Multi-Agent systems meta-modeling: an ontology-based approach

Valentina Cordì

Università degli Studi di Genova – DISI

2 August 2006

Over the last few years...

- Some years ago software architectures could be represented as a function of some input through a single thread of control. Currently they are more complex.

Over the last few years...

- Some years ago software architectures could be represented as a function of some input through a single thread of control. Currently they are more complex.

- Software architectures with many dynamically interacting components, each with its own thread of control, and engaging in complex coordination protocols, are typically more complex to correctly and efficiently engineer than traditional architectures.

Over the last few years...

- Some years ago software architectures could be represented as a function of some input through a single thread of control. Currently they are more complex.
- Software architectures with many dynamically interacting components, each with its own thread of control, and engaging in complex coordination protocols, are typically more complex to correctly and efficiently engineer than traditional architectures.
- Agents and multi-agent systems (MASs) have emerged as a powerful technology to face the complexity of a variety of today's ICT scenarios.

Over the last few years...

- the autonomy of application components reflects the intrinsically decentralised nature of modern distributed systems and can be considered as the natural extension of the notions of modularity and encapsulation.

Over the last few years...

- the autonomy of application components reflects the intrinsically decentralised nature of modern distributed systems and can be considered as the natural extension of the notions of modularity and encapsulation.

- the flexible way in which agents operate and interact is suited to the dynamic and unpredictable scenarios where software is expected to operate.

Over the last few years...

- Several methodologies for the development of MAS have been proposed.
- To support the development and execution of MASs, novel tools and novel software infrastructures are needed. Various tools are being proposed to transform standard MAS specifications (i.e., AUML specifications) into actual agent code.

Over the last few years...

- Several methodologies for the development of MAS have been proposed.
- To support the development and execution of MASs, novel tools and novel software infrastructures are needed. Various tools are being proposed to transform standard MAS specifications (i.e., AUML specifications) into actual agent code.

Cossentino et al.

A unique specific methodology cannot be general enough to be useful for everyone, then some level of personalisation is needed.

A Unified meta-model

The idea of taking advantage of reusable "components" (or "fragments") of existing methodologies, integrating them according to a common meta-model, is currently gaining a wide consensus.

A Unified meta-model

The idea of taking advantage of reusable "components" (or "fragments") of existing methodologies, integrating them according to a common meta-model, is currently gaining a wide consensus.

Methodology Technical Committee of the Foundation for Intelligent Physical Agents (FIPA)

"In order to reuse contributions coming from existing methodologies we will adopt the method engineering as the referring paradigm. In this context the development methodology is constructed by the developer assembling pieces of the process (method fragments) from a method base. In this way he/she could obtain the best process for his/her specific need/problem. We will take method fragments (the composing elements of the development methodology) from several existing methodologies."

A Unified meta-model

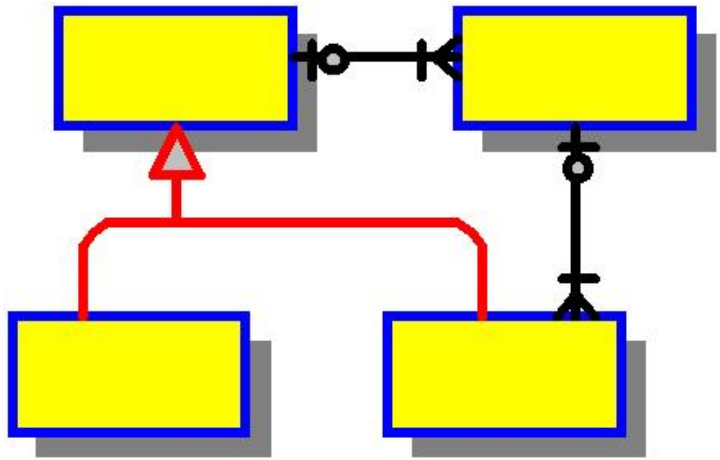
The idea proposed by Cossentino et al.,

Design a unified MAS meta-model, obtained by merging the most interesting aspects of each meta-model.
Then the designers have to select the elements that compose the MAS meta-model they will build.

Our research work is aimed at making a clear and principled analysis of the existing MAS meta-models, and making the analysis result easily accessible by designers.

Meta-models

1. AOSE meta-models
    1. ADELFE Meta-model
    2. PASSI Meta-model
    3. Gaia Meta-model
    4. A unified Meta-model
    5. ROADMAP Meta-model
    6. RICA Meta-model
    7. A generic Meta-model



ADELFE Meta-model (2002)

C. Bernon, M. P. Gleizes, S. Peyruqueou, and G. Picard. *ADELFE, a Methodology for Adaptive Multi- Agent Systems Engineering.*

Based on adaptive MAS and therefore a great effort is done in order to study all the situations that could enable or inhibit the cooperation among agents.

ADELFE Meta-model (2002)

C. Bernon, M. P. Gleizes, S. Peyruqueou, and G. Picard. *ADELFE, a Methodology for Adaptive Multi- Agent Systems Engineering.*

Based on adaptive MAS and therefore a great effort is done in order to study all the situations that could enable or inhibit the cooperation among agents.

The cognitive and behavioural representations of the agent are given in terms of its attitudes, skills and characteristics.

ADELFE Meta-model (2002)

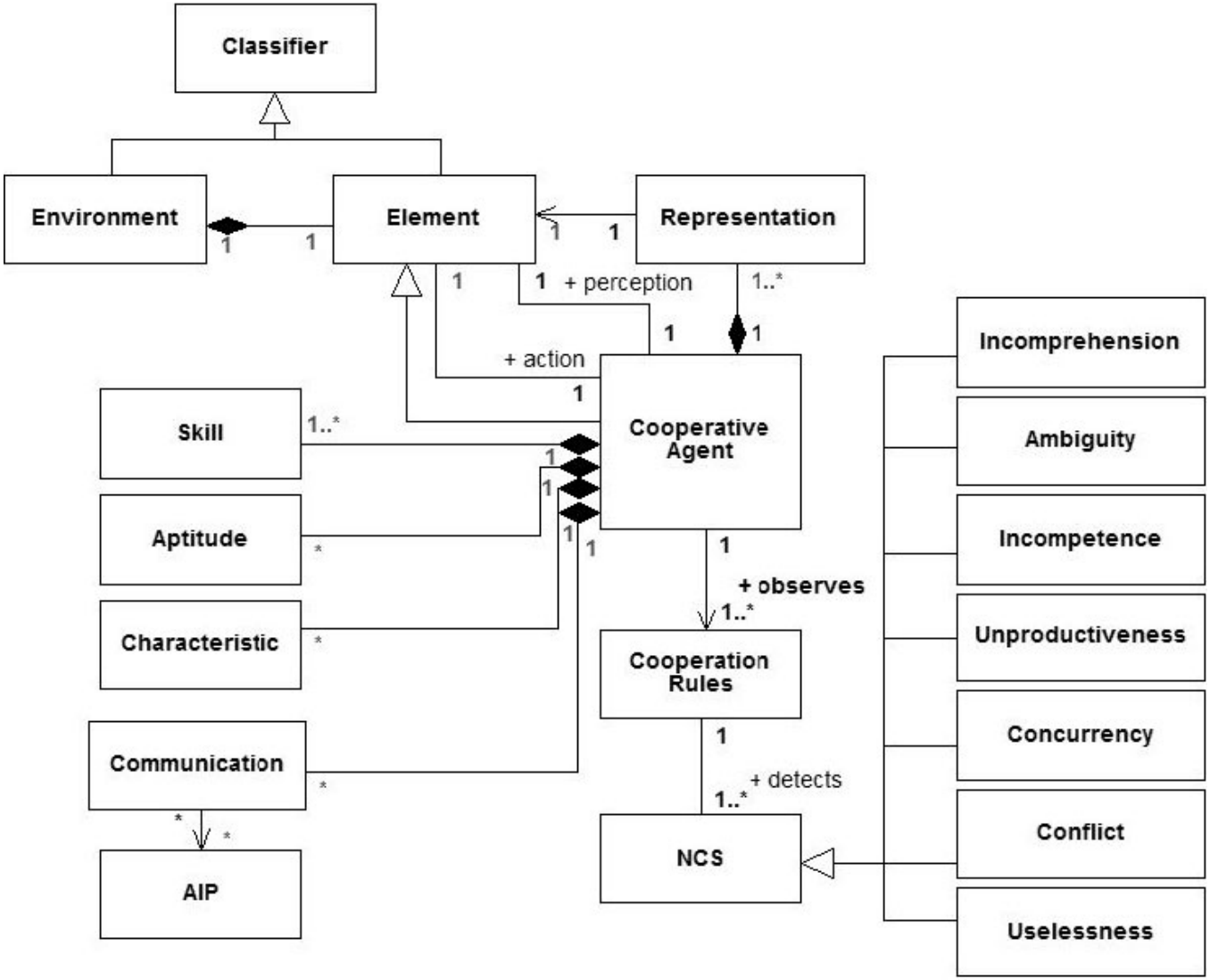
C. Bernon, M. P. Gleizes, S. Peyruqueou, and G. Picard. *ADELFE, a Methodology for Adaptive Multi- Agent Systems Engineering.*

Based on adaptive MAS and therefore a great effort is done in order to study all the situations that could enable or inhibit the cooperation among agents.

The cognitive and behavioural representations of the agent are given in terms of its attitudes, skills and characteristics.

