In Information Technologies in Environmental Engineering, **1** :3-11, Jan 2008.

Semantic mediation for environmental model components integration

Ioannis N. Athanasiadis[1] and Sander Janssen[2]

[1]Dalle Molle Institute for Artificial Intelligence, USI/SUPSI, Lugano, Switzerland
`ioannis@idsia.ch`
[2] Plant Production Systems, Wageningen University, Wageningen, The Netherlands
`sander.janssen@wur.nl`

Abstract — **Within Seamless project, a set of constituent agricultural simulation and optimization models is required to be integrated for facilitating assessment studies. Each one of the models has been developed by a different research group, according to dissimilar modeling approaches, implementation designs, and programming tools. As a mediator among these heterogeneous constituent peers, we introduce the Seamless Knowledge Manager component for incubating the data exchanged by the models. The Seamless Knowledge Manager has been developed following a novel approach that exploits ontologies and semantic modeling. Specifically, a declarative approach has been utilized for specifying the data exchanged by the models and has been used as the basis for software development and integration. This paper presents in detail the methodology used for developing the Knowledge Manager and two alternative implementations. The architecture is demonstrated for integrating modules generating agricultural management alternatives.**

Index Terms — **Modeling and Simulation, Semantic-mediation, Software Engineering, Agricultural Management, Semantic Web, Ontologies, Integrated Modeling Frameworks**

I. INTRODUCTION

A. Managing the complexity of farming systems

Farming Systems Research studies the agro-ecosystems at the farm level, that is agriculture and its interaction with other ecosystems and society. The agro-ecosystems are highly complex (Kropff et al., 2001) due to the many feedbacks between natural processes and human factors, the high, geographical diversity in agro-ecosystems, and the limited understanding of some of the involved processes. This has resulted in an abundant number of field, cropping system or farm level models, each developed for specific purposes. These models are realized as software components that are hardly re-usable and it is difficult if not impossible to integrate them with other models, in order to perform integrated analyses (Rizzoli *et al.* to appear, Athanasiadis *et al.* to appear). One of the reasons for this, is the poor semantics that usually characterize farm model implementations. In this paper, we demonstrate how semantic modeling can help to formalize the knowledge captured by models; in order to subsequently facilitate model knowledge re-usability and exchangeability, and present two alternative architectures for data exchange in integrated modeling frameworks through knowledge management techniques.

B. The SEAMLESS integrated project experience

The SEAMLESS integrated project (Van Ittersum et al., in press) develops a computerized, integrated framework (SEAMLESS-IF) to assess and compare, ex-ante, alternative agricultural and environmental policy options, allowing analysis across different scales (from field, farm to region and EU), dimensions of sustainability (economic, social, environmental and institutional) and for a broad range of issues and agents of change. For the specific questions to be analyzed, a set of models and tools are required to be integrated within SEAMLESS-IF, including:

- *combinatorial models*, as those required for the generation of agricultural management alternatives;
- *biophysical models* for crop growth simulation;
- *economical models* dealing both with farmer income optimization and agricultural product market equilibrium;
- *decision making models*, including social, economic and environmental indicators;
- *databases*, providing with reference agro-economic, meteorological and landscape data, at various temporal and spatial scales.

The diversity of the tools at hand reveals that integration is needed not only at a technical/software level, but also in a conceptual/scientific level, as the constituent approaches employ different paradigms. The SEAMLESS-IF is a platform for farming systems model integration aiming to produce outputs for policy makers. Interdisciplinary research groups contribute with agricultural, environmental, economic and social models, at different scales, which consequently are seamlessly integrated, while maintaining the logical independence of data, models and simulation procedures and optimization algorithms. This is achievable through the development of independent components which are integrated in an open environment (Athanasiadis 2007). The need of sharing scientific knowledge incorporated in the constituent models emerges, and goes further than simply maintaining a shared dictionary for model inputs and outputs. A common reference is required that aims to:

- build a *shared view on the systems modeled*, through identifying and resolving ambiguities in terms and data structures;
- facilitate model integration in a sound way, by overcoming scaling problems that are typically remain hidden in low levels;
- contribute with added value to model development, by targeting reusability, interoperability and extensibility.

These three goals are achieved in SEAMLESS-IF by employing semantic modeling for facilitating agricultural model integration, and developing a Knowledge Manager component, as a mediator for among heterogeneous constituent modules, that incubates the data exchanged by the models.

The rest of the paper structured as follows: The following section (II) introduces semantic modelling and ontologies. In section III we present the Seamless Knowledge Manager and the related design choices. The Seamless Knowledge Manager is demonstrated for the integration of the Agricultural Management components. In Seaction IV we present the problem at hand and the developed ontologies, and in Section V the system implementation and results. The paper concludes with the discussion in section XI.

II. ONTOLOGIES AND SEMANTIC MODELING

The last few years, ontologies, knowledge bases and the semantic web attract the interest of the research community. An ontology in computer science is considered as a specification of a conceptualization (Gruber 1993). It is a formalization that could be expressed in a machine readable format, i.e. the Web Ontology Language (McGuinness & van Harmelen, 2004). This provisionally allows a software system to "comprehend" a conceptual schema and makes it possible to reason on it. A knowledge base is the result of expressing the information related to a domain in line with a given domain ontology. Typically this activity involves instantiating data with respect to an ontological definition. Ontologies can be seen as a medium for open software environments (Willmott et al. 2002), where software agents provide semantic web services under strict contracts, as for example in the AgentLink project.

