşılaştırması (15-07-1999 00:00). . . . .	89

# YER SİSTEM MODELLEMESİ İÇİN İŞ AKIŞ TASARIM VE YÖNETİM SİSTEMİ

## ÖZET

Bilimsel iş akış sistemlerinin karmaşık bileşenler içeren çalışmalarda kullanılması son zamanlarda oldukça popüler bir araştırma alanıdır. Özellikle yer sistem bilimleri açısından bakıldığında, birbirlerine sıkı bir şekilde bağlı, kuple edilmiş ve birçok farklı bileşen içeren yer sistem modellerine ait alt süreçlerin ve birbirleri ile olan ilişkilerin basitleştirilmesi için bilimsel iş akış sistemleri kullanılabilir.

Bu çalışmanın asıl motivasyon kaynağı, karmaşık modelleme sistemlerinin yüksek başarılı hesaplama sistemlerinde çalıştırılmasının gittikçe zorlaşması ve varolan karmaşıklığın ortalama bir kullanıcı seviyesine indirgenmesine olan ihtiyaçtır.

Bu çalışma ile bilimsel iş akışı ve çerçeve yaklaşımlarının birleştirilerek standartlaştırılmış bir çalışma ortamının yaratılması için gerekli yeni bir metodoloji geliştirilmiş ve tanımlanmıştır. Tanımlanan bu metodoloji, iki farklı ve gerçekçi yer sistem uygulaması ile sınanmış ve sonuçları analiz edilmiştir.

İlk uygulama örneği; bir küresel dolaşım modelinin hem bir hesaplama gridi (TeraGrid), hem de bir küme sistem üzerinde, kullanılan modeli ve hesaplama ortamını anlamlı bir şekilde soyutlaştırılarak çalıştırılması ve sonuçların analiz edilmesi süreçlerini içermektedir. Bu amaçla, henüz geliştirilme aşamasında olan CCSM modelinin son sürümü kullanılmıştır. Bu örnek ile daha önce yer sistem modelleri ile uygulaması olmayan bir şekilde köken bilgisi toplanarak, model ve sistem ile ilgili ön bilginin otomatik bir şekilde üretilmesine katkıda bulunulmuştur. Bu şekilde model sonuçlarının tekrar edilmesi ve sonradan detaylı bir şekilde incelenmesinin önündeki zorluklar kaldırılmıştır.

İkinci uygulama örneği ise; çerçeve yaklaşımı kullanılarak kuple edilmiş bölgesel bir iklim modeli sisteminin, Akdeniz bölgesi için tanımlanarak iş akış sistemine entegre edilmesini ve bu şekilde bölge iklim sisteminin daha gerçekçi bir şekilde benzetiminin yapılmasını amaçlamaktadır. Bu test uygulamasının, küresel dolaşım modeli sonuçlarını Türkiye ve bölgesi üzerine dinamik ölçek küçültme yöntemi kullanılarak aktarılması çalışmaları açısından bakıldığında ayrı bir önemi vardır. Birinci uygulamada olduğu gibi, bu uygulama örneğinde de köken bilgisinin toplanarak sonuçların tekrar edilmesi, karşılaştırılması ve hataların ayıklanması süreçlerine katkıda bulunulması amaçlanmaktadır.

Sonuçlar, geliştirilen iş akış çalışma ortamının farklı yer sistem modelleri ve farklı yüksek başarılı hesaplama kaynakları üzerinde, anlamlı bir soyutlama seviyesinde çalıştırılabildiğini ve toplanan köken bilgisinin, yapılan çalışmanın evriminin kayıt altına alınması açısından son derece yararlı olduğunu ortaya koymuştur.

# **WORKFLOW DESIGN AND MANAGEMENT SYSTEM FOR EARTH SYSTEM MODELLING**

## **SUMMARY**

The application of scientific workflow systems for orchestrating complex tasks is still an open research area. In particular, earth system related applications consists of different tasks that are closely related to each other and scientific workflow systems can be used to simplify these sub processes and their relationships.

The motivation for this work is driven by the complexities of running a large modeling system on a high performance network and the need to reduce those complexities, particularly for the average user.

In this study, it is presented and analyzed a new methodology to combine scientific workflow and modeling framework approach together to create a standardized work environment. Then, the proposed methodology is tested using two typical and realistic earth system modeling application. The results of example workflows that are based on the proposed methodology is a part of this study.

The first example allows running and analyzing a global circulation model on both a grid computing environment (TeraGrid) and a cluster system with meaningful abstraction of used model and computing environment. The development version of NCAR Community Climate System Model (CCSM) model is used for this purpose. In this application example, the collection of provenance information has the added benefit of documenting a run in far greater detail than before. This facilitates exploration of runs and leads to possible reproducibility.

In second example, a regional coupled climate modeling system is developed for Mediterranean region and integrated into workflow system to provide better representation of regional climate system. This application has curial importance in downscaling output of the global circulation models over Turkey and near regions and it also can be used to create better representation of regional climate for the future scenarios. As, in the first application example, the workflow application collects provenance information automatically from the coupled earth system modeling system to reproduce, compare and debug the results.

The results show that the developed workflow environment is capable of running different earth system models on a different high performance computing resource with a meaningful abstraction. The proposed work environment acts as an abstraction layer and hides the detail of the used infrastructure and earth system model from user and it also collect standardized provenance information about both model and computing resource to represent the work environment as possible as it can.

## 1. GİRİŞ

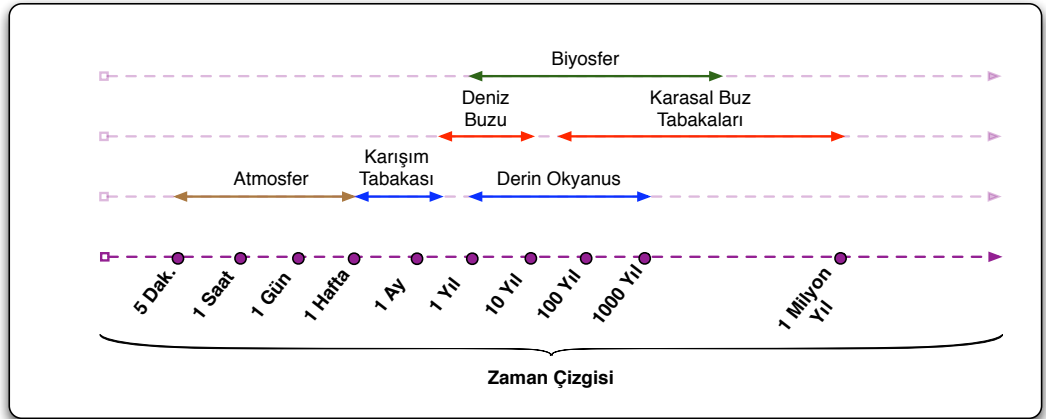
### 1.1 Yer Sistem Biliminde Karşılaşılan Problemler ve Amaç

Son yıllarda bilişim ve bilgi teknolojileri, yüksek başarımlı hesaplama ve depolama sistemleri, ölçüm ve uzaktan algılama sistemleri vb. alanlarında yaşanan hızlı ve düzenli gelişim ile birlikte, yer sisteminin daha iyi anlaşılmasını amaçlayan modelleme ve analiz çalışmaları, gün geçtikçe daha karmaşık ve hesaplama yönünden daha yoğun bilimsel problemlerin çözümüne odaklanmaya başlamıştır. Hacimsel olarak büyük veri kümelerinin işlenmesi ve sorgulanması, farklı yapılarıdaki model ve uygulamaların birlikte uyumlu bir şekilde çalıştırılması, yer sistemini temsil eden bileşenlerin daha gerçekçi ve hassas bir şekilde modellenmesi, bu tip problemlerin en önemlileri arasında sayılabilir. Bu problemlerin çözülmesi, en az problemin kendisi kadar karmaşık alt problemlerin ve süreçlerin de çözülmesini zorunlu hale getirmektedir. Karmaşık modelleme süreçlerinin yönetilmesi ve çalışmada kullanılan bileşenlerin birbirleri ile uyumlu çalışmasının sağlanması, günümüzde de bilim adamları için önemli bir araştırma alanı olmaya devam etmektedir.

Çözülmek istenen problemlerin karmaşıklığı ve birçok alt bileşenden oluşması, araştırmacıları birbirinden farklı düzeyde bilgi ve tecrübe gerektiren matematiksel ve fiziksel modelleri, hesaplama ortamlarını ve uygulamaları bir arada kullanmaya itmektedir. Ancak bu durum, araştırmacının probleme odaklanmasını engelleyerek, çalışmanın asıl amacı olan, problemin çözüm sürecini yavaşlatmakta veya imkansız hale getirmektedir.

Yukarıda kısaca tanımlanan ve problemin çözülmesi aşamasında karşılaşılan sorunlar, birçok araştırma alanı için ortak olsa da, özel olarak yer sistem bilimleri açısından bakıldığında daha karmaşık bir şekil almaktadır. Yer sistemi; atmosfer, okyanus, biyosfer ve buzküre gibi hem zamansal hem de uzaysal bakımdan, birbirinden çok farklı ölçekte birçok alt eleman ve sürecin birleşiminden oluşmaktadır (Bkz. Şekil

1.1). Tüm bu alt süreçlerin birbirleri ile doğrusal ve doğrusal olmayan etkileşimi sonucunda, yer sisteminin bugün çözmeye ve anlamaya çalıştığımız karmaşık yapısı meydana gelmektedir. Örneğin, atmosferdeki kirleticiler veya gazların modellenmesi amacı ile kullanılan kimyasal bir modelin zaman ölçeği saniyeler mertebesindeyken, bu kirleticilerin atmosferdeki dağılımını veya taşınımını sağlayan hava olaylarının benzetimini yapan bölgesel bir atmosfer modelinin zaman ölçeği haftalar mertebesinde olabilir. Ya da, denizlerde meydana gelen küçük ölçekteki akıntıların, aylık ve belirli bir kıyı boyunca olan hareketi, iklim değişimleri göz önüne alındığında, yüzyıllar süren ve tüm dünyayı içine alan genel bir su çevirimi haline dönüşebilir. Bu sebeple, yer sistemini oluşturan tüm bu alt bileşenleri tanımlayan modeller, problemin de doğasına bağlı olarak "kuple" (modellerin bir bütün olarak ve birbirleri ile etkileşecek biçimde) veya tekil olarak kullanılabilir. Farklı fiziksel süreçleri tanımlayan modellerin birbirleri ile etkileşiminin nasıl kurulacağı problemi, gün geçtikçe üzerinde daha fazla araştırma yapılan bir konu haline almıştır. Bu tip problemlerin en yaygın örneklerinden biri; farklı çözünürlüklere ve ızgara tipine sahip atmosfer ve okyanus modelleri arasındaki ısı ve momentum akılarının, birleştirilmiş model sistemi içerisinde nasıl paylaşılacağı ve hesaplanacağıdır. Bu durumda üzerine etkiyen güç (momentum ve ısı akısı) altında farklı tepki süresine sahip iki fiziksel süreç birlikte ele alınmaktadır. Bir diğer örnek ise; küresel dolaşım modellerinin - bölgesel etkileri daha iyi temsil etmesi için - bölgesel atmosfer veya okyanus modelleri ile bir bütün halinde çift yönlü bir etkileşimle kullanılmasıdır.



**Şekil 1.1:** İklim sisteminde karakteristik zaman ölçekleri (www.dkrz.de)

Tüm bu farklı fiziksel, kimyasal ve biyolojik süreçlerin detaylı bir şekilde tanımlanmasına yönelik çalışmaların artması, yer sistem modelleme çalışmaları açısından ele alındığında, birçok farklı uzman ve araştırmacının birlikte ve büyük bir uyumla çalışmasını gerektirmektedir. Yukarıda kısaca açıklanmaya çalışılan karmaşık süreçler düşünüldüğünde, yer sistem modellerinin monolitik yani tek bir parça şeklinde ve tek bir araştırma merkezi tarafından geliştirilmesinin, yönetilmesinin ve kullanılmasının imkansız olduğu açıkça görülebilir. Bu sebeple, yer sistem bilimi içerisinde bulunan farklı alt disiplinler, kendi modellerini geliştirmeye ve kullanmaya eğilimlidir. Başlangıçta avantaj gibi görünen bu durum, belirli standartlara uyulmadan ve tekil olarak geliştirilen bu modellerin birbirleri ile uyumlu bir şekilde kullanılmasını imkansızlaştırmakta ve karmaşıklaştırmaktadır. Sonuç olarak, monolitik olarak üretilmiş ve birbirleri ile bağlantısı bulunmayan uygulama, model ve bilgi kümelerinin etkin bir şekilde ve bir bütün olarak bir araya getirilmesi ve bu amaçla metodolojilerin geliştirilerek standartların oluşturulması, temel bir gereklilik haline gelmiştir.

Bu çalışma, yukarıda genel hatları ile açıklanmaya çalışılan problemlerin çözümüne katkıda bulunmak için var olan yer sistem modellerinin, belirli bir standart yapı altında toplanmasına katkıda bulunacak yeni metodolojilerin geliştirilmesini amaçlamaktadır. Böylece, yapılan modelleme çalışmalarının veya üretilen bilginin kolayca tekrar edilebileceği, paylaşılabilirliği, birbirleri ile karşılaştırılabilirliği ve modelleme süreçlerinde yapılan hataların ayıklanabileceği bir çalışma ortamının geliştirilmesine katkıda bulunacaktır.

## **1.2 Çözüm Yöntemleri ve Tarihsel Gelişim**

Bir önceki bölümde (1.1) kısaca açıklanan problemlerin çözümüne yönelik birçok farklı yaklaşım bulunmaktadır. Bu problemler karşısında üretilen çözüm önerilerini gruplamak istediğimizde, temel iki farklı yaklaşım olduğu görülmektedir. Bunlar; "çerçeve" ve "iş akışı" yaklaşımlarıdır.

Modellerin birbirleri ile belirli bir uyum içinde çalışmasını amaçlayan çerçeve yaklaşımı; farklı yer sistem modellerine ait bileşenleri ve bu modelleri birleştirmek için kullanılan katmanı (kuple bileşeni), standart ve tekil bir ara yüz ile birleştirmeyi amaçlamaktadır. Bu yaklaşım için verilebilecek en önemli örneklerden biri, ESMF [1, 2] kütüphanesidir. ESMF kütüphanesi aracılığı ile model; başlangıç, çalışma ve

son bulma şeklinde üç temel parçaya ayrılır. Böylece, modelin diğer modeller ile olan etkileşimi, standart bir ara yüz ile sağlanmış olur. ESMF, farklı yer sistem modellerinin birleştirilmesi için bir üst yapı ve bu modelin bileşenlerinin kararlı bir şekilde çalışmasını sağlayan bir alt yapı bileşeni içerir. Model ile ilgili iç bileşenler veya kaynak kod yapıları, bu iki ana tabaka arasında bulunur. Bu çok katmanlı yapı ile ESMF kütüphanesi, tipik bir yer sistem modelinin ihtiyaç duyabileceği tüm bileşenleri içeren standart bir yapı olması için tasarlanmıştır. ESMF kütüphanesi ile ilgili detaylı bilgi, Bölüm 3.2'de daha detaylı bir şekilde okuyucuya sunulacaktır.

ESMF kütüphanesi kullanılarak yapılan çalışmaların başında, CCSM modeline ait alt modellerin (atmosfer, okyanus, buz küre, kara küre ve glasiyer) birleştirilmesi gelmektedir. Bu amaçla, CCSM'e ait MCT ile yazılmış kuple bileşeni, ESMF ile tekrar yazılmış ve yeni nesil CCSM (CCSM4) modeli, her iki kuple uygulamasını içerecek şekilde yeniden tasarlanmıştır. Böylece IPCC raporlarında çıktıları önemli yer tutan CCSM modelinin, yüksek başarılı hesaplama sistemlerinde daha performanslı olarak çalışabilmesi ve daha yüksek işlemci sayılarına kadar ölçeklenebilmesi amaçlanmaktadır. CCSM küresel dolaşım modelinin daha yüksek uzaysal çözünürlüklerde çalıştırılarak, sonuçların bir sonraki IPCC raporunda (AR5) kullanılmasının önündeki engeller kaldırılmış olacaktır. ESMF kütüphanesi kullanılan diğer uygulamalar arasında; farklı bölgesel yer sistem modellerinin birleştirilmesi (WRF ve LIS) ve uzay havasının modellenmesi için yeni kuple model sistemlerinin (SWMS) geliştirilmesi sayılabilir.

Çerçeve yaklaşımı için gösterilebilecek diğer örnekler; modellerin kuple edilmesi için geliştirilmiş MCT [3–5] ve Avrupa Birliği 5. ve 7. (METAFOR projesi ile) Çerçeve Programı tarafından desteklenen PRISM/ENES [6, 7] projeleridir.

MCT kütüphanesi ile geliştirilen en önemli uygulamalardan biri, küresel ve bölgesel okyanus modellerinin birbirleri ile kuple edilmesidir (CCSM ve ROMS). Bu amaçla NCAR'da yapılan bir çalışma, okyanuslarda meydana gelen ve küresel modeller ile hassas bir şekilde çözülemeyen bölgesel değişimlerin, küresel modeller ile bölgesel bir okyanus modelinin kuple edilerek çözülmesini amaçlamaktadır. Ayrıca MCT kütüphanesi kullanılarak yapılmış bölgesel modelleri birleştirme çalışmaları da literatürde önemli yer tutmaktadır [8, 9].

PRISM projesinin, atmosferik/okyanus genel dolaşım, atmosfer kimyası, okyanus biyokimyası, kara ve buz yüzeyleri ile ilgili küresel ve bölgesel ölçekte 30'dan fazla modeli bünyesinde bulundurması ve bu modellerin birbirleri ile kuple bir şekilde kullanılmasını sağlayacak yapılar içermesi planlanmaktadır. PRISM uygulaması, temel olarak üç standart bileşenden oluşmaktadır. Bunlar; çalışma (SRE), derleme (SRC) ve ağ servisi (WSS) bileşenleridir. Çalışma bileşeni (ESMF karşılığı alt yapı); kuple edilmiş modellerin çalıştırılmasından sorumludur ve standartlaştırılmış alt araçlar (modellerin birleştirilmesi için geliştirilmiş betikler veri işleme rutinleri, görselleştirme ve veri saklama araçları) içerir [10]. Bu bileşen, yeni bir model veya çalışma ortamı için kolayca uyarlanabilir. Standart derleme bileşeni (ESMF karşılığı üst yapı) ise; çalışma bileşeni gibi, geliştirilmiş alt araçlar yardımı ile model derlenmesi için standart ve esnek bir yapı oluşturmayı amaçlamaktadır. Son olarak ağ servisi bileşeni; kullanıcılara internet ortamından, hazırladıkları sayısal iklim benzetimlerini yapılandırmalarını (ESMF ağ servisleri gibi) ve kullanılan hesaplama ortamına iş vererek izlemelerini sağlar. PRISM; yer sistem modellerinin kuple edilebilmesi için, OASIS adı verilen birleştirici bir yazılımı, modeli hesaplama ortamının karmaşık yapısından ayırmak için ise, grid teknolojilerini kullanmaktadır. Grid servisleri, hesaplama ve veri kaynağının coğrafik ve yönetsel sınırların dışına çıkarılması yoluyla, kullanıcılar tarafından paylaşılmasını sağlar. Grid servisleri ile PRISM servislerini birbirinden ayıran en temel farklardan biri; PRISM'in tekil model bileşenlerini, farklı konumlardaki hesaplama kaynaklarında çalıştıramamasıdır [7].

PRISM projesi ile yakından ilişkili METAFOR projesi; IPCC AR5 raporunda kullanılacak küresel dolaşım modelleri için ön veri ve köken bilgisi tanımlarının standartlaştırılmasını ve çeşitli ontoloji yapılarının geliştirilmesini planlanmaktadır. Böylece, IPCC raporunda kullanılacak model sonuçlarının kolayca karşılaştırılmasını ve analiz edilmesini sağlayan bir yapı kurulmuş olacaktır. METAFOR projesi ile ilgili olarak daha detaylı bilgi Bölüm 1.3'te bulunabilir.

Çerçeve yaklaşımının aksine bilimsel iş akışı uygulamaları; yer sistem bilimi çalışmalarında karşılaşılan problemleri çözmek için, tüm süreci birbirleri ile ilişkili ve daha kolay anlaşılabilir birçok anlamlı parçaya ayırıp tanımlayan standart bir yönetim sisteminin geliştirilmesine odaklanmıştır [11]. Bu şekilde, yer sistem modelinin çalışması için gerekli ön işleme, çalışma ve son işleme aşamalarında kullanılan birçok



farklı teknoloji ve uygulamanın tek bir yapı altında toplanması amaçlanmaktadır. Bu amaçla geliştirilmiş birçok farklı uygulama ve proje mevcuttur. Bunlardan en önemlileri; Kepler [12], Taverna [13], Triana [14] uygulamaları ile , GENIE [15] ve LEAD [16,17] projesidir.

GENIE projesinin amacı; küresel iklimin uzun süreli (özellikle son buzul devri) davranışının incelenerek, gelecekte insan etkisi ve küresel ısınma ile oluşabilecek değişimlerin araştırılmasıdır. Proje, yer sistemini oluşturan bileşenlere ait alt modellerin, hesaplama gridi ile entegre edilerek daha gerçekçi bilgisayar benzetimlerinin yapılabilmesini amaçlanmaktadır. Literatürde, bu yöntemi kullanarak, okyanuslar tarafından ısının enlemler boyunca taşınımını sağlayan termohaline sirkülasyonunun (Conveyor Belt) incelendiği çeşitli [15, 18] çalışmalar bulunmaktadır.

Yer sistem bilimleri temel alındığında, iş akışı yaklaşımını bilimsel ağ geçidi kavramı ile birleştirmesi nedeniyle, LEAD projesi oldukça önemli bir örnek oluşturmaktadır. Bu uygulama sayesinde, meteorolojik veriye ulaşma, analiz etme, modelleme ve görselleştirme aşamalarında kullanılan ileri seviyede karmaşık birçok araç, tek bir merkez altından yönetilebilmektedir. İçerdiği iş akışı uygulaması XBay<sup>1</sup> ile, WRF modelini TeraGrid [19] hesaplama gridi üzerinde farklı veri setleri ile çalıştırıp, model sonuçlarını otomatik olarak incelemek veya belirli bir bölgedeki hava şartlarına göre yeni bir WRF benzetimini sisteme otomatik olarak vermek mümkündür. Tüm bu bileşenleri biraraya getiren bir uygulama olması nedeniyle LEAD projesi, bilimsel iş akış uygulamalarının yer sistem bilimi çalışmalarında kullanıldığında neler yapılabileceğine iyi bir örnek oluşturmaktadır.

İş akışı yaklaşımının önemli örnekleri arasında bulunan Kepler uygulaması, LEAD projesi tarafından kullanılan XBay iş akışı uygulaması ile benzer bir yapıda tasarlanmıştır. Farklı bilimsel disiplinler tarafından kullanılacak genel bir iş akışı uygulaması olan Kepler, köken bilgisinin toplanması ve hesaplama kaynaklarına özel yapılar içermesi açısından önemlidir. Bu çalışmada kullanılan Kepler uygulaması ile ilgili daha detaylı bilgi Bölüm 3.1'te bulunabilir.

Yukarıda kısaca tanımlanan ve çeşitli örnekleri verilen iki farklı yaklaşım, daha önceki bölümlerde tanımlanan sorunların çözümüne odaklanmakla birlikte, yer sistem

---

<sup>1</sup><http://www.extreme.indiana.edu/xgws/xbaya/index.html>

modelleme çalışmalarının geçirdiği süreçler ile ilgili bilginin kayıt altına alınması, en az bu problemlerin çözümü kadar önemlidir. Bu nedenle bir sonraki bölümde, ön veri ve köken bilgisi hakkında detaylı bilgi verilerek, bu bilginin yer sistem bilimlerindeki öneminin okuyucuya aktarılması amaçlanmaktadır.

### **1.3 Ön Veri ve Köken Bilgisinin Yer Sistem Bilimlerindeki Önemi**

Önceki bölümlerde (1.1, 1.2), varolan uygulama ve modelleri birleştiren birçok farklı çalışma ve projeden bahsedilmiştir. Bu çalışmaların her biri; uygulamanın tipine, kullanılan modele ve hesaplama ortamına özgü olarak ön veri ve köken bilgisini daha sonra kullanılmak üzere belirli bir anlamlılık seviyesinde toplayıp saklamakta veya merkezi bir veri tabanında tutmaktadır. Ancak, birbirinden ayrık ve belirli bir standarda uymadan toplanan bu bilgi, yapılan çalışmaların sonuçları arasında karşılaştırma yapmayı engellemekte veya zorlaştırmaktadır. Bu nedenle, toplanan köken bilgisi ve ön verinin standartlaştırılarak kullanılması, yer sistem bilimi açısından son derece önemlidir. Bu bölüm, okuyucuya hem bu konuda yapılan çalışmalar ve araştırmalar hakkında genel bir bilgi vermeyi, hem de çerçeve ve iş akışı yaklaşımlarında bu bilginin kullanılmasının önemine değinmeyi amaçlamaktadır. Köken bilgisi ve ön veri ile ilgili daha detaylı bilgi, sonraki bölümlerde okuyucuya sunulacaktır.

Ön veri; üretilen bilgi, veri veya kullanılan model hakkında bilgi içeren bir veri tipidir. Başka bir deyişle, veri hakkında tutulan veri olarak tanımlanabilir. Ön veri; bir modelin hangi tip dinamik atmosferik çekirdek (sonlu farklar, eulerian, sonlu hacim vb.) kullanılarak tasarlandığı, ne tip ızgara sistemine (yatay ve düşeyde) sahip olduğu ve model tarafından üretilen değişkenlere ait açıklamalar gibi birçok farklı bilgiyi içerebilir. Bu açıdan bakıldığında ön veri, model sonuçlarının analiz edilmesi, hataların ayıklanması ve farklı model sonuçlarının karşılaştırılması gibi birçok alanda son derece önemli bir rol oynamaktadır.

Köken bilgisi ise; yapılan çalışmanın evrimi hakkındaki bilgi veya veri parçalarını içerir. Kullanılan modelin, çalıştığı hesaplama ortamında hangi parametreler ile derlendiği, hangi seçenekler ile çalıştırıldığı, üzerinde yapılan değişikliklerin listesi, çalıştığı işletim sisteminin tipi, işletim sistemi üzerine yapılan yamalar, işlemci mimarisi, girdi verisinin tipi gibi birçok bilgi bu başlık altında değerlendirilebilir. Bu açıdan düşünüldüğünde köken bilgisi; çalışma sonuçlarının tekrar edilmesi, hata

ayıklanması ve model performansının geliştirilmesi aşamalarında önemli bir yardımcı bilgi kümesidir.

Hem köken bilgisinin hem de ön verinin modellerden veya uygulamalardan, standart bir şekilde nasıl toplanacağına dair son zamanlarda literatüre girmiş birçok önemli çalışma vardır. Temel olarak çeşitli ontoloji ve ön veri yapıları geliştirilmesine odaklanmış bu çalışmaların en önemlileri, birbirleri ile yakından ilişkili olarak geliştirilen Curator [20], ESG [21, 22] ve bir Avrupa Birliği projesi olan METAFOR [23] projeleridir.

METAFOR projesi, iklim verisinin ve modellerin standart bir şekilde tanımlanması için CIM adı verilen bir yapı geliştirmeyi amaçlamaktadır. Böylece model ve veri arasındaki farklar, değişik disiplinlerden araştırmacılar tarafından kolayca paylaşılarak anlaşılabilir. METAFOR projesi, standart yapının oluşturulması amacıyla, CF [24], Curator, PRISM ve Gridspec [25] gibi projeler ile yakın bir ilişki içindedir. CIM'de veri, en az üç bileşen olarak tanımlanır. Bunlar; model kodu ile ilgili tanımlayıcı bilgiler, model çıktısı (ön veri) ile ilgili bilgiler ve model uygulamasından (köken bilgisi) elde edilen bilgilerdir. Bu bileşenlerden toplanacak bilgilerin tanımlanması için CIM, RDF [26] ve OWL [27] gibi veri ve ontoloji standartlarından yararlanmaktadır. Proje ile geliştirilecek standart bilgi tabanı yapıları, IPCC raporlarında baz alınan farklı küresel modellere ait çıktuların işlenmesini kolaylaştırmak amacıyla kullanılacaktır. Bu nedenle METAFOR projesi, CMIP projesi ile yakın ilişki içerisindedir. CMIP projesi ile, küresel dolaşım modellerinden toplanan veri kullanılarak, küresel iklim sistemi üzerinde insan etkisi ile oluşan değişim belirlenmeye çalışılmaktadır. Bu proje kapsamında, hem geçmiş iklim kayıtlarından hem de sera gazı salınımlarının yol açabileceği, gelecekteki değişimleri temsil eden senaryo verilerinden yararlanılmaktadır [28]. CMIP projesi uygulamaları arasında, okyanuslardaki genel su çevirimlerinin artan CO<sup>2</sup> konsantrasyonu ile değişimini araştıran çalışmalar da bulunmaktadır [29].

METAFOR projesine benzer olarak ESG ve Curator projeleri; yapılan küresel iklim benzetimlerinin kolayca karşılaştırılmasını ve bu benzetimler ile ilgili ön veri ve köken bilgisinin merkezi bir veri sisteminde saklanmasını amaçlamaktadır. Kurulacak sistem, NCAR'ın CCSM modeli kullanarak yapacağı IPCC AR5 simülasyonlarının saklanması ve sonuçların araştırmacılara ESG yolu ile yayınlanması için kullanıla-

caktır. Geliştirilen ağ geçidi - METAFOR projesinde olduğu gibi - RDF, Ontoloji ve bu bilgilerin saklanması için Sesame RDF veri tabanından [30] yararlanmaktadır. Curator projesi ile üretilen ön veri standartları kullanılarak geliştirilen ESG veri geçidinin son sürümünü test etmek amacıyla yapılan DyCore semineri, bu yaklaşımın kullanışlı olduğunu göstermesi açısından önemli bir örnektir. DyCore ile tasarlanan test uygulamaları, farklı küresel dolaşım modellerinin farklı atmosferik dinamik çekirdekler ile çalıştırılması ve sonuçların ESG üzerinden karşılaştırılması süreçlerini içermektedir.

## 2. TEMEL TANIMLAR VE METODOLOJİ

### 2.1 Köken Bilgisi ve Tipleri

Köken bilgisinin yer sistem bilimi açısından önemi bir önceki bölümde (1.3) aktarılmıştı. Bu bölüm, köken bilgisi ve tipleri hakkında daha detaylı bilgi vermeyi amaçlamaktadır.

Köken bilgisi, basit olarak yapılan herhangi bir çalışmanın gelişimi veya evrimi ile ilgili tutulan kayıt olarak tanımlanabilir. İş akış sistemleri açısından bakıldığında ise köken bilgisi, iş akış sistemi kullanılarak yapılan modelleme çalışmalarının izlenmesi ve bunlar hakkında bilgi toplanmasını amaçlamaktadır. Köken bilgisi, toplanan bilginin türüne göre dört temel gruba ayrılabilir ([31], [32], [33]).

- **Sistem Köken Bilgisi:** Simulasyonun çalıştığı uzak sistem hakkında bilgi içerir. Bu başlık altında hesaplama ortamına, işletim sistemine, uygulamaların sürümlerine, tanımlı çevre değişkenlerine ve kullanılan yazılım kütüphanelerine özgü bilgiler sayılabilir. Yer sistem bilimi çalışmaları göz önünde bulundurulduğunda, sistem köken bilgisinin birçok farklı alt bileşene sahip karmaşık model sistemlerinden tek bir adımda toplanması oldukça zordur.
- **İş Akış Uygulaması Köken Bilgisi:** Bu bilgi, iş akış uygulamasının kendisi hakkında veri toplanmasını içermektedir. Tasarlanan iş akışının farklı sürümleri, iç yapısı ve tasarım sürecinde geçirdiği evreler bu kök bilgisi içerisinde bulunmaktadır. Kullanılacak iş akış uygulaması, bu tip kök bilgisi toplanmasına izin verecek esneklikte olmalıdır. Bu şekilde geliştirilen iş akış tasarımlarının farklı sürümleri arasındaki farklar ve buna bağlı olarak hatalar kolayca takip edilebilir.
- **Veri Köken Bilgisi;** Temel olarak üretilen verinin geçmişi, oluşum süreci ve bu veriyi tanımlayan ön veri hakkında bilgi içerir. Kullanılan modele; özel girdi ve çıktı dosyaları, iş akışı üzerinde yapılan veri dönüşümleri bu tip bilgiye örnek olarak

verilebilir. Bu köken bilgisi, yer sistem modelleme çalışmaları için son derece önemli bir yere sahiptir. Sonuçların analiz sürecinde orjinal veri üzerine uygulanan işlemler veya dönüşümler, analiz sürecinin belirli bir noktasında yapılan hatanın bulunması ve geri dönülmesi için izlenmeli ve kayıt altına alınmalıdır.