Agents interact either via direct communications or via the environment.

The MAS Meta-Model adopted in ADELFE



PASSI Meta-model (2002)

M. Cossentino. *Different Perspectives in Designing Multi-Agent System*.

Conciliates classical software engineering concepts like problem and solution domain with the potentiality of the agent-based approach.

PASSI Meta-model (2002)

M. Cossentino. *Different Perspectives in Designing Multi-Agent System*.

Conciliates classical software engineering concepts like problem and solution domain with the potentiality of the agent-based approach.

The convergence between agents and traditional issues of software engineering is obtained by introducing a new abstraction layer (agency domain)

PASSI Meta-model (2002)

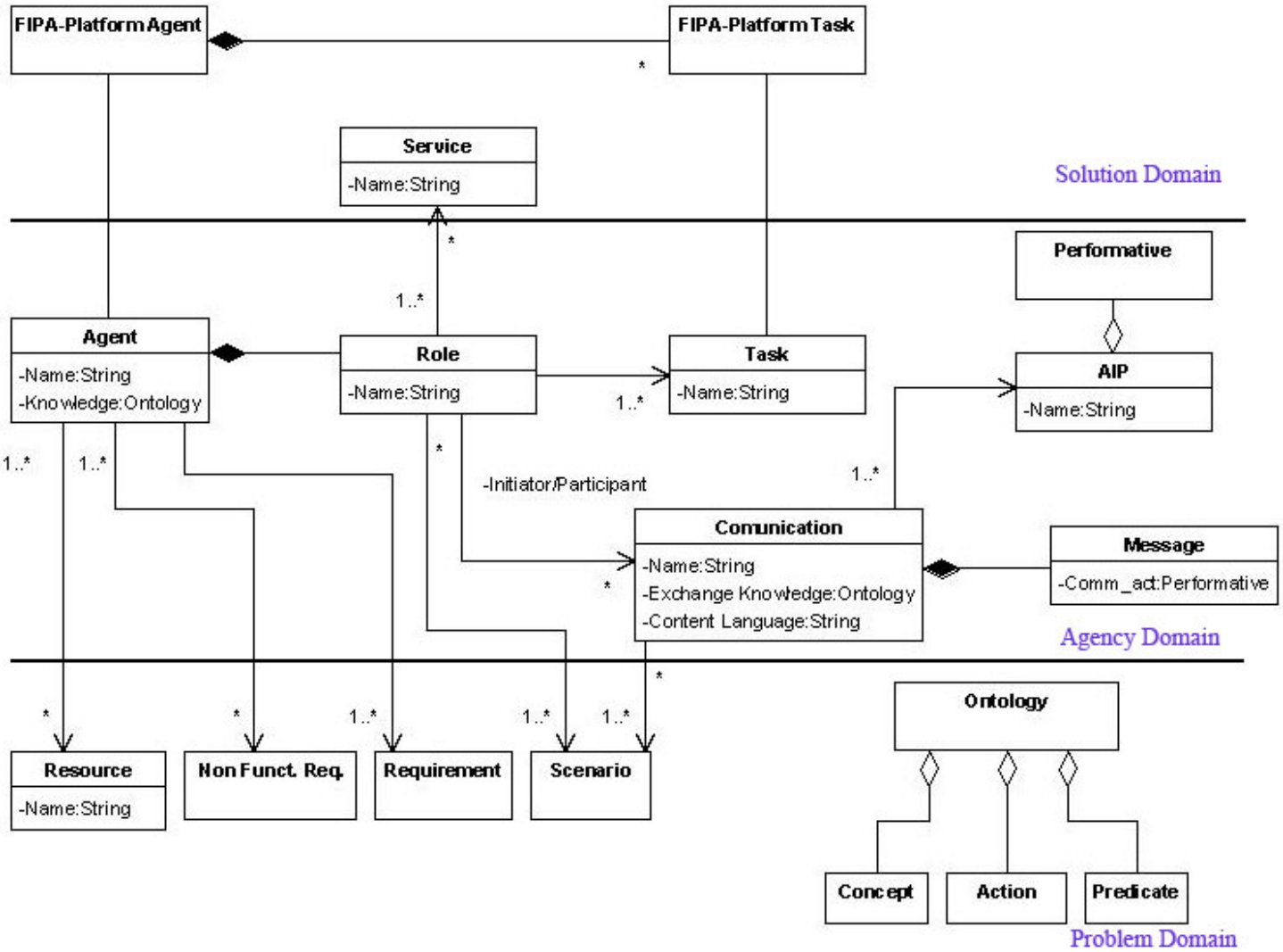
M. Cossentino. *Different Perspectives in Designing Multi-Agent System*.

Conciliates classical software engineering concepts like problem and solution domain with the potentiality of the agent-based approach.

The convergence between agents and traditional issues of software engineering is obtained by introducing a new abstraction layer (agency domain)

Communications and implementation are FIPA-based.

The MAS Meta-Model adopted in PASSI



GAIA Meta-model (2003)

F. Zambonelli, N. R. Jennings, and M.Wooldridge. D*eveloping multiagent systems: the Gaia Methodology*.

Devoted to represent a MAS system as a social organisation.

GAIA Meta-model (2003)

F. Zambonelli, N. R. Jennings, and M.Wooldridge. D*eveloping multiagent systems: the Gaia Methodology*.

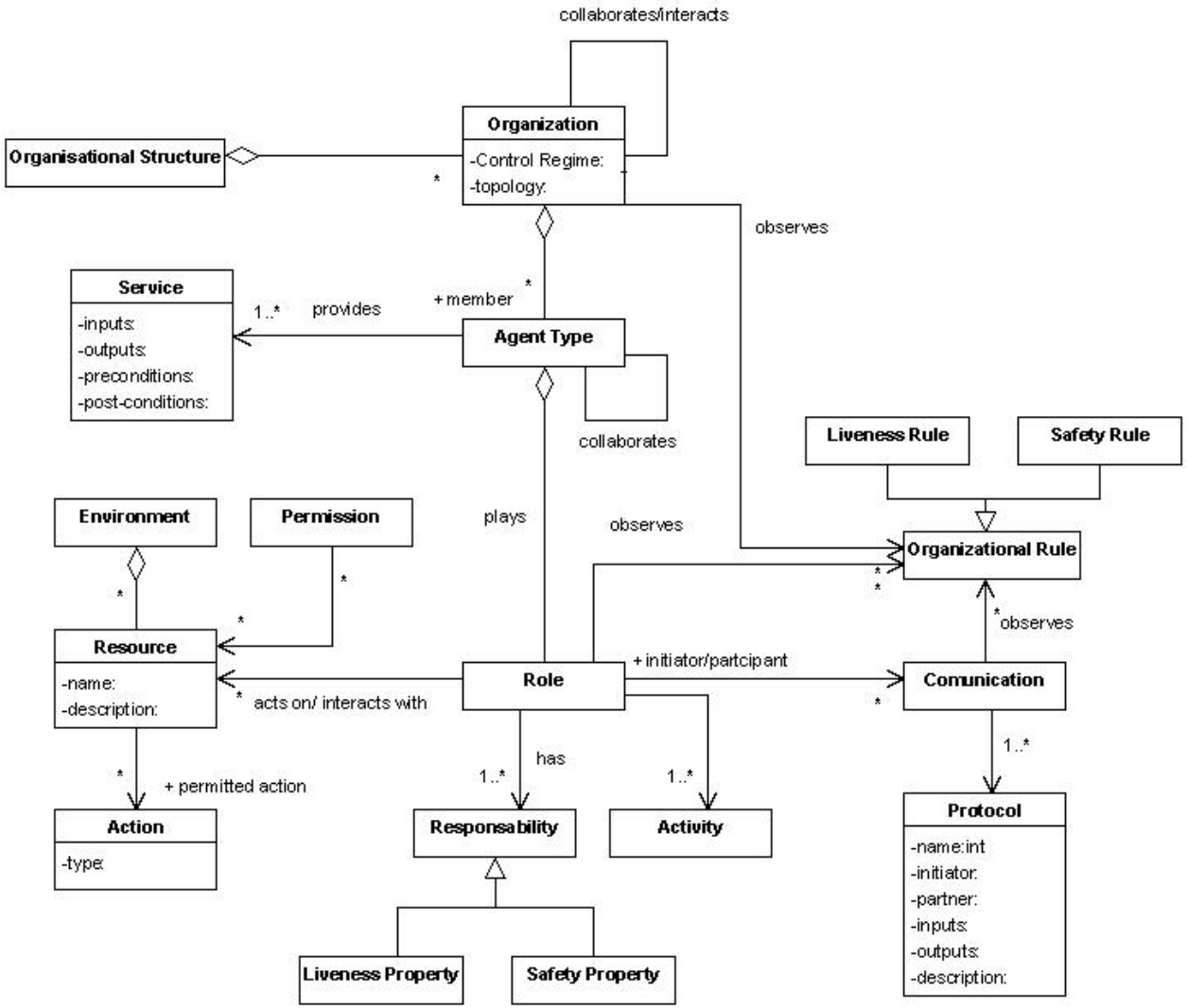
Devoted to represent a MAS system as a social organisation.

An agent is an entity that plays one or more roles; a role is a specific behaviour defined in term of:

- responsibilities
  - safety
  - liveness
- permissions
- activities
- interactions with other roles

An agent plays a role by actualising the behaviour in term of service to be activated and de-activated in dependence of specific pre- post-conditions.