SEAMLESS-IF is considered as an open modular simulation environment for agriculture, science and policy., where modelers are thought of as communities of "knowledge workers" in the fashion of Warren (2006). These knowledge workers translate their domain knowledge both into a model and an ontology, which then can be reused within the framework. The ontology specifies the characteristics of the system at hand, while the model specifies its behavior. *By having part of the modelers' knowledge about the domain at hand defined in an ontology, the model implementation can be realized in a semantically aware fashion, to which we refer as* ***semantic modeling***.

Semantic modeling has been employed in natural systems research for metadata support (Brilhante & Robertson, 2001; Brilhante *et al* 2006), for enabling the transparent integration, reorganization and discovery of natural systems knowledge (Villa 2007), the transparent and sound economic valuation of ecosystem services (Villa *et al.*, 2007), the enrichment of environmental software model interfaces (Athanasiadis et al., to appear). For an in-depth review of semantic modeling approaches in ecological modeling we point the reader to Villa *et al.* (to appear).

III. THE SEAMLESS KNOWLEDGE MANAGER

The objective of this paper is to demonstrate the advantages of semantic modeling in capturing the complexity of farming systems by making knowledge relationships explicit. The development of an ontology formalizing the farming systems studied in SEAMLESS-IF and the use of a Knowledge Manager (KM for short) for linking models and applications together is the key to this process. For the time, let us assume that the knowledge workers (i.e. the modelers) are willing to provide with some ontologies, defining the modeling universe, or that such ontologies are available and ready to be used. The question that arises is how can we utilize practically these ontologies in an integrated framework? In SEAMLESS-IF we considered the Knowledge Manager module as a facility that is capable to:

(a) register and manage domain ontologies;
(b) manipulate data sources and make their contents available, with respect to the domain ontologies;
(c) realize links between components, and facilitate the data exchange;
(d) provide interfaces with external applications.

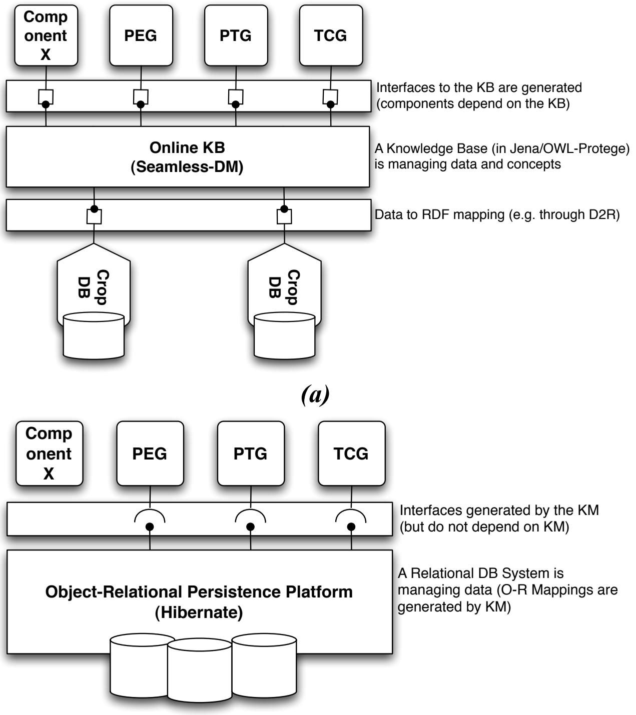
A KM with this behavior is able to mediate between various data sources and model components (see Figure 1), in order to ensure data integration based on strict definitions (specified by the domain ontologies). We identify two major modes of operation for the Knowledge Manager, that imply the design and implementation choices to be made.

First comes the "online operation", which means that the Knowledge Manager is a component at the same level with the model components (Figure 1a). The KM operates as an active mediator that is responsible for receiving data from the models and interfacing with the DB. In this case, through the KM a tight integration can be achieved, implementing a centralized approach.

Second comes the "offline mode" of a KM, in which it provides with the interfaces for models and databases, but does not intervene actively to their communication (Figure 1a). This resembles the "blackboard" architectures, as the model components can directly exchange data, and the mediation is indirect. This approach is more suitable for loosely coupled component integration, as in web-systems.

Both approaches have been implemented during the SEAMLESS project lifetime. The first was realized for the standalone version and the second for the web-based version of SEAMLESS-IF. In both cases the KM was able to manage OWL domain ontologies specifying the data to be exchanged across the models.

The SEAMLESS "online" KM has been implemented on top of ProtegeOWL library and incubates a knowledge base containing both the ontologies (concepts) and the data (instances). Through ontology-derived software interfaces that



(a)

(b)

Fig. 1. *The SEAMLESS Knowledge Manager architecture operating "online" (a) and "offline" (b)*

specify the data structures to be exchanged, the models connect to the KM. The model components in this case depend on the KM component for inter-communicating. Also in this implementation we used the D2R libraries (Bizer 2003) for loading data in the KM. Following the "online" approach it is required to map existing database schemata may to the domain ontology, which is can be a complicated task. The "online" KM approach takes advantage of the semantic technologies, as the KM plays a central role in the system architecture, and its reasoning capabilities can be exploited to the maximum. At the same time, the KM may become a bottleneck for the system, as it manages all the data traffic.

On the other hand, the SEAMLESS "offline" KM architecture employed object-relational mappings for the communication of the model components, through a persistence layer, where an underlying relational database plays the role of the blackboard for data exchange. In this case, both the model software interfaces and the database schema has been generated from ontologies as presented in Athanasiadis *et al*. (2007a,b). Specifically, Enterprise Java Beans and Hibernate technologies have been employed for the software interfaces and object persistency. In this case, the model components are do not depend on the KM implementation or interface, only to the domain ontologies that specify their interfaces. Such a design choice is more suitable for open environments, as for web-based implementations. However, in this case the KM is not active at execution time, which means that reasoning cannot be extensively used. For most simulations this is affordable, however it limits the KM-offered services to semantically-mediated integration (see also Villa *et al*, to appear).