- **Süreç Köken Bilgisi:** Çalıştırılan iş akışı hakkında istatistiksel bilgiler içermektedir. Bu tip köken bilgisine; veri transfer hızı, transfer edilen veri miktarı ve verinin transfer edilmesi için geçen süre veya iş akışının çalışması için gereken süre örnek olarak verilebilir.

Yukarıda sıralanan köken bilgisi türlerinin her biri, iş akış sisteminin çalıştığı sistem ortamının ve/veya işin çalıştırıldığı uzak hesaplama ortamının tekrar üretilmesi için son derece önemlidir. Sistemin tekrar yaratılabilmesi, o sistem üzerinde yapılan benzetimin sonuçlarının daha kolay bir şekilde tekrar üretilebilmesini sağlamaktadır. Bu sebeple, her bir köken bilgisi tipi ile ilgili bilginin, olabildiğince detaylı bir şekilde ve merkezi bir veri tabanında standart bir şekilde toplanması, özellikle yer sistem bilimlerinde yapılan çalışmalar açısından kaçınılmaz bir gerekliliktir.

## 2.2 Ön Veri

Ön veri, köken bilgisinden farklı olarak, üretilen bilgiyi tanımlayan, açıklayan, nerede bulunabileceğini belirten veya kolayca ulaşılmasını, kullanılmasını sağlayan yapısal bir bilgi bütünü veya bilgi kaynağıdır [34]. Ön veri, çoğunlukla veri hakkındaki veri veya bilgi hakkındaki bilgi olarak isimlendirilir. Karmaşık yapıdaki yer sistem modellerine ve bunların oluşturdukları daha kapsamlı model sistemlerine ait her bir elemanın bir şekilde ön veri bilgisine sahip olması, sistemin kendi kendisini tanımlayabilecek bir yapıda olması için gereklidir.

Ön veri, üç temel tipe sahiptir. Bunlar; tanımlayıcı, yapısal ve yönetsel ön veri olarak adlandırılır. Tanımlayıcı ön veri; veri veya bilgiyi tanımlayan ön veridir. Bu başlık altında verinin kim tarafından, ne zaman ve hangi yer sistem modeli kullanılarak üretildiği, veri içerisinde hangi bilgilerin bulunduğu, eğer model ile ilgili değişkenler içeriyor ise bunların isimleri, tanımları ve birimleri, veriyi tanımlayan kısa tanımlayıcı kelimeler vb. tüm bilgiler sayılabilir. Yapısal ön veri; verinin bulunduğu hiyerarşinin ve nasıl kullanılacağına tanınması açısından önemlidir. Örneğin, verinin belirli

bir amaçla kullanılması için hangi aşamalardan geçmesi gerektiği veya bu amaçla kullanılıp kullanılmayacağı bu bilgi içersinde bulunabilir. Son tip olan yönetimsel ön veri; üretilen verinin yönetiminin kolaylaştırılması amacıyla üretilen ve saklanan ön veridir. Bu bileşen altında, veri veya bilginin nasıl üretildiği, hangi formatta olduğu (NetCDF, GRIB, HDF vb.), kimler tarafından ulaşılabileceği ve diğer teknik bilgiler bulunabilir.

Yer sistem bilimleri çalışmaları, yukarıda kısaca açıklanan tüm ön veri tiplerini belirli bir seviyede içermelidir. Çünkü ön veri, üretilen bilginin sürekliliğini ve kalıcı olmasını sağlayan en önemli etkenlerden biridir. Üretilen ön veri bilgisi ayrı bir veri tabanında tutulabileceği gibi, verinin kendisinin içerisinde de saklanabilir. Örnek vermek gerekirse; NetCDF veri formatı içerisinde bulunan her bir değişken, aynı zamanda ön veri bilgisi de içermektedir. Bu bilgiler, değişkenin ismi, tanımı, birimi ve boyutları gibi birçok farklı tipte olabilir. NetCDF formatında, değişkenler dışında dosyanın kendisi de ön veri bilgisi (yaratıldığı tarih, kim tarafından yaratıldığı, hangi standartlar ile uyumlu olduğu vb.) tutabilir.

Ön verinin üretilmesi, verinin veya bilginin organize edilmesi, kolayca ulaşılabilmesi ve kullanılması açısından çok önemli bir yer tutmaktadır. Ancak, veri üretildikten sonra ön veri bilgisinin yaratılması, zor veya imkansız olabilir. Bu nedenle, ön veri bilgisinin veri üretilirken otomatik ve standart bir şekilde üretilmesi ve yine standartlaştırılmış bir yapıda saklanması, en az ön veri bilgisinin üretilmesi kadar önemlidir. Bu amaçla, ön veri ve bir önceki bölümde (2.1) açıklanan köken bilgisinin standart bir şekilde toplanmasını ve sorgulanabilmesini amaçlayan ESC ve METAFOR gibi çalışmalar oldukça önemlidir. Bu çalışmalarda, ön veri ve köken bilgisinin yapısal bir şekilde tutulması ve kullanıcılara sunulması için çeşitli ön veri şemaları, veri yapıları, ontolojiler gibi çeşitli metodolojiler geliştirilmektedir.

### **2.3 Ontoloji Kavramı ve Yer Sistem Bilimlerinde Kullanımı**

En çok kullanılan tanımla ontoloji; herkes tarafından paylaşılan kavramlar bütünü,ün belirli kurallar dahilinde ve açık bir şekilde tanımlanmasıdır [35]. Genel olarak ontoloji, nesnelere kendileri hakkındaki bilgileri ve aralarındaki ilişkileri belirli bir ilgi seviyesinde tutmaya yarayan mekanizmadır. Aristoteles ile başlayıp 1730'larda Christian Wolff'un "Ontologie" eserini yayınlamasıyla devam eden çalış-

malar, 1990'ların başlarında bilgi ve yapay zeka teknolojilerindeki gelişmeyle birlikte daha da hızlanmış ve araştırmacıların yoğun olarak ilgisini çeken bir konu haline gelmiştir.

Temel kullanım alanları arasında; yapay zeka, yazılım mühendisliği, biomedikal, bilişim, kütüphanecilik, bilgi gösterimi gibi araştırmalar bulunmaktadır. Ontoloji üzerine yapılan çalışmalar, üretilen bilginin ve bu bilgi kümeleri arasındaki ilişkinin sadece insanın anlayabileceği bir formda tutulması yerine, makinaların da anlayabileceği bir formda ve belirli standartlar altında nasıl tanımlanabileceği üzerine yoğunlaşmıştır. Örneğin, farklı dillerde ve formlarda üretilmiş, birbirleri ile ilişkili bilgi parçacıklarının makinalar arasında paylaşılması ve bir bütün haline getirilerek sunulması, bu araştırmalar içerisinde yer almaktadır. Bu tür çalışmalara verilecek en önemli örneklerden biri, "Semantik Ağ" kavramıdır. Bu yeni nesil ağ üzerinde, geleneksel ağ yapısının aksine daha karmaşık görevler tanımlanarak, bir çok farklı alandaki bilginin harmanlanması ve kullanıcıya döndürülmesi sağlanmaktadır.

Yapılan çalışmanın özüne bağlı olarak üretilen ontolojilerin bilgiyi tasvir etme biçimleri ve kullanılan ontoloji dilleri farklı olabilir. Ancak her biri en azından aşağıdaki bileşenleri içermelidir [36]:

- **Sınıflar veya Kavramlar:** Düşünce yolu ile bulunan kavramları veya nesnelere takımını belirtir. Üst seviye sınıf yapıları başka sınıfları, nesnelere veya her ikisini birden içerebilir. Örneğin, bir atmosfer veya okyanus modeli küresel dolaşım modeli sınıfının bir alt sınıfı olabilir veya bölgesel bir atmosfer modelinde kullanılan farklı sınır tabaka alt modelleri yine bir alt sınıf olarak tanımlanabilir.
- **İlişkiler:** Alan içerisinde tanımlanan kavramlar arasındaki ilişki tipini gösterir. Biçimsel olarak n takımlı bir ürünün herhangi bir alt takımı:  $R \subset C_1 \times C_2 \times \dots \times C_N$  ifadesi ile tanımlanabilir. Ontolojiler genellikle ikili ilişkilere sahiptir. Örneğin, konum belirtmek için yer sistem modellerinde sıkça kullanılan koordinat değişkenleri (enlem, boylam, derinlik ve yükseklik) "is\_a" ikili ilişkisi ile genel değişken tanımına bağlanabilir. Bir başka örnek vermek gerekirse, boyut nesnesi veya sınıfı "is\_part\_of" ikili ilişkisi ile koordinat sistemi sınıfına bağlanabilir. Tanımlanan ikili ilişkiler, kapsadığı küme veya alan içerisindeki bilgiye göre anlamlı olabilir.



Bu bileşenler dışında ontolojiler nesne ve özellik gibi iki farklı bileşen daha içerebilir;

- **Nesneler:** En temel bileşenlerden biridir. İlgilenilen alan içerisindeki nesnelere tanımlar. Tanımlanan ontoloji içerisindeki nesnelere başka nesnelere de içerebilir. Örneğin, bölgesel okyanus modelleri sınıfı altında ROMS ve POM gibi modeller nesne olarak tanımlanabilir.
- **Özellikler:** Nesnelere arasındaki ikili ilişkilerdir. Örneğin "hasSibling" (kardeşe sahip olmak) ve "hasChild" (çocuğa sahip olmak) özellikleri Ahmet ve Mehmet gibi iki nesneyi birbirine bağlayabilir. Yer sistem bilimlerine özel olarak örnek vermek istersek, bölgesel bir atmosfer modeline ait genel parametreler "hasParameter", alt model tipleri ise "hasParameterization" şeklinde ikili özellikler tanımlanarak, modelin kendisine bağlanabilir. Ayrıca nesnelere arasındaki ikili ilişkilerin terslerini de tanımlamak mümkündür.

Ontoloji dilleri, FOL ile birleştirilmiş çerçeve ve tanımlama mantığına göre gruplanabilir. İlk ontoloji dili, Cyc ontolojisi yaratmak için kullanılan CycL dir. Daha sonra FOL tabanlı KIF geliştirilmiştir. KIF altında direk olarak ontoloji yaratmanın zor olması sebebiyle, 1997'de Ontolingua ontoloji tanımlama dili, KIF'in üzerinde bir katman olacak şekilde geliştirilmiştir. Ontolingua 1990'larda ontoloji geliştirmek için standart olarak kabul edilmişti, ve Lisp benzeri bir sözdizimine sahiptir. Bir diğer ontoloji dili ise 1995'te çerçeve ve FOL birleşiminden geliştirilmiş ve Lisp benzeri bir sözdizimine sahip olmayan Flogic'tir [37].

İnternetin gelişimiyle birlikte, ağ karakteristik özelliklerini kullanan yeni ontoloji dilleri ortaya çıktı. Bu diller genellikle "ağ tabanlı ontoloji dilleri" veya "ontoloji markup dili" olarak adlandırılmaktadır. Bu dillerin sözdizimi, HTML ve XML'in söz dizimlerine çok benzemektedir ve bu nedenle, yeni ontoloji geliştirmek önceki örneklere oranla daha kolaydır. OWL, bu tipte ontoloji geliştirmek için en yaygın olarak kullanılan dildir ve RDF ve RDFS yapısının üzerinde bir katman olarak tasarlanmıştır. Kaynaklar, özellikler ve özellik değerleri RDF ifadeleri veya üçlüleri ile tanımlanır. RDF, ağ kaynaklarının semantik ağ tabanlı bir dil ile tanımlanması amacıyla W3C tarafından geliştirilmiştir. RDF ve RDF şeması genellikle kısaca RDF(S) olarak bilinir ve RDF(S), diğer diller ile karşılaştırıldığında, ontoloji

tanımlamalarını en basit şekilde yapan dildir ancak detay tanımlanması konusunda zayıftır. Sadece kavram, kavram sınıflandırması ve ilişkilerin tanımlanmasına izin verir. Bu nedenle RDF’i temel alan OWL dili geliştirilmiştir.

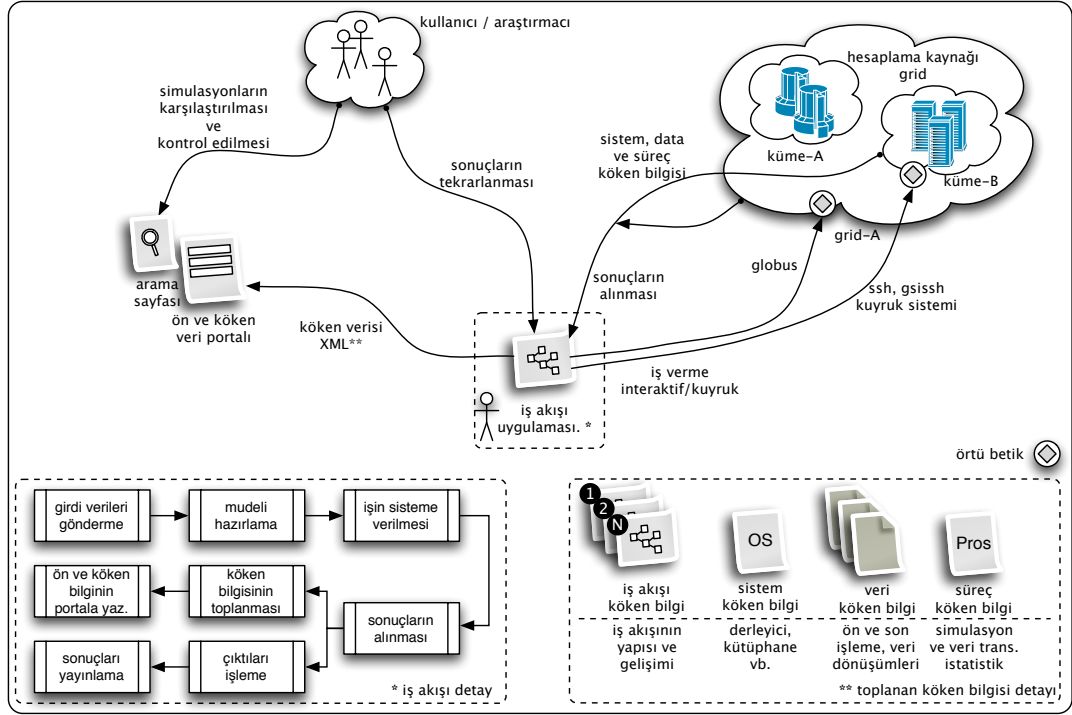
OWL, anlamlılık seviyesine göre üç alt dile ayrılabilir. Bunlar OWL-Lite, OWL-DL ve OWL-Full’dur. OWL-Full, anlamlılık seviyesi en yüksek alt dildir. OWL-Full, OWL-DL’in, OWL-DL ise OWL-Lite alt dilinin daha fazla özelliğe sahip geliştirilmiş biçimleridir. OWL-DL’in avantajlarından biri, içerdiği alt ve üst sınıfları, ilişkiler arasındaki mantıksal boşlukları çeşitli araçlar yardımıyla otomatik olarak tanımlayarak geliştirebilmesidir. Tüm ontoloji dilleri, aynı anlamlılık ve mantık seviyesine sahip değildir. Hangi tip ontoloji geliştirileceğine bağlı olarak seçilen dil son derece önemlidir [37]. Bu sebeple ontoloji geliştirme sürecine başlamadan önce anlamlılık ve mantık bakımından hangi detay seviyesine ihtiyaç duyulduğunun ve sonra hangi dilin gerekleri karşıladığının sorgulanması gereklidir.

Bu çalışma, ileriki bölümde (4.5) detaylı olarak tanımlanacağı üzere, CDO uygulaması ile ilgili alt komutların ve parametrelerin saklanması amacıyla, basit bir ontoloji (OWL bilgi tabanı) içermektedir. Bu bilgi tabanı, daha sonraki çalışmalarda, veri setine özel işlemlerin tanımlanması için, bir alt bilgi tabanı olarak kullanılabilir.

## **2.4 Önerilen Metodoloji**

Yer sistem uygulamaları, genel olarak bir veya birden fazla modelin birlikte yani kuple bir şekilde veya tek olarak, belirli bir problemin çözülmesi için kurulması, çalıştırılması ve sonuçların analiz edilmesi aşamalarını içermektedir. Bu açıdan bakıldığında çalışma, her biri farklı seviyede bilgi ve tecrübe gerektiren birçok farklı bileşen (hesaplama ortamı, yer sistem modeli vb.) içermektedir. Tüm bu modelleme ve analiz sürecini standartlaştırmak ve ortalama bir kullanıcı tarafından kullanılmasını sağlamak amacıyla iş akış sistemleri kullanılabilir. Tipik bir yer sistem modelleme ve analiz çalışmasında kullanılan tüm alt bileşenlerin, bir iş akışı uygulaması ile kontrol edildiği durum için sistemin genel yapısı ve ana bileşenleri Şekil 2.1’de açıkça görülebilir.

Şekilden görülebileceği gibi iş akış sistemi, üç temel bileşenden oluşmaktadır. Bunlar; hesaplama ortamı, iş akışı uygulaması ve toplanan ön veri ile köken bilgisinin



**Şekil 2.1:** İş akış sisteminin genel yapısı ve bileşenleri

kullanıcılara sunulması için geliştirilmiş veri tabanı ve buna bağlı bir portaldır. Bu bileşenler dışında ayrıca kullanılan model sistemi de bir bileşen olarak sayılabilir.

Toplanan ön veri ve köken bilgisinin kullanıcı tarafından sorgulanması amacıyla geliştirilen uygulamalara ESG örnek olarak verilebilir. Dağıtık bir veri depolama sistemi olan ESG, veri ile ön veri bilgisini kullanıcıya birlikte sunarak farklı simulasyonların birbirleri ile karşılaştırılmasını ve sorgulanan bilgiye kolayca erişilmesini sağlamaktadır. Bu amaçla NCAR'da gerçekleştirilen DyCore - farklı atmosferik dolaşım modellerinin birbirleri ile karşılaştırılmasını amaçlayan bir çalışma - etkinliği ile elde edilen sonuçlar, sistemin test edilmesi için kullanılmış ve analiz süreci bu şekilde hızlandırılmıştır. Her iki sistemin de IPCC tarafından hazırlanacak AR5 raporunda kullanılacak CCSM küresel dolaşım modelinin sonuçlarının sunulması ve toplanan ön verinin saklanması amacıyla kullanılması planlanmaktadır. Bu çalışma, toplanan ön veri ve köken bilgisinin sunulması amacıyla bir veri geçidi geliştirilmesi aşamasını kapsamamaktadır, ancak toplanacak ön veri ve köken bilgisi, bu tip bir veri geçidi tarafından kullanılacak standart bir yapıda geliştirilmelidir. Bu amaçla, son derece yaygın olarak kullanılan XML veri saklama ve tanımlama teknolojilerinden yararlanılabilir.

Tanımlanan modelleme ortamının merkezinde diğer tüm bileşenlerin kontrol edilmesi amacıyla kullanılacak bir iş akış uygulaması bulunmaktadır. Bu uygulama, araştırmacı ile diğer modelleme ortamı bileşenleri arasında bir çeşit soyutlama katmanı işlevi görerek, farklı teknolojilerin birbirleri ile uyumlu bir şekilde çalışmasını sağlamaktadır. Diğer bir deyişle, iş akış uygulaması, çalışma sırasında ortaya çıkan ve tekrar eden birçok adımın kendi bünyesi içinde tanımlanmasına izin vererek, tüm bu süreçler için merkezi bir yönetim sisteminin oluşturulmasını sağlar. Bu şekilde, araştırmacının problemin çözümüne yoğunlaşmasını sağlayarak, problemin çözümü ve/veya sonuçların analiz sürecini hızlandıran bir çeşit katalizör görevi görür.

Tanımlanan iş akışı modelleme ortamının diğer önemli görevlerinden biri ise, standartlaştırılmış bir şekilde ön veri ve köken bilgisinin toplanmasıdır. Yer sistem bilimleri ve bu alanda yapılan modelleme çalışmaları göz önüne alındığında, bu bilginin otomatik bir şekilde toplanmasının son derece önemli olduğu kolayca anlaşılabilir. Toplanan köken ve ön veri bilgisi, çalışmanın daha sonraki aşamalarında hata ayıklama, elde edilen sonuçların karşılaştırılması, model performansının geliştirilmesi ve belirli bir simulasyon sonucunun doğrulanması veya tekrar edilmesi amacıyla kullanılabilir. Son zamanlarda bu konuda yapılan çalışmalar, toplanan ön veri ve köken bilgisini kullanarak, tasarlanan iş akışının otomatik olarak tekrar üretilmesi üzerine yoğunlaşmaktadır. Önerilen metodoloji, bu bilgilerin anlamlı bir detay seviyesinde toplanması için çeşitli araçlar ve metodolojiler geliştirilmesi aşamalarını da içermektedir. Bu amaçla, hem kullanılan model sisteminin kendisinden hem de iş akışından standart bir şekilde köken bilgisinin toplanması amacıyla, daha önceki bölümlerde kısaca tanımlanan çerçeve yaklaşımından yararlanılabilir.

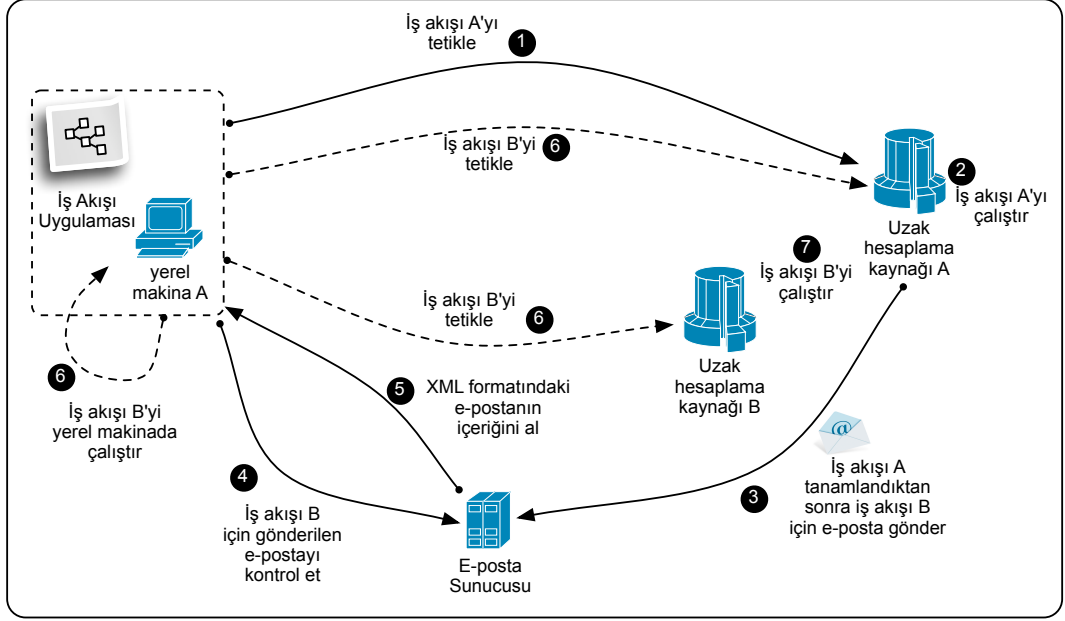
Ön verinin ve köken bilgisinin kullanılan model veya modellerden toplanması, yer sistem modelleri düşünüldüğünde oldukça karmaşık bir süreçtir. Belirli bir standarta uyulmadan geliştirilen yer sistem modellerinden, ön veri ve köken bilgisinin toplanması, bu tip modellerin kuple edilmesi ve üçüncül uygulamalar ile bu modellerin entegre edilmesi son derece karmaşık bir süreçtir. Modellerde kullanılan konfigürasyon dosyalarının XML yerine basit ASCII dosyaları kullanılarak tutulması ve modeller tarafından kullanılan çıktı dosyası formatlarının farklı olması, standartlara uyulmadan geliştirilen modeller de karşılaşılan temel problemler olarak sayılabilir. Bu konuda, son zamanlarda yapılan çalışmalar, yukarıda tanımlanan açığın

kapatılması ve modellerin daha standart bir yapıda tanımlanması ve geliştirilmesi üzerine yoğunlaşmıştır. ESMF [1] bu amaçla geliştirilen uygulamaların başında gelmektedir. ESMF, hem modellerin birbiri ile standart bir arayüz yardımıyla kolayca birleştirilebilmesini, hem de modelin içerisinde bulunan kök ve ön veri bilgisinin diğer uygulamalar içerisinde kullanılmak amacıyla yapısal bir formatta alınmasını sağlamaktadır. Bu nedenle, tasarlanacak iş akış uygulamasının ESMF ile entegre edilmesi, yapılacak standartlaştırma çalışmaları açısından büyük kolaylıklar sağlayacaktır. Bu amaçla, iş akışına entegre edilen kullanılan modellerin ESMF bileşeni olması ve/veya ESMF kütüphanesi ile kuple edilmiş modellerin kullanılması, önerilen metodolojinin temel ve vazgeçilemez bileşenlerinden biridir.

Yer sistem modellerinin iş akışı ortamına aktarılmasında karşılaşılabilecek en önemli zorluklardan biri, kullanılan modellerin çalışma sürelerinin çok uzun olmasıdır. Bu nedenle, bu tür uzun süren simulasyon çalışmalarının başlat, çalıştır ve incele modeli ile yapılması pek mümkün değildir. Problemi çözmek için tasarlanan iş akışı, farklı alt iş akış tasarımlarına ayrılabilir. Örneğin, simulasyonun çalıştırılması ve sonuçların incelenmesi kısımları ayrı birer iş akışı olarak tasarlanabilir. Ancak sonuçların incelenmesi için kullanılacak iş akışının, kullanıcı tarafından çalıştırılması gerekmektedir. Bu nedenle, geliştirilecek çalışma ortamının, hesaplama sistemine verilen iş bittiğinde bir sonraki iş akışını otomatik olarak çalıştıracak şekilde tasarlanması, önerilen metodoloji içerisinde önemli bir yer tutmaktadır.

Şekil 2.2' de birincil iş akışını izleyen iş akışlarının standart e-posta kullanılarak otomatik olarak çalıştırılması için izlenmesi gereken adımlar açıkça görülmektedir. Bu amaçla, öncelikle birincil iş akışı (iş akışı A) daha önceden tanımlanan e-posta adresine bir mesaj gönderir (adım 3). Mesaj içeriğinde bulunan XML tanımlamaları ikincil iş akışı ve bu iş akışına ait değiştirilecek parametreleri içeren özel bir yapıdadır. Sunucuya ulaşan bu mesajın, iş akış uygulaması ile kontrol edilmesi ve mesaj içeriğinin alınması ile (adım 4 ve 5) yeni iş akışı çalıştırılabilir (iş akışı B). Bu yol ile, iş akışlarının yeni iş akışları çalıştırması mümkün olmaktadır. Örneğin, bir model benzetimi bittikten sonra uzak sistemden çıktıların kopyalanması veya çıktıların son işleme tabi tutulması gibi diğer iş akışları bu yapı ile çalıştırılabilir.

Önerilen çalışma ortamı; farklı yer sistem modellerinin kuple edilmesi ve bu model sistemlerinden ön veri bilgisinin toplanması aşamasında ESMF kütüphanesini, tekil



**Şekil 2.2:** Yeni iş akışının otomatik olarak tetiklenmesi için gereken adımlar

veya kuple model sistemlerinin yönetilmesi aşamasında Kepler uygulamasını ve kullanılan yer sistem modellerinden köken bilgisinin toplanması amacıyla ise özel olarak hazırlanan betikleri içermektedir. Bu amaçla, Bölüm 5.5'te kullanılan araçlar ve uygulamalar hakkında kısaca bilgi verilecek ve daha sonraki bölümlerde (Bölüm 4 ve 5), yukarıda tanımlanan metodolojinin sınanması için geliştirilen uygulama örneklerine ait sonuçlar irdelenecektir.

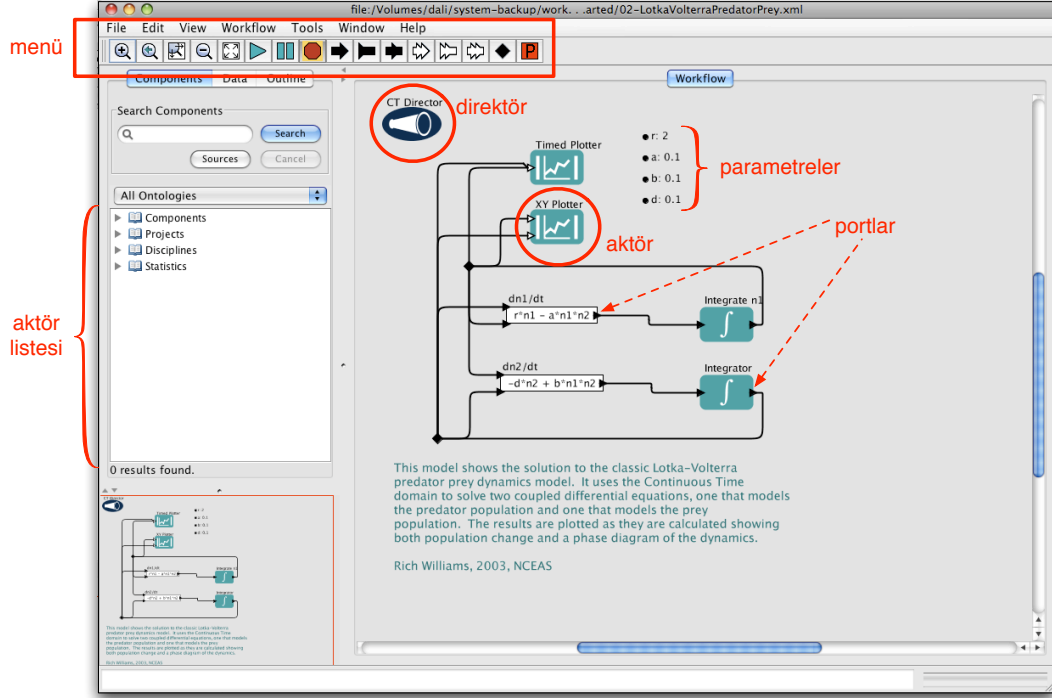
### 3. KULLANILAN ARAÇLAR

#### 3.1 İş Akışı Uygulaması: Kepler

Bir önceki bölümde (2.4) detayları anlatılan metodolojinin en önemli bileşenlerinden biri, diğer tüm bileşenlerin tek bir merkezden yönetilmesini sağlayacak olan iş akış uygulamasıdır. Literatürde geliştirilmiş birçok farklı iş akış uygulaması olmasına rağmen, bu çalışmada Kepler uygulamasının kullanılmasına karar verilmiştir. Kepler, platform bağımsız (Java tabanlı) ve açık kaynak kodlu olması, yer sistem bilimlerinde yapılan farklı çalışmaların gereksinimlerini karşılayacak aktör ve direktör yapısını temel alması ve farklı hesaplama modellerini (MoC) desteklemesi nedeniyle seçilmiştir.

Kepler, Berkeley Üniversitesi tarafından geliştirilmiş olan Ptolemy II iş akış uygulamasının geliştirilmiş ve daha kolay kullanılabilir hale getirilmiş bir versiyonudur. Temel olarak, modelleme ve analiz süreçlerinin kolaylaştırılması için tasarlanmıştır. Kepler'de kullanıcı, bir iş akışı geliştirmek için "aktör" adı verilen yapıları kullanır. Aktörler, yapılarında girdi/çıkış portları ve parametre tanımları bulunduran, belirli bir iş yapmak için özelleştirilmiş, küçük ve modüler bileşenlerdir. Üretilen iş akışının işleyişi ve aktörlerin birbirleri arasında girdi/çıkış portları aracılığı ile nasıl veri alışverişinde bulunacakları, ayrı ve özelleşmiş aktörler olan direktörler ile kontrol edilir. Örneğin, iş akışı senkronize bir şekilde (seri olarak) çalışacak ve belirli bir anda sadece tek bir aktörde iş yapılacaktır ise, bu durumda SDF direktörü kullanılabilir. SDF direktörünün tersine, eğer iş akışı içerisindeki bileşenler paralel olarak çalışacak ise, bu durumda PN direktörü kullanılabilir. Kepler uygulaması ile birlikte en çok kullanılan direktör yapıları, uygulamaya gömülü bir şekilde gelir. Ancak Ptolemy II içerisindeki diğer direktör çeşitleri de, ihtiyaç olduğunda Kepler içerisinde kullanılabilir. SDF ve PN direktörleri dışında, Kepler içerisinde; dinamik sistemlerin modellenmesi için CT, zaman bağımlı sistemlerin ve haberleşme ağlarının modellenmesi için DT ve paralel işleme ihtiyaç duymayan ancak döngüler ve kontrol yapıları içeren iş akışlarının

çalıştırılması için ise DDF direktörü bulunmaktadır. Bunlar dışında, kullanıcı isteğine bağlı olarak yeni direktörler geliştirilebilir ve sisteme entegre edilebilir. Başka bir deyişle Kepler içerisinde tanımlanan iş akışı, aktör ve direktör adı verilen iki ayrık yapı ile tanımlanır. Bu kullanıcıya, direktör yapısını değiştirerek aynı iş akış tasarımını farklı bir hesaplama modeli ile kolayca çalıştırma olanağı verir. Kepler uygulamasının genel görünümü ve içerdiği bileşenler Şekil 3.1’de görülebilir:



Şekil 3.1: Kepler uygulamasının genel görünümü ve içerdiği bileşenler

Daha önce tanımlandığı gibi, Kepler içerisinde hiyerarşik yani yuvalanmış bir yapıda iş akışı tasarlamak mümkündür. Bu amaçla, Kepler içerisindeki kompozit aktörler kullanılır. Kompozit aktörler, daha karmaşık işler yapmak için yapılarında diğer iş akışlarını barındıran özelleşmiş aktörlerdir. Bu aktörler ile üretilen iş akışları, en üstte (ana iş akışı) bulunan iş akışından farklı olarak kendi direktörlerini içerebilecekleri gibi (opak kompozit aktörler), en üst seviyedeki direktör tarafından da kontrol edilebilirler (geçirgen kompozit aktörler).