The MAS Meta-Model adopted in GAIA



A Unified Meta-model (2004)

C. Bernon, M. Cossentino, M. P. Gleizes, P. Turci, and F. Zambonelli. *A Study of Some Multi-agent Meta-models*.

A generic agent is composed of Gaia-like roles complemented by some PASSI features (tasks and a FIPA-compliant communication structure). This generic agent has two choices: belonging to an organisation (like in GAIA)or following cooperation rules (like in ADELFE).

A Unified Meta-model (2004)

C. Bernon, M. Cossentino, M. P. Gleizes, P. Turci, and F. Zambonelli. *A Study of Some Multi-agent Meta-models*.

A generic agent is composed of Gaia-like roles complemented by some PASSI features (tasks and a FIPA-compliant communication structure). This generic agent has two choices: belonging to an organisation (like in GAIA)or following cooperation rules (like in ADELFE).

Agent are implemented (at code level) following the PASSI methodology.

A Unified Meta-model (2004)

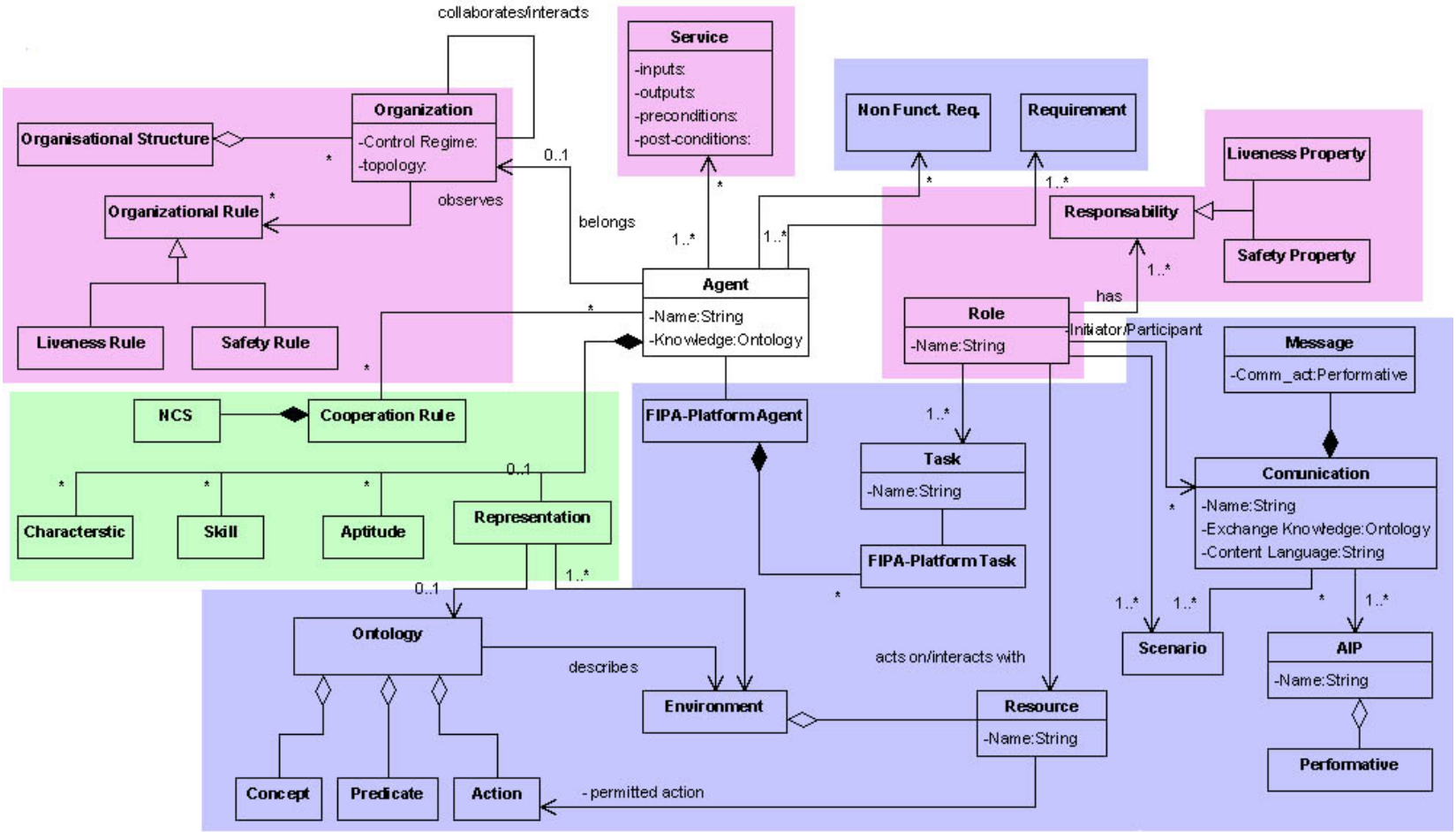
C. Bernon, M. Cossentino, M. P. Gleizes, P. Turci, and F. Zambonelli. *A Study of Some Multi-agent Meta-models*.

A generic agent is composed of Gaia-like roles complemented by some PASSI features (tasks and a FIPA-compliant communication structure). This generic agent has two choices: belonging to an organisation (like in GAIA)or following cooperation rules (like in ADELFE).

Agent are implemented (at code level) following the PASSI methodology.

The proposed meta-model is also characterised by the possibility of identifying in it the three domains (problem, agency, solution) discussed in the PASSI approach.

The Unified MAS Meta-Model



ROADMAP Meta-model (2003)

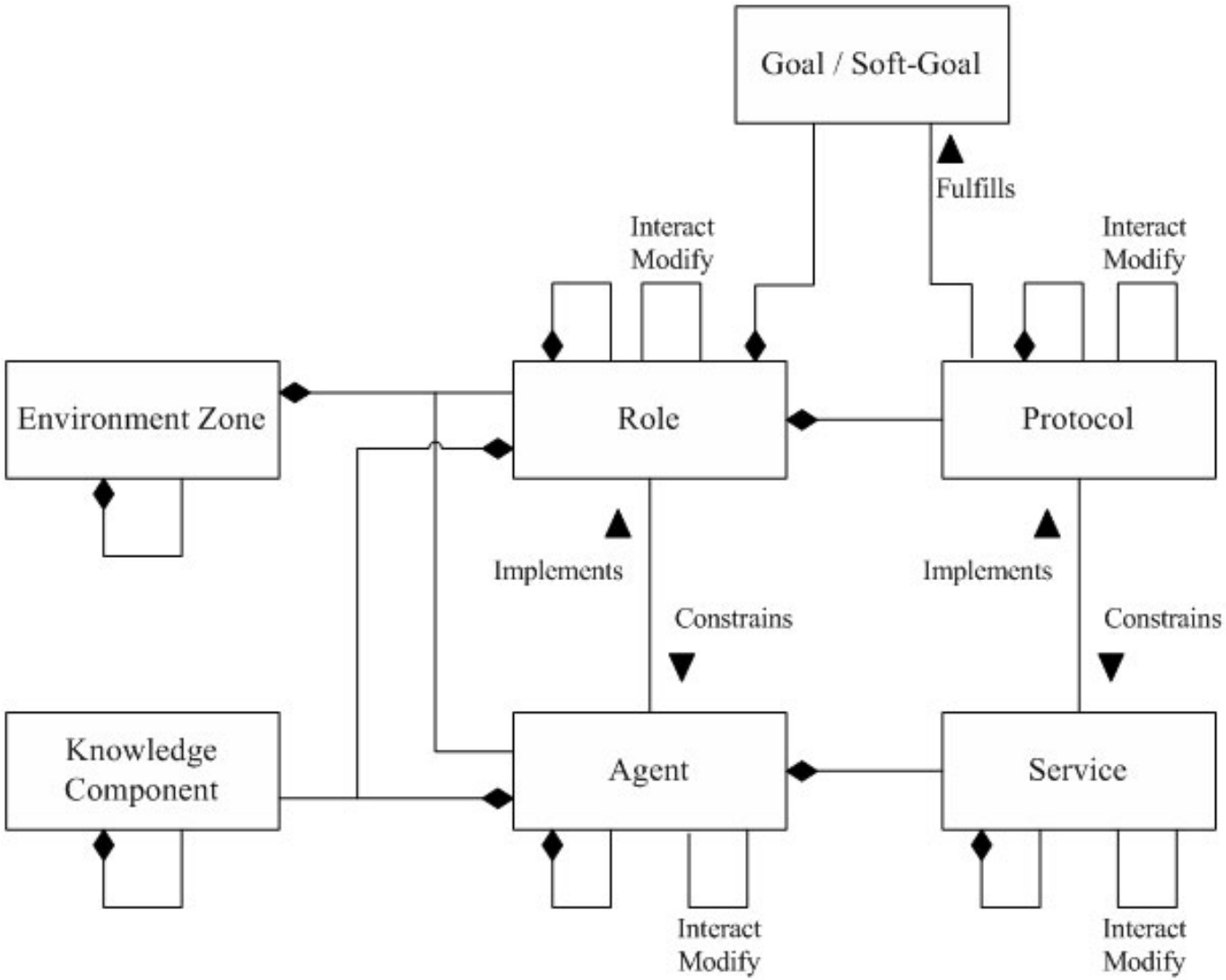
T. Juan, and L. Sterling. *A meta-model for intelligent adaptive multi-agent systems in open environments*.