The use of a Knowledge Manager maximizes the substitutability of SEAMLESS components, by expressing the knowledge related to component interfaces in a declarative way, using a domain ontology (as discussed for a similar application in Athanasiadis *et al* to appear). This is demonstrated in the following sections, where we detail how a semantic modeling approach has been used within SEAMLESS for the integration of the Agricultural Management (AM for short) components. Following the problem definition and the component functionality specification, we come back to the issue of defining the ontologies for the particular domain and the knowledge workers, and exemplify how they have been used within SEAMLESS-IF for component integration.

IV. THE AGRICULTURAL MANAGEMENT MODULE (AM)

A. Problem definition

A pivot element of the SEAMLESS-IF modeling framework is farm level modeling. Farm level modeling aims to assess the impact of policy changes and technological innovations on farmer behavior now and in the future (Janssen and Van Ittersum, 2007). To achieve these goals it is required to specify the possible agricultural activities that a farmer can apply while using his/her resources and satisfying his/her objectives (Donatelli *et al*. 2006). On a purely arable farm, agricultural activities constitute of activities related to growing different crops with a range of different management practices. In this paper, we define an agricultural activity as a coherent set of crops (a crop rotation) with associated crop management and corresponding inputs, e.g. fertilizer, seed, pesticides, and outputs, e.g. marketable products, production of feedstuffs for on-farm use and environmental effects (Ten Berge et al., 2000; Van Ittersum and Rabbinge, 1997). A rotation is a succession of crops in time (cropping sequence) and space (cropping pattern), where the last crop is the predecessor of the first crop (creating a loop). Crop management is a complete set of agronomic inputs (e.g. management practices) characterized by type, level, timing and application technique (Van Ittersum and Rabbinge, 1997).

A distinction can be made between current and alternative agricultural activities. Alternative agricultural activities are agricultural activities that are not currently used, but are technically feasible alternatives for the future, often technological innovations or newly developed cropping practices, while current activities are agricultural activities that are currently being practiced and can be derived from observed data.

In a farm system model, mathematical programming, usually linear constraints and a non-linear objective function, (Janssen and Van Ittersum, 2007) is used to 'simulate' the allocation of current and alternative activities to the available resources, while satisfying the objective and meeting the

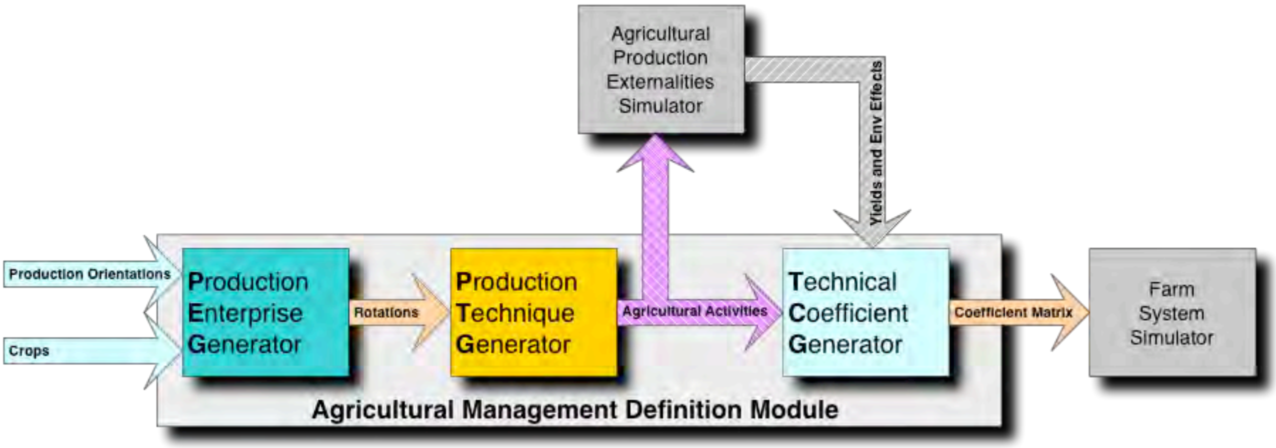


Fig. 2. *The abstract architecture of the AM. Its three components are shown, along with the external applications.*

policy constraints.

B. *Abstract architecture and provided services*

The main objective of the Agricultural Management module (AM) is dual. First is to describe, generate and quantify alternative and current agricultural activities that can be evaluated by a dynamic crop simulation model (Agricultural Production Externalities Simulator: APES, http://www.apesimulator.it) in terms of yields and environmental effects. Second is to generate a set of fully quantified agricultural activities that can serve as inputs to a farm level optimization model in which the possible activities are confronted with farm endowments and farmer's objectives (Farm System Simulator: FSSIM). In this respect, AM serves both biophysical and optimization models by preparing feasible/possible agricultural activities to be simulated or optimized respectively, as shown in Fig. 2. Specifically, in this paper we present the component integration procedure for the part of the AM that formulates alternative agricultural activities for arable farming systems. In principle, a similar procedure could be carried out for livestock farming systems.

