



**System for Environmental and Agricultural Modelling;  
Linking European Science and Society**

**Final version of SeamFrame design**

Rizzoli, A.E., Athanasiadis, I.N., Li, H., Knapen, R., Wien, J.J.F., Villa,  
F., Huber, D., Senaldi, F., Svensson, M.

Partners involved: IDSIA, ALTEERRA, UVM, AntOptima, LU



Report no.: 34  
May 2008  
Ref: D5.3.9  
ISBN no.: 90-8585-122-X  
and 978-90-8585-122-6



Logo's main partners involved in this publication

Sixth Framework Programme

SEAMLESS integrated project aims at developing an integrated framework that allows ex-ante assessment of agricultural and environmental policies and technological innovations. The framework will have multi-scale capabilities ranging from field and farm to the EU25 and globe; it will be generic, modular and open and using state-of-the art software. The project is carried out by a consortium of 30 partners, led by Wageningen University (NL).

Email: [seamless.office@wur.nl](mailto:seamless.office@wur.nl)

Internet: [www.seamless-ip.org](http://www.seamless-ip.org)

Authors of this report and contact details:

Name: Andrea E. Rizzoli	Partner acronym: IDSIA
Address: IDSIA, Galleria 2 Manno, Switzerland	
E-mail: <a href="mailto:andrea@idsia.ch">andrea@idsia.ch</a>	
Name: Ioannis N. Athanasiadis	Partner acronym: IDSIA
Address: IDSIA, Galleria 2 Manno, Switzerland	
E-mail: <a href="mailto:ioannis@idsia.ch">ioannis@idsia.ch</a>	
Name: Hongtao Li	Partner acronym: IDSIA
Address: IDSIA, Galleria 2 Manno, Switzerland	
E-mail: <a href="mailto:hongtao@idsia.ch">hongtao@idsia.ch</a>	
Name: Rob Knapen	Partner acronym: ALTERRA
Address: Postbus 47, 6700 AA Wageningen, The Netherlands	
E-mail: <a href="mailto:Rob.Knapen@wur.nl">Rob.Knapen@wur.nl</a>	
Name: Jan-Erik Wien	Partner acronym: ALTERRA
Address: Postbus 47, 6700 AA Wageningen, The Netherlands	
E-mail: <a href="mailto:Jan-Erik.Wien@wur.nl">Jan-Erik.Wien@wur.nl</a>	
Name: Ferdinando Villa	Partner acronym: UVM
Address: University of Vermont, 590 Main Street, Burlington, Vermont, USA	
E-mail: <a href="mailto:ferdinando.villa@uvm.edu">ferdinando.villa@uvm.edu</a>	
Name: David Huber	Partner acronym: ANTOPTIMA
Address: via Fusoni 4, 6900 Lugano, Switzerland	
E-mail: <a href="mailto:david.huber@antoptima.ch">david.huber@antoptima.ch</a>	
Name: Federico Senaldi	Partner acronym: ANTOPTIMA
Address: via Fusoni 4, 6900 Lugano, Switzerland	
E-mail: <a href="mailto:federico.senaldi@antoptima.ch">federico.senaldi@antoptima.ch</a>	
Name: Mats Svensson	Partner acronym: LUND
Address: Lund University, PO Box 170, S-22100, Sweden	
E-mail: <a href="mailto:mats.svensson@lucsus.lu.se">mats.svensson@lucsus.lu.se</a>	

**Disclaimer 1:**

“This publication has been funded under the SEAMLESS integrated project, EU 6th Framework Programme for Research, Technological Development and Demonstration, Priority 1.1.6.3. Global Change and Ecosystems (European Commission, DG Research, contract no. 010036-2). Its content does not represent the official position of the European Commission and is entirely under the responsibility of the authors.”

"The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability."

**Disclaimer 2:**

Within the SEAMLESS project many reports are published. Some of these reports are intended for public use, others are confidential and intended for use within the SEAMLESS consortium only. As a consequence references in the public reports may refer to internal project deliverables that cannot be made public outside the consortium.

**When citing this SEAMLESS report**, please do so as:

Rizzoli, A.E, Athanasiadis, I.N., Li, H., Knapen, R., Wien, J.E., et al., 2007. Final version of SeamFrame design, SEAMLESS Report No.34, SEAMLESS integrated project, EU 6th Framework Programme, contract no. 010036-2), [www.SEAMLESS-IP.org](http://www.SEAMLESS-IP.org), 32 pp. ISBN no. 90-8585-122\_X and 978-90-8585-122-6.



# Table of contents

<b>Table of contents</b>	<b>5</b>
<b>General Information</b>	<b>7</b>
<b>Executive summary</b>	<b>7</b>
<b>1 The SeamFrame system model</b>	<b>9</b>
1.1 Introduction	9
1.2 Target user groups	11
<b>2 Technologies used in SeamFrame Version 0.2</b>	<b>13</b>
2.1 Overview	13
2.2 SeamFrame Application Server Technologies	13
2.3 SeamFrame Client Technologies	14
2.4 Knowledge Technologies	15
<b>3 The integrated modelling environment (pre-modelling phase)</b>	<b>17</b>
3.1 SEAMLESS project definition	17
3.2 Narrative experiments	18
3.3 Indicator Selector and Indicator FactSheet Editor	18
<b>4 The processing (execution) environment (modelling phase)</b>	<b>21</b>
4.1 Implied model chain choice through scales selection	21
4.2 Application linking through OpenMI+	21
4.3 The pull-push execution	22
4.4 Detailed specification of Experiments	22
4.5 Linking the client application with the server	23
4.5.1 Chain execution, server-side queuing model execution and client-side monitoring	23
4.5.2 Data persistence and linking with data	23
<b>5 The visualization environment (post-modelling phase)</b>	<b>25</b>
5.1 Seam:PRES Workflow	25
5.1.1 Select Visualization	26
5.1.2 Visualization Viewing	26
5.1.3 Visualization Editing	27
5.2 Future Versions of Seam:PRES	28
5.3 Design of Seam:PRES	28
5.3.1 Data Services	28
5.3.2 Asynchronous Processing	29

---

5.3.3	Visualization Factory	30
	<b>References</b>	<b>31</b>

---

## General Information

Task(s) and Activity code(s):	5.3.8
Input from (Task and Activity codes):	5.3.1 - 7
Output to (Task and Activity codes):	5.4 and 5.5

## Executive summary

SeamFrame is the software tool for supporting the SEAMLESS Integrated Framework that aims to facilitate integrated assessment of agricultural and environmental policies, through the integration of models and data, the execution of model chains of ex-ante analyses, the communication with policy experts, and the visualization of results and indicators. This report documents the final design of SeamFrame, presents its current status (as of Version 0.2) and specifies directions for future development (Versions 0.3 and Final).

An overview of the SeamFrame system is presented and the main user types are specified. The technologies to be used in SeamFrame are detailed and the selected client-server architecture is specified. SeamFrame is presented from an integrative modeller perspective: First comes the pre-modelling phase that involves the interaction with policy experts. The system functionalities for facilitating both the integrative modeller and the policy expert to define together a SEAMLESS project, narrative experiments and associate indicators for them is presented. Then comes the modelling phase, that from a SeamFrame point of view involves the selection of model chains through scales, model linking using OpenMI+, the execution of model chains following a push approach, and the detailed specification of experiments. The SeamFrame server provides most of these functionalities, while the SeamFrame client provides services for remote invocation of model chain execution and retrieving of model results. Particular attention is given to the use of Enterprise Java Beans for data persistence and the use of ontologies for domain conceptualization. The last step involves the post-modelling activities, which are supported through the visualization of indicators and model results. An analysis of the visualization workflow is provided along with details on fetching and rendering data on the client.

This report is accompanied by an installation of Version 2 of SeamFrame (publicly available at <http://prototype2.seamless-ip.org/gui/>), the SEAMLESS-IF User Manual (PD5.6.3.2), the source codes of SeamFrame (available at <http://svn.seamless-ip.org/>), and the documentation of the sources (i.e. PD5.4.2.2).





