D4.1 Review of Applications, Users, Software and e-Infrastructures.

Project acronym: *MAPPER*

Project full title: Multiscale Applications on European e-Infrastructures.

Grant agreement no.: 261507

Due-Date:	M 6
Delivery:	M 6
Lead Partner:	UCL
Dissemination Level:	Public
Status:	Final
Approved:	Quality Board, Project Steering Group
Version:	0.9

DOCUMENT INFO

0.1 Authors

Date and version number	Author	Comments
11.04.2011 v0.9	Derek Groen, Alfons Hoekstra	Updated tables and executive summary.
31.03.2011 v0.8	Derek Groen, Joris Borgdorff, Alfons Hoekstra	Updated applications and revised summaries.
29.03.2011 v0.7	Derek Groen, Marian Bubak, Kasia Rycerz, Tomasz Pointek	Incorporated QA feedback, added additional tables to summary.
25.03.2011 v0.6	Derek Groen, Ilya Saverchenko	Updated e-infrastructures
23.03.2011 v0.5	Derek Groen, Peter Coveney, Krzysztof Kurowski	Incorporated external feedback, added information on NRENs
20.03.2011 v0.4	Derek Groen, Stefan Zasada	Added review of software
15.03.2011 v0.3	Derek Groen	Added references and further revised formatting.
10.03.2011 v0.2	Derek Groen	Formatting updates
06.03.2011 v0.1	Derek Groen, Stefan Zasada, Ilya Saverchenko, Joris Borgdorff, David Coster, Par Strand, Bastien Chopard, Alez Mizeranschi, Rupert Nash, James Suter, Krzysztof Kurowski, Mariusz Mamonski, Kasia Rycerz, Tomasz Pointek, Ilya Saverchenko	Initial version, converted from Wiki pages worked on since November 2010.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1 Executive Summary

In this deliverable we provide a comprehensive review of the applications within MAPPER, internal and potential external user communities, multiscale software tools and European production e-infrastructures. The content in this deliverable reflects the work carried out for Task 4.1 by the full MAPPER consortium. In this task, we not only review a number of major aspects in the MAPPER project, but also provide more in-depth information on several topics which relate directly to the future developments within MAPPER. Besides a review, this deliverable also contains preliminary specifications of applications and a consolidation of the fast track versus deep track components.

We concisely describe MAPPER's applications and compare and discuss their technical requirements in Chapter 3. Having formulated these requirements, we find the applications within MAPPER that require supercomputers to be run effectively. The information obtained from this review allows us to proceed with mapping the applications to appropriate e-infrastructures using the software solutions proposed in MAPPER.

In Chapter 4 we reflect on the multiscale user communities and mention a range of promising multiscale projects carried out by researchers worldwide. These include not only the communities present in MAPPER, but also a number of large communities that have clear multiscale computing needs, such as astrophysics, climate and environmental sciences.

In Chapter 5 we review the various components which are part of the MAPPER architecture. We distinguish between the fast-track components, which constitute easily installable solutions for loosely-coupled applications, as well as deep-track components, which require more time to deploy but are essential for supporting the more demanding and tightly-coupled applications.

In Chapter 6 we review the main production e-infrastructures within the EU, and analyse the gap between the functionalities currently provided on these infrastructures and those required by the applications in MAPPER. Each of these gaps can then be resolved by implementing, deploying and integrating parts of the MAPPER software framework into the existing software environment, although several missing functionalities also require modifications in the provision policies of resource providers (see D3.1).

2 Contributors

Although the lead partner for D4.1 is UCL, substantial contributions have been made by all partners in the consortium. The reviews on applications and users were collected and organized by Derek Groen from UCL, the review on software was collected and organized by Stefan Zasada from UCL, and the review on e-infrastructures was collected and organized by Ilya Saverchenko from LMU.

3 Review of Applications

This section provides a concise review of the application portfolio in MAPPER. This section contains a brief description of each application in MAPPER. Details, including extended tables with computational requirements, scale separation maps (SSMs) and Multiscale Modelling Language (MML) diagrams, can be found in Appendix A of this deliverable. An example of such a SSM can be found in Figure 1.

3.1 Simulation of Clay-polymer Composites (Nanomaterials)

In this area we study the formation and materials properties of clay-synthetic composite nanomaterials [2,4] and investigate (among other things) the origin of life by modelling the evolution of hammerhead RNA [1] on a clay surface. Within MAPPER we aim to develop a multiscale simulation mechanism that will, through its advances, allow the study and design of layered mineral composites in such diverse areas as energy applications (oil industry additives) and biomedical applications (drug delivery). The microscopic structure and mechanisms of layered nanomaterials operate over many different length scales, ranging from nanometers to microns, and each length scale needs to be properly simulated to fully understand its features [3]. We will use the MAPPER infrastructure, tools and software to couple three scale levels of simulation across distributed computing infrastructures. Combined with our scientific advances, this will facilitate the understanding of the underlying mechanisms of layered nanominerals on both the atomic and much larger scales.

3.2 In-stent Restenosis 3D (Physiology)

The three-dimensional In-stent Restenosis model (ISR3D) simulates a stent deployment to restore blood flow in coronary arteries and the subsequent processes associated with this procedure. The objective of the model is to study restenosis, a medical condition where the artery narrows some time after an initial stent has been placed. A number of publications provide background on ISR3D, including an extended multiscale model [5], detailed descriptions of the implementation of the single scale submodels [6-8] and reports on two-dimensional simulations run using the code [9]. The ISR3D model consists of four submodels: stent deployment and thrombus (blood clot) formation combined as initial conditions (IC), blood flow (BF), drug diffusion (DD) and smooth muscle cell proliferation (SMC). First, IC initializes the model by placing a stent in an artery and it calculates where thrombus should be formed given the blood circulation. These initial conditions are sent to SMC and for each iteration of SMC, DD and BF are calculated in parallel. For performance reasons BF keeps track of its last state, simplifying subsequent flow calculations.

3.3 Equilibrium Stability Workflow (Fusion)

The equilibrium stability workflow [10] application is one of the scenarios used to simulate aspects of nuclear fusion processes. The equilibrium stability workflow consists of two subcodes: a magnetohydrodynamics (MHD) equilibrium code (HELENA) and a a linear MHD stability code (ILSA). Although listed as one application, several alternative workflows are possible which vary the profiles from the equilibrium code, recalculate the equilibrium for each case and then calculate the MHD stability. The equilibrium stability workflow application is a loosely coupled workflow where the data can be exchanged via files or via structured objects defined by the EFDA Integrated Tokamak Modeling Task Force [17]. Variants of the workflow can be defined which add additional components and then require multiple instances of the equilibrium and stability calculation modules.

3.4 Transport Turbulence Equilibrium (Fusion)

The transport turbulence equilibrium application [11] [53] is a simplified and approximate version of a simulation of the full fusion core in a nuclear fusion reactor. The three main subcodes involved in transport turbulence equilibrium are:

- HELENA: 2D equilibrium solver (elliptic, no explicit time, but equilibrium time dependent).
  - GEM: 3D gyrofluid turbulence code, calculates transport coefficients.
  - ETS: 1D transport code, calculates new profiles.

Both for HELENA and GEM a number of modules could be substituted, with differing trade-offs of speed and accuracy/complexity). There are also some simple service modules in addition to these physics modules.

3.5 HemeLB (Physiology)

In this MAPPER application we are concerned with performing blood flow simulations of vessels in the brain in support of clinical neurosurgery. The behaviour of this blood flow plays a crucial role in the understanding, diagnosis and treatment of cardiovascular disease; problems are often due to anomalous flow behaviour in the neighbourhood of bifurcations and aneurysms within the brain. Simulation offers the possibility of performing patient-specific, virtual experiments to study the effects of courses of treatment with no danger to the patient. For this work, we will use our lattice-Boltzmann code, HemeLB [12] [13], designed to simulate fluid flow in the sparse topologies of the human brain. The code can create visualizations from within a running simulation and send them to a viewing client on a workstation situated in, ultimately, a hospital. A clinician can then steer the parameters of the simulation while viewing the results. Away from the region of direct clinical interest, we require less accuracy in our hydrodynamical simulation and can therefore use a slightly more approximate but much faster method. Still further away, the rest of the circulatory system can be abstracted to a network model of the vasculature and a pump, i.e. the heart.

3.6 Irigation Canals (Hydrology)

In a recent collaboration with ESISAR at Grenoble INP, France, UNIGE has developed multiscale models for the management of a network of irrigation canals. The problem remaining to be solved is the definition of appropriate actions (e.g. opening and closing gates) that need to be taken to always guarantee an adequate water supply throughout the canal system, whatever the external demands or perturbations may be, and with respect to constraints such as water height [33,34]. We have identified four main sub-models in our application, where each sub-model can be instantiated several times inside the global model. The Complex Automata (CxA) formalism [14] based on the lattice Boltzmann approach [15] is used for the implementation of these submodels.

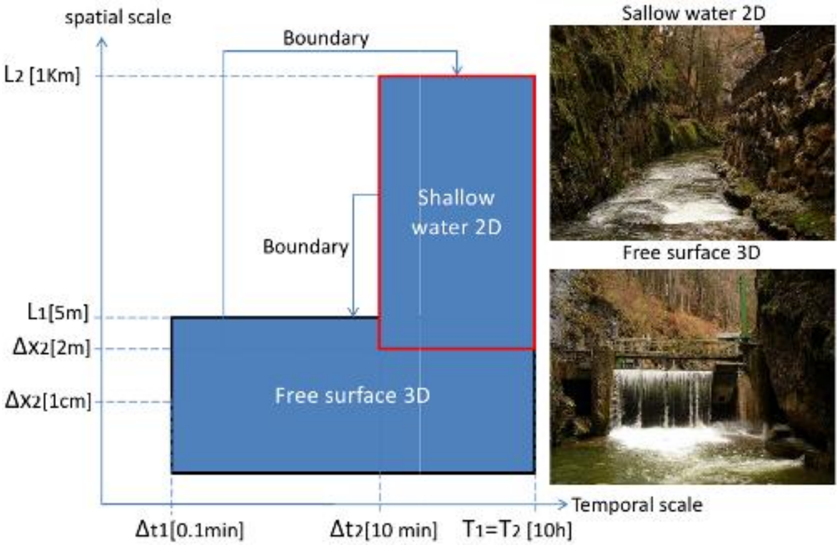


Figure 1: An example scale separation map (in this case of the irrigation canal application). Scale separation maps of each application are prvided in Appendix A.

3.7 Bile Acid and Xenobiotic System (Computational Biology)

The bile acid and xenobiotic system (BAXS) [15] defines an intricate physiological network that detoxifies and removes harmful xenobiotic and endobiotic compounds from the body while ensuring that primary bile acids (essential for the emulsification and absorption of dietary fats and fat-soluble vitamins) are not eliminated and can be re-used. The results generated by using BAXS will help us to understand a range of physiological processes such as drug-drug interactions, intracrine hormone metabolism, xenobiotic clearance and cholesterol/lipid homeostasis. The BAXS involves the coordinated activities of many genes across multiple temporal and spatial scales. Basic BAXS processes and their time scales include the binding of ligands to nuclear receptors (hours), gene expression and regulation (hours), transporter protein (minutes) and metabolic enzyme activity (seconds). Spatially, BAXS components range from molecules (e.g., nuclear receptors) to organs (e.g., the liver). A comprehensive description of the interacting components that govern BAXS gene expression would enable the identification of regulatory “nodes” as targets for treatment regimes, and understanding of the components impacting drug-drug interactions, and provide a framework for the design of large-scale, integrated prediction studies.

3.8 Summary of Technical Requirements

In this section we summarize the computational requirements of the applications within MAPPER. An overview of the coupling characteristics between the single-scale submodels is given in Table 1. The applications cover a wide range of coupling models, with 2 applications using loosely-coupled submodels, 2 applications using tightly-coupled submodels and another 3 applications using both. In addition, four of the applications have submodels which are run concurrently during the simulation. Two of the MAPPER applications (nanomaterials and equilibrium stability workflow) are often run using multiple instances to improve the statistical accuracy. Overall, the 7 exemplar applications in MAPPER are highly diverse in their coupling characteristics.

We provide a summary of the current computational requirements in Table 2 and a summary of the expected requirements in Table 3. Here we observe that the requirements are expected to increase as the project progresses. The MAPPER applications will require about 10-100 times more compute cores, and the size range of the data exchanged in coupled simulations is expected to grow from 0.001-300 MB to 1-1024 MB over the next two years. We do not foresee major changes in the overall completion time and the memory usage per core for these applications. Note that the equilibrium stability workflow is expected to run with up to 121 instances of the same simulation.

Application	# of subcodes	# of instances	Coupling method	Concurrent	Comments
Nanocomposites	3	1 to ~75	L and T	no	
Restenosis	4	1	L and T	yes	
Eq. Stability	2	~121	L	no	Variants possible
Transport Turb.	3+	1	T	yes	Variants possible
HemeLB	3	1	T	yes	Steerable
Irrig. Canals	4	1	L and T	yes	
BAXS	4	1	L	no	

Table 1: Coupling characteristics of the MAPPER application. In the fourth column “l” stands for loosely-coupled and “T” for tightly-coupled.

Application	# of cores	Memory (per core)	Runtime	Coupling frequency	Data size (coupling)	Data size (output)
		[MB]	**[hours]**		**[MB]**	**[GB]**
Nanocomposites	256	200	2 to 6	1/run	1	1
Restenosis	1024	200	2400	3/hour	50	0.05
Eq. Stability	1	3072	~29	2/hour	300	10
Transport Turb.	512	2048	0.5	2/hour	20	10
HemeLB	100	100	0.5	10/sec	<0.01	1
Irrig. Canals	1	100	1	1/hour	1	1
BAXS*	100	10	3	1/min	1	0.001

Table 2: List of computational characteristics of the MAPPER applications in their current form. The numbers for BAXS are estimates.

Application	# of cores	Memory (per core)	Runtime	Coupling frequency	Data size (coupling)	Data size (output)
		[MB]	**[hours]**		**[MB]**	**[GB]**
Nanocomposites	8192	200	2 to 6	1/hour	4	4
Restenosis	10000	100	120	1/min	5120	5
Eq. Stability	128	3072	1.5	4/hour	10240	1000
Transport Turb.	65536	2048	0.2	10/hour	20	1000
HemeLB	20000	100	0.5	10/sec	0.1	10
Irrig. Canals	1000	100	1	10/min	100	100

BAXS	200	10	3	1/min	1	0.001

Table 3: List of required capabilities for running the MAPPER applications as distributed multiscale production simulations.

4 Review of Users

Multiscale and multiscience applications are present in a wide range of scientific and engineering communities. By its nature, multiscale modelling is highly interdisciplinary, with developments occurring independently across fields. In this section we briefly describe several scientific communities which actively seek to develop and apply multiscale simulations, supported by tailored computational resources. This list is obviously not exhaustive. Its goal is principally to illustrate the different flavours of the multiscale approaches and the benefit of developing of a common and reusable methodology. More extensive descriptions of the scientific communities can be found in Appendix B.

Aside from the numerous publications, project websites and domain-specific reviews available, we also identified a few sources of information on multiscale simulations in multiple scientific domains. One source of information about the other communities is the *Journal of Multiscale Modelling and Simulation (MMS)* which is an interdisciplinary journal that focuses on the fundamental modelling and computational principles underlying various multiscale methods". Another relevant source, targeted at the engineering disciplines, is the International Journal for Multiscale Engineering. In addition, there are a series of multiscale modelling workshops, co-chaired by the MAPPER project director [16] in the ICCS conference, where a wide variety of multiscale simulations from different domains have been presented over the years.

4.1 Fusion

Within the EU between 100 and 200 hundred people are involved with modelling of present fusion devices and preparing for modelling of ITER [47] and DEMO. Similar sized communities exist in the USA, with smaller communities in the other ITER partners. Much of the effort within Europe is centered around the EFDA Task Force on Integrated Tokamak Modelling [17].

Example applications: The Equilibrium Stability Workflow application and the Transport Turbulence Equilibrium application within MAPPER are example multiscale applications from the field of Nuclear Fusion.

4.2 Systems Biology

Generally, modelling and simulation of complex (biological) systems involves compute-intensive (and sometimes data-intensive) techniques, for example, for model reverse-engineering or model identification, model validation, sensitivity analysis, etc. For single-scale models, many techniques exist to address these tasks. In contrast to single-scale approaches, multiscale modelling and simulation (MMS) typically involves a collection of (possibly heterogeneous) scale-specific models that together represent the model or simulation. A key challenge in MMS is to find effective and efficient ways for the individual scale-specific models to interoperate. Because of the heterogeneous nature and the need of interoperation of individual MMS components, the computational requirements are considerably more complex than those for single-scale approaches. The paper “Multi-scale modelling and simulation in systems biology” by Joseph O. Dada and Pedro Mendes [18] contains an overview of several large projects in systems biology. In addition, we cover a number of research groups and projects in the Annex.

Example Application: The Bile Acid and Xenobiotic System application within MAPPER is an example multiscale application from this field.

4.3 Astrophysics

Astrophysics is an extensive scientific domain, with a multitude of active multiscale projects. Due to the inherent multiphysics nature of many astrophysical problems, a large number of astrophysical simulations already operate across multiple time and length scales. More recently there has been a movement from large monolithic codes towards more modular frameworks that combine multiple community codes or modules, such as AMUSE and FLASH. Here we highlight these two frameworks, while we also review a number of other projects in Appendix B.

Perhaps the largest and most versatile multiscale framework currently in astrophysics is the Astrophysical Multipurpose Software Environment (AMUSE [19]). The aim of AMUSE, as a successor of the MUSE framework [20], is to provide a software framework for multiscale astrophysical simulations. Within this framework, existing (parallel) codes are connected through a layer of Python and MPI to form coupled simulations. There are numerous astrophysical scenarios in which AMUSE has been applied, e.g. the coupling of a gravitational N-body simulation with a stellar evolution code to model both the dynamical movements and the aging of the stars in a star cluster [20].

The FLASH code [21] couples hydrodynamics with magnetic fields and is applied to simulate the surfaces of compact stars such as white dwarves and neutron stars. The latest version, FLASH 3, consists of interoperable modules that can be combined to generate different applications. The FLASH architecture allows an arbitrary number of alternative implementations of its components to co-exist and be interchanged with each other, which results in greater flexibility. Further, a simple and elegant mechanism exists for customization of code functionality without the need to modify the core implementation of the component.

Example application: We provide a complete review of an external multiscale application in the Appendix B.3.6 of this deliverable. This application simulates a virtual galaxy (see [22]) using gravitational solvers, stellar evolution codes and solvers to model star formation in molecular clouds.

4.4 Engineering and Material Sciences

Many engineering and material sciences applications are inherently multiscale. An active area of research is linking our understanding of the physical world at very short (atomic, molecular) length and time scales with the observable behaviour at the macroscale. The applications are extremely varied; examples include composite and polymer materials, crystalline microstructures, nanomaterials, opto electronics for the development of materials with particular optical, magnetic and electrical properties, fracture propagation and flow through porous media, among many others.

The *International Journal of Multiscale Engineering* [23] is dedicated to multiscale simulation, and several universities have set up centres for multiscale modelling and simulation in the engineering domain. A general review of multiscale simulation in science in engineering can be found in [24]. In Appendix B we present a number of example projects currently developing multiscale coupling schemes. Many of these projects feature nanoscales and/or nanotechnology. Nanotechnology is the creation and utilization of functional materials, devices, and systems with novel properties and functions that are achieved through the control of matter, atom by atom, molecule by molecule or at the macromolecular level. In a sense this represents the ultimate multiscale engineering field by virtue of the enormous range of scales involved.

Example application: The previously discussed nanomaterials application within MAPPER is a good example of a multiscale application in the engineering and material sciences domain.

4.5 Biomedical Sciences

Biological systems span many orders of magnitude through different length and time scales in a continuous way, from the smallest microscopic scales up to the whole body. The sequence from the genome, proteome, metabolome, physiome to population health comprises multiscale, multi-science systems [25-27]. A pedagogical introduction to this concept and the field of multiscale modelling in biology is provided by Schnell et al. [28]. An extensive prior review of multiscale modelling in computational biomedicine has been published by Sloot et al. [29]. The full description in the Annex is largely based on that review. Here we will briefly reflect on a few key projects.

Virtual Physiological Human

The Virtual Physiological Human (VPH) Initiative is a large and active community in the multiscale computing domain, which received substaintial support through the EU 7th framework program through ICT-VPH. Within the VPH, which is also part of MAPPER, and its subprojects, multiscale simulations and models are of fundamental importance. However, not all subprojects directly couple different time and space scales in their models. The VPH is represented in Europe (VPH-NoE, http://www.vph-noe.eu), USA (Physiome, http://www.physiome.org), and Japan (e.g. the RIKEN group of Himeno, http://www.riken.jp/engn/r-world/research/lab/rpcs/next/index.html). Some of the subprojects within VPH are:

- euHeart (http://www.euheart.eu/) - heart modelling
- VPHOP (http://www.vphop.org/) – osteoporosis
- RTreat (http://artreat.org/) – atheroscleroris
- VPH2 (http://www.vph2.eu/) - heart modelling
- ContraCancrum (http://www.contracancrum.eu) – modelling of malignant tumours

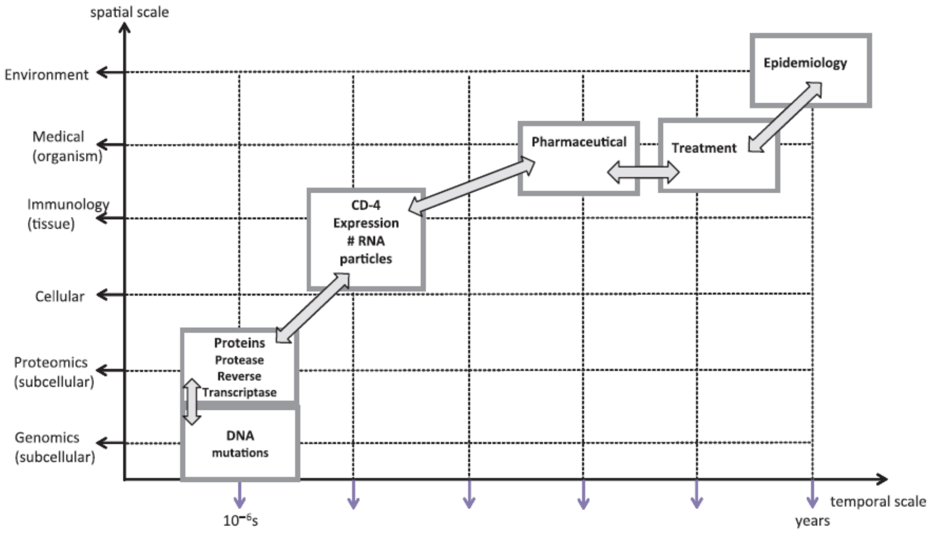


Figure 2: A scale separation map of a sumilation of resitance transmission in HIV (source: Sloot et al. [29]).

ViroLab: Transmission of Resistance in HIV

ViroLab (http://www.virolab.org/) [30-32] was developed to give medical doctors a decision support system to rank drugs targeted at patients and to provide virologists an advanced environment to study trends on an individual, population and epidemiological level. Virolab is a multiscale modelling, simulation and datamining environment for infectious diseases, going from molecule to man and back. Statistical and immunological models are needed to study the dynamics of the HIV populations and molecular dynamics models to study drug-virus binding affinities, in addition to rule-based and parameter-based decision support. All these models operate on a large range of length and time scales as sketched in the scale separation map above.

Example application: The In-stent restenosis and HemeLB applications within MAPPER are examples which originate from the biomedical community.

4.6 Environmental Sciences

Environmental science is multiscale and multi-process in its physical essence. The modelling of interdependent phenomena which operate at different time and length scales is a great scientific and computational challende. Example areas of science and application include ecology studies, climate modelling, geosciences and hydrology.

Hydrology

Several multiscale projects have been undertaken in the hydrology domain. Within the MAPPER project, UNIGE has developed multiscale models for management of a network of irrigation canals. Other multiscale projects include simulating the effects of beach height on tsunamis [35], modelling flood predictions [36], and simulations of watersheds [37].

Earth Modelling (ENES)

The European Network for Earth System Modelling (ENES, [38]) is a large consortium focused on developing of a European network for Earth system (multiscale) modelling. The ENES consortium has helped to establish the ENSEMBLES [39] project, where a consortium of 70 partners use multiscale ensemble simulations, varying across temporal and spatial scales, to simulate the Earth system for a range of applications. These include climate predictions, as well as the resolution of physical, chemical, biological and human-related feedback processes in the Earth System. This system includes water resource use, land use, air quality issues, as well as carbon cycle feedbacks.

Example application: The hydrology simulation of irrigation canals within MAPPER is a clear example of a multiscale application within the Environmental Sciences.

5 Review of Software

This section provides a review of software components which support multiscale simulations. Herein we describe only those components which will we plan to use in the MAPPER infrastructure. However, the full review in the Appendix C of this deliverable also describes several related software tools and popular middleware platforms. These components are arranged in a hierarchical infrastructure, shown in figure 3. The components are further categories into fast and deep tracks (shown in table 3), where fast track components which facilitate basic multiscale applications are deployed before the deep track components which represent the full system infrastructure, required to execute all application scenarios.

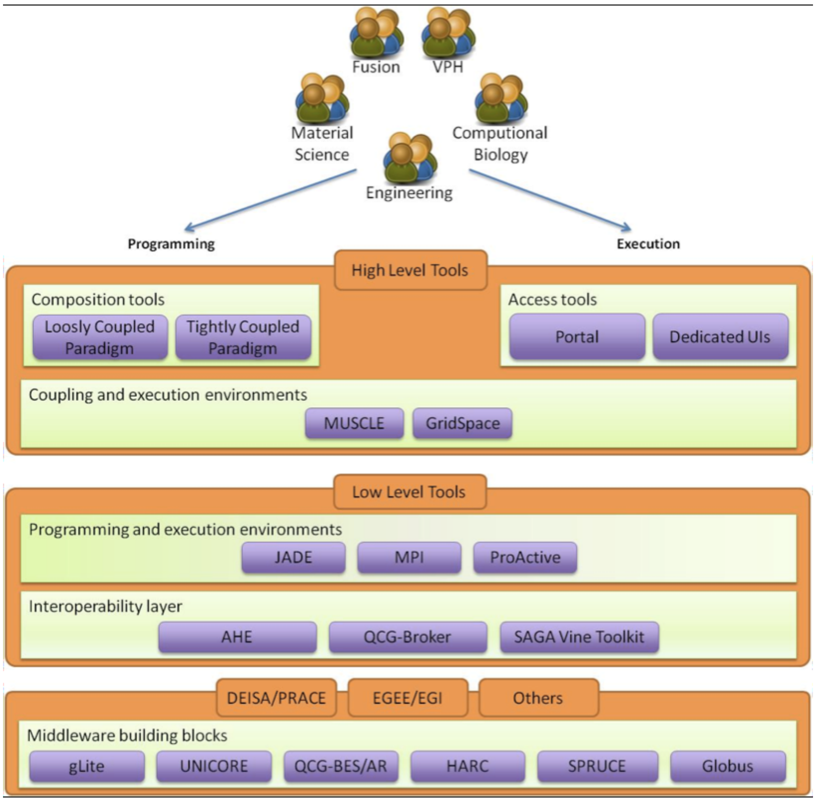


Figure 3: Overview of the MAPPER software infrastructure.

5.1 MPI

Message Passing Interface (MPI) is an API specification that allows processes to communicate with one another by sending and receiving messages. It is typically used for parallel programs running on computer clusters and supercomputers. The MPI interface provides virtual topology, synchronization, and communication functionality between a set of processes in a language-independent way, with language-specific bindings. It is the standard model for parallel software development on most grid and HPC machines, and has several implementations which support execution across sites (e.g., OpenMPI [50] and MPIg [52]).

Fast track components	**Deep track components**
MPI	MUSCLE
AHE	MPWide
GridSpace2	HARC
RealityGrid Steering	SAGA
QosCosGrid Toolkit	Vine Toolkit
	SPRUCE

Table 4: List of MAPPER fast and deep track components.

5.2 Application Hosting Environment

The Application Hosting Environment (AHE) provides simple desktop and command line interfaces to run applications on resources provided by national and international grids, in addition to local departmental and institutional clusters, while hiding from the user the details of the underlying middleware in use by the grid. In addition, a mobile interface for Windows Mobile based PDAs is available, and an iPhone interface is in development. The AHE is able to run applications on both UNICORE and Globus grids, meaning that a user can use a single AHE installation to access resources from the UK NGS and DEISA for example. The AHE is designed to allow scientists to quickly and easily run unmodified, legacy applications on grid resources, manage file transfers and monitor the status of the application. The philosophy of the AHE is based on the fact that very often a group of researchers will all want to access the same application, but not all of them will possess the skill or inclination to install the application on remote grid resources. In the AHE, an expert user installs the application and configures the AHE server, so that all participating users can share the same application.