The AM for alternative agricultural activities consists of three cooperative components:

(i) The *Production Enterprise Generator* (PEG), that given a set of production orientations and crops generates all feasible crop rotations,

(ii) The *Production Technique Generator* (PTG), that is responsible for generating the crop management options of the rotations, and

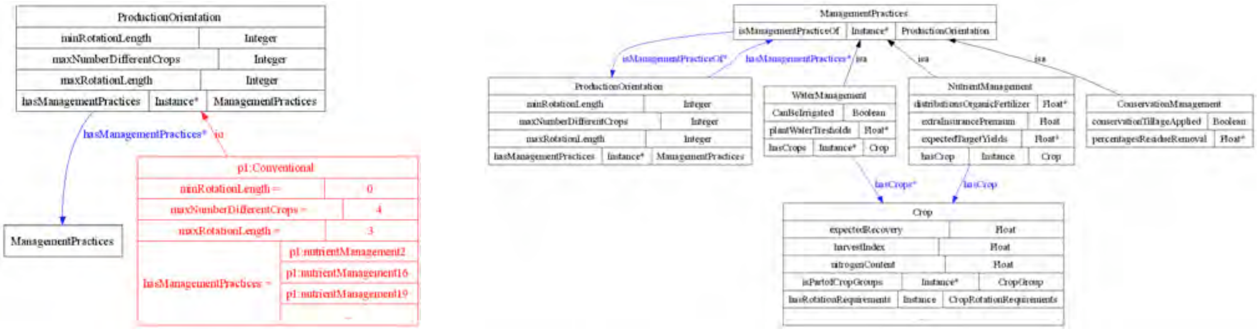
(iii) The *Technical Coefficient Generator* (TCG), that quantifies, collects and formats the input data for the farm model.

An abstract architecture of the AM is presented in Fig. 2. The features of each of the AM components are detailed below.

The **Production Enterprise Generator** (PEG) is the component that generates a set of feasible rotations of the farm based on crop suitability filters and rotation suitability filters. In principle, all crops that may be grown in a given biophysical environment can be combined into rotations. However, not all of these combinations are agronomically feasible or desirable. The first functionality of the PEG is a crop rotation generation facility (Athanasiadis *et al.*, 2007c) that generates as all possible rearrangements of the available crops, while excludes cyclically equivalent rotations from the solution space. The production orientation limits the length of the rotation and the number of different crops in a rotation.

The second functionality of the PEG is to further limit the number of generated rotations by applying crop suitability filters. The latter determine which crops can be grown in a certain biophysical environment, (i.e. the farm where the rotation is applied). This part is based on a component called ROTAT, developed by Dogliotti et al. (2003), that eliminates all rotations that are not feasible according to rotational filters. Ten crop suitability and nine rotation suitability filters are included in PEG, that can be switched on or off as desired by the user. A suitability filter factory and a rotational filter factory are used to create the respective filters at run time.

The **Production Technique Generator** (PTG) is a component to generate alternative agricultural activities on the basis of the feasible set of rotations by attaching crop management information to each crop in the rotation. A crop management is a complete set of agronomic inputs characterized by type, level, timing and application technique (Van Ittersum and Rabbinge, 1997). Crop management exists of five management practices: water management, general management (sowing, harvesting and field inspection), nutrient management, conservation management and weed, pest, and disease management. For each of these management practices the PTG has one management generator, which generates a set of events for an aspect of crop management. An event is one operation that takes place during the growing season of the crop, for example sowing, fertilization, irrigation, harvesting, field inspection, etc. The rule-impact approach discussed in Donatelli *et al.* (2006) has been adopted for generating the events. The five management generators generate the events based on the implements and inputs a farm uses, the specification of the management practices as part of the production orientation and the rotations provided by PEG. Each of the management generators can be switched on or off independently as a management factory is used to create



(a) **(b)**

Fig. 3. *Snapshots of the developed ontology: (a) The Production Orientation class and an example instance "Conventional", (b) The Management Practices class and their characteristics.*

management generator objects at run time. The output of the PTG is fed into the dynamic crop simulation model APES. Note that APES simulates yields and environmental effects for each crop with associated management in a rotation.

Finally, the **Technical Coefficient Generator** (TCG) links the alternative agricultural activities generated by PEG, PTG and APES to socio-economic coefficients by simple calculations and prepares the inputs for the farm model in a technical coefficients matrix (see Fig.2). The TCG can produce a technical coefficients matrix for the farm model on different scales: on a daily, yearly or seasonal basis, and on a rotational or individual crop in the rotation basis, as is dependent on the request from the farm model. The simple calculations carried out by the TCG are on variable costs and labour requirements.

C. An Ontology as a mediator for the Agricultural Management module

Following the semantic modelling paradigm, specifying the model component functionality, that is the component behavior, is not enough. System attributes and characteristics are specified in a declarative fashion through an domain ontology. The modellers, playing the knowledge worker role, through the domain ontology make explicit their specification of the system. Our aim was to develop AM components with open interfaces that adhere to a shared domain ontology, which we discuss here.

In order to systematically formalize our knowledge on Farming Systems Research, we developed an ontology for specifying the interfaces of AM to external applications (Databases, APES, FSSIM), along with the inter-component communication. In this way, AM components behave similarly to software agents providing information services, that are explicitly defined using ontologies (see Athanasiadis 2007). The AM ontology links to the core SEAMLESS ontologies (discussed in Rizzoli *et al*. in press), and are available at http://ontologies.seamless-ip.org).

Domain conceptualization is a communal process that involved the communities of scientist involved in seamless and domain ontologies for agricultural management emerged as a distillation of this process. Several meetings have been held (both on line and face-to-face) for clarifying various aspects of the concepts involved in SEAMLESS-IF and AM in particular.