Kepler, aktörlerin yapısı içinde bulunan girdi ve çıktı portları üzerindeki verinin tipini kontrol etmek için özel bir tip yapısı kullanır. Bu şekilde, aktörlerin birbirlerine uyumsuz tipte (örneğin, tam sayı olarak tanımlanmış bir porta, reel sayı göndermek)



veri göndermesinin önüne geçilmiş olur veya farklı tipteki veriler, eğer mümkün ise birbirlerine otomatik olarak dönüştürülür. Ancak kullanıcı istediği takdirde, genel obje tipi ile yeni veri tipleri geliştirebilir. Böylece, veri ile birlikte veriyi tanımlayan ön verinin de aktörler arasında transfer edilmesi sağlanabilir. Bu bakımdan Kepler, geliştiriciler için oldukça kolay geliştirilebilen bir yapı sunmaktadır.

Kepler için yeni bir aktör geliştirilmek istendiğinde, Java programlama dili kullanılır. Burada amaç, kullanıcının kolay ve platform bağımsız bir şekilde yeni aktörler ve/veya direktörler geliştirmesinin sağlanmasıdır. Bu çalışma içerisinde, yukarıda tanımlanan özellikler kullanılarak, uygulama testlerine ait iş akışlarının tasarlanması için yeni aktörler geliştirilip, sisteme entegre edilmesi sağlanmıştır. Kepler içerisinde yeni bir aktör geliştirmek için izlenmesi gereken metod şu şekildedir:

- Öncelikle, yeni bir java dosyası yaratılır veya daha kolay olması açısından, var olan bir aktör başka bir isim ile kaydedilir. Bu aşamada sadece aktör kodunun kopyalanması yetmez, aynı zamanda aktörü tanımlayan dosyaların da kopyalanması gereklidir. Tanım dosyalarında, her bir aktör için tek olan bir tanımlayıcı rakam (LID), aktör ile ilgili tanımlayıcı açıklamalar ve aktörün hangi semantik tip altında listeleneceği bilgisi bulunmaktadır. Kepler, aktörlerin belirli bir yapı içerisinde saklanması ve sorgulanması amacıyla, ontoloji kavramlarından yararlanmaktadır.
- Aktör ve tanımlayıcı dosyalar kopyalandıktan sonra, aktörün yapılması istenen iş ve alacağı parametrelere göre gerekli tanımlamalar yapılır. Daha ileriki bölümlerde görüleceği üzere, aktörlere ait parametreler, yapılan işin tipine ve/veya belirli bir parametrenin aldığı değere göre otomatik olarak üretilebilir ve değiştirilebilir. İş akışı çalıştırıldığında, tüm aktörlere ait belirli java metodları sıra ile direktör tarafından tetiklenir ve ilgili aktörün tanımlanan işi yapması sağlanır.

Oluşturulan iş akış yapıları ve aktörlerin tanım dosyaları MoML [38] adı verilen özel bir XML dosyası şeklinde saklanır. Böylece, Kepler uygulaması ile üretilen iş akışlarının paylaşılması ve saklanması sağlanmış olur. Kepler içerisinde geliştirilen yeni bir aktörün genel yapısı ve kullanılacak metodlar; ek bölümü Kod A.1'de, yeni Kepler aktörü tanımlamak için gerekli örnek dosyalar ise; Kod A.2 ve A.3'de görülebilir.

### 3.2 Çerçeve Uygulaması: ESMF

Son yıllarda, yer sistem bilimcilerin daha büyük ölçekteki problemleri çözmeye çalışması ve buna bağlı olarak farklı yer sistem modellerinin kuple edilmesine duyulan ihtiyaç ile birlikte, bu işlemi daha basit ve daha performanslı bir şekilde gerçekleştiren çeşitli uygulama ve kütüphaneler geliştirilmiştir. Modellerin kuple edilmesi için izlenebilecek dört temel yöntem vardır. Bunları:

- **Farklı yer sistem model kodlarının birbirleri ile birleştirilmesi:** Bu yöntemde, kuple edilmek istenen her bir yer sistem modeline ait ana program, alt program veya fonksiyon şekline dönüştürülür ve yeni üretilen bu alt programlar, seçilen bir yer sistem modeline ait ana program içerisinden kullanılacak şekilde model yapısı değiştirilir. Böylece, farklı yer sistem modelleri kuple edilmiş olur. Bu yöntem, model kuple etmek için kullanılan en eski yöntemlerden biridir. Yöntemin dezavantajları arasında; ölçeklenemez olması, farklı yer sistem modellerinin eklenmesi ile gittikçe karmaşık bir yapı halini alması ve belirli bir standarda sahip olmaması sayılabilir. Avantajları arasında ise; kolay programlanabilmesi sayılabilir.
- **Var olan mesajlaşma protokolleri kullanılarak, modellerin birleştirilmesi:** Bu yöntemde, MPI, InterComm [39] gibi çeşitli mesajlaşma kütüphaneleri kullanılarak, farklı yer sistem modelleri arasında veri alışverişi sağlanmaktadır. Temel olarak, bir önceki yöntem ile aynı sorunlara sahiptir. Ancak kodun taşınabilirliği (kodun farklı hesaplama sistemlerinde çalıştırılması veya farklı derleyiciler ile derlenmesi) bu yöntem ile biraz daha geliştirilmiştir.
- **Modellerin kuple edilmesi amacıyla geliştirilmiş kütüphanelerin kullanılması:** Bu yöntem, farklı yer sistem modellerinin kuple edilmesi için özelleştirilmiş uygulamalar ve/veya kütüphaneler kullanılmasını amaçlamaktadır. Öncelikle, kod alt elemanlara ayrılarak, bu elemanların yazılan kuple bileşeni ile birleştirilmesi amaçlanmaktadır. Bu şekilde geliştirilen kodun, daha genel bir yapıda tasarlanması, farklı ve yeni yer sistem modellerinin bu sistem ile entegre edilmesinin kolaylaştırılması ve belirli standartlara uyulması sonucunda, geliştirilen kodun taşınabilirliğinin artırılması sağlanmış olur. Bu başlık altında FSM<sup>1</sup>, MCT ve ESMF uygulamaları önemli örnekler olarak sayılabilir.

---

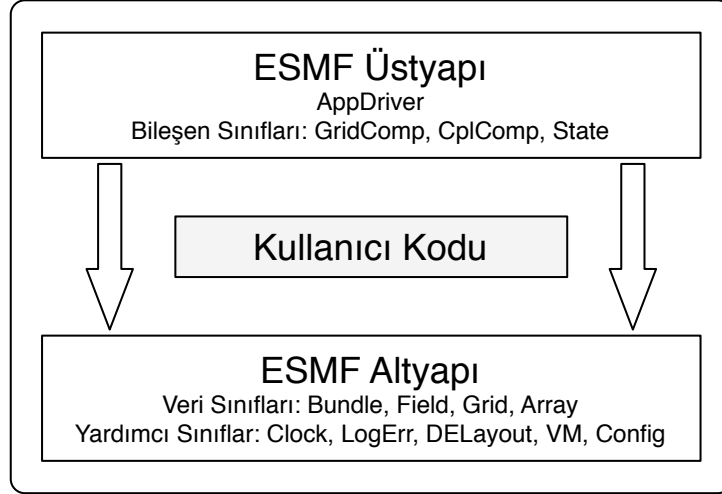
<sup>1</sup><http://www.gfdl.noaa.gov/fms>

- **Modellerin kuple edilmesi amacıyla, geliştirilmiş kuple uygulamalarının kullanılması:** Bağımsız olarak farklı merkezler tarafından geliştirilmiş yer sistem modellerinin birleştirilmesi amacıyla kullanılan uygulamalardır. Yer sistem modeli içerisinde kuple uygulaması ile ilgili metodlar kullanılarak, modeller arasında veri alışverişi sağlanmış olur. Bu grup altında, OASIS [10] uygulaması sayılabilir. OASIS uygulaması ile, kendilerine ait çalıştırılabilir dosyaları bulunmak kaydıyla, farklı yer sistem modelleri arasında alan değişkenlerinin senkronize bir şekilde taşınması sağlanmış olur. Alan değişkenlerinin taşınması sırasında, farklı yer sistem modelleri arasında iki boyutlu interpolasyon yolu ile alan değişkenleri, kaynak gridten hedef gride dönüştürülebilir.

Modellerin kuple edilmesi amacıyla geliştirilen uygulama ve kütüphanelerin en önemli örnekleri; MCT [3, 4], MCEL [40], FMS ve ESMF [1, 2] dir. Bu çalışmada, diğer uygulamalara olan üstünlüğü nedeniyle ESMF kütüphanesi kullanılmıştır. Örneğin MCEL, farklı model gridleri arasında interpolasyon yapmak için kullanıldığında, gridler arasındaki ağırlık matrisini her bir veri değişimi adımıda tekrar oluşturduğundan, performans açısından son derece verimsiz bir şekilde kuple işlemi gerçekleştirilmektedir.

ESMF kütüphanesi, yer sistem bilimlerinde kullanılan mevcut standartlar ile uyumlu, açık kaynak kodlu, paralel hesaplamayı kolaylaştıran ve geliştirilen ve bu şekilde uygulamanın kullanılabilirliğini ve performansını arttırarak, farklı yer sistem modellerini içeren model sistemlerinin kolayca ve standart bir şekilde geliştirilmesini amaçlamaktadır. Kütüphane, GFDL tarafından geliştirilen FMS uygulamasının yapısını temel almaktadır. ESMF kütüphanesinin yapısının daha iyi anlaşılabilmesi için, Şekil 3.2'deki sandviç benzetimi iyi bir örnek oluşturmaktadır.

Şekilde, ESMF uygulamasının üç temel bileşenden oluştuğu görülmektedir. Bunlardan, "kullanıcı kodu" olarak adlandırılan bileşen, modelin bilimsel algoritmalarını tanımlayan ve altyapı ve üstyapı bileşenleri arasında bulunan kısımdır. Bu bileşen örneğin, sonlu farklar hesaplamalarını veya radyasyon ile ilgili alt program tanımlamalarını içerebilir. Kullanıcı kodu, SPMD veya MPMD programlama modellerinden biri ile yazılmış olabilir. ESMF kütüphanesi, her iki programlama modelini de desteklemektedir. Bu nedenle ESMF, daha önce yazılmış birçok yer sistem model



**Şekil 3.2:** ESMF uygulamasının sandviç yapısının şematik gösterimi

kodu ile entegre olabilecek bir yapı sunmaktadır. Üstyapı bileşeninin görevi; kullanıcı kodu ile ESMF arasında bir ilişki kurarak, farklı model bileşenlerinin birbirleri ile veri alışverişlerini kolaylaştırmaktır. Altyapı bileşeni ise, geliştiriciler için standart bir yardımcı araç kümesi sunmaktadır. Bu şekilde, modellerin birleştirilmesi ve performanslı bir şekilde çalıştırılması ve farklı bileşenlerin kararlı bir şekilde çalışması garanti altına alınmış olur [1].

Model bileşenlerinin birbirleri ile iletişim kurabilmesi amacıyla ESMF, durum sınıfı (ESMF\_STATE) adı verilen bir yapı kullanır ve kuple edilmiş modelleme sistemi içerisinde bulunan modellerin, en az bir adet durum değişkeni tanımlaması gereklidir. ESMF durum değişkenleri, iki farklı tipte olabilir: Bunlardan biri; model bileşeninin başka bir model bileşeninden veya kuple bileşeninden veri almak için kullandığı durum değişkeni (ESMF\_IMPORT), diğeri ise; model bileşeninin diğer bileşenlere veri göndermek için kullandığı durum değişkenidir (ESMF\_EXPORT). Bu şekilde, her iki yönde de veri taşınımı için gerekli yapı kurulabilir. ESMF durum değişkenleri içinde, alan değişkenleri (ESMF\_Field), diziler (ESMF\_Array) veya bunların oluşturduğu gruplar bulunabilir (ESMF\_Bundle). Bu açıdan bakıldığında, bileşenler arasındaki iletişimi en etkili ve optimize bir şekilde yapmak için özelleşmişlerdir.

ESMF, farklı model bileşenleri arasında veri interpolasyonu amacıyla, değişik yöntemleri desteklemektedir. Bunların başında SCRIP [41, 42] uygulaması ile üretilen ağırlık matrisleri gelir. Farklı gridler arasında veri dönüşümünü gerçekleştirmek için ESMF

kütüphanesi, modellerin grid bilgilerinden SCRIP uygulaması kullanılarak oluşturulan ağırlık matrislerini kullanarak, paralel seyrek matris çarpımı yapar ve bir model bileşeninden diğerine veriyi interpolasyon yolu ile aktarır. Bu aşamada, kullanılan interpolasyon yöntemi, ağırlık matrisleri yaratılırken SCRIP uygulaması tarafından tanımlanır. SCRIP, kuple edilmiş modeller için son derece önemli olan korunumlu interpolasyon (birincil ve ikincil dereceden) tekniklerini de desteklemektedir. SCRIP uygulaması ile ilgili daha detaylı bilgi Bölüm 5.6'da bulunabilir.

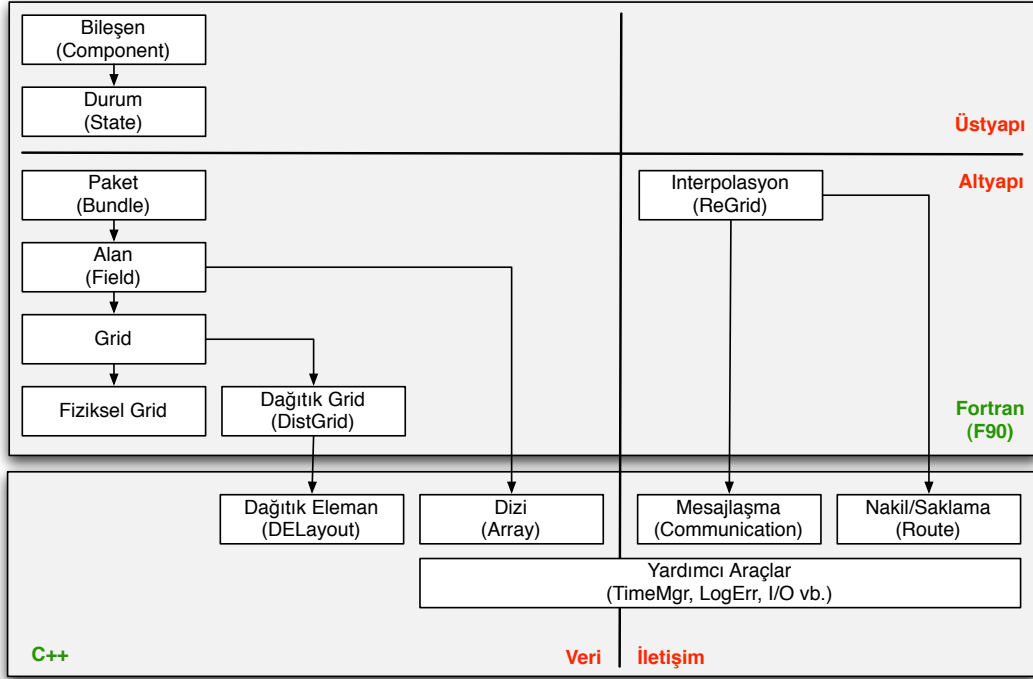
ESMF kütüphanesi son sürümü (4.0.0r), SCRIP uygulaması ile üretilmiş ağırlık matrisi kullanmadan, farklı model bileşenleri arasında interpolasyon yapılmasını sağlayacak şekilde yeniden düzenlenmiştir. Bu yöntemde, ESMF kütüphanesi kullanılarak tanımlanmış fiziksel model gridleri, birbirleri arasında dönüşüm yapabilecek şekilde işleminden geçirilebilir. Bu yöntemin, SCRIP ile yapılan interpolasyon tekniğine üstünlüğü, ağırlık matrisinin okunmasına ve bunun getireceği ek hesaplama yüküne sahip olmamasıdır. Ancak bu yöntem, bir kara kutu modeli gibi çalışmakta ve kullanıcının interpolasyon aşamasındaki müdahalesini en alt seviyeye indirgemektedir. Bu teknik hem korunumlu, hem de korunumsuz interpolasyon tiplerini desteklemektedir ve farklı model gridleri arasındaki ağırlık matrisi, başlangıç aşamasında ESMF tarafından hesaplanır ve daha sonra kullanılmak üzere saklanır.

Şekil 3.3'te, ESMF'i oluşturan temel sınıfların yapısı ve birbirleri ile olan ilişkileri görülebilir. ESMF'i oluşturan çekirdek sınıflar C++ programlama dilinde yazılmışken, daha üst seviyedeki sınıflar, Fortran 90 programlama dili ile geliştirilmiştir. Bu bakımdan ESMF; Fortran 90 ve C++ programlama dilleri ile geliştirilmiş farklı modelleri aynı yapı altında birleştirebilecek bir arayüze sahip olarak tasarlanmıştır.

ESMF, modellerin birleştirilmesini amaçlayan bir kütüphane olmasına rağmen, farklı projeler yardımı ile farklı amaçlar içinde kullanılabilir şekilde evrimleşmektedir. Bunların en önemli örneklerinden biri, henüz ilk evrelerinde bulunan kuple model sistemlerini oluşturan bileşenlerin, ağ servisleri yardımıyla çalıştırılabileceği yapıların geliştirilmesidir. Bu modellerin bilimsel ağ geçitleri ve iş akış yapıları ile kolayca entegre edilmelerini sağlayacak önemli bir özelliktir. Bu, tip uygulamalara örnek olarak OpenMI<sup>2</sup> verilebilir. OpenMI'da ESMF ağ servisleri bileşeni gibi model ile ağ servisleri arasındaki iletişimi kurmak için tasarlanmış bir uygulamadır. ESMF, model

---

<sup>2</sup><http://www.openmi.org/>



**Şekil 3.3:** ESMF’i oluşturan temel sınıflar

veya model bileşenleri içerisindeki değişkenler ile ilgili bilgilerin, yani ön verinin saklanması ve standart bir şekilde diske yazılmasını sağlayacak çeşitli yapıları da (ESMF\_Attribute) bünyesinde barındırmaktadır. ESMF özellik sınıfı, ESMF bileşenlerine veri geçitleri veya iş akış uygulamaları ile kolayca entegre edilebilmelerini sağlayacak, önemli bir özellik kazandırmaktadır. Bu çalışma kapsamında geliştirilen sınıma uygulamalarında kullanılan model ve içerdiği değişkenler ile ilgili ön verinin toplanması amacıyla ESMF özellik sınıflarından yararlanılmıştır.

### 3.3 Hesaplama Kaynağı: TeraGrid ve Küme Sistemler

Bu bölüm, çalışmada kullanılan hesaplama kaynaklarını tanıtmayı amaçlamaktadır. Böylece, geliştirilen iş akış çalışma ortamının hangi hesaplama kaynaklarında test edildiği ve hangi tip sistemler ile uyumlu olduğunu açık bir şekilde okuyucuya aktarılmış olacaktır. Çalışmada temel olarak iki tip hesaplama kaynağı kullanılmıştır. Bunlardan birincisi, bir grid sistemi diğeri ise bir geleneksel küme sistemidir.

Grid, yersel olarak birbirinden ayrık hesaplama kaynaklarını (küme sistemler veya diğeri grid sistemleri) birleştirerek bunların tek bir merkezden yönetimini ve kullanıl-

masını amaçlayan bir hesaplama ortamıdır. Grid sistemlerinin son zamanlarda gittikçe daha popüler olmasının başlıca birkaç nedeni vardır: Bunlardan ilki; tek bir küme veya grid hesaplama sistemi ile çözülemeyecek problemlerin çözülebilmesini sağlamasıdır. Bu amaçla, problemi çözecek uygulamanın grid içerisinde bulunan bir veya birkaç sisteme dağıtılarak aynı anda çalışması sağlanır. Ancak bu durumda, farklı yersel noktalardaki hesaplama kaynakları arasındaki ağ hızları, uygulamanın performansını düşürücü yönde bir etki yapabilir. Bu nedenle; grid sistemler kurulurken, sistemler arasındaki ağ yapısının da geliştirilmesi gereklidir. Grid sistemlerinin kullanım amaçlarından bir diğeri ise; farklı platformlardaki veya mimarideki sistemlerden oluşan grid yapısı içerisinde, farklı merkezler tarafından geliştirilen yazılımların test edilmesidir. Böylece, geliştirilen yazılımların farklı platformlarda denenmesi ve/veya performanslarının ölçülmesi mümkün olmaktadır. Bu tip çalışmalara örnek olarak ESMF kütüphanesi verilebilir. ESMF, belirli aralıklar ile birçok farklı mimariye sahip hesaplama kaynağında, farklı derleyici ve MPI sürümleri ile test edilmekte ve sonuçlar, uygulama geliştiricileri tarafından yazılımın geliştirilmesi, hatalarının ayıklanması ve performansının artırılması amacıyla kullanılmaktadır. Grid sistemlerinin kullanım amaçlarına verilecek son örnek ise; bilimsel ağ geçitlerinin tasarlanması ve kullanılmasına olan ihtiyaçtır. Grid sistemleri, bu tip ağ geçidi uygulamalarına veya LEAD projesi gibi, iş akış sistemlerinin ağ geçitleri ile birleştirildiği çalışmalara hesaplama kaynağı olarak hizmet vermektedir. Bu tip uygulamalar, grid sistemlerinin tek bir yetkilendirme merkezi ile kontrol edilmesinin avantajlarını kullanmaktadırlar.

Grid sistemlere verilebilecek en önemli örneklerden biri; TeraGrid [19] tir. TeraGrid, geliştirilen iş akış sisteminin, grid hesaplama kaynağı üzerinde sınanması için kullanılan grid kaynağıdır. TeraGrid, Chicago Üniversitesi GIG grubu tarafından koordine edilen ve farklı merkezlerin katkı verdiği bir grid sistemidir. Sistem içerisinde bulunan kaynaklar, herhangi bir hatayı bulmak için Inca<sup>3</sup> adı verilen bir uygulama ile belirli aralıklarla test edilmektedir.

Kullanıcı, TeraGrid kaynaklarına iki farklı şekilde erişebilir. Bunlardan birincisi; Teragrid ağ geçididir ve tek bir şifre ile tüm kaynaklara erişim, bu ağ sayfası üzerinden sağlamaktadır. İkinci yöntemde ise kullanıcı, MyProxy sunucusu üzerinden yetkilendirme alarak, TeraGrid içerisinde bulunan kaynaklara, şifresiz bir şekilde

---

<sup>3</sup><http://inca.sdsc.edu/drupal/>

alınan vekil yetkilendirmenin süresine de bağılı olarak ulaşabilir. Tüm TeraGrid kaynaklarına CTSS adı verilen bir paket yüklenmektedir. Bu yazılımın görevi, TeraGrid içerisinde bulunan hesaplama kaynaklarını standart bir hale getirmektir. Örneğin, tüm hesaplama kaynaklarında, kullanıcı alanı ile yüksek performanslı disk alanlarını gösteren çevre değişkenlerinin aynı isimle tanımlanması, grid hesaplama ile ilgili olan yazılımların (Globus, GridFTP vb.) benzer sürümlerinin bulunması veya erişim düğümündeki yazılımların çevre değişkenlerinin SoftEnv [43] ve Modules [44] programları ile kontrol edilmesi, bu yazılımın görevleri arasında sayılabilir. CTSS uygulamasının TeraGrid kaynakları üzerindeki varlığı, geliştirilen iş akış sisteminin TeraGrid ile entegre edilmesini kolaylaştırmıştır.

Çalışmada, TeraGrid üzerinden NCAR tarafından sisteme entegre edilen Frost makinası kullanılmıştır. Frost, üzerinde AIX (bir UNIX versiyonu) işletim sistemi kurulu bir IBM BG/L sistemdir. BlueGene, yapısı itibariyle hesaplama düğümleri üzerinde mikro çekirdeğe sahip özel bir işletim sistemi çalıştırmaktadır ve BG/P modelinden farklı olarak, tek bir hesaplama düğümü üzerinde birden fazla OpenMP iş parçacığı çalıştıramaz. İşletim sistemi, mikro çekirdeğe sahip olduğundan hesaplama düğümleri üzerinde çalıştırılabilecek komutlar sınırlıdır. Bu sebeple, BG/L modeli üzerine verilecek işler, hesaplama düğümüne özel olarak erişim yapılan düğümler üzerinde IBM XL derleyicisi ile derlenir ve sisteme iş, Cobalt adı verilen bir yazılım ile teslim edilir. Frost içerisinde bulunan işlemciler, farklı ve çok kullanılan genel işlemci gruplarına (32, 64, 128, 256, 512 vb.) ayrılmışlardır ve sisteme verilen iş, kendisine en yakın sayıda işlemci içeren grup üzerinde otomatik olarak çalışmaya başlar. TeraGrid ile iş akış sisteminin entegre edilmesi ile ilgili detaylı bilgi Bölüm 4.5'te bulunabilir.

Kullanılan ikinci sistem, küme tipi geleneksel hesaplama sistemlerine iyi bir örnek olarak verilebilecek olan NCAR'a ait Bluefire sistemidir. Bluefire; IBM Power 575 hesaplama düğümlerinden oluşan, 4096 adet işlemciye (veya çekirdeğe) sahip, 7.7 Tflop işlem gücünde ve Infiniband (4X, DDR) performans ağı kullanan bir hesaplama kaynağıdır. Sisteme erişim, OTP (tek kullanımlık şifre) ve SSH protokolünün birleşiminden oluşan bir yetkilendirme sistemi ile sağlanmaktadır.

Yukarıda kısaca tanımlanan hesaplama kaynakları, farklı yetkilendirme ve erişim sistemleri kullanmakla birlikte, geliştirilen çalışma ortamının her iki hesaplama sistemi ile uyumlu bir şekilde çalışması için çeşitli araçlar (farklı dillerde örtü betikler, yeni



Kepler aktörleri vb.) geliştirilmiş ve sınaama uygulamalarının, bu sistemler üzerinde test edilmesi sağlanmıştır. Bu amaçla, sadece hesaplama kaynağına özgü Kepler aktörlerinin değiştirildiğı ve kurulan iş akışının farklı bir hesaplama kaynağı üzerinde çalıştırıldığı örnek iş akışları da tasarlanmıştır.

#### 4. UYGULAMA ÖRNEĞİ - I

İş akışı ve çerçeve yaklaşımlarının birlikte kullanılmasına örnek olması açısından, bu çalışma için iki farklı sına uygulama tasarlanmıştır. Bunlardan birincisi bu bölümde detaylı olarak tanımlanacak olan; CCSM küresel dolaşım modeli kullanılarak geliştirilen uygulamadır. Uygulamada kullanılan CCSM modeli, ESMF kütüphanesi ile belirli bir seviyeye kadar entegre edilmiş bir yer sistem modelidir. Ancak, bu uygulama ile, CCSM modelinin iş akışı ve ESMF kütüphanesi ile daha ileri seviyede birleştirilmesine ve her iki yaklaşımın özelliklerinin kullanılmasına yönelik yeni metodolojilerin sınılanması amaçlanmaktadır.

İkinci uygulama ise; bir kuple model sisteminin aynı metodoloji kullanılarak, iş akışı ortamına aktarılmasını amaçlamaktadır. Kuple model sistemi için kullanılacak sınırlı alan modelleri, atmosfer bileşeni için WRF, okyanus bileşeni için ise ROMS'tur. Bu uygulamada, modeller öncelikle ESMF kütüphanesi kullanılarak entegre edilecek ve daha sonra iş akışı ortamına aktarılacaktır. Kuple model sistemi ile tasarlanan sına uygulama ile ilgili detaylı bilgi bir sonraki bölümde bulunabilir (Bkz. Bölüm 5).

##### 4.1 Küresel Dolaşım Modeli: CCSM4

Bölüm 2.4'de tanımlanan metodolojinin sınılanması amacıyla, birçok alt yer sistem modeli bileşeninden oluşan ve NCAR tarafından geliştirilen, CCSM küresel dolaşım modelinin son sürümü kullanılarak bir sına uygulama geliştirilmiştir.

CCSM4, CCSM model ailesinin geliştirme aşamasındaki son sürümüdür ve 2010 yılı içinde son kullanıcılar tarafından kullanılabilir şekilde yayınlanması planlanmaktadır. CCSM modelinin çalışmada kullanılan sürümü, altı farklı bileşen içermektedir. Bunlar; atmosfer, kara, okyanus, deniz buzu, glasiyer ve tüm bu bileşenlerin bir arada çalışmasını sağlayan kuple bileşenidir. Bir önceki sürümden farklı olarak, CCSM içerisine glasiyerlerin (kara yüzeyi üzerindeki kalıcı buz tabakaları) modellenmesi amacıyla üç boyutlu bir buz tabakası modeli olan GLIMMER [45] entegre edilmiştir.

Ayrıca, eski sürümde aynı olan atmosfer ve kara alt modellerine ait gridler bu sürüm ile birlikte farklı çözünürlüklerde tanımlanabilmektedir.

CCSM bileşenlerinin birbirleri ile kuple edilmesi amacıyla kullanılan yazılım MCT (Bkz. Bölüm 3.2) uygulamasıdır. CCSM'in MCT sürümüne ek olarak, ESMF sürümü üzerinde çalışmalar devam etmektedir ve CCSM modelinin yayınlanacak bir sonraki genel sürümünde her iki kuple bileşenini birlikte desteklemesi amaçlanmaktadır.

CCSM'in çalışmada kullanılan geliştirme aşamasındaki sürümü, diğer yer sistem modellerinden farklı olarak, modelin konfigürasyonunun yapılması için XML standartlarından yararlanmaktadır. XML dosyaları içerisinde bulunan model seçenekleri, derleme seçenekleri ve model ile ilgili ön veriler, PERL betikleri ile ayıklanıp anlamlı bilgilere dönüştürülmekte ve model için gerekli tanımlama dosyaları oluşturulmaktadır. Bu dosyalar içerisinde XML yapılarının kullanılması, CCSM modelinin iş akış sistemine entegre edilmesi aşamasını oldukça kolaylaştırmış ve model konfigürasyon veya çevre dosyalarında meydana gelen değişikliklerin, Kepler uygulaması için yaratılan aktörler üzerindeki etkilerinin en alt seviyeye indirilmesini sağlamıştır.

CCSM modeli ile yapılan simülasyonların doğruluğunun artırılması amacıyla yapılan testlerde, deniz yüzeyi sıcaklıklarının gözlemler ile farklı olduğu bölgelerde, bölgesel okyanus modellerinin çalıştırılması için çeşitli araştırmalar yapılmaktadır. Bu amaçla, CCSM modelinin POP okyanus modeli bileşeni, MCT uygulaması kullanılarak ROMS modeli ile kuple edilmiştir. Bu yolla, bölgesel okyanus modelinin POP bileşenine daha doğru deniz yüzeyi sıcaklığı sağlaması amaçlanmaktadır.

## 4.2 Test Uygulaması ve Amaç

İş akış uygulamasını sınamak için geliştirilen uygulama, bir önceki bölümde tanımlanan CCSM modelinin iki farklı atmosferik model çözünürlüğü ( $2.5^{\circ} \times 1.9^{\circ}$  ve  $1.25^{\circ} \times 0.9^{\circ}$ ) ve B\_2000\_TRACK1 bileşen seti ile TeraGrid ve geleneksel küme sisteminde çalıştırılmasını, otomatik olarak köken ve ön veri bilgilerinin toplanmasını ve bu bilgilerin standart bir şekilde saklanmasını amaçlamaktadır.

B\_2000\_TRACK1, günümüz iklimini modellemek amacıyla geliştirilmiş, tüm CCSM alt model bileşenlerinin aktif durumda olduğu ve atmosfer modeli CAM'ın sürüm 3.5 ve TRACK1 modunda çalıştığı bir bileşen setidir. CCSM içerisinde, buna benzer

şekilde önceden tanımlanmış birçok farklı bileşen seti mevcuttur. Çalışmada, her iki simülasyon için CCSM atmosferik modelinin sonlu hacim (FV) dinamik çekirdeği kullanılmış ve diğer parametreler aynı bırakılmıştır. Ayrıca her bir benzetim için toplam simülasyon süresi bir yıl olarak tanımlanmıştır.