5.3 GridSpace 2

GridSpace 2 [40] is an evolution of the computing and data access platform for viral disease research originally implemented in the ViroLab project. The GridSpace 2 Experiment Workbench, a top-level user interface, is designed to suit the requirements of domain scientists and allow them to exploit distributed computing platforms, including PL-Grid [41]. GridSpace 2 is based on the notion of exploratory programming where each experiment can be decomposed into a number of "*snippets*". Each snippet may be written in a different programming language; moreover, the Workbench enables users to execute entire workflows or just selected snippets. In this way, time-consuming experiments do not have to be started from scratch each time a modification is made during development. The web portal of the Experiment Workbench assists users in iteratively developing simulation workflows with the use of popular scripting languages. The Experiment Execution Environment, which is a layer underneath the Experiment Workbench, evaluates snippets and executes them on remote sites when needed.

5.4 RealityGrid Steering

The RealityGrid steering toolkit is a middleware tool and associated libraries that allows parameters in simulation codes to be marked as steerable, which means that they can be monitored, checkpointed and modified in real time as the simulation is running, and hence the course of the simulation changed. This is useful for a number of reasons: a scientist can monitor the progress of a simulation as it is running, and either stop it or alter the simulation parameters if a problem occurs, thus preventing valuable compute time from being wasted. The RealityGrid toolkit uses a layer of web services to allow users to interact with simulations. Simulation codes publish their steerable parameters to these services, and users use lightweight client desktop tools to check on and modify the state of parameters. Due to its consistent interface for the exchange of simulation parameters, the RealityGrid toolkit can also be used to couple models of different scale, using different simulation codes and potentially running on different computational resources. The HemeLB application in MAPPER is already integrated with RealityGrid Steering.

5.5 QosCosGrid Toolkit

QosCosGrid [49] is a multi-layer architecture which is able to deal with computationally intensive, large-scale, complex and parallel simulations that require multiple computational sites. QosCosGrid is able to combine resources from different administrative domains into a single powerful machine and is tightly integrated with commonly used tools for parallel and multiscale simulations (e.g., OpenMPI [50], ProActive [51] and more recently MUSCLE [42]). Supporting a wide range of development frameworks and programming models relevant for multiscale application developers, QosCosGrid gives the ability to work across heterogeneous computing sites and hides the complexity of underlying e-Infrastructures by simplifying many complex deployment, advance reservation and access procedures.

5.6 MUSCLE

The MUltiScale Coupling Library and Environment (MUSCLE) [42] provides the means to couple distinct computational models of multiple disciplines. Its aim is to provide a uniform platform to implement submodels using varying programming languages, execute them on heterogeneous machines and couple them across diverse networks. In addition, it considers the temporal and spatial scale at which the submodels are operating. It was developed during the COAST EU project and is partially based on the Complex Automata (CxA) formalism [43]. MUSCLE is implemented with Java for interoperability on heterogeneous hardware and its communication layer is built on the Java Agent DEvelopment framework (JADE) [44]. There are Java and C APIs inside the JADE submodel agent, allowing developers to easily use Java code or C, C++ or Fortran as a language of choice for model implementation. MUSCLE allows users to specify at execution time what submodel should run on which machine.

5.7 MPWide

MPWide [45] [46] is a communication library intended for message-passing between supercomputers. It was originally developed for the DEISA Gravitational Billion Body Project [48], where it was applied to run cosmological simulations across multiple supercomputers. The main added value of MPWide is its ability to explicitly define the communication method and configuration for individual network paths, and combine these customized configurations with an MPI-style API. These path-specific tunings are generally essential for obtaining good communication performance over wide area networks. Large simulations have been shown to run efficiently across heterogeneously configured wide area networks (which are quite common nowadays) using MPWide. The library is written in C++ and requires only support for POSIX threads (a default in many OS distributions).

5.8 HARC

A problem faced by users trying to exploit grid infrastructures in novel ways, for example by performing computational steering or real time visualization, is the fact that the user often has to wait until their simulation has progressed through the machine’s work queue and begun to run. The batch queue model employed by most computational resource providers gives users no guarantee of when their simulation will run. HARC, the Highly Available Robust Co-scheduler, developed at Louisiana State University, empowers users by allowing them to make reservations of time on compute, storage and network resources, guaranteeing when their simulation will run. If simulation is to be scheduled in to the clinical workflow this capability is essential, since a clinical user will need to schedule simulation around their clinical practice and surgical interventions. The ability to co-reserve is also essential when running coupled models across different computational resources.

5.9 SAGA

SAGA (the Simple API for Grid Applications) is a set of simple, coordinated and generic application programming interfaces (APIs) for accessing Grid services from generic application codes, portals, data managements systems. It is a standard developed by the Open Grid Forum (OGF) as a basis for a widely available and supported interface usable by application programmers enabling Grid functionality in their applications. SAGA lifts the burden of grid application programmers by providing them with a uniform interface to numerous types of grid middleware. As a result, grid application programmers need only learn a single API to obtain access to the entire grid, which simplifies the application programming for the Grid.

5.10 VINE

The Vine Toolkit is a modular, extensible Java library that offers developers an easy-to-use, high-level Application Programmer Interface (API) for grid-enabling applications. It supports a wide array of middleware and third-party services, including QosCosGrid, OGSA BES, gLite, Globus and Unicore. Using the Vine Toolkit, one composes applications as collections of resources and services for utilizing those resources. Vine offers security mechanisms for authenticating end-users and authorizing their use of resources within a given domain. Other core features include an extensible model for executing tasks (every action persists as a task) and transparent support for persistent information about resources and tasks with in-memory or external relational databases.

5.11 SPRUCE

The traditional high performance computing batch queue model does not allow for simulations to be prioritized by their urgency. Typically a grid will provide general purpose resources to a wide range of different users. If these resources are to be used by clinicians in support of their clinical practice, especially in support of emergency medical intervention planning, then some way is needed of prioritizing clinical simulations above the normal workload on a computational resource. SPRUCE, developed at Argonne National Labs, is a tool which allows this to happen. Clinicians and other users with simulations that are considered high priority are issued with SPRUCE tokens, which allows them to submit emergency jobs to a machine. The SPRUCE middleware takes care of running the job in a high priority mode, pre-empting the work that is already running/queued on the machine.

6 Review of e-Infrastructures

This section provides a review of the existing e-infrastructures. In the main document we focus on the three main EU infrastructures, namely DEISA, EGI and PRACE. However, we also provide a review of several other e-infrastructures and several network providers in Appendix 3.

6.1 PRACE

The Partnership for Advanced Computing in Europe, PRACE, is a persistent pan-European Research Infrastructure for Distributed High Performance Computing. PRACE maintains a HPC service consisting of the top of the line systems integrated into the European HPC ecosystem. Each system provides computing power of several PetaFLOPS and offers modern tools and services supporting a wide range of scientific applications.

6.1.1 Resources

At the moment PRACE combines two HPC systems that are located in Jülich, Germany and Essonne, France. The first system, JUGENE (at FZJ), is an IBM BlueGene/P machine providing more than 800 TeraFLOPS of computing power. The second system, Curie (at CEA), is expected to enter production late 2011 providing more than 1.5 PetaFLOPS of computing power. The number of PRACE systems is expected to increase in the next few years, starting 2012 when several additional systems will be integrated in the e-infrastructure.

6.1.2 Services

PRACE provides users with state of the art services in the following areas:

- Network: PRACE operates a dedicated high speed 10Gbit network that connects all resources. The PRACE internal network is shared among all users and dedicated network paths can not be reserved by research groups working on the e-infrastructure.
- Data: PRACE offers services for data management, including data transfer, storage and archiving. For data transfer PRACE relies on the GridFTP service that delivers up to 1GByte transfer speed.
- Compute: PRACE implements a unified interface to compute resources offering functionality for job submission and management. Unfortunately services MAPPER relies on such as advance reservation and co-allocation are not offered by the e-infrastructure.
- AAA (authentication, authorization, accounting): PRACE relies on PKI (Public Key Infrastructure) and X.509 certificates for user authentication. Each user group/project is assigned a fixed budget for computing resources. PRACE offers user services for accessing detailed information about the allocated and used budgets.
- User services: PRACE implements several technologies (e.g., Modules) to improve the user experience on all resources. PRACE also offers extensive support services, including documentation, help desk, application support, tutorials and training.

6.2 DEISA

DEISA, the Distributed European Infrastructure for Supercomputing Applications, is a consortium of leading national supercomputing centres that deploys and operates a persistent, production quality, distributed supercomputing environment with continental scope. By extending the European collaborative environment in the area of supercomputing, DEISA is paving the way towards the deployment and operation of a persistent cooperative European HPC ecosystem, as suggested by ESFRI.

6.2.1 Resources

DEISA federates 18 national HPC systems of various architectures located across Europe (France, Finland, Germany, Italy, the Netherlands, Spain, Sweden, Switzerland, UK). The combined performance of all systems exceeded 1 PetaFLOPS. The machines are available to European scientists and, due to their unique characteristics, satisfy requirements of all scientific domains.

6.2.2 Services

Services provided by DEISA are to a large extent similar to those offered in PRACE. Although there are some differences, limitations of both e-infrastructures from the MAPPER point of view are mostly identical.

- Network: The majority of DEISA resources are connected by a high speed 10Gbit network. In several cases, due to technical limitations, slower connections are offered.
- Data: DEISA offers services for data transfer and management, including a global GPFS-based file system, GridFTP protocol for file transfer, iRODS for data management. Additional services are offered by individual DEISA partners.
- Compute: DEISA supports a variety of interfaces to compute resources, including batch scheduling systems, UNICORE6 and Globus WS-GRAM. Advance reservation is not supported on compute resources but is supported on visualization systems.
- AAA (authentication, authorization, accounting): Like PRACE, DEISA relies on the PKI for authentication, unless technical limitations require otherwise.
- User services: Services offered by DEISA in this area are similar to PRACE.

6.3 EGI

EGI aims to provide European scientists and their international partners with a sustainable, reliable e-infrastructure that can support their needs for large-scale data analysis. This is essential in order to solve a number of big questions facing science today, and in the decades to come.

6.3.1 Resources

EGI brings together more than 350 resource providers in Europe and more than 550 worldwide. The majority of EGI resources are cluster systems optimized for capacity computing (i.e. large number of serial jobs or parallel jobs requiring several CPUs, for instance data processing) and not supporting parallel interfaces such as MPI.

6.3.2 Services

EGI resources are operated by National Grid Initiatives (NGIs) that have to conform to the general regulations and policies defined by EGI. Each NGI, however, can specify additional internal policies. EGI users are granted access to the e-infrastructure via virtual organizations (VO). Each VO can agree on internal policies and provide VO specific services.

- Network: EGI resources are connected via the public Internet network. Research groups and VOs can request dedicated channels from the local National Research and Education Networks (NRENs).
- Data: EGI resources have to provide several data services including GridFTP for data transfer.
- Compute: The majority of EGI resources support gLite. Other middlewares including Arc, Globus and UNICORE6 are supported on selected resources.
- AAA (authentication, authorization, accounting): Like DEISA and PRACE, EGI relies on PKI. EGI also supports pool accounts, which are shared by a group of people.
- User services: EGI offers help desk, training and other support services to all recognised user communities.

6.4 Gap Analysis

In this section we assess the gaps between several application requirements and the facilities provided by the PRACE, DEISA and EGI infrastructures. A full overview of technical requirements for multiscale simulations and the facilities provided by the e-infrastructures can be found in Table 5.

multiscale requirement	PRACE	DEISA	EGI	Relevant MAPPER tools
MPI support	yes	yes	no	OpenMPI
shared 1GB/s WAN	yes	yes	no	MUSCLE, MPWide
dedicated 1GB/s WAN	no	no	on request	HARC, MUSCLE, MPWide
advance reservation	no	no	no	QCG Toolkit, HARC
automated co-scheduling	no	no	no	QCG, OpenMPI, HARC
resource brokering	no	no	yes	QCG, AHE, SAGA
workflow deployment	no*	no*	no	GridSpace2, VINE, RG Steer
urgent computing	no	no	no	SPRUCE
allocation monitoring	yes	yes	no	-

Table 5: List of requirements of multiscale applications and whether these are met by the three main EU infrastructures on an infrastructure-wide level PRACE and DEISA do support UNICORE-specific workflow deployment tools.

We identify a number of gaps between the facilities required by MAPPER and those offered by production e-infrastructures. These include MPI support (EGI), advance reservation and co-scheduling facilities (all), resource brokering (PRACE,DEISA), workflow deployment (all), urgent computing (all) and user-specific monitoring of compute allocations (primarily EGI). The software tools in MAPPER are able to provide the functionalities which currently are missing (see the rightmost column in Table 5). These tools are already being deployed on several infrastructures, a process which is reported on in detail in D5.1.

Providing appropriate software tools alone will not suffice to mend the gap between the requirements of multiscale applications and the facilities provided by production e-infrastructures. This is because users require a usable environment where they can operate their simulations without being confronted with the complexity of the underlying software tools and their interaction. This can only be accomplished when the underlying software tools are well integrated and interoperate seamlessly with each other and the base e-infrastructure software environment. We present the MAPPER software architecture in D8.1, which aims to provide a tightly-integrated software framework for conveniently composing, scheduling and executing multiscale simulations.

7 Conclusions

In this deliverable we reviewed the applications within MAPPER, a number of scientific communities which use multiscale simulations, a considerable range of multiscale software tools and the major e-infrastructures which support such simulations. The MAPPER applications are diverse, spanning five scientific domains, each having different infrastructural and coupling requirements. In our review of users and user communities we find a similar diversity in multiscale simulations; there are a large number of projects from a wide range of communities both in the loosely-coupled and the tightly-coupled simulation domain.

As a result, we will require a software infrastructure capable of handling both loosely-coupled and tightly-coupled workflows, as well as submodels where different single-scale submodels are expected to run concurrently. In addition we expect the computational demands of the applications to increase to the supercomputing scale within the next two years, if they do not require such resources already.

The European PRACE, DEISA and EGI production infrastructures are able to meet the raw compute power, storage space and network bandwidth requirements of the multiscale applications, but will require additional software tools to provide the technical functionalities (e.g., advance reservation and workflow support) needed to run both loosely and tightly coupled multiscale simulations. MAPPER provides a range of software tools which provide these technical functionalities, as well as a political framework in support of multiscale applications (see D3.1). However, the deployment and integration of these individual components is essential to create a robust and usable environment for distributed multiscale simulations. These actions are already underway, and we report on the deployment status of the MAPPER software environment in D4.1, and provide a blueprint of the MAPPER software architecture in D8.1. Likewise, we are negiotiating with PRACE,DEISA and EGI to make the required policy adjustments for facilitating distributed multiscale simulations in EU production infrastructures.

8 References

[1] R. Anderson, H. Greenwell, J. Suter, P. Coveney and M. Thyveetil, "Determining materials properties of natural composites using molecular simulation", Journal of Materials Chemistry, 19, (39), 7251-7262, (2009), DOI: 10.1039/b909119j.

[2] J. Suter and P. Coveney, "Materials properties of clay nanocomposites: onset of negative Poisson ratio in large-scale molecular dynamics simulation", Soft Matter, 5, (20), 3896-3904, (2009), DOI: 10.1039/b907590a.

[3] J. Suter, R. Anderson, H. Greenwell, and P. Coveney, "Recent advances in large-scale atomistic and coarse-grained molecular dynamics simulation of clay minerals", Journal of Materials Chemistry, 19, (17), 2482-2493, (2009) DOI: 10.1039/b820445d.

[4] J. Suter, P. Coveney, "Computer simulation study of the materials properties of intercalated and exfoliated poly(ethylene)glycol clay nanocomposites", Soft Matter, 5, (11), 2239-2251, (2009), DOI: 10.1039/b822666k.

[5] D. Evans, P. Lawford, and J. Gunn. The application of multiscale modelling to the process of development and prevention of stenosis in a stented coronary artery. Phil. Trans. R. Soc. A, Jan 2008.

[6] A. Caiazzo, D. Evans, J.-L. Falcone, J. Hegewald, E. Lorenz, B. Stahl, D. Wang, J. Bernsdorf, B. Chopard, J. Gunn, R. Hose, M. Krafczyk, P. Law- ford, R. Smallwood, D. Walker, and A. Hoekstra. Towards a complex automata multiscale model of in-stent restenosis. ICCS 2009, Jan 2009.

[7] A. Hoekstra, J.-L. Falcone, A. Caiazzo, and B. Chopard. Multi-scale modelling with cellular automata: the complex automata approach. Cellular Automata, pages 192–199, 2010.
[8] E. Lorenz. Multi-scale modelling with complex automata: In-stent restenosis and suspension flow. Aug 2010.
[9] H. Tahir, A. G. Hoekstra, E. Lorenz, P. Lawford, D. Hose, J. Gunn, and D. Evans. Multiscale simulations of the dynamics of in-stent restenosis: impact of stent deployment and design. Journal of the Royal Society Interface, 2011.
[10] G. Huysmans, J. Goedbloed and W. Kerner, CP90 Conference on Comp. Physics, Word Scientific Publ. Co. 1991, p.371.
[11] D. Coster, V. Basiuk, G. Pereverzev, D. Kalupin, R. Zagórksi, R. Stankiewicz, P. Huynh, and F. Imbeaux. The european transport solver. Plasma Science, IEEE Transactions on, 38(9):2085 -2092, 09 2010.

[12] M. Mazzeo, S. Manos and P. Coveney. In situ ray tracing and computational steering for interactive blood flow simulation. Computer Physics Communications (2010) vol. 181 pp. 355-370.

[13] M. Mazzeo and P. Coveney. HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries. Computer Physics Communications (2008) vol. 178 (12) pp. 894 – 914.

[14] A. Hoekstra, J.-L. Falcone, A. Caiazzo, and B. Chopard. Multi-scale modelling with cellular automata: The complex automata approach. In Hiroshi Umeo, Shin Morishita, Katsuhiro Nishinari, Toshihiko Komatsuzaki, and Stefania Bandini, editors, Cellular Automata, volume 5191 of Lecture Notes in Computer Science, pages 192-199. Springer Berlin / Heidelberg, 2008.

[15] N. Kennedy, W. Dubitzky, P. Thompson, and H. Zheng, Multi-Scale modelling of the Bile Acid and Xenobiotic System, Manuscript submitted for publication.

[16] ICCS Workshop Series: Simulation of Multiphysics Multiscale Systems (SMMS) – http://staff.science.uva.nl/~valeria/SMMS.
[17] EFDA Task Force on Integrated Tokamak Modelling (https://www.efda-itm.eu).
[18] J. Dada and P. Mendes, Multi-scale modelling and simulation in systems biology, December 2010 DOI: 10.1039/c0ib00075b.
[19] Astrophysical Multipurpose Software Environment(AMUSE) – http://www.amusecode.org.
[20] S. Portegies Zwart, S. McMillan, S. Harfst, D. Groen, M. Fujii, B. Ó Nualláin, E. Glebbeek, D. Heggie, J. Lombardi, P. Hut et al. 2008, A multiphysics and multiscale software

environment for modelling astrophysical systems, New Astronomy, Volume 14, Issue 4, p. 369-378.

[21] A. Dubeya, K. Antypasb, M. Ganapathyc, L. Reida, K. Rileyd, D. Sheelerd, A. Siegeld and K. Weidea, 2010, Extensible component-based architecture for FLASH, a massively parallel, multiphysics simulation code, Volume 35, Issues 10-11, Pages 512-522.

[22] A. Hoekstra, S. Portegies Zwart, M. Bubak, P. Sloot, 2007, Towards Distributed Petascale Computing, In: D. Bader (Ed.) Petascale, Computing: Algorithms and Applications, Chapman & Hall / CRC Press, Taylor and Francis Group.

[23] The International Journal of Multiscale Engineering – http://www.begellhouse.com/journals/61fd1b191cf7e96f.html.

[24] Jacob Fish. Multiscale Methods: Bridging the Scales in Science and Engineering. Oxford University Press, 2010.

[25] A. Finkelstein, J. Hetherington, L. Li, O. Margoninski, P. Saffrey, R. Seymour and A. Warner. Computational challenges of systems biology. IEEEComput 2004;37:26–33.

[26] D. Noble modelling the heart—from genes to cells to the whole organ. Science 2002;295:1678–82.

[27] J. Southern, J. Pitt-Francis, J. Whiteley, D. Stokeley, H. Kobashi, R. Nobes, Y. Kadooka, D. Gavaghan. Multi-scale computational modelling in biology and physiology. Progr BiophysMolecul Biol 2008;96:60–89.

[28] S. Schnell, R. Grima, P. Maini. Multiscale modelling in biology. American Scientist 2007;95:134–42.

[29] P. Sloot and A. Hoekstra, 2010, Multi-scale modelling in computational biomedicine, Briefings in Bioinformatics, volume 11, number 1, pages 142--152.

[30] P. Sloot, P. Coveney, M. Bubak, et al. Virolab: a collaborative decision support system in viral disease treatment. RevAntiviralTher 2008;3:4–7.

[31] P. Sloot. VIROLAB: from the molecule to the man. eStrategies Proj 2008;53–5.

[32] P. Sloot, A. Tirado-Ramos, I. Altintas, et al. From molecule to man: decision support in individualized E-health. IEEEComp 2006;39:40–6.

[33] P. van Thang, B. Chopard, L. Lefevre, D. Anda Ondo, and E. Mendes. Study of the 1d lattice boltzmann shallow water equation and its coupling to build a canal network. Journal of Computational Physics, 229(19):7373 -7400, 2010.

[34] O. Marcou, S. El Yacoubi, and B. Chopard. A bi-fluid lattice boltzmann model for water ow in an irrigation channel. In Samira El Yacoubi, Bastien Chopard, and Stefania Bandini,

editors, Cellular Automata, volume 4173 of Lecture Notes in Computer Science, pages 373-382. Springer Berlin / Heidelberg, 2006.

[35] C. Ana and Y. Cai. The effect of beach slope on the tsunami run-up induced by thrust fault earthquakes. Procedia Computer Science Volume 1, Issue 1, May 2010, Pages 645-654, ICCS 2010.

[36] B. Gouldby, V. Krzhizhanovskaya, and J. Simm. Multiscale modelling in real-time flood forecasting systems: From sand grain to dike failure and inundation. Procedia Computer Science Volume 1, Issue 1, May 2010, Page 809, ICCS 2010

[37] P. Ellera, J. Chenga, H. Nguyena and R. Maiera. Improving parallel performance of large-scale watershed simulations. Procedia Computer Science, Volume 1, Issue 1, May 2010, Pages 801-808 ICCS 2010.

[38] European Network for Earth System Modelling – http://www.enes.org.

[39] ENSEMBLES – http://ensembles-eu.metoffice.com/.

[40] GridSpace 2 – http://dice.cyfronet.pl/gridspace/.

[41] PL-Grid – http://www.plgrid.pl/en.

[42] MUSCLE http://muscle.berlios.de/.

[43] A. Hoekstra et al. Towards a complex automata framework for multiscale modelling: formalism and the scale separation map. ICCS 2007 (2007).

[44] JADE – http://jade.tilab.com.

[45] MPWide – http://castle.strw.leidenuniv.nl/software/mpwide.html.

[46] D. Groen, S. Rieder, P. Grosso, C. de Laat and S. Portegies Zwart. A lightweight communication library for distributed computing. Computational Science and Discovery Volume 3, Number 015002, 2010.

[47] ITER – http://www.iter.org.

[48] S. Portegies Zwart, T. Ishiyama, D. Groen, K. Nitadori, J. Makino, C. de Laat, S. McMillan, K. Hiraki, S. Harfst, P. Grosso. Simulating the Universe on an Intercontinental Grid, IEEE Computer, 2010, volume 43, issue: 8, pages 63 – 70.

[49] Kurowski, K., Piontek, T., Kopta, P., Mamoński, M., & Bosak, B. (2010). Parallel Large Scale Simulations in the PL-Grid Environment. Computational Methods in Science and Technology , 47-56.

[50] OpenMPI - http://www.openmpi.org

[51] ProActive – http://proactive.inria.fr.

[52] S. Manos, M. Mazzeo, O. Kenway, P. Coveney, N. Karonis, B. Toonen. Distributed mpi cross-site run performance using mpig. HPDC 2008, pages 229-230.

[53] C.F. Maggi, R.J. Groebner, C. Angioni, T. Hein, L.D. Horton, C. Konz, A.W. Leonard, C.C. Petty, A.C.C. Sips, P.B. Snyder, J. Candy, R.E. Waltz, ASDEX Upgrade and DIII-D Teams. Pedestal and core confinement of hybrid scenario in ASDEX Upgrade and DIII-D. Nuclear Fusion 50(2), 025023 (2010).

9 Appendix A: Extended review of Applications

9.1 Introduction

This appendix contains the complete reviews of the seven applications within MAPPER. The content in this appendix is derived from pages on the MAPPER wiki, where we actively maintain and update the content. Each entry contains a description, a scale separation map (which is a visual representation of the scales of a multiscale model and the interactions between the submodels), a multiscale modelling language (MML) definition of the application, a formulation of the current and expected computational requirements, as well as links to available code repositories and additional literature. As a convenience for the reader, we have placed the application-related literature references within the review sections, close to the other details of the application.

For more information on scale separation maps, please refer to:

A. G. Hoekstra, E. Lorenz, J.-L. Falcone, and B. Chopard. Towards a complex automata framework for multi-scale modeling: formalism and the scale separation map. ICCS 2007, Jan 2007.

For more information on the multiscale modelling langues, please refer to:

J.-L. Falcone, B. Chopard, and A. G. Hoekstra. MML: towards a multiscale modeling language. Procedia Computer Science, 1(1):819–826, 2010.

9.2 Simultaion of nanocomposite materials

9.2.1 Description of the overall application and its subcodes

For the nanomaterials application we aim to develop a multiscale simulation mechanism that will, through its advances, allow the study and design of layered mineral composites in areas having a substantial potential impact such as energy applications (oil industry additives), materials applications (nano composites materials) and biomedical applications (drug delivery). The microscopic structure and mechanisms of layered nanomaterials operate over many different length scales, ranging from nanometers to microns; each length scale needs to be properly simulated to fully understand their features. This knowledge will eventually lead to the design of novel layered mineral systems with properties tailored to their application.

Simulation on the atomic scale can be time consuming; simulations have to compute and track the position and motion of each individual atom. While this may be suitable for simulations of small molecules, when one wishes to simulate a system containing billions of atoms, the computer time required makes such a study unfeasible. Despite increases in computer power (as described by Moore's Law) and improved efficiency in algorithms, there are many situations where we need to simulate a very large system in atomistic detail, but cannot due to the associated computational overheads. An answer to such a bottleneck is to reduce the degrees of freedom in the system by combining several atoms together into a "pseudo"-particle, defined to possess the properties of the atoms it is composed of. By effectively removing the unimportant internal degrees of freedom, we can speed up simulation times by a factor of 100 or even 1000, yet still retain the required accuracy of a full atomistic simulation. These coarse-grained simulations can push atomistic techniques to the macro level, in terms of both length (micron instead of nanometer) and time (microsecond rather than nanosecond), allowing one to study previously unavailable phenomena, such as long-time diffusion and large-scale structure.

Such techniques are becoming more and more commonplace in the biological and polymer science domain. However, problems remain; firstly, how do you split up your atomic system into "pseudo"-particles, and, most importantly, how do you define their properties such that they reproduce those of the atoms that form them? Previous work has side-stepped these problems by creating systems with arbitrary parameters, such that trends could be addressed qualitatively, but no quantitative information can be easily be extracted.

Similarly, bond breaking and formation of atoms requires simulation of the electronic structure. Such techniques are computationally very expensive, yet combining them into a multiscale simulation will allow polymerization mechanisms in layered minerals to be studied and correct parameterisation of classical models to be performed. MAPPER infrastructure, tools and software will be used to transparently couple these three levels of simulation in a loosely coupled scheme across distributed computing infrastructures.

9.2.2 Description of the coupling mechanism between the codes

The nano-materials simulation is initially a loosely coupled scenario where the subcodes run one after another. Here, the simulation is initiated with quantum mechanical simulations, which are followed up by fine-grained molecular dynamics simulations, which in turn provide the input for course-grained molecular dynamics simulations. Several post-processing scripts are run between the simulation to perform data conversion. These scripts may be preprogrammed, but can also (in some cases) be modified by the user between simulations.