A small example of the developed ontology is presented in the following (Figure 3a), where the concept of a Production Orientation is presented in the form of a conceptual map. A *Production Orientation* limits the length of the rotation and the amount of different crops in a rotation and directs the management practices associated with the different crops. So, it specifies three data type properties (the minimum and maximum rotation lengths, and the maximum number of crops), and an object property (*hasManagementPractices*) that associates it with a set of *Management Practices*. In Fig. 3(a), we present an example instance of a *Production Orientation*, called "*Conventional*". The *Management Practices* of a *Production Orientation* on their turn are defined similarly as presented in Figure 2(b), where the characteristics of and the relations between the *Production Orientation* and *Management Practices* are illustrated.

Following an iterative development procedure, all intrinsic concepts that AM deals with and their properties have been defined in the domain ontology. This process involved several iterative reviews of the ontology among the knowledge worker and the knowledge engineer. The result of this activity was a declarative formalization of concepts that AM deals with. In this respect, we do not restrain the components developed by their current implementation languages or internal structures. Rather future extension or substitution can be supported if new components are developed with respect to the same (or an equivalent/orthogonal) ontology.

Based on the AM ontology, the Knowledge Manager may generate for us the software component intefaces for the "online" or the "offline" approach. Specifically, from the ontologies the KM may provide with:

(a) a database registration facility, that enables different data sources to be used directly as system inputs, and
(b) the generation of data types, reflecting the ontology structure, based on which model algorithms can be directly programmed.

Both of them are platform-specific interpretations of the generic domain definitions provided in the ontology. However,

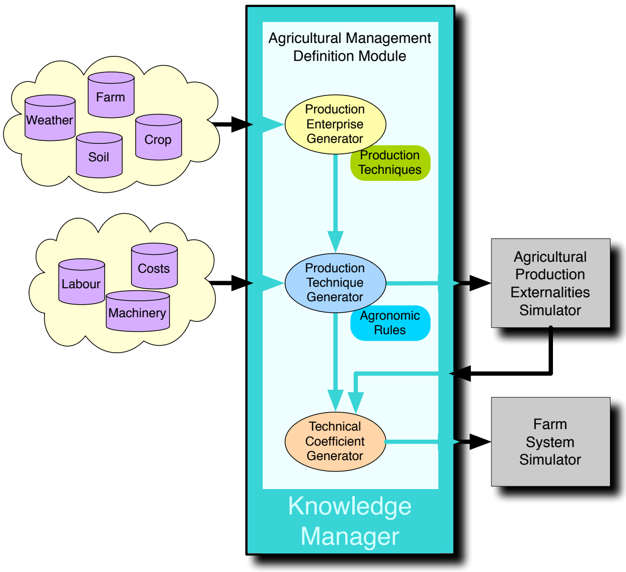


Fig. 4. *AM for alternative activities and its components: algorithms, databases and connections.*

such supporting facilities can speed up the development a lot. In the following Section VI we present how the use of ontologies in conjunction with the KM was realized for enabling all AM components to access a shared knowledge base for accessing existing databases or sharing data generated the system.

V. SYSTEM REALIZATION AND IMPLEMENTATION

A. Agricultural Management module Implementation

The three components of the AM for alternative activities have been developed in Java, while the Knowledge Manager was implemented as an extension of the Java Protégé-OWL API (Horridge et al., 2004). Both "online" and "offline" approaches has been developed in SEAMLESS. The AM implementation is illustrated in Figure 4. The KM shell provides to the AM components with the appropriate interfaces to external data sources and applications and facilitates the linking of AM components. The added value of such an approach is three-fold:

(a) it completely separates algorithms from data and user interfaces,
(b) it facilitates easy linkage to external database sources and user interfaces, and
(c) it makes algorithms easily extensible and comprehensible.

These objectives were achieved by:

(i) linking the algorithms to databases through the ontology,
(ii) developing the user interface at the very end,
(iii) using design patterns, especially factory and strategy pattern where possible in the algorithms.

Following the workflow of Fig.2, the AM process starts with the invocation of PEG. All required inputs related to crops, farm, soils, etc are fetched from external sources via the KM layer. In the "online" approach the KM is a component itself that load the data and provides then as OWL instances. In the "offline" approch, model components are directly acquiring the data as objects through a persistence layer. In both cases, the model components are not tightened to a specific database(s) structure, rather it relies on the KM to acquire mediate for retrieving the required information. In the "online" approach this is achieved using D2R (data to RDF) mappings and in the "offline" approach through hibernate (object-relational) mappings.

When the PEG is executed, it creates the rotations that comply with the suitability and rotational filters that the user selected. These are transferred to PTG through the KM infrastrucutre. Then, PTG calculates all possible agricultural activities, based on the appropriate management practices for each crop, and the farm-specific data related to machinery, labour and costs. Once again the data sources are decoupled from the algorithm implementation. Next, the agricultural activities resulting from the PTG are communicated to APES (an external application), which simulates each one of those, calculating the yields and environmental effects. APES results are captured by the KM interface and fed to the TCG. Finally, TCG is executed for generating the Coefficient Matrix for the FSSIM optimizer. In this respect, the AM facilitates the linking of APES with FSSIM, in an open, "loose" coupling based on ontology-specified interfaces.

In the following paragraphs we present how the ontology was utilized for data type code generation, to facilitate on semantically aware development, and how the KM was utilized for connecting to external data sources.