# 1 The SeamFrame system model

## 1.1 Introduction

During first two years of the SEAMLESS project, many meetings were held, during which requirements were collected and screened, and modelling frameworks were reviewed. The results of this stage of the project are documented in D5.2.1 (Svensson, Rizzoli, *et al*, 2005), D5.2.4 (Svensson, Johnson, *et al*, 2006), and SEAMLESS Report 6 (Rizzoli, Svensson, *et al*, 2005). From the requirements, a first sketch of the SeamFrame architecture emerged (Rizzoli, Athanasiadis, *et al*, 2005), which then has been further improved (van der Wal, Rizzoli, *et al*, 2006) and finally developed, tested and delivered as Prototype 1 (Huber, Knapen, *et al*, 2005).

The SeamFrame system model aimed at supporting a component-oriented approach to software development (Szyperski, 2002). The presentation of the system model is instrumental to the re-organisation of the requirements around the main architectural elements (see Figure 1): the Knowledge Base, the SeamFrame core framelets (Domain Manager, Model Manager, Tool Maker) the SeamFrame Development Environment (Modelling Environment, Processing Environment), and a number of SEAMLESS applications (SeamApps: APES, FSSIM, SEAMCAP).

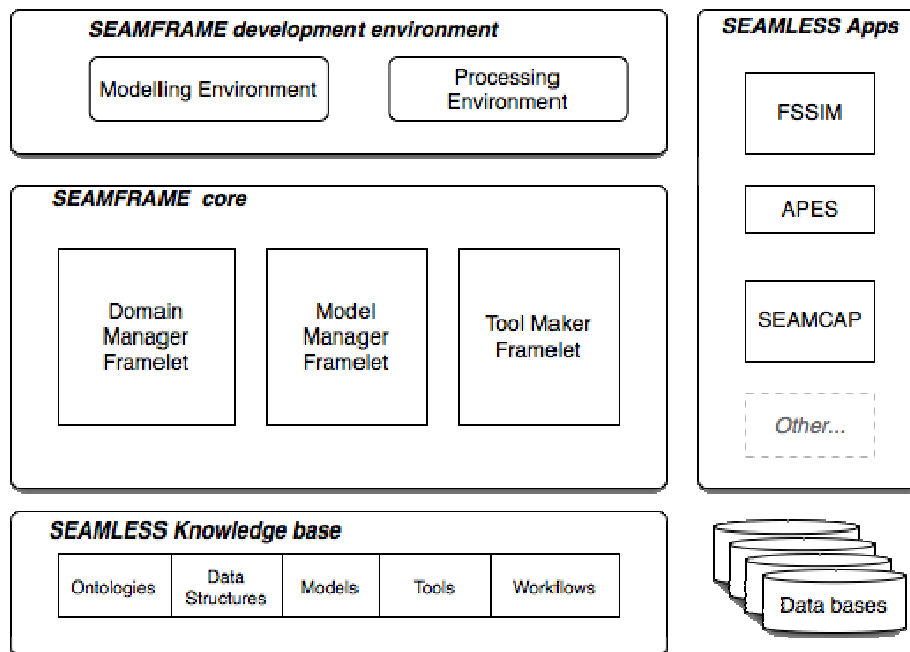


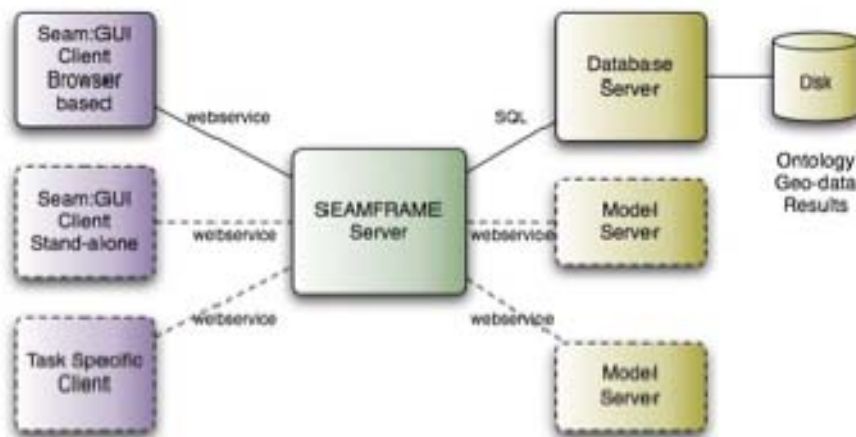
Figure 1: The SeamFrame architecture.

At the Work Package 5 meeting in Bologna (November 2006) it was agreed that based on the experiences and feedback from Prototype 1 (Huber *et al.*, 2005), the next step will not be a “throw-away” prototype that demonstrates the proof of concept of the envisioned final product; instead we are focused towards a functional version of the final SEAMLESS-IF product. Therefore, it was decided that Prototype 2 would eventually be a Version 0.2 of the final product, which signifies **the move to a web based-distributed system** had to be made.

A typical three tier web-based implementation is envisioned for Version 0.2, as shown in Figure 2. A clear separation is introduced between SeamFrame running on a server and a

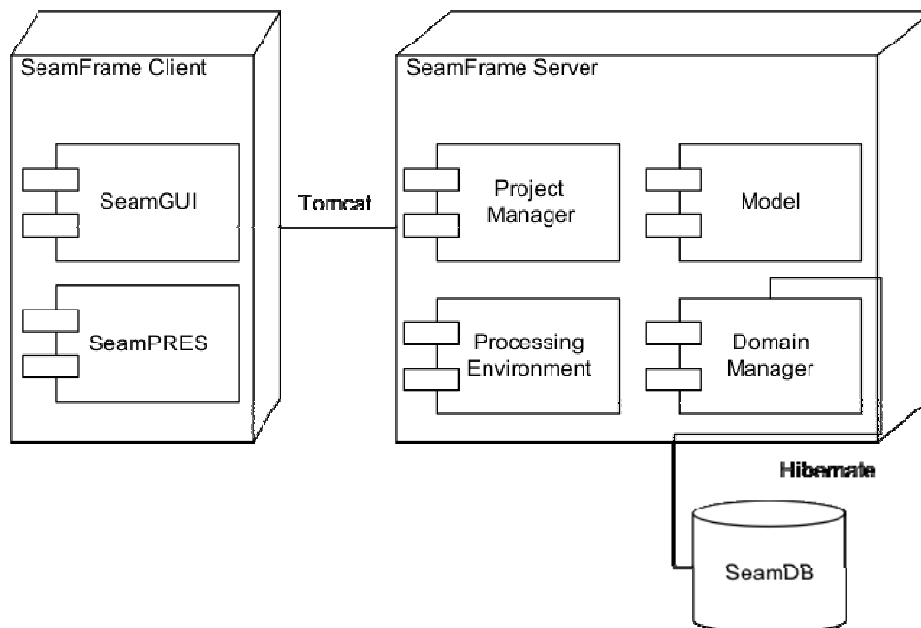
number of client applications that run on different computers. It should be noted however that it remains possible to run both clients and SeamFrame on a single system by using a local server.

The operation and interfaces of the final version of SeamFrame will be well documented, and open standards will be used for its implementation, similarly to the approach we followed for Prototype 1 (i.e. D5.3.2, D5.4.1, D5.4.2, D5.4.3.1), so that new applications can be easily added in the future. An analytical presentation of the software quality assurance plan is detailed in D5.6.2.1 (Svensson, Wallman, *et al*, 2007). Additionally, we envision SeamFrame functionality accessed or re-used in other projects.



**Figure 2: The proposed web-based architecture of SeamFrame**  
*(Dotted lines indicate future extensions to the system, envisioned after the end of the project)*

The conceptual drawing for Version 0.2 (where models run on the SeamFrame server) can be translated into the following UML deployment diagram (Figure 3).



**Figure 3: The UML deployment diagram for SeamFrame**

The SeamFrame client consists of:

- Seam:GUI, the module that guides the end-user to manage projects and request the execution of model chains;
- Seam:PRES, a toolkit for presentation results of computed/simulated projects;

The above client application functionalities are enabled by the components provided by the SeamFrame Server, which include:

- The Project Manager, which enables the appropriate services for initiating, configuring, and managing projects;
- The Processing Environment, the facility that receives user requests for model-chain executions and manages them. It enables model composition and execution;
- The Domain Manager, that provides access to Seam:DB in SeamFrame, through Hibernate object-relational mappings;
- The individual Models included in the SEAMLESS Model chains, namely: SeamCAP, FSSIM-MP, APES and FSSIM-AM. Each model is wrapped for the SEAMLESS needs to the interfaces imposed by the Processing Environment.

