Tomás Rogeiro Brochado de Miranda

Bachelor in Computer Science

Software Language Engineering: Interaction and Usability Modeling of Language Editors

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática

Orientador: Vasco Miguel Moreira do Amaral, Assistant Professor, Universidade NOVA de Lisboa

Co-orientador: Miguel Carlos Pacheco Afonso Goulão, Assistant Professor, Universidade NOVA de Lisboa

Júri

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September, 2017

Software Language Engineering: Interaction and Usability Modeling of Language Editors

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "unlthesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/unlthesis [2] http://www.di.fct.unl.pt

# Acknowledgements

"The hardest thing is to go to sleep at night, when there are so many urgent things needing to be done. A huge gap exists between what we know is possible with today's machines and what we have so far been able to finish.- Donald Knuth

WOW! This moment has finally arrived. All of the sudden, here I am writing my master thesis. Though it seemed like a walk in the park, this part of my life was not simple and for that I have to thank so many people that for sure won't fit in one page.

To start: I would like to thank my parents: Ana Miranda and António Miguel Miranda. Without them I could not be where I am now. They are, indeed, the best parents in the whole world. Everyone needs parents like this on their lives.

I have to thank the Faculty of Sciences and Technology from the NOVA University of Lisbon for all the courses and life friendships that lead me to the person that I am at the moment.

Thank you to some of the amazing teachers that I met on this Faculty: Professor Christopher Auretta, João Moura Pires, João Lourenço and José Legatheaux. their knowledge, know-how and way of teaching helped me to be where I am now.

Thank you to my advisers: Vasco Amaral, Miguel Goulão and co-advisers Ankica Barisic, Geylani Kardas, Moharram Challenger, Omer Faruk Alaca for their valuable support, help, and knowledge when needed, helping me stay on the right direction on producing this thesis.

A special "highfive" for the best colleagues/friends that the faculty has provided me for life: André Lopes, Carlos Machado, Carlos Rijo, Fábio Rosário, Filipe Sena, Giovanny Rodrigues, Katia Duarte, João Tomás, Marta Lopes, Miguel Ferreira, Miguel Oliveira, Paulo Ferreira Sara Rosa, Rui Alves, Telmo Santos and this is just a short-list!

A special thank you to my tutor Carla Ventura, who taught how to think like a mathematician, which was really important during all of the computer science course.

Last, but not least, Professor José Luís Capelo Martínez for his enthusiasm and friendship.

To all the people that, at some part of their lives, shared ideas with me, THANK YOU!

Abstract

Background: *Domain-Specific Languages (DSLs)* are programming languages created to a specific domain that a user has pre-conceived. *Multi-Agent Systems* (MAS) represent a set of systems interacting within an environment, in which many intelligent agents interact with each other. *Usability* is a property of something that is "capable of being used"and "convenient and practicable for use". *Barišić et al.* introduced a conceptual framework that supports the iterative development process of *DSLs* concerning the usability evaluation. *Semantic Web Enabled Agent Modeling Language* (SEA_ML) is a *DSL* that supports the modeling and generation of action-based systems for *MAS* and the *Semantic Web*. It is defined by 44 visual notations.

Objective: Improve *SEA_ML's* usability using "The "Physics"of Notations" principles to create a new visual notation for *SEA_ML*.

Method: (1) Participants test the current notation and the new notation on four exercises. For each exercise, a SUS questionnaire is presented. Participants should have better results on the exercises with the new notation. (2) Participants select the notations for *SEA_ML*. Participants receive a list with figures including the current and the new notation, alongside a set of descriptions for each of the semantic constructs of *SEA_ML*. Participants should select more icons from the new notation.

Results: With the results gathered from each experience it is not clear that the new visual notations are better than the current notations.

Limitation: The results from the guidelines were not evaluated broadly.

Conclusion: The results for each experiment are not clear that the new notation is better than the current notation.

This thesis is part of a scientific and technological co-operation between *NOVA LINCS research center at Universidade Nova de Lisboa*, Portugal, and *Ege University International Computer Institute*, Turkey. regarding the project *Developing a Framework on Evaluating Domain specific Modeling Languages for Multi-Agent Systems*.

Keywords: Domain-Specific Language, DSL, Multi-Agent Systems, SEA_ML, MAS, Usability, Usability Testing, Learnability ...

Resumo

Background: *Linguagens de Domínio Específico (DSLs)* são linguagens de programação criadas para um domínio específico que é pré-determinado por um utilizador. Sistemas Multi-Agente (*MAS*) representam um conjunto de sistemas que interage num ambiente, ambiente esse que possui múltiplos agentes que interagem entre si. *Barišić et al.* propõe uma framework conceptual que suporta o desenvolvimento iterativo de uma *DSL* contemplando a avaliação da usabilidade de uma *DSL*. *Semantic Web Enabled Agent Modeling Language* (SEA_ML) é uma *DSL* que suporta a modelação de artefactos para MAS e Web Semântica. Apresenta 44 notações diferentes.

Objetivo: Melhorar a usabilidade do *SEA_ML* utilizando os princípios do "The "Physics"of Notations".

Método: (1) Os participantes experimentam a notação atual e a nova em quatro exercícios. Por cada exercício é proposto um questionário SUS. Os participantes devem ter melhores resultados com a nova notação. (2) Os participantes selecionam as notações para a linguagem *SEA_ML*. Os participantes recebem uma lista com as notações atuais e novas juntamente com um conjunto de descrições sobre as várias construções semânticas da *DSL*. Os participantes devem selecionar mais notações visuais da nova notação.

Resultados: Observando os resultados de cada experiência não é possível de se concluir que a nova notação tem melhores resultados do que a notação original.

Limitações: As normas propostas não foram avaliadas com grupos de maior dimensão.

Conclusões: Os resultados para cada experiência não são claros de que a nova notação é melhor do que a original.

Esta tese faz parte uma cooperação científica e tecnológica entre a *NOVA LINCS research center at Universidade Nova de Lisboa*, Portugal, e *Ege University International Computer Institute*, Turquia, no âmbito do projecto *Developing a Framework on Evaluating Domain specific Modeling Languages for Multi-Agent Systems*.

Palavras-chave: Linguagens de Domínio Específico, DSL, Sistemas Multi-Agente, MAS, SEA_ML Usabilidade, Testes de Usabilidade ...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

On this chapter we introduce the work developed on this dissertation. It presents a short description of its context, motivations and main objectives. The key contributions and document structure will also be described.

1.1 Context

Domain-Specific Languages (DSLs) are programming languages created to a specific domain that a user has pre-conceived. A *DSL* should reduce the gap between the end-user and the software developer, allowing a better quality of the product being built and, at the same time, reducing the development time of that product. [Mer+05].

Although a *DSL* focuses essentially on a particular domain that is targeted by the domain expert, the *DSL* users should achieve the goals for which the same *DSL* was destined to during its use. The goals should be adequate to the tool. To reach the expected goals, the tool has to meet some usability requirements. If these usability requirements are not met, the *DSL* may be deprecated or may produce results different from the expected.

Multi-Agent Systems represent a set of systems interacting within an environment, in which many intelligent agents interact with each other. These agents are autonomous entities that have their own characteristics and objectives to accomplish [Woo09].

Usability is a property of something that is *"capable of being used"* and *"convenient and practicable for use"*. Applied to the computer science area, *usability* focuses on the *human-computer* interaction (HCI), with this interaction being promoted with the elegance and clarity of a certain computer program interface. This interface can appear in the form of text or diagrams [Nie03].

Since *DSLs* have an interface, users that are going to work with the tool should find it easy to manage and interact with that interface. With this interface well-defined, it is possible to evaluate the usability of a *DSL* by testing its interface.

Usability plays an important role on the communication/expression of a software, so the users who are going to use the tool should be able to provide input to the development process, in order to better align the *DSL* with their needs and expectations for it.

In this thesis the usability of the *DSL* for *Multi-Agent Systems SEA_ML* will be discussed. *SEA_ML* allows users to model and generate architectural artifacts for *Multi-Agent Systems* especially working on the *Semantic Web* [Cha+14].

Applying the principles of "The "Physics" of Notations" [Moo09], a set of improvements are proposed in order to improve the usability of *SEA_ML*. Two experiments related with the improvements gathered were made with end-users that were proposed by the domain experts.

Similar languages to *SEA_ML* are also introduced and a comparison between each language is made. The *DSLs* that are going to be compared with *SEA_ML* are:

1. **DSML4MAS** — Platform independent language for specifying *MAS* [Hah08];

2. **INGENIAS** — Proposes a modeling language for *MAS* with an agent-oriented methodology [Pav+05];

3. **MAS_ML** — Extension of *UML*, incorporating agent related concepts to enable *MAS* [Gon+15].

This thesis is part of a scientific and technological co-operation between *NOVA LINCS research center at Universidade Nova de Lisboa*, Portugal, and *Ege University International Computer Institute*, Turkey. regarding the project *Developing a Framework on Evaluating Domain specific Modeling Languages for Multi-Agent Systems*.

1.2 Motivation

When dealing with new tools, users tend to relate that new instrument with previous experiences that they've had with tools that they find similar to the new one. If, by some reason, the instrument does not react the way the user was thinking, it may take more time than the expected for him to work with it properly. *Learnability* appears in this context: how easy is it for users to complete a certain set of tasks while using the tool for the first time? [Nie03]

The usability of a *DSL* can be measured through a set of metrics. Metrics are defined previously to a usability experiment and enable the extraction of information from the users that are testing the tool. Common metrics can measure at what speed users are doing the tasks they are prompt to do, what is the success rate of those tasks and the subjective users satisfaction while executing each task.

Since *DSLs* try to ease the process of building a tool for a specific domain, usability should be directly related with this process since it deals with the main basis of communication between the user and the interface (in this case, *SEA_ML*). Including end-users on the process of development/evaluation of a *DSL* allows the developer to interact with real DSL users that have their own desires and expectations for it, allowing the *DSL* to be more precise on what the user desires for it.

1.3 Objectives

This thesis applies the usability methodology proposed by [Bar16] and evaluates the usability of *SEA_ML* using the principles of "The "Physics" of Notations" [Moo09]. Two experiments are defined using the results gathered from "The "Physics" of Notations" study, using end-users defined by the domain experts for these experiments. A set of well defined metrics are defined in order to evaluate each one of the two experiments.

Using the results obtained for each experiment, some proposals are made in order to improve *SEA_ML's* visual notation.

1.4 Key Contributions

On this dissertation *SEA_ML*, a *DSL* for *Multi-Agent Systems* and the *Semantic Web* will be evaluated using [Bar16] usability methodology. The *Usability* of the *SEA_ML* language will be tested. *SEA_ML* will be evaluated using the principles of "The "Physics" of Notations"[Moo09]. Using each principle of "The "Physics" of Notations", some modifications are proposed to help improve the usability of the *SEA_ML*. The proposed modifications are tested in two different experiments using end-users that were defined by the domain experts.

1.5 Problem Statement

SEA_ML is a *DSL* language that models *Multi-Agent Systems* and the *Semantic Web*. It is used for educational purposes. *SEA_ML* has a visual workbench, with a set of 44 different visual notations, divided through 8 different viewpoints.

The *Learnability* of *SEA_ML* was issued by the Domain Experts from *Ege University International Computer Institute* as something that can be improved in order to enhance the *DSL*. As *SEA_ML* has a big set of visual notations (44), we believe there is room for improvement in the visual notation and its correlation to the semantic constructs they are linked to.

We will use the methodology proposed by [Bar16] and "The "Physics" of Notations" principles to improve the *SEA_ML* visual notation. With the results gathered from the application of "The "Physics" of Notation" two experiments are going to be made and based on those results we will understand if the improvements that were proposed enhanced the *SEA_ML* language.

Evaluating and improving the usability of *SEA_ML*, a *DSL* for *Multi-Agent Systems*.

1.6 Document Structure

This documents is organized as follows:

1. **Chapter 2** — Background: includes an overview of what is a *DSL*, a discussion of its advantages and disadvantages and its development cycle. An introduction to *Multi-Agent Systems (MAS)* and usability will also be approached. *SEA_ML* will be presented in detail, along with some *DSLs* for *MAS*.

2. **Chapter 3** — Related Work: a brief overview of other usability evaluations and "The Physics Of Notations";

3. **Chapter 4** — applying "The Physics Of Notations" to *SEA_ML*: Understanding the visual notation that SEA_ML presents to the user and applying each principle of "The Physics Of Notations" in order to understand if there is room for improvement on the language's notation. If there is room to improve the language, propose modifications (a new notation) for the language to be according to each principle of "The Physics Of Notations";

4. **Chapter 5** — Selecting a visual notation for SEA_ML: explanation, application and result discussion of the experiment "Selecting a visual notation for SEA_ML";

5. **Chapter 6** — SEA_ML current notation VS SEA_ML new notation: explanation, application and result discussion of the experiment "SEA_ML current notation VS SEA_ML new notation";

6. **Chapter 7** — Conclusions and future work: a synthesis of the work that was made during this thesis and suggestions for future iterations of the experiments made.

Chapter 2

Background

This chapter introduces the main basic concepts upon which this dissertation work will be performed.

2.1 Domain-Specific Languages

Domain-Specific Languages (DSLs) are abstractions created to match the user's mental model of a certain problem domain. Since a *DSL* is focused on a restricted domain, learning it and avoiding errors while using it is allegedly easier than with a *general-purpose language*.

The main focus of a *DSL* is its domain. A domain may be defined as a set of characteristics that describe a group of problems which a certain application intends to solve. The domain combines concepts, processes, restrictions and rules of a certain set. The domain should be relevant to a *stakeholder* (person, group or organization that has interest or concern on the domain that will be addressed). The domain should be restricted to facilitate the creation of the respective *DSL* [Mer+05].

If a language is not confined to a specific domain, then it may be a *General Purpose Language (GPL)*. A *GPL* can be defined as a language that is designed to be used on several domains, in contrast with a *DSL*.

When building a *DSL*, we need to consider:

1. **Problem Domain** — The concepts and rules of the domain;

2. **Solution Domain** — Usually computational terms.

A *DSL* is defined using an abstract, a concrete syntax and the language semantics. The abstract syntax represents the concepts and their relations to other concepts without any consideration of their meaning. The concrete syntax provides a mapping between meta-elements and their representations. The static semantics provides definitions of certain constraint rules on the abstract syntax that are hard or impossible to express in a standard syntactic formalism of the abstract syntax. The *DSL* is defined on a meta-model, which can be described as a model of models. A model is an abstraction of a system, allowing predictions or inferences to be made [Küh06].

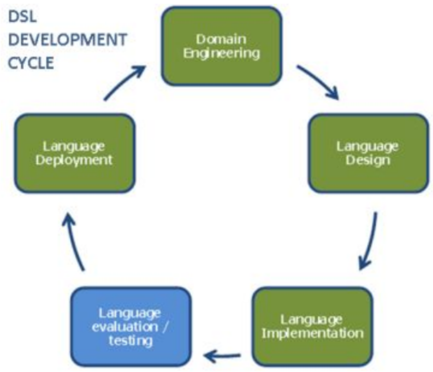


Figure 2.1: DSL Development Cycle. Adapted from [Voe+13]

2.1.1 DSL Development Cycle

The development of a *DSL* can be defined on five important steps, represented in Figure 2.1 (adapted from[Voe+13]):

1. **Domain Engineering** — The activity of collecting, organising and storing past experience in building systems or parts of systems in a particular domain in the form of renewable assets. It also provides an adequate meaning for reusing these assets when building new systems.

2. **Language Design** — A *DSL* can be designed from scratch or it can be based on some other language already developed. The semantic and syntax should be defined during this step.

3. **Language Implementation** — At this point, the *DSL* should already have its domain, semantics and syntax well defined. With this correctly defined, meta-models will be created to implement the solution designed.

4. **Language Evaluation** — This is one of the most important steps of the *DSL* development but usually neglected due to the high costs. Once the *DSL* is finished, the developer should confirm that the language covers the problems meant to be solved and that the solution is easier than the ones already in the market.

5. **Language Deployment** — If every step previously described is correctly completed, then the *DSL* is ready to be delivered and used. Domain Experts/Domain Users

should expect to specify models with the *DSL* created respecting the patterns mentioned in the first two phases of the development of the *DSL*.

2.1.2 Development Paradigms

Since the *DSLs'* main focus is presented in models and meta-models, paradigms are defined to determine the importance of these elements on the construction of a *DSL* [Bra+12]. The essential development paradigms are defined below [Dav09] :

1. **Model-Based Engineering (MBE)** — Paradigm in which a software model is important for the development of the tool, although it is not a key artifact for its development.

2. **Model-Driven Engineering (MDE)** — Goes beyond of the pure development activities and encompasses other model-based tasks of a complete software engineering process (e.g. the model-based evolution the system or the model-driven reverse engineering of a legacy system).

3. **Model-Driven Development (MDD)** — Development paradigm that uses models and transformation (which also have models) as the primary artifact of the development process. Usually, in MDD, the implementation is (semi)automatically generated from the models.

4. **Model-Driven Architecture (MDA)** — A particular vision of *MDD* developed by the *Object Management Group (OMG)*. For *MDA* the modeling of the system if the key artifact of the development of a software tool.

2.1.3 Target Platforms

2.1.3.1 JADE

JAVA Agent DEvelopment Framework (JADE) is a framework implemented in java that simplifies the implementation of *MAS* through a middleware that recognizes the *Foundation for Intelligent Physical Agents (FIPA)* specification, through a set of visual tools that support debugging and deployment phases. *JADE* provides an agent abstraction, used in every *MAS* [Bel+05].

2.1.3.2 JACK

JADEX is a *Belief Desire Intention (BDI)* agent framework composed by *MAS* components like agents, beliefs, plans and goals. *JADEX* is based on the *JADE* platform. As it stands, it is based in the agent beliefs, which consist on the knowledge of an agent [Bus+99].

2.1.4 DSL main characteristics

1. **Domain** — Set of characteristics that describe a group of problems which a certain application intends to solve

2. **Model** — Defines an abstraction of a system, capturing the *DSL* domain.

3. **Meta-Model** — Models of models. Allow the interaction between models.

4. **Syntax** — Set of rules that defines the combination of symbols that are considered to be a correctly structured document or fragment on a certain language. Syntax can be divided in abstract and concrete syntax..

5. **Semantic** — Meaning of various elements on a certain language (program).

2.1.5 DSL Advantages and Disadvantages

Since *DSLs'* are restricted to a domain defined by the domain expert, they are expected to improve productivity. Domain experts should be able to understand, communicate and validate the concept presented by the *DSL* .The *Software Language Engineers (SLE)* will capture and develop the proposed tool. *DSLs'* have a visual editor that enables end-users to interact easily with the tool. After specifying something with the *DSL*, the generation of code and documentation will take place automatically, hiding the complexity of the *DSL*. All of these advantages contribute to leverage the quality and to deliver a productive tool that is automatically generated.

On the other hand *DSLs'* have flaws: the restriction of the domain leads to a limited applicability. This limited applicability is noticed by the lack of web communities for that *DSL*, leading to more difficulties on finding code examples for the *DSL*. With no *DSL standard* defined, this problem is even more serious since there can exist more than one *DSL* for the exact same domain and application on the market, producing unnecessary costs (several *DSLs'* with the same scope are being developed) and restricting the evolution of these *DSLs'*. The maintenance of a *DSL* should also be taken into consideration. Due to its restricted applicability, it is harder to find users that can maintain the tool and, for that reason, it can produce higher costs. [Mer+05].

2.2 Multi-Agent Systems

Multi-Agent Systems (from now on *MAS*) represent a set of systems interacting within an environment, in which many intelligent agents interact with each other. These agents are considered to be autonomous entities, containing certain specifications that can be used to solve their problems or common problems on these systems, allowing them to achieve certain goals [Woo09].

The development of a *MAS* is not trivial due to the vast type of agents available and the vast type of interactions that can exist between agents.

2.2.1 Main Characteristics

1. **Agents** — An agent represents a software system situated within a certain environment in order to meet its design objectives.

   Agents can be passive or active. If an agent is passive, it means that it has no goals to achieve. In contrast, an active agent has goals to achieve. Important agent properties include [Woo09]:

   a) **Autonomy** — Each agent is self-aware, carrying a set of operations with independence of others that may occur at the same time, having some knowledge or representation of what the system needs and wants from it.

   b) **Pro-activeness** — Agents are able to demonstrate goal-directed behaviour by taking the initiative in order to satisfy the goals they are designed to achieve.

   c) **Social-Ability** — Agents are able to interact with other agents in order to satisfy their goals.

2. **Environment** — Represents the state of the system (and its characteristics) where the agents are interacting.

3. **Interaction** — Each agent communicates within an environment with other agents. This communication allows agents to solve their own goals or to help other agents solve their goals.

4. **Organisation** — Sets of norms and rules are defined for the *MAS* to be coordinated.

2.2.2 MAS Metamodels

Several *MAS* metamodels have been introduced to the community that try to model certain aspects of what a *MAS* should be. Some are described below due to its relevance for the *MAS DSLs* that are going to be evaluated.

2.2.2.1 AALAADIN

AALAADIN is a metamodel (Figure 2.2) for *MAS* with three essential concepts: agents, groups and roles [FG98].

For *AALAADIN*, agents are defined by their role inside a certain organisation and the capabilities that they can offer for that group. Agents are atomic, and can only be part of a group at a time.

2.2.2.2 ADELFE

ADELFE can be defined as a methodology to develop an adaptive *MAS (AMAS)*, concentrating on a cooperative behaviour (Figure 2.3). Agents have certain attributes, aptitudes

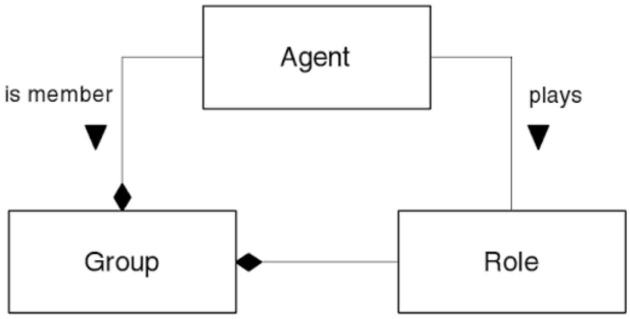


Figure 2.2: AALAADIN Metamodel. Taken from [FG97]

and communication skills that they can share with other agents in order to solve specific problems. An agent of an *AMAS* system gets perceptions from its environment, is autonomous and acts to reach its own goals. Agents have rules to support a correct communication and to not be misunderstood while delivering a certain message [Ber+02].

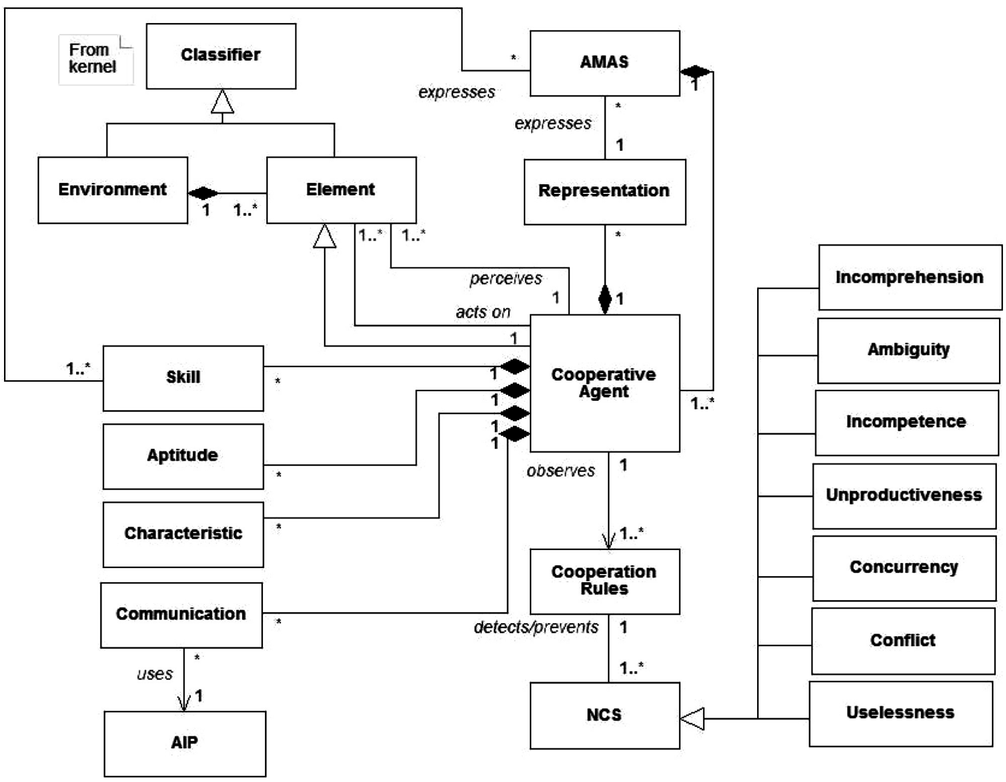


Figure 2.3: ADELFE Metamodel. Taken from [Bey+09]

2.2.2.3 GAIA

GAIA tries to model the social aspects of open agent systems (systems where there is no access to its internal architecture and that we have no guarantee of his behaviour and cannot predict its interactions), with particular attention to its organisational rules, tasks and social goals (Figure 2.4). Agents are part of an organisation, collaborating with other agents that have different attributes (different agent types), providing services and playing different roles. All communications between agents and organizations are defined by a protocol. [Woo+00].

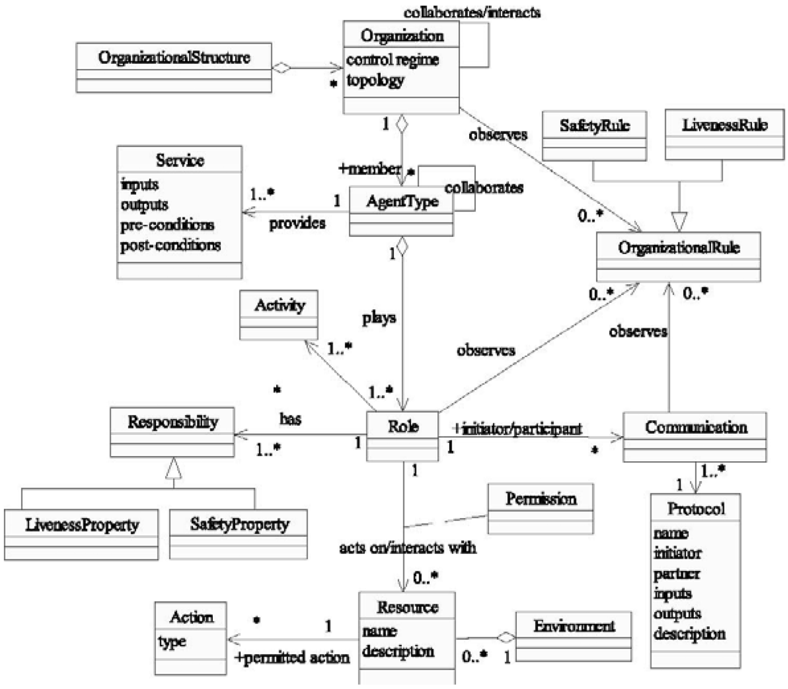


Figure 2.4: GAIA Metamodel. Taken from [Bey+09]

2.2.2.4 PASSI

Process for Agent Societies Specification and Implementation (PASSI) is a methodology for *MAS* defined by an agent-oriented iteration (Figure 2.5). It is concerned with three different domains [Puv+08]:

1. **Solution Domain** — Composed by agents, services and tasks
2. **Agency Domain** — Describes the agent domain. Agents have a set of roles and provide certain services that allow solving tasks (which are composed by actions);
3. **Problem Domain** — Covers live resources, non-functional aspects (accessibility, re-usability, usability) and requirements that can be connected with an agent.