B. Use of the ontology for code generation and semantic-rich development

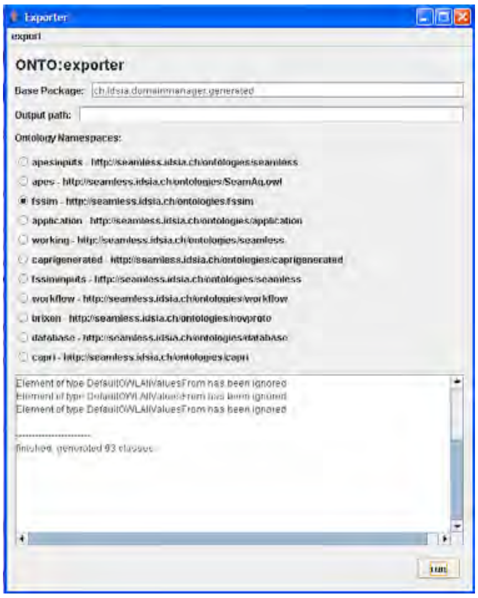
The ontology structure of the AM was used directly for developing the interfaces of the AM by generating the source code of the data types exchanged among components and applications. Although Protégé already includes a plug-in for code generation, a new code generator was developed, called ONTO:Exporter. This was for two reasons. Firstly, because the Protégé code generator is outdated, and takes no advantage of powerful implementation practices, such as code annotations and generics. Secondly, Protégé used a class implementation for each OWL Class. In contrast the ONTO:Exporter uses interfaces for OWL classes. For the "online" approach a common proxy class is used for accessing the knowledge base, while in the "offline" mode the ONTO:Exporter generates with the object-relational mappings as well. In both cases, this ensures that the model development relies only to the software interfaces generated and not to a particular implementation.

In the "online" mode of operation, all information exchanged by the model components flows via the knowledge base, which allows performing semantic checks at run time for ensuring the soundness of the link among components and applications.

The Graphical User Interface of the ONTO:Exporter application for generating code interfaces is shown in Fig. 5 along with a code segment from the generated interface of the Production Orientation ontology class example, discussed above.

```
        package ch.idsia.domainmanager.generated.fs;

        import java.util.Collection;
...


@ClassURI("http://seamless.idsia.ch/ontologies/fssim#ProductionOrientation")
public interface ProductionOrientation extends PEGDatatype {


                        // Datatype Functional Property fs:minRotationLength
                        @PropertyURI("http://seamless.idsia.ch/ontologies/fssim#minRotationLength")
                        public Integer getMinRotationLength();
                        @PropertyURI("http://seamless.idsia.ch/ontologies/fssim#minRotationLength")
                        public void setMinRotationLength(Integer var);

        // (Properties maxNumberDifferentCrops and maxRotationLength omitted for simplicity)

          // Object type non-Functional Property fs:hasManagementPractices
                        @PropertyURI("http://seamless.idsia.ch/ontologies/fssim#hasManagementPractices")
                        public Collection< ManagementPractices> getHasManagementPractices();
                        // Object type non-Functional Propertyfs:hasManagementPractices
                        @PropertyURI("http://seamless.idsia.ch/ontologies/fssim#hasManagementPractices")
                        public void setHasManagementPractices(Collection<ManagementPractices> val);
                        // Object type Propertyfs:hasManagementPractices
                        @PropertyURI("http://seamless.idsia.ch/ontologies/fssim#hasManagementPractices")
                        public void addHasManagementPractices(ManagementPractices val);
                        // Object type Propertyfs:hasManagementPractices
                        @PropertyURI("http://seamless.idsia.ch/ontologies/fssim#hasManagementPractices")
                        public void removeHasManagementPractices(ManagementPractices val);
}
```

(a)



Base Package: ch.idsia.domainmanager.generated

Output path:

Ontology Namespaces:

○ apesinputs - http://seamless.idsia.ch/ontologies/seamless
○ apes - http://seamless.idsia.ch/ontologies/SeamAg.owl
● fssim - http://seamless.idsia.ch/ontologies/fssim
○ application - http://seamless.idsia.ch/ontologies/application
○ working - http://seamless.idsia.ch/ontologies/seamless
○ caprigenerated - http://seamless.idsia.ch/ontologies/caprigenerated
○ fssiminputs - http://seamless.idsia.ch/ontologies/seamless
○ workflow - http://seamless.idsia.ch/ontologies/workflow
○ brixen - http://seamless.idsia.ch/ontologies/novproto
○ database - http://seamless.idsia.ch/ontologies/database
○ capri - http://seamless.idsia.ch/ontologies/capri

```
Element of type DefaultOWLAllValuesFrom has been ignored
Element of type DefaultOWLAllValuesFrom has been ignored
Element of type DefaultOWLAllValuesFrom has been ignored

-----------------------
finished, generated 93 classes
```

(b)

Fig. 5. *(a) Code segment of the generated software interface 'ProductionOrientation' that is exported from the ontology. (b) The Onto:Exporter GUI through which the user may export code from more than one ontologies.*

C. System execution

The AM system developed has been tested on existing data coming from the region of Flevoland, The Netherlands. As mentioned above database sources and schemas are entirely decoupled from the developed system. For testing the "online" approach, we employed the D2R language and library (Bizer 2003) for defining a mapping between an SQL Database (accessed via ODBC) and the AM ontology. This functionality is integrated in the Knowledge Manager for accessing external sources that are transformed to instances of the domain ontology. For testing the "offline" approach we employed Hibernate (Bauer & King, 2006) and that generates a database schema that embodies the data defined by the ontologies.