## 1.2 Target user groups

During the course of work (and specifically during TW2) it emerged that a revision of the target user groups was needed: it was decided to simplify the present classification of user types (i.e. coders, linkers, providers, runners, players, viewers) and to distinguish between three user types:

- Policy experts;
- Integrative modellers;
- Agricultural modellers.

It was clarified that emphasis for this version 0.2 is on the integrative modeller. This user group comprises agricultural and modelling experts from various agricultural research institutions who intend to perform integrated assessment studies and mainly target at ***using the models together – not in isolation***. The development of Version 0.2 does not ignore the demands of the first user group, i.e. policy experts, such as DGs and related consultants. Importantly, agricultural modellers are seen as a group that will utilise the framework to support model use, establishing linkages between models, but are not intended to use the framework as full-fledged environment for model development. However, flexible and transparent access to inputs and outputs are seen as key ingredient to support agricultural modellers.



## 2 Technologies used in SeamFrame Version 0.2

### 2.1 Overview

Prototype 1 (Huber, Knapen, *et al*, 2005) served well as a demonstrator of possibilities for the SEAMLESS-IF software. Starting from a very rough and not very detailed set of requirements, it was necessarily broad in functionality yet shallow in its implementation. Prototype 1 was meant to serve as a basis of building a consensus on system functionalities and it was deployed as a stand-alone, desktop application. Prototype 1 has been extensively discussed in the transition workshops in March and September 2006, which has led to a better understanding of the requirements for the software and to a more limited functionality scope. This was reflected in the Work Package 6 proposals for improvements of Prototype 2 (Therond, Taverne, *et al*, 2007).

It is now clear that the focus must be on the policy experts and the integrated modellers, and not so much on the agricultural modellers. Another point that has been clarified is that SeamFrame and its tools must be seen only as a part of SEAMLESS-IF. Additional software and methods will be used for interaction with users and to conduct the pre-modelling and post-modelling phases. Most likely, software tools are already in use, including standard office software like spreadsheets and word processing applications, but also tools and methods for modelling, i.e. for calibration purposes. Within the SEAMLESS-IF framework, agricultural modellers will evaluate biophysical or economic models independently from SeamFrame, using established tools and methods. SeamFrame purpose is to provide agricultural modellers with *a framework for integrated modelling activities*.

Also abandoned is the idea that a single user will have a desktop version of SEAMLESS-IF, in favour of a more participatory and interdisciplinary vision, where a group of users is interacting in pre- and post- modelling phases with a group of scientists with expertise in SEAMLESS-IF paradigms and tools.

These requirements shifts and clarifications are best met by selecting a multi-tier architecture for SeamFrame (Figure 2). This will allow (controlled) access to data and model simulation *on a centralized server* from several client computers. The software on the client computer can be tailored to specific tasks and / or user roles. On the server, quality of the models and data can be controlled and actions taken for back-ups, etc. It is still possible to deploy such architecture on a single computer, however this requires installing locally a database system and a web server, having one machine playing both roles of server and client.

The client software on the other hand will require no installation, apart from a web browser with a Flash player<sup>1</sup>, that most computers do have preinstalled, and it will simply run inside a web browser.

### 2.2 SeamFrame Application Server Technologies

The Processing Environment of SeamFrame is incarnated into a SeamFrame Application Server. In other words, the SeamFrame Application Server implements the SeamFrame Processing Environment.

---

<sup>1</sup> Flash Player is an independent application or an add-on for a web browser that enables the rendering of Flash multimedia content. In turn, Flash is a multimedia authoring program, that uses vector and raster graphics, a native scripting language (called ActionScript), and bidirectional streaming of video and audio. For more see: <http://www.adobe.com/products/flashplayer/>

Concerning the server tier of SeamFrame it is a Java<sup>2</sup> based solution. Java Enterprise Edition (JEE) is a very mature and frequently used technology for developing this type of server software. SeamFrame uses servlets for server-client communication and Enterprise Java Beans for data persistency. A web application server (the market dominant Tomcat<sup>3</sup>) and a relational database management system (PostgreSQL<sup>4</sup>) have been deployed on the SEAMLESS server in Lund. All communication with the SeamFrame application server uses standard protocols and technology (e.g. http requests, servlets, Enterprise Java Beans).

Although the SeamFrame application server operates in Java environment, the model components can be implemented using other languages, if so required, and as long as they can be integrated. This requirement has been translated into the fact that model components should be OpenMI<sup>5</sup> compliant, by implementing the OpenMI interfaces and allow linking to other components (for data exchange). The model component can take care of this by itself or a wrapper or bridge can be programmed. OpenMI needed to be adapted and extended to work in this new environment. Particular for future versions (beyond the initial 0.2 version) when models could be running on different systems and OpenMI should support web-based, parallel, execution of model chains. Already in Version 0.2 modifications were needed to include push-based data exchange into OpenMI, since this is required for the agro-management and biophysical components. All modifications to (the Java version of) OpenMI will be presented to the OpenMI Consortium to be considered for a future release of the OpenMI software.

## 2.3 SeamFrame Client Technologies

The SeamFrame Server can be accessed by SeamFrame Clients that “consume” the functionalities it offers. Basically we can consider two types of clients: system-to-system and human-to-system.

A **human-to-system client** is a graphical user interface that allows using SeamFrame server functionalities, e.g. to define a project or to visualize results. At least a few of this kind of clients will be developed as part of SEAMLESS-IF. Typically, they are known as Rich Internet Applications (RIA), software applications running on the user’s computer that rely on a server for core functionality but also have some own logic for an enhanced user experience.

A **system-to-system client** is a programming interface that exposes the SeamFrame server functionality. In this case, there is no user interaction, rather there are just computers exchanging information (actually the SeamFrame server provides services that the second computer is consuming). The chosen SeamFrame architecture, using using open standards whenever possible, is capable of supporting system-to-system clients. However, this is not a priority within this course of work, and we do not envision deploying such clients during the duration of the project.

For building Seam:GUI and Seam:PRES (see Figure 2), the two flagship client-applications of SEAMLESS-IF, Adobe Flex 2.0<sup>6</sup> has been selected as the most usable technology. It offers

---

<sup>2</sup>...<http://java.sun.com/>

<sup>3</sup> <http://tomcat.apache.org/>

<sup>4</sup> <http://www.postgresql.org/>

<sup>5</sup> <http://www.openmi.org/>

<sup>6</sup> <http://www.adobe.com/products/flex/>

declarative user interface programming with a rich library of professional and functional components. Code is compiled and runs in a Flash 9 virtual machine (inside a web browser). Protocols for data exchange (pull, server push and data binding) with a server are included. The basic Flex SDK is free, and also limited use of Flex Data Services (required to push data from the server to multiple clients). To facilitate development, a Flex Builder 2 license can be bought, which can include a set of useful charting components. Since a Flex application basically is deployed as a Flash application running within a web browser it is easy to create GUI mock-ups and distribute them to get feedback. This process helped a lot in clarifying the end-user requirements. Specifically, during the past period Work Package 5 followed an iterative process for communicating the status and functionality of Seam:GUI and Seam:Pres, by publishing non-functional Flash application on an internal server. End-users were accessing through a single point the client application and provide with their feedback on design and usability issues.

Finally, it worths mentioning that web based clients use extensively the Cascading Style Sheets (CSS) standard whenever possible to allow future synchronization (or changes) of look and feel.

## 2.4 Knowledge Technologies

In SEAMLESS, we employ knowledge-intensive technologies, such as semantic modelling and ontologies for specifying data, models, projects and their relationships. The Domain Manager plug-in of the Knowledge Manager (Villa, 2005), (Athanasiadis, Huber, *et al*, 2006), (Villa, Athanasiadis, Rizzoli, 2007) provides with the tools for persistently storing generated classes using the EJB/Hibernate technology: through Hibernate a relational database may store Java objects persistently. The database design is directly mandated by the object structure and the data can be stored in the DB based on an adjunct persistence XML document provided along with a class. This approach replaces the OWL file based persistency used in Prototype 1. The reason for this shift is for converging to programming practices that enable client-server application development. Other SEAMLESS working groups (WP4 and WP3) have been populating the database following the schema generated by the object structure, adopted by the modellers. This is a very important shift in everyday practice that SEAMLESS has achieved: Modellers specify the data requirements of their models, which are transformed into a database model, to which “data collecting” activities need to comply with. Data may originate from third-party, pre-existing databases or may have been directly collected (i.e through surveys). Either of the two is the case, data collection activities need to facilitate the filling-up of the generated database, which is fit to the model data requirements.