The ROADMAP meta-model is designed to describe intelligent adaptive systems in open environments, using agent concepts such as roles.

The meta-model is special in the sense that it describes runtime systems and all constructs shown in the meta-model have concrete physical runtime manifestations.

The role hierarchy represents a high-level abstract specification of requirements, capturing: organisational structures, regulations, processes, goals, responsibilities and various permissions for the agents to function in the system.
The agent hierarchy provides concrete implementation of system functionalities.

The MAS Meta-Model adopted in ROADMAP



RICA Meta-model (2004)

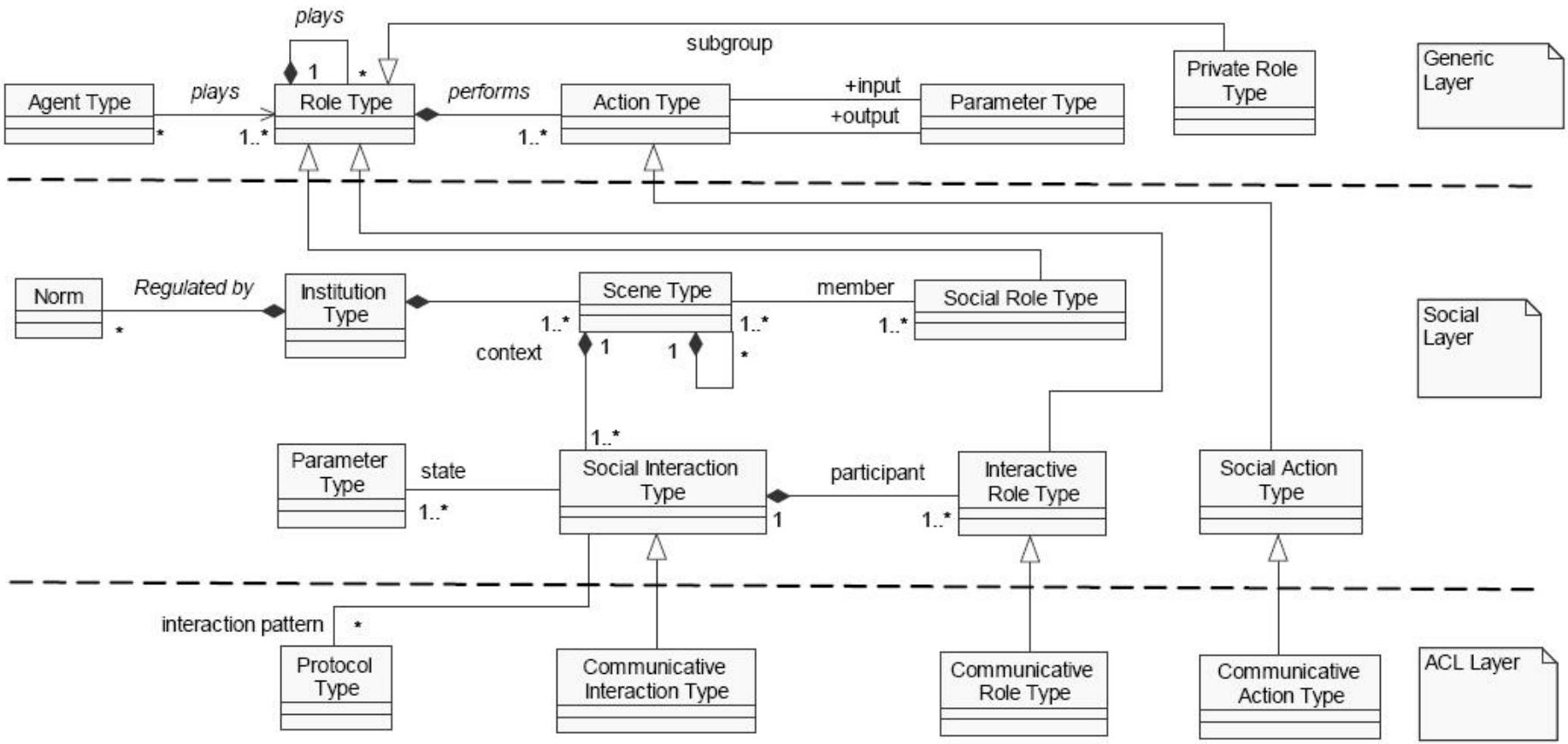
J. M. Serrano, and S. Ossowski. *On the Impact of Agent Communication Languages on the Implementation of Agent Systems*.

The RICA meta-model establishes a systematic link between the specifications of the organizational model and the Agent Communication Language (ACL) of a multiagent application.

The organisational model is specified in terms of entities such as agents, roles and interactions, while the specification of the ACL is based on the definition of comunicative actions and protocols.

The RICA MAS meta-model is organized in three different layers: Generic Layer, Social Layer and ACL Layer.

The MAS Meta-Model adopted in RICA



A generic Meta-model (2005)

G. Beydoun, C. Gonzalez-Perez, G. Low, and B. Henderson-Sellers. *Synthesis of a Generic MAS Metamodel*.

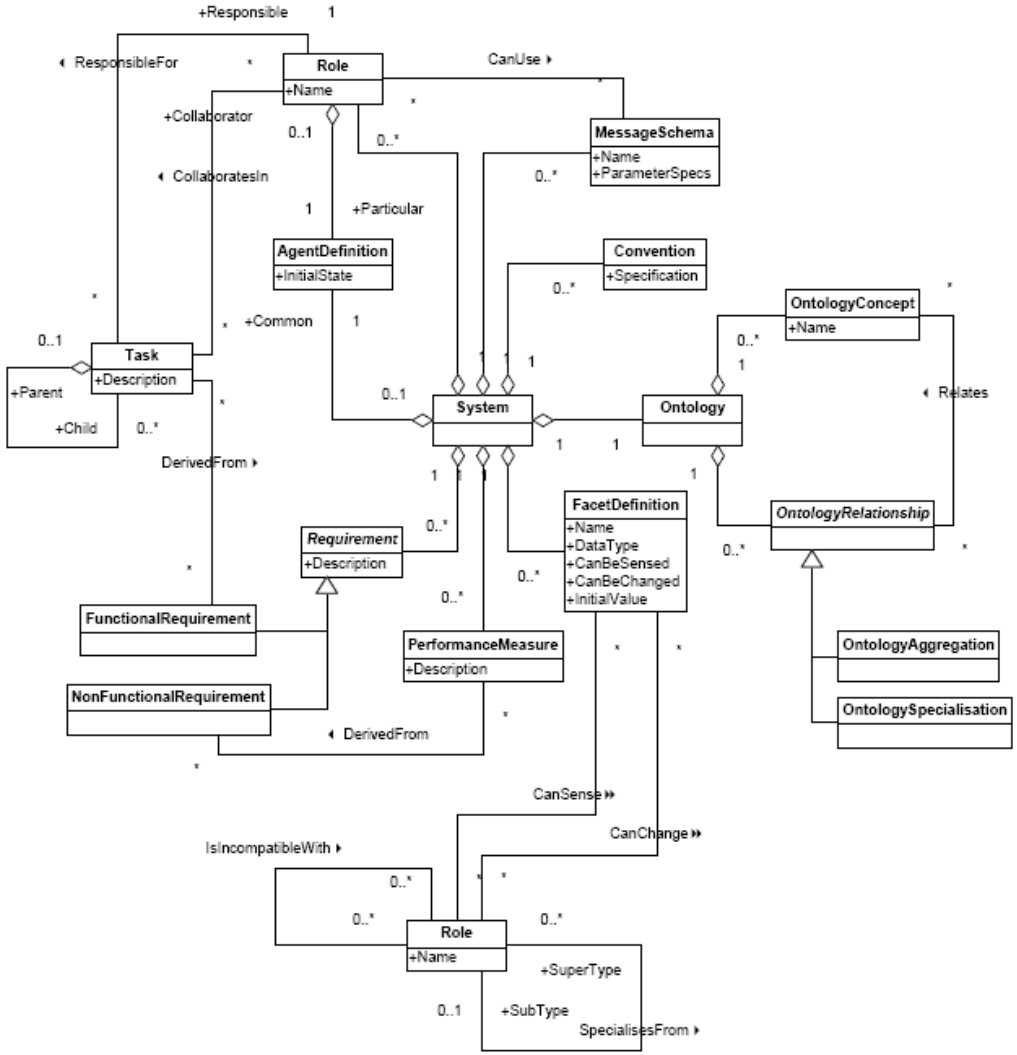
This meta-model is the only one that does not built upon a single AOSE methodology.

The meta-model has two layers: design-time and runtime layers.

Each layer may have two scopes: System related or agent related scope.

It is presented in four different diagrams to clearly group classes into the four different areas of concern: design-time system related classes, runtime system-related (environment) classes, design time agent-internals classes and run-time agent-internals classes.

The generic MAS Meta-Model



Some questions?

- Is it possible to identify a meta-model from which all the meta-models used in the multi-agent community could be derived?

Some questions?

- Is it possible to identify a meta-model from which all the meta-models used in the multi-agent community could be derived?
- What description level has to be reached in the meta-model? For instance, skills and aptitudes in ADELFE are certainly used to implement the role notion of Gaia or PASSI.