2.2.2.5 Unified MAS Metamodel Proposal

Unified MAS Metamodel is a tentative metamodel (Figure 2.6) that tries to merge the best characteristics of *AALAADIN*, *ADELFE*, *GAIA* and *PASSI*. It tries to cover the cooperative behaviour of *ADELFE*, the organisational scheme of *GAIA* and the communication

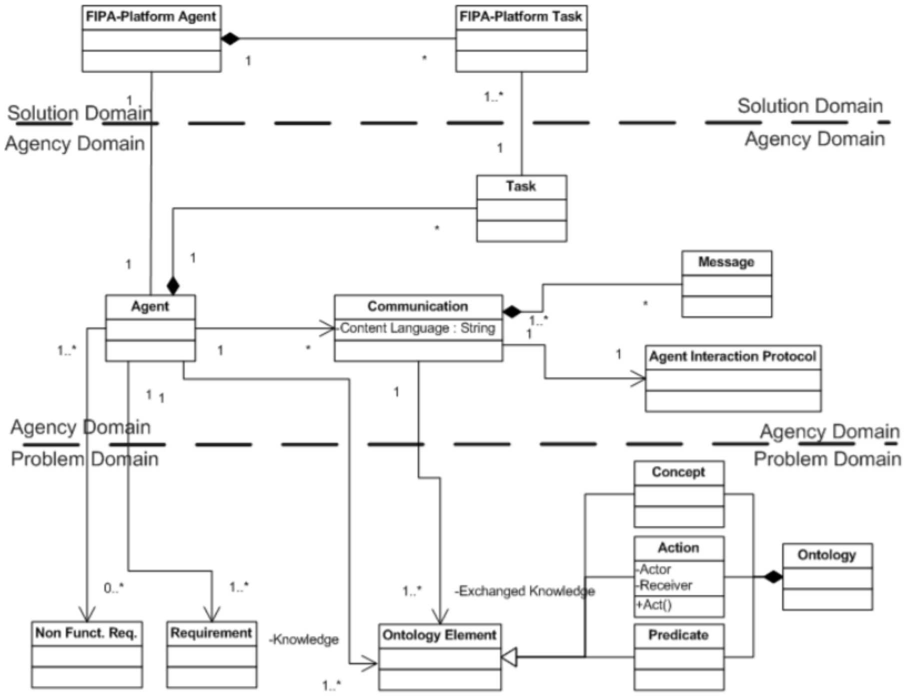


Figure 2.5: PASSI Metamodel. Taken from [Sei+10]

structure of *PASSI* [Hah+09].

2.3 Usability

In computer science, usability can be defined as a qualitative attribute that determines how comfortable user interfaces are to use. It defines if a certain tool is ready for a specific audience with a big impact in efficiency, effectiveness and satisfaction in a certain context of use [Nie03].

For *ISO*, the definition of usability refers it as *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."* [Jok+03]. With this definition, we can define some important quality components [Nie03]:

1. **Learnability** — How easy it is for users to accomplish the tasks they are prompt to do when they encounter the program for the first time?

2. **Efficiency** — After facing the tool and its design, how much time will users consume to perform a certain type of task?

3. **Memorability** — After a while of not using the tool, how much time will the user need to know how to use it again?

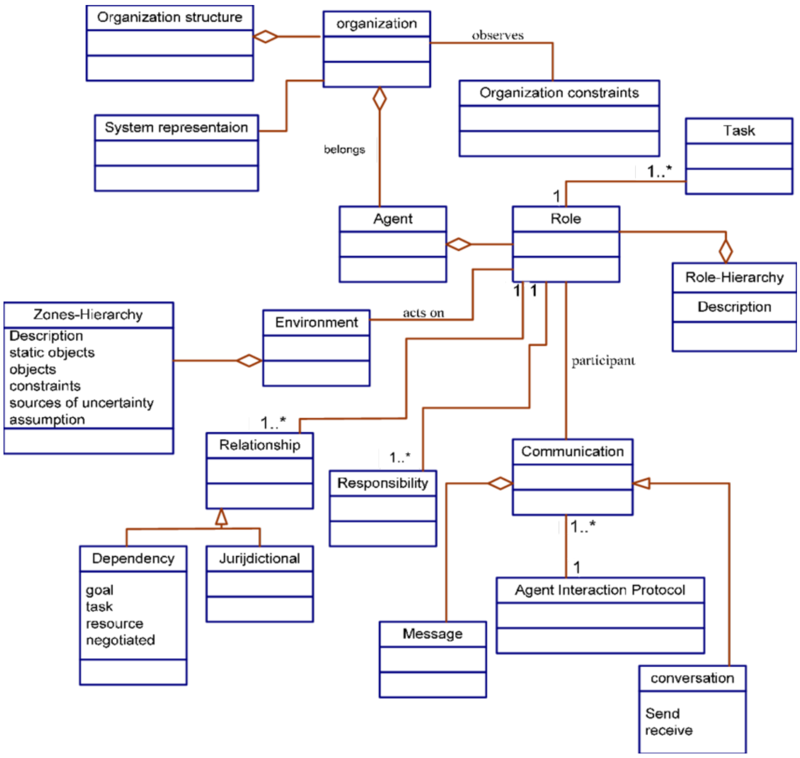


Figure 2.6: Unified MAS Metamodel Proposal. Taken from [EE+11]

4. **Errors** — When testing the tool, how many errors do the users make, how easily do they recover from them and how severe are the errors made?

5. **Satisfaction** — Does the user enjoy working with the tool?

6. **Utility** — Does the tool do what the user needs?

The usability of a *DSL* can be measured through a set of metrics. Metrics are defined previously to a usability experiment and enable the extraction of information from the users that are testing the tool. Common metrics can characterize at what speed users are doing the tasks they are prompt to do, what is the success rate of those tasks and the subjective users' satisfaction while executing each task, among others.

In the context of this thesis, we intend to evaluate the cognitive effectiveness (how easy it is for the user to understand the interface, how much time he takes to deal with a given problem and what is the users successful rate when given a specific task) of the four evaluated *DSLs*. Since all languages under study have a visual workbench, the metrics that are going to be applied will be related to the syntax of each language and how users respond with the type of problem they are facing with. [Moo09] presents a set of principles that allow the evaluation of visual notations. It proposes modifications for visual languages in order to make them more usable and easier to be understood by stakeholders.

2.3.1 Why is Usability Important?

It is essential for a tool to be usable. If it is not easy and intuitive to use a tool, users tend to leave it. If users get lost on a tool, find it hard to information to solve their problems or even do not find answers to their key questions, the tool eventually will be left aside by the user [Nie03].

Users tend to not read a manual or obtain previous information of a tool before they use it [NW06]. Their first impression with it is usually when they test the tool for the first time. This experiment takes place through the tool's interface.

For a *DSL*, usability problems become even more clearer: the lack of examples and related problems with ideas on how to solve it tend to frustrate users. A poorly designed DSL may be too confusing and fail to capture the essential abstractions of the domain, resulting in a mismatch with its intended end users' expectations. When using a new *DSL*, users should be satisfied on using it, not frustrated. They should enjoy the experience and understand fast how the *DSL* is and how they can use it, to achieve their goals efficiently and effectively. Doing so, users will feel confident and will enjoy the experience using the *DSL*.

2.4 DSL to be Evaluated

This sub-chapter will introduce the *DSL* that will be evaluated in terms of its usability.

2.4.1 SEA_ML

Semantic Web Enabled Agent Modeling Language (*SEA_ML* from now on) is a model driven (MDD) *DSL* that supports the modeling and generation of action-based systems [Cha+14]. *SEA_ML* allows users to model and generate architectural artifacts for *MAS*, with its main focus on the semantic web. Developers can model agent systems in a platform independent level.

SEA_ML introduces new viewpoints (Agent-Semantic Web Services - Agent-SWS from now on - interaction and ontology) for supporting the development of software agents working within the semantic web environment. Its syntax covers aspects of the domain such as the environment, plan and role.

2.4.1.1 Semantic Web

The *semantic web* is an extension of the current web in which information is given in a well-defined meaning. This structured web is presented to the end-users according to their needs. The web can be interpreted with ontologies, helping machines understanding the web content. Such interfaces of the semantic web can be discovered by software agents, with these agents interacting with the services in order to complete tasks. [BL+01]

In a *semantic web* enabled *MAS*, software agents can gather web contents from various resources, process the information, exchange the results and negotiate with other agents.

2.4.1.2 Metamodels

SEA_ML metamodel is divided into eight viewpoints, with each one describing a different aspect of the semantic web for *MAS*:

1. **MAS** — General view of the *MAS* system. It is composed by a *semantic web organization (SWO)* that interacts within an environment. The *SWO* has different types of roles, representing the organization aims (e.g. for health-care, trading, and so on). On this *SWO* we have *Semantic Web Agents (SWA)* working on it. These agents inside the organization can be at more than one organization at the same time. Agents have different skills and attributes and may collaborate between them in order to solve their problems.

2. **Agent Internal** — Focuses on the agent internal structure in the *MAS* organization. Each *SWA* plays a role, determining the way he behaves in the *MAS* system. As such, the *SWA* can be from a different type (e.g. in a health-care system we have medics, nurses, and so on). Each *SWA* has certain capabilities and beliefs that are used to reach a certain goal previously planned.

3. **Plan** — The internal structure of the *Agent Internal Viewpoint* plan section. When an agent applies a plan, it needs to execute certain actions in order to complete several tasks (to solve the prompt plan). Each action is connected with an entity called *message*, that sends and receives them (actions are atomic, which means we receive the same quantity of messages we send).

4. **Interaction** — Focuses on the interaction and communication of agents in *MAS*. Each interaction includes a message, that is structured in a certain order (it has a sequence in order for it to make sense). Each agent interaction is based on its social abilities.

5. **Role** — *SWA* and *SWO* can play roles and use ontologies to maintain their internal structure and knowledge, inferring the behavioural facts of the environment. Each agent has a certain role and does certain interactions within the *MAS* (with other agents, for example).

6. **Environment** — The environment viewpoint focuses on the relationships between the agents and to what they access. It's where the agents reside, containing resources and services that are used by the agents. The environment also acknowledges facts (e.g. there are 20 patients on the hospital), that can relate important information for the way agents interact with each other on the environment.

7. **Agent-SWS** — Models the interaction between the agents and the *semantic web services (SWS)*. This viewpoint deals with the services and agreements that have to be accomplished to support successful communications and relationships. *MAS* and *SWS* are viewed as two different systems that can interact with each other in order to realize successfully certain tasks. A *SWS* is composed of web services, called to action when needed in order to complete a certain task that is previously planned. These plans are applied by a *SWA* that plays a certain role on the system, interacting directly with the *SWS*.

8. **Ontology** — Viewpoint that reunites all the ontology sets and concepts together. An ontology represents any information gathering and reasoning resource for the *MAS* agents. A collection of ontologies creates a knowledge basis of the *MAS* that provides a domain context. In *SEA_ML*, ontologies are represented as instances of *OrganizationOntology*.

These viewpoints deal solely with the construction of a *MAS* as an overall aspect of the metamodel. It includes the main blocks, which compose the complex system, as an organization. The software ontology can include several agents at any time and each organization can be composed of several sub-organizations recursively. This composition has agents with similar goal duties.

2.4.2 Example: Stock Exchange System

On [Cha+14], a case study is presented for a *Stock Exchange System*. An investor intends to buy some stock. The investor has access to the trade information service and should be able to consult a broker that finds the proper stock and seller for the investor. For the order to be processed, the investor should be informed of the rate of exchange and conditions for the transaction to be successful. It is assumed that exists a web service for finding, negotiating and exchanging stocks on the stock exchange market.

Three viewpoints are focused on this example: *MAS* (Figure 2.7) , Agent Internal (Figure 2.8) and Agent-SWS (Figure 2.9).

2.5 Similar *DSLs* available on the market

This sub-chapter will introduce *DSLs* that are similar to *SEA_ML*.

2.5.1 DSML4MAS

Domain Specific Modeling Language for Multi-Agent Systems (DSML4MAS) is a platform independent language for specifying multi-agent systems [Hah08]. It is developed based on the principles of *MDD*, linking design and code through transformations to generate executable code. The abstract syntax of *DSML4MAS* is defined by *PIM4AGENTS*, which divides the *MAS* into seven viewpoints.

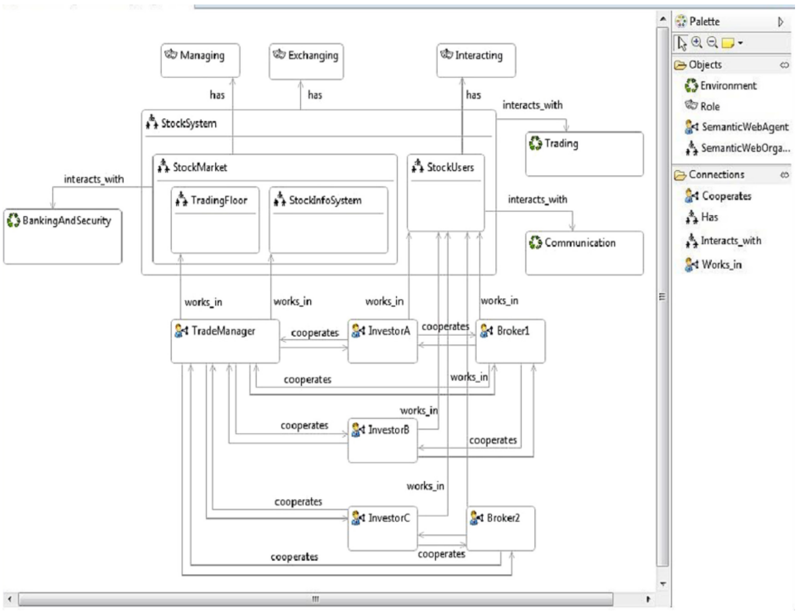


Figure 2.7: MAS Viewpoint for the Stock Exchange System. Taken from [Cha+14]

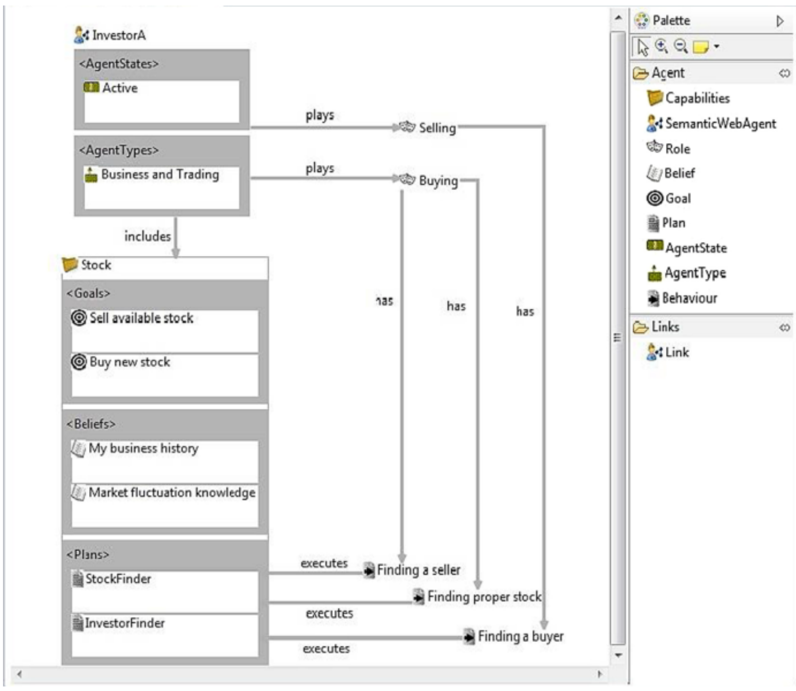


Figure 2.8: Agent Internal Viewpoint for the Stock Exchange System. Taken from [Cha+14]

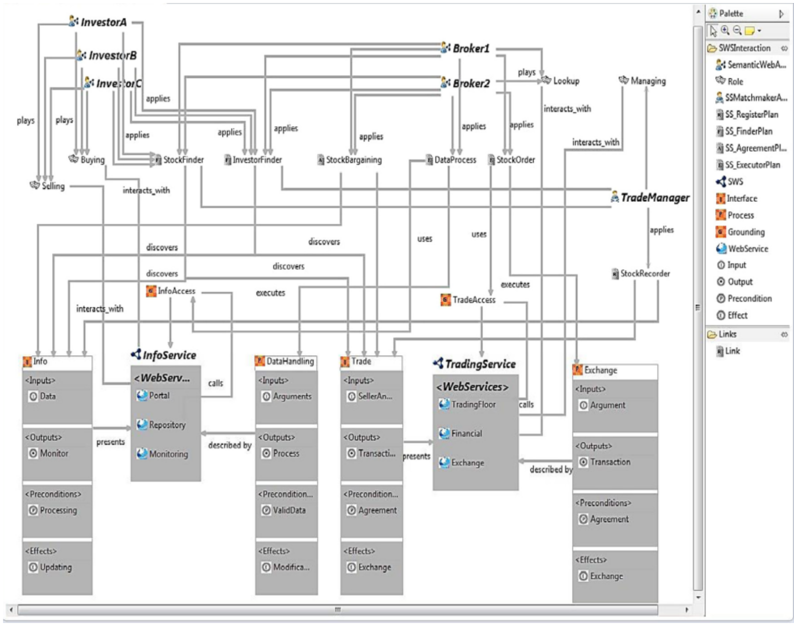


Figure 2.9: Agent-SWS Viewpoint for the Stock Exchange System. Taken from [Cha+14]

2.5.1.1 PIM4AGENTS

Platform-Independent Metamodel for Agents (PIM4AGENTS), as the name suggests, is a platform independent metamodel for *MAS* that can be used to model agent systems in an abstract manner without focusing on a certain platform specific requirements [Hah+09].

It is developed using the *MDD* approach. *PIM4AGENTS* allows users to design agent-based systems, reducing the gap between traditional software engineering approaches and agent-based system design. It divides *MAS* in seven important viewpoints: MAS, Agent, Organization, Role, Behaviour, Interaction and Environment.

Since *PIM4AGENTS* tries to be a platform-independent metamodel, it needs to make some transformations in order to work with similar agent-specification tools. For that reason, *PIM4AGENTS* has transformations for *JACK* and *JADE*, due to their focus on different aspects (*JACK* focuses on *BDI* and *JADE* on *FIPA*). The new *JACK* metamodel is defined as *JACKMM* and the new *JADE* metamodel as *JADEMM*.

2.5.1.2 Metamodels

DSML4MAS is divided into seven viewpoints: *MAS*, Agent, Role, Organization, Interaction, Behaviour and Environment:

1. **MAS** — General view of the *MAS* system. It contains the main blocks of a *MAS*: Agent, Instance, Cooperation, Capability, Interaction, Role, Behaviour and Environment.

2. **Agent** — Focuses on the agent internal structure in the *MAS* organization. An agent is part of a certain organization, having access to a set of resources that can

include ontologies of the environment he is inserted in. On the organization he is inserted in, he performs a certain role, defining the specific context he is acting and behaviours that demonstrate how particular tasks are being achieved. Each agent has a certain set of capabilities that group with a particular type of behaviours.

3. **Role** — Covers achievable specializations and how they could be related with each other. An actor is called for a certain role, adapted to the problem domain. This role gives the actor a certain set of resources, enabling new capabilities for him that can be used to solve new problems on the domain he is working on. A role is an abstraction of the agents social behaviour in a given social context. The actor inherits from the role its capabilities and resources that are necessary for exchanging messages with other actors.

4. **Organization** — The organization viewpoint describes how each entity cooperates within the *MAS* and how complex organizational structures can be defined. An organization defines the social structure in which agents can take part. It is composed by agents that have roles attributed to them. In order to solve problems, agents cooperate among them, with this communication being defined by the organization that has a certain protocol for those circumstances.

5. **Interaction** — Covers how entities and organizations interact between them. Actors interact with others with messages. These messages are received and, as soon as possible, replied. The messages are sequenced in order for the communication to be successful. Also, the interaction is based on a protocol. If, for instance, a message does not receive a reply on a certain time, a timeout can be triggered.

6. **Behaviour** — Describes how plans are composed by a complex control structure with simple atomic tasks such as sending a message and how the information flows between those artifacts. Behaviour connects with the agent behavioural aspect. Depending on the plan and the activities to be done, behaviours will be made in order to solve the respective tasks.

7. **Environment** — The environment aspect delivers resources that are dynamically created, shared or used by agents or organizations. A resource contains a set of documents, including classes with attributes. Agents can influence the environment to make it change or to extract information. Also, agents can communicate indirectly via the environment, adding and reading information from it.

2.5.2 Example: Conference Management System

On [War13] a Conference Management System is presented using *DSML4MAS*. The Agent viewpoint (Figure 2.10) presents two different agents: the Senior Researcher, that can be part of the program chair as its chairman or as member only. The Senior Researcher can send call for papers for researchers, assign papers for other chair member or partition papers. Also, the Senior Researcher can submit or review a paper. A Researcher can only submit or review other papers.

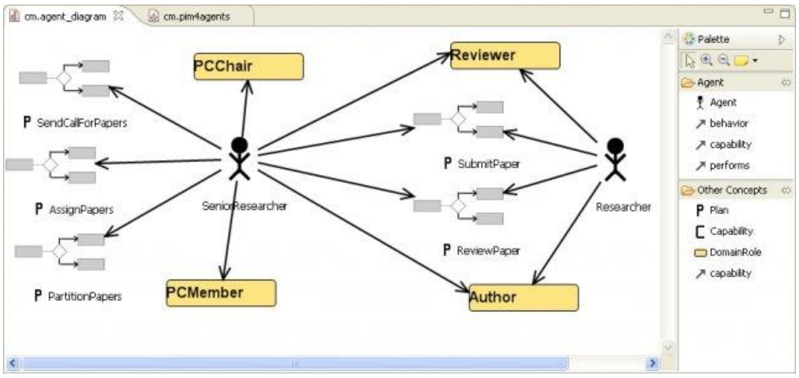


Figure 2.10: Agent Viewpoint for the Conference Management System. Taken from [War13]

The Behaviour viewpoint (Figure 2.11) presents how the paper submission is managed on the system.

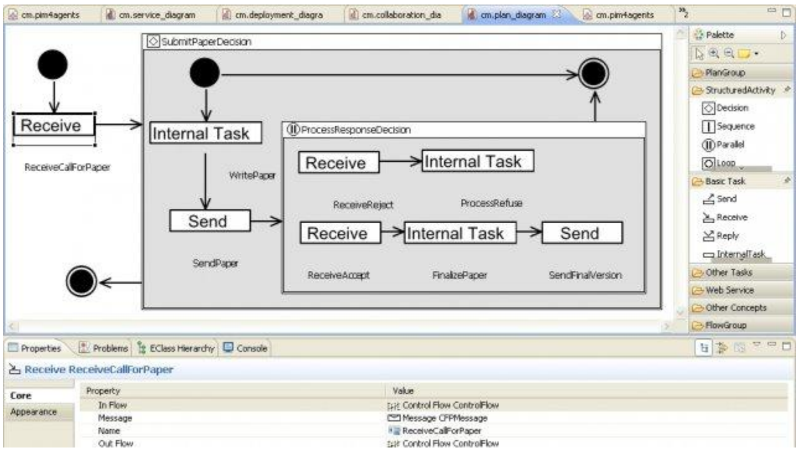


Figure 2.11: Behaviour Viewpoint for the Conference Management System. Taken from [War13]

The Organization viewpoint (Figure 2.12) shows how the system is represented as a general view. It is composed by the conference organization, represented by authors, reviewers, program committee members and chairs. The conference organization has protocols defined for the paper submission, paper assignment and paper partition.

2.5.3 INGENIAS

INGENIERIA de Agentes de Software (INGENIAS) proposes a modeling language for *MAS*. It has recommendations for the development of these types of systems. *INGENIAS* is a *model-driven engineering (MDE)* approach, having its own support tools, which concerns code generation, documentation and a visual editor for the specification to be made.

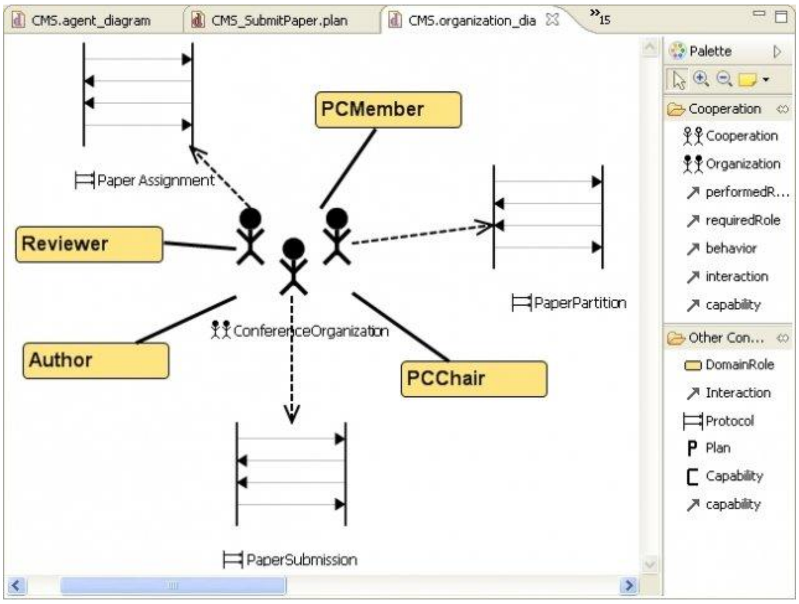


Figure 2.12: Organization Viewpoint for the Conference Management System. Taken from [War13]

For *INGENIAS*, an agent is the basis to develop complete software systems. As such, *INGENIAS* is an agent-oriented software engineering methodology [Pav+05].