Farklı çözünürlükte çalıştırılan modellerin sonuçları, farklı iş akış diyagramları ile son işlemde geçirilerek, sonuç ürünlerin (görsel figürler, ortalamaları alınmış dosyalar) elde edilmesi sağlanmıştır. İş akışı tarafından, hesaplama ortamı üzerinde çalıştırılan CCSM benzetimi tamamlandıktan sonra, model çıktılarının işlenmesi amacıyla geliştirilen son işleme iş akışı, otomatik olarak çalıştırılır. Model, düşük çözünürlüklü benzetim için TeraGrid üzerinden erişilen ve NCAR tarafından kurulan Frost makinasında, toplamda 384 işlemci üzerinde, yüksek çözünürlüklü benzetim ise, NCAR'a ait Bluefire (IBM küme) makinası üzerinde çalıştırılmıştır. CCSM modelinin çalışma zamanına örnek vermek gerekirse, Frost makinası üzerinde bir yıllık benzetim, yaklaşık olarak dokuz saat sürmektedir.

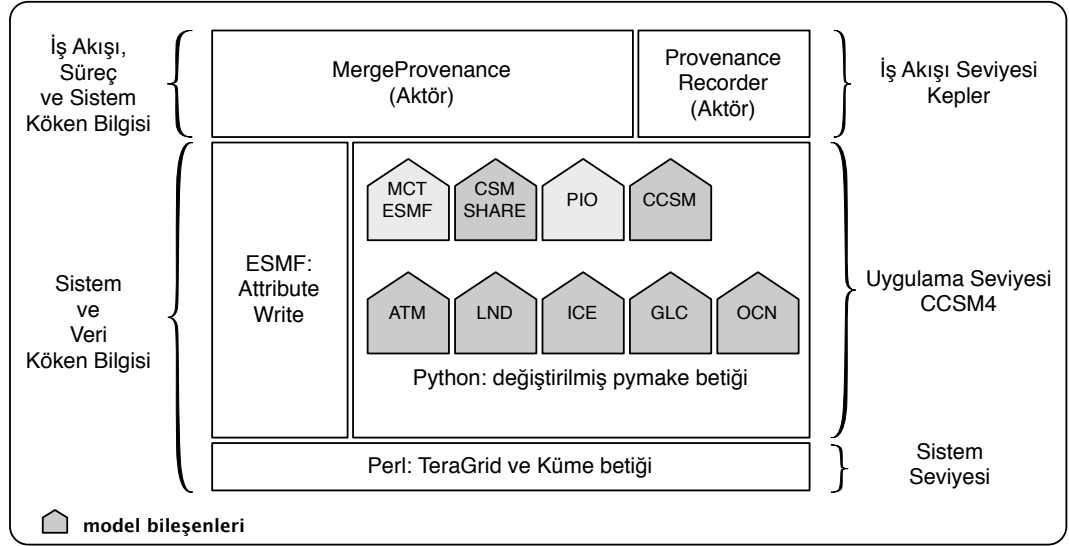
Bu çalışma açısından bakıldığında, benzetimden gelen bilimsel sonuçların analiz edilmesi ve farklı çözünürlükteki benzetimler arasındaki farkların araştırılması, tezin asıl amacı değildir. CCSM modeli kullanılarak tasarlanan uygulama örneğinin asıl amacı modelin iş akış sistemine entegre edilmesi, ön veri ve köken bilgisinin otomatik olarak ve standartlaştırılmış bir şekilde toplanması ve bu amaçla metodolojiler geliştirilmesidir.

### **4.3 Köken ve Ön Veri Bilgisinin CCSM Modelinden Toplanması**

Bir önceki bölümde kısaca tanımlanan sınağa uygulamasında kullanılan CCSM modelinin, kendi kendisini tanımlaması amacıyla, köken bilgisinin iş akış sistemi tarafından otomatik bir şekilde toplanması gereklidir. Bu amaçla, model sistemi içerisinde farklı uygulama seviyelerinde bulunan köken bilgisinin bir araya getirilmesi önemlidir. Birçok farklı bileşenden oluşan CCSM gibi bir yer sistem modeli için bu bilgiyi toplamak ve standart bir şekilde sunmak, modelin iş akış sistemi ile entegre edilmesi kadar karmaşık ve zor bir süreçtir. CCSM küresel dolaşım modeli özelinde düşünüldüğünde, köken bilgisinin model içerisinde farklı noktalarda, uygulama seviyelerinde ve farklı alt yer sistem model bileşenlerinde bulunduğu görülür. IPCC raporlarında sonuçları kullanılan CCSM küresel dolaşım modeli için

köken bilgisinin toplanarak, kullanıcı tarafından hata ayıklama, sonuçları tekrar etme, farklı benzetimlerdeki sonuçları karşılaştırma ve model performansının artırılması açısından kullanılması oldukça önemlidir.

Daha önceki bölümlerde kısaca bahsedilen zorlukların aşılması ve köken bilgisinin toplanması amacıyla, hiyerarşik bir model yaratılması gerekmektedir. Şekil 4.1’de, köken bilgisinin CCSM modelinden toplanması için önerilen hiyerarşik yapı görülebilir:



Şekil 4.1: Köken bilgisinin toplanması için geliştirilen hiyerarşik yapı

Şekil dikkatlice incelendiğinde görülebileceği gibi, üç uygulama tabakasından oluşan bir yapı tasarlanmıştır ve her bir seviyenin topladığı bilgiler ve içerdikleri özellikler, aşağıdaki listede bulunabilir:

- **Sistem Seviyesi:** En alt seviye olan sistem seviyesinden toplanan bilgi, sistem köken bilgisidir. Bu amaçla, PERL dilinde bir betik yazılmıştır ve temel olarak iki ana göreve sahiptir. Bunlardan birincisi; uzak sistemde çalıştırılacak uygulama için gerekli ortamı hazırlamaktır. TeraGrid gibi karmaşık bir hesaplama ortamında uygulama çalıştırmak için birçok farklı problemin çözülmesi gereklidir. Örneğin, TeraGrid sistemine bağlandıktan sonra, iş akış sistemine entegre edilmiş uygulamanın çalışması için bazı çevre değişkenlerinin ayarlanmasına (derleyici ve kütüphaneler ile ilgili özel çevre değişkenleri vb.) veya bazı yazılımların kurulmasına (ESMF, MCT vb.) ihtiyaç olabilir. Genel olarak bu bilgiler, uygulama

çalıştırılmadan tanımlanmalı ve/veya gerekli yazılımlar kurulmalıdır. Ancak bu durumda, gerekli uygulamaların kurulu olduğu dizinlerin, uzak sistem üzerinde otomatik olarak bulunması gereklidir. Yazılan PERL betiği, bu gerekleri yerine getirmek ve iş akış sistemi için standart bir çevre yaratmaktan sorumludur.

Geliştirilen betik, hem SoftEnv [43] hem de Modules [44] uygulamalarını desteklemektedir ve çalışma mantığı son derece basittir. İş akış uygulaması ile sisteme iş verildiğinde, betik öncelikle SoftEnv veya Modules uygulamalarını bularak, bunları kullanıcı çevre değişkenine ekler. Bu aşamadan sonra betik, daha önce kullanıcı tarafından tanımlanmış uygulamaları, SoftEnv ve/veya Modules uygulamalarına ait veri tabanından bularak, komut çalışmadan önce kullanıcının çevre değişkenlerine ekler. Sonuç olarak, betik tarafından yaratılan komut zinciri, işin çalışması için kullanılır. Geliştirilen bu betik ile ayrıca iş akış uygulaması tarafından kullanılmış değişkenlerin, işin çalıştığı sisteme özel karşılıkları ile değiştirilmesi sağlanır.

Betiğin ikinci görevi ise; uzak hesaplama kaynağından otomatik olarak sistem köken bilgisini toplayarak, bu bilgiyi XML formatında yazmaktır. Bu dosya temel olarak, bağlanılan sistemin ismini, işlemci mimarisi ile ilgili bilgileri, işletim sisteminin tipini ve işletim sistemine yapılan yamaların sürüm bilgileri gibi birçok farklı bilgiyi içermektedir. Şekil 4.2’de, bu betik tarafından NCAR’a ait Frost makinasından toplanmış sistem köken bilgisi görülebilir.

```
1 |<?xml version="1.0"?>
2
3 <system_provenance>
4   <machine>
5     <property name="login_node" value="fr0103en" />
6     <property name="cpu_vendor" value="" />
7     <property name="cpu_model" value="POWER5 (gr)" />
8     <property name="os_distribution" value="SUSE LINUX Enterprise Server 9 (ppc)" />
9     <property name="os_version" value="9" />
10    <property name="os_patch_level" value="4" />
11    <property name="os_type" value="Linux" />
12    <property name="os_kernel" value="2.6.5-7.316-pseries64" />
13    <property name="os_architecture" value="ppc64" />
14  </machine>
15 </system_provenance>
16
```

Şekil 4.2: Örnek sistem köken bilgisi çıktısı (NCAR Frost, IBM BG/L)

- **Uygulama Seviyesi:** İkinci seviye olan uygulama seviyesinde, CCSM modelinden derleme aşamasındaki bilgileri toplamak amacıyla, Oak Ridge Ulusal Laboratuvarı (ORNL) ve North Carolina State Üniversitesi tarafından geliştirilmiş bir betik serisi kullanılmıştır. Python programlama dilinde yazılmış olan betik, temel olarak

sistem köken bilgisinin toplanmasını amaçlamaktadır. Geliştirilen betik, çalışılan uygulamaya özeldir, ancak kullanılan yöntem başka çalışmalar için uyarlanabilir. Bu amaçla, betiğin genel yapısı temel alınarak, sınama uygulamasında kullanılan CCSM modeli için baştan yazılmış ve bazı iyileştirmeler yapılmıştır. Betik üzerinde yapılan temel değişiklikler aşağıdaki listede bulunabilir:

- Uygulama örneğinde, köken bilgisinin saklanması amacıyla XML veri formatı temel olarak kabul edilmiştir. Bu sebeple, orjinal betikte bulunan veri tabanı bağlantısı (MySQL) ile ilgili kısımlar çıkartılarak, yerine daha önce belirlenmiş bir yapıda XML dosyası üretmek için gerekli alt programlar eklenmiştir. Betik tarafından üretilen XML dosyası içerisinde, PERL betiğinde toplandığı gibi sistem köken bilgisi, CCSM modelinin derleme aşamasında kullanılan derleyiciler, derleyici seçenekleri ve bunların sürümleri, tanımlanan çevre değişkenleri ve CCSM alt bileşenine ait özel bilgiler bulunmaktadır. CCSM model bileşenine ait bilgiler içerisinde; bileşen ismi, çalıştırılabilir veya kütüphane dosyasının hesaplanmış MD5 sınama değeri ve ne zaman yaratıldığı gibi bilgiler bulunmaktadır. MD5 sınama değeri, belirli bir benzetimin diğerlerinden ayrılması için tekil anahtar olarak kullanılabilir. CCSM alt model bileşenlerinden köken bilgisinin toplanması için betik, öncelikle alt modele ait "makefile" dosyasının yapısını (veritabanı) oluşturur. Oluşturulan veri tabanı içerisinden derleyici ile ilgili komutlar, betik tarafından bulunarak, bu komutlar ile ilgili bilgiler (derleyici tipi, sürümü vb.) toplanır. Daha sonra yine oluşturulan veritabanından CCSM alt modeli derlenirken, tanımlanan çevre değişkenleri toplanır. Bu çevre değişkenleri, model ile ilgili konfigürasyon bilgilerini tutmaktadır ve bu nedenle son derece önemlidir. Daha sonra, toplanan bu bilgi XML dosyası içerisine belirli bir yapıda yazılır. Betik ile CCSM modelinden toplanmış örnek sistem köken bilgisi Şekil 4.3'de görülebilir.
- Python betiği, iç içe yuvalanmış dizin yapısı bulunan uygulamalardan (örneğin, MCT) sistem köken bilgisi toplamak için değiştirilmiştir. "Makefile" dosyasından oluşturulan veritabanının incelenmesi ve bunun içerisinden gerekli bilgilerin alınması, özellikle çok kaynak dosyası içeren yazılımlar için zaman alan bir işlemdir. Bu sebeple, betik CCSM modelinin derleme

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <system_provenance>
3   <run>
4     <component>
5       atm
6     </component>
7     <executable>
8       /ptmp/turuncu/kepler.case.BTR1.f19_g16/lib/libatm.a
9     </executable>
10    <checksum>
11      aaff4ea9b7e01b33083206fdf07b07ad
12    </checksum>
13    <user>
14      turuncu
15    </user>
16    <time>
17      09-06-30 11:33:33
18    </time>
19  </run>
20  <machine>
21    <property name="mach" value="bluefire"/>
22    <property name="site" value="NCAR"/>
23    <property name="arch" value="IBM"/>
24    <property name="os_name" value="AIX"/>
25    <property name="arch_bit" value="64bit"/>
26    <property name="login_node" value="be1005en"/>
27    <property name="os_arch" value=""/>
28    <property name="os_version" value="5"/>
29    <property name="os_release" value="3"/>
30  </machine>
31  <compilers>
32    <compiler>
33      <property name="command" value="mpcc_r"/>
34      <property name="description" value="IBM XL C/C++ Enterprise Edition for AIX, V9.0"/>
35      <property name="flags" value="-DFORTRAN_SAME -DMAXPATCH_PFT=4 -DLSMLAT=1 -DLSMLON=1
-DPLON=144 -DPLAT=96 -DPLEV=26 -DPCNST=3 -DPCOLS=16 -DPTRM=1 -DPTRN=1 -DPTRK=1 -DSTAGGERED
-DSPMD -DSPMD -DCO2A -DAIX -DSEQ_MCT -q64 -O2"/>
36      <property name="language" value="C"/>
37      <property name="version" value="09.00.0000.0002"/>
38    </compiler>
39    ...
40  </compilers>
41  <environment>
42    <variable name="AIXTHREAD_SCOPE" value="S"/>
43    <variable name="ARCH" value="IBM"/>
44    ...
45  </environment>
46 </system_provenance>

```

**Şekil 4.3:** CCSM modelinin atmosferik bileşeninden (CAM) toplanmış örnek sistem köken bilgisi çıktısı (NCAR Frost, BG/L)

yapısına uydurulmuş ve model ile ilgili bilgilerin toplanmasının daha hızlı ve performanslı bir şekilde yapılması sağlanmıştır. Ancak bu çözüm, CCSM modeline özeldir ve genelleştirilemez.

- Geliştirilen Python betiğinin CCSM modeli ile kullanılması amacıyla, CCSM modelinin derleme betikleri değiştirilmiş ve "make" komutları yerine CCSM tarafından Python betiklerinin çalıştırılması sağlanmıştır. CCSM alt modellerinin her biri kendisine ait derleme yapısı içerdiği için, sonuçta Python betiği ile oluşturulacak çıktı dosyaları da, herbir alt model bileşeni için ayrı olacaktır. Bu dosyalar, daha sonra Kepler uygulaması içerisindeki bir aktör yardımıyla birleştirilebilir yapıdadır.

Yukarıda, kısaca görevi ve yapısı tanımlanan betiğin çalışması ile ilgili iş akış ekler kısmında bulunan Şekil B.1’de detaylı olarak görülebilir.



- **İş Akışı Seviyesi:** Şekil 4.2’te görülen en üst seviye olan iş akışı seviyesinde, tasarlanan iş akışı ile ilgili köken bilgisinin toplanması amacıyla Kepler "ProvenanceRecorder" aktörü kullanılmaktadır. Bu aktör, varsayılan olarak köken bilgisini SQL ve türevi bir veri tabanında saklamak üzere geliştirilmiştir. Köken bilgisi, ESG portalına veya başka bir veri portalına aktarmak istenildiğinde, temel veri formatı olarak XML kullanılmalıdır. Bu sebeple, aktör değiştirilerek XML çıktısı üretmesi sağlanmıştır. Daha alt seviyelerde toplanan köken ve ön veri bilgisinin birleştirilmesi, yine bu seviyede yapılabilir. Ancak, birçok alt bileşenden oluşan CCSM modeli için toplanan tüm köken bilgisinin, tek bir veri yapısı altında toplanması, oldukça karmaşık olabilir.

#### 4.4 CCSM Modeli Üzerinde Yapılan Değişiklikler

Daha önceki bölümde (Bkz. Bölüm 2.1) detaylı olarak bahsedildiği gibi, dört farklı tip köken bilgisi tanımlanabilir. Bunlardan en önemlilerinden biri, üretilen veya kullanılan verinin evrimini tanımlayan veri köken bilgisidir. Bu örnek uygulama için CCSM küresel sirkülasyon modelinden, veri köken bilgisinin toplanması amacıyla ESMF özellik (ESMF\_Attribute) yapısından yararlanılmıştır. ESMF özellik yapısı, temel olarak model sistemini oluşturan bileşenler, bu bileşenlere ait değişken veya değişken grupları ve bileşenler arasında transfer edilen veri ile ilgili ön veriyi tutmak üzere geliştirilmiş ve tasarlanmış bir yapıdır. CCSM modelinin kuple bileşeni içerisindeki değişken bilgilerinin, model çalışmaya başladığında XML formatında diske yazılmasını ve bu bilginin, toplanan köken bilgisi içinde kullanılmasını sağlamak amacıyla, aşağıdaki listede detayları görülen CCSM modeli üzerinde çeşitli değişiklikler yapılmıştır.

- CCSM modelinin varsayılan kuple veya farklı yer sistem modellerinin birleştirilmesi için kullanılan bileşeni, MCT kütüphanesini kullanmaktadır. Ancak CCSM, model içindeki zamansal hesaplamaların yapılabilmesi için, amacıyla ESMF kütüphanesinin saat ile ilgili metodlarına (ESMF\_Clock) ihtiyaç duymaktadır. Varolan bu yapının, ESMF kütüphanesinin son geliştirici sürümü (ESMF\_4\_0\_0r\_beta\_snapshot\_02) ile çalışmasını sağlamak amacıyla, CCSM modelinin zaman yönetimini yapan modülü üzerinde gerekli değişiklikler

yapılmıştır. Bu şekilde, CCSM modelinin ESMF kütüphanesinin son geliştirici sürümü ile derlenmesi ve çalışması sağlanmıştır.

- CCSM modeli içinde bulunan merkezi kuple modülü, farklı yer sistem modellerine ait değişkenlerin XML formatında yazılmasını sağlamak amacıyla değiştirilmiştir. Bu aşamada, CCSM kuple bileşeni içerisinde bulunan değişken listesi, boş ESMF alan değişkenleri (ESMF\_Field) yaratmak için kullanılmış ve bu boş ESMF değişkenlerine değişken ismi, değişken tanımlaması, standart isim ve birim gibi farklı ön veri bilgileri eklenmiştir. Elde edilen bu alan değişkenleri, ESMF girdi (ESMF\_Import) ve çıktı (ESMF\_Export) değişkenleri şeklinde ayrılarak, ESMF durum yapıları içerisine gömülmüş ve ESMF özelliklerinin model içinden alınması için tasarlanmış ESMF\_AttributeWrite metodu kullanılarak diske yazılmıştır. Yazılan dosya, aynı zamanda kuple bileşen için kuple yönünü gösteren durum bilgisini de içermektedir (Bkz. Şekil 4.4).
- Alan değişkenlerine ait ön veri bilgisinin hazırlanması amacıyla, kuple bileşenine ait standart isim bilgileri, CF standartlarına uygun olarak tamamlanmıştır. Bu aşamada, standart isim bilgisi olmayan değişkenlere ait bilgiler için CF karşılıkları kullanılmış veya yeni standart isimler, CF standardına uygun olarak üretilmiştir. Ön veri tamamlama aşamasında her bir CCSM alt model bileşeninin geliştirilmesinden sorumlu gruptan, önerilen standart isim tablosunun kontrol edilmesi istenmiş ve olası yanlış tanımlamalardan bu şekilde sakınılmıştır. Üretilen bu bilgiler CCSM'in bir sonraki sürümü olan CCSM4'te kullanılacaktır.

CCSM modelinden elde edilen veri köken bilgisine ait örnek bir XML çıktısı, Şekil 4.4'te görülebilir. Örnek XML dosyası, CCSM'in kuple bileşeninden toplanmış ve CCSM'in atmosferik alt model bileşeni olan CAM tarafından kuple bileşenine gönderilen değişkenleri içermektedir. Çıktı dosyasının tanımlayıcı kısmında bulunan N/A (Not Available) değerleri, model bileşeni ile ilgili herhangi bir ön veri bilgisinin eklenmediğini göstermektedir. Bu kısım, CCSM modelinin ESMF ile entegrasyonu tamamlandığında ve CCSM modelinin ESMF sürümü yayınlandığında doldurulacaktır. Bu şekilde, her bir alt model bileşenine ait ön veri bilgisi, modelin içine gömülebilecek ve modelin kendi kendisini tanımlaması sağlanacaktır.