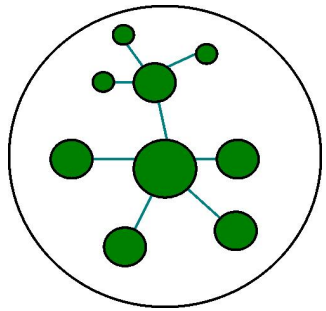
Some questions?

- Is it possible to identify a meta-model from which all the meta-models used in the multi-agent community could be derived?
- What description level has to be reached in the meta-model? For instance, skills and aptitudes in ADELFE are certainly used to implement the role notion of Gaia or PASSI.
- How may a designer choose meta-model elements he is interested in? What kind of tools can we provide him to ease his choices?

Ontology

Definition

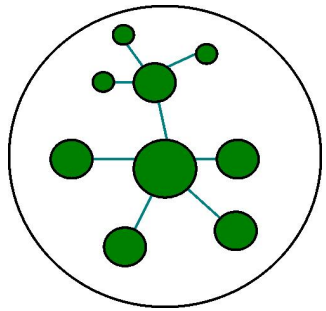
An ontology is an explicit specification of a conceptualisation. Such a conceptualisation consists of objects or entities that are assumed to exist in the domain of interest as well as the relationships that hold between them.



Ontology

Definition

An ontology is an explicit specification of a conceptualisation. Such a conceptualisation consists of objects or entities that are assumed to exist in the domain of interest as well as the relationships that hold between them.

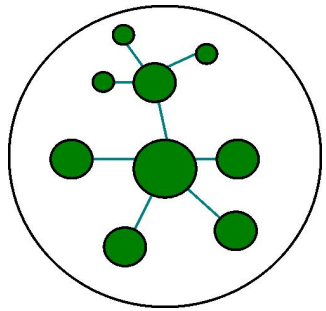


- Ontologies are important because they make explicit every possible relationship between concepts associated to a domain.

Ontology

Definition

An ontology is an explicit specification of a conceptualisation. Such a conceptualisation consists of objects or entities that are assumed to exist in the domain of interest as well as the relationships that hold between them.



- Ontologies are important because they make explicit every possible relationship between concepts associated to a domain.
- The presence of a unified structure is also useful to reduce terminological and conceptual confusion.

Ontology

The use of ontologies, therefore, clarifies the structure of knowledge: given a domain, its ontology constitutes the heart of whichever system representing knowledge on that domain.

Various ontology editors and languages are available to facilitate the developers in defining and representing ontologies, one of the most popular ontology editor is Protégé.

Protégé is a free, open source ontology editor and knowledge-base framework developed by Stanford University. It is based on Java, it is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.

Protégé 2000

Class modeling. Protégé provides a graphical user interface (GUI) that models classes (domain concepts) and their attributes and relationships.

Protégé 2000

Class modeling. Protégé provides a graphical user interface (GUI) that models classes (domain concepts) and their attributes and relationships.

Instance editing. From these classes, Protégé automatically generates interactive forms that enable you or domain experts to enter valid instances.

Protégé 2000

Class modeling. Protégé provides a graphical user interface (GUI) that models classes (domain concepts) and their attributes and relationships.

Instance editing. From these classes, Protégé automatically generates interactive forms that enable you or domain experts to enter valid instances.

Model processing. Protégé has a library of plug-ins that help you define semantics, ask queries, and define logical behavior.

Protégé 2000

Class modeling. Protégé provides a graphical user interface (GUI) that models classes (domain concepts) and their attributes and relationships.

Instance editing. From these classes, Protégé automatically generates interactive forms that enable you or domain experts to enter valid instances.

Model processing. Protégé has a library of plug-ins that help you define semantics, ask queries, and define logical behavior.

Model exchange. The resulting models (classes and instances) can be loaded and saved in various formats, including XML, UML, and RDF (Resource Description Framework). Protégé also provides a very scalable database back end.

An Ontology for the Meta-model of MAS Interaction and Communication

We have developed the communication and interaction portion of the "MAS meta-model ontology", taking six different AOSE methodologies and related meta-models into account.

This ontology, designed following well-established criteria and implemented using the tool Protégé, is aimed at helping the MAS designer in finding the right method fragment to do the right thing, by answering queries such as:

An Ontology for the Meta-model of MAS Interaction and Communication

We have developed the communication and interaction portion of the "MAS meta-model ontology", taking six different AOSE methodologies and related meta-models into account.

This ontology, designed following well-established criteria and implemented using the tool Protégé, is aimed at helping the MAS designer in finding the right method fragment to do the right thing, by answering queries such as:

"What is a communicative act, and which methodologies use it?",

An Ontology for the Meta-model of MAS Interaction and Communication

We have developed the communication and interaction portion of the "MAS meta-model ontology", taking six different AOSE methodologies and related meta-models into account.

This ontology, designed following well-established criteria and implemented using the tool Protégé, is aimed at helping the MAS designer in finding the right method fragment to do the right thing, by answering queries such as:

"What is a communicative act, and which methodologies use it?",
"What is a message according to the methodology XYZ?",

An Ontology for the Meta-model of MAS Interaction and Communication

We have developed the communication and interaction portion of the "MAS meta-model ontology", taking six different AOSE methodologies and related meta-models into account.

This ontology, designed following well-established criteria and implemented using the tool Protégé, is aimed at helping the MAS designer in finding the right method fragment to do the right thing, by answering queries such as:

"What is a communicative act, and which methodologies use it?",
"What is a message according to the methodology XYZ?",
"Which AOSE methodologies take Agent Interaction Protocols into account?".

An Ontology for the Meta-model of MAS Interaction and Communication

Note

It will provide a support to human designers that must build a MAS, by helping them in better understanding which AOSE method fragments are already available, which is their intended meaning, and which AOSE methodologies provide them.

The ontology is not aimed at being used by software agents running inside a MAS.

Overview of the development of the ontology

We have followed the "Ontology Development 101" guide, developed by Noy and McGuinness in 2001.

That guide proposes an iterative approach to ontology development: it starts with a rough first pass at the ontology, then it revises and refines the evolving ontology and fills it in the details.

The methodology is divided into seven steps. In the following, we discuss how we have faced each of them.

Overview of the development of the ontology

1. Determine the domain and scope of the ontology by answering a set of basic questions and by identifying the ontology competency questions.

1. What is the domain that the ontology will cover?
   Answer: The ontology is about MAS meta-model.

Overview of the development of the ontology

1. Determine the domain and scope of the ontology by answering a set of basic questions and by identifying the ontology competency questions.

1. What is the domain that the ontology will cover?
   Answer: The ontology is about MAS meta-model.
2. For what we are going to use the ontology?
   Answer: The ontology will be used to help the MAS designer in finding the right fragment of existing AOSE methodologies to do the right thing.

Overview of the development of the ontology

1. Determine the domain and scope of the ontology by answering a set of basic questions and by identifying the ontology competency questions.

1. What is the domain that the ontology will cover?
   Answer: The ontology is about MAS meta-model.
2. For what we are going to use the ontology?
   Answer: The ontology will be used to help the MAS designer in finding the right fragment of existing AOSE methodologies to do the right thing.
3. Who will use and maintain the ontology?
   Answer: The users of the ontology will be the MAS designers, while the "maintainers" will be the ontology developers (us).

Overview of the development of the ontology

1. Determine the domain and scope of the ontology by answering a set of basic questions and by identifying the ontology competency questions.

1. What is the domain that the ontology will cover?
   Answer: The ontology is about MAS meta-model.
2. For what we are going to use the ontology?
   Answer: The ontology will be used to help the MAS designer in finding the right fragment of existing AOSE methodologies to do the right thing.
3. Who will use and maintain the ontology?
   Answer: The users of the ontology will be the MAS designers, while the "maintainers" will be the ontology developers (us).
4. For what types of questions the information in the ontology should provide answers?
   Answer: The ontology has to provide answers to an exhaustive list of Competency Questions.

Overview of the development of the ontology

Competency Questions

- "What is a message according to the PASSI methodology?"
  A: *A message is a part of communication and it is expressed in an encoding language (e.g. ACL) that is totally transparent to agents. The message content could be expressed in several different content languages (e.g. SL, KIF, RDF, etc.)*
- "Which AOSE methodologies take Agent Interaction Protocols into account?"
  A: *The methodologies that take Agent Interaction Protocols into account are PASSI and ADELFE.*

Overview of the development of the ontology

- "What is a resource according to the PASSI methodology?"
  A: *A Resource is an entity that can be acquired (also using sensors), shared, or produced by agents*
- "What is a communicative act, and which methodologies use it?"
  A: *A communicative act is the act of communication by a specific communicator over a specific time interval and originating from a specific location. Possible communicators include: people, organisations, and computational agents. The methodology that uses communicative acts is PASSI.*

Overview of the development of the ontology

2. Consider re-using existing ontologies.

Currently, existing MAS meta-models are almost all represented using UML Class Diagrams. The recent literature shows that there are strict relationships between UML Class Diagrams and Ontologies.

Thus, by basing our ontology on existing MAS meta-models, although represented in UML and not in a standard ontology language, we have actually re-used and integrated existing ontologies.

Overview of the development of the ontology

3. Enumerate the terms that the developer wants either to make statements about or to explain to a user.

A part of these terms derives directly from the MAS meta-models, while the other part derives from the need of describing and understanding the MAS meta-model.