INGENIAS is a living methodology, not finished, growing to incorporate new knowledge on the modeling languages for *MAS* [GS14].

2.5.3.1 Metamodels

INGENIAS is divided in five viewpoints: Organization, Agent, Goals/Tasks, Interactions and Environment:

1. **Organization** — Describes where agents, resources, tasks and goals coexist. An organization has a structure, a certain functionality and social relationships. It has a set of entities with relationships of aggregation and inheritance. An organization pursues a certain goal. This goal is to be achieved by a group. A group may contain agents, roles, resources or applications. Groups are useful when the number of elements of the *MAS* increases. These groups must obey to some organizational rules and purposes, facilitating workflows (groups usually have, for example, agents with some common abilities). An organization can also be defined by its purpose and tasks, and can have one or more goals depending on the agents performing the tasks to achieve them.

2. **Agent** — An agent is defined by its purpose (what goals the agent intends to reach), his responsibilities (what tasks should he do to achieve it) and capabilities (what roles does he play). The agent behaviour is defined in three components, all rounding a similar aspect: the agent mental aspect. Each agent has mental state, consisting

in an entity that aggregates his beliefs, facts, goals and settlements, a mental state manager, that allows the agent to create, destroy or modify the entities referred above, and finally a mental state processor, that determines how this entities evolve in terms of planning and rules.

3. **Goals/Tasks** — This viewpoint proposes the decomposition of goals/tasks, describing the consequences of performing a task and why the task should be performed in order to reach a certain goal. It determines, for each task, what elements are required and what results should be expected. Goals are influenced by the tasks to be executed and their success/failure. This viewpoint also explains how solved goals affect other existing goals.

4. **Interactions** — The interaction viewpoint discusses the exchange of information or requests between agents. For each interaction we have a specification of the interaction and the agents who are addressing the interaction. Interactions should previously be specified in order to avoid misunderstandings. The agent role also is important for the interaction (e.g. on a stock exchange system, an investor has the role to buy and sell stock, while a trade manager supervises the exchanges being made in the system). These interactions are intended to reach a certain goal.

5. **Environment** — Defines the entities with which the *MAS* interact. They can be resources (elements required by tasks in order to achieve goals), other agents (from the same or different organizations) and applications (that offer application interfaces that can be useful to solve a certain problem).

2.5.4 Example: Bookstore Electronic Sales System

On [GSP05], *INGENIAS* is used to define a bookstore electronic sales sytem. This bookstore sells books to students of one university. It has an agreement with professors and students in order to obtain the books that will be used on their courses, selling them at special prices. The bookstore is an organization, having a department for sales and logistics. The bookstore negotiates with other organizations (e.g. publishers), in order to acquire books at the best possible price. The bookstore sells books on a physical store or via web.

The case study pretends to define an electronic sales system. It divides the bookstore on two main departments: Logistics and ESales(Electronic Sales). The first is responsible for delivering the books to the customers from the publishers. The second for interacting with its customers.

Figure 2.13 presents the Bookstore Electronic Sales System using INGENIAS viewpoints and notation. Rectangles with three circles above represent an organization, the rectangle with two circles a group, circles represent goals, workflows are represented with linked ovals and roles with the hollowed squares. On the example, the JuulMoller

Enterprise intends to SellBooks in order to ObtainBenefits. It is structured in two groups: ESales and a Logistics department, each one with each set of roles.

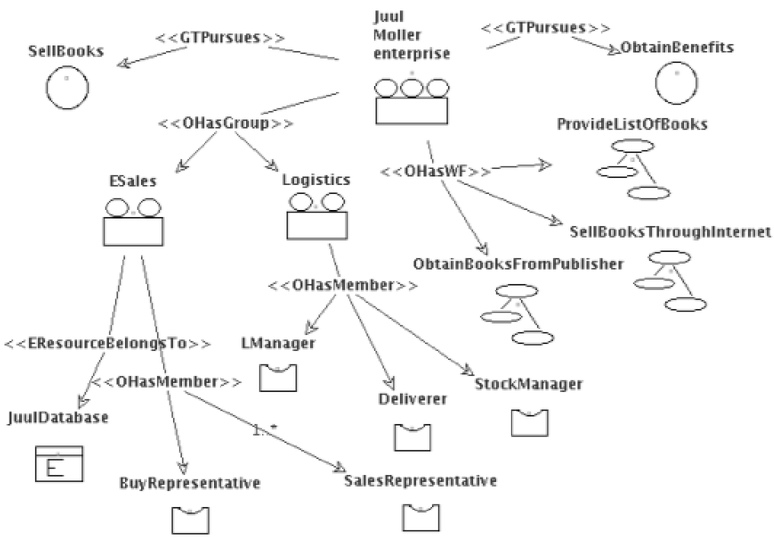


Figure 2.13: Bookstore Electronic Sales System on INGENIAS. Taken from [GSP05]

2.5.5 MAS-ML 2.0

Multi-Agent Systems Modeling Language 2.0 (MAS-ML 2.0 is a *MAS* modeling language that is made through the extension of *UML*, incorporating agent related concepts to represent proactive agents. It is based on the *Model-Driven Architecture (MDA)* approach.

For *MAS-ML* developers, a modeling language for *MAS* should be an incremental extension of a known and trusted modeling language. With the basis of *UML* and since agents and objects can coexist in *MAS*, *UML* can be used for modeling these types of systems. *MAS-ML* performs a conservative extension of *UML*, enabling users to model agent characteristics, and it is based on the agent-oriented defined in *Taming Agents and Objects (TAO)*.

MAS-ML gives support to the modeling of *MAS* entities (and their respective static and dynamics properties), Agents (and agents roles), Organizations and Environments [Gon+15].

MAS-ML agents have goals and beliefs and structural features, having plans and actions on their behavioural feature-plans, that are executed to achieve goals, composed by actions.

MAS-ML divides agents in four possible types:

1. **Simple Reflex Agent** — Simple agents that base their work on the system on a condition-action basis. Are used to select the actions based on the current perception of the system. Similar to an if clause (if a certain condition is verified, then

some action will occur), where an action will be executed if a perception occurs. At any time the agent can receive information from the environment through sensors.

2. **Model-Based Reflex Agent** — On the same line of the Simples Reflex Agents. Deals with the information by using condition-action rules. An agent is also able to store its current state in an internal model.

3. **Goal-Based Agent** — Similar to model-based agents but with a set of specific goals and actions that lead to those goals. This allows the agents to choose a goal state among multiple possibilities, involving the formulation of problems and goals, with plans and actions to reach them.

4. **Utility-Based Agent** — Utility-based agents optimize the agent performance. It is responsible for mapping a possible state or group of states according to the current goal that he is dealing with.

2.5.5.1 UML

Unified Modeling Language (UML) is a standard for object-oriented modeling. It is commonly used in both academia and industry for modeling object oriented systems. It allows users to visualize the architecture of a certain in the form of a diagram. Activities, components and interactions are covered by *UML* [Rum+04]. *UML* diagrams are divided in structural and behavioural diagrams:

1. **Structural Diagrams** — Represent what the system must have in order to work. Class, component, deployment and object diagrams are covered here.

2. **Behavioural Diagrams** — Represent what should be happening in a system. Activity, interaction, sequence, state and use case diagrams are covered on this division.

2.5.5.2 Taming Agents and Objects (TAO)

TAO is a conceptual framework that provides users an ontology that covers the fundamentals of software engineering based on agents and objects, making possible the development of *MAS* in large scale [Sil+02]. *TAO's* ontology corrects consolidated abstractions like classes and objects, and *MAS* new abstractions such as agents, organizations and roles, that are fundamental foundations for *agent and object based software engineering* (also known as *AOSE*).

2.5.5.3 Metamodels

MAS-ML 2.0 models all aspects defined in the *TAO* metamodel by extending the UML metamodel. *TAO* metamodel consists of six viewpoints:

1. **Agent** — An agent is an autonomous, adaptive and interactive element that has a mental state, having beliefs, plans, goals and action to achieve them;

2. **Agent Role** — Guides and restricts the agent behaviour, having a role assigned to him on an organization.
3. **Environment** — Agents, objects and organizations habitat. It has a state and a behaviour.
4. **Object** — Passive or reactive element that has a state and behaviour and can be related to other elements.
5. **Object Role** — Guides and restricts the object behaviour. Can add information to the object instance.
6. **Organization** — Groups agents and sub-organizations that play roles and have common goals. An organization hides intracharacteristics, properties and behaviours of their agents and sub-organizations through axioms, which constraint the organization.

These aspects are covered on *MAS-ML 2.0* with static and dynamic viewpoints from *UML*.

1. **Static Viewpoint** — Contains the class, organization and role diagrams. Class diagram can represent an entity, with the same relationships gathered on *UML* (association, aggregation and specialization). The organization diagram represents organizations, sub-organizations, classes, agents, agent roles, objects, object roles and its environment. The Role diagram represents the agent and object role and its relationships.
2. **Dynamic Viewpoint** — Represents the sequence and activity diagrams presented in *UML*. The sequence diagram represents objects, agents, organizations and environment, while the sequence diagram identifies agents or organizations, the roles they are playing and the flow of a certain task to be accomplished.

2.5.6 Example: MAS to Schedule Lectures in Conferences

In [Cam+10] an example of *MAS-ML* modeling is presented. The example consisted in using *MAS* to have the best schedule for lectures in a conference (Figure (2.14)). Each agent has a list of subjects of its interest and a timetable with session of one hour, sessions that can occur from 8 a.m. to 6 p.m. On each hour the timetable may be free or busy with a certain activity. A lecture of a specific subject can be scheduled on the timetable on a free hour. The list of interest and the initial state of the timetables are started randomly. The agents can play two roles: the organizer of the event or the participant. The organizer schedules a lecture about a subject of its interest and tries to achieve the highest possible number of participants. Participants try to attend the highest possible number of lectures of their interest. The lectures promoted by the organizer require the organizer to be present. Agents can also change its role. (Figure (2.15))

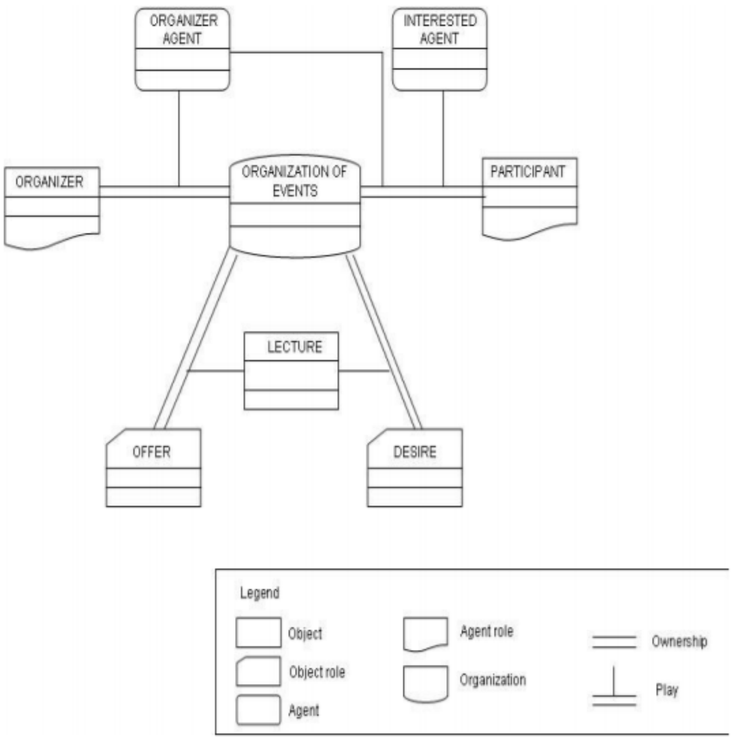


Figure 2.14: Organization Diagram for the Conference Lecture Scheduler on MAS-ML 2.0. Taken from [Cam+10]

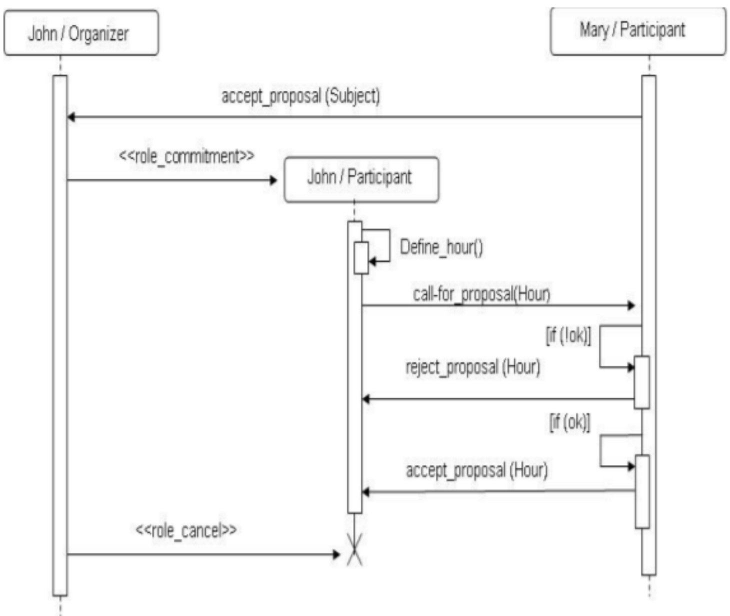
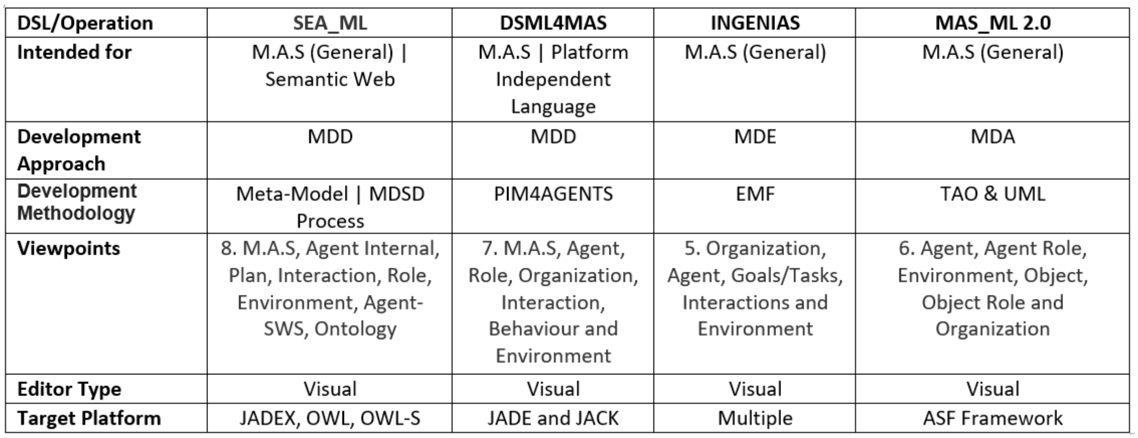


Figure 2.15: Change of Role Diagram for the Conference Lecture Scheduler on MAS-ML 2.0. Taken from [Cam+10]

2.5.7 Comparison between each *DSL*

Figure 2.16 compares each *DSL* that was presented above. We can verify that every *DSL* is intended for *MAS*, with *SEA_ML* allowing also the modelling of the *Semantic Web*. *SEA_ML* and *DSML4MAS* use the same development approach (MDD), while *INGENIAS* and *MAS_ML 2.0* have different approaches (MDE and MDA, respectively). Each *DSL* has a different development methodology. As for viwpoints, *SEA_ML* delivers more options than the remaining *DSLs'*, as it adds more options due to the fact that is modules artifacts from the *Semantic Web*. All of the *DSLs'* presented have visual editors and are intended for different target platforms.



DSL/Operation	SEA_ML	DSML4MAS	INGENIAS	MAS_ML 2.0
Intended for	M.A.S (General) | Semantic Web	M.A.S | Platform Independent Language	M.A.S (General)	M.A.S (General)
Development Approach	MDD	MDD	MDE	MDA
Development Methodology	Meta-Model | MDSD Process	PIM4AGENTS	EMF	TAO & UML
Viewpoints	8. M.A.S, Agent Internal, Plan, Interaction, Role, Environment, Agent-SWS, Ontology	7. M.A.S, Agent, Role, Organization, Interaction, Behaviour and Environment	5. Organization, Agent, Goals/Tasks, Interactions and Environment	6. Agent, Agent Role, Environment, Object, Object Role and Organization
Editor Type	Visual	Visual	Visual	Visual
Target Platform	JADEX, OWL, OWL-S	JADE and JACK	Multiple	ASF Framework

Figure 2.16: *MAS DSL* comparison

2.5.8 Summary

This chapter presented the main topics of this dissertation. The definition of what is a *DSL*, what is *MAS* and how usability is important in computer science for the evolution of a programming language should be retained for the remaining of this document. A presentation of *SEA_ML*, the language to be evaluated, was also made. *SEA_ML* was also compared with similar *DSLs'* available on the market.

Chapter 3

Related Work

In this chapter some work will be introduced that allow the evaluation of the usability of *DSLs*.

3.1 Using the USE-ME methodology to evaluate the usability of a DSL

Barišić et al. introduce a conceptual framework that supports the iterative development process of *DSLs* concerning the usability evaluation [Bar16]. This framework allows *DSL* developers to evaluate the usability of a *DSL* while it is being developed or evolved. *Barišić et al.* divides the *DSL* lifecycle in six major steps (Figure (3.1)):

1. **Decision** — Identifies the need of that *DSL* for that domain and justifies that the efforts to be made are worth it for its creation. The *DSL* requirements are defined on this section. Language engineers and domain experts are needed for this phase;

2. **Analysis** — Defines the domain, feature, functional and goal concepts for the *DSL*. Similar to the decision phase, language engineers and domain experts are needed for this phase;

3. **Design** — Formalises the language, introducing its abstract and concrete syntax as well as its semantics. This is made possible by the language engineer, who defines the expected behaviour of the language elements;

4. **Implementation** — The development of the tool designed on the previous steps. It focuses on the domain, goals and requirements previously defined. It includes the *DSL* artifacts and the needed transformations. Language engineers will develop the tool;

5. **Testing/Evaluation** — The most important topic of this section. It allows testing and validating the created *DSL*, checking if the proposed functionalities are being provided by the *DSL* in an adequate way for its intended users. With the help of tools such as model checkers and simulators, language engineers and domain experts perform verification and validation activities, which may include end users;

6. **Deployment** — If all previous steps are correctly completed, the *DSL* is delivered to its users with its respective documentation and proposed maintenance service. Language engineers should address this phase.

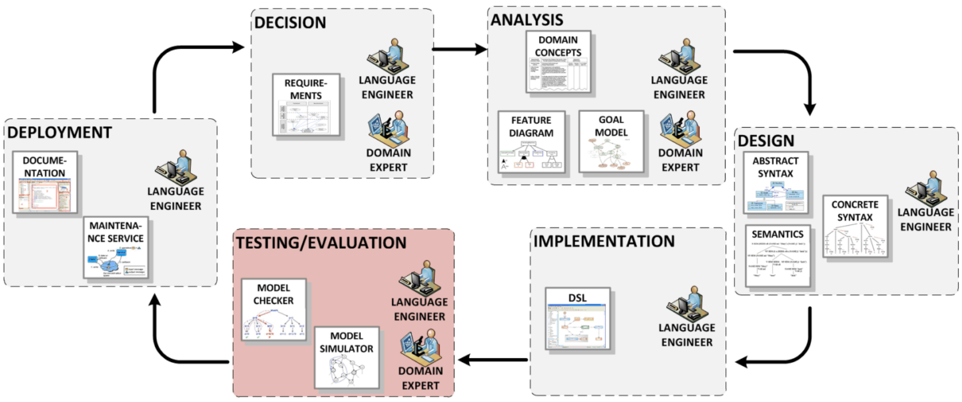


Figure 3.1: DSL Lifecycle. Taken from [Bar16]

A *DSL's* usability can be tested in several aspects, usually requiring users to experiment the tool on a controlled experience that allows the developers to retrieve relevant information that can help finding ways to improve the usability aspects of the language.

Usability evaluation is the main focus of *USE-ME*, allowing the modelling activities that are expected to be performed by the expert evaluator. To test if a language is usable 5 major steps are defined (Figure (3.2)): Context modelling, goal modelling, evaluation modelling, evaluation execution and report modelling.

3.1.1 Context Modelling

The context modelling activity defines the context of use for the *DSL*. Questions such as who will use the *DSL*, where will it be used and how is it expected to be used are managed in this activity. This activity engages all stakeholders that are involved in the development of the tool. Domain experts and language engineers are involved on this activity. User profiling helps to decide which users should be engaged in which phases of the evaluation and how their input can be leveraged in the language evolution. Who will use the *DSL* and where it is going to be used will support the evaluator in creating a workflow for the respective testers. Conditions such as the technical and social environment where the experiment will take place are also modeled here.

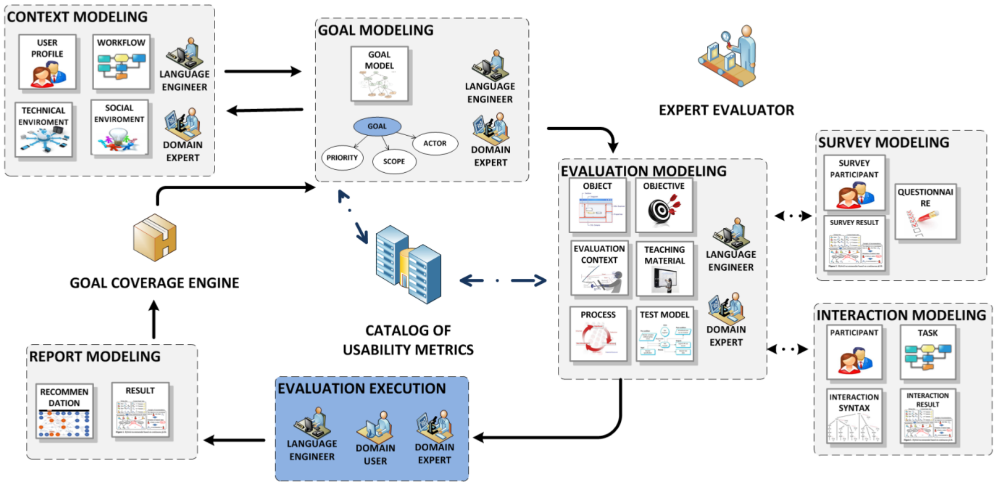


Figure 3.2: USE-ME Lifecycle. Taken from [Bar16]

3.1.2 Goal Modelling

In this section the goals expected for the user testing the tool are defined. The goals will have a certain priority defining whether or not the goal to reach is crucial to be reached on the *DSL* that is being tested. Questions such as why this language is being developed and whether the objectives for the *DSL* are reachable are discussed here. A goal captures, at different levels of abstraction, the various objectives the system should achieve. Usability is meant to be the highest goal of the developed *DSL* and should have actors (specialization of a stakeholder), a scope of the goal (tasks that should be accomplished in order to reach the goal), methods (requirements that contribute to achieve a goal) and, finally, the goals success coverage (percentage of achieved goals).

3.1.3 Evaluation Modelling

The evaluation model expresses the purpose of the evaluation (objectives to be achieved) in a specific context. It is composed by seven elements:

1. **Evaluation Goal** — Hypothesis and questions to be evaluates, with goals to be achieved;

2. **Language** — Which languages will be evaluated? Comparison between languages

3. **Evaluation Context** — Which users will test the tool? How are they going to test the tool and why?

4. **Participant** — Who will participate on the study?

5. **Documentation** — Materials that are provided to the user who is going to test the language;

6. **Test Model** — Questionnaires, interviews or observations that are made in order to gather information from the study being made;

7. **Process** — Concrete test that is going to be done.

For this phase we can define an interaction modelling and also a survey modelling. The first one is where the test models are defined, with summative methods for measuring the usability with concrete tasks that involve at least one language (to be evaluated). The survey modelling is similar to the previous but with a set of questions that are important for the evaluation

3.1.4 Evaluation Execution

On this phase users will be executing the study that was defined on the previous steps.

3.1.5 Report Modelling

The last phase of the usability testing. It helps on the construction of the final report for the experimented assessment. The evaluation results are based on the analysis of the result models for the different tasks modeled. Using the goal model as a reference, the report describes the obtained results. It further adds details concerning how participants met the goals, or why they failed to achieve those goals.

3.2 An integrated tool environment for experimentation in DSL Engineering

In 2016, *Häser et al.* introduced an integrated end-to-end tool environment to perform controlled experiments in *DSLs* [Häs+16]. This experiment environment is integrated in the *Meta Programming System (MPS)* (workbench developed by *JetBrains*). Similar to the eclipse workbench, it allows *DSL* creator to build a *DSL* on its own visual tool.

The environment supports steps in order to experiment a language. The experiment is formulated by the language engineer, defining details of what and how is the *DSL* going to be tested. The supported steps are as follows (Figure (3.3)):

1. **Experiment Planning** — The language engineer will define goals and hypothesis to test the *DSL*. Metrics and variables are also defined on this section.

2. **Experiment Operation** — Based on the experiment planned on the previous step, the users that are selected to test the *DSL* will, on this section, experiment the *DSL*;

3. **Analysis & Interpretation** — After the experiment has been completed, a descriptive statistic based on obtained results of the experiment will be available to the tester, allowing him to compare with the hypothesis previously defined;

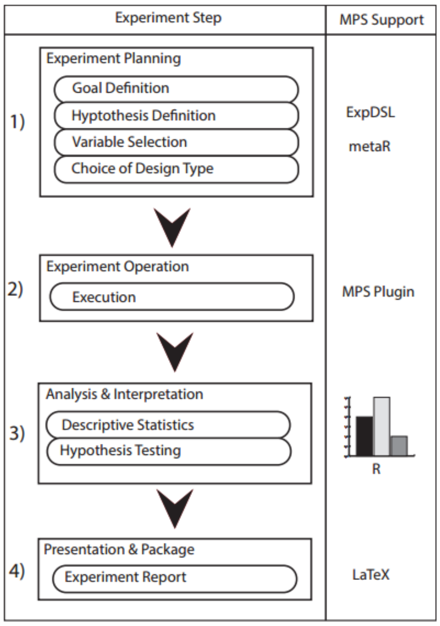


Figure 3.3: Integrated DSL Engineering Environment Step-By-Step. Taken from [Häs+16]

4. **Presentation & Package** — After the experiment is conducted, a process report is automatically created, generating a PDF with all the information that was generated with the experiment made. This PDF is defined using the structure of a *LATEX* document.