The **SEAMLESS Knowledge Manager**, is the Thinklab platform, a facility, jointly developed by UVM and IDSIA, as an open source project<sup>7</sup> for accessing and managing ontologies. It is built upon the Protege-OWL libraries, and has been designed using the plugin approach (Villa, 2005). A web-based extension of the Knowledge Manager (called Thinkcap) is currently under development independently from Seamless, and will be able to visualize semantic models on the web. A Thinkcap customization for browsing the SEAMLESS ontologies will become available with the delivery of Version 0.3, as decided during TW3.

The **SEAMLESS Domain Manager** is a tool for delivering java objects that are used by models and tools for exchanging information. The Domain Manager uses the Knowledge Manager for accessing ontologies and following the Domain Class principle, it may:

---

<sup>7</sup> <http://www.integratedmodelling.org/thinklab>. Terms of use are defined by the General Public Licence (<http://www.gnu.org/gpl>).

- 1 Generate domain classes from ontologies
- 2 Access domain objects at runtime (by using EJB/Hibernate)

All the commonly shared data types in SEAMLESS are declared in the domain ontologies (starting from projects, up to the model exchange items) and through the domain manager, they become available through a shared interface. In this sense, the Domain Manager is the "knowledge processing" component of SeamFrame, providing access to and modification of SEAMLESS data, through ontology-specified interfaces. The Domain Manager accesses the set of SEAMLESS ontologies through the Knowledge Manager.

Fetching data from the Knowledge Base actually means to retrieve data from a relational database (i.e. SeamDB) using information provided by Hibernate<sup>8</sup>. The Knowledge Manager only deals with metadata (i.e. information on data structure, accessing details as jdbc connections, etc). Data are delivered by Seam:DB and are accessible by using the generated JavaBeans through hibernate.

---

<sup>8</sup> <http://www.hibernate.org/>



### 3 The integrated modelling environment (pre-modelling phase)

#### 3.1 SEAMLESS project definition

Through the Seam:GUI the user may build up a project that specifies an integrated assessment exercise. A *project* is characterized by the definition of the problem it tries to solve or study, and it incorporates at least one experiment configuration, that is, the configuration of the models to be executed during the analysis. An *experiment*, in turn, is associated with a single model chain and is parameterized by the specification of an outlook, a context and a policy option. Also the *indicators* are associated with a problem; they are used for quantifying the analysis results. Through a single project, there are several alternatives that can be investigated. An experiment, when configured, corresponds to a single run of one of the model chains, which has two parts: the biophysical chain and the policy assessment chain. Based on the results of the computation, the calculated indicators become available in the framework and can be used for visualizing results in Seam:PRES, the visualization client.

The project definition is a result of the SEAMLESS DOT.Force<sup>9</sup> team (Janssen, *et al*, 2007, Janssen, Li, *et al*, 2007) and has been specified as an ontology in OWL. Thereafter, the Persistence Plugin of Thinklab has been used for generating object-oriented source code as Java Beans and the database schema in relational SQL<sup>10</sup>.



Figure 4: Partial view of the Project Ontology

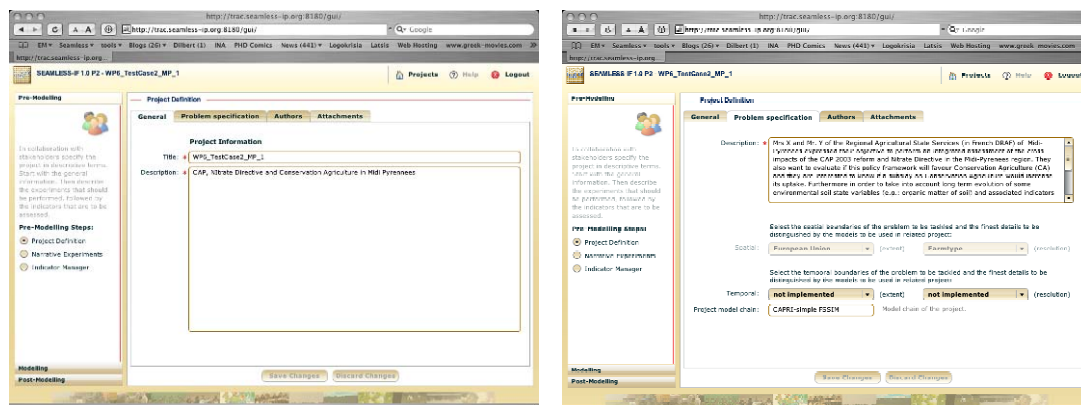
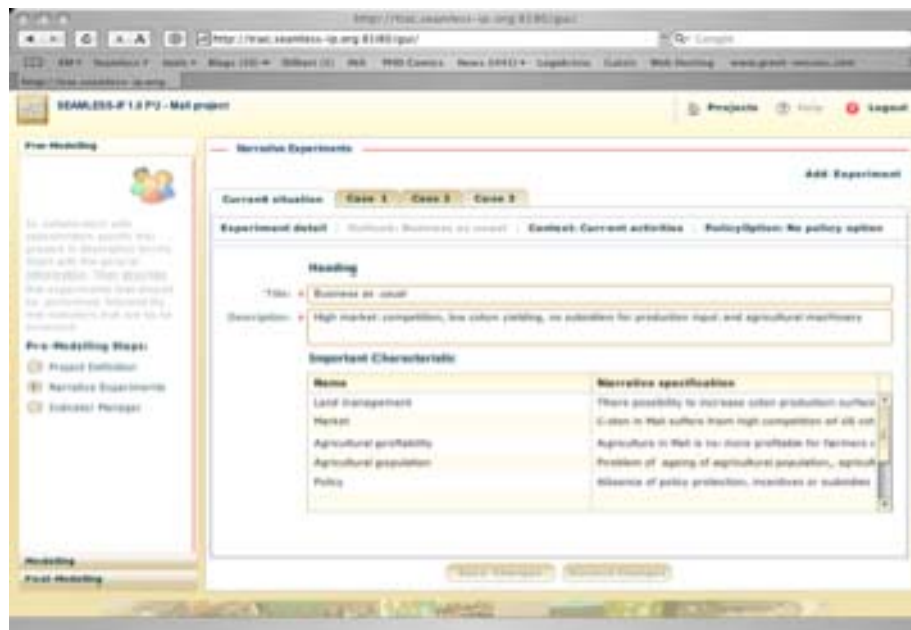


Figure 5: Screenshots of the project definition panels in Seam:GUI

<sup>9</sup> Seamless Data-Ontology Task Force

<sup>10</sup> Structured Query Language

Through the Seam:GUI client application, the end user may configure a SEAMLESS-IF



**Figure 6: The narrative experiment configuration**

Project, by specifying the narrative descriptions of problem and experiments, context, outlook and policy options, but also, by selecting indicators applicable for the exercise at hand. Figure 5 below presents two screenshots of the project definition panel.

### 3.2 Narrative experiments

Another important aspect in the pre-modelling phase is the narrative specification of the experiments (Figure 6). The narrative specification is required, so that the integrative modeller will communicate its requirements to the agricultural modeller, who will materialize them in turn, as project parameters through the Experiment configuration in the modelling phase. There is a one-to-one relationship between the narrative and the configuration: Each context, outlook and policy option has a narrative description, that will be accessible to the integrative modeller while detailing the configuration.

### 3.3 Indicator Selector and Indicator FactSheet Editor

The pre-modelling phase concludes with a choice of indicators for the project. The indicator manager allows the selection of indicators and the user may specify thresholds and targets for each one of them. The following Figure 7 illustrates the panels for managing the indicators.

In Version 0.3 (Prototype 3), more features will be added, as the policy expert user requirements will be come more clear. For example, functionalities for indicator filtering or querying could be additionally become available. There are also options for further developing and integrating the indicator calculator, based on the policy-expert user needs.