In particular the last part contains terms as: Meta Description, Bibliographical reference, Authors of the meta-model, Methodology, Description, etc.

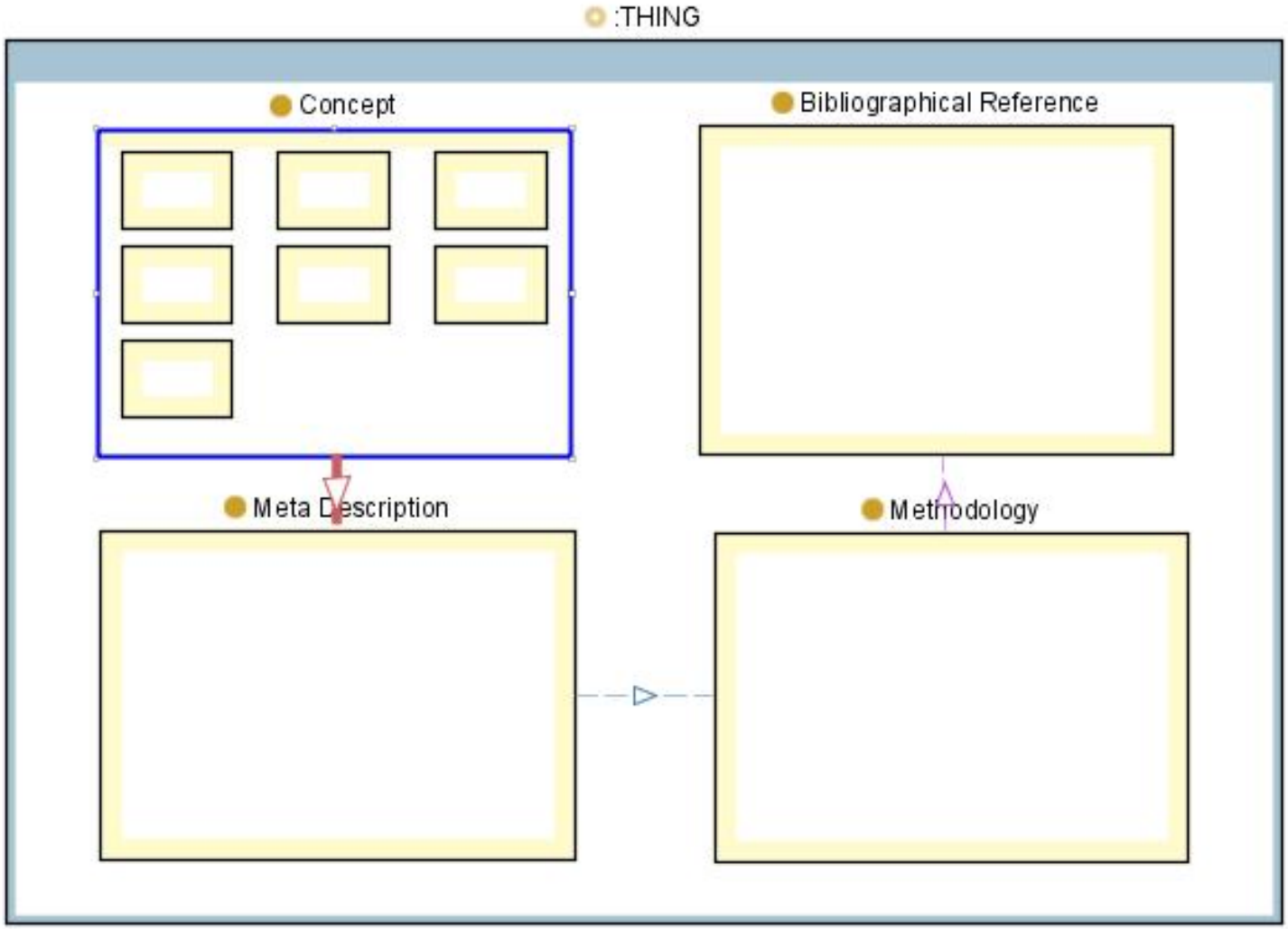
Class hierarchy

4. Define the classes and the class hierarchy by following one of the several existing approaches (top-down, bottom-up, middle-out).

Class hierarchy

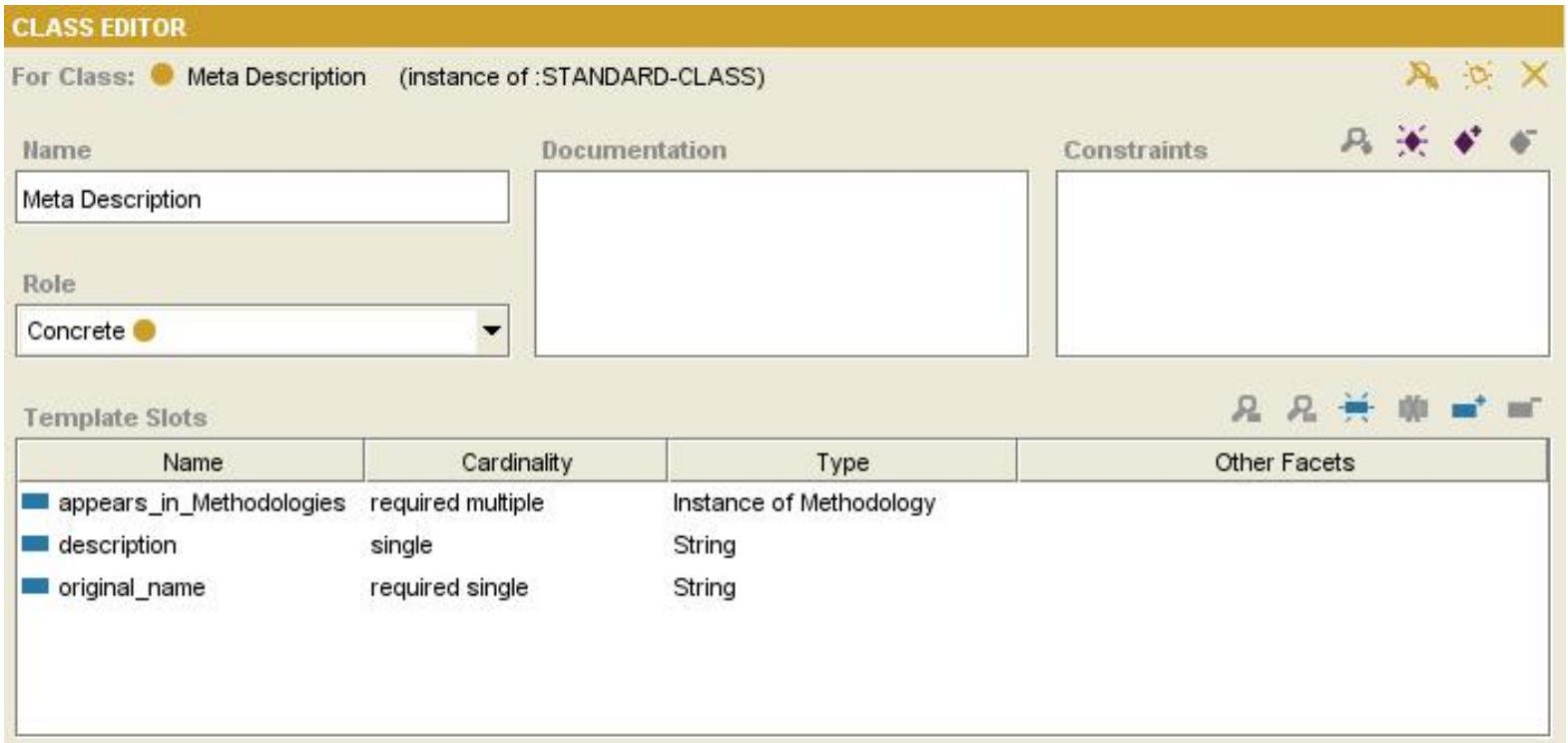
4. Define the classes and the class hierarchy by following one of the several existing approaches (top-down, bottom-up, middle-out).