The benefit of coupling the codes is to produce a course-grained nanomaterials simulation which exhibits statistically accurate behavior on finer levels (e.g., on the resolution of clay-polymer interactions). The fine-grained and quantum mechanical simulations consist of multiple instances (typically one or two dozen), while we run a single instance of the course-grained simulation in the final stage of the simulation. Because the simulations do not run concurrently, and the frequency (and required performance) of data exchange between subcodes is limited we plan to use GridSpace here to perform the coupling between the submodels.

9.2.3 Scale Separation Map

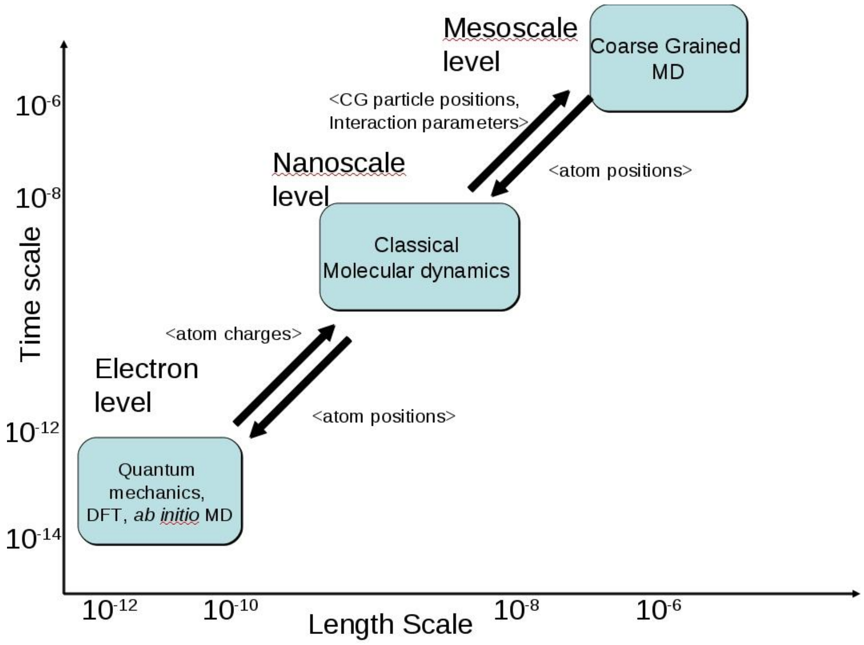


Figure 4: Scale separation map of the nanomaterials application.

9.2.4 Characterisation of the application coupling template

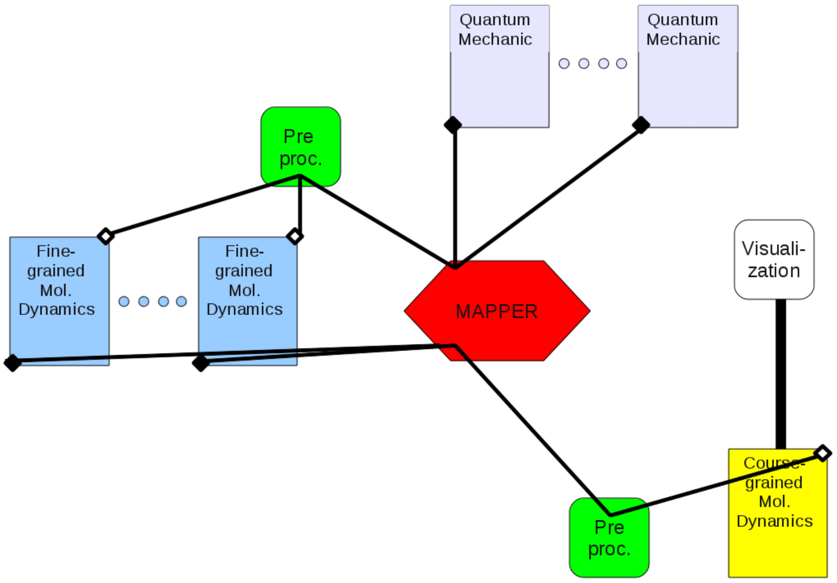

Quantum Mechanic
Quantum Mechanic
Pre proc.
Fine-grained Mol. Dynamics
Fine-grained Mol. Dynamics
MAPPER
Visuali-zation
Course-grained Mol. Dynamics
Pre proc.

Figure 5: Multiscale Modelling Language diagram of the nanomaterials application.

9.2.5 Computational requirements

9.2.5.1 Course grained molecular dynamics

Requirement	State of the art	Required capabilities
# of cores	256	8192
memory per core [MB]	200	200
runtime	2-6 hours	21 days
coupling frequency	1/run (at the start)	1/hour
max. data size (coupling)	1MB	32MB
max. data size (output)	1GB	32GB

9.2.5.2 Fine-grained molecular dynamics

Requirement	State of the art	Required capabilities
# of cores	256	8192
memory per core [MB]	200	200
runtime	2-6 hours	2-6 hours
coupling frequency	1/run (at the end)	1/hour

Requirement	State of the art	Required capabilities
max. data size (coupling)	1MB	32MB
max. data size (output)	1MB	32MB

9.2.5.3 Quantum mechanical simulation

Requirement	State of the art	Required capabilities
# of cores	16	64
memory per core [MB]	200	200
runtime	1 hour	1 hour
coupling frequency	1/run (at the end)	1/hour
max. data size (coupling)	1MB	32MB
max. data size (output)	1MB	32MB

9.2.6 Software Requirements

- Any MPI distribution.
- FFTW 2.1.5.

9.2.7 Other Characteristics

The final production runs are expected to run without user interaction. However, before that stage is reached the user will need to tune the data processing scripts such that an optimal configuration is obtained. This is an iterative process that will require scripts to be modified after simulations have been run, and where simulations possible have to be rerun to provide more accurate results. So in summary, this scenario does involve user interaction during development, testing and tuning, but could be performed without user interaction once the final simulation is run.

There is a LAMMPS version for GPUs available, but we will use the non-GPU version within MAPPER. As for accessing the simulation codes, CASTEP (http://www.castep.org) is available for free to UK researchers but must be purchased otherwise. Alternatively, CPMD (http://www.cpmd.org) may be used in place of CASTEP. In the case of fine grained and course grained molecular dynamics, we use LAMMPS (http://lammps.sandia.gov), of which both the binaries and the source code are freely available.

9.2.8 References to literature to the application:

[1] R. L. Anderson, H. C. Greenwell, J. L. Suter, P. V. Coveney and M. Thyveetil, "Determining materials properties of natural composites using molecular simulation", Journal of Materials Chemistry, 19, (39), 7251-7262, (2009), DOI: 10.1039/b909119j.

[2] J. L. Suter and P. V. Coveney, "Materials properties of clay nanocomposites: onset of negative Poisson ratio in large-scale molecular dynamics simulation", Soft Matter, 5, (20), 3896-3904, (2009), DOI: 10.1039/b907590a.

[3] J. L. Suter, R. L. Anderson, H. C. Greenwell, and P. V. Coveney, "Recent advances in large-scale atomistic and coarse-grained molecular dynamics simulation of clay minerals", Journal of Materials Chemistry, 19, (17), 2482-2493, (2009) DOI: 10.1039/b820445d.

[4] J. L. Suter, P. V. Coveney, "Computer simulation study of the materials properties of intercalated and exfoliated poly(ethylene)glycol clay nanocomposites", Soft Matter, 5, (11), 2239-2251, (2009), DOI: 10.1039/b822666k.

9.3 3D Simulation of in-stent restenosis

9.3.1 Description of the overall application and its submodels

The multiscale three-dimensional In-stent Restenosis model (ISR3D) allows 3D simulation of a stent deployment in a coronary artery and subsequent processes. The objective of the model is to help understand restenosis and to indicate improvements in stent design. An extended multiscale model, in terms of an SSM is described by Evans et al. [3]. A simplified version as well as a detailed description of the implementation of the submodels and the coupling with MUSLE is provided by Caiazzo et al. [2], Hoekstra et al. [7], Lorenz [8]. Some detailed two- dimensional simulations are reported by Tahir et al. [9]. Here, we will further describe the ISR3D model.

The ISR3D model consists of four submodels: stent deployment and thrombus forming combined as initial conditions (IC), blood flow (BF), drug diffusion (DD) and smooth muscle cell proliferation (SMC). First, IC initializes the model by placing a stent in an artery and it calculates where thrombus should be formed given the blood circulation. These initial conditions are sent to SMC and for each iteration of SMC, DD and BF are calculated in parallel. For performance reasons BF keeps track of its last state, simplifying subsequent flow calculations.

9.3.2 Description of the coupling mechanism between the submodels

The submodels of ISR3D listed above all have a single instance running, all throughout the simulation. However, they do their calculations at disjunct times, except BF and DD.

The IC submodel performs initialization and is loosely coupled to the rest of the model; it stops after sending the adjusted structural data to DD, BF and SMC. The model then enters a tightly coupled loop, alternating SMC with BF and DD, where the number of iterations of SMC determines the number of data transfers. The number of iterations of SMC is fixed, thus so is the number of synchronization points. Since the mainstay of the model is the tightly coupled part, we generally describe the ISR3D model as being tightly coupled. The submodels are coupled using the MUSCLE [1], which uses the Java Agent DEvelopment framework (JADE) [4] which in turn uses TCP/IP for its communication.

9.3.3 Scale Separation Map of the application

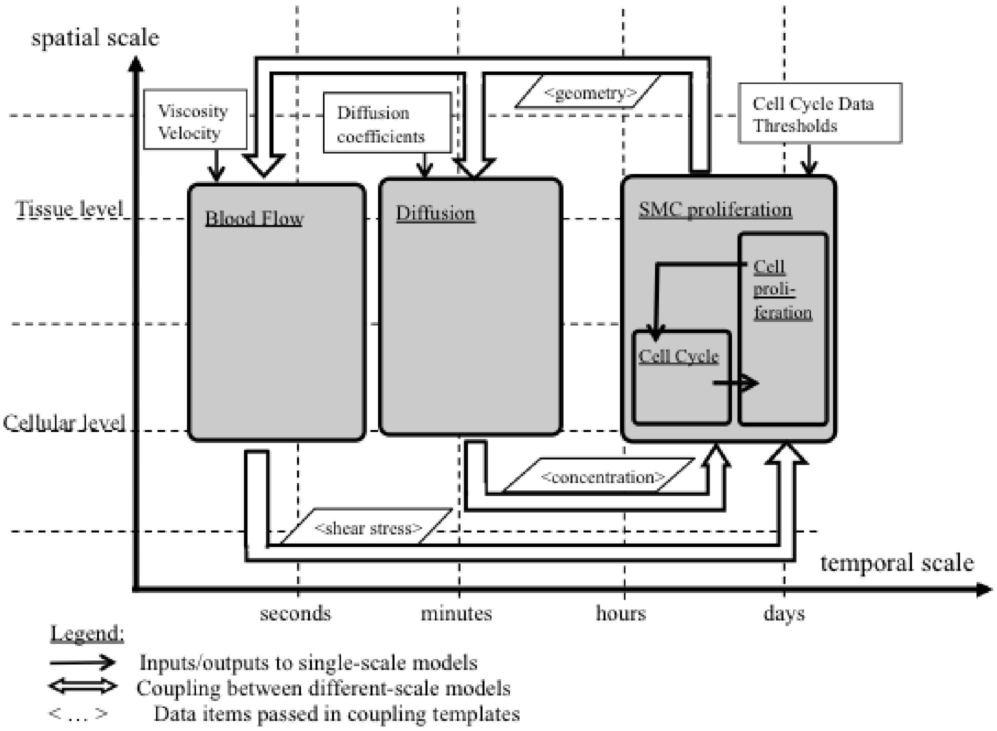


Figure 6: A scale separation map (SSM) of the ISR3D model, excluding initial conditions.

9.3.4 Characterisation of the application coupling template

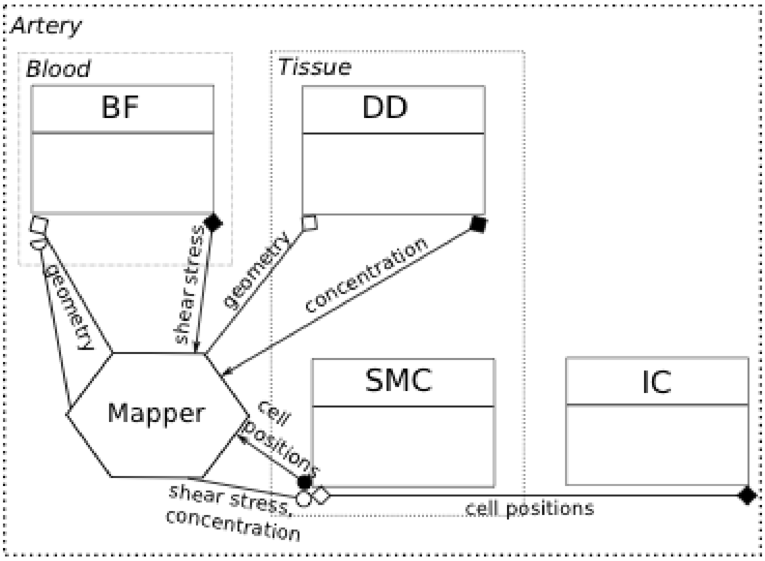


Figure 7: A multiscale modeling language figure of the ISR3D model. The other parts of the characterization of the application coupling template are listed in the Computational Requirements section.

9.3.5 Computational Requirements

9.3.5.1 IC

Requirement	State of the art	Required capabilities
# of cores	1	256
memory per core [MB]	200	40
runtime	20min	2min
coupling frequency	once	once
max. data size (coupling)	150MB	15GB
max. data size (output)	150MB	15GB

9.3.5.2 SMC

Requirement	State of the art	Required capabilities
# of cores	1	256
memory per core [MB]	100	20
runtime	100d	5d
coupling frequency	1/(20min)	1/(1min)

Requirement	State of the art	Required capabilities
max. data size (coupling)	100MB	10GB
max. data size (output)	100MB	10GB

9.3.5.3 BF

Requirement	State of the art	Required capabilities
# of cores	1024	10000
memory per core [MB]	200	100
runtime	10min	1min
coupling frequency	1/(20min)	1/(1min)
max. data size (coupling)	50MB	5GB
max. data size (output)	50MB	5GB

9.3.5.4 DD

Requirement	State of the art	Required capabilities
# of cores	1	128
memory per core [MB]	50	10
runtime	1min	6s
coupling frequency	1/(20min)	1/(1min)
max. data size (coupling)	50MB	5GB
max. data size (output)	50MB	5GB

9.3.6 Software Requirements

Besides the MUSCLE, the BF submodel needs Intel Fortran and for visualization, libpng and libboost are used.

- Intel Fortran Compiler libraries.
- libpng++.
- libboost-dev.

9.3.7 Other characteristics

No visualization is required, other than the PNG library. None of the submodels require specialized hardware, although we have a version of BF optimized for a NEC SX-8 machine. The application is not publicly available at the moment.

9.3.8 References to literature related to the applications:

[1] MUSCLE – http://muscle.berlios.de/.

[2] A. Caiazzo, D. Evans, J.-L. Falcone, J. Hegewald, E. Lorenz, B. Stahl, D. Wang, J. Bernsdorf, B. Chopard, J. Gunn, R. Hose, M. Krafczyk, P. Law- ford, R. Smallwood, D. Walker, and A. G. Hoekstra. Towards a complex automata multiscale model of in-stent restenosis. ICCS 2009, Jan 2009.

[3] D. Evans, P. Lawford, and J. Gunn. The application of multiscale modelling to the process of development and prevention of stenosis in a stented coronary artery. Phil. Trans. R. Soc. A, Jan 2008.

[4] JADE – http://jade.tilab.it/.

[5] A. G. Hoekstra, E. Lorenz, J.-L. Falcone, and B. Chopard. Towards a complex automata framework for multi-scale modeling: formalism and the scale separation map. ICCS 2007, Jan 2007.

[6] A. G. Hoekstra, S. Zwart, M. Bubak, and P. Sloot. Towards distributed petascale computing. Arxiv preprint astro-ph, Jan 2007.

[7] A. G. Hoekstra, J.-L. Falcone, A. Caiazzo, and B. Chopard. Multi-scale modeling with cellular automata: the complex automata approach. Cellular Automata, pages 192–199, 2010.

[8] E. Lorenz. Multi-scale modeling with complex automata: In-stent restenosis and suspension flow. Aug 2010.

[9] H. Tahir, A. G. Hoekstra, E. Lorenz, P. V. Lawford, D. R. Hose, J. Gunn, and D. J. Evans. Multiscale simulations of the dynamics of in-stent restenosis: impact of stent deployment and design. Journal of the Royal Society Interface, 2011.

9.4 Equilibrium stability workflow

9.4.1 Description of the overall application and its submodels

The basic workflow would consist of an MHD equilibrium code (HELENA) followed by a linear MHD stability code (ILSA). Variants are possible which vary the profiles from the equiilibrium code, recalculate the equilibrium for each case and then calculate the MHD stability.

9.4.2 Description of the coupling mechanism between the submodels

This is a loosely coupled workflow where the data can be exchanged wither via files or via structured objects (CPOs) defined by the EFDA-TF-ITM. Variants of the workflow can be defined which add additional components and then require multiple instances of the equilibrium and stability calculation modules.

9.4.3 Scale Separation Map and Characterisation of the application coupling template.

Both HELENA and ILSA cover the same physical domain (0.01 - 2 m) and are not time-dependent, although the ILSA does calculate the growth time of associated instabilies. A full SSM and application template for this application can be found in the entry for the Full Core Workflow.

9.4.4 Computational requirements

9.4.4.1 HELENA

Requirement	State of the art	Required capabilities
# of cores	1	1
memory per core [MB]	3072	3072
runtime	300	300
max. data size (coupling)	10MB	100MB
max. data size (output)	1GB	10GB

9.4.4.2 ISLA

Requirement	State of the art	Required capabilities
# of cores	1	128
memory per core [MB]	3072	3072
runtime	100000	10000
max. data size (coupling)	300MB	10GB
max. data size (output)	10GB	1000GB

9.4.5 Software Requirements

- EFDA-TF-ITM.

- UAL Library.
- BLAS.
- LAPACK .

9.4.6 Other characteristics

The source code of the applications can be found at https://www.efda-itm.eu/.

9.4.7 References to literature related to the applications;

[1] G.T.A. Huysmans, J.P. Goedbloed and W. Kerner, CP90 Conference on Comp. Physics, Word Scientific Publ. Co. 1991, p.371.

9.5 Simulation of transprot turbulence equilibrium and the full fusion core workflow.

9.5.1 Descritopn of the overall application and its submodels

The transport turbulence equilibrium application is a reduced version of the full fusion core workflow that still exhibits key elements of the desired complexity. The three main subcodes involved in transport turbulence equilibrium are:

- HELENA: A 2D equilibrium solver (elliptic, no explicit time, but equilibrium time dependent).
- GEM: A 3D gyrofluid turbulence code which calculates transport coefficients.
- ETS: A 1D transport code which calculates new profiles.

Both for HELENA and GEM a number of modules could be substituted, with differing trade-offs of speed and accuracy/complexity). There are also some simple service modules in addition to these physics modules. The full fusion core workflow is an elaboration of the equilibrium-turbulence-transport workflow.

9.5.2 Description of the coupling mechanism between the submodels

The modules are tightly coupled and data flows from HELENA to GEM and ETS; from GEM to ETS; and from ETS to HELENA and GEM. In this reduced version of the problem none of the modules can run concurrently, although 8 or 16 copies of GEM would run in parallel, each an MPI job using 8 to 64 cores. However, in the full core workflow the subcodes do run concurrently. The data exchange consists of structured objects (CPOs) defined by the EFDA-TF-ITM.

9.5.3 Scale Separation Map

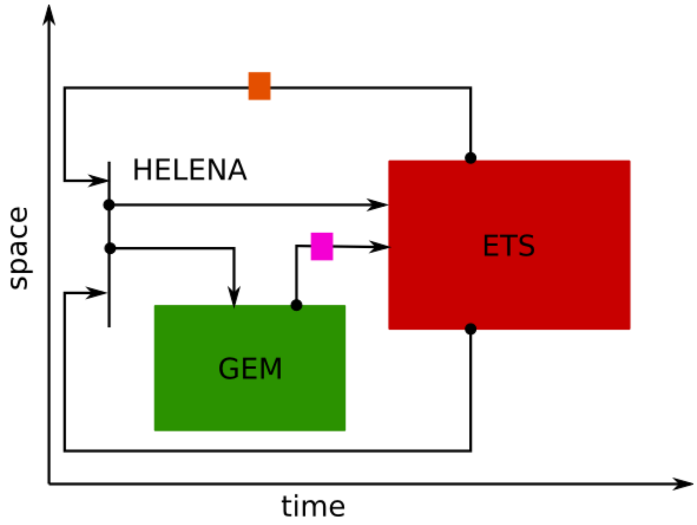


Figure 8: Scale separation amp of the transport turbulence equilibrium application.

9.5.4 Characterisation of the application coupling template

The table below provides an overview of the coupling template within the application in MML.

Submodel	HELENA	GEM	ETS	Transport Combiner	Convergence Checker
HELENA		sD O_f->f_{init} 16/30min 16x100x10MB	sD O_f->f_{init} 1/30min 100x10MB	sD O_f->f_{init} 1/30min 100x10MB	sD O_f->f_{init} 1/30min 100x10MB
GEM				sD O_f->f_{init} 16/30min 16x100x1MB	
ETS					sD O_f->f_{init} 1/30min 100x10MB

Submodel	HELENA	GEM	ETS	Transport Combiner	Convergence Checker
Transport Combiner			sD O_f->f_{init} 1/30min 100x10MB		
Convergence Checker	sD O_f->f_{init} 1/30min 100x10MB	sD O_f->f_{init} 1/30min 100x1MB	sD O_f->f_{init} 1/30min 100x1MB		

9.5.5 Computational requirements

9.5.5.1 HELENA

Requirement	State of the art	Required capabilities
# of cores	1	1
memory per core [MB]	100	100
runtime	30	30
coupling frequency	2/hour	10/hour
max. data size (coupling)	20MB	20MB
max. data size (output)	1GB	100GB

9.5.5.2 GEM/FEFI

Requirement	State of the art	Required capabilities
# of cores	128-512	4096-65536+
memory per core [MB]	2048	2048
runtime	1800	360
coupling frequency	2/hour	10/hour
max. data size (coupling)	10MB	10MB
max. data size (output)	10GB	1000GB

9.5.5.3 ETS

Requirement	State of the art	Required capabilities
# of cores	1	1
memory per core [MB]	100	100
runtime	10	10
coupling frequency	2/hour	10/hour
max. data size (coupling)	10MB	10MB
max. data size (output)	1GB	100GB

9.5.5.4 Full core flow

These requirements are sums over all modules involved in the full core workflow.

Requirement	State of the art	Required capabilities
# of cores	128	65536+
memory per core [MB]	1024	4096
runtime	hours to days elapsed	hours to days
coupling frequency	from1/sec to 1/hour	from 1/sec to 1/hour
max. data size (coupling)	10MB	1000MB
max. data size (output)	100GB	10000GB

9.5.6 Software requirements

Currently the software uses the EFDA-TF-ITM UAL library which uses:

- MDSplus
- BLAS
- LAPACK
- PETSC
- BLITZ

9.5.7 Other characteristics

In what couplings or submodels, if any, is user interaction involved? Think of steering or vizualization.

The full core workflow scenario supports computational steering through feedback loops.

Address of the application's website or code repository:

https://www.efda-itm.eu/

9.5.8 References to literature related to the application:

[1] G.T.A. Huysmans, J.P. Goedbloed and W. Kerner, CP90 Conference on Comp. Physics, Word Scientific Publ. Co. 1991, p.371.

[2] For the models, DALF: B Scott, PPCF 39 (1997) 1635, updated for warm ions PPCF 40 (1998) 823; GEM model starts from Beer/Hammett Phys Plasmas 3 (1996) 4046, made collisional/electromagnetic: B Scott, Phys Plasmas 7 (2000) 1845; GEM3: PPCF 45 (2003) A385; GEM free energy conserving model end of 2003 (preprint under preparation). For the geometry, globally consistent flux tube/field line connection: B Scott, Phys Plasmas 5 (1998) 2334; shifted metric to treat slab modes: B Scott, Phys Plasmas 8 (2001) 447; stellarator metric treatment: A Kendl et al, PPCF 42 (2000) L23

[3] D.P. Coster, V. Basiuk, G. Pereverzev, D. Kalupin, R. Zagórksi, R. Stankiewicz, P. Huynh, and F. Imbeaux. The european transport solver. Plasma Science, IEEE Transactions on, 38(9):2085 -2092, 09 2010.

[4] F. Imbeaux, J.B. Lister, G.T.A. Huysmans, W. Zwingmann, M. Airaj, L. Appel, V. Basiuk, D. Coster, L.-G. Eriksson, B. Guillerminet, D. Kalupin, C. Konz, G. Manduchi, M. Ottaviani, G. Pereverzev, Y. Peysson, O. Sauter, J. Signoret, and P. Strand. A generic data structure for integrated modelling of tokamak physics and subsystems. Computer Physics Communications, 181(6):987 - 998, 2010.

[5] C.F. Maggi, R.J. Groebner, C. Angioni, T. Hein, L.D. Horton, C. Konz, A.W. Leonard, C.C. Petty, A.C.C. Sips, P.B. Snyder, J. Candy, R.E. Waltz, ASDEX Upgrade and DIII-D Teams. Pedestal and core confinement of hybrid scenario in ASDEX Upgrade and DIII-D. Nuclear Fusion 50(2), 025023 (2010).

9.6 Simulation of cerebral blood flow (HemeLB)

9.6.1 Description of the overall application and its submodels

We are concerned with performing neurovascular blood flow simulations in support of clinical neurosurgery. Cerebral blood flow behaviour plays a crucial role in the understanding, diagnosis and treatment of cardiovascular disease; problems are often due to anomalous blood flow behaviour in the neighbourhood of bifurcations and aneurysms within the brain. Simulation offers the possibility of performing patient-specific, virtual experiments to study the effects of courses of treatment with no danger. For this work, we will use our lattice-

Boltzmann code, HemeLB [1] [2], designed to simulate fluid flow in the sparse topologies of the human brain. The code can create visualizations from within a running simulation and send them to a viewing client on a workstation situated in, eventually, the hospital. The clinician can then steer the parameters of the simulation.

This work uses the HemeLB code - which is well-suited to describe fluid behaviour - within sparse systems such as neurovasculatures. The code is written in C/C++ and parallelized using MPI. It employs a number of algorithmic optimizations and techniques to efficiently compute and communicate flow data in sparse and complex geometries. The domain decomposition ensures that computational domains are equally partitioned. Efficient layout of memory and memory access patterns further optimizes the fluid simulation. HemeLB has built in real-time rendering and steering capabilities such that the time-varying behaviour of the blood flow can be visualized live by a parallel in situ ray tracing algorithm. Each frame is transmitted over the network to a workstation which visualizes the frame in real time. The visual and physical parameters can be interactively modified through steering capabilities. The analysis and visualization, particularly in real-time, is essential to this successful use of this code in clinical environments.

Away from the region of interest, the accuracy demanded of the hydrodynamic simulation is lower and so can be simulated at a lesser computational expense at a lower resolution, without significantly affecting the accuracy. Further still away, the rest of the circulatory system can be abstracted to a network model of the vasculature and a pump, i.e. the heart. We will couple this hierarchy on models together.

9.6.2 Description of the coupling mechanism between the submodels

The submodels are all tightly coupled, each providing pressure and velocity boundary conditions to the concurrently-executing, adjacent submodels. The frequent communication between models places demanding requirements on the latency of the coupling library, although in the first instance the volume of data to be exchanged is low. The coupling from the network model to HemeLB will require construction of a flow profile (typically a parabolic Poiseuille flow profile) and the reverse coupling will require computation of the the average pressure and velocity. Coupling of the different resolution HemeLB simulations will require resampling of the velocity and pressure to the new resolution. The number of submodels and the coupling between them is determined during simulation setup.