```

1 <model_component name="N/A" full_name="N/A" version="N/A"
2 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xsi:schemaLocation="http://www.esmf.ucar.edu file:/esmf_model_component.xsd"
4 xmlns="http://www.esmf.ucar.edu">
5
6 <variable_set>
7   <variable   Export="false"
8             Import="true"
9             StandardName="height"
10            LongName="Height at the lowest model level"
11            Name="Sa_z"
12            Units="m" />
13  <variable   Export="false"
14            Import="true"
15            StandardName="eastward_wind"
16            LongName="Zonal wind at the lowest model level"
17            Name="Sa_u"
18            Units="m s-1" />
19  <variable   Export="false"
20            Import="true"
21            StandardName="northward_wind"
22            LongName="Meridional wind at the lowest model level"
23            Name="Sa_v"
24            Units="m s-1" />

```

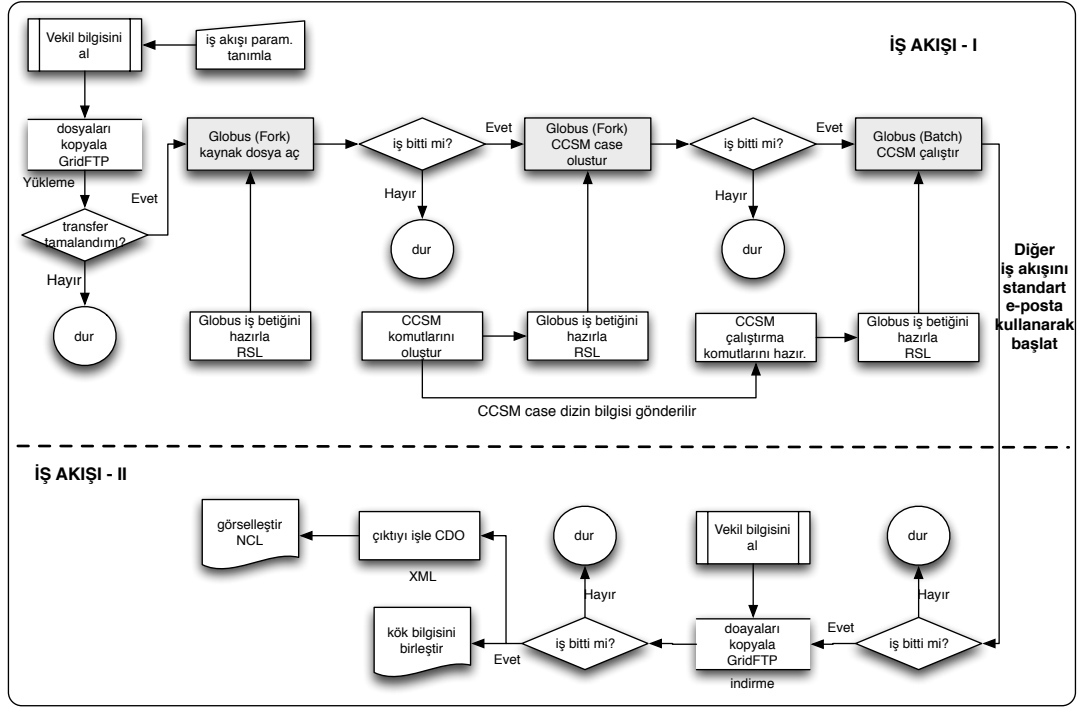
**Şekil 4.4:** Örnek veri köken bilgisi çıktısı (CCSM Kuple bileşeni, atmosferik model değişkenleri)

#### 4.5 Geliştirilen Aktörler ve Tasarlanan İş Akışı

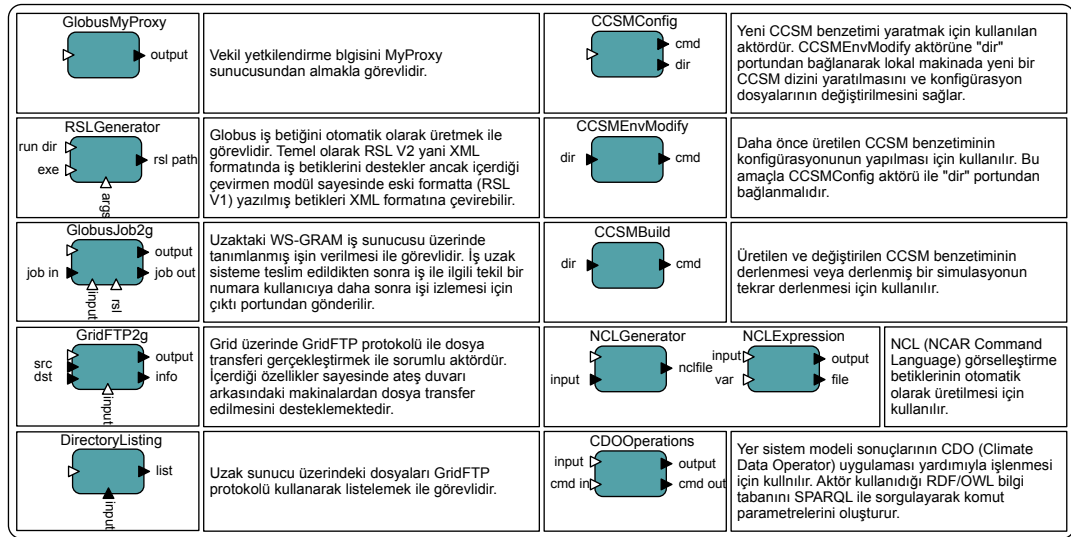
Önceki bölümlerde detaylı olarak tanımlanan test uygulamasının Kepler iş akış sistemine aktarılması için tanımlanan mantıksal iş akış diyagramı, Şekil 4.5’de görülebilir.

Şekil, detaylı olarak incelendiğinde iki temel iş akışı görülmektedir. Bunlardan birincisi; CCSM modelinin TeraGrid ortamında çalıştırılmasından sorumludur. Temel olarak, merkezi TeraGrid yetkilendirme sunucusundan gerekli yetkilerin alınmasını ve seçilen TeraGrid kaynağı üzerinde CCSM modelinin kurularak, tanımlanan benzetimin sisteme verilmesini içermektedir. İş, uzak sisteme verildikten sonra birincil iş akış tasarımı son bulur. Bu aşamadan sonra, işin bitmesi ve biten işin otomatik olarak ikinci iş akışını tetiklemesi beklenir. İkinci iş akışı, biten birinci iş akışı sonucunda oluşan çıktıları, yerel ve/veya uzak bir sisteme kopyalayarak, sonuçları son işlemde geçirmekten ve sonuçları görselleştirmekten sorumludur.

Şekil 4.5’te mantıksal iş akışı görülen test uygulamasının, Kepler iş akış ortamına aktarılması için birçok yeni Kepler aktörü geliştirilmiştir (Şekil 4.6). Bu aktörler; hesaplama ortamı ile ilgili aktörler, yer sistem modeli ile ilgili aktörler, görselleştirme amacıyla geliştirilen aktörler ve yardımcı aktörler olarak dört temel grup altında toplanabilir. Aşağıdaki listede, her bir grup ile ilgili detaylı bilgi bulunabilir:



Şekil 4.5: CCSM modeli mantıksal iş akışı tasarımı



Şekil 4.6: CCSM iş akışı için tasarlanmış aktörler

- Hesaplama Ortamı Aktörleri:** Kepler uygulaması, grid ile etkileşimi sağlayacak aktörler içermesine rağmen bu aktörler, Globus'un eski sürümleri ile uyumlu çalışmaktadır ve bu nedenle yeni grid aktörlerinin tasarlanmasına ihtiyaç duyulmuştur. Hesaplama sistemine özgü aktörler; temel olarak grid sistemini kullanmak için gerekli olan yetkilendirme aktörü (MyProxyLogon), dosya transferi

için "GridFTP2g", sisteme WS-GRAM (Grid Resource Allocation and Management) sunucusu üzerinden iş verilmesini sağlayan "GlobusJob2g" ve GRAM iş betiklerinin otomatik oluşturulması için "RSLGenerator" aktörlerini içermektedir. Bu aktörlere ek olarak, uzak sistemde bulunan dosyaların, GridFTP protokolü kullanarak listelenmesi için "DirectoryLisiting" aktörü tasarlanmıştır. Aktör, CCSM küresel dolaşım modeli sonuçlarına ait dosyaların bir listesini üreterek, bu bilgiyi GridFTP aktörüne dosya transferinin başlatılması için aktarmaktadır. Bu aşamada, tüm grid hesaplama ortamı ile ilgili aktörlerin geliştirilmesi için Globus CoG (Commodity Grid Kits) Java kütüphanesi [46] kullanılmıştır.

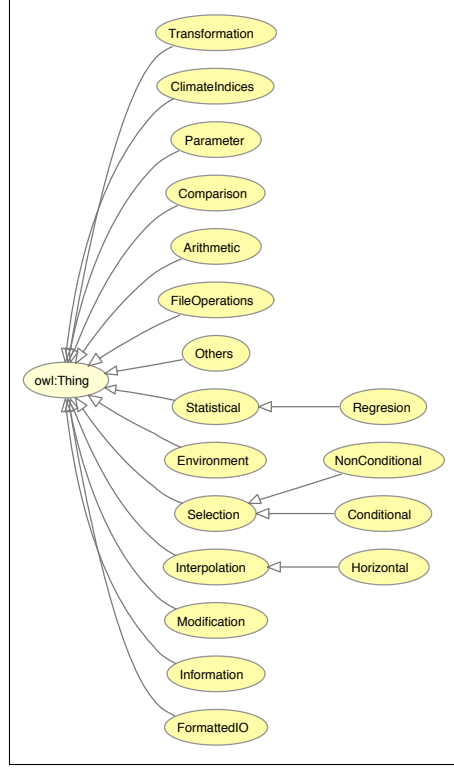
- **Yer Sistem Modeline Özgü Aktörler:** CCSM modelinin, Kepler iş akışı üzerinden kontrol edilmesini sağlamak amacıyla geliştirilen aktörler; "CCSMConfig", "CCSMEnvModify" ve "CCSMBuild" aktörleridir. Bu aktörler, yeni bir CCSM simülasyonu yaratmak ve bunu derlemek için gerekli olan komutları, kullanıcının daha önceden tanımladığı parametreler doğrultusunda üretir. "CCSMConfig" aktörü, "CCSMEnvModify" aktörü ile birlikte kullanıldığında, CCSM konfigürasyon dosyalarını değiştirecek komutların üretilmesi için kullanıcı dizini altında yaratılan dizin (.earth) içerisine, CCSM simülasyonunun bir kopyasını oluşturur. Bu aktörler, CCSM'in XML formatındaki ayar dosyalarını kullanarak aktör parametrelerini otomatik ürettikleri için CCSM sürümlerindeki değişikliklerden en az düzeyde etkilenirler. Kullanılacak yer sistem modelinin konfigürasyon dosyalarının, basit ASCII dosyaları yerine CCSM modelinde tanımlandığı gibi XML formatında olması, modelin Kepler iş akışı gibi diğer üçüncül uygulamalarla entegrasyon sürecini hızlandıran en önemli etkenlerden biridir.
- **Sonuçların Analiz Edilmesi İçin Geliştirilen Aktörler:** Bu amaçla kullanılan aktörler, NCL<sup>1</sup> ve CDO<sup>2</sup> uygulamalarının Kepler ile kullanılmasını sağlamak için oluşturulmuştur. CDO uygulamasına ait bir komut, parametreler ve çevre değişkenleri gibi iki temel bileşenden oluşur. Bu bileşenlerin, aktörler tarafından dinamik olarak üretilmesi için OWL<sup>3</sup> dili kullanılarak, bir bilgi tabanı oluşturulmuştur (Şekil 4.7). Komut parametreleri ve bu komutlara bağlı özelliklerin, geliştirilen Kepler aktörü tarafından sorgulanması amacıyla bu bilgi tabanı SPARQL sorgulama dili

<sup>1</sup>NCAR Command Language: <http://www.ncl.ucar.edu/>

<sup>2</sup>Climate Data Operator: <http://www.mpimet.mpg.de/fileadmin/software/cdo/>

<sup>3</sup>Web Ontology Language: <http://www.w3.org/TR/owl-features/>

kullanır. Oluşturulan bu bilgi tabanı belirli bir ve/veya çıktı tipine özgü işlemlerin tanımlanması amacıyla diğer bir bilgi tabanı içerisinde kullanılabilir şekilde tasarlanmıştır.



**Şekil 4.7:** CDO uygulaması için geliştirilmiş OWL bilgi tabanı genel sınıf yapısı

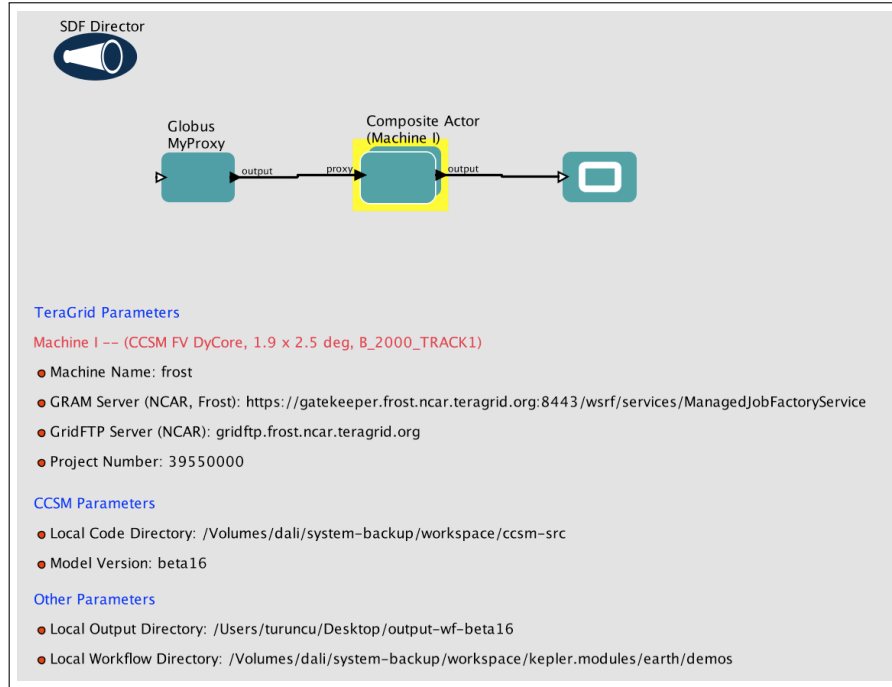
CDO komutları temel olarak; komut, parametreler, girdi ve çıktı dosyaları ve komutun işleyişini etkileyen çevre değişkenlerinin birleşiminden oluşmaktadır. OWL dili ile üretilmiş CDO bilgi tabanı, komutların birbirleri ile olan ilişkilerini tanımlamak amacıyla "hasParameter" ilişkisini kullanmaktadır. Bu ilişki ile, belirli bir komuta ait parametrenin başka bir CDO komutu ile bağlantısı olup olmadığı daha sonra sorgulanmak üzere tanımlanabilir. CDOOperations Kepler aktörü, bu bilgiyi kullanarak ilgili parametreye ait komutu otomatik olarak aktör parametresi olarak tanımlar ve komut parametresini bu komutun çıktısı ile ilişkilendirir. Bu aşamada, ilgili ilişkinin yaratılması için "isCalculatedBy" ilişkisi kullanılır. Seçilen CDO komutunun işleyişini etkileyen çevre değişkenleri ve bu değişkenlerin alabileceği değerler, "hasEnvironment", "hasOptions" ve "hasDefaultValue" ilişkileri kullanılarak, ilgili komuta bağlanır. "hasInput" ve "hasOutput" ilişkileri ise, ilgili

komutun parametre olarak girdi ve çıktı dosyası alıp almadığı bilgisini saklamak için kullanılmaktadır.

Geliştirilen bilgi tabanının sorgulanması daha önceki bölümlerde bahsedildiği gibi SPARQL sorgulama dili kullanılmaktadır. SPARQL ile yapılan sorgulamalara örnek olması için ekler bölümünde bulunan Kod C.1'deki sorgu örnekleri incelenebilir.

- **Yardımcı Aktörler:** Bu aktörler, diğer başlıklar altında tanımlanan aktörlerin bir arada kullanılması amacıyla geliştirilmiş aktörlerdir. Bunlar arasında "ConstantToArray" ve "ArrayColumn" sayılabilir.

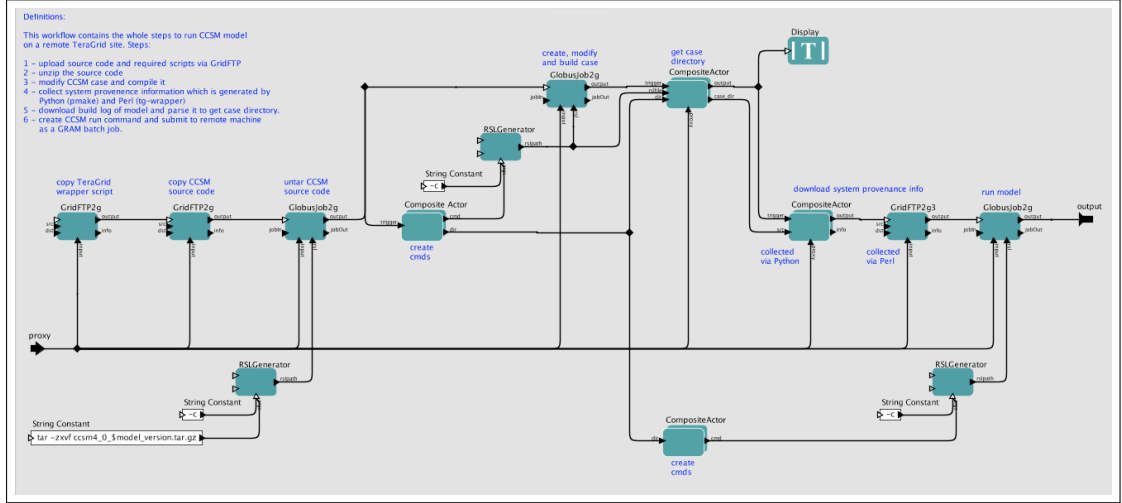
Şekil 4.5'da görülen mantıksal iş akış diyagramının birinci bölümünün Kepler uygulamasına dönüştürülmüş sürümü, Şekil 4.8'de görülebilir.



**Şekil 4.8:** CCSM modeli için geliştirilmiş Kepler üst seviye iş akışı

İş akış tasarımında, ana iş akışını kontrol eden direktör (SDF, Synchronous Dataflow) iki farklı CCSM simülasyonu yaratma aşamasında PN (Process Network) direktörü ile değiştirilerek her bir CCSM simülasyonunun yaratılması için farklı Java iş parçacığının oluşturulması ve farklı çözünürlükteki CCSM benzetimlerinin paralel bir şekilde üretilmesi sağlanmıştır. Bu durumda, birleşik aktörlerin içerisine ilgili iş akışını

kontrol etmesi için SDF direktörleri yerleştirmek gereklidir. Aksi takdirde, birleşik aktörün içindeki iş akışı, varsayılan olarak PN direktörü ile çalıştırılacaktır. Ana iş akışında görülen birleşik aktör kopyalanarak, yeni bir CCSM simülasyonunun oluşturulması ve/veya iş akışında bulunan parametrelerin değiştirilmesi ile geliştirilen iş akışının başka bir TeraGrid kaynağında çalışması kolayca sağlanabilir. CCSM modeli için geliştirilmiş bu Kepler iş akışı, toplamda 88 aktör kullanmaktadır. Ana iş akışında görülen birleşik aktörün iç yapısı, Şekil 4.9’de detaylı bir şekilde görülebilir.

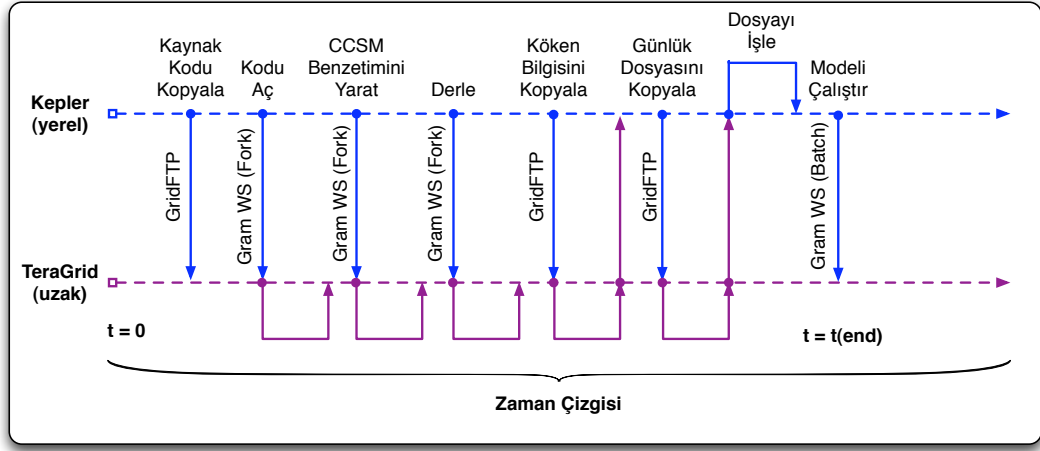


Şekil 4.9: Şekil 4.8’te görülen birleşik aktörün iç yapısı

İş akışını ve yapılan işlemleri bir zaman çizelgesi üzerinde göstermek istersek, iş akışının izlediği adımlar daha kolay anlaşılabilir. Şekil 4.10’de görüldüğü gibi iş akışı öncelikle CCSM benzetiminin hazırlanması için gerekli dosyaları (kaynak kod, betikler vb.) GridFTP protokolü kullanarak, uzaktaki TeraGrid kaynağına kopyalamaktadır. Bu aşamadan sonra CCSM modeline özel Kepler aktörleri kullanılarak, kullanıcı tarafından belirlenen CCSM konfigürasyonu, uzaktaki hesaplama kaynağında yaratılır ve CCSM modeli derlenir. Derleme aşamasında Globus üzerinde interaktif bir iş çalıştırılır ve derleme sırasında Python ve Perl betikleri ile toplanan sistem köken bilgisi, iş akışının çalıştığı yerel bilgisayara kopyalanır. CCSM modelinin derleme aşamasında ürettiği günlük dosyası kullanılarak belirlenen kurulum lokasyonu, CCSM modelinin çalıştırılması için kullanılır. Bu aşamada, modelin çalıştırılması için uzak TeraGrid kaynağında kullanılan ve işleri belirli bir sırada çalıştırmak için özelleşmiş uygulamaya özgü komutlar ile model sisteme iş olarak verilir. Uzak sistem ile ilgili

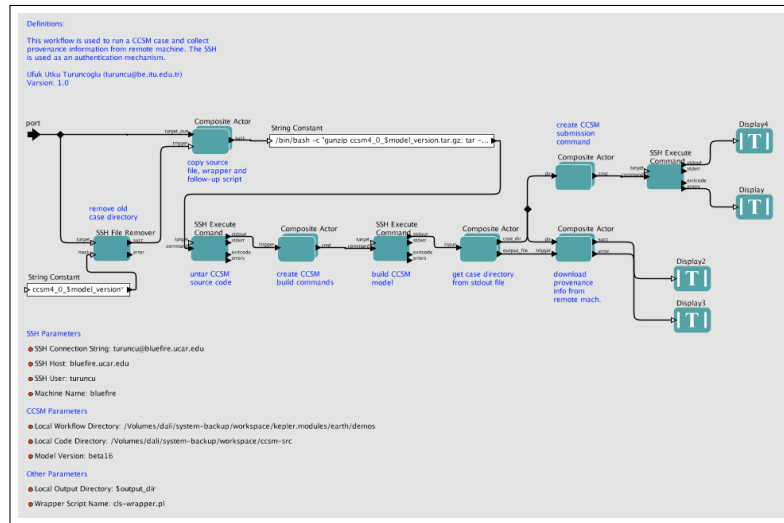


bilgilerin bir kısmını, CCSM içerisinde bulunan XML konfigürasyon dosyalarından alınan bilgiler oluşturur.



Şekil 4.10: CCSM iş akışının zaman çizelgesi üzerine aktarılmış hali

TeraGrid hesaplama kaynağı üzerinde çalışan iş akışını (Bkz. Şekil 4.9) NCAR'a ait Bluefire küme makinası üzerinde çalıştırmak için kolayca değiştirebiliriz. Yapılması gereken, TeraGrid veya hesaplama gridine özel Kepler aktörlerini temel SSH aktörleri ile değiştirmektir. Bluefire küme sistemi, daha önceki bölümlerde okuyucuya aktarıldığı gibi tek kullanımlık şifre (OTP) kullanmaktadır ve SSH aktörleri, bu yetkilendirme sistemini desteklemektedir. CCSM iş akışının Bluefire küme sistemi göz önünde bulundurulurak değiştirilmiş sürümü, Şekil 4.11'de görülebilir.



Şekil 4.11: TeraGrid CCSM alt iş akışının Bluefire küme sistemine aktarılmış hali

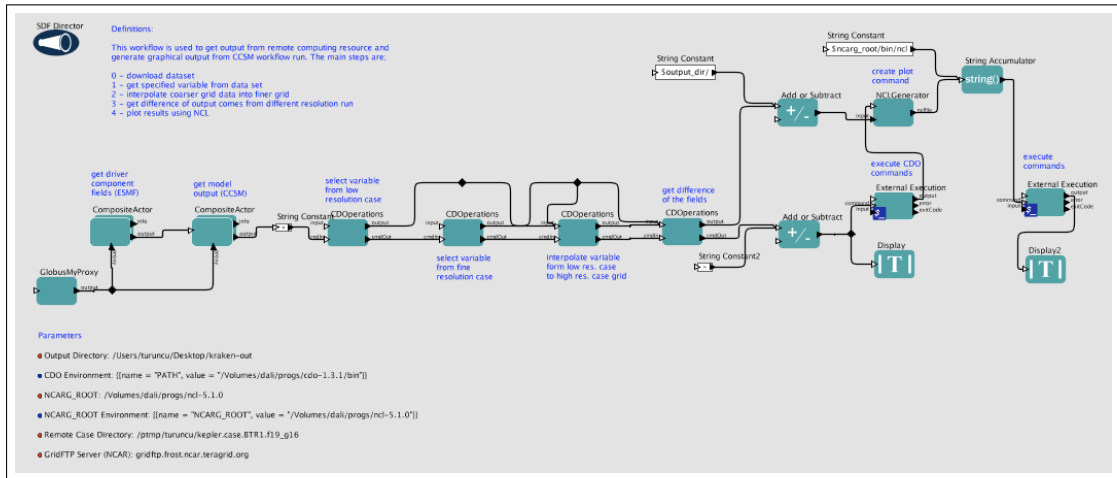
Temel CCSM aktörleri değişmeden, sadece hesaplama ortamına özgü aktörler değiştirilerek, aynı iş akışı farklı bir hesaplama ortamında kolayca çalıştırılabilmektedir. Geliştirilen metodolojinin ve üretilen iş akışı sisteminin tasarımının doğru yapıldığının bir kanıtı olarak görülebilir. Bir başka deyişle; üretilen iş akışı, farklı hesaplama kaynaklarına kolayca uyabilen bir şekilde tasarlanmıştır. Gönderilen e-postanın içeriği son derece basittir (Şekil 4.12).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <workflow>
3   <name>07-follow-up.xml</name>
4   <directory>/Volumes/dali/system-backup/workspace/kepler.modules/earth/demos</directory>
5   <parameters>
6     <gridftp_srv>gridftp.frost.ncar.teragrid.org</gridftp_srv>
7     <output_dir>/Users/turuncu/Desktop/output-wf-beta16</output_dir>
8     <remote_case_dir>/ptmp/turuncu/kepler.case.BTR1.f19_g16</remote_case_dir>
9   </parameters>
10 </workflow>

```

**Şekil 4.12:** CCSM iş akışı tarafından üretilen ve e-posta olarak gönderilen XML dosyası



**Şekil 4.13:** CCSM iş akışı tarafından otomatik olarak tetiklenen iş akışı

Yukarıda, içeriği görülebilecek elektronik posta; Kepler uygulaması tarafından tetiklenecek iş akışının ismini, hangi dizinde bulunduğunu ve çalıştırılacak iş akışına ait değiştirilecek parametrelerin bir listesini içerir. Bu amaçla, Kepler uygulamasının yapısında değişikliğe gidilmiş ve uygulama açılırken, kullanıcının posta hesabını belirli aralıklar ile dinleyen ve ayrı bir iş parçacığı olarak çalışan yeni bir servisin başlatılması sağlanmıştır. Bu şekilde Kepler, elektronik posta ile tanımlanan ilgili iş akışını bularak, üzerindeki değişiklikleri yapar ve çalıştırır. Şekil 4.13'deki

örnekte, iş akışı daha önce çalıştırılmış CCSM benzetimine ait çıktıları yerel sunucu veya makinaya kopyalayarak, sonuçlar üzerinde CDO komutları ile analiz işlemi gerçekleştirmektedir. Son ürün olarak ise, farklı çözünürlüklerde çalıştırılmış model çıktıları arasındaki farkın NCL uygulaması ile görselleştirilmiş hali kullanıcıya sunulmaktadır.

#### **4.6 Uygulama Örneği Sonuçları**

Bu bölümde CCSM modeli için geliştirilen mantıksal iş akışı, Bölüm 2.4'te önerilen metodolojiyi kullanarak, Kepler iş akışı ortamına aktarılmıştır. Sonuçlar, geliştirilen metodolojinin, kullanılan yer sistemi modeli ile hesaplama ortamını anlamlı bir seviyede soyutlaştırılabildiğini kanıtlamaktadır. Ayrıca bu örnekte, model ile ilgili köken bilgisi kullanıcıdan bağımsız ve standart bir şekilde toplanabilmiştir. Köken bilgisinin, yersistem bilimlerinde yapılan çalışmalar için ne denli önemli olduğu açıktır. Bu sebeple, köken bilgisinin geliştirilen iş akışı tarafından otomatik bir şekilde toplanabilmesi, oldukça önemli bir adımdır. Tüm bunlara ek olarak sına uygulama, yersistem modelleme çalışmaları için son derece önemli olan birbirine bağımlı veya birbirini izleyen iş akışlarının çalıştırılması için yeni bir yöntem içermekte ve bunu başarı ile uygulamaktadır.

Bir sonraki bölüm, kuple edilmiş yer sistem modelleri (atmosfer ve okyanus) kullanılarak geliştirilen iş akış ortamının sınanmasını amaçlamaktadır. Bu şekilde, önerülen metodolojinin işlevselliğinin farklı bir sına uygulama ile incelenmesi amaçlanmaktadır.

## 5. UYGULAMA ÖRNEĞİ - II

### 5.1 Kuple Edilmiş Model Sistemi: ROMS ve WRF

Bu bölüm, bir önceki bölümde (Bölüm 4.1) CCSM4 küresel sirkülasyon modeli ile geliştirilen iş akışına ek olarak, önerilen metodolojinin sınanması için diğer bir gerçekçi uygulama senaryosunun iş akış ortamına aktarılmasını amaçlamaktadır. Bu şekilde yer sistem bilimciler tarafından yapılan gerçek çalışmalara benzer iki farklı test uygulaması, önerilen metodolojinin ve geliştirilen iş akış sisteminin sınanması için kullanılmış olacaktır.

Geliştirilen ikinci uygulama, yer sistem bilimlerinde yaygın olarak kullanılan küresel dolaşım modellerinin düşük çözünürlükte (IPCC AR4 raporunda atmosfer modeli çözünürlükleri  $\sim 1.5^\circ \times 1.5^\circ$ ) olan çıktılarını, bölgesel ölçeğe indirgemeyi amaçlayan dinamik ölçek küçültme yaklaşımını temel almaktadır.

Genellikle günümüzde sıkça kullanılan yaklaşım, tekil olarak belirli bir bölge üzerindeki atmosferi, iklimsel anlamda uzun bir süre boyunca farklı senaryolar ve model çıktıları kullanarak modellemektir. Ancak son yıllarda yapılan çalışmalar [47, 48], tıpkı küresel dolaşım modellerinin farklı bileşenler (okyanus, kara, buz vb.) içermesi gibi, bölgesel iklim modellerinin de farklı bileşenler ile birlikte kullanılmasına yönelmiştir. Bu tip çalışmalar için verilebilecek en önemli örnek, bir atmosfer modeli ile bir okyanus modelinin kuple bir şekilde çalıştırılmasına dayanmaktadır. Bu şekilde, tekil olarak atmosferin modellenmesi yerine, ilgilenilen bölgeyi içine alan atmosfer ve okyanus kütleleri birlikte modellenerek, daha gerçekçi sonuçlar elde edilmeye çalışılmaktadır.

Yukarıda, temel olarak açıklanmaya çalışılan dinamik ölçek küçültme yaklaşımının, kuple edilmiş bir atmosfer ve okyanus model sistemi kullanılarak iş akış ortamına aktarılması ve geliştirilen çalışma ortamının sınanması, bu test uygulamasının temel amacıdır. Bu amaçla geliştirilen test uygulaması, Akdeniz bölgesindeki iklim

değişimini modellemek amacı ile kuple edilmiş bir model sisteminin çalıştırılmasını amaçlamaktadır. Ayrıca geliştirilen test uygulaması, Türkiye ve bölgesi için yapılacak iklim çalışmaları, göller ve sığ sular gibi bölgesel su kütlelerinin hidrodinamik özelliklerinin araştırılması gibi birçok farklı amaç için de kullanılabilir.

Aşağıdaki bölümlerde öncelikle, kullanılan yer sistem modelleri ve bu modellerin kuple edilmesi için izlenen yöntem ile ilgili okuyucuya genel bir bilgi verilecektir. Daha sonraki bölümlerde ise, geliştirilen kuple model sisteminin iş akışı çalışma ortamına aktarılması ve bununla ilgili sonuçlar irdelenecektir.

## **5.2 WRF Sınırlı Alan Atmosfer Modeli**

WRF modeli, NCAR tarafından hem tahmin, hem de atmosfer araştırmaları için geliştirilmiş, hidrostatik olmayan bir sınırlı alan sayısal atmosfer modelidir [49]. Temel olarak, bir önceki model olan MM5 modelinin yapısı üzerine kurulmuştur. Model, içerdiği farklı atmosferik dinamik çekirdekler ile değişik araştırma alanları için özelleştirilebilir ve bu dinamik çekirdekler için tasarlanmış fiziksel alt modeller değiştirilerek, farklı çözünürlükteki atmosferik olaylar modellenebilir. Alt modeller; belirli bir yer sistem bileşeni ile atmosfer modelinin etkileşimini ve/veya belirli bir atmosferik olayı modellemek için geliştirilmiş daha küçük model parçacıklarıdır. WRF modeli, bulut fiziği, sınır tabaka, kara yüzeyi ve radyasyon alt modelleri gibi birçok farklı alt model tipi içermektedir.

Bu çalışmanın amacı, farklı alt modellerin sonuçlara etkisini araştırmak veya hassasiyet analizi yapmak olmadığından, seçilen fiziksel parametrelerin birçoğu varsayılan değerlerinde bırakılmıştır ve model ESMF-IO bileşenini kullanacak şekilde ayarlanmıştır. Farklı yer sistem modelleri için gerekli olan değişkenlerin atmosfer modelinden ESMF kütüphanesi kullanılarak diğer bileşenlere gönderilmesi için, WRF, ESMF-IO yapısı kullanılır. Bu şekilde, alan değişkenleri ESMF durum nesnelere şeklinde diğer model bileşenlerine gönderilmiş olur. WRF modelinin ihtiyaç duyduğu değişkenler ise, yine bu yapı sayesinde tanımlanarak, WRF modeli tarafından kullanılabilir. Başka bir deyişle ESMF-IO, modelin diğer yer sistem modelleri ile veri alışverişi yapmasını sağlayan temel yapılarından biridir. ESMF-IO yapısı ile ilgili detaylı bilgi, kuple model sisteminin yapısının anlatıldığı Bölüm 5.7'de bulunabilir.

### 5.3 ROMS Okyanus Modeli

Bu sınaama uygulamasında, kuple model sisteminin okyanus bileşeni olarak ROMS modeli kullanılmıştır. ROMS; üç boyutlu, düşeyde sigma koordinatlarını kullanan, hidrostatik ve Navier Stokes denklem sistemini Arakawa C gridi ve curvilinear koordinatlar üzerinde Boussinesq kabulü (denklemlerin basitleştirilmesi amacıyla akışkan içindeki yoğunluk değişiklikleri küçük ve ihmal edilebilir olması) ile çözen bir okyanus modelidir. Ayrıca model, momentum ve iz değişkenlerinin (sıcaklık, tuzluluk) yatay ve düşey adveksiyon şeması olarak ikinci, üçüncü ve dördüncü dereceden çözümleri desteklemektedir.

ROMS modelinin, diğer yer sistem modelleri ile kuple edilebilmesi amacıyla, MCT kütüphanesi ile geliştirilmiş bir kuple bileşeni bulunmaktadır. Bu bileşen ile model, teorik olarak WRF, SWAN gibi farklı modeller ile kuple edilebilmektedir. Ancak, geliştirilen metodoloji kullanılarak - kuple sistemin iş akış ortamına entegre edilmesine duyulan ihtiyaç nedeniyle - kuple bileşeninin ESMF kütüphanesi kullanılarak tekrar yazılması gerekmiştir. Bu amaçla, model içinde bulunan ve MCT uygulaması ile yazılmış kod parçalarından yararlanılarak, ESMF kütüphanesini kullanan yeni bir kuple bileşeni yazılmıştır (Bkz. Bölüm 5.7).

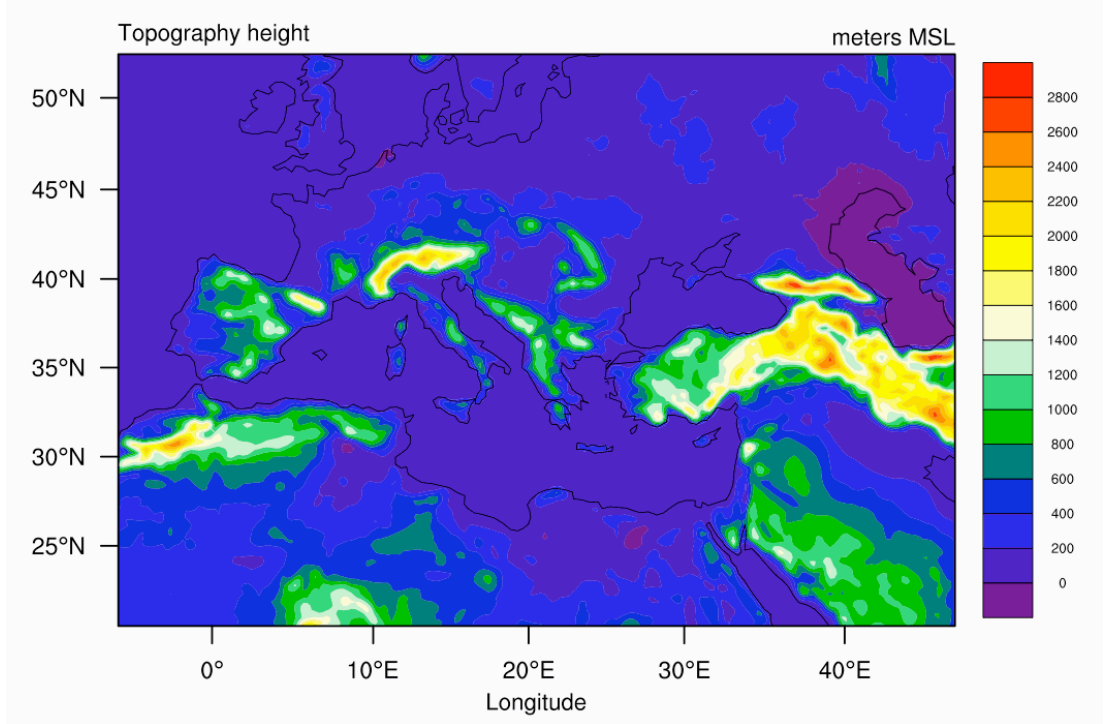
### 5.4 Test Uygulaması

Bu uygulama için seçilen atmosferik model WRF, okyanus modeli ise ROMS'tur. Çerçeve ve iş akışı yaklaşımlarının birleştirilmesi için WRF ve ROMS modellerinin ESMF kütüphanesi ile birleştirilmesi planlanmıştır. Bu test uygulamasında amaç, geliştirilmiş olan kuple modelleme sisteminin detaylı bir şekilde incelenmesi olmayıp, bu sistemin geliştirilerek, iş akışı çalışma ortamına aktarılmasıdır. Bu nedenle, modelleme sistemi ile uzun süreli iklimsel simülasyonlar yapılması yerine sadece sistemin düzgün bir şekilde çalıştığını gösteren kısa bir benzetim tasarlanmıştır. Ancak, istenildiği takdirde aynı sistem ile geleceğe yönelik senaryoların üretilmesini amaçlayan, uzun süreli iklimsel simülasyonlar da yapılabilir.

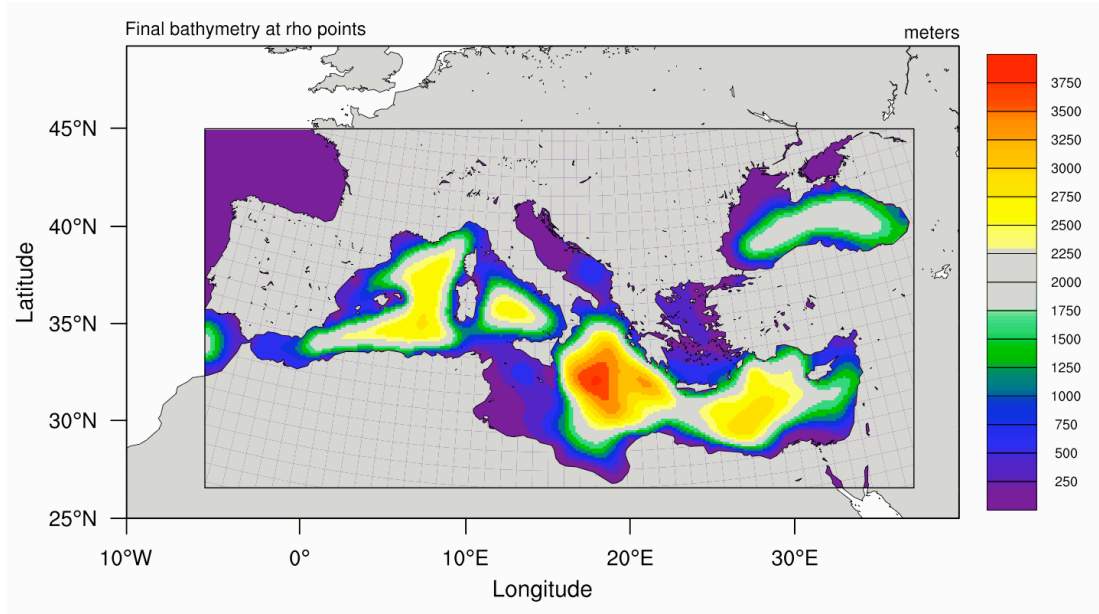
Hem WRF modeli, hem de ROMS modeli CCSM küresel dolaşım modeline ait çıktı verileri kullanılarak çalıştırılacak şekilde ayarlanmıştır. Bu amaçla, WRF ve ROMS modelleri için Bölüm 5.5'te daha detaylı olarak değinilecek iki farklı uygulama

geliştirilmiştir. Bu uygulamaların amacı, CCSM modeli çıktılarını işleyerek, WRF ve ROMS modelleri için uygun formatta veri hazırlamaktır. Çalışmada kullanılan girdi verisi, CCSM modelinin IPCC AR4 raporunda kullanılmış SRES 20. yüzyıl (20C) sonuçlarıdır. Kullanılan simülasyonun referans adı ise b30.030e'dir. WRF modeli için CCSM'e ait CAM ve CLM alt modellerinin çıktıları kullanılarak, başlangıç ve sınır koşullarını içeren girdi dosyaları üretilirken, CAM modeli çıktısı ROMS modeli için yüzey şartları üretmek, POP alt modellerine ait veriler ise yine ROMS modeli için okyanus kütleleri için sınır koşullarının üretilmesi için kullanılmıştır.

Simülasyon süresi olarak 1999-07-01\_00:00 ile 1999-07-31\_18:00 tarihleri arasındaki bir aylık periyot seçilmiştir ve atmosfer modelinin zaman adımları 150 sn iken, okyanus modeli için bu süre 300 sn'dir. Modeller arasındaki veri alışverişleri ise 300 sn lik aralıklar ile yapılmaktadır. Atmosfer modeli, yatayda 210x144 grid noktası, düşeyde ise 27 sigma seviyesi içermektedir ve ortalama çözünürlüğü 27 km'dir. Okyanus modeli ise, 11 km çözünürlüğe sahip ve yatayda 319x159 grid noktası içerirken, düşeyde 32 sigma seviyesine sahiptir. WRF ve ROMS modellerinin kapsadıkları alanlar Şekil 5.1 ve 5.2 'de görülebilir.