3.3 A systematic approach to evaluation DSML environments for *MAS*

[Cha+16] presented a framework to evaluate language environments for *MAS*. The evaluation criteria of the language tries to catch the essential elements of a language such as its abstract and concrete syntax, model transformations that are made, the input and output *MAS* models and its overall performance and development time. Evaluation regarding the end-user perspective is also taken in consideration (users are asked to answer a questionnaire concerning the experiment).

The experiment consisted in two groups (group A and group B) trying to build 4 case studies, all from different business domains, but all *MAS* based. Group A used SEA_ML, while group B used a generic software tool. The hypothesis concerned the development time, the testing time, the maintenance effort and the software quality. For the SEA_ML group, a questionnaire was proposed in order to understand the experience they had with the tool.

The test results show that users using the non-dsl version had spent half the time developing, testing and extending the proposed case studies. Also, the results from the questionnaires refer that the automatically generated code that was produced by the *DSL* included a better architecture, which increased the quality of the software developed.

3.4 The "Physics" of Notations

Along with evaluations relying on the feedback collected with language users, a language engineer can also reason about the merits of a *DSL*. One prominent approach to support this reasoning is known as "The "Physics" of Notations" [Moo09]. "The "Physics" of Notations" targets diagrammatic languages and is therefore suited for diagrammatic DSLs, rather than textual *DSLs*. It presents a set of principles to enhance the communication properties of a given language. "The "Physics" of Notations" defines the term *cognitive effectiveness* as the speed, ease, and accuracy with which a representation can be processed by the human mind, which is the basis of [Moo09] work. *Cognitive effectiveness* determines the ability of visual notations to communicate with stakeholders and support design and problem solving by software engineers. Visual notations should be well planned and evaluated in order to facilitate the interaction between users and diagrammatic *DSLs* (having every feature of the program with a graphical form may not guarantee the best interface).

It should be planned, implemented and evaluated with reason since transforming information into graphical form may not guarantee the best possible notation.

[Moo09] presents a set of nine principles (Fig.3.4) that improve the *cognitive effectiveness* of visual tools:

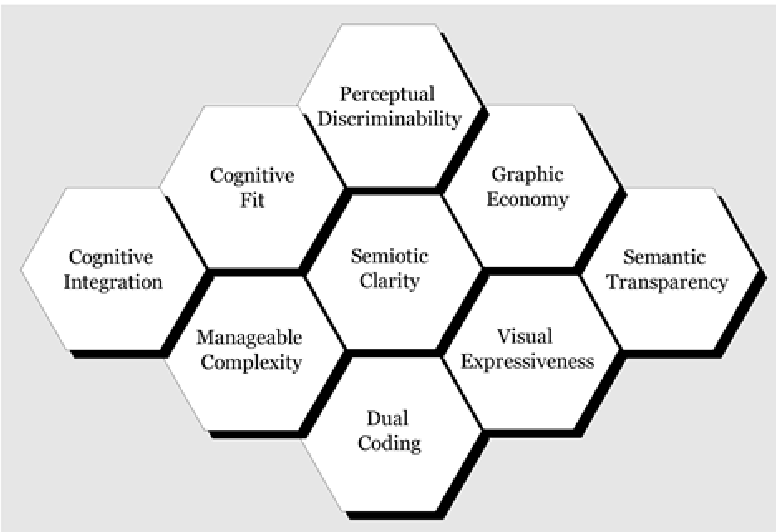


Figure 3.4: "The "Physics" of Notations" 9 principles. Taken from [Moo09]

1. **Semiotic Clarity** — There should be a 1:1 correspondence between semantic constructs and graphical symbols;

2. **Perceptual Discriminability** — Different symbols should be clearly distinguishable from each other;

3. **Semantic Transparency** — Use visual representations whose appearance suggests this meaning;

4. **Complexity Management** — Include explicit mechanisms for dealing with complexity;

5. **Cognitive Integration** — Include explicit mechanisms to support integration of information from different diagrams;

6. **Visual Expressiveness** — Use the full range and capacities of visual variables;

7. **Dual Coding** — Use text to complement graphics;

8. **Graphic Economy** — The number of symbols presented in a notation may affect the handling of the tool.

9. **Cognitive Fit** — Use different visual dialects for different tasks and audiences;

3.4.1 Semiotic Clarity

There should be a 1:1 correspondence between semantic constructs and graphical symbols (Fig.3.5). If we are on a situation where 1:1 correspondence is not verified, some misconceptions may occur such as:

1. **Symbol Redundancy** — Multiple symbols may represent the same semantic construct;

2. **Symbol Overload** — Two different options can be represented by the same symbol;

3. **Symbol Excess** — There are symbols that do not correspond to any semantic construction;

4. **Symbol Deficit** — A semantic construct is not represented by any symbol.

3.4.2 Perceptual Discriminability

Different symbols should be clearly distinguishable from each other. To allow a better comprehension of the information provided, some advises can be made:

1. **Visual Distance** — The greater the visual distance between symbols, the faster and more accurately these symbols will be recognized;

2. **The Primacy of Shape** — Symbol shapes should be used as the primary visual variable for distinguishing between different semantic constructs;

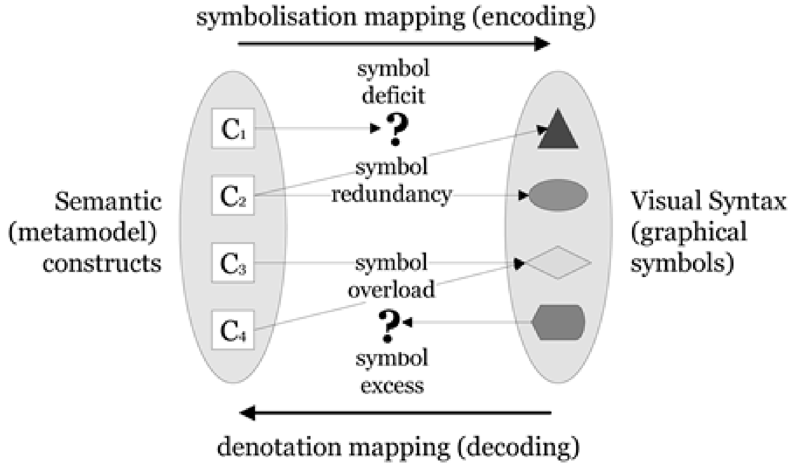


Figure 3.5: Semiotic Clarity attributes. Taken from [Moo09]

3. **Redundant Coding** — Using multiple visual variables to distinguish between them. Visual variables such as colours and shapes can improve the discriminability between agents and tasks, for example;

4. **Perceptual Popout** — Visual elements should be distinguishble and unique by at least one visual property;

5. **Textual Differentiation** — Text notation should not distinguish different symbols.

3.4.3 Semantic Transparency

Semantic transparency defines that a user can understand the meaning of a symbol by only looking at its appearance (Fig.3.6). Symbols can be:

1. **Semantically Perverse** — Symbol appearance suggests a different or opposite meaning;

2. **Semantically Opaque** — Symbol has a relationship with its meaning;

3. **Semantically Translucent** — Symbol has an association with its meaning but requires an initial explanation before using it;

4. **Semantically Immediate** — The meaning of the symbol can be inferred from its appearance without any initial explanation.

3.4.4 Manageable Complexity

Complexity management is the ability of a visual notation to represent information without overloading the human mind. Limits associated with this can be:

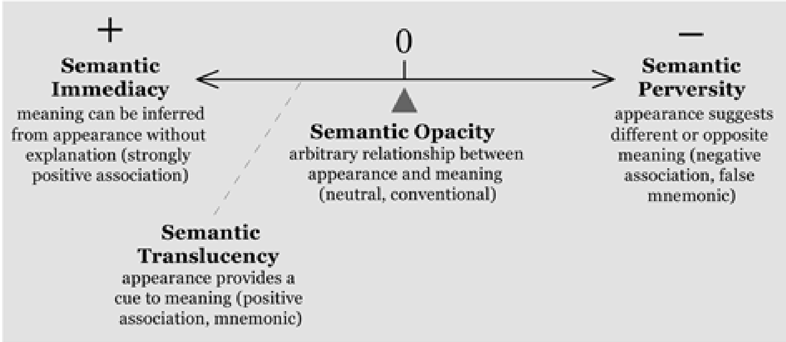


Figure 3.6: Semantic Transparency classification line. Taken from [Moo09]

1. **Perceptual Limit** — The ability to discriminate between diagram elements increases with the size of the diagram;

2. **Cognitive Limit** — The number of diagram elements that can be comprehended at a time is limited by the user working-memory capacity. If exceeded, we are on the verge of a *cognitive overload*, where the comprehension of its elements degrades quickly.

3.4.5 Cognitive Integration

Cognitive integration includes explicit mechanisms to support integration of information from different diagrams. This only applies when we have more than one diagram representing a system. The integration can be:

1. **Conceptual Integration (CI)** — Help the reader collect information from separate diagrams into a coherent representation of the system;

2. **Perceptual Integration (PI)** — Perceptual suggestions that simplify the users transitions and navigation between diagrams

3.4.6 Visual Expressiveness

Visual expressiveness is defined as the number of visual variables that are used in a notation. Variables can be used to encode information in a notation (*Information-carrying variables* or not formally used *free variables*.

3.4.7 Dual Coding

Dual coding can be defined as the text that accompanies a certain visual notation. Usually the text follows the icon when its not obvious what the icon is meant to do.

3.4.8 Graphic Economy

Graphic complexity is defined by the number of symbols presented in a notation. It affects mostly beginners due to their lack of practice with the respective tool. To deal with the complexity of a language, developers may reduce the semantic complexity, increase the visual expressiveness of the language and try to not show some constructs graphically (symbol deficit).

3.4.9 Cognitive Fit

The cognitive fit theory states that using different visual methods can be suitable for different tasks and different scopes of users. Novice users tend to have more difficulty understanding the difference between symbols and are more affected by complexity due to their inexperience. Also, it is more difficult for them to remember what are those symbols due to their lack of experience with the tool.

Chapter 4

Applying "The "Physics" of Notations" to SEA_ML

In this chapter the SEA_ML semantic constructs will be introduced and we will be analyzing it using "The "Physics"of Notations" nine principles, one by one.

4.1 SEA_ML Semantic Constructs

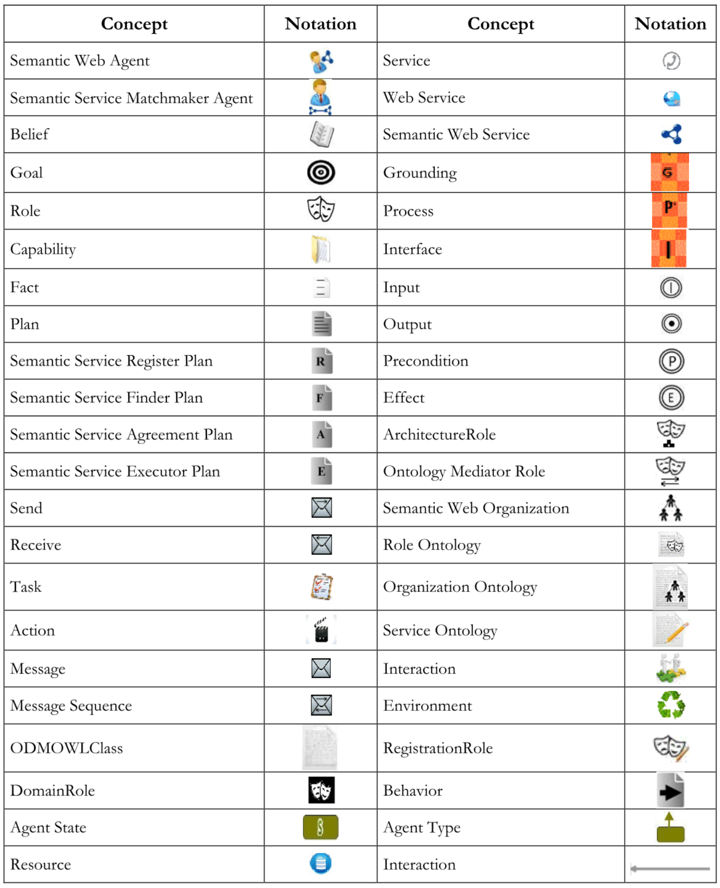
SEA_ML presents users with 43 (44 including the arrows that relate each entity) different semantic constructs. These semantic constructs are presented through a visual notation in 8 different viewpoints. Each concept and its respective notation is presented in Fig. 4.1.

4.2 Applying the principles of "The "Physics" of Notations" in SEA_ML

In this section we will apply each principle of "The "Physics" of Notations" in SEA_ML and understand if there is room for improvement of the visual notation of the language.

4.2.1 Semiotic Clarity

SEA_ML presents a set of symbols that define different semantic constructs. For each viewpoint, different symbols are presented. Each symbol represents a different semantic construct and there i no such case where two symbols represent the same semantic construct. Also, when viewing a certain viewpoint, all symbols that represent that viewpoint are presented to the users. There is no such case where a symbol is not connected to a



Concept	Notation	Concept	Notation
Semantic Web Agent		Service	
Semantic Service Matchmaker Agent		Web Service	
Belief		Semantic Web Service	
Goal		Grounding	G
Role		Process	P
Capability		Interface	I
Fact		Input	I
Plan		Output	
Semantic Service Register Plan	R	Precondition	P
Semantic Service Finder Plan	F	Effect	E
Semantic Service Agreement Plan	A	ArchitectureRole	
Semantic Service Executor Plan	E	Ontology Mediator Role	
Send		Semantic Web Organization	
Receive		Role Ontology	
Task		Organization Ontology	
Action		Service Ontology	
Message		Interaction	
Message Sequence		Environment	
ODMOWLClass		RegistrationRole	
DomainRole		Behavior	
Agent State		Agent Type	
Resource		Interaction	

Figure 4.1: SEA_ML current visual notation

semantic construct (all semantic constructs have well-defined symbols for that construction).

4.2.2 Perceptual Discriminability

The distance between each visual syntax is predefined by the language editor when generating the tool, with the distance between each symbol not greater than one symbol (by default), resulting in a short space between each representation. For the primacy of shape, we can relate that similar semantic constructs have similar symbols. Each symbol has a unique attribute that distinguishes it from others, although their differences may be short to understand the intent of that visual notation. Finally, some symbols are only differentiated by some label (defined by the user) or letter, which is proven to be cognitively ineffective [Moo09].

4.2.3 Semantic Transparency

SEA_ML presents the user with 43 (44 including the arrows that relate each entity) different visual notations.

In Fig. 4.2 we present 19 visual notations of SEA_ML that we think they have room for improvement on this principle. A description of what can be improved is provided for each symbol.

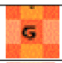
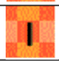
Concept	Notation	Description
Capability		A folder with several files has no direct relationship with a capability.
Grounding		Similar to "Process" and "Interface", with no direct relation between its semantic construct and the visual notation presented.
Process		Similar to "Grounding" and "Interface", with no direct relation between its semantic construct and the visual notation presented.
Interface		Similar to "Grounding" and "Process", with no direct relation between its semantic construct and the visual notation presented.
Precondition		Similar to the "Effect" symbol. There is no relationship between the icon and the semantic construct presented.
Effect		Similar to the "Precondition" symbol. There is no relationship between the icon and the semantic construct presented.
Fact		There is no relationship with a fact. Similar to the "Plan" icon.
Plan		There is no relationship with a plan. Similar to the "Fact" icon.
Semantic Service Register Plan (SSRP)		Similar to "SSFP", "SSAP" and "SSEP", with no direct relation between its semantic construct and the visual notation presented.
Semantic Service Finder Plan (SSFP)		Similar to "SSRP", "SSAP" and "SSEP", with no direct relation between its semantic construct and the visual notation presented.
Semantic Service Agreement Plan (SSAP)		Similar to "SSRP", "SSFP" and "SSEP", with no direct relation between its semantic construct and the visual notation presented.
Semantic Service Executor Plan (SSEP)		Similar to "SSRP", "SSFP" and "SSAP", with no direct relation between its semantic construct and the visual notation presented.
Send		The difference between "Send" and "Receive" is just a short arrow above the envelope that is hard to view and understand the difference at a first glance.
Receive		The difference between "Send" and "Receive" is just a short arrow above the envelope that is hard to view and understand the difference at a first glance.
ArchitectureRole		The icon below the masks is not clear enough to differentiate it from the "Role" symbol.
Ontology Mediator Role		The icon below the masks is not clear enough to differentiate it from the "Role" symbol.
Semantic Web Organization		The notation doesn't directly relate with the Semantic Web. May infer to a generic organization.
Behavior		There is no direct relation between the symbol and the semantic construct.
Agent Type		There is no direct relation between the symbol and the semantic construct.

Figure 4.2: Icons with room for improvement on the Semantic Transparency principle

4.2.4 Complexity Management

SEA_ML does not have any direct mechanism for dealing with complexity on the view-points that are being developed.

4.2.5 Cognitive Integration

For the Conceptual Integration, *SEA_ML* requires a name for every diagram and label used during the modelling. Every procedure is required to be connected to some entity.

As for the Perceptual Integration, *SEA_ML* presents only the viewpoint that is being edited (one at a time), not having any direct relation with each viewpoint that can easily correlate the system being modelled.

4.2.6 Visual Expressiveness

SEA_ML presents similar colors and symbols to similar semantic constructs, although in some cases not the best metaphors. Some of the semantic figures are only differentiated by a letter, which is not the best way to do it since the icons should all be different enough in order to the user to automatically distinguish each semantic construct only looking to its visual notation. Alongside with these details, SEA_ML also has some symbols bigger than others that may induce users to think that those constructions are more important than others. All symbols of SEA_ML should have the same size due to the fact that every semantic construct is important when building a viewpoint on SEA_ML.

SEA_ML presents the user with 43 (44 including the arrows that relate each entity) different visual notations, being a language with a high level of visual expressiveness.

4.2.7 Dual Coding

SEA_ML has eleven visual notations that are only differentiated through letters or short differences that are difficult to see, which are impossible to differentiate without it. These icons can be found on Fig. 4.3.

Analyzing each of the eleven icons, we can conclude that:

1. **Grounding** — The background is similar with the "Process" and "Interface" visual notations, being differentiated only by the letter "P";

2. **Process** — The background is similar with the "Grounding" and "Interface" visual notations, being differentiated only by the letter "P";

3. **Interface** — The background is similar with the "Process" and "Grounding" visual notations, being differentiated only by the letter "G";

4. **Precondition** — The double circled icon is only differentiated by the letter "P" and is similar to the "Effect" visual notation;

5. **Effect** — The double circled icon is only differentiated by the letter "E" and is similar to the "Precondition" visual notation;

6. **Semantic Service Register Plan (SSRP)** — The paper sheet icon is similar to the "SSFP", "SSAP" and "SSEP" visual notations, being differentiated only by the letter "R";

7. **Semantic Service Finder Plan (SSFP)** — The paper sheet icon is similar to the "SSRP", "SSAP" and "SSEP" visual notations, being differentiated only by the letter "F";

8. **Semantic Service Agreement Plan (SSAP)** — The paper sheet icon is similar to the "SSRP", "SSFP" and "SSEP" visual notations, being differentiated only by the letter "A";

9. **Semantic Service Executor Plan (SSEP)** — The paper sheet icon is similar to the "SSRP", "SSFP" and "SSAP" visual notations, being differentiated only by the letter "E";

10. **Send** — The envelope symbol is only differentiated by a short arrow, difficult to read, at the top of the visual notation. The envelope is the same from the "Receive" visual notation, being differentiated by the position of the arrow (pointing to different sides);

11. **Receive** — The envelope symbol is only differentiated by a short arrow, difficult to read, at the top of the visual notation. The envelope is the same from the "Send" visual notation, being differentiated by the position of the arrow (pointing to different sides).

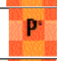
Concept	Notation
Grounding	
Process	
Interface	
Precondition	
Effect	
Semantic Service Register Plan (SSRP)	
Semantic Service Finder Plan (SSFP)	
Semantic Service Agreement Plan (SSAP)	
Semantic Service Executor Plan (SSEP)	
Send	
Receive	

Figure 4.3: Icons with room for improvement on the Dual Coding principle

4.2.8 Graphic Economy

Merging all of the visual notation of SEA_ML, we can verify that it has 43 (44 including arrows that reflect the interaction between items) different visual notations. To ease the process, each set of symbols is divided through different viewpoints (8 in total). On each viewpoint, the user is presented with a palette with every notation he can use on it, limiting the visual notations that users can apply when modifying a viewpoint.

4.2.9 Cognitive Fit

As referred above, SEA_ML presents the users with 43 (44 including arrows that reflect the interaction between items) different symbols, divided by 8 different viewpoints (users cannot use some symbols on some viewpoints).

Some of the proposed visual notation can be improved in order to have a better relation with other similar symbols presented on the SEA_ML language, which may turn the language easier to understand and to be worked for novice users.

4.2.10 Synthesis of each principle applied to SEA_ML

Table 4.1 synthesizes the information detailed on each sub-section above. Plus (+) refers that SEA_ML is according the presented principle, while a minus (-) refers that SEA_ML has room for improvement under that principle.

4.2.11 SEA_ML new notation

In this section, we propose some improvements for the SEA_ML visual notation in order to follow "The "Physics" of Notations" principles.

Of the 43 (44 including the arrows that relate each entity), 32 symbols were modified to follow the principles presented above.

Principle	Room for Improvement
Semiotic Clarity	+
Perceptual Discriminability	-
Semantic Transparency	-
Complexity Management	+/-
Cognitive Integration	+
Visual Expressiveness	-
Dual Coding	-
Graphic Economy	+
Cognitive Fit	-
+ OK | - There is room for improvement	

Table 4.1: Feedback of SEA_ML visual notation according to each principle of "The "Physics" of Notations"

Fig.4.4 shows the SEA_ML visual notations that were not modified.These notations reflect correctly its semantic constructs and are according to "The "Physics" of Notations" principles.

In contrast to the referred above, Fig.4.5 shows the SEA_ML visual notations that were modified in order to be according to "The "Physics" of Notations" principles. An explanation for each new notation is defined below:

1. **Goal** — The new notation adds color to the target, making it more appropriate to be selected when using viewpoints that use this semantic construct;

2. **Capability** — The current visual notation may induce users wrong. The new notation reflects that users have a set of capabilities in order to solve their problems;

3. **Fact** — The current notation is similar to other notations present in SEA_ML. The new notation (check mark) reflects something that is correct and concrete;

4. **Plan** — The notation addresses a plan that can be made to reach a goal from X to Y;

5. **Semantic Service Register Plan (SSRP)** — The current notation has four similar symbols, being distinguished through different letters. The new notation adds the "Semantic Web Services" notation and a person registering to a customer's list;

6. **Semantic Service Finder Plan (SSFP)** — The current notation has four similar symbols, being distinguished through different letters. The new notation adds the "Semantic Web Services" notation and a magnifying glass;

7. **Semantic Service Agreement Plan (SSAP)** — The current notation has four similar symbols, being distinguished through different letters. The new notation adds the "Semantic Web Services" notation and a handshake between two people;

8. **Semantic Service Executor Plan (SSEP)** — The current notation has four similar symbols, being distinguished through different letters. The new notation adds the "Semantic Web Services" notation and a "Play"icon;

9. **Send** — It is not clear what the current notation is addressing. The new notation states clearly that the message is going to be sent elsewhere;

10. **Receive** — It is not clear what the current notation is addressing. The new notation states clearly that the message is going to be received;

11. **Action** — Removed the round border. The clapperboard is enough to understand the semantic construct;

12. **Message** — The new notation attempts to be similar to the new notations adopted in "Message Sequence", "Send" and "Receive";

13. **Message Sequence** — Similar to the notations presented in "Send" and "Receive", the new notation hints a sequence of message being transmitted by those parties;
14. **ODMOWLClass** — The new notation is similar to the previous "Plan" symbol. It tries to remove two similar from the visual notation (as the "Plan" symbol is totally different from the original one);
15. **DomainRole** — The current visual notation does not have any relation with a domain. The metaphor tried on the new notation tries to reflect the web domains, inserting its roles on a web browser window;
16. **Agent State** — The current visual notation does not have any relation with an Agent State. The new notation attempts to add a "Secret Agent" to a typical rounded "State Icon" that appears on some loading screens;
17. **Resource** — The new notation reflects a box full of resources, which reflects more what the semantic construct is;
18. **Web Service** — The new notation adds a gear to a normal icon that relates to the web;
19. **Grounding** — Proposed by the Ege SER-Lab Group;
20. **Process** — Proposed by the Ege SER-Lab Group;
21. **Interface** — Proposed by the Ege SER-Lab Group;
22. **Precondition** — Proposed by the Ege SER-Lab Group;
23. **Effect** — The current visual notation does not have any direct relation with an "Effect". The new notation tries to adapt the "Magic" metaphor for an effect cause;
24. **ArchitectureRole** — The current visual notation does not have any direct relation with an "ArchitectureRole". The new icon adds the "Role" symbol to a common architecture plan;
25. **Ontology Mediator Role** — Proposed by the Ege SER-Lab Group;
26. **Semantic Web Organization** — The current visual notation does not have any direct relation with a Semantic Web organization. The new symbol adds that relation;
27. **Role Ontology** — The new visual notation adapts to the new “ODMOWLClass” proposed above;
28. **Organization Ontology** — The new visual notation adapts to the new “ODMOWLClass” and “SemanticWebOrganization” proposed above;
29. **Service Ontology** — The new visual notation adapts to the new “ODMOWLClass” proposed above;

30. **Interaction** — Although it is perceptible what the current visual notation proposes, there is room for improvement by adding a clearer symbol;

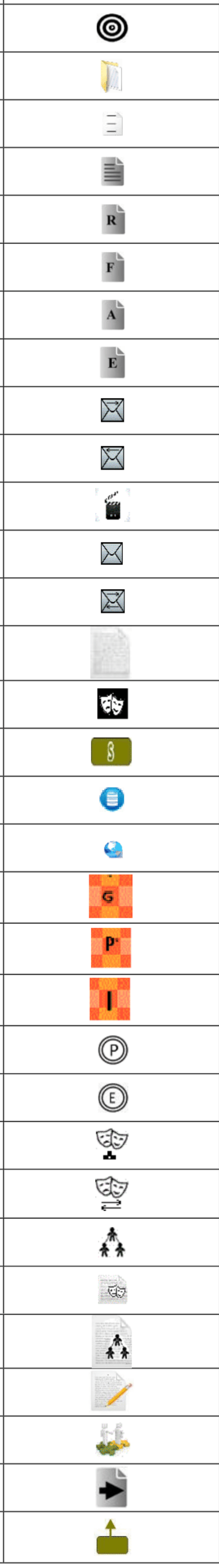
31. **Behavior** — The current visual notation does not have any relation with the "Behavior" semantic construct. The new symbol tries to apply a metaphor related to the human behavior;

32. **Agent Type** — Proposed by the Ege SER-Lab Group.