9.6.3 Scale Separation Map

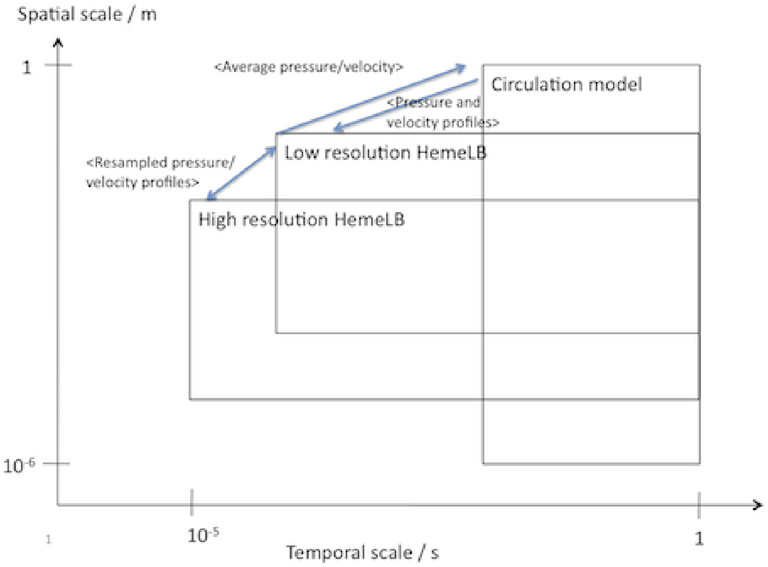


Figure 9: Scale Separation Map of the HemeLB application

9.6.4 Characterication of the application coupling template

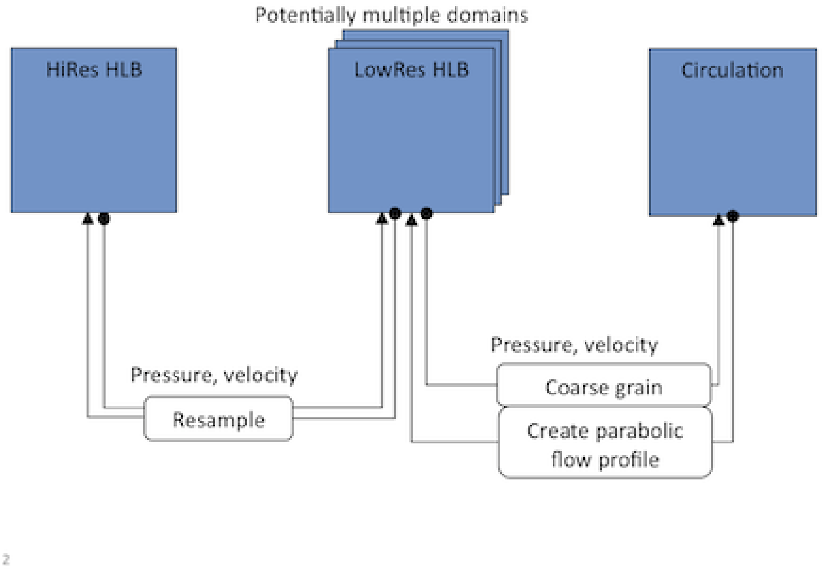


Figure 10: Multiscale Modelling Langugae diagram of the HemeLB application.

9.6.5 Computational requirements

9.6.5.1 HemeLB

Requirement	State of the art	Required capabilities
# of cores	1000	20000
memory per core [MB]	100	100
runtime	30 min	30 min
coupling frequency	1/time step	1/time step
max. data size (coupling)	1 kB	100 kB
max. data size (output)	1GB	10GB

9.6.5.2 Network flow model

Requirement	State of the art	Required capabilities
# of cores	1	100
memory per core [MB]	10	100
runtime	minutes	minutes
coupling frequency	1/time step	1/time step
max. data size (coupling)	1 kB	100 kB
max. data size (output)	10 MB	100GB

9.6.6 Software Requirements

- MPI
- ParMETIS graph partitioning library.

9.6.7 Other characteristics

In what couplings or submodels, if any, is user interaction involved?

HemeLB has built in real-time rendering and steering capabilities such that the time-varying behaviour of the blood flow can be visualized live by a parallel in situ ray tracing algorithm. Each frame is transmitted over the network to a workstation which visualizes the frame in real time.

Address of the application's website or code repository:

HemeLB is not yet publicly available.

9.6.8 References to literature related to the applications:

[1] Mazzeo et al. In situ ray tracing and computational steering for interactive blood flow simulation . Computer Physics Communications (2010) vol. 181 pp. 355-370.

[2] Mazzeo and Coveney. HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries. Computer Physics Communications (2008) vol. 178 (12) pp. 894 - 914.

9.7 Simulation of irrigation canals

9.7.1 Description of the overall application and its submodels

Canals or rivers are a central infrastructure in all populated areas. They ensure an adequate supply of water for agriculture and are a key component of electricity production or transportation. An optimal management and the control of such resources can be a critical issue for long term planning or to react to natural hazards. In a recent collaboration with ESISAR (Ecole nationale supérieure des systèmes avancés et réseaux) at Grenoble INP, France, UNIGE has developed multiscale models for management of a network of irrigation canals. The problem to be solved is to define appropriate actions (e.g. opening and closing gates) that need to be taken to always guarantee an adequate water supply throughout the canal system, whatever the external demands or perturbations can be, and respecting constraints such as water height. Our simulation example is a model of the canal de la Bourne irrigation network. This network was built in the late 19th century to irrigate the plains around Valence in France.

Figure 11: A French La Bourne irrigation canal, which is modeled within the hydrology MAPPER applications.

It is still in use now and its fine modelling and control has become a new challenge. Indeed, the demand on water considerably increased these last few decades as the constraints on the quantity of water which may be withdrawn from the up-stream natural river La Bourne became more and more limiting.Four main sub-models have been identified where each sub-model can be instantiated several times inside the global model. The CxA formalism [1] based on the lattice boltzmann (LB) approach [2] is used for the implementation of these submodels.

9.7.2 Shallow Water 1D (SW1D)

Shallow Water 1D (SW1D) The one-dimensional shallow water equation can be used to describe long canal sections. In [3], a model has already been developed based on the D1Q3 lattice Boltzmann (LB) shallow water equation, analyzed in detail and compared with other numerical schemes. Remarkably, it has been shown how such D1Q3 LB models for canal reaches may be easily coupled with various hydraulic interconnection structures to build models of complex irrigation networks.

9.7.3 Shallow Water 2D (SW2D)

It describes branching regions or pools in which the water height varies from the left to the right side. 2D LB-SW models have been considered in several papers [4][5][6]. The coupling SW1D-SW2D has been considered in [7].

9.7.4 Free Surface 3D

A full 3D, free-surface model is required to describe in detail the flow around gates or to describe the transport of sediments that can deposit along the canal. This 3D single scale model requires supercomputing capabilities. Several relevant bi-fluid lattice Boltzmann (LB) models [8][9][10] has been developed. Recently, Marcou et al. have adapted these approaches to implement a (LB) model which describes the flow near a gate, with or without sediment transport and erosion [11], fully resolving all the components of the velocity flow. To manage a full system of irrigation canals one needs to couple descriptions at all these three scales and then investigate response scenarios to given perturbations. In [12][13], animations of open water phenomena have been performed based on the (LB) model and particularly with coupled 2D shallow water and 3D free surface simulations.

9.7.5 Sediment transport

Sedimentation strongly influences the water flow, i.e. reduces the efficiency of the irrigation network. Sediment can be defined as a fragmented material from rocks that has been formed by different physical and/or chemical process and may be transported in three ways:

bedload, saltation and suspension. In [14], a complete description of shallow water models including sediment transport is presented. Moreover, sedimentation phenomenon has been widely investigated using the LB approach [15][16][17][18].

9.7.6 Description of the coupling mechanism between the submodels

The models within the hydrology application are all tightly coupled. The main components consist of the Shallow water 2D model and the Free Surface 3D model.

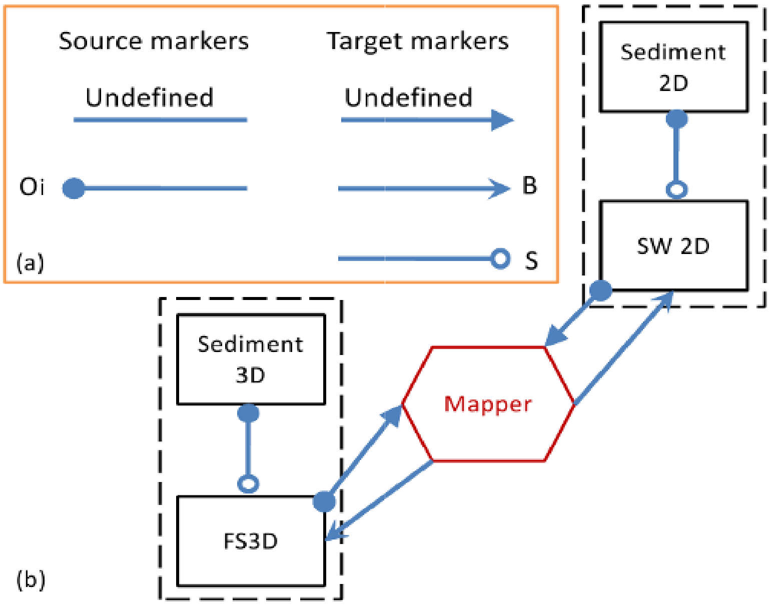


Figure 13: A multiscale modeling figure of an irrigation canal application.

9.7.7 Scale Separation Map and Characterication of the application coupling template

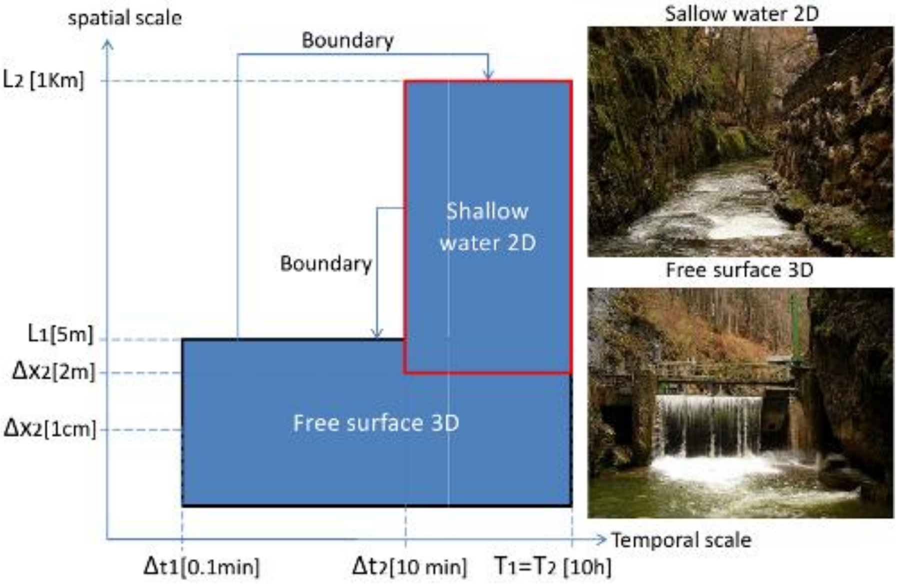


Figure 12: A scale separation map (SSM) of an irrigation canal application.

9.7.8 Computational requirements

9.7.8.1 SW1D

Requirement	State of the art	Required capabilities
# of cores	1	1
coupling frequency	1/hour	10/min
max. data size (coupling)	1MB	100MB
max. data size (output)	1GB	100GB

9.7.8.2 SW2D

Requirement	State of the art	Required capabilities
# of cores	1	1
coupling frequency	1/hour	10/min
max. data size (coupling)	1MB	100MB
max. data size (output)	1GB	100GB

9.7.8.3 FS3D

Requirement	State of the art	Required capabilities
# of cores	1	1000
memory per core [MB]	100	100
runtime	1 hour	1 hour
coupling frequency	1/hour	10/min
max. data size (coupling)	1MB	100MB
max. data size (output)	1GB	100GB

9.7.8.4 ST

Requirement	State of the art	Required capabilities
# of cores	1	1000
memory per core [MB]	100	100
runtime	1 hour	1 hour
coupling frequency	1/hour	10/min
max. data size (coupling)	1MB	100MB

max. data size (output)	1GB	100GB

9.7.9 Software requirements

- C++
- Java

9.7.10 Other characteristics

This code is still largely being worked on, and is not yet available in the public domain.

9.7.11 References to literature of the related application:

[1] Alfons Hoekstra, Jean-Luc Falcone, Alfonso Caiazzo, and Bastien Chopard. Multi-scale modeling with cellular automata: The complex automata approach. In Hiroshi Umeo, Shin Morishita, Katsuhiro Nishinari, Toshihiko Komatsuzaki, and Stefania Bandini, editors, Cellular Automata, volume 5191 of Lecture Notes in Computer Science, pages 192-199. Springer Berlin / Heidelberg, 2008.

[2] Shiyi Chen and Gary D. Doolen. Lattice boltzmann method for fluid fows. Annual Review of Fluid Mechanics, 30(1):329-364, 1998.

[3] Pham van Thang, Bastien Chopard, Laurent Lefevre, Diemer Anda Ondo, and Eduardo Mendes. Study of the 1d lattice boltzmann shallow water equation and its coupling to build a canal network. Journal of Computational Physics, 229(19):7373 -7400, 2010.

[4] Salmon R. The lattice boltzmann method as a basis for ocean circulation modeling. Journal of Marine Research, 57:503-535(33), 1 May 1999.

[5] Paul J. Dellar. Nonhydrodynamic modes and a priori construction of shallow water lattice boltzmann equations. Phys. Rev. E, 65(3):036309, Feb 2002.

[6] J.G. Zhou. Lattice Boltzmann Methods for Shallow Water Flows. Springer, 2004.

[7] Pham van Thang. Modlisation et commande des systèmes non-linaires paramètres distribus par la méthode de boltzmann sur réseau: application aux canaux d'irrigation. Master's thesis, Grenoble INP ESISAR, France, 2009.

[8] Xiaowen Shan and Hudong Chen. Lattice boltzmann model for simulating flows with multiple phases and components. Phys. Rev. E, 47(3):1815-1819, Mar 1993.

[9] Xiaowen Shan and Hudong Chen. Simulation of nonideal gases and liquid-gas phase transitions by the lattice boltzmann equation. Phys. Rev. E, 49(4):2941-2948, Apr 1994.

[10] Nicos S. Martys and Hudong Chen. Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice Boltzmann method. Phys. Rev. E, 53(1):743-750, Jan 1996.

[11] Olivier Marcou, Samira El Yacoubi, and Bastien Chopard. A bi-fluid lattice boltzmann

model for water ow in an irrigation channel. In Samira El Yacoubi, Bastien Chopard, and Stefania Bandini, editors, Cellular Automata, volume 4173 of Lecture Notes in Computer Science, pages 373-382. Springer Berlin / Heidelberg, 2006.
[12] Nils Thürey. Physically based Animation of Free Surface Flows with the Lattice Boltzmann Method. PhD thesis, University of Erlangen-Nuremberg, Germany, 2007.
[13] Nils Thürey, Ulrich Rüde, and Marc Stamminger. Animation of open water phenomena with coupled shallow water and free surface simulations. In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '06, pages 157-164, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
[14] W.H. Graf. Hydraulique fluviale. Presses polytechniques et universitaire romandes, 1993.
[15] Bastien Chopard, Alexandre Masselot, and Alexandre Dupuis. A lattice gas model for erosion and particles transport in a uid. Computer Physics Communications, 129(1-3):167-176, 2000.
[16] A. Dupuis. From a Lattice Boltzmann model to a parallel and reusable implementation of a virtual river. PhD thesis, University of Geneva, Switzerland, 2002.
[17] Alexandre Dupuis and Bastien Chopard. Lattice gas modeling of scour formation under submarine pipelines. J. Comput. Phys., 178:161-174, May 2002.
[18] A. Masselot and B. Chopard. A lattice boltzmann model for particle transport and deposition. EPL (Europhysics Letters), 42(3):259, 1998.
[19] Jean-Luc Falcone, Bastien Chopard, and Alfons Hoekstra. Mml: towards a multiscale modeling language. Procedia Computer Science, 1(1):819-826, 2010. ICCS 2010.

9.8 The bile acid and xenobiotic system

9.8.1 Description of the overall application and its submodels

The bile acid and xenobiotic system (BAXS) defines an intricate physiological network of chemoprotective and transporter-related functions that ensure the detoxification and removal from the body of harmful xenobiotic and endobiotic compounds while ensuring that primary bile acids (essential for the emulsification and absorption of dietary fats and fat-soluble vitamins) are not eliminated and can be re-used. The overall metabolic flux of the BAXS is primarily achieved through the activities of nuclear receptors, which have the ability to directly bind to DNA and regulate gene expression. Nuclear receptors can be thought of as metabolic sensors of exogenous and endogenous toxins and a detailed knowledge of the factors that govern their activity has critical implications to a range of physiological processes such as drug-drug interactions, intracrine hormone metabolism, xenobiotic clearance and cholesterol/lipid homeostasis.

The BAXS involves the coordinated activities of many genes across multiple temporal and spatial scales. Basic BAXS processes and their time scales include the binding of ligands to nuclear receptors (hours), gene expression and regulation (hours), transporter protein (minutes) and metabolic enzyme activity (seconds). Spatially, BAXS components range from molecules (e.g., nuclear receptors) to organs (e.g., liver). Given the complex multiscale nature the BAXS, it is difficult to assess the exact importance and impact of individual receptors and their activating/deactivating ligands with respect to the overall BAXS flux and how it can vary throughout a number of participating organ systems. A comprehensive description of the interacting components that govern BAXS gene expression would enable the identification of regulatory "nodes" as targets for treatment regimes, and understanding of the components impacting drug-drug interactions, and provide a framework for the design of large-scale, integrated prediction studies.

The aim of this application is to explore and develop novel modelling and simulation techniques facilitating a multiscale characterization of the BAXS. Conceptually, we adopt a bottom-up multiscale systems biology approach (Kitano, 2002), aiming to derive system behaviour on higher spatial or temporal scales from the dynamics and interactions of BAXS model components at lower, more detailed scales. The coarse graining that connects the different scales involves identifying which types of collective behaviour on a fundamental scale give rise to a coherent phenomenon on a higher scale. The great disadvantage of bottom-up models, namely the requirement to invest much care into the construction of the fundamental modelling layer, is at the same time their main advantage: the process of assembling the model unveils gaps in our knowledge and points out new directions for experimental studies that without the modelling effort would be less apparent. Another problem with the bottom-up multiscale approach in systems biology is the lack of sufficient data. Because of this, we will initially develop reference models manually, based on information from the literature, and concentrate on the exploration, evaluation and development suitable multiscale modelling methodology and technology(Meier-Schellersheim et al., 2009; Schnell et al., 2007).

For our BAXS application, we've identified four scale-specific models, as follows:

- Process A – Ligand/Receptor binding – involves detection and binding of ligands which thus activate the nuclear receptors. This process takes place in the cytosol.
- Process B – Receptor activates gene – B results in increased gene expression. This process takes place in the nucleus.
- Process C – Enzyme activity on substrate – involves metabolism of substrates including competitive inhibition by another substrate. This process takes place in the cytosol.

- Process D – Transport of substrate from cell – involves the transport of metabolites from the cell including competitive inhibition by another substrate. This process takes place across the cellular membrane.

9.8.2 Description of the coupling mechanism between the submodels

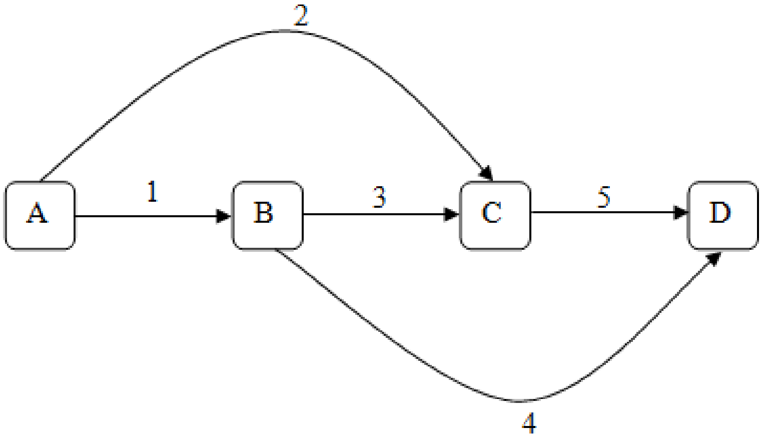


Figure 14: Data Flow diagram of the BAXS

- Process A: Ligand/Nuclear receptor binding
- Process B: Gene expression
- Process C: Metabolism with competitive inhibition
- Process D: Transport with competitive inhibition

Process A, Ligand/Nuclear receptor binding, is modelled over a 10 minute period and requires the concentrations of 6 components as inputs: 2 nuclear receptors PXR and FXR, and 4 substrates CDCA, LCA, Hyperforin and Ritonavir. Only a small concentration of substrate is needed to activate the nuclear receptors (approx 1µmol/l). The products of process A are bound nuclear receptors which translocate to the nucleus. The data exchanged between processes A and B therefore is a concentration of activated nuclear receptors (PXR:Hyp, PXR:Rit, FXR:CDCA, FXR:LCA). The remaining substrate (unbound) is exchanged with processes C and D. Hyperforin, LCA and ritonavir concentrations are exchanged with process C, and ritonavir concentrations are also exchanged with process D.

Process B, gene expression, is modelled over a period of 27 hours and requires the products of process A as inputs. The products of process B are CYP3A4 enzymes which are exchanged with process C, and MDR1, MRP2 and BSEP transporter proteins which are exchanged with process D.

Process C, metabolism, is modelled over a period of 160 seconds and requires substrate concentrations from process A and the products of process B as inputs. The products of process C are metabolised substrates (Hyp(a) and LCA(a)) which are exchanged with process D. Ritonavir concentrations from process A effect the rate of metabolism as this substrate inhibits CYP3A4 activity through competitive inhibition.

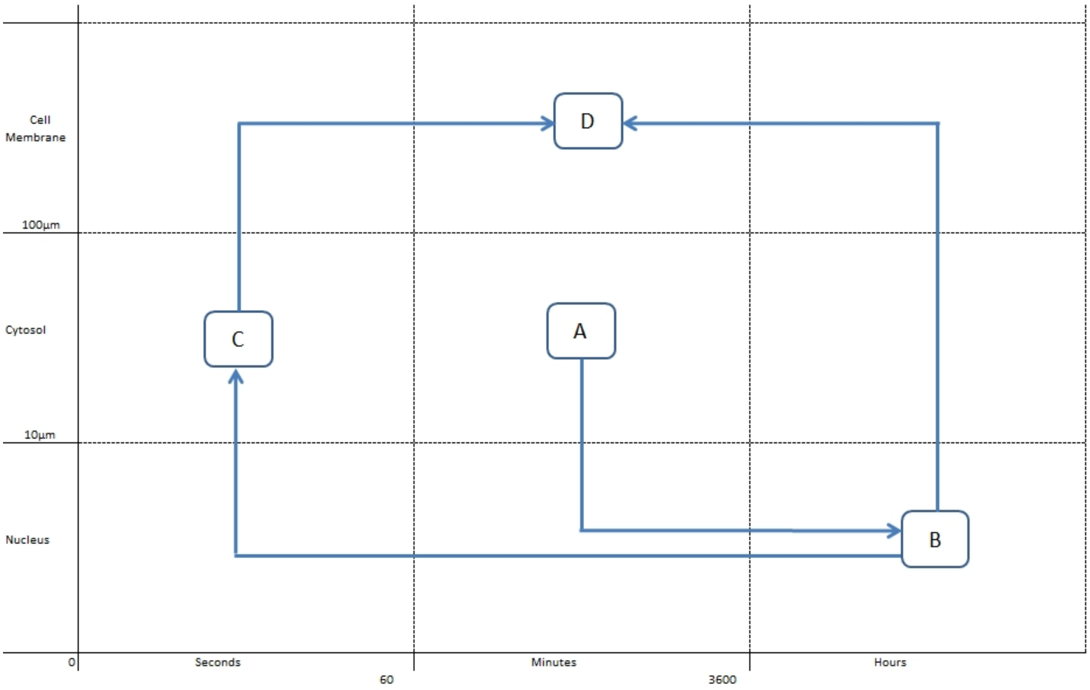
Process D, transport, is modelled over a period of 5 minutes and requires the products of process C, ritonavir concentrations from process A and transporter protein concentrations from process B as inputs. The effect of this process is a decrease in metabolized substrates in the cell as they are transported across the cell membrane. The rate of transport is affected by the concentration of ritonavir as this inhibits the activity of transporter proteins through competitive inhibition

Description of the coupling between the submodels:

1. A to B: For every 20 time steps of A data is exchanged with B (40 seconds). The data exchanged is a concentration of four products.
2. A to C: exchange is only once at the end of process A. A to C exchanges concentrations of remaining substrates Hyperforin, LCA and Ritonavir.
3. B to C: exchanges only once at the end of the process B. The data exchanged is CYP3A4 enzyme concentrations.
4. B to D: exchanges only once at end of process B. The data exchanged is concentrations of MDR1, MRP2 and BSEP enzymes.
5. C to D: for every 10 time steps of C data is exchanged with D. The data exchanged is concentrations of metabolized Hyperforin and LCA, and remaining concentration of Ritonavir.

Due to the relatively short running time of process C as compared to process B, process C will only start after process B has finished. The same is true for the B and D pair of processes. There will, however, be an overlap between the running of processes A and B, and between C and D. For an instance of the model, the submodels have a fixed number of instances, always 1:1. We will need, however, to span a large number of instances of the model when doing reverse engineering or sensitivity analysis.

9.8.3 Scale Separation Map



9.8.4 Characterication of the application coupling template

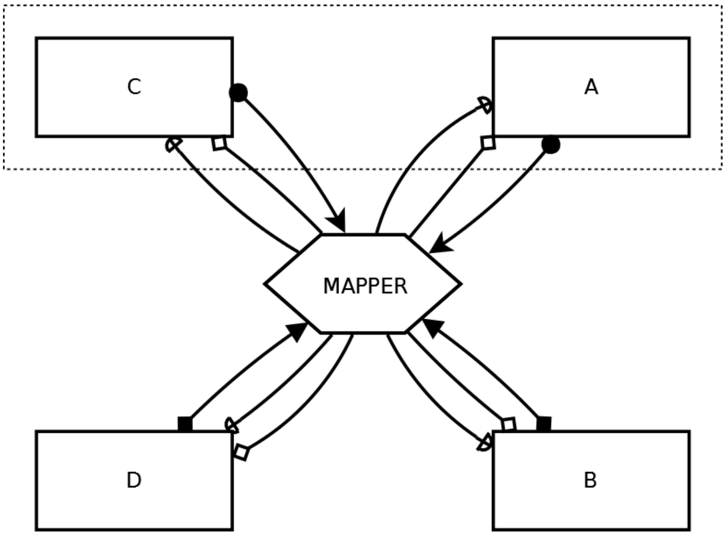


Figure 16: Multiscale Modelling Language diagram of the BAXS.

9.8.5 Computational requirements

9.8.6 Submodel A:

Requirement	State of the art	Required capabilities
# of cores	100	200
memory per core [MB]	10	100

runtime	1-5 hours	1-5 hours
coupling frequency	1/min	1/min
max. data size (coupling)	1MB	100MB
max. data size (output)	1MB	100MB

9.8.7 Submodel B

Requirement	State of the art	Required capabilities
# of cores	100	200
memory per core [MB]	10	100
runtime	1-5 hours	1-5 hours
coupling frequency	1/run (at the end)	1/run (at the end)
max. data size (coupling)	1MB	100MB
max. data size (output)	1MB	100MB

9.8.8 Submodel C

Requirement	State of the art	Required capabilities
# of cores	100	200
memory per core [MB]	10	100
runtime	1-5 hours	1-5 hours
coupling frequency	1/min	10/min
max. data size (coupling)	1MB	100MB
max. data size (output)	1MB	100MB

9.8.9 Submodel D

Requirement	State of the art	Required capabilities
# of cores	100	200
memory per core [MB]	10	100
runtime	1-5 hours	1-5 hours
coupling frequency	1/run (at the end)	1/run (at the end)
max. data size (coupling)	1MB	100MB

max. data size (output)	1MB	100MB

9.8.10 Software Requirements

- COPASI [1] [6] biochemical network simulator, which also provides SBML support;
- SBRML [7] support (e.g. through a tool like SBRML.NET [8]);
- An ODE solver, such as XPPAUT [9].

9.8.11 Other characteristics

The BAXS code is not yet available in the public domain.

9.8.12 References to literature related to the application

[1] Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P. & Kummer, U.: COPASI – a COmplex PAthway Simulator, Bioinformatics, 22(24):3067–3074, 2006.

[2] Kitano, H.: Computational systems biology, Nature, 420: 6912, 206–10, 2002.

[3] Meier-Schellersheim, M., Fraser, I. D. C., & Klauschen, F., Multi-scale modelling for biologists, WIREs Systems Biology and Medicine, 1:4–14, 2009.

[4] Schnell, S., Grima, R. & Maini, P. K., Multi-scale modelling in biology: New insights into cancer illustrate how mathematical tools are enhancing the understanding of life from the smallest scale to the grandest, American Scientist, 95:134–142, 2007.

[5] Kennedy, N., Dubitzky, W., Thompson, P. & Zheng, H., Multi-Scale modelling of the Bile Acid and Xenobiotic System, Manuscript submitted for publication.