Şekil 5.1: WRF modeli alanı ve yükselti verisi



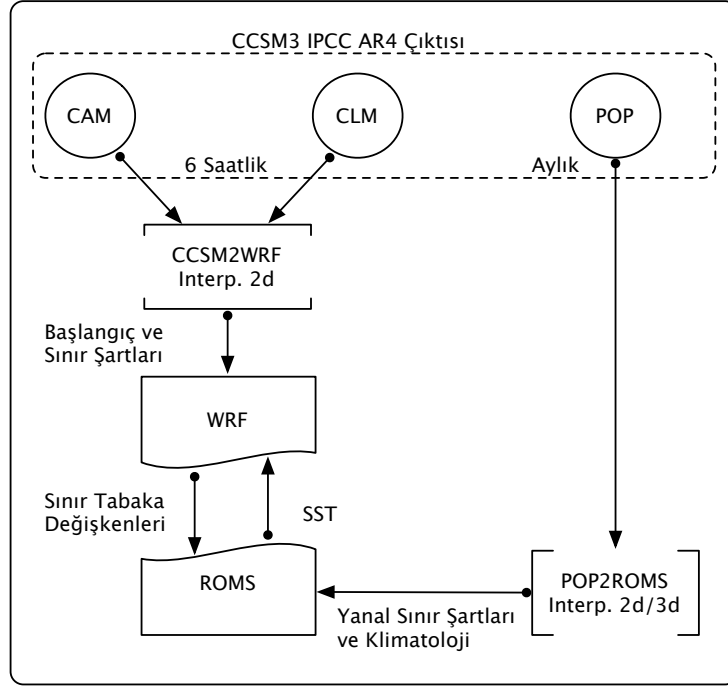
**Şekil 5.2:** ROMS modeli alanı ve derinlik verisi

Şekil 5.3, test uygulamasının genel yapısını ve içerdiği bileşenleri göstermektedir. Şekilden görülebileceği gibi, simülasyona başlamadan önce her iki model için de girdi verileri, geliştirilen araçlar yardımı ile oluşturulur. CCSM modelinin okyanus modeli olan POP modelinin çıktıları aylıktır ve klimatolojik olarak ROMS modelinin çalışacağı bölgedeki su kütesinin üç boyutlu durumunu içermektedir. WRF modeli için ise, girdi verileri 6 saatliktir. Veriler, her bir model için hazırlandıktan ve modeller çalıştırıldıktan sonra, atmosfer modeli okyanus modeline ısı ve momentum akılarının hesaplanması için gerekli değişkenleri gönderirken, okyanus modeli de atmosfer modeline deniz yüzeyi sıcaklığını göndermektedir. Sistemin işleyişi ile ilgili daha detaylı bilgi Bölüm 5.7’de bulunabilir.

### 5.5 Geliştirilen Araçlar

Bir önceki bölümde tanımlanan model sistemini (Bkz. Şekil 5.3) oluşturan alt modellere, girdi verisi (başlangıç ve sınır koşulları) üretmek amacıyla, CCSM küresel dolaşım modelinin bileşenleri olan CAM, CLM ve POP modellerinden alınan, NetCDF formatındaki çıktılar işlenerek, WRF ve ROMS modellerinin yatay ve düşey grid sistemine aktaracak iki adet uygulama geliştirilmiştir. Bu uygulamalar, WRF için CCSM2WRF ve ROMS modeli için CCSM2ROMS olarak adlandırılmıştır.





**Şekil 5.3:** Test uygulamasının genel yapısı

WRF modeli için geliştirilen araç, temel olarak Harvard Üniversitesi araştırmacıları tarafından geliştirilmiş olup, CAM2WRF uygulamasını temel almaktadır. Ancak, program tekrar yazılmış ve hataları ayıklanarak, CLM'den alınan toprak sıcaklığı ile ilgili kısmı düzeltilmiştir. Program, CAM verilerini hybrid sigma seviyesinden WRF modelinin düşey seviyelerine doğrusal interpolasyon yöntemi kullanarak, yatayda ise doğrusal olmayan kübik interpolasyon metodu kullanarak aktarmaktadır. Uygulama, NetCDF formatında aldığı üç boyutlu verileri (sıcaklık, karışım oranı, rüzgar bileşenleri, geopotansiyel yükseklik) işleyerek, WRF modeline girdi verisi üretmek için kullanılan WPS uygulaması ile uyumlu bir formatta diske yazmaktadır. WPS için gerekli veriler üretildikten sonra, WPS uygulamasının "metgrid" aşaması ile veri WRF modelinin okuyabileceği formattaki NetCDF dosyasına dönüştürülür. Temel olarak CCSM2WRF uygulaması, WPS modelinin "ungrib" aşamasında yapılan işlemleri yapmaktadır ve bu nedenle, bu aşamanın çalıştırılmasına gerek yoktur.

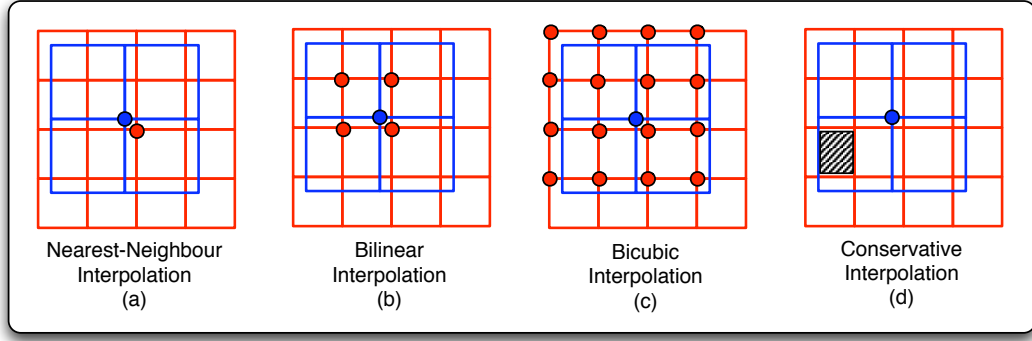
ROMS modeli için geliştirilen uygulama, temel olarak WRF modeli için geliştirilen uygulamaya benzemektedir. CCSM2ROMS, CCSM modelinin POP bileşeninden aldığı verileri kullanarak, başlangıç ve ROMS okyanus modelinin atmosfer ile ara kesiti olan sınır tabaka için gerekli değişkenlerin (yüzey sıcaklığı, yüzey basıncı, yüzey

nemi, rüzgar bileşenleri, kısa dalga boylu radyasyon, uzun dalga boylu radyasyon vb.) bulunduğu veri dosyalarını üretir. Modelin ihtiyaç duyduğu yanal sınırlar için gerekli olan girdi dosyası ise, bir MATLAB betiği kullanılarak CCSM2ROMS uygulaması ile yaratılan dosyalardan üretilmektedir. ROMS için hazırlanan tüm bu veriler NetCDF formatındadır. Uygulama, yatay interpolasyon yapmak amacıyla SCRIP programı tarafından üretilmiş bir dosyayı kullanır. Bu dosyada, küresel okyanus modeli (POP) ile bölgesel okyanus modeli (ROMS) gridleri için üretilmiş bir ağırlık matrisi bulunmaktadır. Oluşturulan matris, seyrek bir matris olduğu için bu dosya içerisinde ağırlık değerlerinin sıfırdan farklı olduğu indislerde tutulmaktadır. Bu şekilde, basit bir seyrek matris çarpımı ile veriler, POP gridinden ROMS gridine kolayca ve yüksek performanslı bir şekilde aktarılmaktadır. Örneğin, 320x384 (122880) grid noktası bulunan POP modeli gridinden, 317x157 (49769) grid noktası bulunan ROMS modeli gridine, seyrek matris çarpımı yöntemi kullanılmadan interpolasyon yapmak, performanslı bir şekilde ve hesaplama yapılan bilgisayarın hafıza sınırları içerisinde gerçekleştirilemez. Üretilen ağırlık matrisi, her bir değişken ve zaman adımı için kullanılarak, ROMS modeli için gerekli veri oluşturulmaktadır. Uygulama ayrıca düşey boyutta (z ve sigma seviyeleri) interpolasyon yapmak amacıyla doğrusal interpolasyon yöntemini kullanmaktadır.

## 5.6 SCRIP Uygulaması ve Yapılan Değişiklikler

SCRIP [41, 42], değişkenlerin farklı geometriye sahip gridler arasında aktarılmasını sağlamak amacıyla, interpolasyon ağırlıklarını hesaplayan bir uygulamadır. Uygulama, temel olarak dört farklı interpolasyon tipini desteklemektedir (Şekil 5.4): Bunlardan birincisi, korunumlu interpolasyon şemasıdır. Bu şema, kuple edilmiş modelleme sistemleri için geliştirilmiş ve korunması gerekli alansal değişkenlerin (nem veya ısı akısı) farklı gridler arasında aktarılmasını sağlayan bir şemadır. Bu yöntemde ilgili değişken, kaynak ve hedef grid alanı ile orantılı bir şekilde hedef gride aktarılır ve sistem içinde interpolasyon yolu ile yapay bir şekilde enerji üretilmesinin önüne geçilmiş olur.

Bu çalışmada ısı akısının WRF modelinden ROMS modeline aktarılması için korunumlu bir interpolasyon yöntemi kullanılması amaçlanmış ancak SCRIP uygulamasında bulunan eksiklikler nedeniyle bu yapılamamıştır. Çalışmanın ileriki

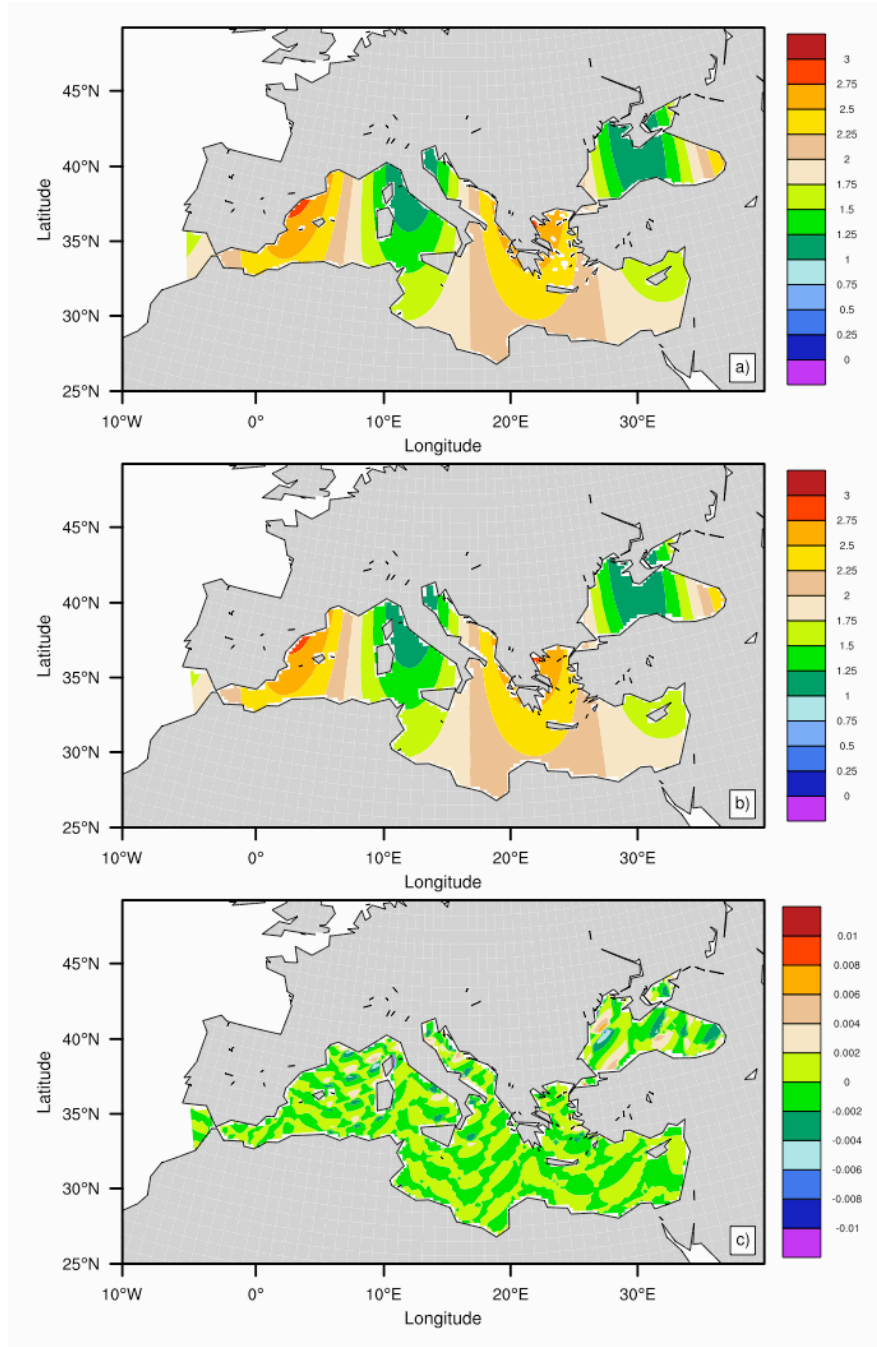


**Şekil 5.4:** SCRIP uygulaması tarafından desteklenen interpolasyon tipleri

aşamalarında, kuple model sistemini oluşturan model bileşenleri arasında ısı akılarının (gizli ısı, hissedilir ısı, uzun dalga boylu radyasyon vb.) interpolasyonu amacıyla korunumlu ağırlık matrislerinin üretilmesi veya ESMF kütüphanesinde bulunan ve interpolasyon ağırlıklarına ihtiyaç duymayan metodlar ve alt programların kullanılması planlanmaktadır. İkinci interpolasyon tipi ise, yerel doğrusal interpolasyon yönteminin geliştirilmiş bir şekli olan doğrusal interpolasyon yöntemidir. Üçüncü metod, doğrusal interpolasyon yöntemine benzeyen kübik interpolasyon yöntemidir. SCRIP uygulaması tarafından desteklenen son interpolasyon yöntemi ise komşu noktaları baz alan ve bu noktaları birbirlerine göre uzaklıkları ile ağırlıklandıran bir yöntemdir.

ROMS ve WRF modelleri arasında değişkenlerin interpolasyonu için kullanılan yöntem, yukarıda sayılan yöntemlerin en sonuncusudur. Şekil 5.5’de yapay olarak üretilmiş bir alan değişkenin ROMS gridinden WRF gridine komşu noktalar yöntemi ile aktarılması ve oluşan hata grafiği görülebilir. Şekilden görülebileceği gibi maksimum hata değeri  $\pm \% 0.1$  mertebesindedir.

Yapılan testlerde her iki yönde de (WRF’tan ROMS’a ve ROMS’tan WRF’ye) bu interpolasyon yöntemi daha iyi sonuç vermiştir. Modellerde bulunan maskeleme değişkenlerinin daha doğru bir şekilde interpolasyon aşamasında kullanılması amacıyla, SCRIP uygulaması üzerinde ufak değişiklikler yapılmıştır. SCRIP, bu maskeleme değişkenlerini kullanarak, ağırlık matrisinin boyutunu küçültmektedir. Bu nedenle, modellere ait maskeleme ve grid alanı değişkenlerinin son derece hassas bir şekilde temsil edilmesi gereklidir. SCRIP uygulamasının çalıştırılabilmesi için, modelin çalıştığı grid yapısının belirli bir formda ve NetCDF formatında uygulamaya verilmesi



**Şekil 5.5:** SCRIP uygulaması test sonuçları: (a) ROMS gridindeki test verisi, (b) WRF gridine interpolasyon ile aktarılmış veri, (c) hata. Test fonksiyonu  $f = 2 + \sin^{16}(2\theta) \cos(16\phi)$  küresel harmonik,  $l = 32, m = 16$

gereklidir. Bu dosya içerisinde, her bir grid noktasının (grid merkezi veya ilgilenilen diğer nokta; u, v vb.) koordinatları, merkez alınan grid noktasını çevreleyen noktaların (köşe noktaları) koordinatları, maskeleme değişkeni ve grid alanı değişkeni tek boyutlu bir dizi şeklinde bulunmaktadır. Tüm bu değişkenlerin WRF ve ROMS modelleri için üretilmesi amacıyla hem WRF, hem de ROMS modeli gridlerini okuyarak SCRIP

uygulamasını formatına dönüştürecek iki alt uygulama, SCRIP içerisinde bu çalışma kapsamında entegre edilmiştir.

## 5.7 Modellerin Kuple Edilmesi

Bölüm 5.1 ve 5.4'te detaylı olarak tanımlanan test uygulamasının geliştirilmesi amacıyla WRF ve ROMS modelleri, ESMF kütüphanesi kullanılarak birleştirilmiştir. Bu bölüm, bu iki farklı modelin birleştirilmesi sırasında izlenen metodolojiyi ve kuple edilmiş modelleme sisteminin iç yapısını tanıtmayı amaçlamaktadır.

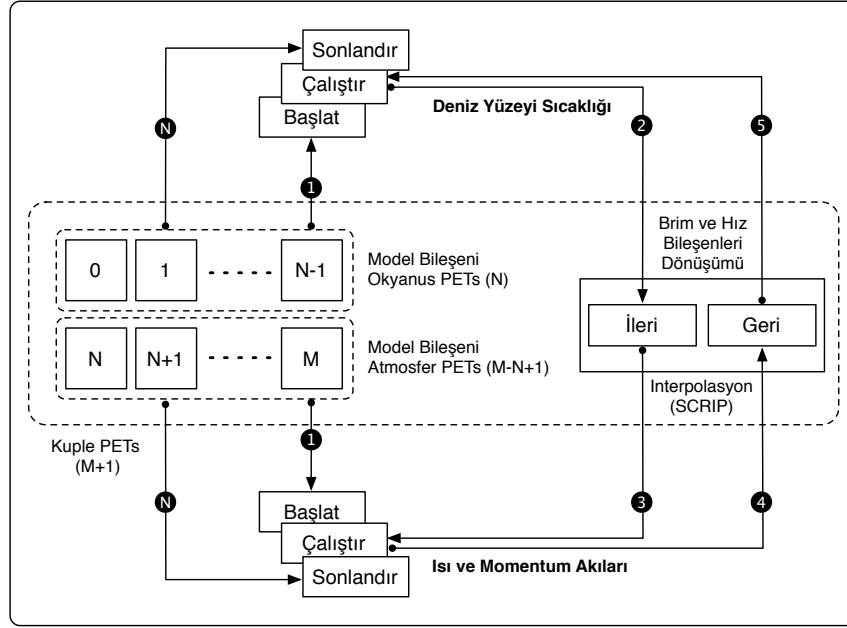
ESMF, farklı yer sistem modellerinin birleştirilmesi ve standart bir yapıya kavuşturulması için geliştirilmiş bir kütüphanedir (Bkz. Bölüm 3.2). Hem WRF hem de ROMS modeli, ESMF kütüphanesini kullanan çeşitli kod yapılarına ve parçalarına sahiptir. Ancak bu parçalar, modellerin bir arada çalıştırılması için yeterli değildir. Aşağıda, bu modeller içerisinde bulunan ESMF yapıları ve bunlara işlerlik kazandırmak amacıyla yapılan değişiklikler hakkında bilgi verilecektir.

WRF modeli içerisinde bulunan ve deneysel yapıdaki ESMF-IO bileşeni, modelin ESMF ile olan etkileşiminin sağlanması amacıyla yazılmıştır. Bu bileşen, standart bir IO (girdi/çıkıtı) yapısı gibi davranmakla birlikte, veriyi diske yazmak yerine, veriyi oluşturan değişkenleri ESMF veri alanları (ESMF\_Field) şekline dönüştürüp, bunlardan ESMF durum (ESMF\_State) yapıları oluşturur ve bunları dağıtık olarak saklar. ESMF durum tipleri, ESMF içerisinde verinin bir noktadan diğer noktaya taşınması için tasarlanmış özel yapılardır ve iki tipte olabilir. Bunlardan birincisi; ilgilenilen model bileşeninin ihtiyaç duyduğu değişkenleri saklayan ESMF veri alma (ESMF\_IMPORT) yapıları, diğeri ise; bu modelin diğer yer sistem modellerine göndermek için hazırladığı ESMF veri gönderme (ESMF\_EXPORT) yapılarıdır. Örneğin, WRF atmosfer modeli için okyanus modelinden gelen deniz suyu sıcaklığı (SST), ESMF veri alma yapısı ile tanımlanırken aynı değişken ROMS modelinde veri gönderme yapısı olarak adlandırılır, WRF atmosfer modelinden okyanus modeline gönderilmesi gereken rüzgar hızı bileşenleri, yüzey basıncı vb. değişkenler, ESMF veri gönderme yapıları olarak tanımlanır. ESMF veri alma ve gönderme yapılarının hangi aralıklar ile hazırlanacağı ve bunların ne zaman kullanılacağı, kuple edilmiş model sistemi için veri alışverişi sıklığını belirlemektedir.

WRF modelinin aksine ROMS modeli, ESMF veri alanlarını ESMF durum yapıları içerisinde saklamak yerine, ESMF dizilerini (ESMF\_Array) bu yapılar içerisinde saklayarak, farklı model bileşenleri ile paylaşmayı amaçlamaktadır. Ancak bu yapı, WRF modeli ile uyumlu olması açısından model sisteminin daha sonra tasarlanacak sürümlerinde, ESMF alan değişkeni yapıları ile tanımlanacaktır. Bu şekilde, ESMF içinde bulunan alan değişkenlerine özel metotların kullanılması planlanmaktadır. Her iki modelin kuple edilmesi için, ROMS modeli içerisinde bir ESMF ana programı, bir de ROMS bileşenini tanımlayan ESMF bileşen kodu bulunmaktadır. Bu hali ile ROMS modelini başka bir model ile, ESMF kütüphanesini kullanarak birleştirmek mümkün değildir. Bu amaçla hem WRF, hem de ROMS modelinde bulunan eksiklikler ve hatalar giderilerek, bu iki modelin birleştirilmesi gerekmektedir.

Modelleme sisteminin oluşturulması için kullanılan temel programlama mimarisi, Şekil 5.6’te görülmektedir. Öncelikle, her bir yer sistem modelinin ESMF bileşeni haline getirilmesi için model kodunun başlangıç, çalışma ve sonlanma şeklinde üç farklı parçaya ayrılması gereklidir. Bu açıdan bakıldığında hem ROMS, hem de WRF modelleri bu gerekliliği sağlanmaktadır. Parçalardan her birinin sırasıyla ESMF tarafından tetiklenmesi ile, modellerin bir arada ve uyumlu bir şekilde çalışması sağlamaktadır. Tasarlanan yapı içerisinde her modelin kendi PET (işlemci kümesi) kümesi içerisinde çalışması amaçlanmıştır. Modeller arasındaki veri alışverişini ve model gridleri arasındaki dönüşümü yapan kuple bileşeni ise tüm PET’lerde çalışmaktadır. Ancak bu yapı, kolayca değiştirilebilir ve tüm PET veya işlemcilerin bir arada ve aynı model bölümünü çalıştırması sağlanabilir.

Kuple model sisteminin çalışması için öncelikle her iki modelin başlangıç aşamasının ESMF tarafından çalıştırılması gereklidir. Bu aşamada ESMF veri dizileri, alanlar ve bunları içerisinde barındıran ESMF durum yapıları hazırlanır. Bu aşama ayrıca modellere ait iç saatlerin sıfırlandığı ve kuple model sisteminin iç saatinin (ESMF\_Clock, ESMF\_StartTime, ESMF\_StopTime ve ESMF\_TimeStep) hesaplandığı veya ayarlandığı aşamadır. Kuple modelin iç saati, her bir model bileşenin iç saati incelenerek kuple bileşeni tarafından otomatik olarak yaratılır. Bu aşamadan sonra, üretilen iç saat ve zaman adımı göz önünde bulundurularak, her bir model sırasıyla belirli bir zaman aralığı için çalıştırılır. Bu aşamadan sonra kuple bileşeni devreye girerek, modeller arasındaki veri alışverişini gerçekleştirir. Bu



**Şekil 5.6:** Kuple modelleme sisteminin yapısı

işlemler sırasıyla daha önceden tanımlanan zaman aralığı süresince birbirini izleyerek devam eder. Kuple bileşeni, WRF modelinden ROMS modeline veri aktarımı ve ROMS modelinden WRF modeline veri aktarımı yapacak şekilde iki yönü olarak çalışır ve bu şekilde modellerin kendileri tarafından oluşturulan ESMF durum yapıları üzerinden veri alışverişi sağlanmış olur. Tüm bu aşamalardan sonra ise iki modelin ve bunlara bağlı MPI süreçlerinin sonlandırıldığı son aşamaya geçilir.

Yukarıda kısaca genel yapısından bahsedilen uygulamanın yaratılması amacıyla modeller üzerinde yapılan temel değişiklikler aşağıdaki listede görülebilir:

- ESMF tarafından, WRF modelinin başlangıç, çalışma ve sonlanma aşamalarını içeren WRF model bileşen kodu yazıldı.
- WRF ve ROMS modellerinin gridleri arasında veri değişimi yapabilmeleri amacıyla SCRIP ağırlık matrisini kullanarak interpolasyon yapan ve veri dönüşümlerini (rüzgar hızı alanlarının okyanus gridine interpolasyon yolu ile aktarıldıktan sonra döndürülmesi, birim dönüşümleri vb.) sağlayan kuple bileşeni yazıldı.
- ESMF ana programı içerisinde her bir model bileşeninin çalışması için gerekli değişiklikler yapıldı. Bunlar; model iç saatlerinin ayarlanması, saat bilgilerinin tüm PET'lere gönderilerek, kuple bileşeni için saatin üretilmesi olarak sayılabilir.

- ROMS modeli içerisinde MCT kütüphanesi için yazılmış kısımlar, ESMF kütüphanesinin kullanılabilceđi veya destekleneceđi şekilde deđiştirildi.
- WRF modeli kayıt (registry) dosyası içerisinde, ROMS modeline gönderilecek ve ROMS modelinden alınacak meteorolojik deđişkenler tanımlandı. WRF, bu dosya yardımı ile otomatik olarak ESMF durum yapılarını ESMF-IO aracılıđı ile oluřturmaktadır.
- Her iki kuple yönü için (WRF'den ROMS'a ve ROMS'tan WRF'ye) SCRIP ađırlık dosyaları, SCRIP uygulaması üzerinde deđişiklikler yapılarak oluřturuldu.

Yukarıda sıralanan deđişiklikler yapılırken, WRF modelinin **3.1.1** sürümü, ROMS modelinin 3.2 sürümü (revizyon **382**) ve ESMF kütüphanesinin yayınlanmış en son açık sürümü olan **3.1.0rp2** kullanılmıştır. ESMF modelinin daha güncel sürümlerinden birinin kullanılmamasının nedeni, WRF modelinin ESMF-IO bileşeninin ve ESMF zaman bileşeninin, ESMF kütüphanesinin bu sürümü ile uyumlu olması zorunluluđundan kaynaklanmaktadır. Ancak bu durum, çalışmanın daha sonraki safhalarında WRF modeli içinde bulunan ESMF-IO yapısının, ESMF kütüphanesinin en son sürümü (4.0.0r) ile derlenecek şekilde deđiştirilmesi ile düzeltilecek ve ESMF kütüphanesinde bulunan yeni interpolasyon yöntemlerinin kullanılması sağlanacaktır.

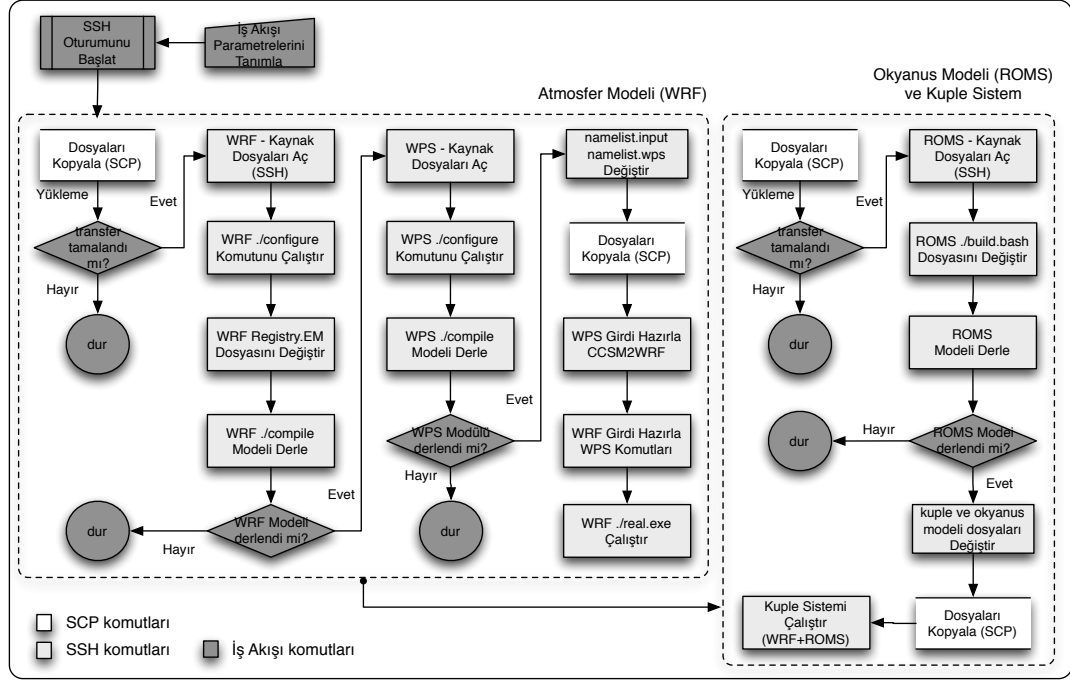
Geliştirilen kuple modelleme sistemi kullanılarak yapılan benzetimlerden elde edilen sonuçlar, EK D'de görülebilir. Sonuçlar, kuple model sistemi ile yapılmış bir aylık benzetim sonuçlarını ve bu sonuçlardan elde edilen karşılařtırmalı çizimleri içermektedir. Şekiller incelendiđinde, sistemin özellikle atmosferik karışma oranı ve okyanus yüzey akıntıları üzerindeki etkisi açıkça görülmektedir. Modelden elde edilen sonuçların daha uzun süreli simülasyonlar yapılarak ve gözlemler ile karşılařtırılarak incelenmesi gerekliliđi açıktır. Ancak bu aşamalar tezin konusu dahilinde deđildir ve daha sonraki arařtırmalara bırakılmıştır.

## **5.8 Geliştirilen Aktörler ve Tasarlanan İş Akışı**

Bu bölüm, önceki bölümlerde (5.4 ve 5.7) detaylı olarak tanımlanan kuple edilmiş model sisteminin iş akış sistemine entegre edilmesi için geliştirilen aktörler ve iş akış sistemi hakkında detaylı bilgi vermeyi amaçlamaktadır. Model sisteminin iş



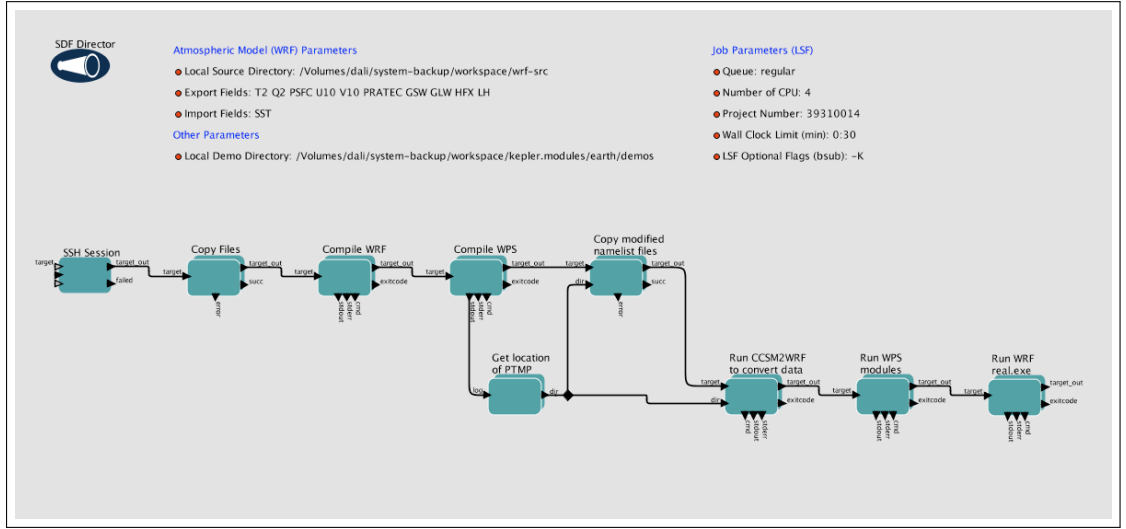
akış sistemi ile birleştirilme sürecini kolaylaştırması için öncelikle her bir alt model (atmosfer ve okyanus), tekil olarak iş akış sistemi ile entegre edilmiş, daha sonra üretilen iş akışları birleştirilerek kuple model sistemi için gerekli iş akış tasarımı elde edilmiştir. Şekil 5.7’de görülebilecek mantıksal iş akışı, kuple model sisteminin çalıştırılması için gerekli tüm temel adımları içermektedir.