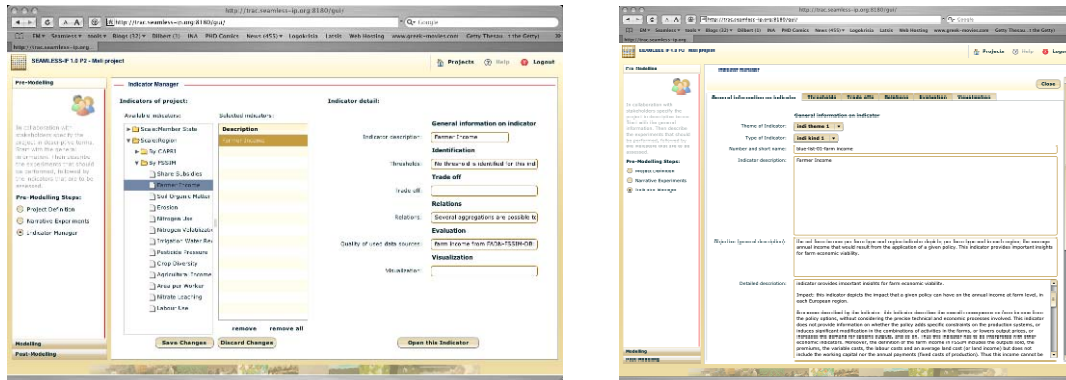


Figure 7: Indicator Manager screenshots



## 4 The processing (execution) environment (modelling phase)

### 4.1 Implied model chain choice through scales selection

The scales that are relevant when performing an impact assessment are the scales of the problem definition, including policy options, contexts, outlooks and indicators. The scales (spatial and temporal extents/resolution) of the problem are defined by the integrative modeller jointly with the policy expert. The scale of a model is defined by the modeller, taking into account the scales of the available data. The interpretation of the problem scale into the model scale is made by the modeller when the model is integrated in the SEAMLESS-IF. In this way, the project definition was decoupled from the inputs and parameters required by the models.

The selection of the extent and the resolution of the problem scale in SEAMLESS (in the pre-modelling phase) implies the scale of the models and also the model chain to be used. Specifically, the following combinations (Table 1) are foreseen for SeamFrame.

Extent	Resolution	Model chain
Regional	AEnvZ	FSSIM-SCA
EU25	Farmtype	CAPRI-FSSIM-SCA
Farmtype	AEnvZ	FSSIM-AM-APES
Global	Trading block	GTAP-CAPRI
Global	Farmtype	GTAP-CAPRI- simple FSSIM

**Table 1: Problem scales and model chains in SEAMLESS-IF**

### 4.2 Application linking through OpenMI+

As discussed above, we assume that the end-user of SEAMLESS-IF intends to perform an integrated assessment exercise. The end user specifies her/his model chain to be executed through the definition of a project, which is facilitated by the GUI. In general, we identify two subchains to be executed.

- The first one is the biophysical subchain that, based on the user specifications, defines the boundaries of the problem (i.e. geographical extent, number of crops or techniques available, etc). The biophysical chain includes the following components: PEG, PTG, TCG, that commonly constitute AM, SCA, and APES. Given the biophysical configuration, the biophysical chain simulates the biophysical system and produces a set of technical coefficients to be evaluated. The biophysical configuration is the part of the project definition where the user specifies (basically) regions (spatial extent), crops, products, and agricultural management options (technology availability) all associated to context of the experiment, along with the related model configurations (i.e. filters for the generators, APES internal chain, MODCOM integrator, etc). There are two options for setting the biophysical subchain. The first is when the biophysical chain consist only of a single component, the Simple Current Activities component (SCA) that reads from the DB and transforms the current activities to a set of technical coefficients. The second is when the biophysical chain consists of AM and APES, and then the chain involves PEG-PTG-APES-TCG.

- The second chain is the policy assessment subchain, which performs farm optimization and market equilibrium evaluation within the problem boundaries calculated by the "biophysical chain". Given the results of the biophysical chain, a set of user-defined policy-related options (i.e. premiums, subsidies, etc) and some external data (CAPRI database) the "policy assessment chain" concludes with quantified measures of policy consequences in the form of indicators. The policy assessor consists of FSSIM, EXPAMOD, and CAPRI. In principle, based on a specific set of production coefficients that defines the boundaries of the problem, several policy options can be evaluated.

### 4.3 The pull-push execution

The policy assessment chain implements fully the OpenMI pull approach, as all element sets of the data exchanged are a priori known (actually calculated by the biophysical chain). In the biophysical chain, a pull approach is not applicable, as the dimensions of the exchange items are not known a priori, rather they are computed by the models. Therefore, a push approach is adopted in the biophysical model chain. However, a smooth integration is foreseen between the two chains, as in both cases the same datatypes are used (*IResources*), as specified in OpenMI+. The only difference between the two chains is the sequence of execution of the models.

### 4.4 Detailed specification of Experiments

A project may encapsulate several experiments. Each experiment corresponds to a single run of the model chain. All experiments in the same project refer to the same model chain, as the latter is implied by the scales of the problem (see also §4.1). An experiment consists of three parts: the *Outlook*, that defines the trends of the envisioned future, the *Context* that specifies the boundaries of the problem and is specific to the biophysical subchain, and the *Policy option* that defines the conditions for the policy assessment subchain.

The experiment is defined narratively in the pre-modelling phase by the policy expert, and in the modelling phase is materialized by the integrative modeller that translates it to specific parameters and inputs of the selected model.

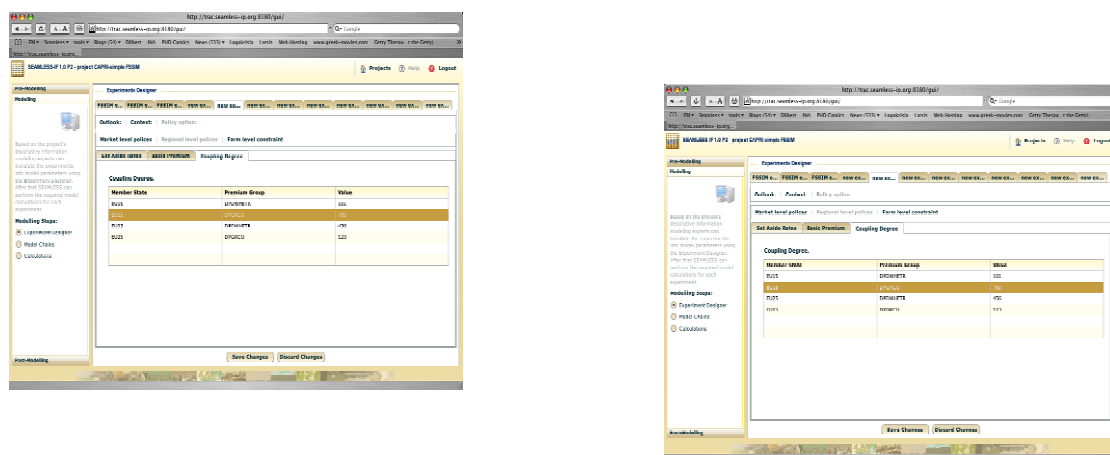


Figure 8: Detailed experiment configuration in the Seam:GUI

## 4.5 Linking the client application with the server

### 4.5.1 Chain execution, server-side queuing model execution and client-side monitoring

When the modeller has completed the configuration of a model chain (that is, of the experiment), s/he can start its execution. This adds the newly configured model chain to the queue of the model chains to be executed on the server. The SeamFrame server provides the queuing facility. The end-user may monitor the queue on his/her client and see which model is next to be executed. When an experiment arrives to the top of the queue, it is passed to the processing environment that executes it on the server. The model chain type is retrieved from the experiment, instantiated, initialized with the experiment id and executed. While models of the chain are finishing, intermediate results and indicator values are linked to the experiment. When the chain is finished, it links the final results to the experiment. As the results are attached to the experiment, they can be visualized with Seam:PRES (Wallman, Almeded, *et al*, 2007).

At present, only one model chain can be executed at a time, but the current design also allows for parallel execution of more model chains at once, if the computing architecture allows for it.

### 4.5.2 Data persistence and linking with data

The Seam:DB data are available through the SEAMLESS Domain manager that has been developed with a Hibernate back-end. The SEAMLESS Domain manager has two incarnations: one is for communicating with the Seam:GUI client and is offered through servlet and XML technology. The second is the Java implementation through which the models are communicating with the DB. In the first case, it obviously has a limited functionality.