[6] COPASI – http://www.copasi.org.

[7] SBRML – http://www.comp-sys-bio.org/tiki-index.php?page=SBRML.

[8] SBRML.net - http://sourceforge.net/projects/sbrml/files.

[9] XPPAUT – http://www.math.pitt.edu/~bard/xpp/xpp.html.

10 Appendix B: Review on Users and User Communities

Multiscale and multiscience applications are present in a wide range of scientific and engineering communities. By its nature, multiscale modeling is highly interdisciplinary, with developments occurring independently across fields. A broad range of scientific and engineering problems involve multiple scales. Traditional monoscale approaches have proven to be inadequate, even with the largest supercomputers, because many physical processes are inherently taking place across a range of spatial and temporal scales. As a result, there is a growing need to develop systematic modeling and simulation approaches for multiscale problems.

In this section we provide an overview of several scientific communities which actively seek to develop and apply multiscale simulations. This list is obviously not exhaustive. Its goal is mostly to illustrate the different flavors of the multiscale approaches and the benefit of developing of a common and reusable methodology.

Aside from the numerous publications, project websites and domain-specific reviews available, we also identified a few sources which provide information on multiscale simulations in multiple scientific domains. One source of information about the other communities is the Journal of Multiscale Modeling and Simulation (MMS) which defines itself as “an interdisciplinary journal focusing on the fundamental modeling and computational principles underlying various multiscale methods". In addition, there is a series of multiscale workshops which are part of the ICCS conference (http://staff.science.uva.nl/~valeria/SMMS), where a wide variety of multiscale simulations from different domains have been presented over the years.

10.1 Fusion

10.1.1 Description of the user community

Within the EU between 100 and 200 hundred people are involved with modelling of present fusion devices and preparing for modelling of ITER and DEMO. Similar sized communities exist in the USA, with smaller communities in the other ITER partners. Much of the effort with in Europe is centered around the EFDA Task Force on Integrated Tokamak Modelling (https://www.efda-itm.eu).

10.1.2 User needs

A number of workflows have been identified as needed and are under development. Two workflows have been identified as exhibiting characteristic behaviour and are being investigated in the MAPPER project.

10.1.3 Main usage templates

Some workflows are loosely coupled, most are expected to be tightly coupled.

10.1.4 Short term user needs

Workflows as described elsewhere in this document.

10.1.5 Long term user needs

Once the basic workflows have been implemented, the workflows will be elaborated with additional modules.

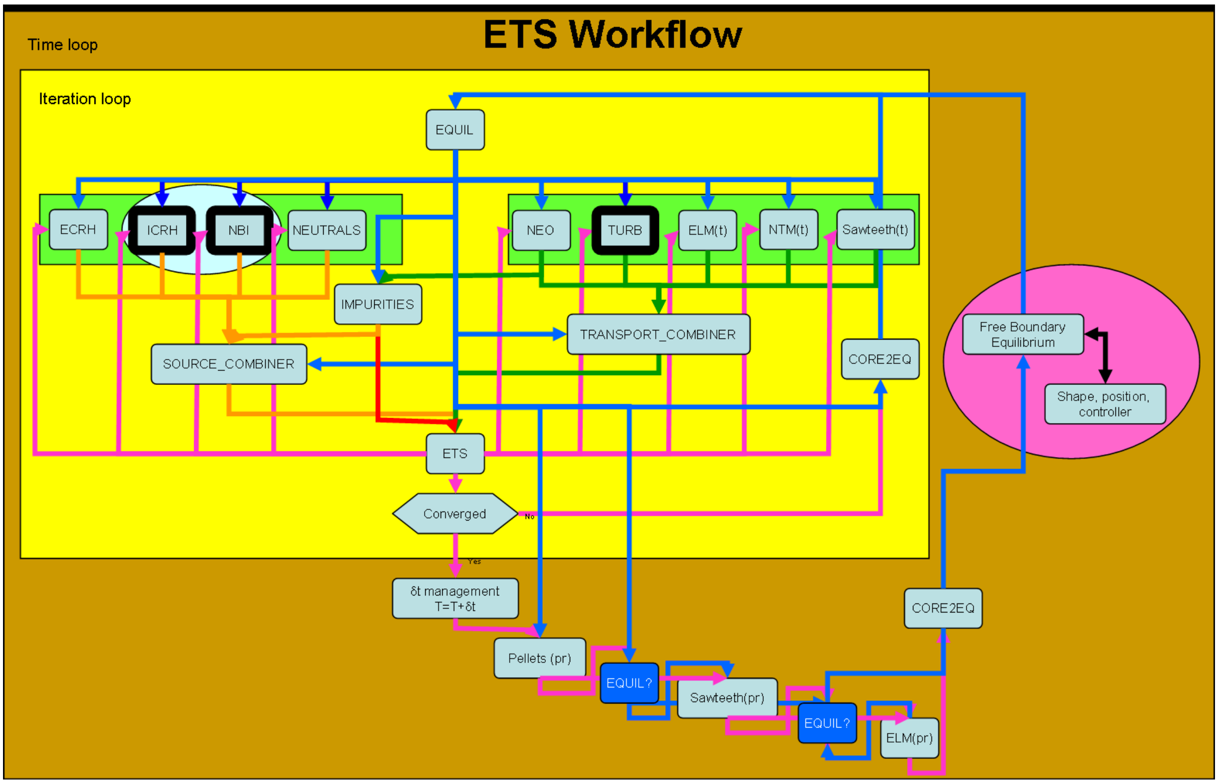


Figure 17: An example workflow within the Fusion community.

10.1.6 Example application

See Appendix A: Application Review Equilibrium Stability Workflow and Appendix A: Application Review Transport Turbulence Equilibrium.

10.2 System biology

10.2.1 Description of the user community

Generally, modeling and simulation of complex (biological) systems involves compute-intensive (and sometimes data-intensive) techniques, for example, for model reverse-engineering or model identification, model validation, sensitivity analysis, etc. For single-scale models, many techniques exist to address these tasks. In contrast to single-scale approaches, multiscale modeling and simulation (MMS) typically involves a collection of (possibly heterogeneous) scale-specific models that together represent the model or simulation. A key challenge in MMS is to find effective and efficient ways for the individual scale-specific models to interoperate. Because of the heterogeneous nature and the need of interoperation of individual MMS components, the computational requirements are considerably more complex than those for single-scale approaches.

The paper “Multi-scale modelling and simulation in systems biology” by Joseph O. Dada and Pedro Mendes [1] contains a an overview of several large projects in systems biology. In addition, we will cover a number of research groups and projects here:

10.2.1.1 University of Southampton-Computational Modelling Group

The Computational Modelling Group brings together Southampton university researchers using applied Computational Modelling to support the understanding and advancement of physical and natural sciences, engineering, medicine, economy, society, psychology, and other fields. This includes both the development of new computational methods and the application of existing simulation tools and software packages.

They have a broad list of research projects, some of which have HPC demands. For these, they have their own supercomputer named Iridis (rank 74 in Top500).

Typical research project: Cellular Automata Modelling of Membrane Formation and Protocell Evolution. Excerpt from the project description: "This research aims to assess some basic mechanisms by which amphiphilic molecules, synthesised through simple chemical reaction networks, can form cell-like entities which must adapt to their environment in order to persist. Having extended a model first put forward by Ono and Ikegam (2001), we have already observed behaviour that could be defined in basic ecological terms. And yet our artificial system consists only of chemical substances interacting on a purely local scale. All long range order and coherent behaviours are emergent properties. Due to this short interaction scale, the model is eminently parallelizable. All simulations are performed by Iridis using a C code with MPI routines. These routines allow for the division of computational labour amongst many separate computing cores (typically 8 nodes using all 8 cores per node). The MPI routines facilitate communication between the individual cores so that particles and forces can be exchanged across domain boundaries."

10.2.1.2 University of Edinburgh – Centre for Systems Biology at Edinburgh (CSBE)

Their focus is on modelling within systems biology, having the aim to make theoretical and practical developments to support all stages of the modelling process by developing large-scale modelling frameworks.

Notable research project: "Model Analysis Project", which deals with Ordinary differential equation (ODE) models (http://csbe.bio.ed.ac.uk/model_analysis.php). Excerpt from the project’s description: "Software tools are available for detailed analysis of small-scale dynamic systems however CSBE has proposed to modify and extend these to create new software tools that will specifically use high-performance, parallel computation to explore the parameter space of large ODE models including thousands of equations."

10.2.1.3 The Integrative Biology Project

The Integrative Biology Project is bringing together an international consortium of leading biomedical and computing researchers to address two of the most important problems in clinical medicine today: understanding what causes heart failure and how cancer tumours develop and grow.

The aim is to develop detailed, accurate multi-scale computational models of the heart and of cancer tumours, exploiting the new Grid infrastructure to run these models on the most powerful supercomputers available for research in the UK today.

10.2.1.4 Dublin City University – SCI –SYM

The Centre for Scientific Computing & Complex Systems Modelling (SCI-SYM) is a centre of excellence for researchers working in high performance computing (HPC) applied to computational and mathematical models for complex systems in engineering, natural and applied sciences. It explores models of the natural and artificial world, through high performance computer solutions of problems, which, due to their complexity, are intractable by conventional methods such as experimental, mathematical or semi-analytical methods alone. Complex systems arise in a variety of fields, e.g. physics, biology, chemistry, eco- and other hybrid sciences, finance, socio-economic phenomena, and others and are truly interdisciplinary. In some cases, a formal model may be proposed and investigated; in others large amounts of data may be mined and empirically analysed or computational models may be designed and tested against available data.

Example PhD project: "Genetic regulatory network (GRN) inference from microarray data with evolutionary computation" (http://sci-sym.computing.dcu.ie/index.php?option=com_content&view=article&id=63&Itemid=77).
Excerpt from the project’s description: "This work attempts to develop a novel evolutionary computation approach to GRN inference. The aim is to integrate, along with microarray data, other types of biological data sets, in order to enhance the discovery of parameters for a finely-grained GRN model . The method will avail itself of parallel computation in order to make the approach scalable to larger scale networks."

10.2.1.5 MUPHY

MUPHY[2] is a multi-physics/multi-scale code which combines microscopic Molecular Dynamics with a hydro-kinetic lattice Boltzmann method. This code was then applied to simulate the translocation of DNA through a nanopore, to work towards achieving very fast electronic genome sequencing. MUPHY has been ported and tested on large supercomputers, and scales well up to 10000s of cores. In addition, the code has been ported to run on GPUs (http://www.chem-quantum.info/scigpu/?p=229).

10.2.1.6 Voth group

The Voth group [3] has a key focus on the development of powerful multiscale theory and computationalmethods for complex biomolecular systems. These multiscale methods include systematic coarse-graining approaches, mesoscopic modeling, and multiscale bridging between all of the relevant scales.

10.2.2 Example application

See the systems biology application within MAPPER for an example multiscale application within this field.

10.2.3 Relevant multiscale literature references

[1] J.O. Dada and P. Mendes, Multi-scale modelling and simulation in systems biology, December 2010 DOI: 10.1039/c0ib00075b

[2] M. Bernaschi, S. Melchionna, S. Succi, M. Fyta, E. Kaxiras, J.K. Sircar, MUPHY: A parallel MUlti PHYsics/scale code for high performance bio-fluidic simulations, Computer Physics Communications, Volume 180, Issue 9, September 2009, Pages 1495-1502, ISSN 0010-4655, DOI: 10.1016/j.cpc.2009.04.001. (http://www.sciencedirect.com/science/article/B6TJ5-4W0SK3Y-1/2/c87a36b26733c1dd3c641c30f0b10ccd).

[3] Gary S. Ayton, and Gregory A. Voth Center for Biophysical Modeling and Simulation and Department of Chemistry, University of Utah, Multiscale Theory and Simulation. Website: http://www.chem.utah.edu/directory/faculty/voth.html.

10.3 Astrophysics

10.3.1 Description fo the user community

Astrophysics is an extensive scientific domain, with a multitude of active multiscale projects. Due to the inherent multiphysics nature of many astrophysical problems, a large number of astrophysical simulations already operate across multiple time and length scales. More recently there has been a movement from large monolithic codes towards more modular frameworks the combine multiple community codes or modules, such as AMUSE and FLASH. Here we will highlight a few major multiscale projects within the astrophysics domain:

10.3.1.1 Multiscale software frameworks

AAMUSE – http://www.amusecode.org

Perhaps the largest and most versatile multiscale framework currently in astrophysics is the Astrophysical Multipurpose Software Environment (AMUSE). The aim of AMUSE, as a successor of the MUSE framework [1], is to provide a software framework for multi-scale astrophysical simulations. Within this framework, existing codes are connected through a layer of Python and MPI to form coupled simulations. AMUSE can be used for parallel simulations, though it does not yet support simulations across multiple computational sites. There are numerous astrophysical scenarios in which AMUSE has been applied. One example is coupling a gravitational N-body simulation with a stellar evolution code to model both the dynamical movements and the aging of the stars in a star cluster. Another example involves using a high-accuracy N-body integrator to simulate the center of a galaxy, while using more approximate methods (e.g., Barnes-Hut tree integration) to simulate the outskirts.

FLASH - http://flash.uchicago.edu/website/home/

The FLASH code [3] couples hydrodynamics with magnetic fields and is applied to simulate the surfaces of compact stars such as white dwarves and neutron stars. The latest version, FLASH 3, consists of inter-operable modules that can be combined to generate different applications. The FLASH architecture allows arbitrarily many alternative implementations of its components to co-exist and interchange with each other, resulting in greater flexibility. Further, a simple and elegant mechanism exists for customization of code functionality without the need to modify the core implementation of the source.

10.3.1.2 Other multiscale projects

GADGET - http://www.mpa-garching.mpg.de/gadget/

GADGET [2] is a cosmological code which couples gravitational dynamics with smooth particle hydrodynamics. The code operates on a variety of physical scales and has been applied to perform simulations of very large cosmological systems. It should be noted that the gravitational dynamics part also uses a hybrid method, where short range forces are resolved using a Barnes-Hut tree algorithm and long range forces are resolved using particle-mesh integration. There are a variety of projects which adopted GADGET, some of which extended it to simulate a wider range of physical phenomena observed in the universe.

Starlab - http://www.manybody.org/manybody/starlab.html

Starlab[4] is a "monolithic" N-body integrator that has been extended to support the separate treatment of binaries on smaller scales, and to support the evolution of individual stars.

Multiscale simulations of Type Ia supernovae

Röpke et al [5] have performed multiscale simulations of thermonuclear supernovae. Their implementation uses a large eddy simulation (LES) strategy for treating turbulence effects and a level-set technique to represent the thin thermonuclear flames. They also developed a visualization engine which aims to allow interactive explorations of the simulation data using out-of-core algorithms.

N-body codes with regularization algorithm.

There are several N-body codes which handle close encounters using seperate algorithms. These algorithms operate on a different temporal and spatial scale, yet are directly added into the existing codes. Examples of N-body code with regularization schemes are NBODY7 [6] and phiGRAPE [7].

10.3.2 User needs

With such a large number of multiscale projects, the needs of the astrophysical community are diverse.

10.3.3 Main usage templates

Though there are several loosely coupled multiscale simulations described in the literature, most of the scenarios within astrophysics describe concurrent physical processes modelled by different solvers, and are therefore tightly coupled.

10.3.4 Short term user needs

Recent developments in astrophysics seem to move further towards hybrid simulations which encompass multiple physical processes. Also, observing the developments in AMUSE and FLASH there is a current trend away from monolithic large production codes to more modular software toolkits which can be applied for a variety of scientific purposes. The scale on which these codes run seems to remain limited at the moment, as the main focus still lies on the improvement of these recently emerged frameworks. So far, only cosmological dark matter simulations have been frequently run on supercomputers.

10.3.5 Long terms user needs

The aim to perform a large multiscale simulation of a complete galaxy has been expressed nearly 10 years ago in MODEST (http://www.manybody.org/modest/) collaboration, and has been reiterated and further worked out in [8]. Here, Hoekstra et al. describe a simulation of a Virtual Galaxy which will require Petascale resources. With Petascale machines now in existence and funds received for astrophysical projects working towards this goal, it is not unreasonable to expect simulations of galaxies to increase in scale once the multiscale software frameworks have fully matured.

10.3.6 Example external application: the Virtual Galaxy

This application involves a high accuracy simulation of a Galaxy, including the modeling of star clusters, field stars and molecular clouds. The main motivation for choosing this example, though 4 years old, is the fact that it concerns a popular topic in astronomy, it is well documented in public literature [8] and it has yet to be realized in full.

10.3.6.1 Description of the example application and its subcodes

"A grand challenge in computational astrophysics, requiring at least Petascale resources, is the simulation of the physics of formation and evolution of large spiral galaxies like the Milky-way. This requires the development of a hybrid simulation environment to cope with the multiple time scales, the broad range of physics and the sheer number of simulation operations [9] [10]. The nearby grand design spiral galaxy M31 in the constellation Andromeda, as displayed below, provides an excellent birdseye view of how the Milky-way probably looks."

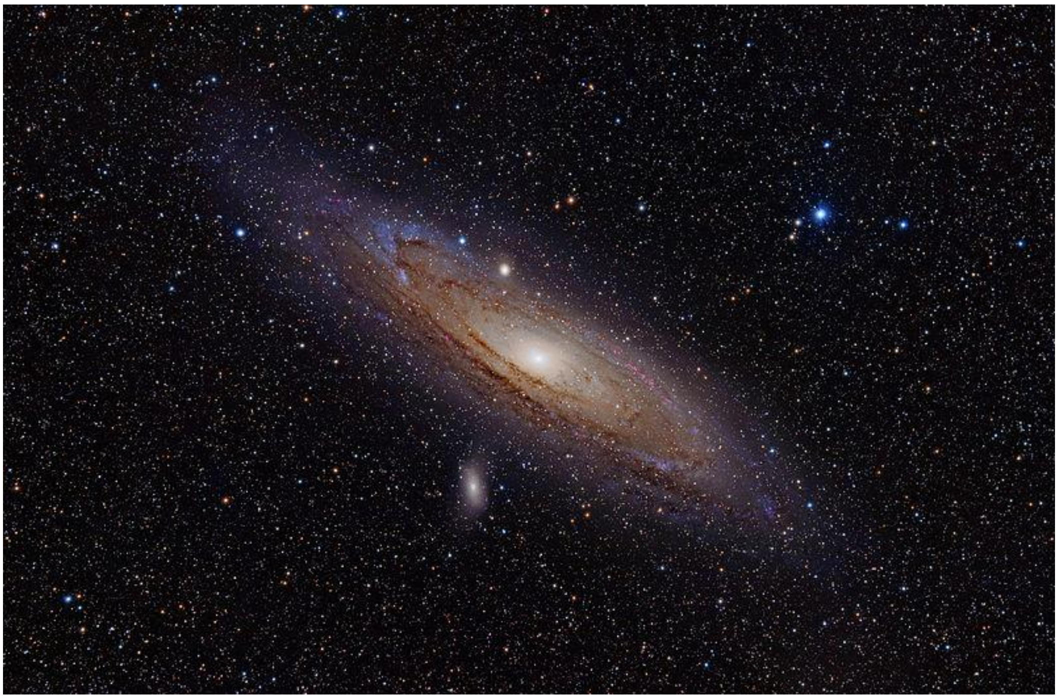


Figure 18: Andromeda Galaxy

The Virtual Galaxy requires the coupling between a number of solvers to perform. These include N-body codes to model the dynamic evolution of star clusters and field stars within the galaxy, stellar evolution kernels to model the nuclear evolution of individual stars as well as adaptive mesh refinement and smoother particle hydrodynamics simulations to model the molecular gas clouds and the process of star formation. The individual codes have diverse hardware requirements. For example, stellar evolution routines are often embarrassingly parallel for different stars (except in the case of e.g. very tight binaries) and can be distributed across multiple machines while N-body simulations require intense intercommunication and benefit strongly from using specialized (GPU) hardware.

10.3.6.2 Short description of the coupling mechanism between the codes

The coupling within the Virtual Galaxy is essential for the overall simulation, and contains aspects of both tight and loose coupling. The stellar evolution may be coupled to the other simulations using a loose coupling mechanism, but the modeling of a galaxy using both N-body and hydrodynamics requires both codes to be run concurrently and tightly coupled.

10.3.6.3 Scale Separation Map

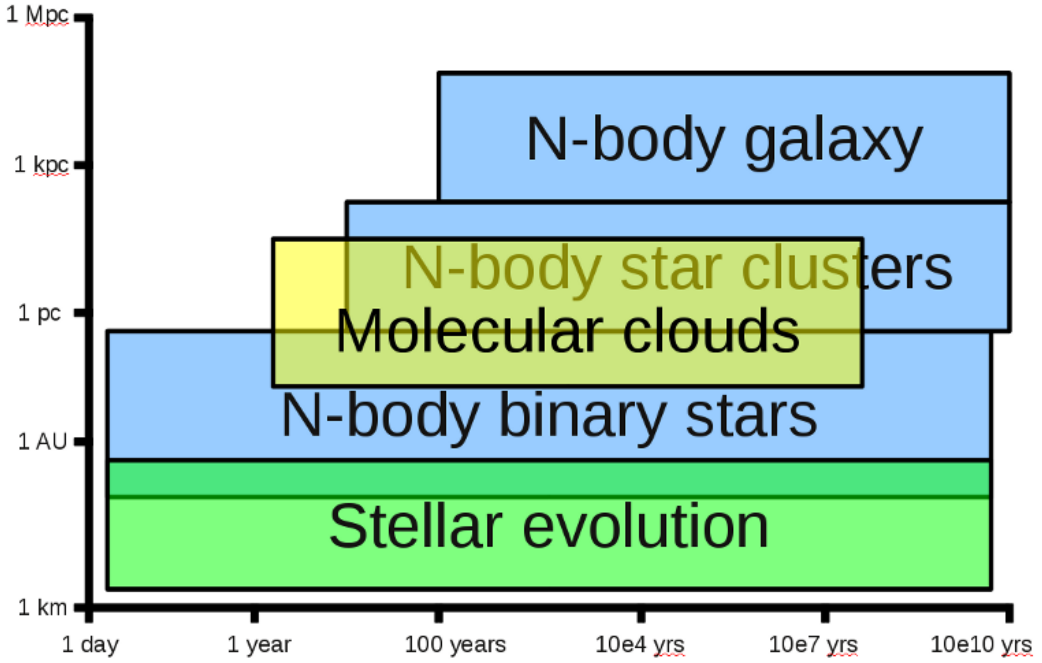


Figure 19: A scale separation map of a Virtual Galaxy application. The estimates for the time and length scales of molecular cloud simulations were based on Klessen et al. [9].

10.3.6.4 Computational requirements

Below we provide an estimate of the computational requirements of this scenario. As the full simulation has not be put into practise yet, these numbers should be treated as rough indicators.

For the requirements, we assume the following scenarios: Now: ~100.000 stars using direct-method, coupled with a Barnes-Hut tree code for the other stars and a basic stellar evolution. Required: The scenario as described in [8].

Requirement	State of the art	Required capabilities
# of cores	~100	100000s
memory per core [MB]	200	1000
runtime	1 month	1 month
coupling frequency	1/second	1/second
max. data size (coupling)	1MB	100MB
max. data size (output)	50MB	10GB

10.3.6.5 Relevant multiscale literature references

[1] Portegies Zwart, S et al., 2008, A multiphysics and multiscale software environment for modeling astrophysical systems, New Astronomy, Volume 14, Issue 4, p. 369-378.

[2] Springel, V., 2005, The cosmological simulation code gadget-2, Monthly Notices of the Royal Astronomical Society Volume 364, Issue 4, pages 1105–1134, December 2005.

[3] Dubeya, A., Antypasb, K., Ganapathyc, M., Reida, L., Rileyd, K., Sheelerd, D., Siegeld, A. and Weidea, K., 2010, Extensible component-based architecture for FLASH, a massively parallel, multiphysics simulation code, Volume 35, Issues 10-11, Pages 512-522.

[4] Portegies Zwart, S., McMillan, S., Groen, D., Gualandris, A., Sipior, M. and Vermin, W., A parallel gravitational N-body kernel, 2008, New Astronomy, Volume 13, Issue 5, Pages 285-295.

[5] Röpke, F. and Bruckschen, R., 2008, Thermonuclear supernovae: a multi-scale astrophysical problem challenging numerical simulations and visualization, New J. Phys. 10 125009.

[6] Mikkola, S. and Merrit, D., 2006, Algorithmic regularization with velocity-dependent forces, Monthly Notices of the Royal Astronomical Society, Volume 372, Issue 1, pages 219–223.

[7] Harfst, S., Gualandris, A., Merritt, D., Mikkola, S.,2008, A hybrid N-body code incorporating algorithmic regularization and post-Newtonian forces, Monthly Notices of the Royal Astronomical Society, Volume 389, Issue 1, pages 2–12.

[8] Hoekstra, A., Portegies Zwart, S., Bubak, M. and Sloot, P., 2007, Towards Distributed Petascale Computing, In: D. Bader (Ed.) Petascale, Computing: Algorithms and Applications, Chapman & Hall / CRC Press, Taylor and Francis Group

[9] Klessen, R.; Krumholz, M.; Heitsch, F. Numerical Star-Formation Studies -- A Status Report. eprint arXiv:0906.4452, 2009.

10.4 Engineering and material sciences

10.4.1 Description of the user community

Engineering and material sciences applications are inherently multiscale. The macroscale level may be the level in which we can see and interact with materials, but it is the smaller levels of interaction, whether they are on the atomic and molecular scale, which play an important role in their properties. An active area of research is linking our understanding of the physical world at very short scales with the observable behaviour at the macroscale. The applications are incredibly varied; examples include composite and polymer materials, crystalline microstructures, nanomaterials, opto electronics for the development of materials with particular optical, magnetic and electrical properties, fracture propagation and flow through porous media, among many others.

The International Journal of Multiscale Engineering [1] is a new journal dedicated to multiscale simulation, and several universities have set up Centers for Multi-scale Modeling and Simulation. A general review of multiscale simulation in science in engineering can be found in Multiscale Methods, bridging the scales in science and engineering [2]. In the rest of this article, we identify key topic areas in engineering and material science, which reflect the external communities. A full list of projects and researchers active in these areas is too large to list here; instead, we have identified a number of example projects currently developing multiscale coupling schemes. Many of these topics feature nanoscales and/or nanotechnology. Nanotechnology is the creation and utilization of functional materials, devices, and systems with novel properties and functions that are achieved through the control of matter, atom by atom, molecule by molecule or at the macromolecular level. In a sense this represents the ultimate multi-scale engineering field by virtue of the enormous range of scales involved.

10.4.1.1 Nanoscale experiments and model development for materials.

RADINTERFACE (Multiscale Modelling and Materials by Design of interface-controlled Radiation Damage in Crystalline Materials)[3].

In the RADINTERFACE project, multiscale simulation is used to predict the mechanical behaviour (macroscale) of materials that have undergone radiation damage (microscale). Such material damage can lead to failure and thus is of critical importance to lifetime and safety within nuclear reactors. Observations and physical models have shown that the most important damage contributions arise from point defect localization –leading to void swelling- and creep. It was recently found that void swelling can be prevented via use of non coherent heterophase interfaces. It is very likely that other interface types may exhibit similar trends. Unfortunately, no tool is available to generally predict the effect of interface composition (monophase, heterophase) and structure (geometry, roughness) on its propensity to resist radiation damage (both via defect localization and creep). In RADINTERFACE project, this problem will be addressed using the familiar micro-macro coupling scheme. The lowest level is ab initio methods (computing the electronic degrees of freedom), classical molecular dynamics and continuum modeling. Multiscale issues involve the coupling of atomistic and continuum descriptions. Using the MAPPER based terminology, the multiscale scheme used in this project is “loosely coupled”. The project will be validating the multiscale models through mechanical testing and characterization of a wide array of materials undergoing irradiation.

10.4.1.2 Simulation of material behavior from the atomistic to the continuum level.

M3-2S (Multiscale Modelling for Multilayered Surface Systems)[4]

There is a demand for high performance engineering components which can withstand severe working conditions, for new tools in new micro techniques and for light weight alloys. The design of multi-layered surfaces for theseapplications requires surface modelling techniques to provide reliable and high performance surfaces across the scale of multiscale systems from nanometre to millimetre. The aim of this project is to establish integrated, generic, robust multiscale materials modelling techniques for the design and performance prediction of multilayer surface systems, under different working conditions. This project is developing the coupling between an a molecular description (molecular dynamics) with a finite element and finite volume description (where the molecular detail is no longer present) for these multlayered systems.

A similar project, coupling atomic level resolution with finite element analysis, is underway at Sandia National Labs (DoE): multiscale schemes for the predictive description and virtual engineering of materials. (Greg J. Wagner / Anatole Lilienfeld). This project is examining the fracture and failure of solids, surface interactions, and design of nano-structured materials such as thermoelectrics.