Finally, we demonstrated the use of the system with data from Flevoland. Specifically, PEG was run for 10 crops, a conventional production orientation with a maximum rotation length of 3 years and a farm type in Flevoland. The crops were `Carrot`, `Onion`, `Pea`, `Springbarley`, `Springwheat`, `Tulip`, `Lucerne`, `Fibrehemp`, `Grassseed` and `Sunflower`. All Suitability Filters and Rotational Filters in the PEG were used, and this ultimately led to 42 rotations as a result of the PEG execution. Two of them were:
- `GrassSeed-SpringBarley`, and
- `FibreHemp-GrassSeed-SpringBarley`.

Next, the two rotations were fed into the PTG, which used data obtained from the database and specification of management practices on the production orientation, and ultimately generated 252 alternative agricultural activities. Every agricultural activity contains a set of crops (in a rotation), each of which is associated with a year in the rotation and with a set of management events, which is all the information required by APES. The grass-seed crop in `AgriculturalActivity1` (that corresponds to the rotation `GrassSeed-SpringBarley`) has 5 different events, e.g. one sowing event, one harvest event, three nutrient events, and no irrigation events. The execution of APES associated yields and environmental effects to each agricultural activity, that ultimately was directed to the TCG.

TCG execution ultimately resulted in 720 production coefficients that were finally forwarded to the FSSIM to select the optimal set of production coefficients given the farmer objectives. For the creation of production coefficients again the data was retrieved from the database via the KM, as in the TCG information on variable costs and labour requirements were attached to each production coefficient. For example, a grass seed crop in production coefficient `ProdCoeff-1` with rotation `GrassSeed-SpringBarley` has associated a labour requirement of 20 hours per hectare per year and variable costs of 450 euros per hectare per year.

VI. DISCUSSION

Through the case of the AM it was explained how semantic modelling, ontologies and the knowledge management practices can be used in modelling agricultural systems. The ontology structure of AM helped to capture the complexity of the AM by making explicit the knowledge that the agricultural scientist holds. By the use of ontologies, the agricultural scientist is *forced* to define concepts he/she commonly refers to by specifying their properties (data-type properties) and relationships to other concepts (object properties) in a detailed formalisation. Also, the concepts in the ontology could subsequently be made available for modelling by allowing the agricultural scientist to write the algorithms. As farming systems approaches require interdisciplinary studies (Kropff et al., 2001) and therefore require the use of several different models and techniques, the different techniques and models need to be able to communicate with meaningful objects. In the AM the developed ontology played a vital role in clarifying exchanged information between AM and its external peers (i.e. APES, FSSIM, and the databases), as well as among the components of AM (PEG, PTG and TCG). This allowed the AM to operate on two different scales, both in space in time. Firstly, AM is able to exchange meaningful objects with a point scale model, operating on a daily basis, as APES is a dynamic crop simulation model. Secondly, it is able to exchange meaningful with a farm scale model, using annual data, as FSSIM is a static, mathematical programming farm model. Using ontologies in farming systems research requires a close co-operation between disciplines, in this case agronomy, agricultural economy and information technology. A close co-operation can be achieved by frequent iterations and discussion on the concepts used in both disciplines. Using ontologies implies an additional layer to the modelling exercise, which strengthens the other layers (databases, algorithms and model structure), while at the same time making the modeling exercise more distributed.

In particular, in this paper contributed with a modular architecture and implementation for the Agricultural Management module, by exploiting ontologies and semantic modelling, and with two alternative modes of operation for the Seamless Knowledge Manager. Also, we examined the performance of existing knowledge engineering tools, particularly related to linking ontologies with legacy database sources and generating programming interfaces. Both "online" and "offline" models of the Knowledge Manager have been deployed and tested. Although both effective, D2R mapping language and Object-Relational Mapping have shown pros and cons in the development and performance, each one being suitable for different type of deployment. An important issue revealed with both is the maintenance of the domain ontologies and their alignment with third-party data sources. This part could be significantly improved, by employing semantic alignment methods, as this will semi-automatize the procedure. However it reveals issues for formalization of data source registration and environmental dataset standardization.

Future work will concentrate on expanding the current implementation. We plan to exploit further the developed ontology by expressing both production techniques filters and agronomic rules in a declarative fashion. Due to the nature of the rules, it is very hard, if not impossible, to express them using description logics in OWL-DL. Thus extended frameworks like RuleML and SWRL will be considered for incorporating reasoning capabilities. This will further advance the benefits of using ontologies and semantic modelling in the agricultural modelling domain.

Finally, we consider within the context of the SEAMLESS project to promote further the use of semantic modelling, and the development of ontologies for the agricultural sector in order to maximize the reusability and the extensibility of the systems developed. Parallel efforts are adopting declarative approaches for the APES and FSSIM applications. The adoption of a set of shared ontologies within these modularised applications will lead us to a semantic-aware modelling and simulation framework for the agricultural sector.

ACKNOWLEDGEMENT

This publication has been partially funded through the SEAMLESS integrated project, EU 6th Framework Programme for Research, Technological Development and Demonstration, (contract no. 010036-2). For further reference, the SEAMLESS project website is: http://www.seamless-ip.org

REFERENCES

[1] Athanasiadis, I.N. (2007). Towards a virtual enterprise architecture for the environmental sector. *In Agent and Web Service Technologies in Virtual Enterprises*, 256-266, Information Science Reference.

[2] Athanasiadis, I.N.; Villa, F. & Rizzoli, A.E. (2007a). Enabling knowledge-based software engineering through semantic-object-relational mappings. *In Proc. of the 3rd International Workshop on Semantic Web Enabled Software Engineering, 4th European Semantic Web Conference.*, 16-30, KnowledgeWeb.