Concept	Current Notation
Semantic Web Agent	
Semantic Service Matchmaker Agent	
Belief	
Role	
Service	
Semantic Web Service	
Input	
Output	
Environment	
RegistrationRole	
Interation	

Figure 4.4: Visual notations of SEA_ML that were not modified

Concept	Current Notation	Proposed Notation
Goal		
Capability		
Fact		
Plan		
Semantic Service Register Plan		
Semantic Service Finder Plan		
Semantic Service Agreement Plan		
Semantic Service Executor Plan		
Send		
Receive		
Action		
Message		
Message Sequence		
ODMOWLClass		
DomainRole		
Agent State		
Resource		
Web Service		
Grounding		
Process		
Interface		
Precondition		
Effect		
ArchitectureRole		
Ontology Mediator Role		
Semantic Web Organization		
Role Ontology		
Organization Ontology		
Service Ontology		
Interaction		
Behavior		
Agent Type		



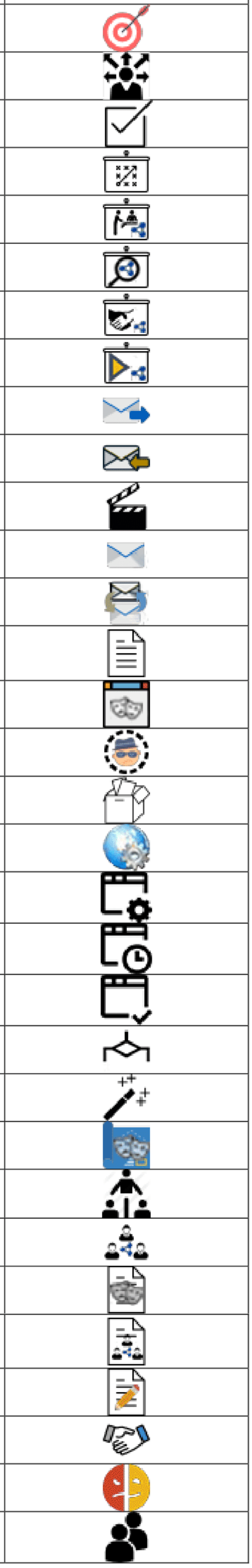


Figure 4.5: New visual notations for certain semantic constructs of SEA_ML

Chapter 5

Selecting a visual notation for SEA_ML

In this chapter we will look in detail to the experiment "Selecting a visual notation for SEA_ML". Some concepts of *SEA_ML* will be presented to a set of users and they will try to connect these concepts to the visual notations presented on chapter 4.

5.1 Introduction

In this experiment users will choose the visual notation they think is more suitable to *SEA_ML* concepts that will be provided to them.

These concepts are *SEA_ML* semantic constructs for which we proposed an alternative visual notation on chapter 4.

The documentation with all the visual notations available to choose from has a mix of the original notation of *SEA_ML* and the proposed new notation. Users may select one or more notations for each concept and can repeat a notation if they think it is more suitable.

The experiment should take around 30 minutes to be completed.

5.2 Objectives

For this experiment we want to understand if the new proposed visual notation is more suitable for SEA_ML than the current notation. In order to do so, a set of participants will be selecting the visual notations they find more useful for a set of semantic constructs of SEA_ML. Using only short descriptions about the concepts that they are dealing with, participants should connect to a visual notation they find more appropriate to.

For each semantic construct that is proposed to be modified we expect that participants select the new visual notation instead of the current notation. The new notations are all according to "The "Physics" of Notations", as discussed on chapter 4.

5.3 Research Questions

In this section we present the Research Questions (RQ from now on) we intend to solve during this experiment:

RQ1: Will participants select the visual notations that correlate with the correct semantic constructs?

RQ2: Does the new visual notation of SEA_ML define better the semantic constructs comparing to the current notation?

5.4 User Profile

For this experiment we intend to understand which symbols are selected by participants from the computer science area that have poor or no knowledge about *DSLs* and *MAS*. These users will select the most convenient symbol they find more adequate based on their previous experiences with other computer science subjects.

For this reason, students of Computer Science with a level degree that ranged from *BSc*, *MSc* or *PhD* were contacted to participate on the experiment, as these users have different levels of experience on the computer science area.

5.5 Experiment Planning

The execution phase (Fig.5.1) is divided into the following steps:

1. — **Letter of Consent** — Participants will read and accept a consent letter regarding the data that will be collected on the experiment. This data will only be used for the purpose of the study and participants remain anonymous;

2. — **Descriptions Questionnaire** — Participants will select the visual notation they find more appropriate for each description of a semantic construct of SEA_ML. Since SEA_ML has 43 (44 including the arrows that relate each entity) different constructions, users are only defining symbols for the constructions that have been modified on chapter 4 (33 different constructions in total).

3. — **Profile Questionnaire** — A questionnaire where participants are asked to insert their gender, age, nationality, field of studies, completed education, current occupation and previous experience with *M.A.S* and *Semantic Web*.

Fig. 5.2 shows the table presented to each user in order to fulfill the descriptions questionnaire (the full questionnaire is presented in appendix D).

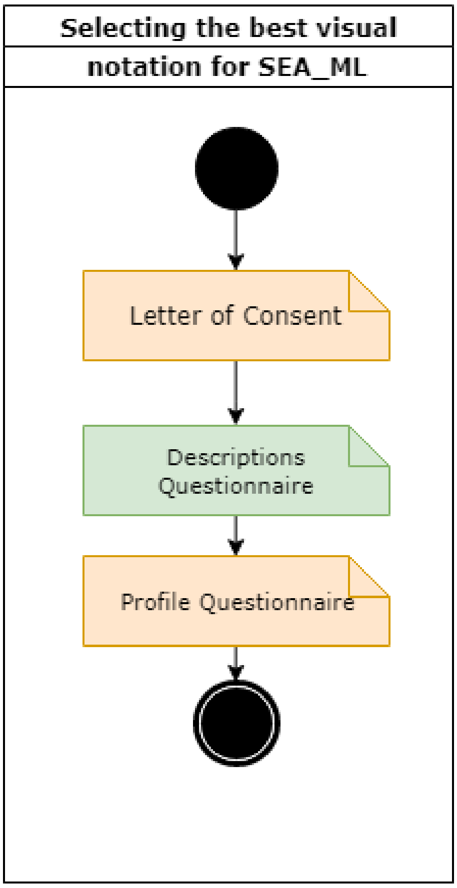


Figure 5.1: Execution plan activity diagram for the experiment "Selecting a visual notation for SEA_ML"

5.6 USE-ME methodology for the experiment

In this section we define the models for the experiment based on the USE-ME methodology as explained on chapter 3. These models synthesize the information of all of the execution phase steps described above.

5.6.1 Context Specification

The Context Specification activity defines the context of use for the *DSL*. Questions such as who will use the *DSL*, where will it be used and how is it expected to be used are added in this activity. This activity engages all stakeholders that are involved in the development of the tool. Domain experts and language engineers are involved on this activity. User profiling helps to decide which users should be engaged in which phases of the evaluation and how their input can be leveraged in the language evolution. Who will use the *DSL* and where it is going to be used will support the evaluator in creating a workflow for the respective testers. Conditions such as the technical and social environment where the experiment will take place are also modeled here.

The context specification presented in Fig.5.3 defines the technical aspects of SEA_ML,

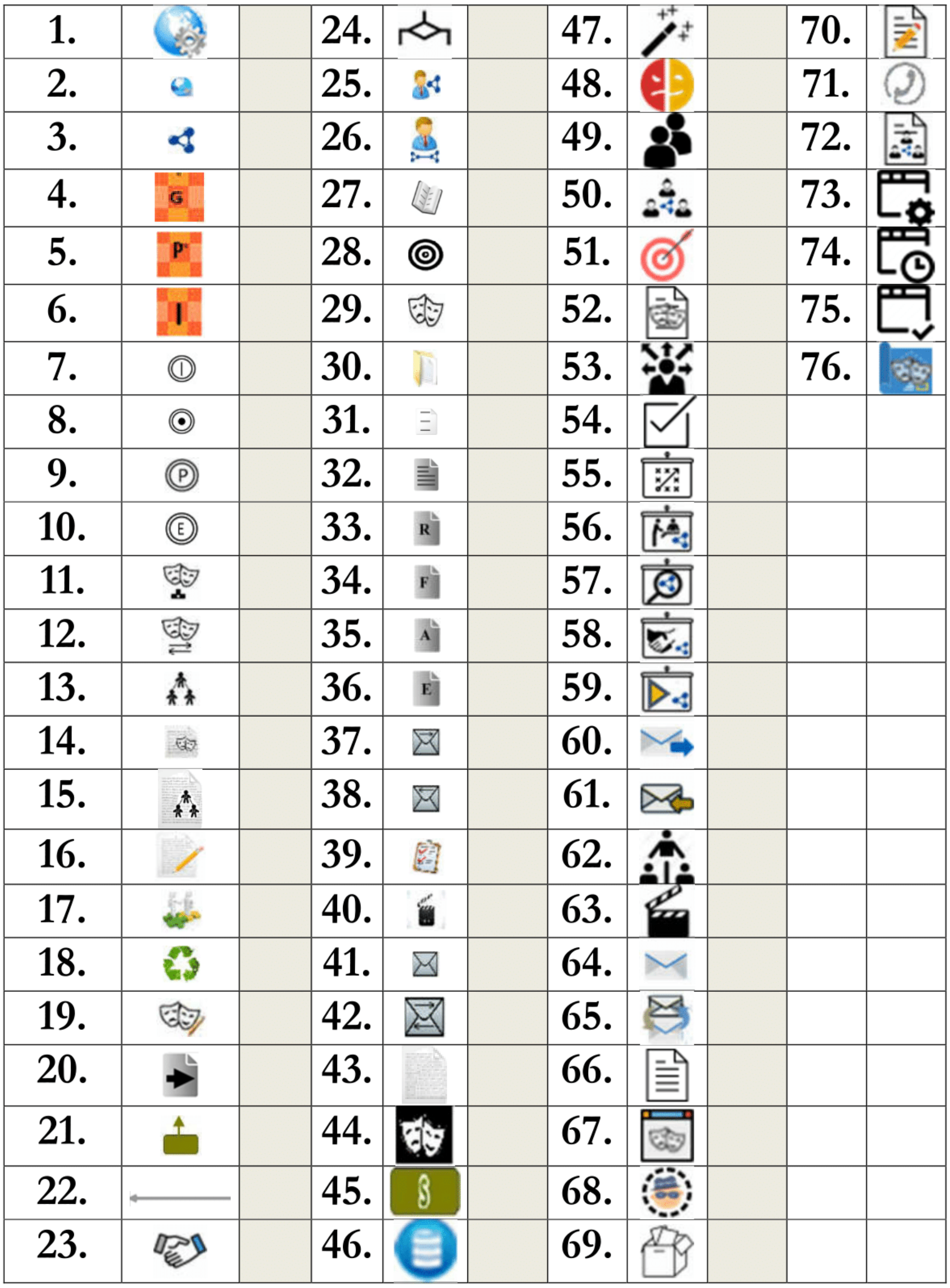


Figure 5.2: Figures presented to the users in order to select the best visual notation for SEA_ML's semantic constructs

defined by the Domain Experts from EGE University. The type of user that uses SEA_ML is defined here, being composed by the Domain Expert, the Expert Evaluator, the Language Engineer, the *DSL* Stakeholder, the Architecture Programmer, the Plan Developer and the End User. Each of these users is defined by a template that defines a set of characteristic for them. As the majority of this specification was managed by the Domain Experts from EGE University, we have added a set of profile templates that are crucial for this experiment: four different End Users from the Computer Science (or similar) area, with a level degree that ranges from BSc, MSc and PhD. Users that have some knowledge from the area are also considered for this experiment. The Environment Specification was also defined by the Domain Experts from EGE University and it illustrates where SEA_ML should run (e.g. computer with a certain type of specification) correctly. The execution plan already defined above is also defined on the Context Specification through the Workflow Specification. All steps that define this experiment are defined on this section.

5.6.2 Goal Specification

The Goal Specification was defined by Domain Experts from the EGE University, as these goals refer to the objectives of the SEA_ML. Although we worked in close cooperation with our colleagues from EGE University, defining the goals for SEA_ML is beyond the scope of this dissertation.

5.6.3 Evaluation Specification

The Evaluation Specification expresses the purpose of the evaluation (objectives to be achieved) in a specific context.

Fig.5.4 presents the Evaluation Specification for the "Selecting a visual notation for SEA_ML" experiment. It defines the goals for this evaluation (the effectiveness and efficiency of the new notation, and the users' satisfaction), the documentation that is provided in order to solve the experiment (list of concepts and figures to be defined for the SEA_ML) and the process of the evaluation (defined on the section "Execution Phase" above).

5.6.4 Interaction Specification

The Interaction Specification expresses the tasks that users are going to manage on the experiment in study.

For this experiment we want to understand which icons are going to be selected by the participants. The icons that are going to be given for the participants are from the current and the new notation of SEA_ML. Participants should correlate the set of descriptions that is given to them with the best visual notations for those constructions. Analyzing the results we want to measure the effectiveness and efficiency of the participants when

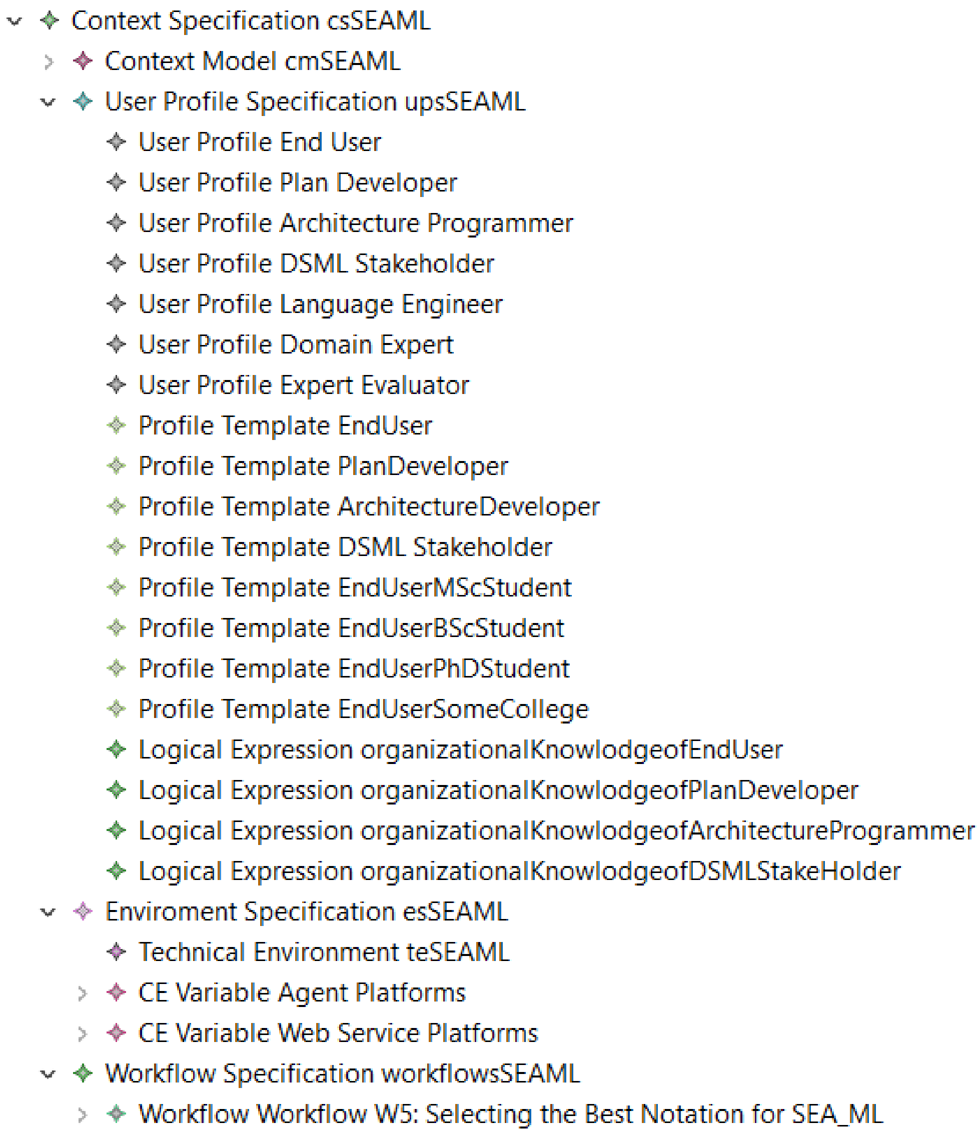


Figure 5.3: Context Specification model using the USE-ME methodology for the "Selecting a visual notation for SEA_ML" experiment.

choosing a set of visual notations for the concepts that are provided to them. Fig.5.5 presents the Interaction Specification for the "Selecting a visual notation for SEA_ML" experiment.

5.6.5 Report Specification

The report specification helps on the construction of the final report for the experiment. The evaluation results are based on the analysis of the result models for the different tasks proposed for the experiment.

For this experiment we want to understand which were the most selected visual notations by the participants and if these visual notations are the most adequate for the semantic constructs they were defined. Fig.5.6 presents the Report Modeling for the

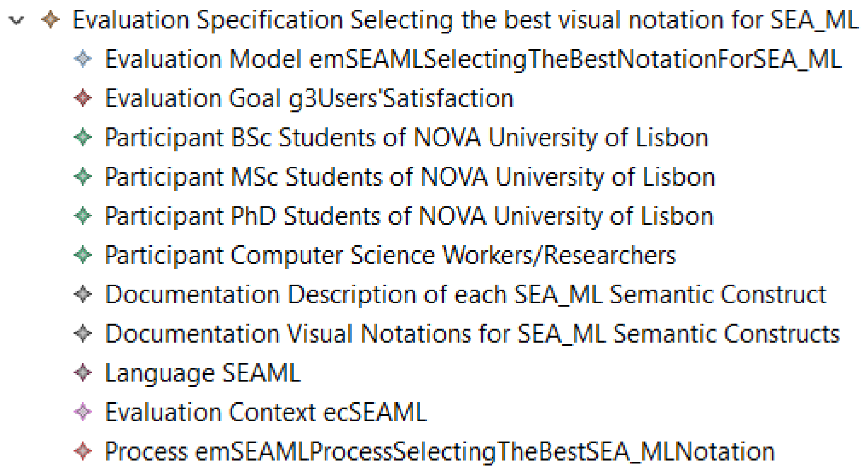


Figure 5.4: Evaluation Specification model using the USE-ME methodology for the "Selecting a visual notation for SEA_ML" experiment.

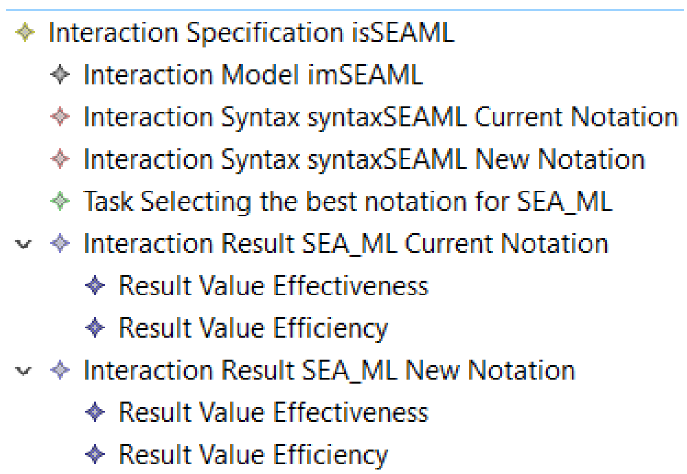


Figure 5.5: Interaction Specification model using the USE-ME methodology for the "Selecting a visual notation for SEA_ML" experiment.

"Selecting a visual notation for SEA_ML" experiment.

5.6.6 Survey Specification

The Survey Specification is similar to the previous but with a set of questions that are important for the experiment.

For this experiment participants are presented to two different questionnaires: a questionnaire regarding the semantic construct where they should select the visual notation they find more suitable for that construction and a profile questionnaire to understand the type of participant that responded to the experiment. For the first questionnaire, participants are proposed to correlate 33 different concepts of SEA_ML to a set of visual notations from the current and the new notation of SEA_ML. Each visual notation has a

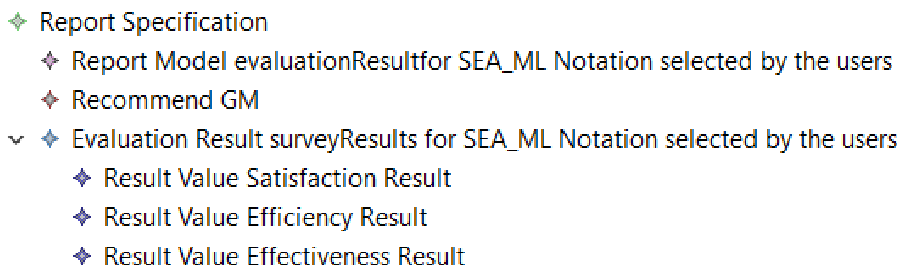


Figure 5.6: Report Specification model using the USE-ME methodology for the "Selecting a visual notation for SEA_ML" experiment.

number assigned for it and the participants should select the number they find more adequate for a specific semantic construct. Participants can select the same visual notation for different semantic constructs. For the profile questionnaire, participants are asked to define their gender, age, nationality, field of studies, completed education, current occupation and their experience with *MAS* and the *Semantic Web*. It is expected that users with more experience on the Computer Science area have results closer to the expected results rather than users with less experience. Fig.5.7 presents the Survey Modeling for the "Selecting a visual notation for SEA_ML" experiment.

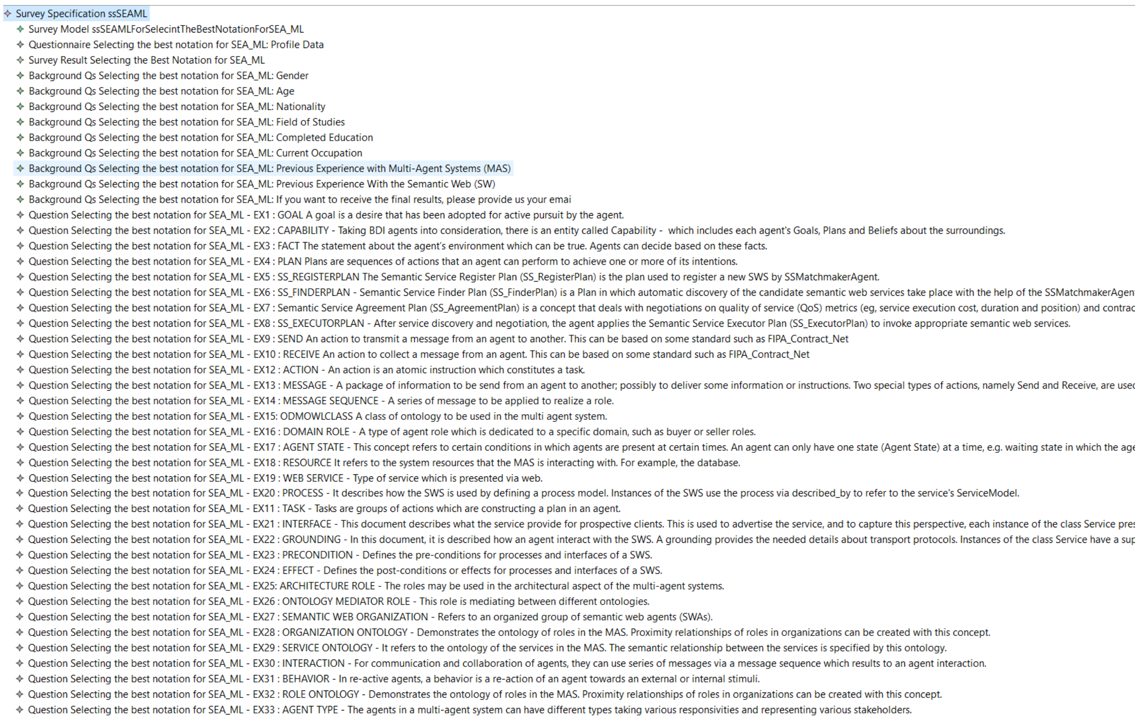


Figure 5.7: Survey Specification model using the USE-ME methodology for the "Selecting a visual notation for SEA_ML" experiment.

5.6.7 Specification Modeling

Fig.5.8 defines the Specification Modeling for the "Selecting a visual notation for SEA_ML" experiment. It defines the syntax and functional goals of SEA_ML, which are defined by the Domain Experts from EGE University and the process model for the experiment of SEA_ML, which it was defined on the specifications above.

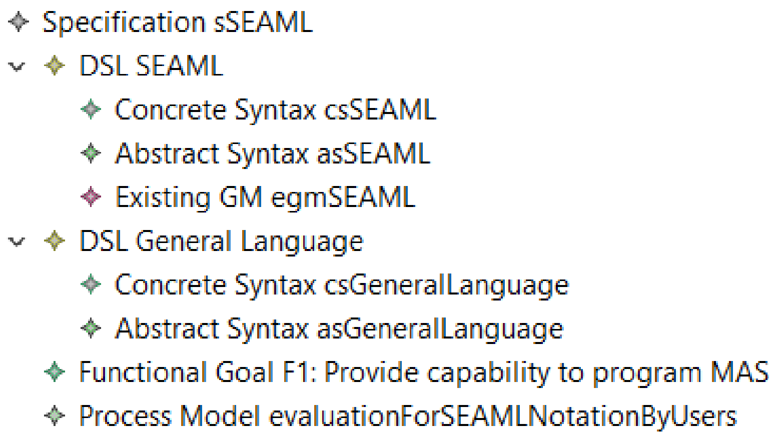


Figure 5.8: Survey Specification model using the USE-ME methodology for the "Selecting a visual notation for SEA_ML" experiment.

5.7 Experiment Results

In this section the results of the "Selecting a visual notation for SEA_ML" experiment will be discussed.