**Şekil 5.7:** Kuple (WRF ve ROMS) edilmiş model sistemine ait mantıksal iş akışı tasarımı

Mantıksal iş akışı incelendiğinde, birbiri ile ilişkili iki grup iş akışı bulunduğu görülmektedir. Bunlardan birincisi; atmosfer modelinin derlenmesi ve çalışmaya hazırlanması ile ilgili aşamaları içermektedir. İkinci grup ise; okyanus modeli ile ilgili adımları içermektedir. İş akışının Kepler uygulaması kullanılarak tasarlanması ve çalıştırılması için hesaplama kaynağı olarak, bir küme sistem olan NCAR’a ait Bluefire makinası seçilmiştir. Bu nedenle, uzaktaki sunucu ile iletişim sağlayarak, komutların çalıştırılması ve dosyaların kopyalanması amacıyla Kepler içinde bulunan SSH ve SCP aktörleri kullanılmıştır. Kuple modelleme sistemi, ROMS okyanus modeli içerisine entegre edildiği ve ROMS modelinin derlenmesi için WRF modelinin modüllerine ihtiyaç olduğu için öncelikle WRF modelinin iş akış sistemine aktarılması gereklidir. Bu amaçla, WRF modeline ve bu modele gerekli girdi verisinin üretilmesi amacıyla kullanılan WPS bileşenine ait ayar dosyasının (namelist.wps) değiştirilmesi

için WRFEnvModify olarak isimlendirilen bir Kepler aktörü tasarlanmıştır. Aktör, temel olarak WRF ve WPS uygulamalarına ait ayar dosyalarını (namelist.input ve namelist.wps) değiştirerek, bunları eşsiz bir dosya ismi ile (.earth dizini altında) kaydetmektedir. Geliştirilen Kepler aktörü, ayar dosyaları içindeki değişikliklere uyumlu bir şekilde tasarlanmıştır ve aktör parametreleri, bu ayar dosyaları kullanılarak otomatik olarak üretilmektedir. Şekil 5.7’te görülen mantıksal iş akışının atmosferik model bileşenine ait bölümünün Kepler uygulamasına dönüştürülmüş hali, Şekil 5.8’te görülebilir.

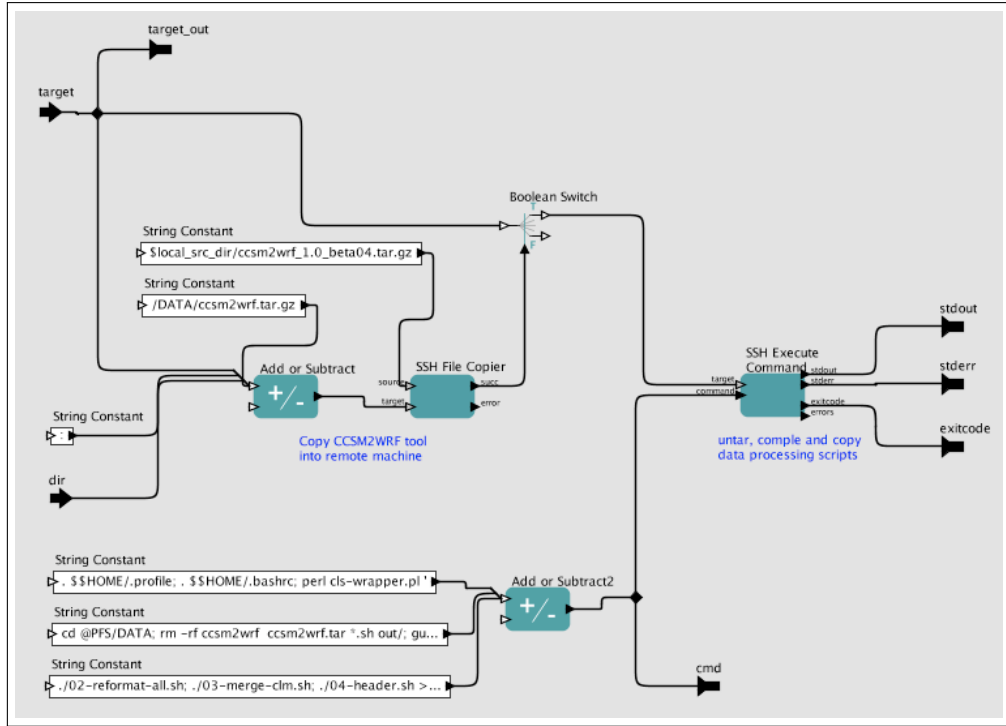


**Şekil 5.8:** WRF iş akışı tasarımı

Kepler iş akışı incelendiğinde, iş akışının işleyişini değiştiren ve tüm aktörler tarafından kullanılacak üst seviye parametreler görülebilir. Bu parametreler ile, modele ait kaynak dosyaların bulunduğu dizinler, atmosfer modelinin okyanus modeli ile kuple aşamasında kullanacağı alan değişkenlerinin listesi ve uzak sisteme verilecek iş ile ilgili seçenekler (örneğin, LSF parametreleri) tanımlanabilir.

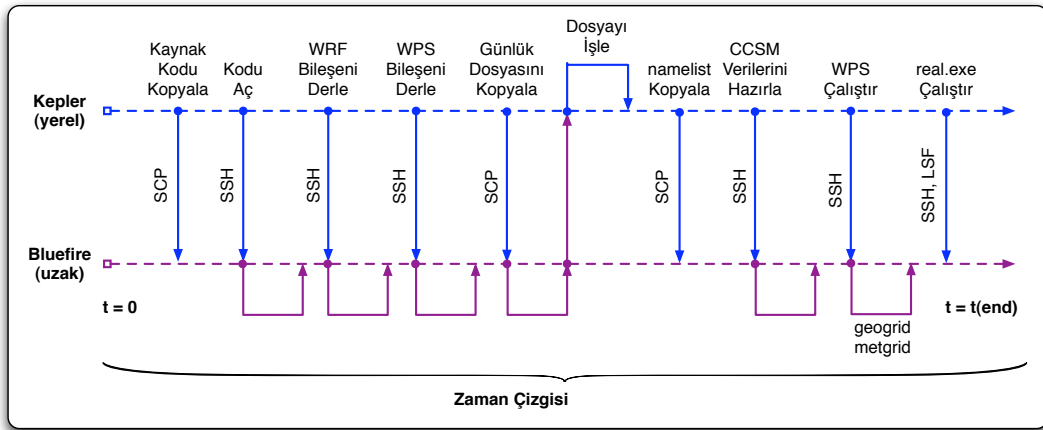
Daha önceki bölümlerde detaylı olarak aktarıldığı gibi Kepler farklı seviyelerde ve hiyerarşik bir şekilde iş akışı tasarlanmasını kompozit aktörler yolu ile desteklemektedir. Şekil 5.8’de görülen ana iş akışı kompozit aktörler kullanarak farklı süreçlerin modellenmesi için alt iş akışlarının tanımlanmasını sağlamaktadır (Bkz. Şekil 5.9).

Alt iş akışı, uzak sisteme kopyalanmış olan veri dönüştürme uygulamasını sisteme kurarak çalıştırmakta ve girdi olarak verilen CCSM küresel modeline ait CAM ve CLM bileşenlerinin çıktılarını WRF modelinin okuyabileceği bir formatta diske



Şekil 5.9: WRF iş akış tasarımı (CCSM2WRF kompozit aktörü)

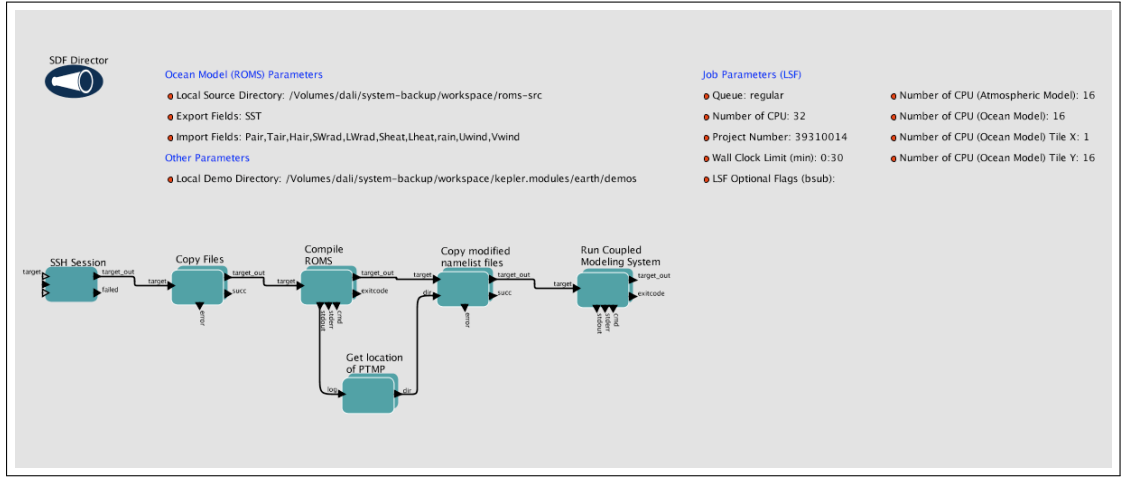
yazmaktadır. Bu şekilde her bir süreç için özelleşmiş iş akışları, kompozit aktörlerin kullanılması ile geliştirilerek, var olan aktörlerin kullanıldığı yeni aktörler tasarlanmıştır. İş akışını ve yapılan işlemleri bir zaman çizelgesi üzerinde göstermek istersek, iş akışının izlediği adımlar daha kolay izlenebilir (Bkz. Şekil 5.10).



Şekil 5.10: WRF iş akışının zaman çizelgesi üzerine aktarılmış hali

Daha önce okuyucuya aktarıldığı gibi, kuple iş akışının yaratılması amacıyla her bir alt modelin iş akışının ayrı ayrı yaratılması gereklidir. Atmosferik model bileşeni

olan WRF modelinin iş akışının tamamlanması ile, okyanus bileşeni olan ROMS modelinin kurulması ve çalıştırılması ile ilgili diğer iş akışı tanımlanabilir. Bu amaçla, Şekil 5.7'deki mantıksal iş akışını tanımlayan diyagramın sağ tarafındaki süreçlerin de, Kepler iş akışı ortamına aktarılması gereklidir. Şekil 5.11'de ROMS modeli için hazırlanmış iş akışı görülebilir. Tasarlanan iş akışı, WRF modelinin kurulduğu dizini otomatik olarak bulur ve bu dizindeki gerekli dosyaları, ROMS modelini derlemesi amacıyla, derleme aşamasında kullanılan dosyanın içinde tanımlar. Bu aşamadan sonra iş akışı; modelin kurulumu, ayar dosyalarının değiştirilmesi ve kuple model sisteminin çalıştırılması aşamalarını izleyerek son bulur. WRF modeli için tanımlanan iş akışında olduğu gibi ROMS modeli için de, modeller arasında değiştirilecek veya transfer edilecek alan değişkenlerinin listesinin tanımlanması mümkündür.



**Şekil 5.11:** ROMS iş akışı tasarımı

Geliştirilen her iki iş akışı için aşağıda sıralanan adımlar, Kepler iş akışı çalıştırılmadan önce yapılmalıdır. İş akışlarının sonraki sürümlerinde bu adımların da otomatikleştirilmesi ve iş akışına entegre edilmesi planlanmaktadır.

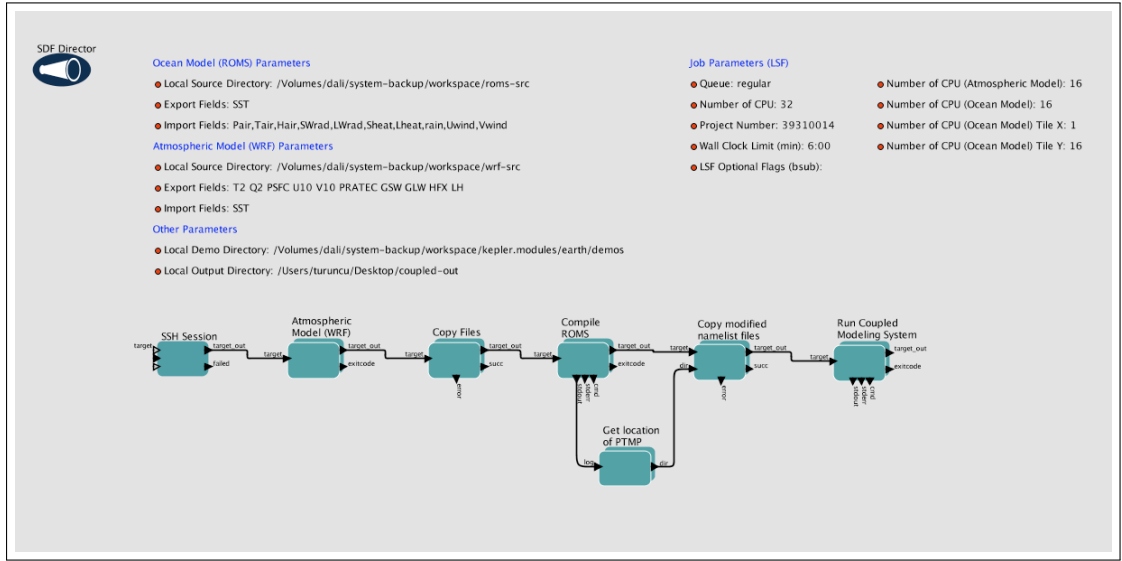
- Girdi CCSM verilerinin, uzak hesaplama sistemine ait paralel dosya sisteminin bulunduğu dizine (\$PTMP/DATA) daha önceden kopyalanması gereklidir. Verilerin bulunduğu ESG veri geçidinin şu anki sürümü sistem içerisinde bulunan veri kümelerinin dışarıdan sorgulanmasına izin vermemektedir. Ancak, tasarlanan yeni sürümün Hessian<sup>1</sup> ağ servisini kullanarak bu desteği vermesi ve üçüncül yazılımlar

<sup>1</sup><http://hessian.caucho.com/>

ile veri kümelerinin sorgulanmasını desteklemesi planlanmaktadır. ESG veri geçidinin yeni sürümü ile birlikte bu desteğin iş akışlarına eklenmesi sağlanacaktır.

- ROMS modeli için gerekli olan girdi veri dosyalarının ve modelin çalışacağı coğrafik alanı belirten grid dosyasının daha önceden hazırlanarak, \$PTMP/DATA dizini altına konulması gereklidir. Bu aşama, grid dosyasının hazırlanmasının zorluğu ve araştırmacının bu grid dosyasına müdahale etmesinin gerekliliği nedeniyle otomatikleştirilmemiş ve iş akışına entegre edilmemiştir. Paralel dosya sisteminin bulunduğu veya erişilebileceği dizin (\$PTMP), yazılan Perl betiği ile otomatik olarak iş akışı tarafından bulunmaktadır.

Tekil olarak tasarlanan iş akışlarını birleştirilerek, kuple model sistemi akışını oluşturmak için, atmosfer modelini temsil eden iş akışı bir kompozit aktör yardımı ile, ROMS okyanus modeli iş akışına entegre edilmiştir (Şekil 5.12).



Şekil 5.12: Kuple model sistemi için iş akış tasarımı

CCSM modeli sınama örneğinde olduğu gibi, kullanılan yer sistem modellerinden köken bilgisinin otomatik olarak toplanması amacıyla, her iki model bileşeninin derleme aşamasında kullanılan betikleri, değiştirilerek geliştirilen Python betiklerini kullanması sağlanmıştır. Bu şekilde, modellerin derleme aşamasında kullandığı parametreler ve çevre değişkenleri, bir XML dosyası içinde toplanarak kayıt altına alınmıştır. Şekil 5.13’de ROMS modeli tarafından üretilmiş XML dosyasının bir bölümü görülebilir.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <system_provenance>
3   <run>
4     <component>
5       rev_382
6     </component>
7     <executable>
8       /ptmp/turuncu/case/oceanM
9     </executable>
10    <checksum>
11      d2c898797c9f6aaa2ef678b0bdd2a2a9
12    </checksum>
13    <user>
14      turuncu
15    </user>
16    <time>
17      09-09-28 15:35:41
18    </time>
19  </run>
20  <compilers>
21    <compiler>
22      <property name="command" value="mpxlf95_r"/>
23      <property name="description" value="IBM XL Fortran Enterprise Edition for AIX, V11.1"/>
24      <property name="flags" value="-qsuffix=f=f90 -qmaxmem=-1 -qarch=pwr4 -qtune=pwr4 -q64 -
03 -qstrict -I/ptmp/turuncu/esmf_3.1.0rp2/mod/mod0/AIX.default.64.mpi.default -I/ptmp/turuncu/
esmf_3.1.0rp2/src/include"/>
25      <property name="language" value="Fortan"/>
26      <property name="version" value="11.01.0000.0007"/>
27    </compiler>
28  </compilers>
29  <environment>
30    ...
31    <variable name="ESMF_ABI" value="64"/>
32    <variable name="ESMF_BOPT" value="0"/>
33    <variable name="ESMF_COMM" value="mpi"/>
34    <variable name="ESMF_COMPILER" value="default"/>
35    <variable name="ESMF_DIR" value="/ptmp/turuncu/esmf_3.1.0rp2"/>
36    <variable name="ESMF_FORTRANSYMBOLS" value="default"/>
37    <variable name="ESMF_INSTALL_PREFIX" value="/ptmp/turuncu/esmf_3.1.0rp2/install_dir"/>
38    <variable name="ESMF_MACHINE" value="default"/>
39    <variable name="ESMF_MPIBATCHOPTIONS" value="-K -qshare -P 7700014 -W60"/>
40    ...
41    <variable name="ROMS_APPLICATION" value="MED_COUPLED"/>
42    <variable name="SCRATCH_DIR" value="/ptmp/turuncu/case/Build"/>
43    <variable name="SHELL" value="/bin/sh"/>
44    ...
45  </environment>
46 </system_provenance>

```

**Şekil 5.13:** ROMS modeli örnek sistem köken bilgisi çıktısı (NCAR Bluefire, IBM Küme)

XML dosyasının içeriğinden görülebileceği gibi, modelin derlenmesi aşamasında kullanılan derleyici ve derleyici seçenekleri XML dosyası içerisinde özel bir grup olarak tutulmaktadır. Bu grup dışında, modellerin kuple edilmesinde kullanılan ESMF kütüphanesine ait çevre değişkeni tanımlamaları da, bu dosya içinde saklanmaktadır. Böylelikle, kuple model sisteminin nasıl derlendiği ve ESMF kütüphanesinin hangi sürümünü kullandığı, daha sonraki incelemeler için saklanmış olmaktadır. Model sisteminde oluşabilecek hataların bulunması, otomatik olarak tutulan bu kayıtların aralarındaki farkların incelenmesi ile mümkün olmaktadır.

Sistem köken bilgisi dışında, model bileşenleri içindeki değişkenlerin bir XML dosyası olarak diske yazılması ve veri köken bilgisi olarak kullanılması amacıyla, her iki modelin kodu içerisinde alan değişkenlerine ait ön verinin tutulması için, ESMF özellik sınıfları (ESMF\_Attribute) kullanılarak tanımlamalar yapılmıştır. Bu amaçla, WRF modeli ESMF-IO bileşeni üzerinde değişiklik yapılarak, hem alan

değişkeni seviyesinde hem de model bileşeni seviyesinde ön veri, durum nesnelere (ESMF\_State) eklenmiştir. Değiştirilen ESMF durum nesnelere, ROMS modeli içerisinde bulunan ESMF ana programı tarafından ESMF\_Attribute\_Write metodu kullanılarak, XML dosyasına yazılır. Veri köken bilgisine örnek olması açısından, ROMS ve WRF modeline ait XML dosyaları Şekil 5.14 ve 5.15'te görülebilir.

```

1 <model_component name="ROMS" full_name="Regional Ocean Modeling System" version="ESMFized 3.2"
2 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xsi:schemaLocation="http://www.esmf.ucar.edu file:/esmf_model_component.xsd"
4 xmlns="http://www.esmf.ucar.edu">
5
6 <discipline_set>
7   <discipline name="Ocean" />
8 </discipline_set>
9
10 <physical_domain_set>
11   <physical_domain name="Earth system" />
12 </physical_domain_set>
13
14 <agency_set>
15   <agency name="Rutgers University" />
16 </agency_set>
17
18 <institution_set>
19   <institution name="Institute of Marine and Coastal Sciences" />
20 </institution_set>
21
22 <author_set>
23   <author name="Hernan G. Arango" />
24 </author_set>
25
26 <coding_language_set>
27   <coding_language name="Fortran 90" />
28 </coding_language_set>
29
30 <model_component_framework_set>
31   <model_component_framework name="ESMF (Earth System Modeling Framework)" />
32 </model_component_framework_set>
33
34 <variable_set>
35   <variable name="Pair"
36     standard_name="N/A"
37     long_name="surface air pressure"
38     units="millibar"
39     import="true"
40     export="false" />
41   ...
42 </variable_set>
43
44 </model_component>

```

**Şekil 5.14:** ROMS örnek veri köken bilgisi çıktısı

Şekilden görülebileceği gibi, XML dosyası başlık kısmında model bileşenin ismi, tanımlaması ve sürümü bulunmaktadır. XML dosyasının sonraki bölümü, iki ana kısma ayrılmıştır. Bunlardan birincisi; model bileşeni ile ilgili olan bilgiler, ikincisi ise; model bileşenin sahip olduğu değişkenler ile ilgili olan bilgilerdir. İlk grup içinde, model bileşenin hangi disipline ait olduğu, hangi merkez tarafından ve hangi programlama dili kullanılarak geliştirildiği gibi bilgiler bulunmaktadır. İkinci grup köken bilgisi içinde ise, kuple model bileşenleri arasında, değiştirilecek alan değişkenlerinin ismi, standart ismi, tanımlaması, birimi ve değişkenin model bileşenine olan yönünü (girdi veya çıktı gibi) içeren çeşitli ön veri bilgileri saklanmaktadır.

```

1 <model_component name="WRF" full_name="Weather Research and Forecasting"
  version="Standard 3.1.1"
2 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xsi:schemaLocation="http://www.esmf.ucar.edu file:/esmf_model_component.xsd"
4 xmlns="http://www.esmf.ucar.edu">
5
6 <discipline_set>
7   <discipline name="Atmosphere" />
8 </discipline_set>
9
10 <physical_domain_set>
11   <physical_domain name="Earth system" />
12 </physical_domain_set>
13
14 <agency_set>
15   <agency name="NCAR (National Center for Atmospheric Research)" />
16 </agency_set>
17
18 <institution_set>
19   <institution name="MMM (Mesoscale and Microscale Meteorology Division)" />
20 </institution_set>
21
22 <author_set>
23   <author name="M3 Team" />
24 </author_set>
25
26 <coding_language_set>
27   <coding_language name="Mixed Fortran 90 and C" />
28 </coding_language_set>
29
30 <model_component_framework_set>
31   <model_component_framework name="ESMF (Earth System Modeling Framework)" />
32 </model_component_framework_set>
33
34 <variable_set>
35   <variable name="SST"
36     standard_name="sea_surface_temperature"
37     long_name="Sea surface temperature"
38     units="Kelvin"
39     import="true"
40     export="false" />
41   <variable name="Q2"
42     standard_name="surface_mixing_ratio"
43     long_name="Mixing ratio at 2 meter"
44     units="kg kg-1"
45     import="false"
46     export="true" />
47   ...
48 </variable_set>
49
50 </model_component>

```

**Şekil 5.15:** WRF örnek veri köken bilgisi çıktısı

Yukarıda detaylı olarak açıklanan farklı tipteki ön veri ve köken bilgileri, geliştirilen iş akışının ve kuple model sisteminin kendi kendisini tanımlaması açısından son derece önemlidir. Böylece, tüm modelleme sisteminin işleyişinin kayıt altında tutulması mümkün olmaktadır.

## 5.9 Uygulama Örneği Sonuçları

Kuple model sistemi için geliştirilen mantıksal iş akışının, bir önceki sınama uygulaması gibi, geliştirilen metodoloji kullanılarak Kepler iş akışına aktarılması bu çalışma için oldukça önemlidir. Bu şekilde, önerilen iş akışı ortamının farklı yer



sistem uygulamaları için esnek bir yapıda olup olmadığı ve farklı uygulamaların aynı metodoloji ile olan uyumu sınanmış olmaktadır.

Bu sınama uygulaması ile birlikte geliştirilen kuple modelleme sistemi, başarılı bir şekilde Kepler iş akışı ortamına aktarılmıştır. Böylece, araştırmacı ile, iş akış ortamı tarafından kontrol edilen hesaplama kaynağı ve yer sistem modeli gibi farklı bileşenler arasında, bir soyutlaştırma katmanı geliştirilmiştir. Bu şekilde, kullanımı oldukça zor ve karmaşık olan kuple model sistemi, ortalama bir araştırmacının kullanabileceği basitlik seviyesine indirgenmiştir. Geliştirilen kuple model sistemi, farklı tipteki köken bilgilerinin otomatik olarak toplanabileceği bir çalışma ortamını araştırmacıya sunmaktadır. Köken bilgisinin standart bir şekilde toplanması açısından bakıldığında, model bileşeni bazında toplanan tanımlayıcı bilgi, bir önceki sınama uygulamasında, değişken bazında toplanan köken ve ön veri bilgisinden farklıdır.

## 6. SONUÇLAR

Giriş bölümünde okuyucuya kısaca aktarılan problemlerin çözülmesi için geliştirilmesi amaçlanan iş akışı çalışma ortamı, bu ortamın tasarımı için önerilen metodolojinin farklı yer sistem modelleri ve bunlara ait uygulamalar ile sınanması, daha önce de anlatıldığı gibi, çalışmanın asıl amacını oluşturmaktadır. Bu amaçla, daha önceki bölümlerde detaylı olarak tanımlanan yeni bir metodoloji geliştirilmiş ve geliştirilen metodoloji, gerçekçi uygulamalar ile sınanmıştır.

Önerilen metodoloji, problemlerin çözülmesi için literatürde bulunan, iş akışı ve çerçeve yaklaşımlarının birleştirilerek ve bir arada kullanılmasını gerektirmektedir. Bu şekilde, her iki yöntemin kendine özgü avantajları kullanılarak, birbirlerinin eksik yanlarını tamamlayacağı ve kullanılan yer sistem modeli ile ilgili ön veri ve köken bilgisinin otomatik olarak toplanabileceği yeni bir çalışma ortamı tasarlanmıştır.

İş akışı, karmaşık süreçler bütününi daha basit alt parçalara ayırarak, modüler bir hale getirmeyi ve tüm süreçlerin modellenmesi için her bir süreci birbirleri ile olan ilişkileri doğrultusunda sıralı bir şekilde birleştirmeyi amaçlamaktadır. Modüler bir yapı altında toplanan alt süreçler, farklı amaçlar doğrultusunda kullanılarak, değişik iş akışlarının tasarlanabileceği yeni bir çalışma ortamı yaratmış olur. Önceki bölümlerde detaylı olarak anlatılan sına örneklerinin iş akış ortamına aktarılması için, iş akış ortamı içinde yeni süreçlerin aktör yapıları ile tanımlanmasının kolay olması ve iş akışının, farklı hesaplama modellerini desteklemesi gereklidir. Sına uygulamaları incelendiğinde, her bir örnek için hem hesaplama kaynağı, hem de yer sistem modeli ile ilişkili birçok aktörün geliştirildiği ve farklı seviyelerde iş akışlarının tasarlandığı görülebilir. Çalışmada, iş akışlarının tasarlanması ve her bir alt süreci tanımlayan aktörlerin geliştirilmesi için Kepler uygulaması kullanılmıştır. Kepler, bu aşamada platform bağımsız olması, hiyerarşik bir şekilde iş akışları geliştirilmesine izin vermesi, farklı hesaplama modellerini (MoC) desteklemesi, köken bilgisi toplanması amacıyla özelleşmiş aktörlerinin bulunması ve kolayca geliştirilebilir bir

şekilde tasarlanmış olması nedeniyle, sına ma uygulamalarının geliştirilmesi sürecini kolaylaştırmıştır.

İş akışı uygulaması ile çerçeve yaklaşımının birleştirilmesi amacıyla kullanılan ESMF kütüphanesi, sına ma uygulamalarında kullanılan yer sistem modellerinden, köken bilgisinin otomatik bir şekilde toplanması ve farklı yer sistem modellerinin birbirleri ile kuple edilmesi için kullanılmıştır. ESMF, benzer kütüphanelerden farklı olarak, bünyesinde bulundurduğu ön veri ile ilişkili yapılar sayesinde, yer sistem modelleri içerisine alan değişkeni ve model seviyesinde ön verinin eklenmesi için oldukça uygun bir çalışma ortamı sunmaktadır. Ayrıca ESMF, OpenMI<sup>1</sup> uygulaması benzeri ek özellikleri ile, modellerin ağ servislerine entegre edilmesini kolaylaştıran çeşitli yapılar içermektedir. Bu özelliklerin birlikte kullanılmasıyla, LEAD gibi bilimsel ağ geçitlerinin geliştirilmesinin ve tasarlanmasının kolaylaştırılması ve standartlaştırılması amaçlanmaktadır. Bu açıdan bakıldığında, özellikle ikinci sına ma uygulamasından elde edilen sonuçlar, kendi kendisini tanımlayan yer sistem modellerinin önerilen metodoloji ile geliştirilebileceğini göstermesi açısından oldukça önemli bir örnek oluşturmaktadır.

Uygulama örneklerinden elde edilen sonuçlar incelendiğinde, her bir sına ma uygulamasının başarı ile geliştirilen çalışma ortamı üzerinden kullanılabilirdiği ve yer sistem bilimleri çalışmaları için çok önemli bir yer tutan köken bilgisinin otomatik bir şekilde toplanabilirdiği açıkça görülmektedir. İş akış sistemlerinin kullanılması ile, hesaplama kaynağının ve yer sistem modelinin anlamlı bir seviyede soyutlaştırılarak, modelleme sisteminin karmaşıklığının ortalama bir araştırmacı seviyesine indirgenmesi, elde edilen sonuçların önemli bir parçasını oluşturmaktadır. Bu çalışma ile, iş akışı sistemlerinin çerçeve uygulamaları ile birleştirilmesinin, yer sistem bilimciler için problem olan birçok süreci kolaylaştırdığı veya ortadan kaldırdığı açık bir şekilde gösterilmektedir. Bu bakımdan, önerilen metodolojinin ileride geliştirilmesi planlanan yer sistem modelleme sistemleri için önemli bir çalışma ve örnek oluşturduğu açıktır.

Yapılan çalışmanın geliştirilmesi için ESMF grubu, tasarlanan iş akış sisteminin ve metodolojinin ESMF ağ servisleri ile entegre edilmesini amaçlayan bir projeyi sunmuş ve proje TeraGrid tarafından iki yıl desteklenecek şekilde kabul edilmiştir. Proje, temel olarak var olan CCSM iş akışını ağ servisleri ile entegre ederek, CCSM bileşenlerini

---

<sup>1</sup><http://www.openmi.org/>

ađ servisleri yolu ile alıřtırmayı amalamaktadır. İlk ařamada, CCSM modeline ait atmosfer bileřeni olan CAM'ın kullanılması planlanmaktadır. Bu amala, Kepler uygulamasının ESMF web servisleri ile uyumlu bir řekilde alıřacađı yeni bir sistem iř akıřı tasarlanacaktır. Bu, retilen sistemin uygulanabilirliđini gstermesi aısından nemli bir adımdır.

## KAYNAKLAR

- [1] **Hill, C., DeLuca, C., Balaji, V., Suarez, M. and da Silva, A.**, 2004. The Architecture of the Earth System Modeling Framework, *Computing in Science and Engineering*, **6(1)**, 18–28.
- [2] **Collins, N., Theurich, G., Deluca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., Yang, W., Hill, C. and Da Silva, A.**, 2005. Design and Implementation of Components in the Earth System Modeling Framework, *Int. J. High Perform. Comput. Appl.*, **19(3)**, 341–350.
- [3] **Larson, J., Jacob, R. and Ong, E.**, 2005. The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models, *Int. J. High Perform. Comput. Appl.*, **19(3)**, 277–292.
- [4] **Jacob, R., Larson, J. and Ong, E.**, 2005. M X N Communication and Parallel Interpolation in Community Climate System Model Version 3 Using the Model Coupling Toolkit, *Int. J. High Perform. Comput. Appl.*, **19(3)**, 293–307.
- [5] **Craig, A.P., Jacob, R., Kauffman, B., Bettge, T., Larson, J., Ong, E., Ding, C. and He, Y.**, 2005. CPL6: The New Extensible, High Performance Parallel Coupler for the Community Climate System Model, *Int. J. High Perform. Comput. Appl.*, **19(3)**, 309–327.
- [6] **Valcke, S., Guilyardi, E. and Larsson, C.**, 2006. PRISM and ENES: a European approach to Earth system modelling, *Concurr. Comput. : Pract. Exper.*, **18(2)**, 247–262.
- [7] **Valcke, S., Declat, D., Redler, R., Ritzdorf, H., Vogelsang, R. and Bourcier, P.**, 2003. The PRISM Coupling an I/O System, VECPAR 2004.
- [8] **Warner, J.C., Perlin, N. and Skillingstad, E.D.**, 2008. Using the Model Coupling Toolkit to couple earth system models, *Environ. Model. Softw.*, **23(10-11)**, 1240–1249.
- [9] **Warner, J.C., Sherwood, C.R., Signell, R.P., Harris, C.K. and Arango, H.G.**, 2008. Development of a three-dimensional, regional, coupled wave, current, and sediment-transport model, *Comput. Geosci.*, **34(10)**, 1284–1306.
- [10] **Valcke, S. and Redler, R.**, 2006. *OASIS4 User Guide*, CERFACS and NEC-CCRLE.

- [11] **Bowers, S. and Ascher, B.**, 2005. Actor-oriented design of scientific workflows, In 24st Intl. Conference on Conceptual Modeling, Springer, pp. 369–384, <http://citeseer.ist.psu.edu/bowers05actororiented.html>.
- [12] **Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J. and Zhao, Y.**, 2005. Scientific workflow management and the Kepler system, *Concurr. Comput. : Pract. Exper.*, **18**, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.6066>.
- [13] **Oinn, T., Addis, M., Ferris, J., Marvin, D., Carver, T., Pocock, M.R. and Wipat, A.**, 2004. Taverna: A tool for the composition and enactment of bioinformatics workflows, *Bioinformatics*, **20**, 2004.
- [14] **Majithia, S., Shields, M., Taylor, I. and Wang, I.**, 2004. Triana: a graphical Web service composition and execution toolkit, Web Services, 2004. Proceedings. IEEE International Conference on, pp. 514–521, <http://dx.doi.org/10.1109/ICWS.2004.1314777>.
- [15] **Price, P., Edwards, D., Cox, B., Krznaric, M., Newhouse, S. and Darlington, J.**, 2003. GENIE: Delivering e-Science to the Environmental Scientist, UK e-Science All Hands Meeting (AHM2003), September 2003, Nottingham, UK,, pp. 145–152, <http://pubs.doc.ic.ac.uk/Allhands2003-Genie-scientist/>.
- [16] **Plale, B., Gannon, D., Brotzge, J., Droegemeier, K., Kurose, J., McLaughlin, D., Wilhelmson, R., Graves, S., Ramamurthy, M., Clark, R.D., Yalda, S., Reed, D.A., Joseph, E. and Chandrasekar, V.**, 2006. CASA and LEAD: Adaptive Cyberinfrastructure for Real-Time Multiscale Weather Forecasting, *Computer*, **39(11)**, 56–64.
- [17] **Brewster, K., Weber, D., Thomas, K., Droegemeier, K., Wang, Y., Xue, M., Marru, S., Gannon, D., Alameda, J. and Jewett, B.**, 2008. Use of the LEAD portal for on-demand severe weather prediction, *Sixth Conference on Artificial Intelligence Applications to Environmental Science, 88th Annual Meeting of the American Meteorological Society, New Orleans*.
- [18] **Fairman, M.J., Price, A.R., Xue, G., Molinari, M., Nicole, D.A., Lenton, T.M., Marsh, R., Takeda, K. and Cox, S.J.**, 2009. Earth system modelling with Windows Workflow Foundation, *Future Generation Computer Systems*, **25(5)**, 586 – 597.
- [19] **Charlie Catlett, William E. Allcock, P.A. et al.**, 2008. TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Application, *High Performance Computing and Grids in Action*, **16**.
- [20] **Dunlap, R., Mark, L., Rugaber, S., Balaji, V., Chastang, J., Cinquini, L., Deluca, C., Middleton, D. and Murphy, S.**, 2008. Earth system curator: metadata infrastructure for climate modeling, *Earth Science Informatics*, **1(3)**, 131–149, <http://dx.doi.org/10.1007/s12145-008-0016-1>.
- [21] **Allcock, B., Foster, I., Nefedova, V., Chervenak, A., Deelman, E., Kesselman, C., Lee, J., Sim, A., Shoshani, A., Drach, B. and Williams, D.**, 2001.

High-performance remote access to climate simulation data: a challenge problem for data grid technologies, Supercomputing '01: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM), ACM, New York, NY, USA, pp. 46–46.

- [22] **David Bernholdt, Shishir Bharathi, D.B. et al.**, 2005. The Earth System Grid: Supporting the Next Generation of Climate Modeling Research, Proceedings of the IEEE, pp. 485–495.
- [23] METAFOR: Common Metadata for Climate Modelling Digital Repositories, <http://metaforclimate.eu/>.
- [24] CF Metadata: NetCDF Climate and Forecast (CF) Metadata Convention, <http://cf-pcmdi.llnl.gov/>.
- [25] Gridspec: a standard for the description of grids used in earth system models, <http://www.gfdl.noaa.gov/vb/gridstd/gridstd.html>.
- [26] RDF: Resource Description Framework, <http://www.w3.org/RDF/>.
- [27] OWL: Web Ontology Language Guide, <http://www.w3.org/TR/owl-guide/>.
- [28] **Gregory, J.M., Dixon, K.W., Stouffer, R.J., Weaver, A.J., Driesschaert, E., Eby, M., Fichefet, T., Hasumi, H., Hu, A., Jungclaus, J.H., Kamenkovich, I.V., Levermann, A., Montoya, M., Murakami, S., Nawrath, S., Oka, A., Sokolov, A.P. and Thorpe, R.B.**, 2005. A model intercomparison of changes in the Atlantic thermohaline circulation in response to increasing atmospheric CO<sub>2</sub> concentration, *Geophysical Research Letters*, **32**, L12703+, <http://dx.doi.org/10.1029/2005GL023209>.
- [29] **Covey, C., AchutaRao, K.M., Cubasch, U., Jones, P., Lambert, S.J., Mann, M.E., Phillips, T.J. and Taylor, K.E.**, 2003. An overview of results from the Coupled Model Intercomparison Project, *Global and Planetary Change*, **37(1-2)**, 103 – 133, evaluation, Intercomparison and Application of Global Climate Models.
- [30] Sesame, <http://www.openrdf.org/>.
- [31] **Bowers, S., McPhillips, T.M., Ludäscher, B., Cohen, S. and Davidson, S.B.**, 2006. A Model for User-Oriented Data Provenance in Pipelined Scientific Workflows, **Moreau and Foster** [33], pp. 133–147.
- [32] **Klasky, S., Barreto, R., Kahn, A., Parashar, M., Podhorszki, N., Parker, S., Silver, D. and Vouk, M.**, 2008. Collaborative visualization spaces for petascale simulations, Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on, pp. 203–211.
- [33] **Moreau, L. and Foster, I.T.**, editors, 2006. Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers, volume 4145 of *Lecture Notes in Computer Science*, Springer.

- [34] **National**, 2004. Understanding Metadata, Technical report, National Information Standards Organization, Bethesda, MD.
- [35] **Gruber, T.R.**, 1993. A translation approach to portable ontology specifications, *Knowl. Acquis.*, **5(2)**, 199–220.
- [36] **Coral, C., Francisco, R. and Mario, P.**, 2006. *Ontologies for Software Engineering and Software Technology*, Springer-Verlag, Berlin, Heidelberg.
- [37] **Gomez-Perez, A., Corcho, O. and Fernandez-Lopez, M.**, 2004. *Ontological Engineering : with examples from the areas Curator of Knowledge Management, e-Commerce and the Semantic Web. First Edition (Advanced Information and Knowledge Processing)*, Springer.
- [38] **Lee, E.A. and Neuendorffer, S.**, 2000. MoML - A Modeling Markup Language in XML - Version 0.4, Technical report, University of California at Berkeley, <http://www.gigascale.org/pubs/16.html>.
- [39] **Sussman, A.**, 2006. Building complex coupled physical simulations on the grid with InterComm, *Eng. with Comput.*, **22(3)**, 311–323.
- [40] **Bettencourt, M.T.**, 2002. Distributed Model Coupling Framework, *International Symposium on High-Performance Distributed Computing*, **0**, 284.
- [41] SCRIP user guide, <http://climate.lanl.gov/Software/SCRIP/SCRIPusers.pdf>.
- [42] **Jones, P.W.**, 1999. First- and Second-Order Conservative Remapping Schemes for Grids in Spherical Coordinates, *Monthly Weather Review*, **127(9)**, 2204–2210.
- [43] SoftEnv, Argonne National Laboratory, The MCS Systems Administration Toolkit, <http://www.mcs.anl.gov/hs/software/systems/softenv/softenv-intro.html>.
- [44] **Furlani, J.L. and Osel, P.W.**, 1996. Abstract Yourself With Modules, LISA '96: Proceedings of the 10th USENIX conference on System administration, USENIX Association, Berkeley, CA, USA, pp. 193–204.
- [45] **Tang, X., Zhang, Z., Sun, B., Li, Y., Li, N., Wang, B. and Zhang, X.**, 2008. Antarctic ice sheet GLIMMER model test and its simplified model on 2-dimensional ice flow, *Progress in Natural Science*, **18(2)**, 173 – 180.
- [46] **von Laszewski, G., Foster, I.T., Gawor, J. and Lane, P.**, 2001. A Java commodity grid kit, *Concurrency and Computation: Practice and Experience*, **13(8-9)**, 645–662.
- [47] **Raisanen, J., Hansson, U., Ullerstig, A., Döscher, R., Graham, L.P., Jones, C., Meier, H.E.M., Samuelsson, P. and Willen, U.**, 2004. European climate in the late twenty-first century: regional simulations with two driving global models and two forcing scenarios, *Climate Dynamics*, **22(1)**, 13 – 31.



- [48] **Somot, S., Sevault, F., Dèquè, M. and Crèpon, M.**, 2008. 21st century climate change scenario for the Mediterranean using a coupled atmosphere-ocean regional climate model, *Global and Planetary Change*, **63(2-3)**, 112 – 126, mediterranean climate: trends, variability and change.
- [49] **Skamarock, W.C., Klemp, J.B., Dudhia, J., Gill, D.O., Barker, D.M., Duda, M.G., Huang, X.Y., Wang, W. and Powers, J.G.**, 2008. A Description of the Advanced Research WRF Version 3, Technical Report NCAR/TN-475+STR, National Center for Atmospheric Research, [http://www.mmm.ucar.edu/wrf/users/docs/arw\\_v3.pdf](http://www.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf).

## **EKLER**

**EK A** : Kepler İş Akışı Uygulaması Örnek Aktör Tanımlama Dosyaları

**EK B** : Sistem Köken Bilgisi ve Betik Yapısı

**EK C** : Örnek SPARQL Sorguları

**EK D** : WRF ve ROMS Kuple Model Sistemi Benzetim Sonuçları

## EK A

### Kod A.1: Kepler örnek aktör yapısı