Accessing the DB through a single point requires that the data will be provided by WP4 in a universal schema (documented in Belzepkina *et al*, 2007), which has ultimately been generated from the ontologies (i.e provided by WP5 to WP4). Privileges on data modification is mandated by the database access rights, managed by WP4. The D2R approach adopted for Prototype 1 has been discontinued for enabling the use of Enterprise Java Beans and related technologies for client-server applications. In practice, WP3 modellers with the assistance of WP5 have developed ontologies for their model data requirements and using some of the DomainManager tools the ontologies have been translated to EJB classes and database schemata for all exchanged/required data. WP4 and WP3 are filling the data following this schema. The same will be done with project definitions.

Related to intermediate results data exchange and saving, Hibernate technology with a DB backend has been adopted for storing them as well. The Domain Manager of Prototype 1 has been completely re-implemented to be able to use a database persistency storage, rather than a file system (as it used to be). The domain entities specifications are defined in the ontologies and the Domain manager provides two facilities for using them:

- A code generator (similar to the one in Prototype 1). In Version 0.2 instead of generating interfaces for accessing directly instances in the KB, the code generator generates Enterprise Java Beans with default Hibernate annotations, as XML assortments.
- An online utility for managing universally the Enterprise Java Bean persistency layer and its functionality in a server environment. This utility generates and manages

Hibernate sessions through which data can be generated in, retrieved from and searched from the database.



## 5 The visualization environment (post-modelling phase)

Seam:PRES is the visualization tool in SEAMLESS-IF, and is aiming at being a powerful state-of-the-art application that enables the users of SEAMLESS-IF to interactively display impacts, indicators and model outputs. It is a tool that can digest and visually display SEAMLESS model outputs in various ways, to improve the analysis and the dissemination of the model results. These model outputs are either available directly, processed (or copied) into indicator results or, when compared based on scenario's and expected changes, as impacts.

The scope of the first version of Seam:PRES is to retrieve calculated indicator values and display those in three major ways: tables, graphs and maps. Version 0.2 will provide only with tables and simple graphs. Future versions of Seam:PRES will include additional functionality such as options for exporting results, advanced graph visualisations and maps.

Seam:PRES is designed specifically to support the post-modeling related tasks of SEAMLESS-IF. As such it is a task specific client and its detailed requirements are driven by the workflow process and the allocation of tasks to all the tools in SEAMLESS-IF. For the current version all other tasks (those related to pre-modeling and modeling) are implemented in the Seam:GUI tool.

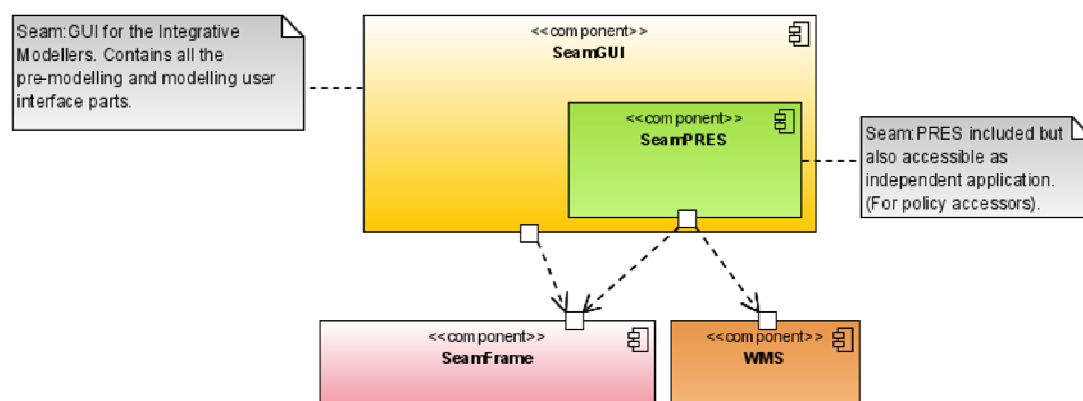


Figure 9: The Seam:PRES Component

In Figure 9 it is shown how Seam:PRES communicates with the same SeamFrame server as Seam:GUI, using the same protocols. Additionally it will retrieve map images from a WMS<sup>11</sup> server.

### 5.1 Seam:PRES Workflow

This section illustrates in general the workflow used in Seam:PRES by users to create different kinds of graphs, maps and other outputs. In Seam:PRES all types of output that can be generated are called *Visualizations*. In the user interaction diagram shown below the boxes in the top line (*Login* and *Available Projects*) are part of the shared functionality with Seam:GUI, and allow a user to log on to the system and select a project to work on.

<sup>11</sup> WMS stands for Web Map Server

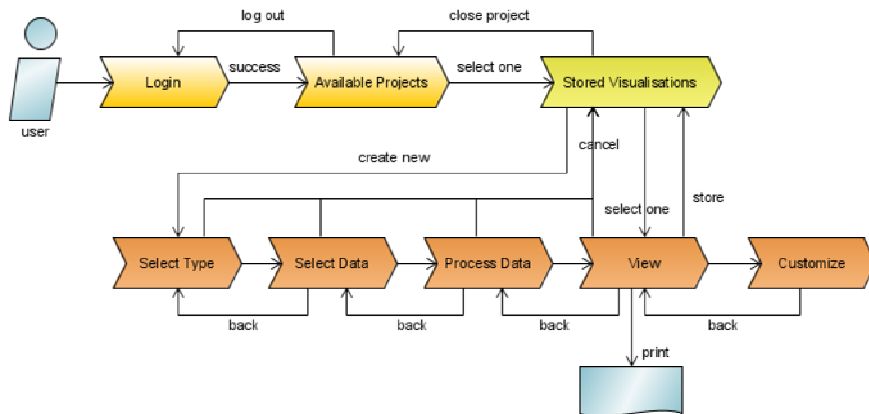


Figure 10: Seam:PRES User Interaction Diagram

Figure 10 shows that once a project has been selected (Seam:PRES does not allow editing of the project, or creating new ones – this is done through Seam:GUI), a user can choose from visualizations already available for the project or create and add new visualizations.

The selected visualization can be opened directly for viewing, or for editing. Editing visualization involves a few steps: *Output type selection, data selection, data processing* and *output customization*. Visualization editing is not available in Version 0.2 of Seam:PRES, but it will be implemented for Versions 0.3 and Final.

### 5.1.1 Select Visualization

Information about visualizations is stored within the project. After a project is opened the visualizations are available in Seam:PRES, with the possibility to add new ones.

### 5.1.2 Visualization Viewing

From the list of available visualizations for the project the user can directly open one for viewing. In order to create the output, Seam:PRES needs to download all the required data from the SEAMLESS server first. This is done asynchronously, and the user can track the progress while the data is being transferred (Figure 11).

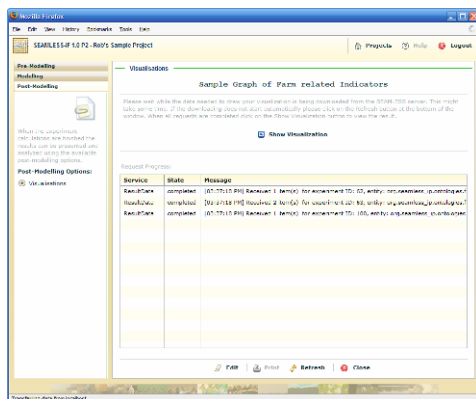


Figure 11: Fetching Data

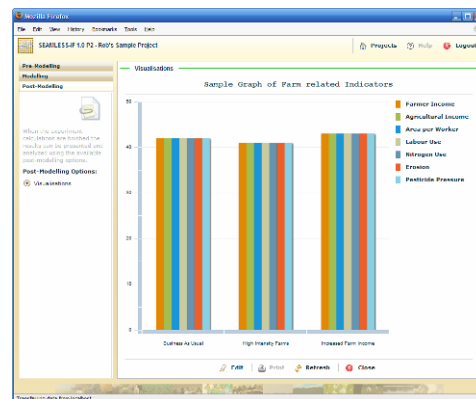


Figure 12: Created Output

When all requested data is retrieved the output can be created and displayed (Figure 12).