5.7.1 Profile Data

The following bullets synthesize the profile data of users that did this experiment:

1. The experiment was applied to 25 Portuguese users. 80% were male while 20% were female;
2. 60% were Students, 20% were Workers, 12,0% Work while studying and 8,0% are Researchers (Fig.5.9). All of them studied/are studying Computer Science;
3. 84,1% of the users has their age between 22-25, while the rest are between 17-22 and over 26 years (Fig.5.10);
4. 64,0% of the users have completed their BSc, 20,0% completed their MSc and 16,0% only completed High School (Fig.5.11);
5. 36,0% learned about the Semantic Web in the context of a course, 36,0% know what the Semantic Web is but never used it and finally 28,0% never heard of the Semantic Web (Fig.5.12);
6. 44,0% of the users learned about *M.A.S* in the context of a course, while 56,0% never heard of it (Fig.5.13).

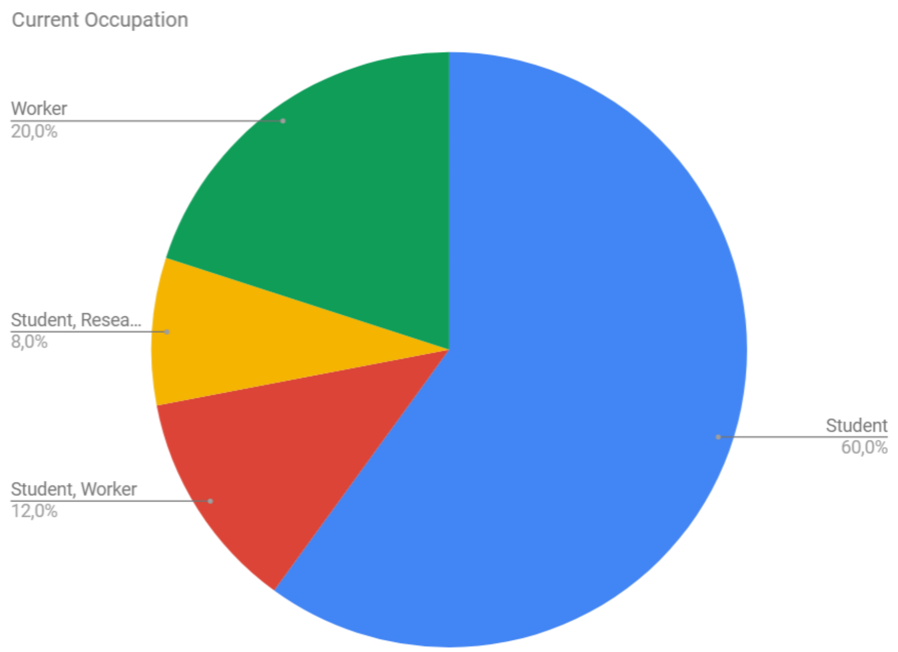


Figure 5.9: Current occupation of the users on "Selecting a visual notation for SEA_ML" experiment

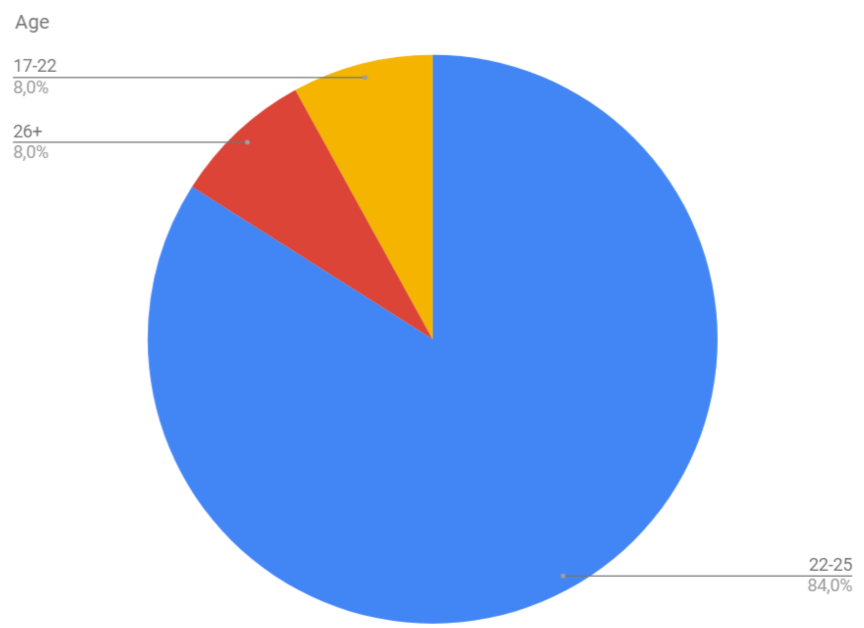


Figure 5.10: Age of the users on the "Selecting a visual notation for SEA_ML" experiment

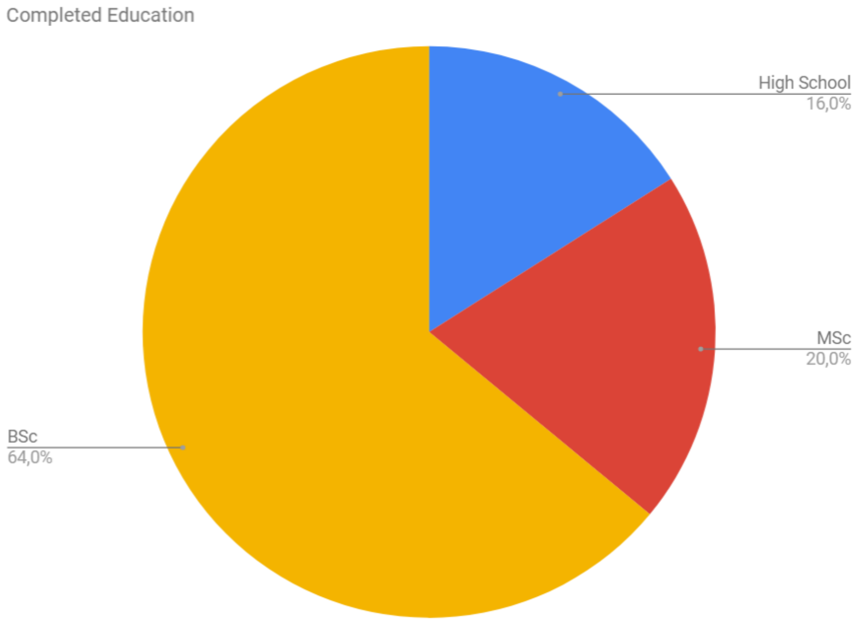


Figure 5.11: Completed education of the users on "Selecting a visual notation for SEA_ML" experiment

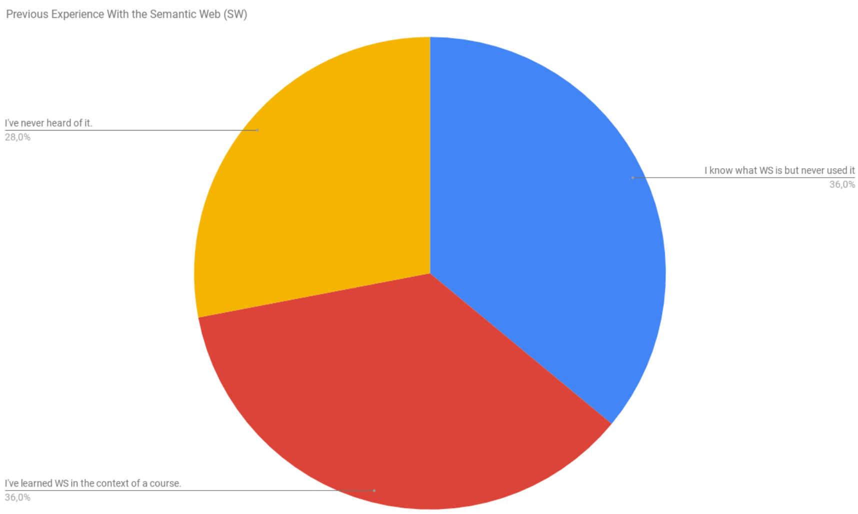


Figure 5.12: Previous experience with Semantic Web of the users on the "Selecting a visual notation for SEA_ML" experiment

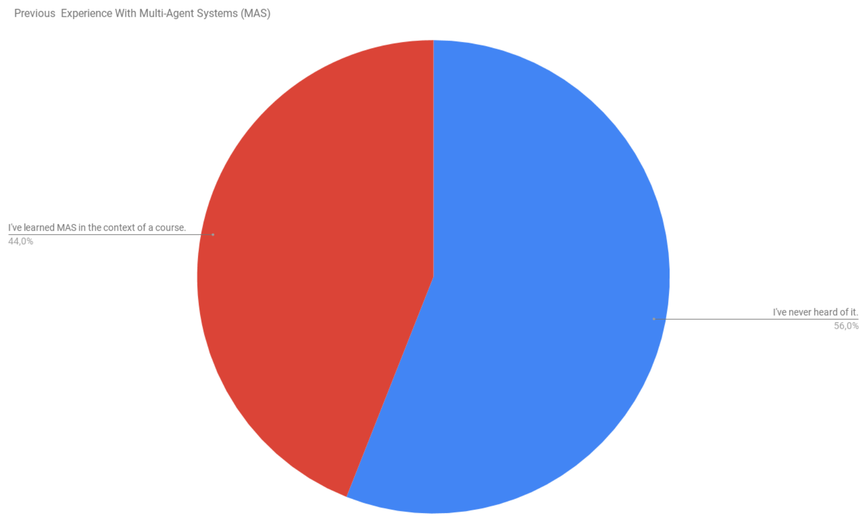


Figure 5.13: Previous Experience with *M.A.S* on the "Selecting a visual notation for SEA_ML" experiment

5.8 Results

In Figures 5.14, 5.15 and 5.16 we present the results for the experiment "Selecting a visual notation for SEA_ML". 33 of the 43 (44 including the arrow that relates each entity) semantic constructs of SEA_ML were proposed to be modified by each participant in order to create the best notation for SEA_ML.

For each semantic construct we present the visual notations that were selected by the participants. The visual notation with the percentage underlined represents the most selected visual notation for that semantic construct, with the remaining percentages (not underlined) representing the other visual notation selected by the participants for that semantic construct.

5.8.1 Discussion of the Results

For this experiment, participants select a visual notation they find more adequate for a certain semantic construct. For each semantic construct, participants have two visual notations that are assigned for that semantic construct, with one of those visual notations being from the current visual notation of SEA_ML and the other from the new proposed notation.

The most selected visual notations proposed by the participants of this experiment has a mix of visual notations of the current and the new proposed notation of SEA_ML. Below we discuss the results for each semantic construct that was proposed to be selected on this experiment:

1. — **Service Ontology** — For this semantic construct users selected 13 different visual notations, with the most selected visual notation gathering a percentage of 24%.

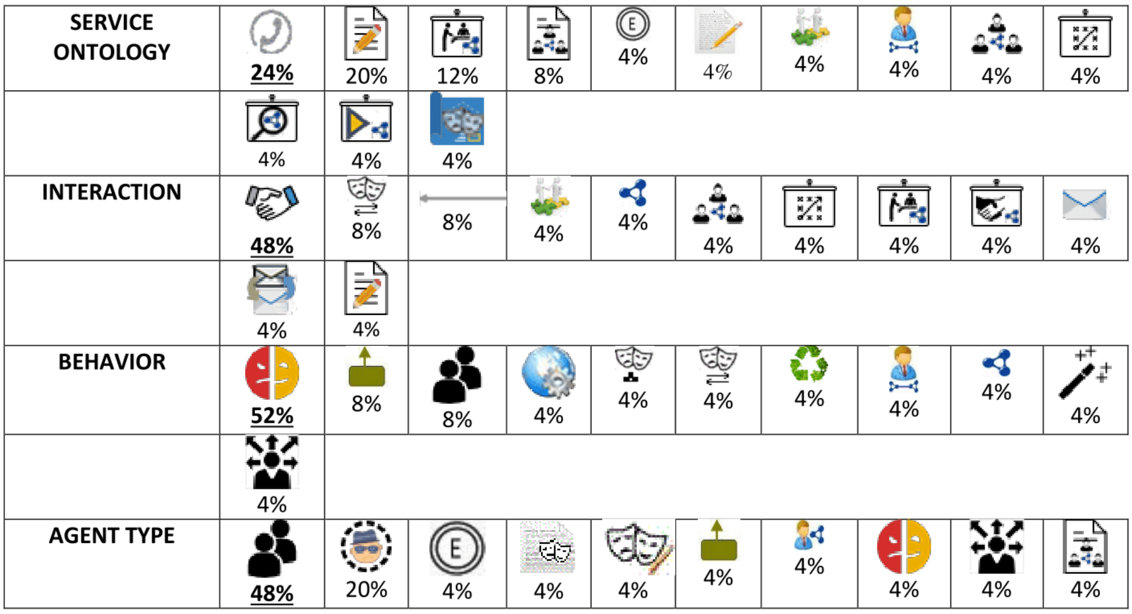


Figure 5.14: Results of the experiment "Selecting a visual notation for SEA_ML" (Part 1/3)

The selected visual notation does not represent the most adequate notation for this semantic construct as it does not correlate with the assigned notation from the current or the new visual notation of SEA_ML. The selected visual notation represents the "Service" semantic construct, not proposed to be modified on this experiment. The visual notations from the current and the new notation of SEA_ML for this semantic construct have a total percentage of 24%, divided on 20% for the new visual notation and 4% for the current visual notation;

2. — **Interaction** — For this semantic construct users selected 12 different visual notations, with the most selected visual notation gathering a percentage of 48%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 4%;

3. — **Behavior** — For this semantic construct users selected 11 different visual notations, with the most selected visual notation gathering a percentage of 52%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. Participants did not select the current visual notation for this semantic construct on this exercise;

4. — **Agent Type** — For this semantic construct users selected 10 different visual notations, with the most selected visual notation gathering a percentage of 48%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 4%;

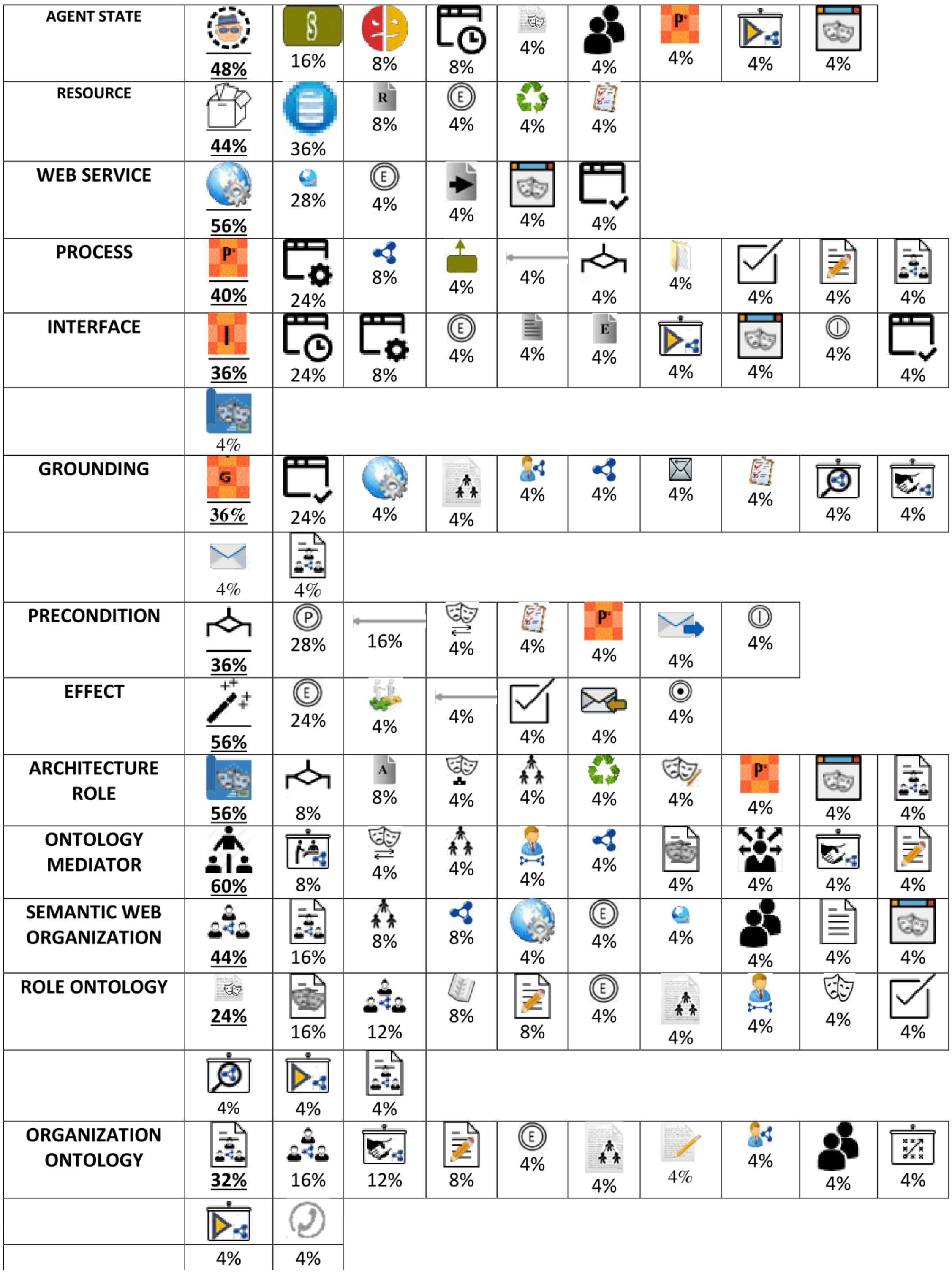


Figure 5.15: Results of the experiment "Selecting a visual notation for SEA_ML" (Part 2/3)

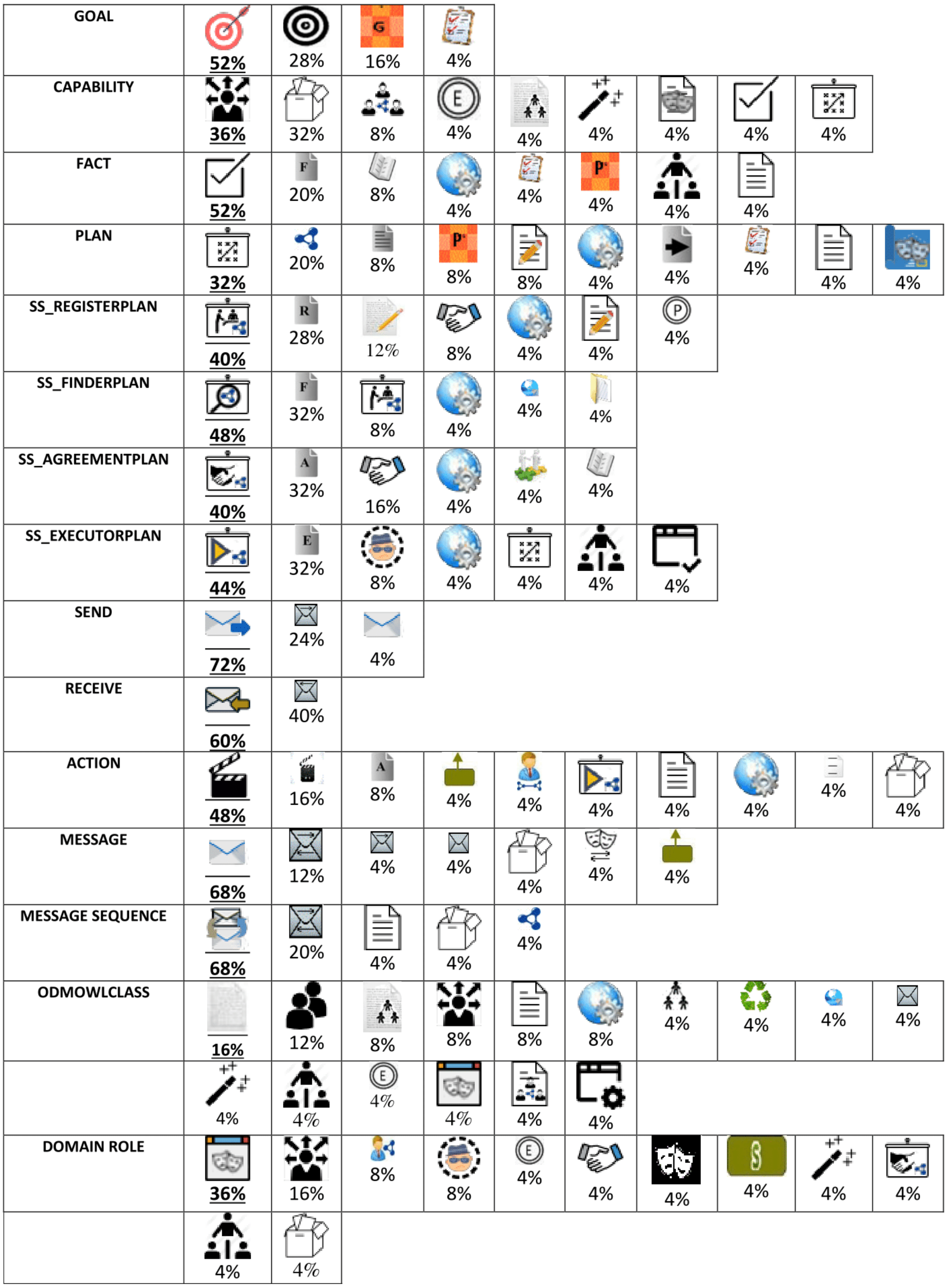


Figure 5.16: Results of the experiment "Selecting a visual notation for SEA_ML" (Part 3/3)

5. — **Agent State** — For this semantic construct users selected 9 different visual notations, with the most selected visual notation gathering a percentage of 48%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 16%;

6. — **Resource** — For this semantic construct users selected 6 different visual notations, with the most selected visual notation gathering a percentage of 44%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 36%, being the second most selected visual notation by the participants;

7. — **Web Service** — For this semantic construct users selected 6 different visual notations, with the most selected visual notation gathering a percentage of 56%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 28%, being the second most selected visual notation by the participants;

8. — **Process** — For this semantic construct users selected 10 different visual notations, with the most selected visual notation gathering a percentage of 40%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the current visual notation. The proposed new visual notation for this semantic construct had a result of 24%, being the second most selected visual notation by the participants;

9. — **Interface** — For this semantic construct users selected 11 different visual notations, with the most selected visual notation gathering a percentage of 36%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the current visual notation. The new proposed visual notation for this semantic construct had a result of 36%, being the second most selected visual notation by the participants;

10. — **Grounding** — For this semantic construct users selected 12 different visual notations, with the most selected visual notation gathering a percentage of 36%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the current visual notation. The new proposed visual notation for this semantic construct had a result of 24%, being the second most selected visual notation by the participants;

11. — **Precondition** — For this semantic construct users selected 8 different visual notations, with the most selected visual notation gathering a percentage of 36%. The selected visual notation represents the most adequate notation for this semantic

construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 28%, being the second most selected visual notation by the participants;

12. — **Effect** — For this semantic construct users selected 7 different visual notations, with the most selected visual notation gathering a percentage of 56%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 24%, being the second most selected visual notation by the participants;

13. — **Architecture Role** — For this semantic construct users selected 10 different visual notations, with the most selected visual notation gathering a percentage of 56%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 28%, being the second most selected visual notation by the participants;

14. — **Ontology Mediator** — For this semantic construct users selected 10 different visual notations, with the most selected visual notation gathering a percentage of 60%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 4%;

15. — **Semantic Web Organization** — For this semantic construct users selected 10 different visual notations, with the most selected visual notation gathering a percentage of 44%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 8%;

16. — **Role Ontology** — For this semantic construct users selected 13 different visual notations, with the most selected visual notation gathering a percentage of 24%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the current visual notation. The new proposed visual notation for this semantic construct had a result of 16%, being the second most selected visual notation by the participants;

17. — **Organization Ontology** — For this semantic construct users selected 12 different visual notations, with the most selected visual notation gathering a percentage of 32%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. Participants did not select the current visual notation for this semantic construct;

18. — **Goal** — For this semantic construct users selected 4 different visual notations, with the most selected visual notation gathering a percentage of 52%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 28%, being the second most selected visual notation by the participants;

19. — **Capability** — For this semantic construct users selected 9 different visual notations, with the most selected visual notation gathering a percentage of 36%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. Participants did not select the current visual notation for this semantic construct;

20. — **Fact** — For this semantic construct users selected 8 different visual notations, with the most selected visual notation gathering a percentage of 52%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. Participants did not select the current visual notation for this semantic construct;

21. — **Plan** — For this semantic construct users selected 10 different visual notations, with the most selected visual notation gathering a percentage of 32%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 8%;

22. — **SS_RegisterPlan** — For this semantic construct users selected 7 different visual notations, with the most selected visual notation gathering a percentage of 40%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 28%, being the second most selected visual notation by the participants;

23. — **SS_FinderPlan** — For this semantic construct users selected 6 different visual notations, with the most selected visual notation gathering a percentage of 48%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 32%, being the second most selected visual notation by the participants;

24. — **SS_AgreementPlan** — For this semantic construct users selected 6 different visual notations, with the most selected visual notation gathering a percentage of 40%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current

visual notation for this semantic construct had a result of 32%, being the second most selected visual notation by the participants;

25. — **SS_ExecutorPlan** — For this semantic construct users selected 7 different visual notations, with the most selected visual notation gathering a percentage of 44%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 32%, being the second most selected visual notation by the participants;

26. — **Send** — For this semantic construct users selected 3 different visual notations, with the most selected visual notation gathering a percentage of 72%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 24%, being the second most selected visual notation by the participants;

27. — **Receive** — For this semantic construct users selected 2 different visual notations, with the most selected visual notation gathering a percentage of 60%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 40%;

28. — **Action** — For this semantic construct users selected 10 different visual notations, with the most selected visual notation gathering a percentage of 48%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 16%, being the second most selected visual notation by the participants;

29. — **Message** — For this semantic construct users selected 7 different visual notations, with the most selected visual notation gathering a percentage of 68%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 4%;

30. — **Message Sequence** — For this semantic construct users selected 5 different visual notations, with the most selected visual notation gathering a percentage of 68%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 20%, being the second most selected visual notation by the participants;

31. — **ODMOWLCLASS** — For this semantic construct users selected 16 different visual notations, with the most selected visual notation gathering a percentage of 16%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the current visual notation. The new proposed visual notation for this semantic construct had a result of 8%;

32. — **Domain Role** — For this semantic construct users selected 12 different visual notations, with the most selected visual notation gathering a percentage of 36%. The selected visual notation represents the most adequate notation for this semantic construct, correlating with the new proposed visual notation. The current visual notation for this semantic construct had a result of 4%.

RQ1: Will participants select the visual notations that correlate with the correct semantic constructs?