```
/** One line description of the class and copyright notice.
*/

package name;

import method_name; /-- in alphabetical order --//

////////////////////////////////////
/// ClassName

/** Class documentation

    @author Author Name
    @version $Id$
*/

public class ClassName extends TypedAtomicActor { /-- Transformer, Source etc. --//
    /** Constructor description
    */
    public ClassName (CompositeEntity container, String name)
        throws NameDuplicationException, IllegalArgumentException {
        super(container, name);

        /-- Constructors create the actor and its parameters & ports --//
    }

    //////////////////////////////////////
    /// public ports and parameters ///

    /** Public variable description
    */
    public variables; /-- in alphabetical order --//

    //////////////////////////////////////
    /// public methods ///

    /** Public method description
    */
    public methods; /-- in alphabetical order --//
    /-- Example methods: attributeChanged(), fire(), prefire(), initialize() etc. --//

    //////////////////////////////////////
    /// protected methods ///
```

```

/** Protected method description
*/
protected methods; //-- in alphabetical order --//

////////////////////////////////////
//// protected variables ///

/** Protected variable description
*/
protected variables; //-- in alphabetical order --//

////////////////////////////////////
//// private methods ///

/** Private method description
*/
private methods; //-- in alphabetical order --//
//-- User defined methods --//

////////////////////////////////////
//// private variables ///

/** Private variable description
*/
private variables; //-- in alphabetical order --//
//-- User defined private variables --//
}

```

Kod A.2: Kepler örnek MANIFEST.MF dosyası

```

Manifest-Version: 1.4.2
KAR-Version: 1.0
lsid: urn:lsid:org.earth.actors.util:kar:12:1

Name: ArrayColumn.xml
lsid: urn:lsid:org.earth.actors.util:actor:12:1
type: actorMetadata

```

Kod A.3: Kepler örnek aktör tanımlama dosyası (XML)

```

<?xml version="1.0"?>
<entity name="ArrayColumn" class="ptolemy.kernel.ComponentEntity">
  <!-- Actor short definition -->
  <property name="entityId" value="urn:lsid:org.earth.actors.util:actor:12:1"
    class="org.kepler.moml.NamedObjId"/>

  <property name="KeplerDocumentation"
    class="ptolemy.vergil.basic.KeplerDocumentationAttribute">

  <property name="description"
    class="ptolemy.kernel.util.ConfigurableAttribute">

```

```
<configure>This actor is used to split specific column of the array.</configure>
</property>
```

```
<!-- Actor creator -->
```

```
<property name="author" class="ptolemy.kernel.util.ConfigurableAttribute">
  <configure>Ufuk Utku Turuncoglu</configure>
</property>
```

```
<!-- Actor version -->
```

```
<property name="version" class="ptolemy.kernel.util.ConfigurableAttribute">
  <configure>1.0</configure>
</property>
```

```
<!-- Actor documentation -->
```

```
<property name="userLevelDocumentation"
class="ptolemy.kernel.util.ConfigurableAttribute">
  <configure>
    <p>This actor is used to extract specified column data from multi-column
    ASCII file. The file must be read using String Splitter actor. It uses
    two parameter, the first one is number of column in ASCII file and the
    second one is the index of the column that will be extracted. The index
    values start from zero (0), it means that it uses zero based indexing.
    The output is an array that is the same data type with input.</p>
  </configure>
</property>
```

```
<!-- port definition (input) -->
```

```
<property name="port:input" class="ptolemy.kernel.util.ConfigurableAttribute">
  <configure>An input port that receives the multi-dimensional array.</configure>
</property>
```

```
<!-- port definition (output) -->
```

```
<property name="port:output" class="ptolemy.kernel.util.ConfigurableAttribute">
  <configure>An output port that sends the selected column of the input array.</configure>
</property>
```

```
<!-- parameter defintiton (size of array) -->
```

```
<property name="prop:ndims" class="ptolemy.kernel.util.ConfigurableAttribute">
  <configure>This parameter sets the size of the multi-dimensional array.</configure>
</property>
```

```
<!-- parameter definition (dimension index) -->
```

```
<property name="prop:index" class="ptolemy.kernel.util.ConfigurableAttribute">
  <configure>This parameter sets the index value of array column that will
  be extracted.</configure>
</property>
```

```
</property>
```

```
<property name="class" value="org.earth.actors.util.ArrayColumn"
class="ptolemy.kernel.util.StringAttribute">
```

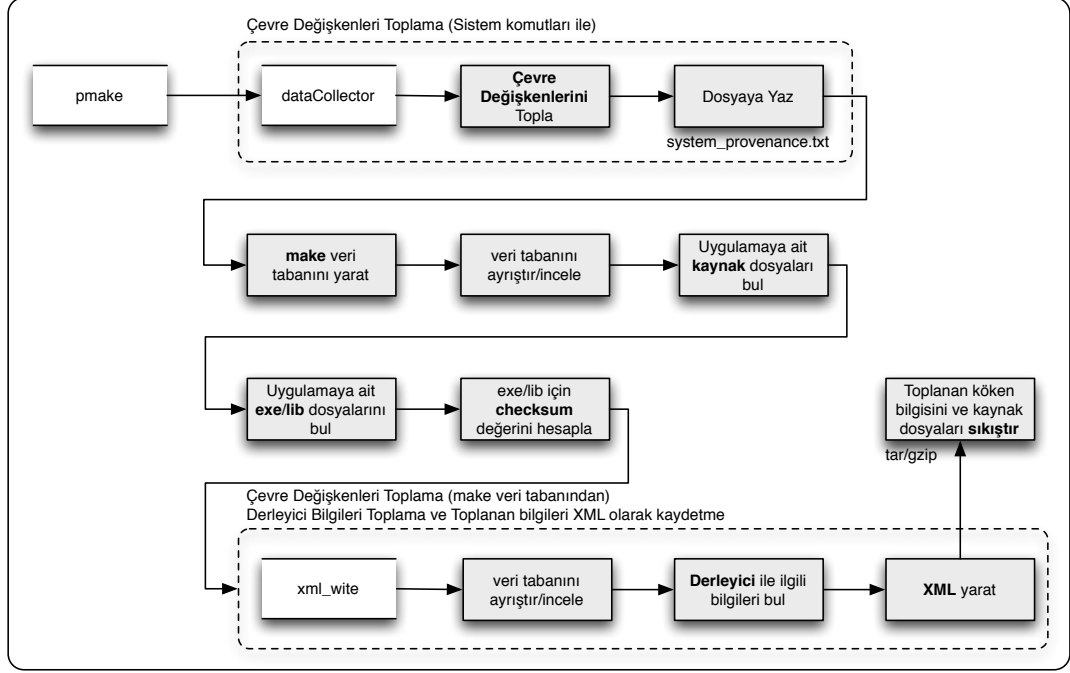
```
<property name="id" value="urn:lsid:org.earth.actors.util:class:12:1"
  class="ptolemy.kernel.util.StringAttribute"/>
</property>

<!-- Each port is listed with its properties and documentation -->
<property name="input" class="org.kepler.moml.PortAttribute">
  <property name="direction" value="input"
    class="ptolemy.kernel.util.StringAttribute"/>
  <property name="dataType" value="unknown"
    class="ptolemy.kernel.util.StringAttribute"/>
  <property name="isMultiport" value="false"
    class="ptolemy.kernel.util.StringAttribute"/>
</property>

<property name="output" class="org.kepler.moml.PortAttribute">
  <property name="direction" value="output"
    class="ptolemy.kernel.util.StringAttribute"/>
  <property name="dataType" value="unknown"
    class="ptolemy.kernel.util.StringAttribute"/>
  <property name="isMultiport" value="false"
    class="ptolemy.kernel.util.StringAttribute"/>
</property>

<property name="semanticType00" value="urn:lsid:localhost:onto:2:1#Constant"
  class="org.kepler.sms.SemanticType"/>
</entity>
```

## EK B



Şekil B.1: Sistem köken bilgisini toplayan betiğin yapısı

## EK C

### Kod C.1: CDOOperations aktöründe kullanılan örnek SPARQL sorguları

#### OWL bilgi tabanında tanımlanmış tüm CDO komutları nedir?

```
PREFIX base:<http://www.owl-ontologies.com/Ontology1241930231.owl#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?x ?y ?z ?t
WHERE {
  ?x rdfs:comment ?y.
  ?x base:hasInput ?z.
  ?x base:hasOutput ?t.
}
ORDER BY ?x
```

#### selname CDO komutuna ait olan parametreler nedir?

```
PREFIX base:<http://www.owl-ontologies.com/Ontology1241930231.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?p ?c
WHERE {
  ?x base:hasParameter ?p.
FILTER (?x = base:cdo.selname)
OPTIONAL { ?p base:isCalculatedBy ?c }
}
```

#### selname CDO komutu varsayılan bir değere sahip mi? ve Değer nedir?

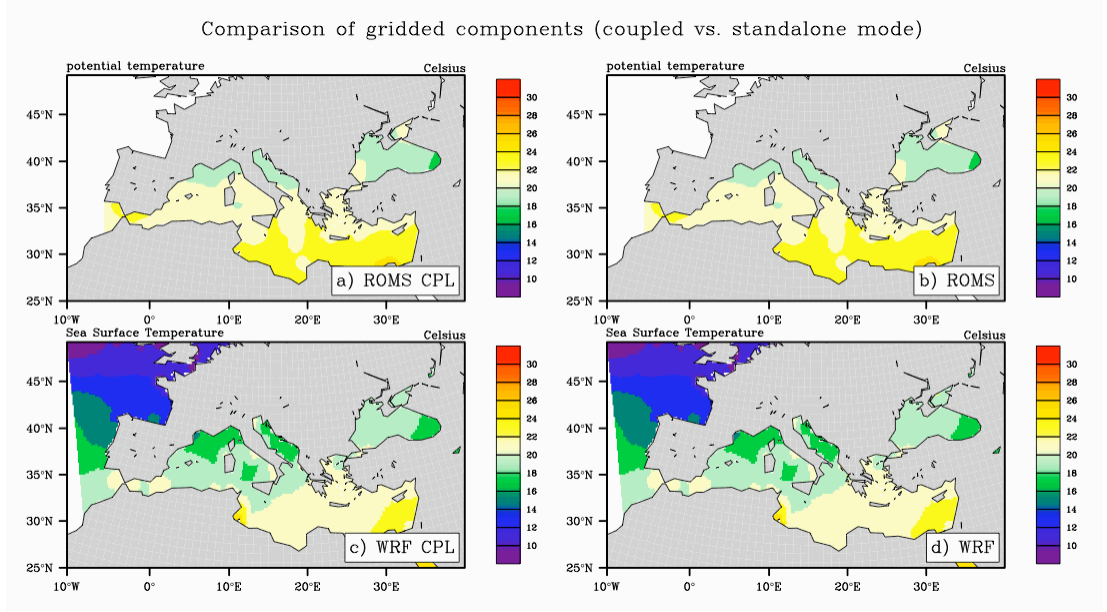
```
PREFIX base:<http://www.owl-ontologies.com/Ontology1241930231.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?p ?d
WHERE {
  ?x base:hasEnvironment ?p.
FILTER (?x = base:cdo.selname)
OPTIONAL { ?p base:hasDefaultValue ?d }
}
```

#### CDO komutuna ait REMAP\_EXTRAPOLATE çevre değişkeni hangi opsiyonlara sahip?

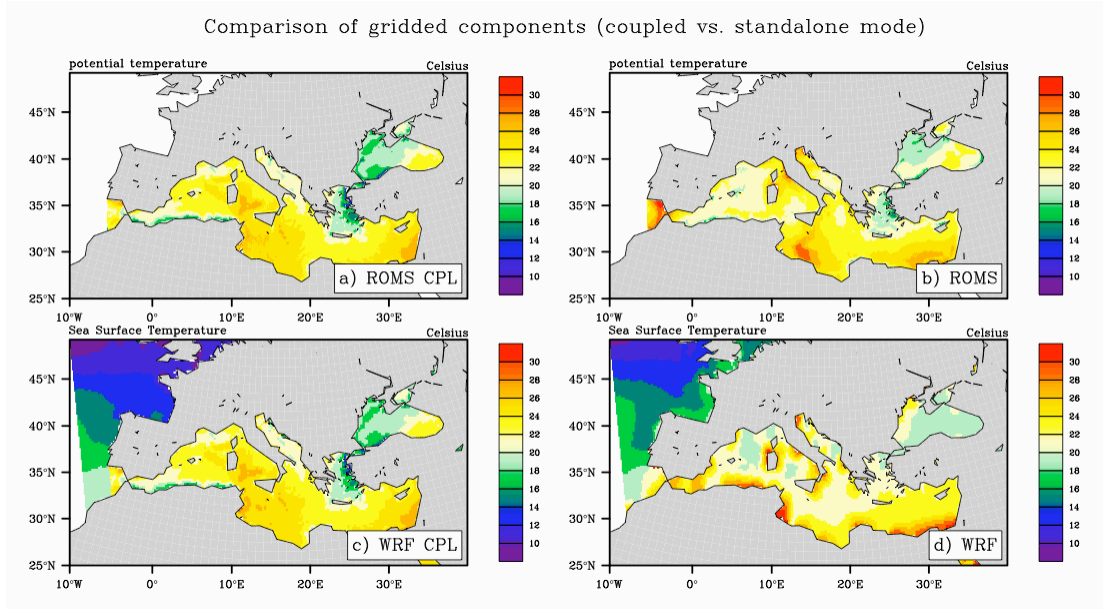
```
PREFIX base:<http://www.owl-ontologies.com/Ontology1241930231.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?o
WHERE {
  ?x base:hasOptions ?o.
FILTER (?x = base:cdo.REMAP_EXTRAPOLATE)
}
```



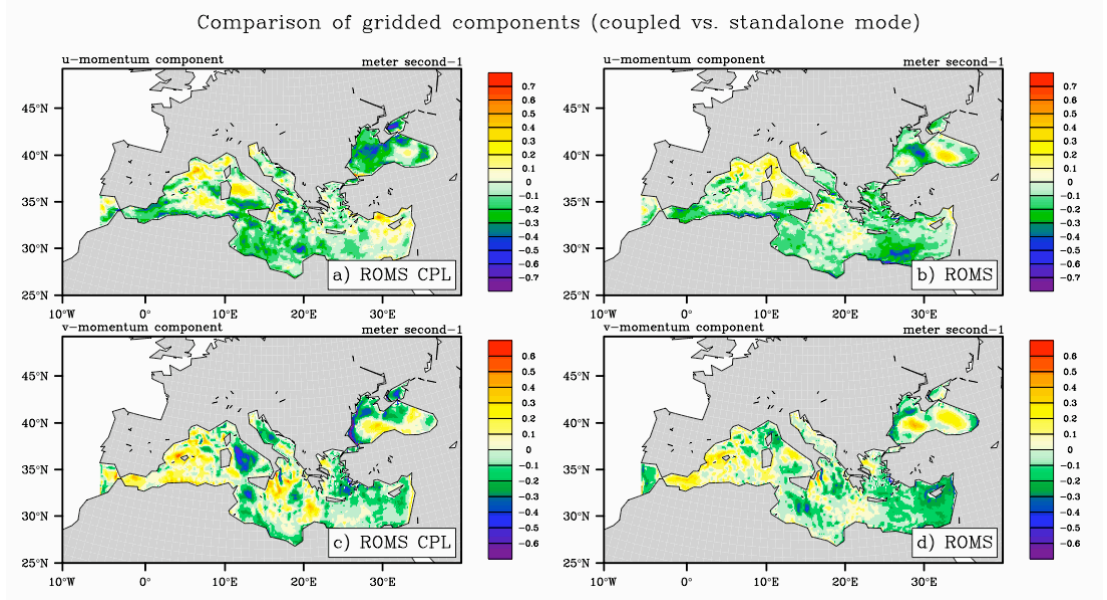
## EK D



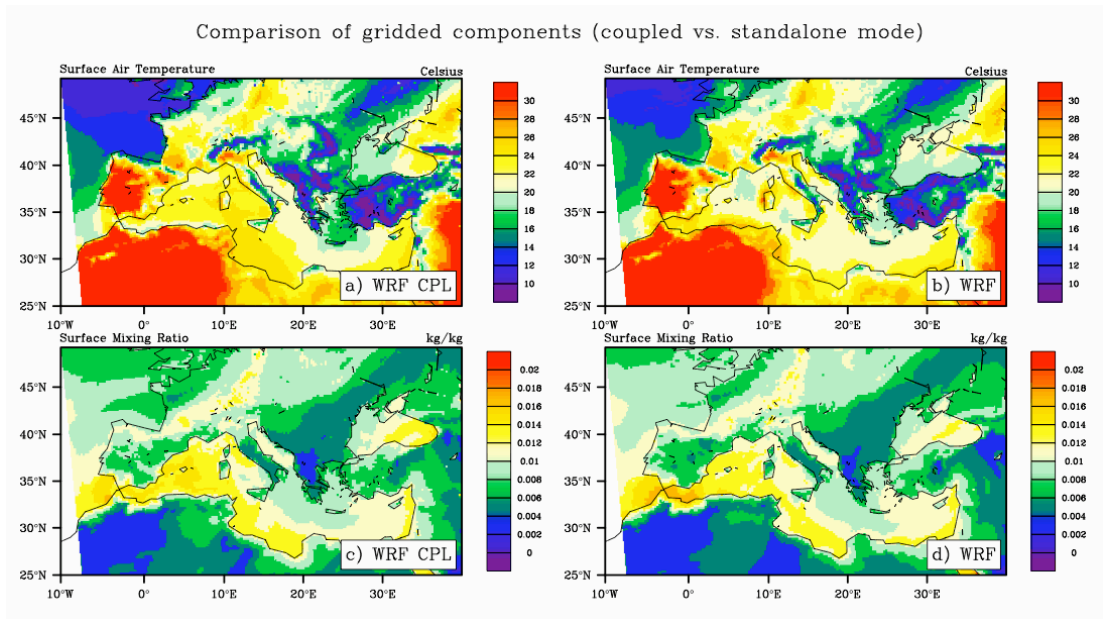
**Şekil D.1:** Kuple ve tek olarak çalıştırılmış modellerin deniz yüzeyi sıcaklığı karşılaştırmaları (01-07-1999 00:00). ROMS modeli (a ve b) için değerler en üst sigma seviyesinden alınmıştır.



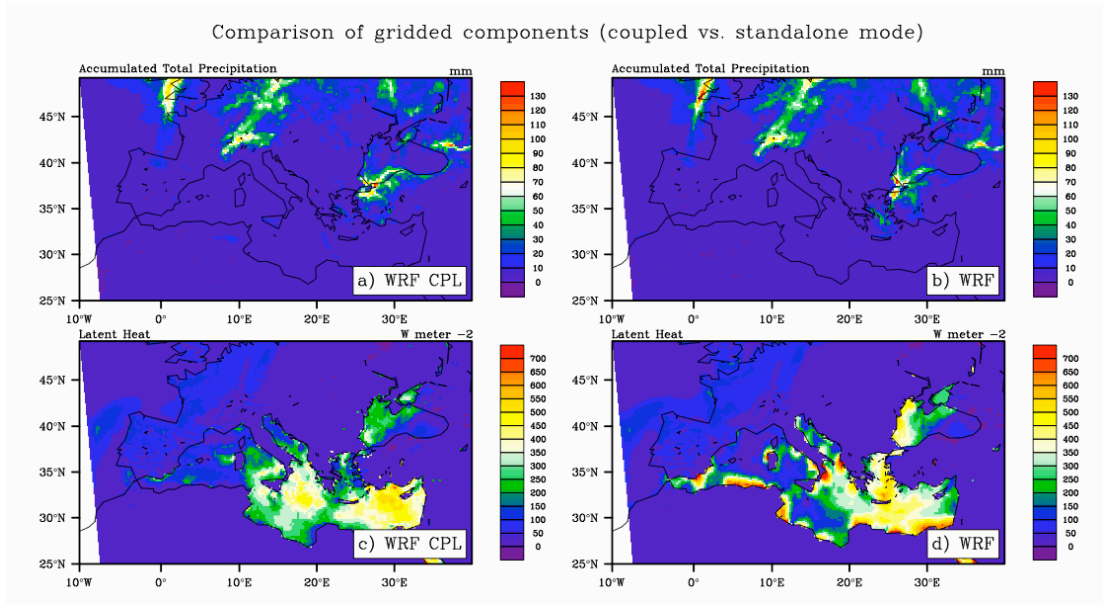
**Şekil D.2:** Kuple ve tek olarak çalıştırılmış modellerin deniz yüzeyi sıcaklığı karşılaştırmaları (15-07-1999 00:00). ROMS modeli (a ve b) için değerler en üst sigma seviyesinden alınmıştır.



**Şekil D.3:** Kuple ve tek olarak çalıştırılmış okyanus modelinin yüzey akıntıları karşılaştırması (15-07-1999 00:00).



**Şekil D.4:** Kuple ve tek olarak çalıştırılmış atmosfer modelinin 2 metre yüzey sıcaklığı ve karışım oranı karşılaştırmaları (15-07-1999 00:00).



**Şekil D.5:** Kuple ve tek olarak çalıştırılmış atmosfer modelinin yağış ve gizli ısı karşılaştırması (15-07-1999 00:00).

## ÖZGEÇMİŞ

Ufuk Utku Turunçođlu 1977'de Tokat'ta doğdu. Lisans derecesini 2000 yılında İstanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri Fakültesi, Meteoroloji Mühendisliği bölümünden aldı. Yüksek lisansını ise yine aynı üniversitenin Avrasya Yerbilimleri Enstitüsü, İklim ve Okyanus Anabilim Dalı, Yer Sistem Bilimi Programı'ndan 2003 yılında almış ve 2004 yılında İstanbul Teknik Üniversitesi bünyesinde bulunan Bilişim Enstitüsü, Hesaplamalı Bilim ve Mühendislik Programında doktora başlamıştır. Bu sırada Bilişim Enstitüsü, Yüksek Başarımlı Hesaplama Laboratuvarı ve Ulusal Yüksek Başarımlı Hesaplama Merkezinde sistem yöneticisi grubunun bir üyesi olarak çalıştı ve Ulusal Yüksek Başarımlı Hesaplama Merkezi'nin yılda iki kez düzenlediği paralel programlama eğitimlerinde eğitmen olarak görev aldı. 2008 yılında merkezi Amerika Birleşik Devletleri'nde bulunan NCAR (National Center for Atmospheric Research) Mesa Lab'ta CISL (Computational and Information Systems Laboratory) birimine bağlı ESMF (Earth System Modeling Framework) projesi grubunda doktora tezi ile ilgili bir yıl süre ile burslu araştırmacı olarak çalıştı. Kendisi halen İstanbul Teknik Üniversitesi, Bilişim Enstitüsü'nde Araştırma Görevlisi olarak görev yapmaktadır.