### 5.1.3 Visualization Editing

From the visualization selection screen or from the output view the user can start editing the visualization (Figure 13). This includes general information such as a title, description and author(s), and specifications for the desired output type (Figure 14: line graph, bar chart, column chart, map, spider diagram, and so on).

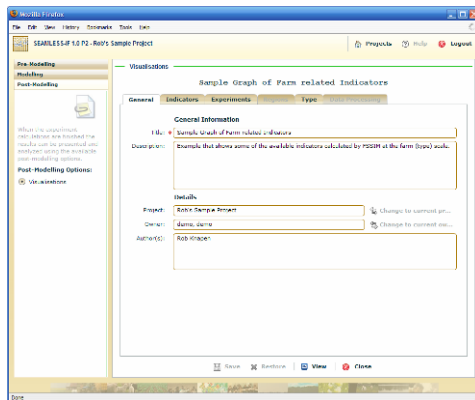


Figure 13: General Settings

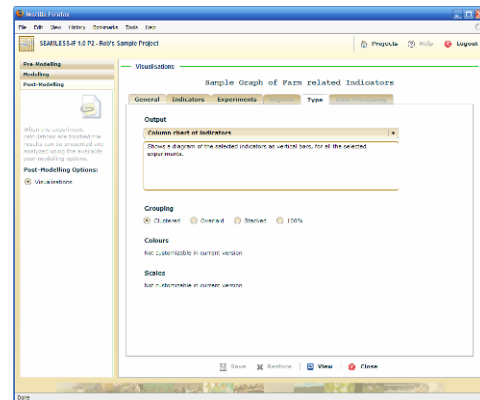


Figure 14: Output Type

Outputs are created based on the principle information available in the projects: the (selected) indicators (Figure 15), experiments and regions. After running the models for the defined experiments the calculation results should be available in the SEAMLESS database (Figure 16). Seam:PRES will attempt to retrieve them by accessing the SeamFrame server.

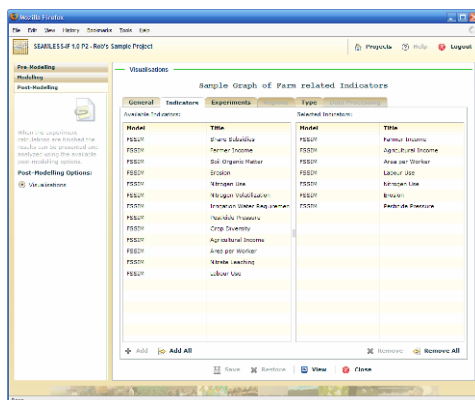


Figure 15: Indicator Selection

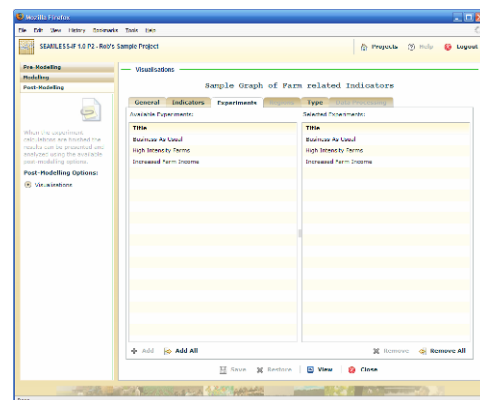


Figure 16: Experiment Selection

## 5.2 Future Versions of Seam:PRES

Seam:PRES is developed incrementally. Functionality of Version 0.2 is limited and mainly serves to show the intended use of the tool in order to get proper feedback for further development.

Additional functionality might include:

- 1) More types of supported outputs.
- 2) Drill down functionality on data / outputs.
- 3) A special type of visualization that can be used as a grouping mechanism, specifying how multiple (child) visualizations should be presented on the screen (e.g. using tabs, a grid, overlays, etc.).
- 4) Data processing capabilities included in Seam:PRES (client side processing), for example to:
  - a) Select on data ranges
  - b) Perform data filtering
  - c) Run calculations on raw data
  - d) Set a specific map extent (select a region to display)
  - e) Change the spatial projection of map data
- 5) More customization options for created outputs. Change colors, scales, labels, and so on.

It should be clarified that Seam:PRES does not intent to visualize all the data in Seam:DB, but only the project outputs and indicators. There are other tools (suggested by Work Package 4 in Belzepkina *et al*, 2007) that are appropriate for accessing the SeamDB contents.

## 5.3 Design of Seam:PRES

The design of Seam:PRES is split into a client part running on the computer of the user and a server part. The server part is included into SeamFrame and consists of the few specific servlets needed by Seam:PRES. These are written in the Java programming language. In general they take care of the retrieval of the calculation result data and the spatial data from the SEAMLESS database. A *GetIndicatorDataManager* class is used to isolate the knowledge of how several database tables need to be traversed to get to the results for a specific experiment and indicator. Once found the data is returned in the usual SeamFrame XML return format.

The client part is programmed in Adobe Flex 2 and integrated into the Seam:GUI. It is possible to create a stand-alone Seam:PRES application as well. This would only allow viewing of project results, but that might be sufficient for a certain group of users.

### 5.3.1 Data Services

The client application uses the *ServiceCenter* to get access to the data services that can be used to retrieve and store information. Access is through an interface, and concrete implementations of the data services might provide different types of local or remote storage. Seam:PRES uses the *ResultDataServices*, *SpatialDataServices* and *VisualizationServices*.

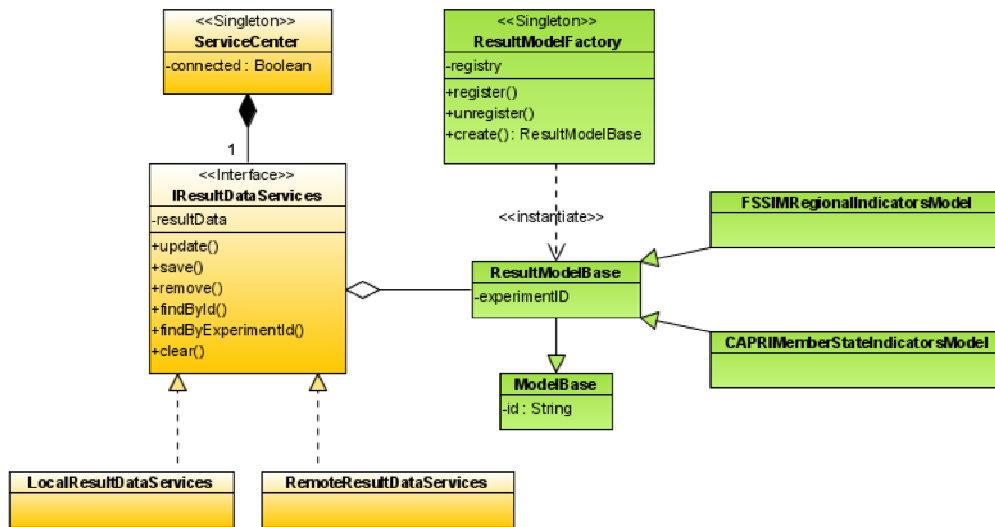


Figure 17: Sample Data Services

In Figure 17 it is shown how data services make use of client-side model classes. These represent the domain model and reflect the domain model used on the SeamFrame server. Using data transfer objects and data transfer object factories information is exchanged between the domain model layers of the client and the server.

### 5.3.2 Asynchronous Processing

In order to create the output for a visualization Seam:PRES needs to receive the required data from the SeamFrame server. Potentially this involves a large amount of data and might take some time. Figure 18 describes how requests to the server are handled asynchronously, using the *Asynchronous Completion Token (ACT)* pattern. Support for this, including the needed event mechanism is already available in Flex 2. The ACT pattern is implemented in the *ResultDataProcessor* class.