Based on the descriptions above, from the 32 visual notations that were modified, participants selected 27 symbols from the new proposed visual notation and 5 from the current visual notation. From the selected visual notations, only one notation did not correlate with any of the two visual notations that were assigned to that semantic construct (the Service Ontology semantic construct). This may due to the fact that the visual notation assigned to it is not intuitive enough for the participants to correlate it to that semantic construct.

The lack of a clearly suitable candidate icon for some of the concepts, combined with the inexperience of the participants with *MAS* made the selection of some of the icons for the visual notation challenging for our participants. This can be shown on, for example, the "ODMOWLCLASS" construct. This semantic construct is defined as a class for building ontologies to be used in *MAS*. The most selected icon for this semantic construct had a 16% of selection by the participants and 16 different icons were selected to be the most suitable visual notation for this construct. This result is poor as it represents that a short percentage of participants selected this icon as the correct one for this semantic construct. The second most selected visual notation for this semantic construct had a percentage of 12%, which is similar to the winning visual notation, where we can conclude that the selected visual notation may not be the most suitable for this semantic construct.

Users find the visual notations that have letters the most suitable for the "Process", "Interface" and "Grounding" semantic constructs, which is directly related to the semantic constructs that are provided on the current version of SEA_ML. These notations were discussed on Chapter 4 as not according to "The "Physics" of Notations", with the possibility of room for improvement.

The remaining semantic constructs have the most selected visual notation with a percentage above 30%, which induces that this visual notations are more intuitive than the remaining selected visual notations.

To conclude, from the 32 visual notations that were proposed to be selected, participants selected 27 symbols from the new visual notation and 5 symbols from the current visual notation. 31 of these visual notations correlate with the correct semantic constructs, proving that the participants selected the most suitable visual notation for almost every semantic construct of this experiment.

RQ2: Does the new visual notation of SEA_ML define better the semantic constructs comparing to the current notation?

Analyzing the gathered results for this experiment, participants selected 27 symbols from the new visual notation and 5 from the current notation. We verify that the icons that were selected from the new notation are directly related to the visual notations proposed on chapter 4, which means that the participants selected the correct new notations for the semantic constructs that they were proposed to define them. For the current visual notations that were selected, 4 correlate with the correct semantic construct, while one of the visual notations is not adequate for the semantic construct that participants defined it to.

The new visual notations that were selected by the participants on this experiment were modified in order to improve the correlation between the visual notation and its semantic construct. To do so, we intended to understand correctly the semantic construct and try to connect it to a metaphor or something that participants would directly relate to (e.g. a semantic construct defines phone calls between users, so a phone would be a correct metaphor for that semantic construct). This would enhance SEA_ML as this *DSL* interaction point is through a visual workbench.

The selected visual notation by the participants mixes symbols from the current and the new visual notation of SEA_ML. Since the participants selected more symbols from the new visual notation and this symbols have been proposed with the objective of enhancing the visual communication of SEA_ML, we can verify that these new visual notations are more user friendly and correlate better with the semantic constructs they are defined to. The remaining visual notations are from the current visual notations of SEA_ML, where we can conclude that these visual notations may not have the better metaphor that is assigned to them, where there can be room for improvement as described on chapter 4.

5.8.2 The new notation for SEA_ML based on the results from the experiment

In this section the new notation based on the selection of the users is presented. It presents the visual notations that were not modified and the solutions proposed by the users. As SEA_ML is a fairly large language (44 different semantic constructs), this section will be divided on the notations that were not included on the experiment and the notations that were included.

The notations that were not modified (11 in 44 notations) are presented on Fig.5.17.

Concept	Current Notation
Semantic Web Agent	
Semantic Service Matchmaker Agent	
Belief	
Role	
Service	
Semantic Web Service	
Input	
Output	
Environment	
RegistrationRole	
Interation	

Figure 5.17: Visual notations of SEA_ML that were not modified

As discussed above, from the 33 notations that were put up to test, 5 are the same as on the original SEA_ML visual notation. From the remaining 28, 27 are relative to the new proposed notation, while 1 refers to the original notation to one notation that was not on the current experience (users have chosen for the "Service Ontology" visual notation the symbol that is currently from the "Service" semantic construct).

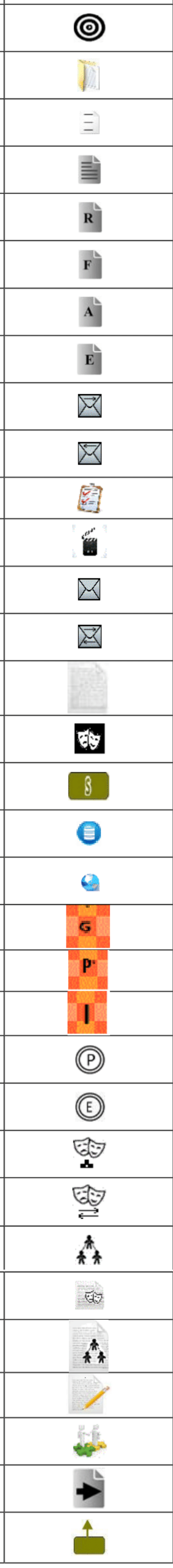
The notations proposed by the users are presented on Fig.5.18.

5.8.3 Threats to Validity

[Woh+12] presents some threats to the validity Experimentation in Software Engineering.

The population selection for this experiment is one of the concerning threats. Users that test the experiment should provide a representative collection of the population. Due

Concept	Current Notation	New Notation
Goal		
Capability		
Fact		
Plan		
Semantic Service Register Plan		
Semantic Service Finder Plan		
Semantic Service Agreement Plan		
Semantic Service Executor Plan		
Send		
Receive		
Task		
Action		
Message		
Message Sequence		
ODMOWLClass		
DomainRole		
Agent State		
Resource		
Web Service		
Grounding		
Process		
Interface		
Precondition		
Effect		
ArchitectureRole		
Ontology Mediator Role		
Semantic Web Organization		
Role Ontology		
Organization Ontology		
Service Ontology		
Interaction		
Behavior		
Agent Type		



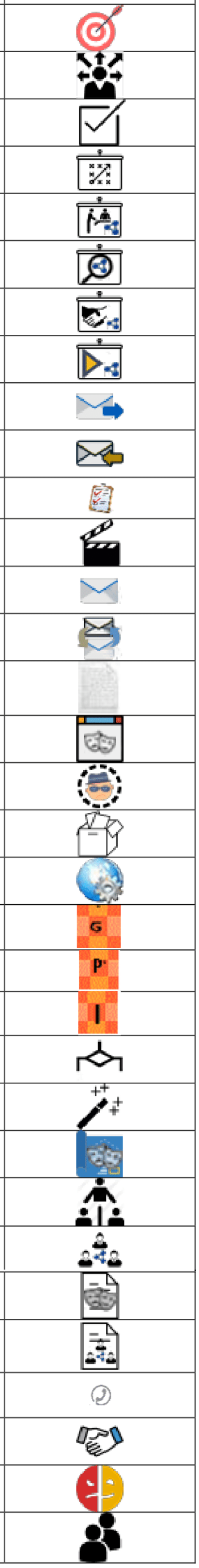


Figure 5.18: Visual notations of SEA_ML proposed by the users on the experiment

to timing and resource constraints, almost all testers of the experiment were students/ex-students of the Computer Science course in NOVA University of Lisbon (only one student is from a different university). The low number of participants on the experiment may reduce the probability of gathering a certain pattern on the results, which means that the conclusions that were discussed above may be different if the population of the experiment was another one completely different. The experience of the participants should also be taken in consideration, as mixing different types of users from the same course can lead to results different than the expected ones. Ideally the experiment should have more users from different colleges and countries, from the same course.

A construction validity to the experiment should also be taken in consideration. The chosen perspective behind this experience may not be representative or the best for the presented scenario. It is being tested that the new visual notation is better than the current notation of SEA_ML, but it is not clear that the visual notations proposed are the more suitable for the semantic constructs of SEA_ML. As the participants of this experiment are only allowed to chose between the current and the new visual notation, participants are restricted to a small sample and cannot add a different visual notation for some of the semantic constructs (if they believe that none of the visual notations proposed is the more suitable for a certain semantic construct). The descriptions for each semantic construct may induce the participants on choosing a different visual notation to that construct, which can lead to a result different than the expected.

The time spent by the participants on the experiment when reading the details of the semantic constructs and comparing them to the visual notations may lead to imprecise data, as participants may not be focused enough when making the whole experience (participants should read 32 different descriptions).

5.9 Summary

In this chapter we presented the experiment "Selecting a visual notation for SEA_ML". The experiment asks the participants to select the best visual notation for 33 semantic constructs of SEA_ML using the original notation and the new proposed notation of SEA_ML based on a set of concepts that were provided. From the 33 semantic constructs that were proposed for modification, 28 notations were proposed for modifications by the participants.

Chapter 6

SEA_ML current notation VS SEA_ML new notation

In this chapter we will look in detail to the experiment "SEA_ML current notation VS SEA_ML new notation". The experiment will be testing 4 viewpoints of *SEA_ML* using the current notation and the new notation that was proposed on chapter 4.

6.1 Introduction

In this experiment participants will test SEA_ML using its current notation and the new notation proposed in this dissertation. The experiment is composed of four exercises, taking place on four different viewpoints of SEA_ML. The selected viewpoints were identified, with the support of the *Domain Experts* from Ege University, as the most important of *SEA_ML*.

These exercises make use of two different case studies: Music Trader and Expert Finding. Each case study has two different exercises. The Music Trader exercise uses the *M.A.S* and *Agent-SWS* viewpoints, while the Expert Finding uses the *Agent Internal* and *Ontology* viewpoints.

Each case study has a different notation (two exercises for each notation). The exercises ask participants to assess if a certain viewpoint is correct according to the text provided to them. If the viewpoint is wrong or incomplete, participants should complete it in order to have it correctly done.

After each exercise, participants answer a questionnaire about the exercise they have made.

6.2 Objectives

For this experiment we want to understand if the new proposed visual notation is more suitable for SEA_ML than the current notation.

In order to do so, a set of participants will be interacting with SEA_ML using the current and the new visual notation on a series of exercises. It is expected that participants find the new notation easier than the current notation, solving the proposed exercises correctly, with less errors, and in less time.

6.3 Research Questions

In this section we present the Research Questions (RQ from now on) we intend to solve during this experiment:

RQ1: Are the results gathered from the experiment better using the current notation or the new notation of SEA_ML?

RQ2: Do participants find it easier to interact and solve problems using the new visual notation of SEA_ML?

6.4 User Profile

For this experiment we intend to evaluate the *Learnability* of *SEA_ML*. Participants that have no background on *DSLs* and *MAS* will need to understand the information that will be provided to them during the experiment and apply it directly at the experiment runtime to the tasks that they have in hands. Participants with some experience with *DSLs* and *MAS* are expected to understand the problems and solve them correctly in a shorter time than the remaining participants.

As such, the participants contacted were from the Computer Science area with a level degree that ranged from *BSc*, *MSc* or *PhD* that had some or no knowledge of *DSLs* and *MAS*.

6.5 Metrics

In order to evaluate correctly the participants performance during the experiment, some metrics are defined:

1. **Time spent with a task** — We want to understand if participants take less time and less effort working on a task using the new visual notation instead of the current notation. In order to do so, we analyze the time the participant needs to answer a certain task using both notations. The time is gathered in minutes and seconds;

2. **Error Rate** — We want to understand if participants make less errors using the new visual notation instead of the current notation of SEA_ML. In order to do

so, we analyze the errors that each participant makes while working on tasks with both notations and assimilate the rate of results on a percentage. This percentage represents the average error rate of the participants on the experiment;

3. **Error Analysis** — We want to understand which errors participants make while developing tasks on the new and the current notation of SEA_ML in order to enhance the language afterwards. To do so, the errors that the participants do on the experiment are considered and afterwards described in order to improve the language;

4. **Success Rate** — We want to understand if participants make more successful actions using the new visual notation instead of the current notation of SEA_ML. In order to do so, we analyze the correct answers that each participant makes while working on tasks with both notations and assimilate the rate of results on a percentage. This percentage represents the success rate of the participants on the experiment;

5. **Participants' Satisfaction** — We want to have a feedback of the participants after interacting with both notations of SEA_ML. In order to do so, a standard questionnaire after each exercise to understand what the exercise was like for the user. An average of the results of the participants for each question is gathered after the experiment.

6.6 Case Studies

Two different case studies are presented to the participants: The "Music Trader" and The "Expert Finding".

1. **Music Trader** — In this case study, participants are requested to develop a system that allows agents to trade their music albums without using any currency. Agents want to trade their music albums for other albums, with this trade being made on an N to N basis (Agent A wants to trade the album A1 for the album B1 from Agent B and vice versa. Agents are not able to trade more than one album for only one album.);

2. **Expert Finding** — In this case study, participants are requested to develop a system that allows agents to find information about other agents that they are searching for in order to communicate with them. Agents have some information about the other agent they are looking for (they are family related or were friends at the past), which is crucial in order to find the correct Semantic Web Service to search the right person. The communication between agents can be made through Social Networks, E-Mail, VoIP or Phone Call. This case study is an adaptation of the case study "Expert Finding" shortly presented in [Cha+16] for an evaluation of SEA_ML.

A full description of each case study (and each exercise) is presented on appendix A.

6.7 Exercises

Each case study presents the participants with 2 different exercises (4 exercises in total).

Each exercise has a description that defines all variables of the system to be modelled. For each exercise an incomplete version of this system is presented. Participants should read the description that is provided to them and compare to the model they have in hands. When the participant thinks the model is according to the description, the exercise is complete, passing to an inquiry about the system they have modeled and afterwards to the next exercise.

Each exercise should take around 10 minutes to be completed. The total experiment should take around 40 minutes.

A short description of each exercise is as follows:

1. **Music Trader: Exercise 1** — In this exercise the participants will be modeling the *M.A.S* viewpoint. An environment and one customer are missing from the original model. Fig.6.1 or Fig.6.2 are presented to the participant for this exercise (depending on the *SEA_ML* version participants are presented to);

2. **Music Trader: Exercise 2** — In this exercise the participants will be modeling the *Agent-SWS* viewpoint. A SS_RegisterPlan is missing from the original model. Fig.6.3 or Fig. 6.4 are presented to the participant for this exercise (depending on the *SEA_ML* version participants are presented to);

3. **Expert Finding: Exercise 1** — In this exercise the participants will be modeling the *Agent Internal* viewpoint. A goal, a belief and a behaviour (and its respective connections) are missing from the original model. Fig. 6.5 or Fig.6.6 are presented to the participant for this exercise (depending on the *SEA_ML* version participants are presented to);

4. **Expert Finding : Exercise 2** — In this exercise the participants will be modeling the *Ontology* viewpoint. A fact and a semantic web organization are missing from the original model. Fig.6.7 or Fig.6.12 are presented to the participant for this exercise (depending on the *SEA_ML* version participants are presented to).

A full description of each exercise is presented on appendix A.

6.8 Possible Solutions

Each exercise has some parts missing, parts that the participants should add to the model they are presented with. The following parts need to be added in order for the exercises to be correctly completed:

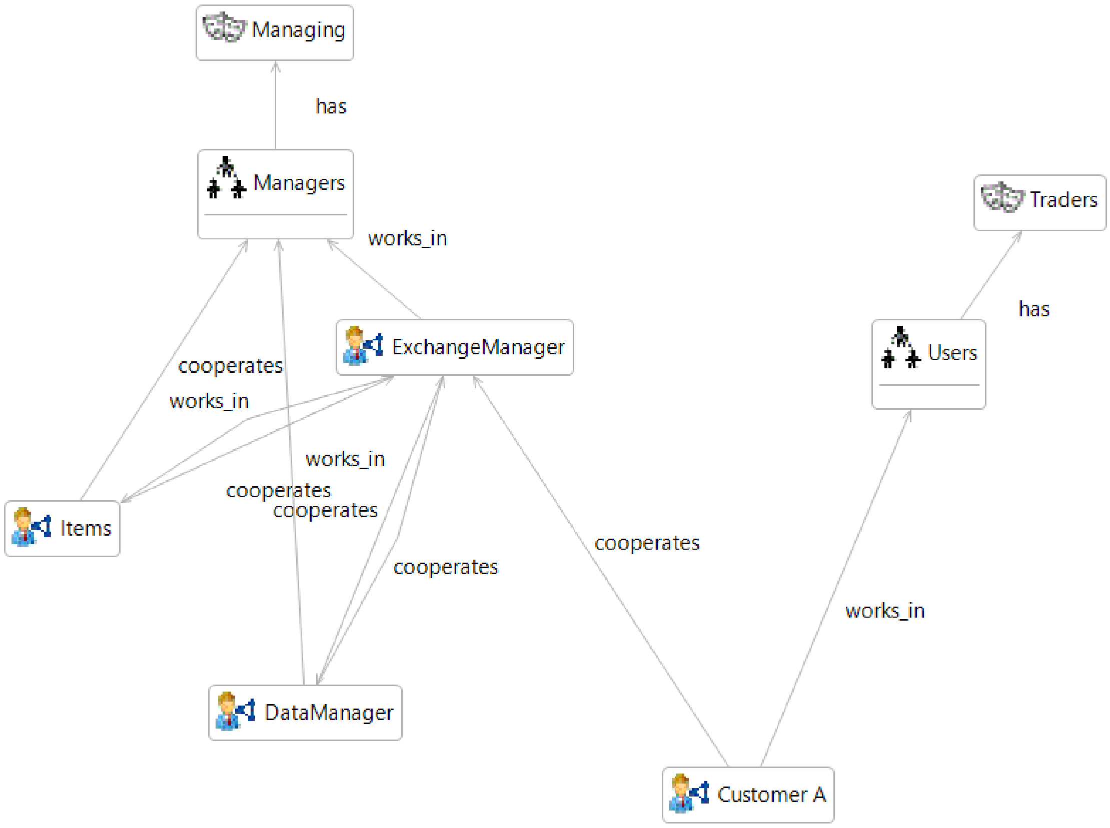


Figure 6.1: Music Trader: Exercise 1 - current notation. Problem for participants to solve with the current notation.

1. **Music Trader: Exercise 1** — An environment and one customer are missing from the original model;
2. **Music Trader: Exercise 2** — a SS_RegisterPlan is missing from the original model;
3. **Expert Finding: Exercise 1** — A goal, a belief and a behaviour (and its respective connections) are missing from the original model;
4. **Expert Finding : Exercise 2** — A fact and a semantic web organization are missing from the original model.

6.9 Experiment Planning

The execution phase (Fig.6.13) is divided on the steps below:

1. — **Letter of Consent** — Participants will read and accept a consent letter regarding the data that will be collected on the experiment. This data will only be used for the purpose of the study and will remain anonymous;

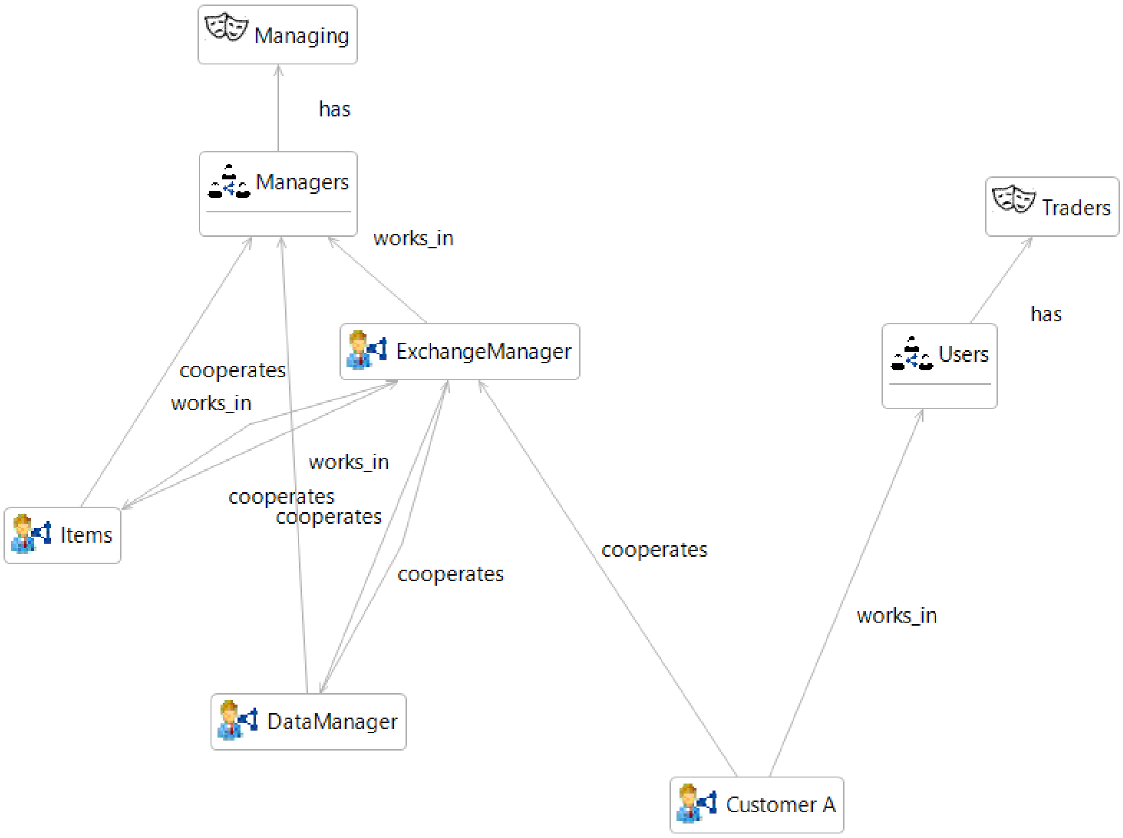


Figure 6.2: Music Trader: Exercise 1 - New Notation. Problem for participants to solve with the new proposed notation.

2. — **Exercise number 1** — Participants start by reading a short description of the system they are going to model. Afterwards, an incomplete model of one of the case studies (Music Trader for version 1 and version 3, Expert Finding for version 2 and version 4) is presented to the user, related to the text participants just read. Participants should complete the model according to the description it was provided to them. The exercise may be from the current notation or from the new notation (version 1 and version 3 start on the new notation, while version 2 and version 4 start on the current notation);

3. — **Questionnaire about exercise number 1** — Participants will be asked how was the experience with the system they just modelled. The first page of the questionnaire is a standard (System Usability Scale - *SUS*) inquiry, while the second one is to understand if participants thought the visual notation was relevant to solve the problem;

4. — **Exercise number 2** — An incomplete model of one of the case studies (Music Trader for version 1 and version 3, Expert Finding for version 2 and version 4) is presented to the participant, related to the system of exercise number 1. Participants should complete the model according to the description it was provided to them;

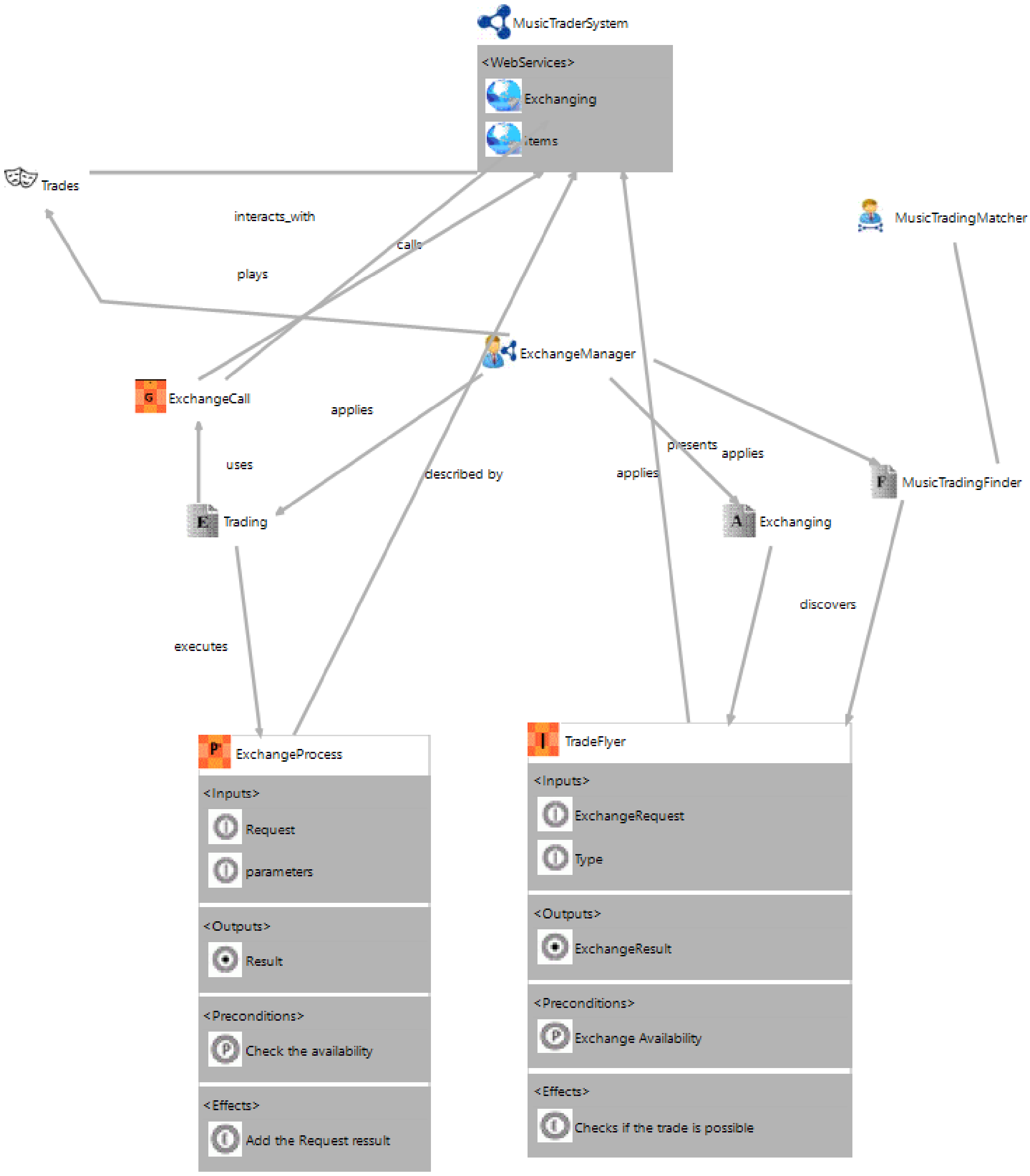


Figure 6.3: Music Trader: Exercise 2 - current notation. Problem for participants to solve with the current notation.