10.4.1.3 Simulation of mass, energy, and momentum transport processes from the nanoscale to the continuum level.

The building physics group at Katholieke Universiteit Leuven is using multi-scale approaches to model transport, chemical, physical and damage related phenomena in porous materials. The aim is to generate incremental constitutive laws for coupled multi-physical problems, which can be used in (continuum) macroscopic models. Transport in porous materials requires multi-scale (network) modeling, while transport in cracks are modeled by hybrid multi-scale approaches. Heat, air and mass transport in boundary layers is modeled by a multi-scale approach. We will also consider nonlinear mechanical behaviour (e.g. softening damage, permanent deformations, etc.) originating from behaviour of cracks and defects situated at lower scales. Coupled physical, chemical and mechanical phenomena generate complex coupled poromechanical constitutive laws, which can only adequately be described by multi-scale modeling.

The previous topic areas have a multiscale scheme with information passing between levels in a hirechical scheme (see the scale separation map below). However, a very active research area is the class of multiscale approaches for systems whose behaviour depends on physics at multiple scales and require a concurrent resolution of multiple scales. Various mathematical models are required to pass information between the sub domains. Friction and fracture are among many problems falling into this category. In fracture, the crack tip bond breaking can be described with a quantum- mechanical model of bonding, while the rest of the sample is described with empirical potentials. In friction, it might be necessary to describe the surface interaction using quantum-chemical approaches while using continuum elasticity to simulate the contact forces. For these types of problems, multiple scales have to be simultaneously resolved in different portions of the problem domain. Although the research in this area is vast, one method we would like to highlight is “Learn on the Fly”‘, a hybrid ab initio-atomistic computational schemes introduced in recent years, which has been used to model large materials systems, and in particular the crack propagation instabilities in silicon [5].

10.4.2 User needs

Tha applications in the engineering and materials sciences are frequently run on large scale infrastructures, and most multiscale scenarios involve the use of loosely coupled simulations, where the statistical sampling of one subcode may be used as the input for another. However, a number of engineering and materials science projects also feature tightly coupled multiscale simulations.

10.4.3 Example application

The nanomaterials application within MAPPER is a good example of a multiscale application in the engineering and material sciences domain.

10.4.4 Relevant multiscale literature references

[1] The International Journal of Multiscale Engineering – http://www.begellhouse.com/journals/61fd1b191cf7e96f.html.

[2] Jacob Fish. Multiscale Methods: Bridging the Scales in Science and Engineering. Oxford University Press, 2010.

[3] RADINTERFACE – http://www.materials.imdea.org/Research/Projects/RADINTERFACE/tabid/3423/Default.aspx

[4] M3-2S - http://www.m3-2s.bham.ac.uk/.

[5] Kermode J R, Albaret T, Sherman D, Bernstein N, Gumbsch P, Payne M C, Csa´nyi G and De Vita A 2008 “Low-speed fracture instabilities in a brittle crystal” Nature 455 1224–7.

10.5 Biomedical sciences

10.5.1 Description of the user community

Biological systems span many orders of magnitude through the scales in a continuous way, from the smallest microscopic scales up to the whole body. The sequence from the genome, proteome, metabolome, physiome to health comprises multi-scale, multi-science systems [2] [3] [4]. A pedagogical introduction to this concept and the field of multi-scale modelling in biology is provided by Schnell et al. [5]. An extensive prior review of multiscale modelling in computational biomedicine has been published by Sloot et al. [1]. The description which we provide here is largely based on that review.

10.5.1.1 Virtual Physiological Human

The Virtual Physiological Human (VPH) Initiative is a large and active community in the multiscale computing domain. Within the VPH, which is also part of MAPPER, and its subprojects, multiscale simulations and models are of fundamental importance. However, not all subprojects directly couple different time and space scales in their models. The VPH is represented in Europe (VPH-NoE, http://www.vph-noe.eu), USA (Physiome, http://www.physiome.org), and Japan (e.g. the RIKEN group of Himeno, http://www.riken.jp/engn/r-world/research/lab/rpcs/next/index.html). Some of the subprojects within VPH are:

- euHeart (http://www.euheart.eu/) - heart modelling
- VPHOP (http://www.vphop.org/) - osteoporosis
- RTreat (http://artreat.org/) - atheroscleroris
- VPH2 (http://www.vph2.eu/) - heart modelling
- ContraCancrum (http://www.contracancrum.eu) – malignant tumour modelling

10.5.1.2 Multiscale modelling of sub-cellular processes

Burrage et al. [6][7] describe their approaches to multi-scale modelling for sub-cellular processes in the cell and on the cell membrane. Southern discusses in some detail the case of multi-scale modelling of ion channels [4] coupling simulations on the molecular scale, to obtain diffusion coefficients of potassium ions with Brownian dynamics simulations to obtain ion fluxes. In these examples the main assumption is that the time scales are separated, such that lower-level processes are much faster, implying that the lower-level processes are in quasi-equilibrium with the slower higher level processes and can be included at the higher level via e.g. constitutive equations or force fields. This type of multi-scale modelling, where microscopic processes (small spatial scales, fast dynamics) are coupled to macroscopic processes (large spatial scales, slow dynamics) has received a lot of attention in the literature.

10.5.1.3 Multiscale modelling of thrombus development

Another multiscale model has been proposed by Xu et al. [8], in this instance for thrombus development. They couple a discrete Cellular Potts model of cellular behaviour to continuous models of blood flow and biochemical reactions. They assume that the growth of a thrombus is slow process, allowing a time splitting technique to be used where the transport equations are solved first, providing boundary conditions to the growth model. This is then iterated over many time steps of the slow model. This allows them to study in detail the growth of the thrombus, where initially activated platelets arrive at the front side of the thrombus, but as the thrombus grows, the flow fields change and activated platelets and blood cell clusters are pushed back and attached to the backside. This could explain inhomogeneity and later thrombus instability.

10.5.1.4 IUPS Physiome project

The most advanced multi-scale models and simulations in computational biology to date are those emerging from the IUPS Physiome project [9]. Here we will only mention a few highlights from the literature dealing specifically with multi-scale modelling. The Physiome aims at developing a standardized computational framework for human physiology. The main idea behind the multi-scale modelling in the physiome is ‘the application of continuum field concepts and constitutive laws, whose parameters are derived from separate, finer-scale models', which is 'the key to linking molecular systems biology (with its characterization of molecular processes and pathways) to larger-scale systems physiology (with its characterization of the integrated function of the body's organ systems)' [10].

This is again the micro-macro coupling as discussed earlier. Most advanced is the multi-scale model of the heart (see [3] [11] and also [12] for deeper discussion on a multi-scale modelling paradigm to exploit temporal scale separation in systems of coupled ODEs). For other organ systems multi-scale models are on the drawing table, and with the current wave of projects in e.g. the Virtual Physiological Human initiative (see http://ec.europa.eu/information_society/activities/health/research/fp7vph/index_en.htm), one may expect to see rapid progress. For instance, in case of the pulmonary system, detailed models on the organ level (anatomically based models of the lung, airway and vascular trees), the tissue level (tissue mechanics, microcirculatory flow) and cell levels (erythrocyte gas kinetics) are available [13] but their integration into a multiscale model for a virtual lung is only beginning to appear in the literature [13] [14]. Another challenging organ system is the musculoskeletal system. The case of multi-scale modelling of the human femur is discussed in [15], again demonstrating that on many levels detailed models are available, and that linking them together through the scales is where the current challenge lies. Interestingly, this article nicely demonstrates the 'middle out' approach (as opposed to bottom–up or top–down), where one picks the biological level of interest as the starting point, and then works up and down the scales as required [16].

10.5.1.5 Example Project: ViroLab: Transmission of Resistance in HIV

ViroLab (http://www.virolab.org/) [17][18][19] was developed to give medical doctors a decision support system to rank drugs targeted at patients and to provide virologists an advanced environment to study trends on an individual, population and epidemiological level. Virolab is a multi-scale modelling, simulation and datamining environment for infectious diseases, going from molecule to man and back. Statistical and immunological models are needed to study the dynamics of the HIV populations and molecular dynamics models to study drug affinities, in addition to rule-based and parameter-based decision support. We added cellular automata (CA) and molecular dynamics modelling of HIV infection and AIDS onset. All these models operate on a large range of length and time scales as sketched in the scale separation map.

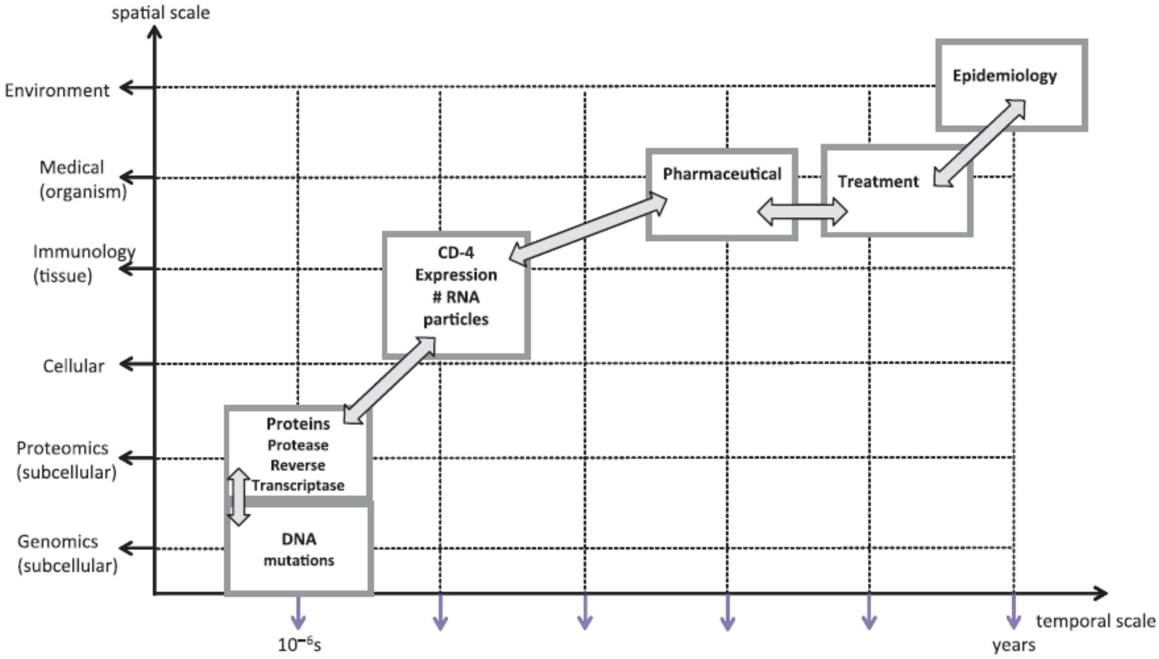


Figure 20: Scale separation map of a simulation of resistance transmission in HIV (source: Sloot et al. [1]).

10.5.2 User needs

The biomedical sciences community has numerous examples of successful multiscale simulations. This particular community strongly emphasizes on integrating the vastly different spatial and temporal scales and building up the multiscale simulation infrastructure which is ready to assist clinicians and biomedical researchers in their work. The biomedical sciences community not only requires a large-scale compute infrastructure (e.g., to execute simulations with high resolution and to transport large data sets), but also faces a number of usability and privacy constraints.

Two major user groups in the biomedical community are the scientific researchers and the clinicians. These two groups have differing, but overlapping needs. Scientific researchers require a robust compute, networking and storage infrastructure which provides complete support for their multi-scale applications. However, for clinicians it is even more essential that, despite their relative unfamiliarity with large e-infrastructures, they are able to access the infrastructure and run the simulations they require. This requirement can only be met if the simulation environment is sufficiently easy to use. The biomedical community is also subject to strong security policies regarding medical data. A large portion of medical data either cannot be shared between organizations at all, or needs additional security and provenance measures to be shared on a distributed environment.

10.5.3 Main usage templates

Most of the multiscale simulations in the biomedical sciences community are at least partially tightly coupled, and many of them involve concurrent execution and a coupling between three or more subcodes. However, there are also a large number of biomedical models which have yet to be integrated in a multiscale environment.

10.5.4 Short term user needs

The biomedical sciences community is well represented within MAPPER, in part because MAPPER's goal of forging a robust infrastructure for multiscale simulations is strong and short term user need for the biomedical community as well. The infrastructure we are building within MAPPER will be essential for deploying the numerous biomedical multiscale applications, and through that be of importance to the biomedical sciences community as a whole.

Another challenge which biomedical science projects face is the evolution from proof-of-concept multiscale simulations in the academic environment to validated multiscale problem solvers which have a proven clinical benefit. As a result, several projects are now in the process of formulating methods to validate their simulation results against the results of physical clinical experiments.

10.5.5 Long term user needs

The validation of multiscale simulation is a crucial phase towards introducing such methods in the medical environment, though validating simulation results against physical experiments is a complicated and time-consuming procedure which requires the close collaboration between academic and clinical institutions. A long term need of the community is to prove that the developed multiscale biomedical simulations indeed have added clinical value. Once this need has been met, the next step may well be to implement the use of biomedical simulations for practical use in hospitals and other medical institutions.

10.5.6 Example application

The In-stent restenosis and HemeLB applications within MAPPER are clear examples which originate from the biomedical community.

10.5.7 Relevant multiscale literature references

[1] Sloot, P. and Hoekstra, A., 2010, Multi-scale modelling in computational biomedicine, Briefings in Bioinformatics, volume 11, number 1, pages 142--152.

[2] Finkelstein A, Hetherington J, Li L, et al. Computational challenges of systems biology. IEEEComput 2004;37:26–33.

[3] Noble D. Modeling the heart—from genes to cells to the whole organ. Science 2002;295:1678–82.

[4] Southern J, Pitt-Francis J, Whiteley J, etal. Multi-scale computational modelling in biology and physiology. Progr BiophysMolecul Biol 2008;96:60–89.

[5] Schnell S, Grima R, Maini PK. Multiscale modeling in biology. American Scientist 2007;95:134–42.

[6] Burrage K, Hancock J, Leier A, et al. Modelling and simulation techniques for membrane biology. Brief Bioinform 2007; 8:234–44.

[7] Burrage K, Tian T, Burrage P. A multi-scaled approach for simulating chemical reaction systems. Progr Biophys Mol Biol 2004;85:217–34.

[8] Xu Z, Chen N, Kamocka MM, et al. A multiscale model of thrombus development. JRoy Soc Interface 2008;5:705–22.

[9] Hunter PJ, Crampin EJ, Nielsen PMF. Bioinformatics, multiscale modeling and the IUPS Physiome Project. Brief Bioinform 2008;9:333–43.

[10] Hunter P, Nielsen P. A strategy for integrative computational physiology. Physiology 2005;20:316–25.

[11] Plank G, Zhou L, Greenstein JL, et al. From mitochondrial ion channels to arrhythmias in the heart: computational techniques to bridge the spatio-temporal scales. PhilTrans RSocA 2008;366:3381–409.

[12] Burrowes KS, Swan AJ, Warren NJ, et al. Towards a virtual lung: multi-scale, multi-physics modelling of the pulmonary system. PhilTrans RSoc A 2008;366:3247–63.

[13] Chakraborty S, Balakotaiah V, Bidani A. Multiscale model for pulmonary oxygen uptake and its application to quantify hypoxemia in hepatopulmonary syndrome. J Theoret Biol 2007;244:190–207.

[14] Cristofolini L, Taddei F, Baleani M, et al. Multiscale investigation of the functional properties of the human femur. PhilTrans RSoc A 2008;366:3319–41.

[15] Noble D. The Music of Life. Oxford, New York: Oxford University Press, 2006.

[16] Sloot PMA, Coveney PV, Bubak MT, et al. Virolab: a collaborative decision support system in viral disease treatment. RevAntiviralTher 2008;3:4–7.

[17] Sloot PMA. VIROLAB: from the molecule to the man. eStrategies Proj 2008;53–5.

[18] Sloot PMA, Tirado-Ramos A, Altintas I, et al. From molecule to man: decision support in individualized E-health. IEEEComp 2006;39:40–6.

10.6 Environmental sciences

10.6.1 Description of the user community

Environmental science is multiscale and multi-process in essence. A numerical challenge to address the environment as a whole is to combine very different phenomena that are coupled and that interact at different scales. This applies for instance to ecology studies, climate modeling, geosciences, hydrology, etc.

10.6.1.1 Hydrology

Several multiscale projects have been undertaken in the hydrology domain. Within the MAPPER project, UNIGE has developed multiscale models for management of a network of irrigation canals. The problem to be solved is to define appropriate actions (e.g. opening and closing gates) that need to be taken to always guarantee an adequate water supply throughout the canal system, whatever the external demands or perturbations can be, and respecting constraints such as water height [1] [2]. Other multiscale projects include simulating the effects of beach height on tsunamis [3], modelling flood predictions [4], and simulations of watersheds [5].

10.6.1.2 Weather and climate modelling

Weather prediction is often performed using multiscale simulations. These simulations take into account global effects in terms of climate and ocean currents, but also interpret weather data on the mesoscale to perform predictions. The details available on the exact nature of these simulations are somewhat limited, as many of these predictions are performed by commercial companies. Examples of weather prediction include, but are by no means limited to the prediction of hurricanes [6], the prediction of mesoscale weather [7] and air quality [8].

10.6.1.3 Earth Modelling (ENES)

The European Network for Earth System Modelling (ENES, [10]) is a large consortium focused on developing of a European network for Earth system modelling. ENES intends (among other things) to facilitate focused model intercomparisons, to encourage exchanges of software and model results, and to help in the development of high performance computing facilities dedicated to long high-resolution *multi-model* ensemble integrations. The ENES consortium has helped to establish the ENSEMBLES [11] project, where a consortium of 70 partners use multiscale ensemble simulations, varying across temporal and spatial scales, to simulate the Earth system for a range of applications. These include climate predictions, as well as the resolution of physical, chemical, biological and human-related feedback processes in the Earth System. This system includes water resource use, land use, air quality issues, as well as carbon cycle feedbacks.

10.6.1.4 Earth Modelling (Bespoke Framework Generator)

In the context of Earth modeling, the Manchester University has developed the bespoke software and methodological framework (http://intranet.cs.man.ac.uk/cnc/projects/bfg.php). According to their website: “The Bespoke Framework Generator (BFG) is a prototype implementation of the Flexible Coupling Approach (FCA). The BFG specifies single model rules to which a conformant model implementation must adhere; it also defines XML schemas to capture metadata describing the conformant models, their scientific composition and their deployment onto resources. The BFG engine (written in xsl) then processes the resultant (user specified) XML, producing appropriate "wrapper code" within which the models can execute.

10.6.2 User needs

10.6.2.1 Main usage templates

A large number of multi-scale applications within the environmental sciences seek to capture concurrent processes across multiple scales, and are therefore tightly-coupled. However, weather simulations are also frequently run in loosely-coupled ensembles to achieve better statistical accuracy (e.g., see [9]).

10.6.2.2 Short and Long term user needs

The user needs within the environmental sciences community are widespread, with numerous organizations running their simulations on self-purchased facilities. It is therefore difficult in this particular case to formulate concise user needs for the community as a whole.

10.6.3 Example application

The Hydrology application within MAPPER is a clear example of a multiscale application within the Environmental Sciences.

10.6.4 Relevant multiscale literature references

[1] Pham van Thang, Bastien Chopard, Laurent Lefevre, Diemer Anda Ondo, and Eduardo Mendes. Study of the 1d lattice boltzmann shallow water equation and its coupling to build a canal network. Journal of Computational Physics, 229(19):7373 -7400, 2010.

[2] Olivier Marcou, Samira El Yacoubi, and Bastien Chopard. A bi-fluid lattice boltzmann model for water ow in an irrigation channel. In Samira El Yacoubi, Bastien Chopard, and Stefania Bandini, editors, Cellular Automata, volume 4173 of Lecture Notes in Computer Science, pages 373-382. Springer Berlin / Heidelberg, 2006.

[3] Chao Ana and Yongen Cai. The effect of beach slope on the tsunami run-up induced by thrust fault earthquakes. Procedia Computer Science Volume 1, Issue 1, May 2010, Pages 645-654, ICCS 2010.

[4] Ben Gouldbya, Valeria Krzhizhanovskayab, and Jonathan Simma. Multiscale modelling in real-time flood forecasting systems: From sand grain to dike failure and inundation. Procedia Computer Science Volume 1, Issue 1, May 2010, Page 809, ICCS 2010

[5] Paul R. Ellera, Jing-Ru C. Chenga, Hung V. Nguyena and Robert S. Maiera. Improving parallel performance of large-scale watershed simulations. Procedia Computer Science, Volume 1, Issue 1, May 2010, Pages 801-808 ICCS 2010.

[6] Gopalakrishnan, S. G. Bacon, D. P. Ahmad, N. N. Boybeyi, Z. Dunn, T. J. Hall, M. S. Jin, Y. Lee, P. C. S. Mays, D. E. Madala, R. V.. An Operational Multiscale Hurricane Forecasting System. Monthly Weather Review - USA, 2002, vol 130; part 7, pages 1830-1847.

[7] 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.; Chandrasekar, V.. CASA and LEAD: Adaptive Cyberinfrastructure for Real-Time Multiscale Weather Forecasting. IEEE Computer, volume 39, issue 11, pages 56-64, 2006.

[8] Grell, G.; Fast, J.; Gustafson, W.; Peckham, S.; McKeen, S.; Salzmann, M.; andFreitas, S. On-Line Chemistry Within WRF: Description and Evaluation of a State-of-the-Art Multiscale Air Quality and Weather Prediction Model. Book chapter in Integrated Systems of Meso-Meteorological and Chemical Transport Models By Alexander Baklanov et al., 2011.

[9] (in Dutch) KNMI Verwachte ijsdikte/watertemperatuur Midden-Nederland - http://www.knmi.nl/waarschuwingen_en_verwachtingen/extra/pluimijsaangroei.html.

[10] European Network for Earth System Modelling – http://www.enes.org.

[11] ENSEMBLES – http://ensembles-eu.metoffice.com/.

11 Appendix C: Review on Software

The MAPPER software stack seeks to deploy a set of services to facilitate the execution of multiscale scientific applications, that is single applications composed of multiple different single scale models, with each model usually executed by a different application code. This set of services, building on computational resources from the EGI and PRACE initiatives, as well as testbed resources and services run by the MAPPER project, constitute a European wide infrastructure for multiscale modelling and science. These components are arranged in a hierarchical infrastructure, shown in the figure below.

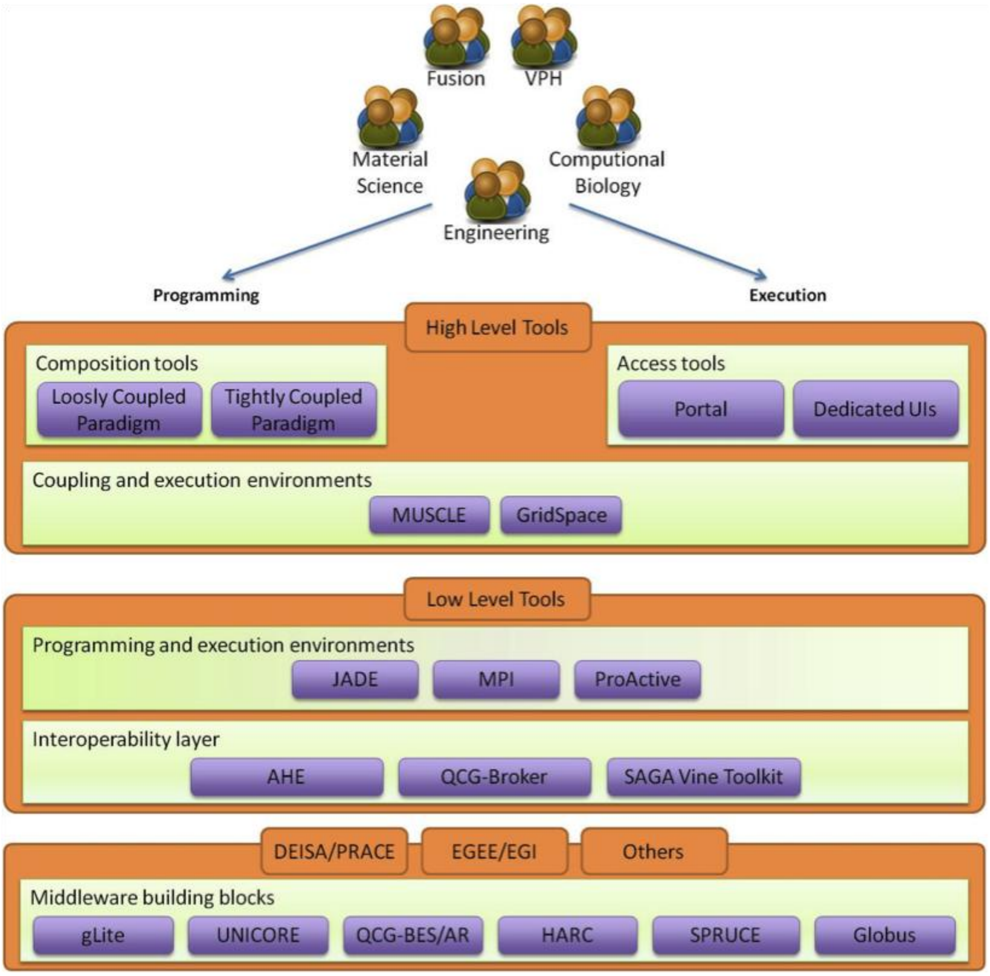
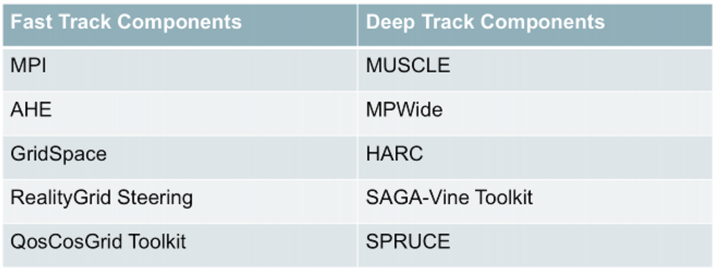


Figure 21: the MAPPER software aarchitecture

By necessity, some of these components run on on target compute resources (such as those operated by PRACE and EGI) and some components run at a higher level, on resources operated by the MAPPER project. In this section we review the main MAPPER software components needed to fully provision the MAPPER infrastructure.

11.1 Fast Track vs Deep Track

The MAPPER software components are further categorised into fast and deep tracks (shown in the table below), where the fast track components constitute the minimally effective infrastructure requred by the MAPPER project in order to execute basic multiscale applications, and the deep track components represent the full system infrastructure, required to execute all application scenarios. The MAPPER project is organized in such a way to initially deploy the fast track components so an initial infrastructure is available, then subsequently to adapt and deploy the deep track components.



Fast Track Components	Deep Track Components
MPI	MUSCLE
AHE	MPWide
GridSpace	HARC
RealityGrid Steering	SAGA-Vine Toolkit
QosCosGrid Toolkit	SPRUCE

Figure 22: MAPPER Fast and Deep track classification.

11.2 The Application environment

The Application Hosting Environment, AHE, developed at University College London, provides simple desktop and command line interfaces, to run applications on resources provided by national and international grids, in addition to local departmental and institutional clusters, while hiding from the user the details of the underlying middleware in use by the grid. In addition, a mobile interface for Windows Mobile based PDAs is available, and an iPhone interface is in development. The AHE is able to run applications on both UNICORE and Globus grids, meaning that a user can use a single AHE installation to access resources from the UK NGS and DEISA for example. Development of an EGEE connector for AHE is currently underway.

The AHE is designed to allow scientists to quickly and easily run unmodified, legacy applications on grid resources, manage the transfer of files to and from the grid resource and monitor the status of the application. The philosophy of the AHE is based on the fact that very often a group of researchers will all want to access the same application, but not all of them will possess the skill or inclination to install the application on remote grid resources. In the AHE, an expert user installs the application and configures the AHE server, so that all participating users can share the same application. This community model draws a parallel with the modus operandi of numerous scientific application communities. A diagram of the AHE's architecture is given in the figure below.