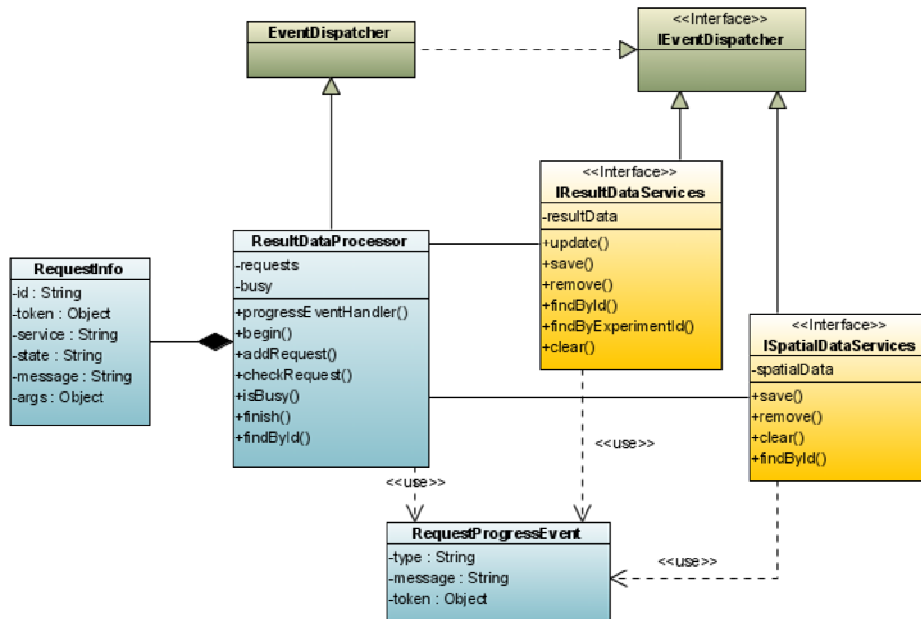


Figure 18: ResultDataProcessor for Asynchronous Data Retrieval

### 5.3.3 Visualization Factory

A factory design pattern is used for the creation of the different types of visualization outputs that must be supported (Figure 19). This ensures the required flexibility and extensibility. Specific classes will be then generated by the factory to handle data pertaining to experiments, region, classes and indicators.

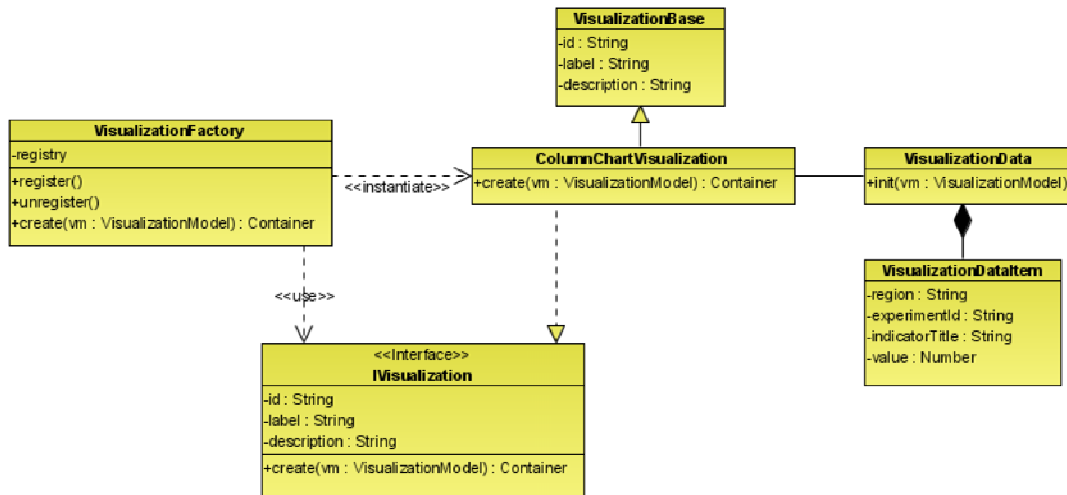


Figure 19: Visualization Factory

---

## References

- Athanasiadis, I., Huber, D., Rizzoli, A.E. & Villa, F. (2006). First prototype components for knowledge and data management. SEAMLESS-IP Project Deliverable D5.4.2
- Belzepkina, I. & others (2007). Prototype 2 of SEAMLESS-IF. SEAMLESS-IP Project Deliverable D1.4.2.
- van Evert, F.K., Rizzoli, A.E., Li, H., Svensson, M. & Jonsson, B. (2006). First prototype of the modelling environment. SEAMLESS-IP Project Deliverable D5.4.1.
- Huber, D., Knapen, R., Athanasiadis, I., Li, H., Jonsson, B., Oliverio, F., Rizzoli, A.E. & van der Wal, T. (2006). SeamFrame alpha prototype. SEAMLESS-IP Project Deliverable D5.4.3.
- Janssen S., Li, H., Athanasiadis I. N., & David, H., 2007. Explanation of the Project Definition, Working paper of the DOT.Force, Internal SEAMLESS-IP communication.
- Janssen, S.J.C, Wien, J.J.F., Li, H. Athanasiadis, I. N., Ewert, F., Knapen, M.J.R, Huber, D., Therond, O., Rizzoli, A.E., Belhouchette, H., Svensson, M. & van Ittersum, M.K., Defining projects and scenarios for integrated modelling using ontology, MODSIM 2007 Conference.
- Knapen, R., Huber, D. & Rizzoli, A. (2007). Processing Environment Prototype. SEAMLESS-IP Project Deliverable D5.4.3.1.
- Rizzoli, A., Athanasiadis, I., Donatelli, M., Huber, D., Muetzelfeldt, R., van Evert, F., van den Broek, M., van der Wal, M. & Villa, F. (2005). Overall architectural design of SeamFrame. SEAMLESS-IP, Seamless report 7.
- Rizzoli, A.E., Athanasiadis, I.N., Li, H., Knapen, R., Wien, J., Donatelli, M., Villa, F., Huber, D., Senaldi, F., Svensson, M. & Jonsson, B. (2006). First iteration of SEAMFRAME design. SEAMLESS-IP Project Deliverable D5.3.8.
- Rizzoli, A.E., van Evert, F.K., Li, H., Athanasiadis, I., van der Wal, T., Knapen, R. & Huber, D. (2006). Design of the modelling and processing environments. SEAMLESS-IP Project Deliverable D5.3.3-6.
- Rizzoli, A.E., Svensson, M., Rowe, E., Donatelli, M., Muetzelfeldt, R., van der Wal, T., van Evert, F. & Villa, F. (2005). Modelling Framework (SeamFrame) requirements. Seamless Report 6, SEAMLESS-IP, ISBN: 90-8585-034-7.
- Svensson, M.G.E., Jonsson, B. & Wallman, P. (2006). User Requirement Analysis: Analysis for Prototype 1. SEAMLESS-IP Project Deliverable D5.2.4.
- Svensson, M.G.E., Rizzoli, A., van der Wal, T., Donatelli, M., F.v. Evert, Muetzelfeldt, R. & Villa, F. (2005). Analysis of requirements: the user perspective. SEAMLESS-IP Project Deliverable D5.2.1.
- Svensson, M.G.E., Wallman, P., Almeded, K., Wien, J., Knapen, R. & Rizzoli, A. (2007). Software Quality Assurance. SEAMLESS-IP Project Deliverable D5.6.2.1.
- Therond, O., Taverne, M., Wery, J., Josien, E., Belhouchette, H., Bergez, J., Rapidel, B., Thenard, V. & Bigot, G. (2007). Proposals from WP6 for revision of SEAMLESS-IF procedure and functionalities in Prototype 2. SEAMLESS-IP Project Deliverable D6.5.4.2.

- 
- Villa, F. (2005). Design of the Domain Manager component of SeamFrame. SEAMLESS-IP Project Deliverable D5.3.2.
- Villa, F. & Athanasiadis, I.N. (2006). Guidelines for populating the demo version knowledge base. SEAMLESS-IP Project Deliverable D4.2.1.
- Villa, F., Athanasiadis, I.N. & Rizzoli, A.E. (2007). Software deliverable: SEAMLESS Knowledge Manager (and accompanied documentation). SEAMLESS-IP Project Deliverable D4.2.2.2.
- van der Wal, T., Rizzoli, A., Svensson, M., Villa, F., Knapen, R. & Athanasiadis, I.N. (2006). The conceptual design of SeamFrame. SEAMLESS-IP Seamless report 21.
- Wallman, P., Almehed, K., Svensson, M. & Knapen, R. (2007). Manual of the SEAMLESS-IF Prototype 2. SEAMLESS-IP Project Deliverable D5.6.3.2
- Wallman, P., Bezlepkina, I, Knapen, I. (2006) How to Handle SEAMLESS-IF Prototype 1: A short how-to description. SEAMLESS-IP Project Deliverable D5.6.3.1.