[3] Athanasiadis, I.N.; Villa, F. & Rizzoli, A.E. (2007b). Ontologies, JavaBeans and Relational Databases for enabling semantic programming. *Proc. of the 31th IEEE Annual International Computer Software and Applications Conference (COMPSAC)*, vol 2, 341-346, IEEE.

[4] Athanasiadis, I.N., Rizzoli, A.E., Janssen, S. & van Ittersum, M. (2007c). Generation of artificial crop rotation schemes as cyclic permutations. In *Farming Systems Design 2007: Methodologies for Integrated Analysis of Farm Production Systems*, vol 2: 225-226, La Golliardica Pavese.

[5] Athanasiadis, I.N.; Rizzoli, A.E.; Donatelli, M. & Carlini, L. (to appear). Enriching environmental software model interfaces through ontology-based tools. *International Journal of Applied Systemic Studies.*

[6] Bauer, C. & King, G. (2006). *Java Persistence with Hibernate*. Manning Publications Co.

[7] Bizer, C. (2003). D2R MAP - A Database to RDF Mapping Language. *In Proc. of the 12th International World Wide Web Conference (WWW2003)*, Budapest, Hungary.

[8] Brilhante, V. & Robertson, D. (2001). Metadata-supported automated ecological modelling. In Rautenstrauch, C. & Patig, S. (ed.) Environmental Information Systems in Industry and Public Administration. Idea Group Publishing.

[9] Brilhante, V.; Ferreira, A.; Marinho, J. & Pereira, J. (2006). Information Integration through Ontology and Metadata for Sustainability Analysis. *In Proc. of the 3rd biennial meeting of the International Environmental Modelling and Software Society*, Burlington, VT, USA, iEMSs.

[10] Dogliotti, S.; Rossing, W.A.H. & van Ittersum, M.K., (2003). ROTAT, a tool for systematically generating crop rotations. *European Journal of Agronomy* 19, 239-250

[11] Donatelli, M.; van Evert, F.K.; Di Guardo, A; Adam, M. & Kansou, K. (2006). A component to simulate agricultural management. *In Proc. of the 3rd biennial meeting of the International Environmental Modelling and Software Society*, Burlington, VT, USA, iEMSs.

[12] Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition* 5(2):199-220.

[13] Horridge, M.; Knublauch, H.; Rector, A.; Stevens, R. & Wroe, C. (2004). A Practical Guide to Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools. *Technical Report,* The University Of Manchester and Stanford University.

[14] Janssen, S. & van Ittersum, M. (2007). Assessing farm innovations and responses to policies: a review of bio-economic farm models. *Agricultural Systems* 94: 622-636.

[15] McGuinness, D. & van Harmelen, F. (2004). OWL Web Ontology Language Overview. *WWW Consortium Technical report*, www.w3.org/TR/owl-features.

[16] Kropff, M.J.; Bouma, J. & Jones, J.W. (2001). Systems approaches for the design of sustainable agro-ecosystems. *Agricultural Systems* 70(2-3): 369-393.

[17] Rizzoli, A., Donatelli, M., Athanasiadis, I., Villa, F., & Huber, D. (to appear) Semantic links in integrated modelling frameworks. *Mathematics and Computers in Simulation*.

[18] Rizzoli, A.; Li, H.; Athanasiadis, I.N. & Marechal, F. (2007). Semantically Rich Interfaces for Simulation Interoperability. *In Proc. of the 2007 European Simulation Interoperability Workshop*, SISO.

[19] Ten Berge, H.F.M; van Ittersum, M.K.; Rossing, W.A.H.; van de Ven, G.W.J.& Schans, J. (2000) Farming Options for the Netherlands explored by multi-objective modelling. *European Journal of Agronomy* 13: 263-277

[20] Van Ittersum, M.K. and Donatelli, M. (eds.) (2003). Modelling Cropping Systems: Science, Software and Applications. Special Issue of *European Journal of Agronomy* 18 (3-4)

[21] Van Ittersum, M.K., F. Ewert, T. Heckelei, J. Wery, J. Alkan Olsson, E. Andersen, I. Bezplepkina, F. Brouwer, M. Donatelli, G. Flichman, L. Olsson, A. Rizzoli, T. van der Wal, J.E. Wien and J. Wolf, 2007. Integrated assessment of agricultural systems – A component-based framework for the European Union (SEAMLESS). *Agricultural Systems*, in press.

[22] Van Ittersum, M.K. and Rabbinge (1997). Concepts in production ecology for analysis and quantification of agricultural input-output combinations. *Field Crops Research* 52, 197-208.

[23] Villa, F. (2007). A semantic framework and software design to enable the transparent integration, reorganization and discovery of natural systems knowledge. Journal of Intelligent Information Systems. 29, 79-96

[24] Villa, F.; Athanasiadis, I.N. & Rizzoli, A.E. (to appear). Modelling with knowledge: emerging semantic approaches to ecological modelling. *Environmental Modelling & Software*.

[25] Villa, F., Ceroni, M. & Krivov, S. (2007). Intelligent Databases Assist Transparent and Sound Economic Valuation of Ecosystem Services. Environmental Management. 39, 887-899

[26] Warren, P. (2006), Knowledge Management and the Semantic Web: From Scenario to Technology. *IEEE Intelligent Systems* 21(1), 53-59.

[27] Willmott, S.; Dale, J. & Charlton, P. (2002). Agent Communication Semantics for Open Environments, *Technical report* f-in-00066, FIPA.