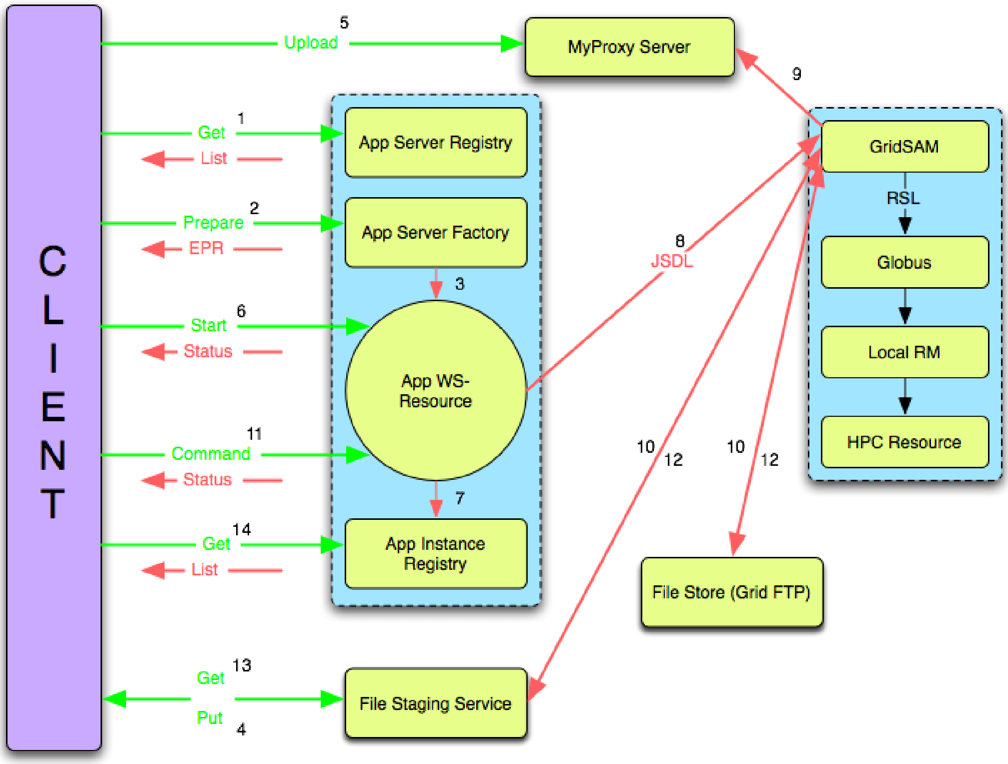


Figure 23: Architecture of the Application Hosting Environment

The AHE client is easily installed on an end user's machine, requiring only that they have a Java installation and an X.509 certificate for the grid, which they want to access. The client package contains both GUI and command line clients which interoperate, allowing jobs launched with the GUI client to be manipulated with the command line tools and vice versa, and also application workflows to be easily constructed.

A current EPSRC funded project in the UK, Es ist kein InterWiki-Link ttp definiert!, is developing the AHE further to remove the need for users to obtain individual grid certificates in order to use resources. Instead a shared certificate is stored in a central repository which users can access through username/password authorisation. This shared certificate can then be used to access grid resources. The technology is built in to the AHE client portal interface, meaning that access to ther certificate is seamless to the user.

11.1 RealityGrid Steering Toolkit

The RealityGrid steering toolkit is a middleware tool and associated libraries that allow parameters in simulation codes to be marked as steerable, which means that they can be monitored and modified in real time as the a simulation is running, and hence the course of the simulation changed. This is useful for a number of reasons: a scientist can monitor the progress of a simulation as it is running, and either stop it or alter the simulation parameters if a problem occurs, thus preventing valuable compute time from being wasted. Steering also allows the scientist to checkpoint a simulation as it is running should an interesting state occur, and then use this checkpoint to spawn new jobs and investigate the parameter state further. The UK EPSRC funded GENIUS project uses steering in a clinical setting to allow clinicians to interact with patient specific whole brain blood flow simulations, to check how modifying a certain part of the model (to simulate a surgical intervention for example) can cause increases in pressure and stress elsewhere in the model.

The RealityGrid toolkit uses a layer of web services to allow users to interact with simulations. Simulation codes publish their steerable parameters to these services, and users use lightweight client desktop tools to check on and modify the state of parameters. The architecture of the RealityGrid system is shown in the figure below. Since the RealityGrid services provide a consistent interface for the exchange of simulation parameters, the RealityGrid toolkit can also be used to couple models of different scale, using different simulation codes and potentially running on different computational resources.

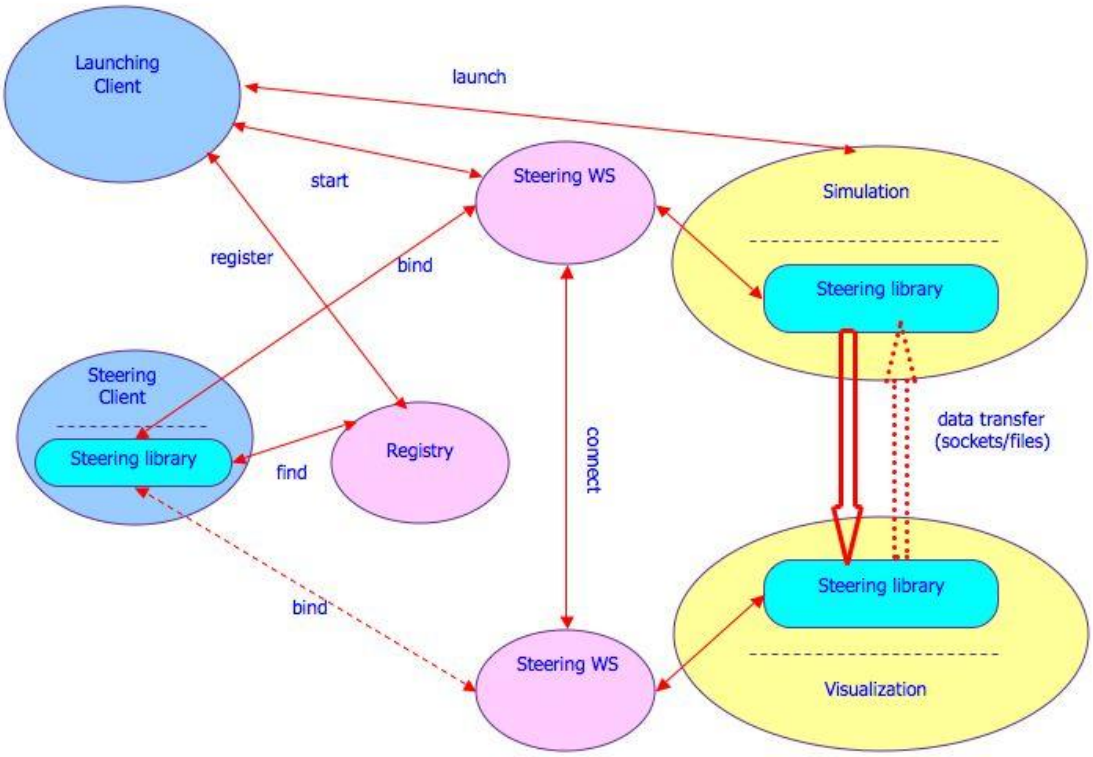


Figure 24: Architecture of the RealityGrid Steering System

11.2 HARC

A problem faced by users trying to exploit grid infrastructures in novel ways, for example by performing computational steering or real time visualization, is the fact that the user often has to wait until their simulation has progressed through the machine's work queue and begun to run. The batch queue model employed by most computational resource providers gives users no guarantee of when their simulation will run. HARC, the Highly Available Robust Coscheduler, developed at Louisiana State University, empowers users by allowing them to make reservations of time on grid resources, guaranteeing when their simulation will run. If simulation is to be scheduled in to the clinical workflow this capability is essential, since a clinical user will need to schedule simulation around their clinical practice and surgical interventions. The ability to co-reserve is also essential when running coupled models across different computational resources.

11.3 SPRUCE

The traditional high performance computing batch queue model does not allow for simulations to be prioritized by their urgency. Typically a grid will provide general purpose resources to a wide range of different users. If these resources are to be used by clinicians in support of their clinical practice, especially in support of emergency medical intervention planning, then some way is needed of prioritizing clinical simulations above the normal workload on a computational resource. SPRUCE, developed at Argonne National Labs, is a tool which allows this to happen. Clinicians and other users with simulations that are considered an emergency are issued with SPRUCE tokens, which allows them to submit emergency jobs to a machine. The SPRCUE middleware takes care of running the job in a high priority mode, preempting the work that is already running/queued on the machine.

11.4 GridSpace

GSEngine (GridSpace 1), developed by the FP6 ViroLab project, consists of two important parts to automatically run and manage ViroLab virtual experiments. The first is an interpreter of the JRuby language in which experiments are written. The second part is the runtime library that provides all the specific functionalities of the virtual laboratory which fall beyond the capabilities of the interpreter's standard library. The runtime library provides a defined set of routines that allow the developer to perform the following actions:

- instantiate Grid Objects (in general, or ViroLab Gems in the case of tools of ViroLab origin) on a suitable computing element in the ViroLab infrastructure
- destroy previously created Grid Object instances
- execute remote operations on those instances with required input parameters and receiving specified output results
- store any important data outside the experiment execution machine in a safe Laboratory Data Store (not yet supported)

- access remote data sources in a unified way, especially the ViroLab Data Access system that integrates crucial ViroLab databases
- interact with experiement executor asking him to provide some input
- freely use any kind of functionality that the JRuby standard library provides

The library is loaded into the interpreter every time execution of a ViroLab experiment plan is commended. The Runtime is responsible for holding and managing the entire experiment execution context. This may involve taking care of unique execution identification, passing authorization tokens on the user's behalf and maintaining information regarding the state of the experiment run.

Furthermore, the GSEngine interpreter with its runtime library may be referred from within any arbitrary Java application via its API. The API provided is intended to be complied by several implementations. One can perform script interpretation within the JVM of calling application, while the other is allowed to call dedicated remotely staged service, namely GSEngineServer.

11.4.1 GridSpace 2

In version 2 GridSpace applied different approach to establish Virtual Laboratory-like environment. The GridSpace 2 is an evolution of the computing and data access platform for viral disease research originally implemented in the ViroLab project. Following extensive discussions with representatives of various fields of e-science we have reengineered our software to better suit the daily work habits of domain scientists. This group includes users who frequently conduct virtual experimentation, relying on applications which can easily be ported to distributed computing environments. We have also drawn upon the experience of to-date scientific computing projects and the issues which have emerged during the course of their implementation (see [4] for an insightful discussion of these issues).

The GridSpace 2 Experiment Workbench, a top-level user interface, is designed to suit the habits and requirements of such users and allow them to exploit of the potential of distributed computing platforms, including PL-Grid (http://www.plgrid.pl/en). The GridSpace2 bases upon the notion of exploratory programming where each experiment can be decomposed into a number of so-called snippets. Each snippet may be written in a different programming language; moreover, the Workbench enables its user to execute entire experiments or just selected snippets. In this way, time-consuming experiments do not have to be started from scratch each time a modification is made during development.

The web portal of Experiment Workbench assists its users in iteratively developing virtual experiments with the use of scripting languages, including Ruby, Python and Perl.

Underneath the Experiment Workbench the Experiment Execution Environment (as depicted in below) resides where snippets are evaluated. If needed the computation is submitted to underlying computation resources of cluster.

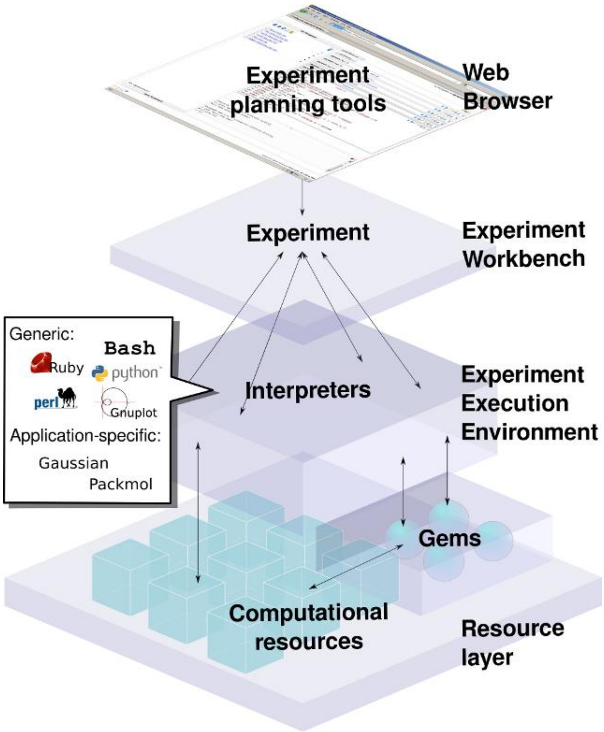


Figure 25: Architecture of GridSpace 2.

11.5 MPWide

MPWide is a communication library intended for message-passing between supercomputers. It was originally developed for the Gravitational Billion Body Project, where it was applied to run cosmological simulations across multiple supercomputers. The main added value of MPWide is its ability to explicitly define the communication method and configuration for individual network paths, and combine these customized configurations with an MPI-style API. For example, one can choose to use 128 parallel streams on a particular network path, adjust the tcp buffer sizes or apply packet pacing to improve the performance.

As a result, MPWide allows simulations to run efficiently across heterogeneously configured wide area networks (which are quite common nowadays). The library is written in C++ and requires only support of POSIX threads (a default in many OS distributions).

More information on MPWide can be found at
http://castle.strw.leidenuniv.nl/software/mpwide.html.

11.1 QosCosGrid Broker

QosCosGrid middleware was designed to build multilayered infrastructure being capable of dealing with computationally intensive large scale simulations. The developed middleware enables computing clusters in different administrative domains to be virtually welded into a single powerful computing resource that can be treated as a quasi-opportunistic supercomputer, which computational power exceeds the power offered by a single administrative domain (data center). The QosCosGrid infrastructure was primarly designed and developed in the QosCosGrid FP 6 project and then adopted and extended in PL-Grid national grid initiative.

11.1.1 QCG middleware architecture

The QosCosGrid middleware consists of two logical levels: grid level and administrative domain (AD) one. Grid-level services control, schedule and generally supervise the execution of tasks, which are spread between independent administrative domains. The Administrative domain (AD) represents a single organization (e.g. HPC center or research lab) participating in a virtual organization (VO) and sharing resources. The logical separation of administrative domains coresponds with the fact that they are possesed by different institutions. Each institution contributes its resources for the benefit of the entire VO, while controlling its own administrative domain and own resource allocation/sharing policies. The organizations agree to connect their resource pools exposed by AD-level services to a trusted “grid level” middleware which tries to achieve optimal resource utilization and ensure requested level of Quality-of-Service. The key component of every AD is the SMOA Computing service, which gives the remote access to queuing systems resources and features including the advance reservations, parallel execution environments - OpenMPI and ProActive with coordinators responsible for synchronization of cross-clusters executions and Data Transfer services for managing data. AD-level services, in turn, are connected to the Grid-level service, called Grid Resource Management System (GRMS)

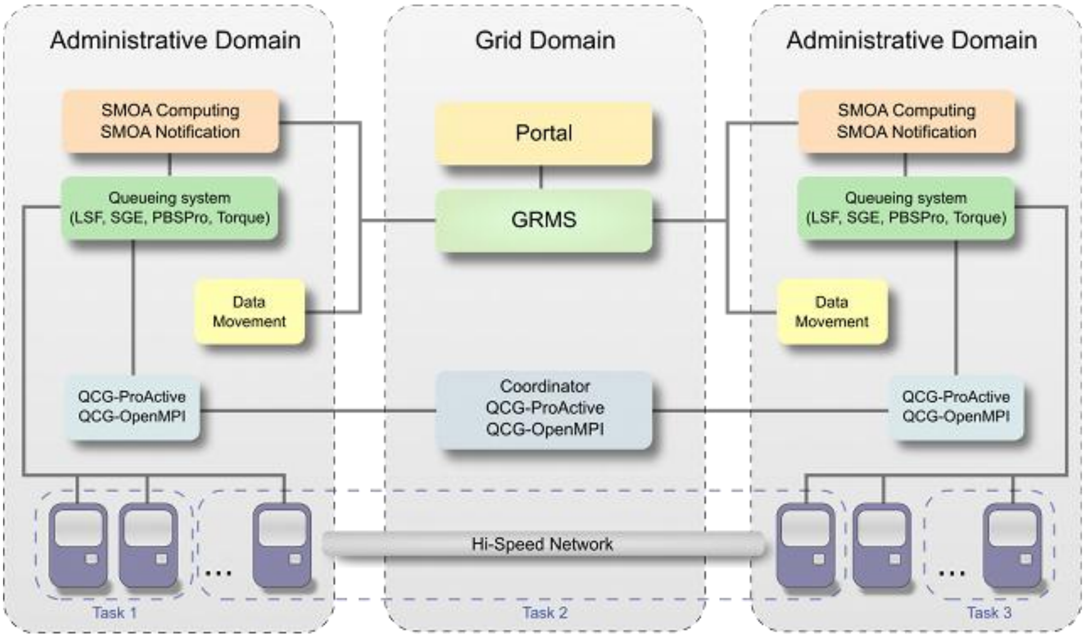


Figure 26: Architecture of QGC

11.1.2 SMOA Computing

SMOA Computing (the successor of the OpenDSP project) is an open architecture implementation of SOAP Web Service for multi-user access and policy-based job control routines by various Distributed Resource Management systems. It uses Distributed Resource Management Application API (DRMAA) to communicate with the underlying DRM systems.SMOA Computing has been designed and implemented in the way to support different plugins and modules for external communication. Consequently, it can be used and integrated with various authentication, authorization and accounting infrastructures and other external services. SMOA Computing service is compliant with the OGF HPC Basic Profile specification (a document which serves as a profile over the JSDL and OGSA® Basic Execution Service OGF standards). In addition it offers remote interface for Advance Reservations management, and support for basic file transfer mechanisms. The service was successfully tested with the following Distributed Resources Management systems:

- Sun Grid Engine (SGE),
- Platform LSF,
- Torque/PBSPro,
- PBS Pro,
- Condor,
- Apple XGrid.

The Advance Reservations capabilities were exposed for SGE, LSF and Maui (a scheduler that is typically used in conjunction with Torque) systems.

11.1.3 SMOA Notification

SMOA Notification is an open source implementation of the family of WS-Notification standards (version 1.3). It supports the topic-based publish/subscribe pattern for the asynchronous message exchange among Web Services and other involved entities. The main architecture of the notification system is based on a highly efficient, extended version of the NotificationBroker managing all items participating in notification events. Today, SMOA Notification offers sophisticated notification capabilities, e.g. topic and message content notification filtering, pull and push styles of transporting messages, and it was successfully integrated with different communication protocols as well as various Web Services security mechanisms. The modular architecture of SMOA Notification also provides a great opportunity for developers to build new extensions and plugins to meet other specific requirements. Within the QosCosGrid project the SMOA Notification Provider service is used for brokering notification messages about the job state changes. All instances of the SMOA Computing services act as the information Producers while the GRMS service as the Consumer.

11.1.4 Grid Resource Management System

The Grid Resource Management System (GRMS) is an open source meta-scheduling system, which allows developers to build and deploy resource management systems for large scale distributed computing infrastructures. The GRMS, based on dynamic resource selection, mapping and advanced scheduling methodology, combined with feedback control architecture, deals with dynamic Grid environment and resource management challenges, e.g. load-balancing among clusters, remote job control or file staging support. The main goal of the GRMS is to manage the whole process of remote job submission to various batch queuing systems, clusters or resources. It has been designed as an independent core component for resource management processes which can take advantage of various low-level Core and Grid Services and existing technologies, such as SMOA Computing and Notifications or GridFTP, as well as various Grid middleware services, e.g. Data Management Service, authorization service and more. All these services working together to provide a consistent, adaptive and robust Grid middleware layer which fits dynamically to many different distributing computing infrastructures enabling large scale simulations and to ensure requested Quality of Services.

One of the main assumptions for GRMS is to perform remote jobs control and management in the way that satisfies Users (Job Owners) and their applications requirements as well as constraints and policies imposed by other stakeholders, i.e. resource owners and Grid or Virtual Organization administrators. Simultaneously, Resource Administrators (Resource

Owners) have full control over resources on which all jobs and operations will be performed by appropriate GRMS setup and installation. Note that GRMS together with Core Services reduces operational and integration costs for Administrators by enabling Grid deployment across previously incompatible cluster and resources.

Addressing the need of large scale complex simulations concerning as well the computational power, that in many cases can exceed capabilities of single cluster, as topology requirements GRMS supports multi-cluster topology-aware parallel applications that can be executed on single cluster but also, if it doesn't stand in contradiction to application requirements, to be dispersed between resources belonging to many administrative domains to meet requirements exceeding resources available on single cluster. The cross-cluster scheduling allows not only to run tasks which requirements exceed capabilities of single cluster but also to use managed resources in more efficient way increasing the overall system throughput what is important without need to change the code of application. In order to create, synchronize and manage simultaneously co-allocation of computing resources located in different administrative domains an advance reservation mechanisms are used. Large scale parallel applications requirements together with complex parallel communication topologies can be easily expressed in a formal way using the XML-based job definition language called Job Profile. Consequently, applications developers and end-users are able not only to run their experiments in parallel onto many clusters, but also to perform various benchmark-based experiments as alternative topologies are taken into account during meta-scheduling processes in GRMS. Defined topologies may contain definitions of group of MPI or ProActive processes with resource requirements, using resource and network attributes, for the internal and external group-to-group communication. Therefore, various application-specific topologies such as master-slave, all-to-all or ring are supported in the Job Profile language.

Addressing requirements of complex scenarios consisting of many cooperating tasks GRMS is able to deal with jobs defined as a set of tasks with precedence relationships (workflows). The Workflow model used in GRMS is based on direct acyclic graphs (DAG). In this approach a user specifies task's precedence constraints in the form of task's states relationships. This what differs GRMS from other middlewares supporting workflows is that every tasks can depend on many others and the relationship can be expressed using nested AND/OR operators and any statuses of the parent tasks.

GRMS supports parameter sweep and allows to start in one call many instances of the same task with different set of arguments. For every task in the collection, the value of one or more of the task parameters may be changed in some preordained fashion creating a so called “parameters space”. Task of parameter sweep type can be of course part of a bigger workflow experiment and all parent-child dependencies are automatically converted by the system to take into consideration the whole collection of generated tasks. This is very useful feature and gives the user an easy way to search the parameters space for the concrete set of parameters that meet the defined criterion. What differs GRMS from other middleware dealing with parameter sweep tasks is the support for multi dimensional parameters spaces in which many parameters can be changed to build the aforementioned space of parameters.

The heart of GRMS is the Meta Scheduling framework which is responsible for scheduling tasks in the environment controlled by it. GRMS has been successfully integrated with the Scheduling Framework which have been designed, implemented and used in the Grid Scheduling SIMulator, what allows the GRMS administrator in a very easy and flexible way to change the scheduling policies using different scheduling plug-ins. The scheduling framework allows developers to easily implement specific algorithms hiding low-level technical details, so that it helps to focus on the scheduling problem and algorithm itself. Because GSSIM and GRMS share the same scheduling interfaces it is possible to test new scheduling plug-ins in a simulated environment before they will be used in production.

11.1.5 Programming and Execution Environments

In order to make the cross-cluster execution of different applications possible in firewalled and NAT (Network Address Translation) environments, the QosCosGrid project provides two parallel programming and execution implementations, namely: QCG-OpenMPI (aiming at C/C++ and FORTRAN applications developers) and QCG-ProActive (aiming at Java applications developers).

11.1.5.1 QCG-OpenMPI

The MPI (Message Passing Interface) is de facto a standard in the domain of parallel scientific applications which demands computational resources that are beyond what single machine can provide. It delivers end users both the programming interface consisting of simple communication primitives and the environment for spawning and monitoring MPI processes. A variety of implementations of the MPI standard is available (both as commercial and open source). In QosCosGrid, it was decided to use OpenMPI implementation of the MPI 2.0 standard as input for further enhancements. Of key importance were the inter-cluster communication techniques that deal with firewalls and Network Address Translation. In addition, the mechanism for spawning new processes in OpenMPI needed to be integrated with QosCosGrid-developed middleware. The extended version of the OpenMPI framework was named QCG-OMPI (where QCG stands for QosCosGrid).

The extensions were three-fold:

1. QCG-OMPI improves the MPI library by featuring multiple connectivity techniques to enable, when possible, direct connections between MPI ranks that are located in remote clusters potentially separated by firewalls.
2. The MPI standard was extended to comply with the QosCosGrid semi-opportunistic approach by providing a new interface to describe the actual topology provided by the meta-scheduler.
3. Many MPI collective operations were upgraded to be hierarchy-aware, and optimized for the Grid. We briefly present here each of these extensions, and a few results that demonstrate the performance achievable with QCG-OMPI.

11.1.5.2 QCG-ProActive

ProActive Parallel suite, which uses the standard Java RMI framework as a portable communication layer, was selected as a Java equivalent of OpenMPI library. With a reduced set of simple primitives, ProActive (version 3.9 as used in QosCosGrid) provides a comprehensive toolkit that simplifies the programming of java applications distributed on local area networks, clusters, Internet grids and peer-to-peer intranets. However, when QosCosGrid project was desiged, the standard ProActive framework did not provide any support for multi-user environments, advance reservation and cross-cluster co-allocation. To satisfy the requirements of complex system simulation applications and users, we developed extensions to the ProActive library (called QCG-ProActive) with the following goals:

1. To preserve standard ProActive library properties (i.e., allow legacy ProActive applications to be seamlessly ported to QosCosGrid).
2. To provide end users with a consistent GRMS Job Profile schema as a single document used to describe application parameters required for execution as well as resource requirements (in particular network topology and estimated execution time).
3. To prevent end users from the necessity to have direct (i.e., over SSH) access to remote clusters and machines.

11.1.5.3 OpenMPI and ProActive coordinators

In the QCG infrastructrure, spawning of parallel application processes on co-allocated resources is supported by additional Web Services, called coordinators. The necessity of using these services is caused by the fact that standard deployment methodologies used in OpenMPI and ProActive relay on either RSH/SSH or specific batch systems functionality, which both are limited for single cluster runs (for e.g. the SSH based deployment methods are problematic if at least one cluster has worker nodes that have private IP addresses).

11.1.6 Requirements

Taking into account different existing cluster configurations, in general, we distinguish:

1. A computing cluster with public IP addresses - both the front-end and the worker nodes have public IP addresses. Typically, a firewall is used to restrict the access to internal nodes.
2. A computing cluster with private IP addresses - only the front-end machine is accessible from the Internet, all the worker nodes have private IP addresses. Typically, NAT is used to provide out-bound connectivity.

Those two different cluster configuration types influence inter-cluster communication techniques supported in QosCosGrid, called port range and proxy respectively, and the requirements which must be satisfied to use each of the techniques.

11.1.7 Port Range Technique

The Port Range technique is a simple approach that makes the particular parallel environment firewall friendly. Most of the existing parallel environments use random ports by default to listen for incoming TCP/IP traffic. This makes cross-domain application execution almost impossible as most of system administrators often forbid to open all inbound ports to the Internet due to security reasons. By forcing the parallel environments to use only predefined, unprivileged range of ports, it is much easier for administrators to configure the firewall in a way to allow incoming MPI and ProActive traffic without exposing critical system services to the Internet.

The Port Range technique is visualized on the picture below. Each of the site administrators has to choose a range of ports to be used (e.g. 5000-5100) for the parallel communication and configure the firewall appropriately. Please note that the port range technique solves the problem of the cross-cluster connectivity for computing clusters where all worker nodes have public addresses.

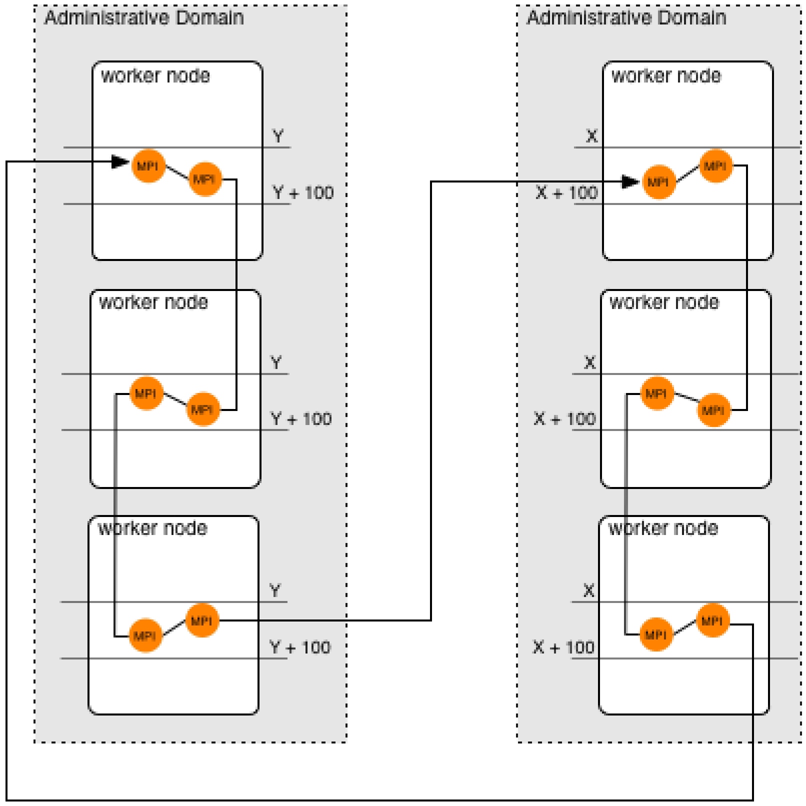


Figure 27: Operation of the Port Range technique

11.1.8 Proxy

In the second category of the clusters (cluster where worker nodes have IP private addresses) the Port Range technique is not sufficient as all the worker nodes are not addressable from the outside networks. Therefore, in addition to the Port Range technique (which helps to separate traffic) the SOCKS proxy service has to be deployed on front-end machines to route incoming traffic to the MPI/ProActive processes running at the local worker nodes.