Class hierarchy



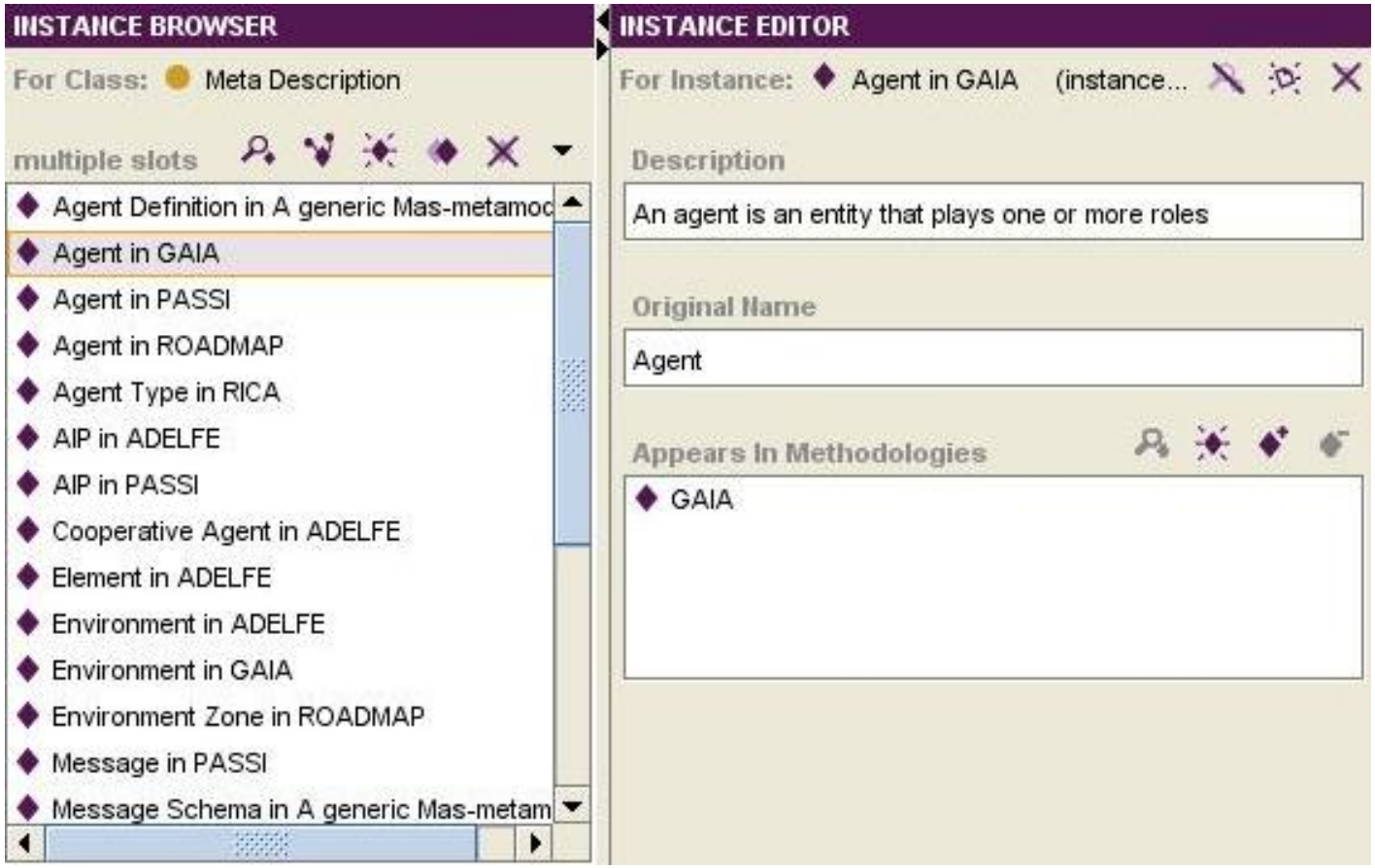
Properties of class Meta Description

5. Define the properties of classes by means of slots, namely, the class attributes.
6. Define the features ("facets") of the slots, such as the value type, the allowed values, the cardinality, etc.

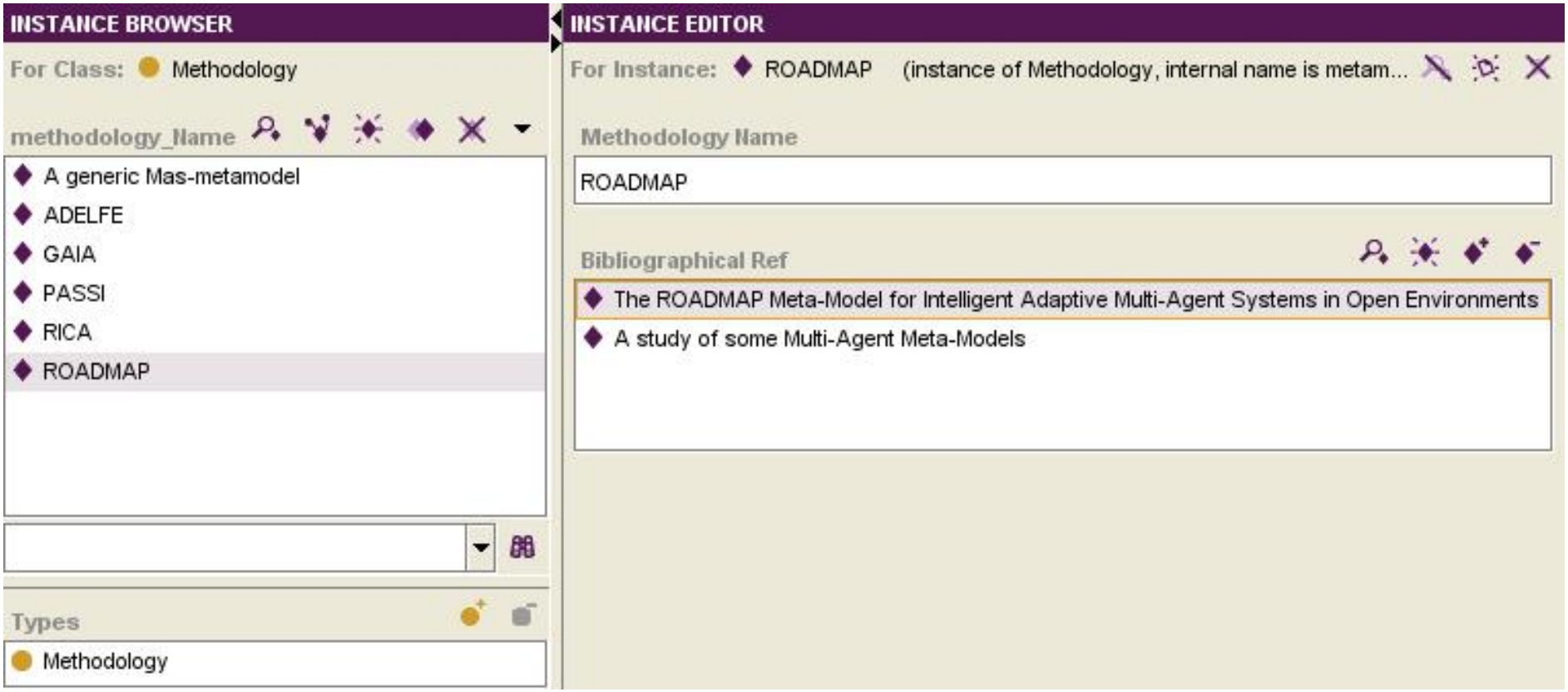


Instances of Meta Description

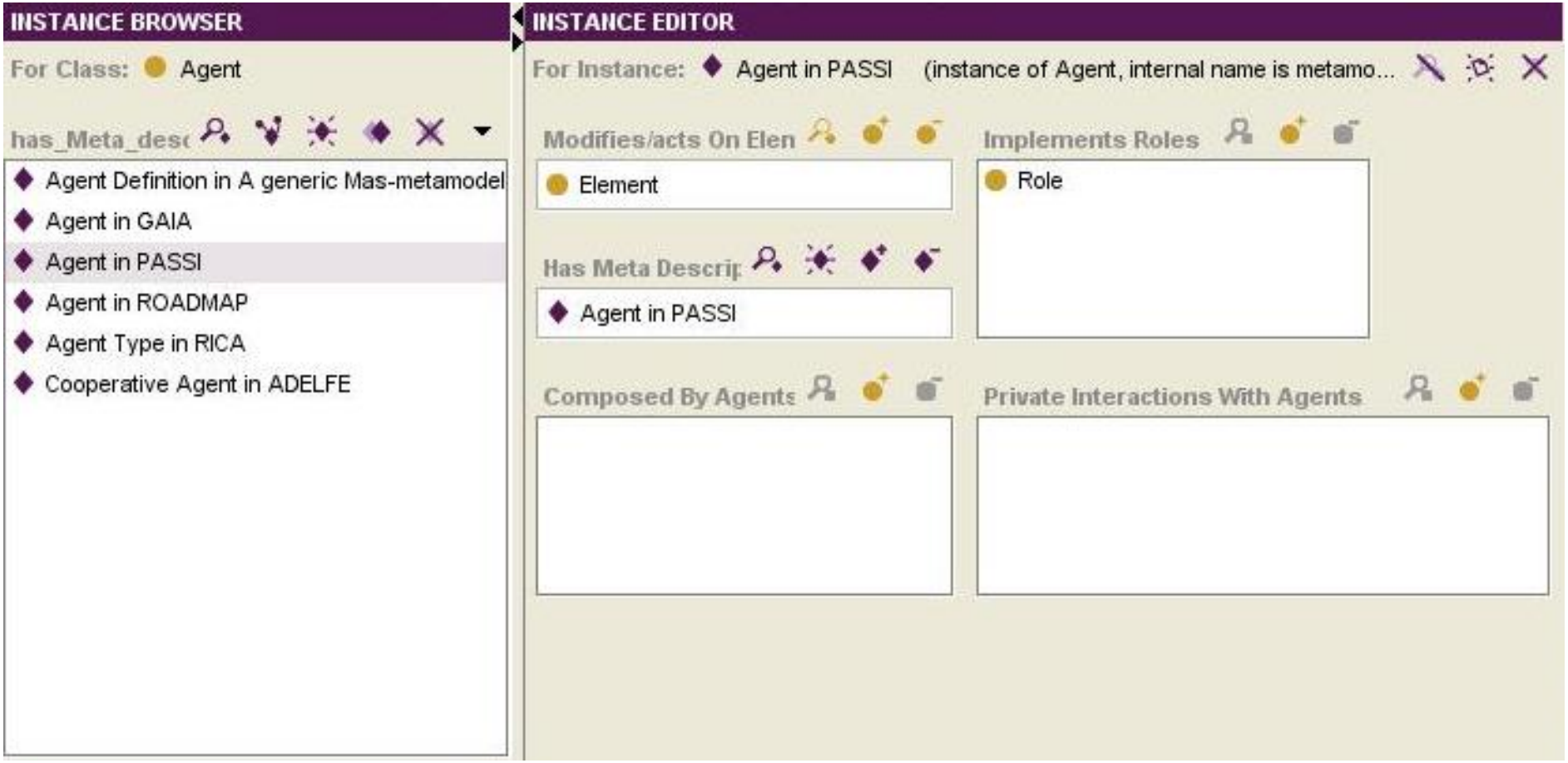
7. Create instances of classes in the hierarchy.