The Proxy operation (used in conjunction with Port Range technique) is visualized in the picture below.

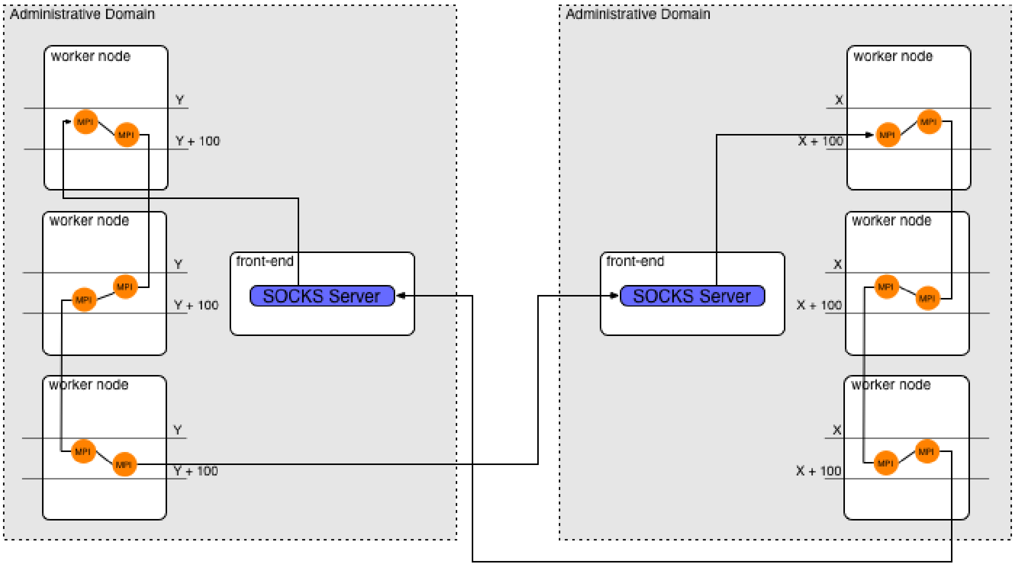


Figure 28: Operation of the Proxy technique.

11.1.9 E-Infrastructure requirements

- QCG-BES/AR service
  - must be started as root, but the main service process runs as unprivileged user (i.e. it adheres to the privilege separation model). The service must be deployed on any of the cluster submission host (not needed to be a frontend machine, e.g. in QosCosGrid we had deployment in TU Dortmund, where all the QCG components were located on one virtual machine - see: QosCosGrid instant install).
  - needs one open port (GSI/HTTPS enabled web service).
  - access to the relational database system (e.g. PostgresSQL).
  - Advance Reservation (AR) capable resource management system (AR is a base requirement for most of the co-allocation algorithms). Currently supported systems (in terms of AR): LSF, Grid Engine, Torque/Maui. Planned: PBS Pro, LoadLeveler.
  - Local Resource manager must be configured in such way that the QCG-BES/AR service is allowed to issue advance reservation requests.

- cross-cluster MUSCLE/OpenMPI
  - installation does not require administration privileges, the problem is to enable cross cluster communication. There are three options:
    - port range - works only for clusters with all worker nodes having public IPs - the administrator must open some range of unprivileged ports

- proxy - every worker node can be accessed via SOCKS proxy server, which is deployed on the machine having both public and private addresses. A secured version of SOCKS server can be used.
- external relay - e.g. TCP relay or XMPP server - no need to open any inbound ports, but this is seems to be less efficient technique.

11.2 Globus

Globus is deployed by many international grid projects, constituting the back- bone of both the US TeraGrid and UK National Grid Service. From Globus Toolkit version 3 (GT3), Globus followed a service oriented approach, tracking to a certain extent the development of the Open Grid Services Architecture (OGSA). The most widely deployed version (GT4) comprises a set of Web ser- vices and a Web services container, as well as a set of non-Web services related tools. The services are used to manage tasks such as job launching and data transfer. Re- cently, Globus has moved away from Web services architecture, with GT5 reverting to the client–server system previously used in GT2. The client tools supplied with all versions of Globus are command line based.

All versions of Globus feature a security infrastructure based on X.509 public key cryptography. Users must first convert their digital certificate to the PEM format, and place it, with the correct file permissions, into a specified directory in their home directory. Prior to issuing a Globus command, users must create a proxy version of their certificate, a short lived credential generated from the full grid certificate, which allows grid middleware tools to perform actions on their behalf. The client tools invoke a mutual authentication protocol to establish the user’s identity, via the proxy certificate, with the Globus job manager running on the remote resource.

11.3 UNICORE

UNICORE (UNiform Interface to COmputing REsources) is a grid middleware system that has been adopted for use by several international grid initiatives, for example the EU funded DEISA grid. It implements a three-tier architecture in the form of client-gateway-server. Jobs are passed around the grid as Abstract Job Objects (AJOs), which are serialised Java objects. The first tier of the architecture consists of a client with which the user prepares and submits jobs and also receives output back from the job. The middle tier consists of a gateway which controls authentication to the target resource. This tier also contains the Network Job Supervisor (NJS), UNICORE User Database (UUDB) and the Incarnation Database (IDB) . The NJS manages jobs and performs authorisation of a user on a target resource. The UUDB maps user certificates onto logins on the resource and the IDB translates AJOs to platform specific commands that the resource understands. The authenticated job is then submitted to the server tier (Target System Interface or TSI) which takes care of running the job on the target resource. With the exception of the TSI, the UNICORE system is implemented as a Java API and set of related programs. Version 6 of the UNICORE middleware presents a Web services interface based on the Open Grid Services Architecture framework.

Security is maintained in the UNICORE system though the use of X.509 certificates. Prior to using the system, the user must configure their client to use their digital certificate, imported into the Java keystore format. The user must also im- port the Certificate Authority root certificates for any of the resources that they want to access. When the client is launched, the user is prompted for their password to unlock the keystore. When submitting a job to the UNICORE system, the user's client uses the certificate to digitally sign the AJO before it is transmitted to the NJS. From here, the signature is verified using a copy of the public part of the user's certificate maintained at the NJS site, thus establishing the identity of the submitting user.

11.4 gLite

gLite is the product of the Enabling Grids for E-SciencE (EGEE) project, and uses components from a number of different sources to produce a middleware stack with a wide range of basic grid services. gLite is the mid- dleware underpinning the EGEE grid, designed to process the predicted 15 petabytes of data produced annually by the Large Hadron Collider instrument based at CERN.

11.5 SAGA

SAGA (the Simple API for Grid Applications) is a set of simple, coordinated and generic application programming interfaces (APIs) for accessing Grid services from generic application codes, portals, data managements systems. It is a standard developed by the Open Grid Forum (OGF) as a basis for a widely available and supported interface usable by application programmers enabling Grid functionality in their applications.

SAGA is designed in a modular plug-and-play manner, such that tools developed anywhere can be plugged into SAGA. SAGA lifts the burden of grid application programmers by providing them with a uniform interface to numerous types of grid middleware. As a result, grid application programmers need only learn a single API, to obtain access to the entire grid, which simplifies the application programming for the Grid. The grid middleware-dependent code (Globus, Condor, Unicore, etc.) still exists in SAGA, but it is hidden from the application programmer inside the SAGA libraries. Adjustable parameters are automatically optimized, based on available information on the current environment.

11.6 VINE

Vine Toolkit is a modular, extensible Java library that offers developers an easy-to-use, high-level Application Programmer Interface (API) for Grid-enabling applications. Vine can be deployed for use in desktop, Java Web Start, Java Servlet 2.3 and Java Portlet 1.0 environments with ease. Additionally, Vine Toolkit supports a wide array of middleware and third-party services, including QosCosGrid, OGSA BES, gLite, Globus and Unicore. Using the Vine Toolkit, one composes applications as collections of resources and services for utilizing those resources (basic idea in Vine - generic resource model - any service and data source can me modeled as an abstract entity called resource and can be integrated with web applications using high-level APIs).

The Vine Toolkit makes it possible to organize resources into a hierarchy of domains to represent one or more virtual organizations (VOs). Vine offers security mechanisms for authenticating end-users and authorizing their use of resources within a given domain. Other core features include an extensible model for executing tasks (every action is persisted as Task) and transparent support for persisting information about resources and tasks with in-memory or external relational databases.

11.7 SAGA vs VINE

Both SAGA and VINE set out to provide high-level abstraction levels for different distributed systems, and shield details of lower level middle-ware and system issues from the user.

SAGA is an API that provides the basic functionality required to build distributed applications, tools and frameworks, the main intention being that it is used to develop standalone applications. VINE is intended as a toolkit to build web applications (such as portlets, servlets or web services). The grid is only one of core components of VINE, meaning that it offers more possibilities and flexibility than the SAGA API. However, SAGA supports a wide range of programming languages, including C, C++, Python and Java, whereas VINE only supports Java 1.5 and newer.

11.8 MPI

Message Passing Interface (MPI) is an API specification that allows processes to communicate with one another by sending and receiving messages. It is typically used for parallel programs running on computer clusters and supercomputers.

The MPI interface provides virtual topology, synchronization, and communication functionality between a set of processes (that have been mapped to nodes/servers/computer instances) in a language-independent way, with language-specific bindings. MPI library functions include point-to-point rendezvous-type send/receive operations, choosing between a Cartesian or graph-like logical process topology, exchanging data between process pairs (send/receive operations), combining partial results of computations (gather and reduce operations), synchronizing nodes (barrier operation) as well as obtaining network-related information such as the number of processes in the computing session and current processor identity that a process is mapped to. It is the de facto standard model for parallel software development on most grid and HPC machines.

11.9 IBIS

The Ibis project (http://www.cs.vu.nl/ibis/) aims to drastically simplify the programming and deployment process of high-performance distributed applications. The Ibis philosophy is that applications should be developed on a local workstation and simply be launched from there. This write-and-go philosophy requires minimalistic assumptions about the execution environment, and sends most of the environment's software (e.g., libraries) along with the application. The Ibis system is designed to run in a hostile environment that is dynamic, heterogeneous, and suffers from connectivity problems.

11.10 Vertical Integration Scenarios

To illustrate the role of the various MAPPER software components, we will consider two application scenarios, paradigmatic of the multiscale applications considered throughout MAPPER. The first application has been classified as being loosely coupled, meaning that the submodels that comprise the application only communicate in one direction (i.e. the smaller scale passes parameters up to the bigger scale). The second application has been classified as being tightly coupled, meaning that its submodel components communicate in both directions, from the smallest scale to the larger scale and then back again.

11.10.1 Loose Coupling

The loosely coupled application is taken from the field of materials science described elsewhere in this document. The loosely coupled scheme consists of three levels of simulation. The lowest level simulates the electronic degrees of freedom, using the Car-Parrinello Molecular Dynamics (CPMD) code. This code is a parallelized plane wave / pseudopotential implementation of Density Functional Theory, particularly designed for ab-initio molecular dynamics. The high level of accuracy of this method provides a mechanism for deriving accurate atomic charges which can be used in classical molecular dynamics, where the electronic degrees of freedom are removed. The atomic charges are passed to the initial models simulated using LAMMPS classical molecular dynamics code. To increase the size and length of simulation we use the classical molecular dynamics simulation to create input model parameters for Coarse-Grained Molecular Dynamics (CGMD) simulations, again using the LAMMPS code. The CGMD simulations have reduced degrees of freedom, by combining atoms into single larger particles. The parameters which are transferred between these levels are the interparticle positions and interactions, calculated to reduce the structural details of the simulation.

The three components of the loosely coupled application scenario, are deployed on resources appropriate to their processor requirements. First, the CPMD application is executed, with it's output processed into a form that can be used as input to the LAMMPS MD code, and the data transferred to the resource running LAMMPS. Once the first LAMMPS model has completed, its output is similarly processed into a form that can be used by the second LAMMPS model, and the data transferred to a PRACE HPC resource. The sequential execution of the different scale models is managed by the GridSpace workflow engine, using the Application Hosting Environment as an interoperability layer. The QoS Broker is used to co-reserve resources, to ensure that the different submodels are able to execute before their subsequent submodels are are executed.

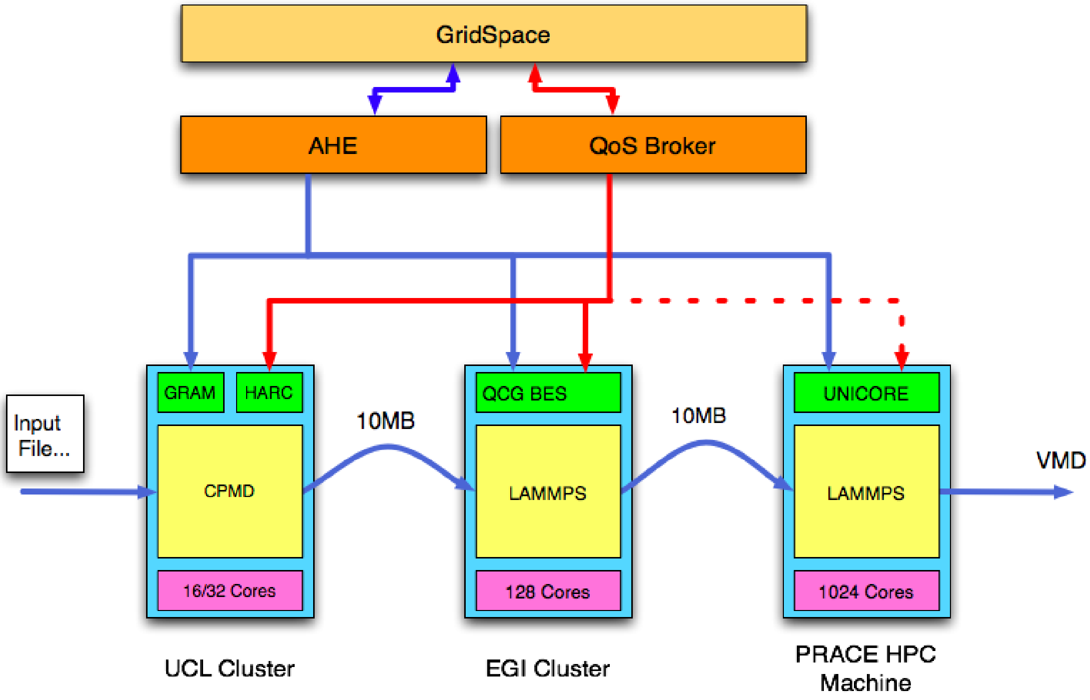


Figure 29: Architecture of the loosely coupled application scenario

11.10.2 Tight Coupling

The multiscale three-dimensional In-stent Restenosis model (ISR3D) allows 3D simulation of a stent deployment in a coronary artery and subsequent processes. The objective of the model is to help understand restenosis and to indicate improvements in stent design. The simplified ISR3D model considered in this scenario consists of three submodels: blood flow (BF), drug diffusion (DD) and smooth muscle cell proliferation (SMC). First, the model is initializes by placing a stent in an artery and then calculates where thrombus should be formed given the blood circulation. These initial conditions are sent to SMC and for each iteration of SMC, DD and BF are calculated in parallel. For performance reasons BF keeps track of its last state, simplifying subsequent flow calculations.

The submodels of ISR3D shown in the diagram below all have a single instance running, all throughout the simulation. However, they do their calculations at disjunct times, except BF and DD.

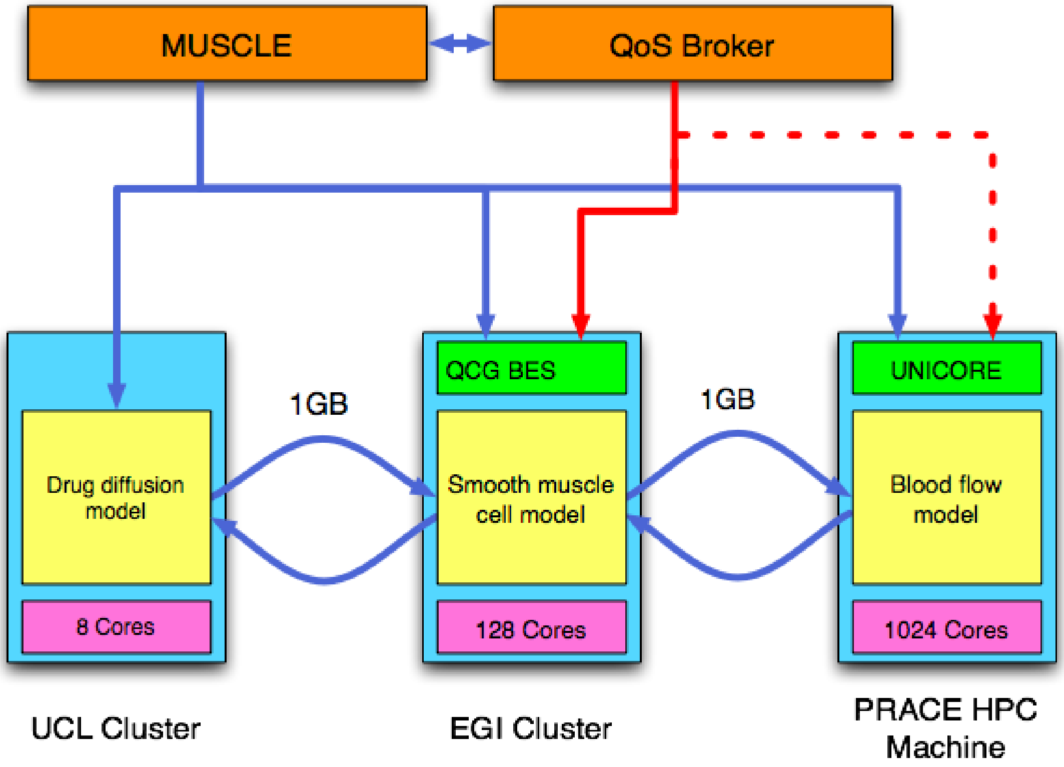


Figure 30: Architecture of the tightly coupled application scenario.

After the initialisation routine, the model enters a tightly coupled loop, alternating SMC with BF and DD, where the number of iterations of SMC determines the number of data transfers. The number of iterations of SMC is fixed, thus so is the number of synchronization points. Since the mainstay of the model is the tightly coupled part, we generally describe the ISR3D model as being tightly coupled. The coupling is done using the MUSCLE framework, with data transfers between the submodels in the gigabyte range. The QoS Broker is used to coallocate resources in order to ensure that they are available simultaniously as required.

12 Appendix D: Review of e-Infrastructures

12.1 PRACE

The Partnership for Advanced Computing in Europe, PRACE, is a persistent pan-European Research Infrastructure for Distributed High Performance Computing. PRACE provides European scientific community with the state of the art resources and systems for world-class science and strengthens Europe's scientific and industrial competitiveness. PRACE maintains a pan-European HPC service consisting of the top of the line systems integrated into the European HPC ecosystem. Each system provides computing power of several PetaFLOPS and offers modern tools and services supporting a wide range of scientific applications.

12.1.1 Resources

At the moment PRACE combines two HPC systems that are located in Jülich, Germany and Essonne, France. The first system, JUGENE, hosted by "Forschungszentrum Jülich" is an IBM BlueGene/P machine providing more than 800 TeraFLOPS of computing power. The second system, Curie, hosted by "Très grand centre de calcul du CEA" is being constructed by Bull and should enter production by the end of 2011 providing more than 1.5 PetaFLOPS of computing power. The number of PRACE systems is expected to increase in the next few years, starting 2012 when several additional systems will be integrated in the e-Infrastructure.

12.1.2 Services

PRACE provides users with state of the art services in the following areas:

- Network: PRACE operates a dedicated high speed 10Gbit network that connects all resources. The network offers users a high quality connection with fast data transfer rates. Multiscale simulations spanning multiple resources often depend on such network characteristics as constant low latency. Unfortunately, this can not always be guaranteed as the PRACE internal network is shared among all users and dedicated network paths can not be reserved by research groups working on the e-Infrastructure.

- Data: PRACE offers services for data management, including data transfer, storage and archiving. For data transfer PRACE relies on the GridFTP service that can fully utilize the PRACE internal network and deliver up to 1GByte transfer speed.
- Compute: PRACE implements a unified interface to compute resources offering functionality for job submission and management. Unfortunately services MAPPER relies on such as advance reservation and co-allocation are not offered by the e-Infrastructure.
- AAA (authentication, authorization, accounting): PRACE relies on PKI (Public Key Infrastructure) and X.509 certificates for user authentication. Every user has to poses a unique certificate before access to the e-Infrastructure can be granted. Each user group/project is assigned a fixed budget for computing resources. PRACE offers user services for accessing detailed information about the allocated and used budgets.
- Monitoring: All PRACE services are monitored to ensure their availability and correct functionality. Information describing the real time status of PRACE services is available to all users.
- User services: PRACE implements a number of technologies, for instance Modules, to improve user experience and provide a unified user environment on all resources. PRACE offers extensive user support services, including documentation, help desk, application support, tutorials and training.

12.2 DEISA

DEISA, the Distributed European Infrastructure for Supercomputing Applications, is a consortium of leading national Supercomputing centres that aims at fostering the pan-European world-leading computational science research. DEISA deploys and operates a persistent, production quality, distributed supercomputing environment with continental scope. It aims at delivering operational solutions for a future European HPC ecosystem. By extending the European collaborative environment in the area of supercomputing, DEISA is paving the way towards the deployment and operation of a persistent cooperative European HPC ecosystem, as suggested by ESFRI.

12.2.1 Resources

DEISA federates 18 national HPC systems of various architectures located across Europe (France, Finland, Germany, Italy, the Netherlands, Spain, Sweden, Switzerland, UK). The combined performance of all systems exceeded 1 PetaFLOPS. The machines are available to European scientists and, due to their unique characteristics, satisfy requirements of all scientific domains.

12.2.2 Services

Services provided by DEISA are to a large extent similar to those offered in PRACE. Although there are some differences, limitations of both e-Infrastructures from the MAPPER point of view are mostly identical.

- Network: The majority of DEISA resources are connected by a high speed 10Gbit network. In several cases, due to technical limitations, slower connections are offered.
- Data: DEISA offers services for data transfer and management, including a global GPFS-based file system, GridFTP protocol for file transfer, iRODS for data management. Additional services, for instance data archiving, are offered by DEISA partners on a case by case basis.
- Compute: DEISA supports a variety of interfaces to compute resources, including batch scheduling systems, UNICORE6 and Globus WS-GRAM. Advance reservation is not supported on compute resource but is supported on visualization systems.
- AAA (authentication, authorization, accounting): As well as PRACE, DEISA relies on the PKI for authentication. In exceptional cases, i.e. due to technical limitations, other authentication methods are supported.
- Monitoring: All DEISA services and e-Infrastructure components are monitored for availability and functionality.
- User services: Services offered by DEISA in this area are similar to PRACE.

12.3 EGI

Building a world-class pan-European High Performance Computing Service and infrastructure involves the scientific and industrial user communities with their leading edge applications. This needs to be done in a rapidly evolving context, were technologies change continuously and where the science focus changes as results are obtained and new directions are explored.

The ultimate goal of EGI is to provide European scientists and their international partners with a sustainable, reliable e-Infrastructure that can support their needs for large-scale data analysis. This is essential in order to solve the big questions facing science today, and in the decades to come.

12.3.1 Resources

EGI brings together more than 350 resource provider in Europe and more than 550 worldwide. The majority of EGI resources are cluster systems optimized for capacity computing (i.e. large number of serial jobs or parallel jobs requiring several CPUs, for instance data processing) and not supporting parallel interfaces such as MPI.

12.3.2 Services

EGI resources are operated by National Grid Initiatives (NGIs) that have to conform to the general regulations and policies defined by EGI. Each NGI, however, can specify additional internal policies. EGI users are granted access to the e-Infrastructure via virtual organizations (VO). Each VO can agree on internal policies and provide VO specific services.

- Network: EGI resources are connected via the public Internet network. Research groups and VOs can request dedicated channels from the local National Research and Education Networks (NRENs).
- Data: EGI resources have to provide several data services including GridFTP for data transfer.
- Compute: The majority of EGI resources support gLite. Other middlewares including Arc, Globus and UNICORE6 are supported as well, however are only available on selected resources.
- AAA (authentication, authorization, accounting): As well as DEISA and PRACE, EGI relies on PKI and X.509 certificates for user authentication. Unlike the other two e-Infrastructure, EGI supports so-called pool accounts, i.e. accounts and certificates that are shared by a group of people.
- Monitoring: All EGI supported services are monitored by the respective NGIs using Nagios.
- User services: EGI offers help desk, training and other support services to all recognised user communities.

12.4 Networking

A high quality network connection is essential for multiscale simulations spanning across multiple geographical locations. Poor network performance caused by, for example, high latency or random jitter, can compromise the ability of the individual simulation components to exchange data which may result in a failure of the simulation. As such it is important that network policies enforced in an e-Infrastructure offer users the possibility to satisfy their network requirements either by using available resources and services or by requesting support from external providers.

12.4.1 National Research and Education Networks (NRENs)

An NREN is responsible, on a national basis, for the provision of data communications networks and services to the research and education community of its country. The NREN network typically connects other networks at regional or metropolitan level. NRENs in Europe cooperate together and provide the access to European wide networking infrastructure operated by DANTE [11]. DANTE is owned by European NRENs, and works in partnership with them and in cooperation with the European Commission.

DANTE provides the data communications infrastructure essential to the development of the global research community. However, in order to meet new scientists requirements defined in MAPPER various new networking services will be required for networking provisioning. Thus, DANTE is coordinating joint research and development activities in the GEANT3 project [12]. In GEANT3 project key services that are under development are related to network provisioning.

The concept of dedicated capacity via point-to-point connections one step further, providing bandwidth-on-demand to users with exacting requirements who will be able to reserve the bandwidth they need, as and when they need it. The AutoBAHN service, which partially developed at PSNC, is able to calculate a route across the multi-domain environment for the required capacity at the required time, and balance the demanding users'requirements with those of traditional users so that each receives the agreed level of service.

12.4.2 PRACE

PRACE resources are interconnected with a dedicated high-speed 10Gb network that is shared by all users. Although offering a very good performance in theory, it can not always be achieved in practice, such as in situations where many users are concurrently transferring large amount of data. At the moment PRACE operates only two machines, JUGENE located in Forschungszentrum Jülich and CURIE operated by the French Alternative Energies and Atomic Energy Commission. As such, utilization of the PRACE internal network is not high at present. This will however change later this year when additional systems will be brought in production.

Connection to the public network is available on all PRACE resources and is managed by the resource provider. PRACE does not define policies for this type of connection and therefore their availability might vary from one resource to another.

12.4.3 DEISA

The majority of DEISA resources are interconnected with a dedicated high-speed 10Gb network. In a few cases where a high-speed interface cannot be installed due to technical limitations a 1Gb link is provided. Availability and performance of all network interfaces and segments is monitored by DEISA and the respective network operators to ensure high quality of service provided to the users. The DEISA internal network is shared and can be concurrently used by all DEISA users. Therefore all limitations described in the previous section apply to DEISA.

Due to the heterogeneous nature of the DEISA e-Infrastructure the network performance optimization capabilities are technically limited. As a consequence, in some cases research groups might not be able to utilize the network to its full potential. To circumvent this a research group may request a dedicated connection between specific DEISA resources. Such network links can be provided by the National Research and Education Networks operating in the regions respective to the resource location. Depending on the agreement between DEISA and the NRENs involved the research group issuing the request might be asked to bear a part of the outstanding costs.

12.4.4 EGI

NGIs and resource providers are responsible for establishing the necessary network connectivity using the public Internet. Each NGI and VO may establish internal regulations and policies that, however, will not be uniform across the EGI e-Infrastructure. Research groups that require special networking functionality, such as links with low latency for computational steering workflows, may request dedicated links from the respective NREN or the DANTE project. However, as in the example describe in the previous section, the requester might be asked to bear a part of the costs associated with setup and maintenance of the network channel.