Instances of Methodologies



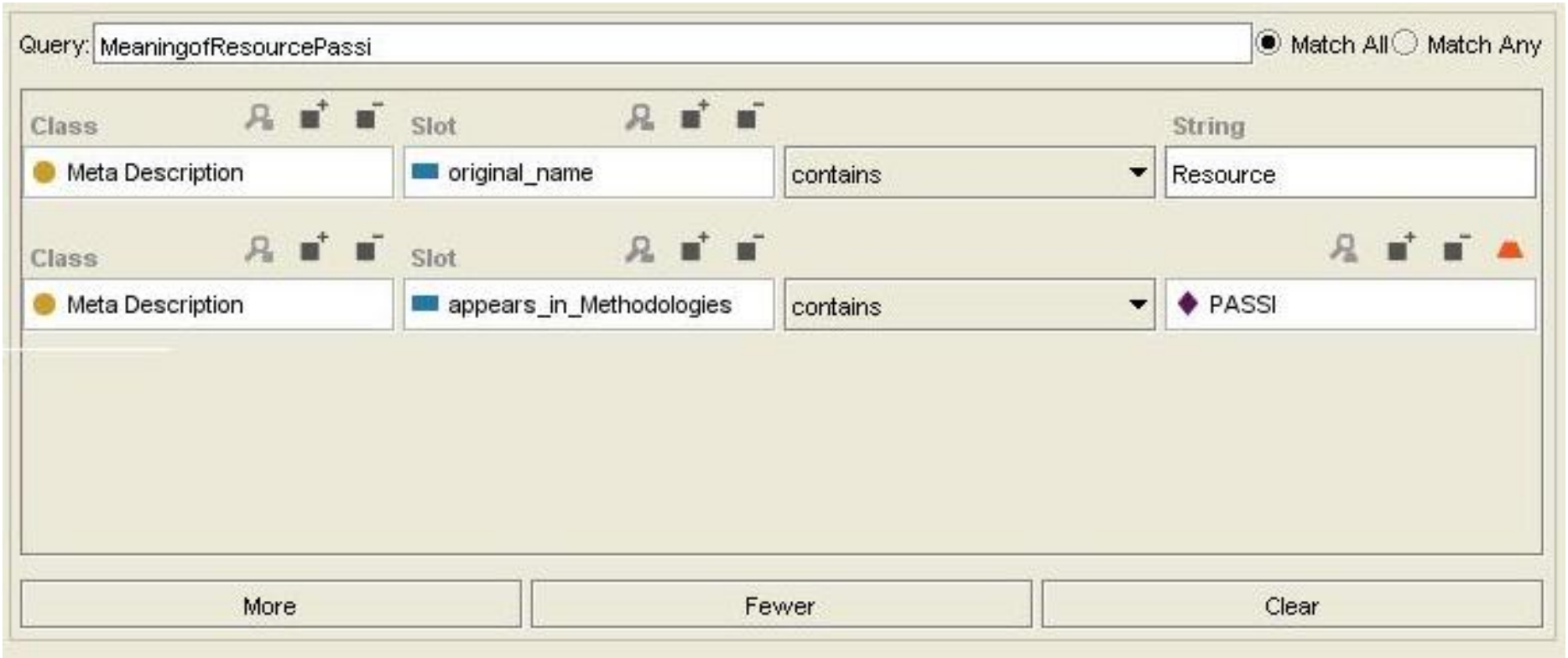
Instances of Agent



Querying the ontology

Final Step: Querying the ontology.

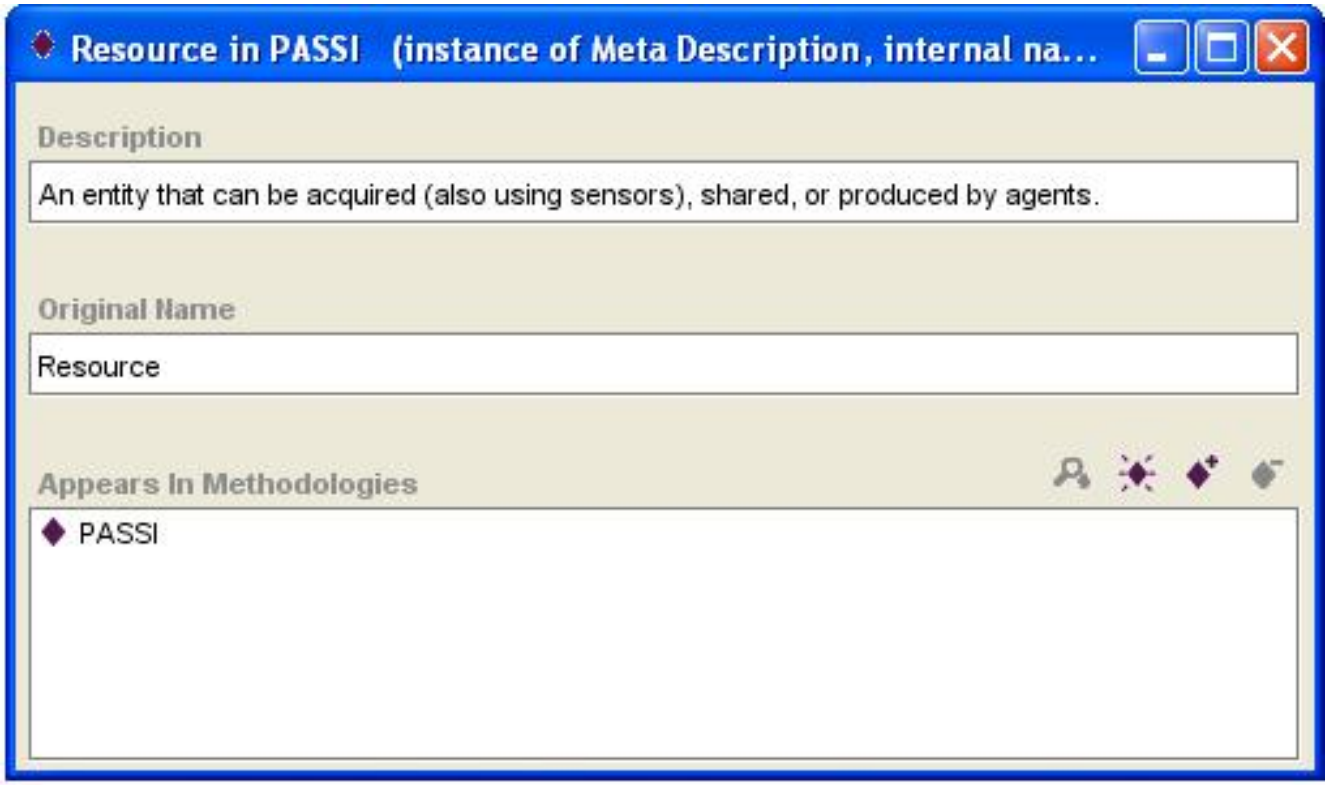
"What is a resource according to the PASSI methodology?"



Querying the ontology

Final Step: Querying the ontology.

"What is a resource according to the PASSI methodology?"



Related work

The "unified meta-model" - Cossentino et al.

They developed a quite a huge model including all the relevant features of the three original meta-models (GAIA, PASSI, and ADELFE).

The "unified meta-model" provides the means neither for understanding from which meta-model each fragment derives, nor for querying the model.

Our ontology adds three more meta-models to those considered by the "unified meta-model", namely the "generic MAS meta-model", RICA, and ROADMAP, and associates meta-data to them that can be analysed and queried,

Related work

INGENIAS- Jorge Gómez Sanz et al.

It consists of five MAS-related models, Tasks and Goals, Interaction, Agent, Environment, and Organisation, plus the models for AUML interaction and activity diagrams, and for UML use case diagrams.

All the models are available on the Web, and can be easily browsed through a graphical user interface developed using SVG.

Related work

INGENIAS- Jorge Gómez Sanz et al.

The interaction model of Ingenias is much more detailed than the ones of the six metamodels previously cited, and for this reason we plan to integrate it in our ontology in the very near future.

With respect to Ingenias, that refers to a unique AOSE methodology, our meta-model, although currently simpler, copes with more methodologies, providing thus a "meta-meta-model".

Conclusion

The purpose of our ontology is to help the MAS designer in untangling the choice of which method fragment and which AOSE methodology are more suitable for modeling a given application component, and in understanding if and when similar fragments have a similar meaning in different methodologies.

Our ontology can be downloaded from `ftp://ftp.disi.unige.it/person/CordiV/Ontologia/OntoMetaModel.owl`.

We are working at equipping it with a user-friendly interface for better navigating and querying it.