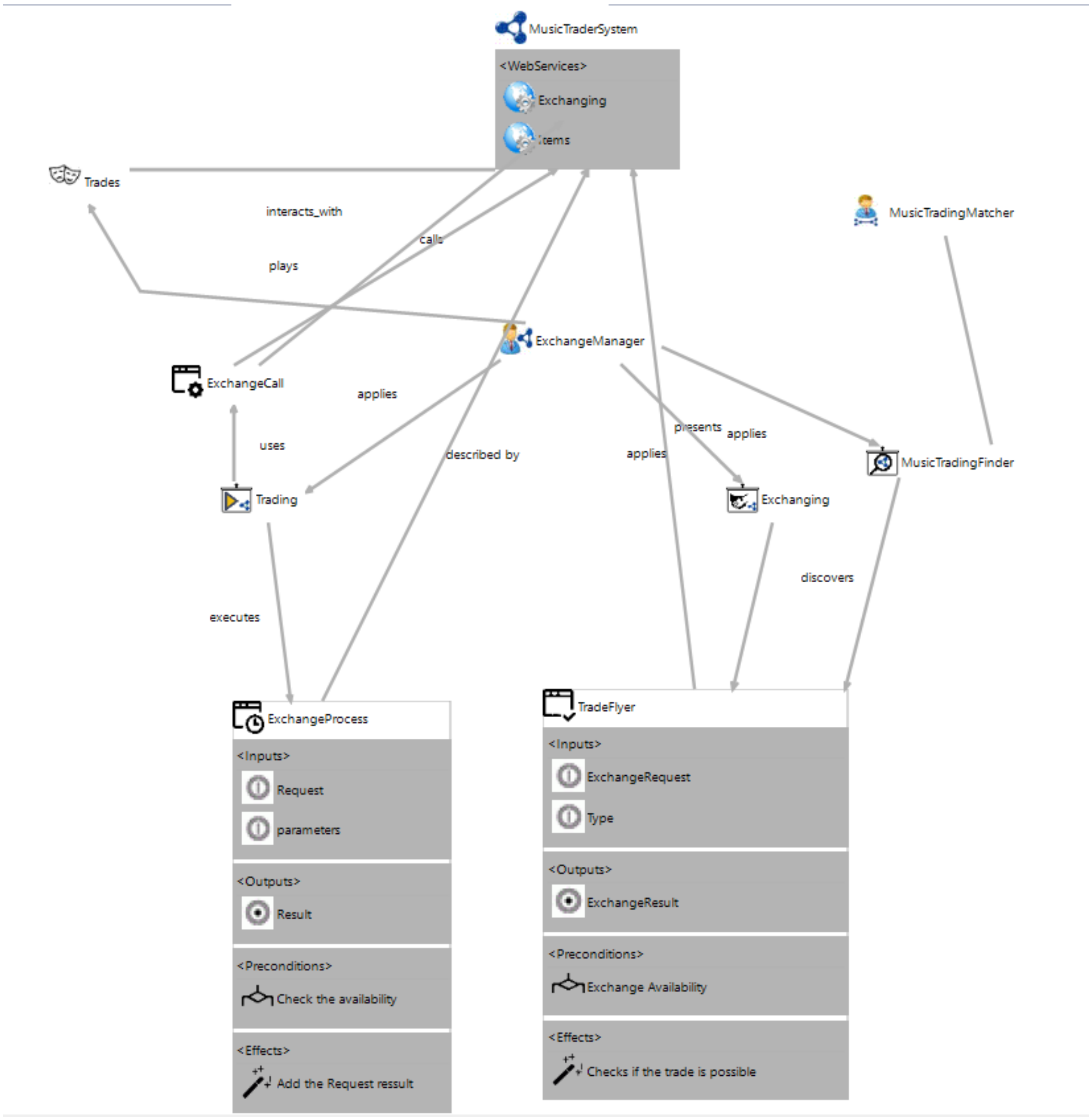


Figure 6.4: Music Trader: Exercise 2 - New Notation. Problem for participants to solve with the new proposed notation.

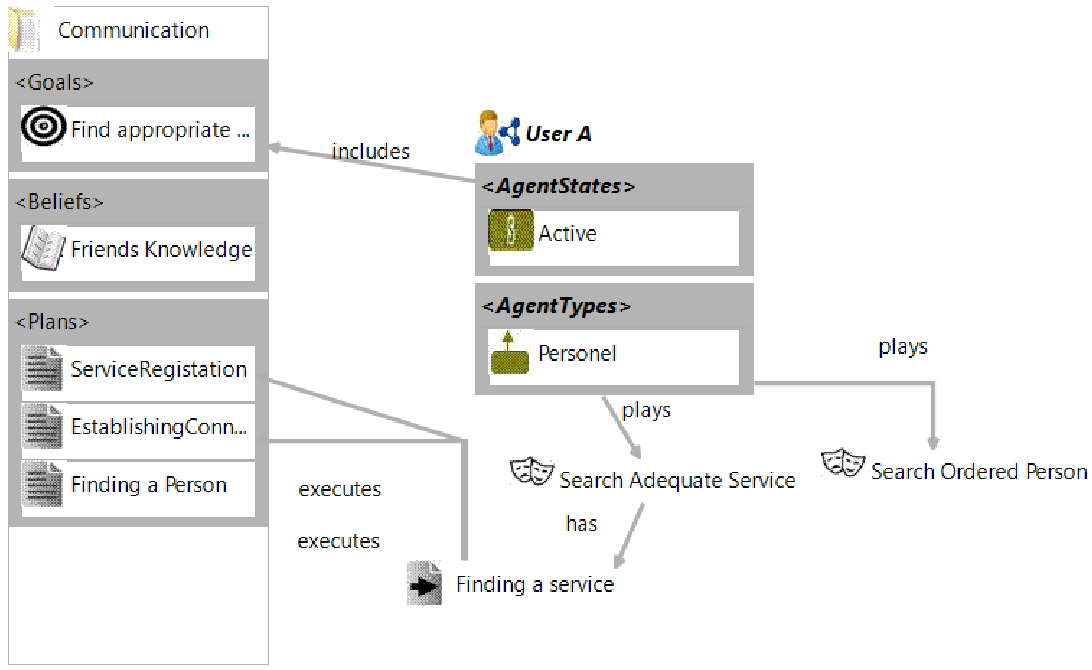


Figure 6.5: Expert Finding: Exercise 1 - current notation. Problem for participants to solve with the current notation.

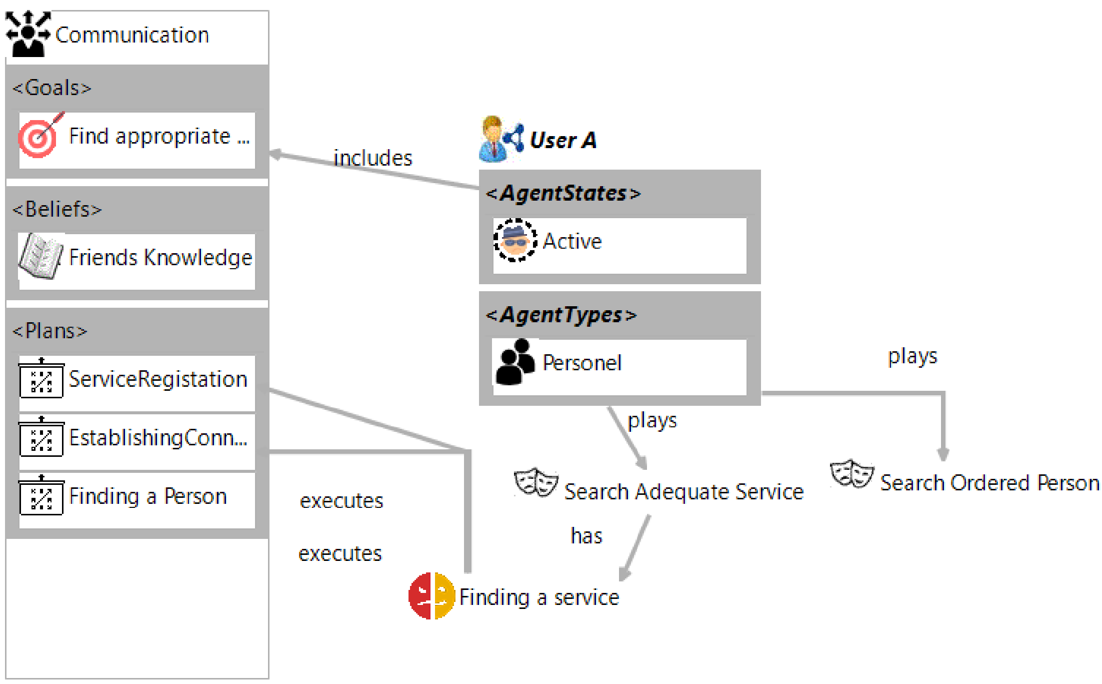


Figure 6.6: Expert Finding: Exercise 2 - New Notation. Problem for participants to solve with the new proposed notation.

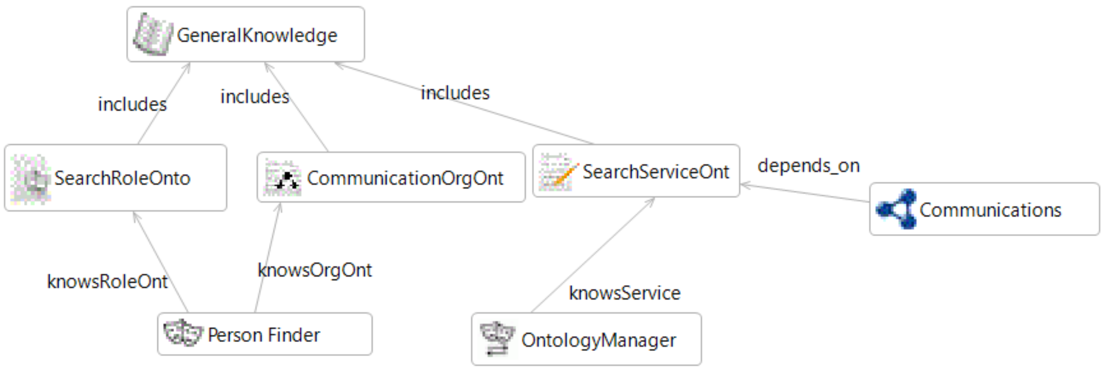


Figure 6.7: Expert Finding: Exercise 2 - current notation. Problem for participants to solve with the current notation.

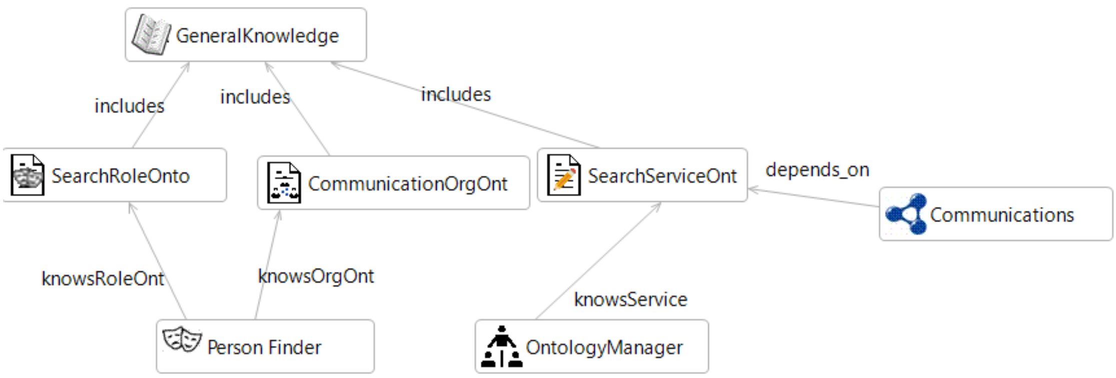


Figure 6.8: Expert Finding: Exercise 2 - New Notation. Problem for participants to solve with the new proposed notation.

5. — **Questionnaire about exercise number 2** — Participants will be asked how was the experience with the system they just modelled. The first page of the questionnaire is a standard (System Usability Scale - *SUS*) inquiry, while the second one is to understand if participants thought the visual notation was relevant to solve the problem;

6. — **Exercise number 3** — Participants start by reading a short description of the system they are going to model. Afterwards, an incomplete model of one of the case studies (Expert Finding for version 1 and version 3, Music Trader for version 2 and version 4) is presented to the user, related to the text participants just read. Participants should complete the model according to the description it was provided to them. The exercise may be from the current notation or from the new notation (version 1 and version 3 now interact with the current notation, while version 2 and version 4 interact with the new notation);

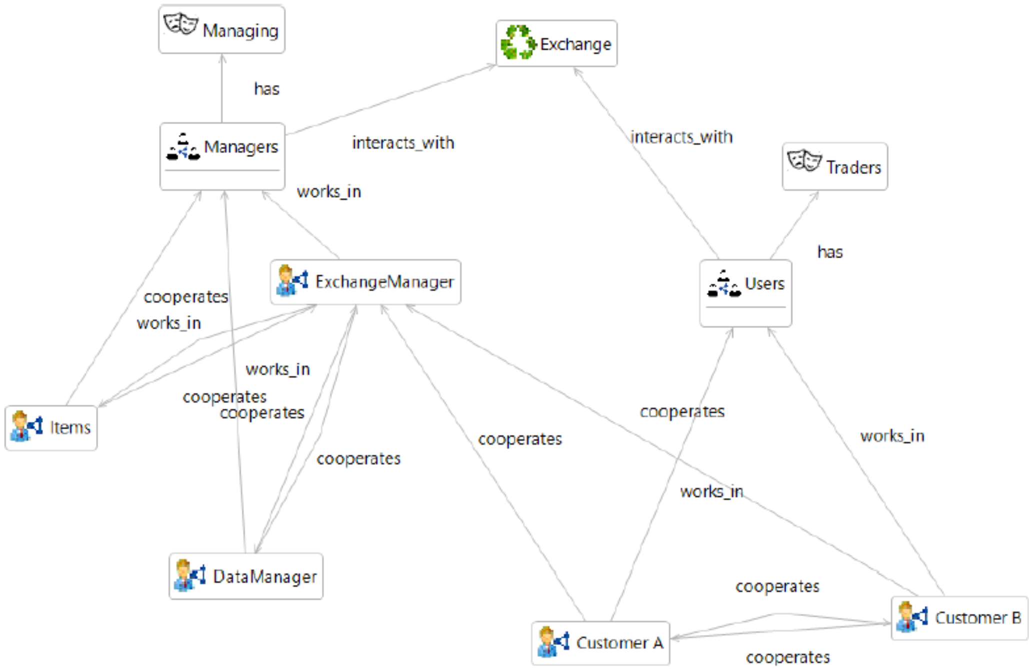


Figure 6.9: Solution for the Exercise 1 of the Music Trader case study.

7. — **Questionnaire about exercise number 3** — Participants will be asked how was the experience with the system they just modelled. The first page of the questionnaire is a standard (System Usability Scale - *SUS*) inquiry, while the second one is to understand if participants thought the visual notation was relevant to solve the problem;

8. — **Exercise number 4** — An incomplete model of one of the case studies (Expert Finding for version 1 and version 3, Music Trader for version 2 and version 4) is presented to the participant, related to the system of exercise number 3. Participants should complete the model according to the description it was provided to them;

9. — **Questionnaire about exercise number 4** — Participants will be asked how was the experience with the system they just modelled. The first page of the questionnaire is a standard (System Usability Scale - *SUS*) inquiry, while the second one is to understand if participants thought the visual notation was relevant to solve the problem;

10. — **Profile Questionnaire** — A questionnaire where participants are asked to insert their gender, age, nationality, field of studies, completed education, current occupation and previous experience with *M.A.S* and *WS*.

The execution phase presented above has four different versions (one of the four versions is presented to each user):

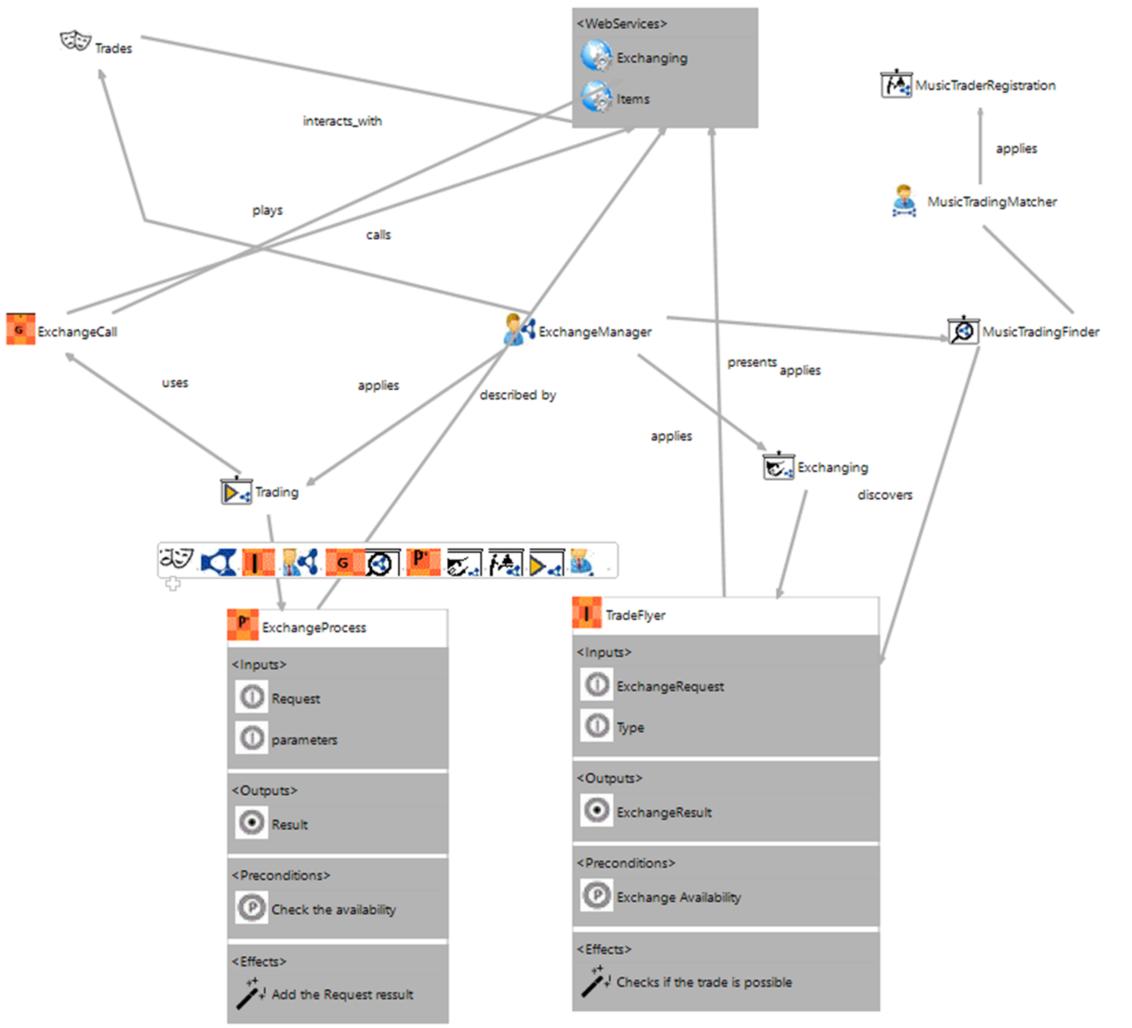


Figure 6.10: Solution for the Exercise 2 of the Music Trader case study.

1. — **Version 1** — Participants start on the case study "Music Trader", finishing on the "Expert Finding" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled with the new notation, while the second case study is modeled with the current notation;

2. — **Version 2** — Participants start on the case study "Expert Finding", finishing on the "Music Trader" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled with the actual notation, while the second case study is modeled with the new notation;

3. — **Version 3** — Participants start on the case study "Music Trader", finishing on the "Expert Finding" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled

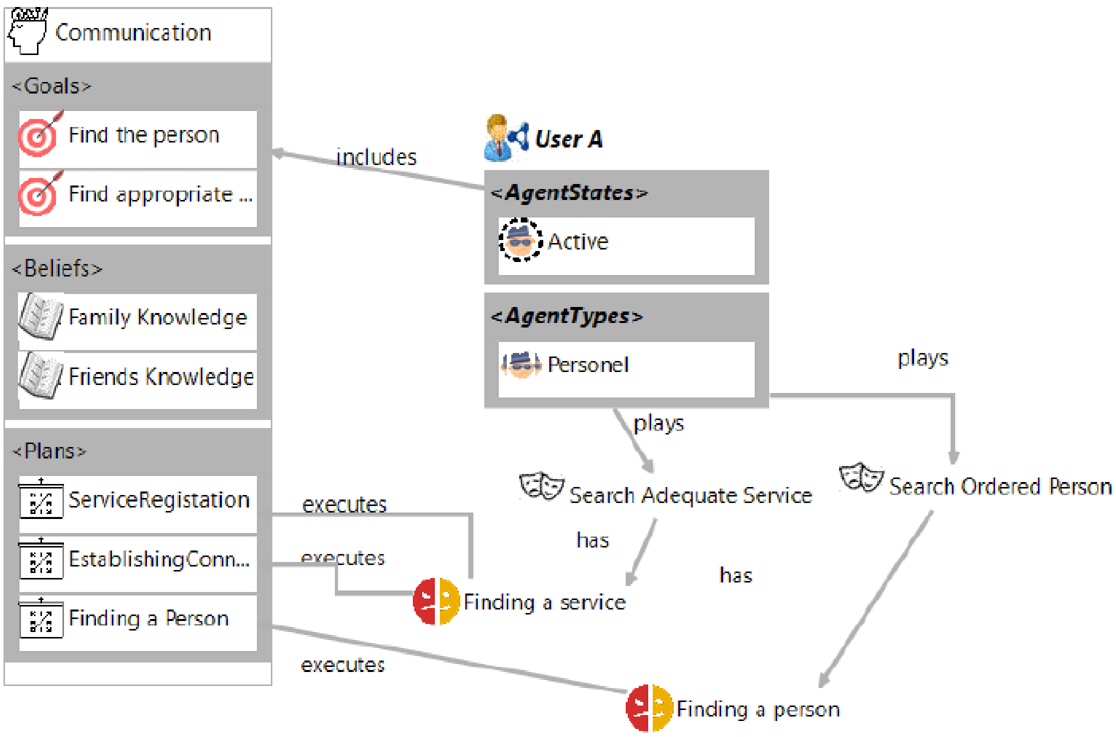


Figure 6.11: Solution for the Exercise 1 of the Expert Finding case study.

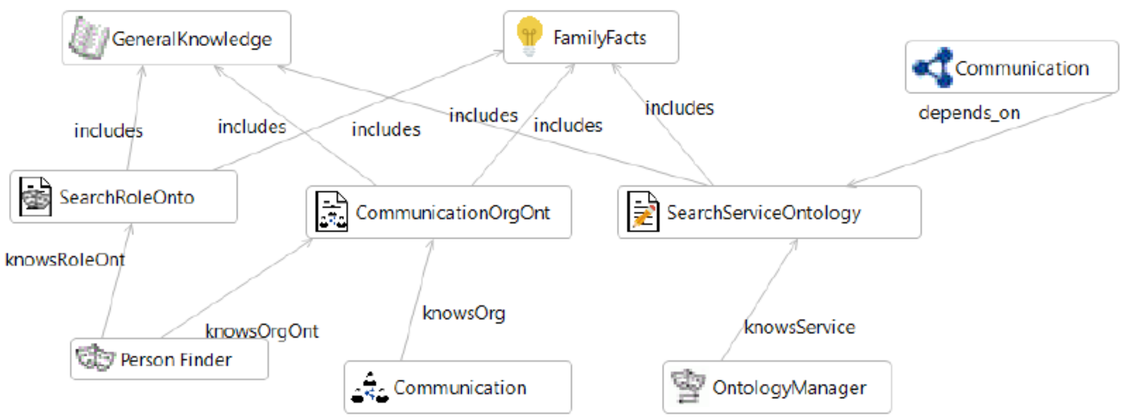


Figure 6.12: Solution for the Exercise 2 of the Expert Finding case study.

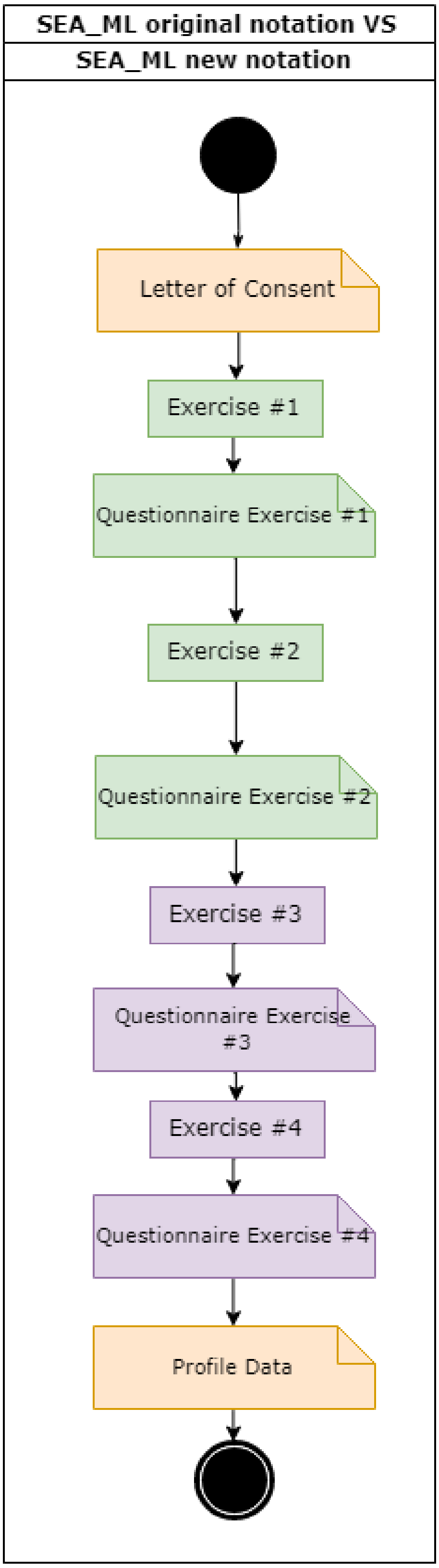


Figure 6.13: Execution plan activity diagram for the experiment "SEA_ML current notation VS SEA_ML new notation"

with the current notation, while the second case study is modeled with the new notation;

4. — **Version 4** — Participants start on the case study "Expert Finding", finishing on the "Music Trader" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled with the new notation, while the second case study is modeled with the current notation;

Fig.6.14 synthesizes the process each user has to do on each version of this experiment based on the information detailed above.

The letter of consent, profile data and inquiries for each version can be found on appendix B and appendix C .

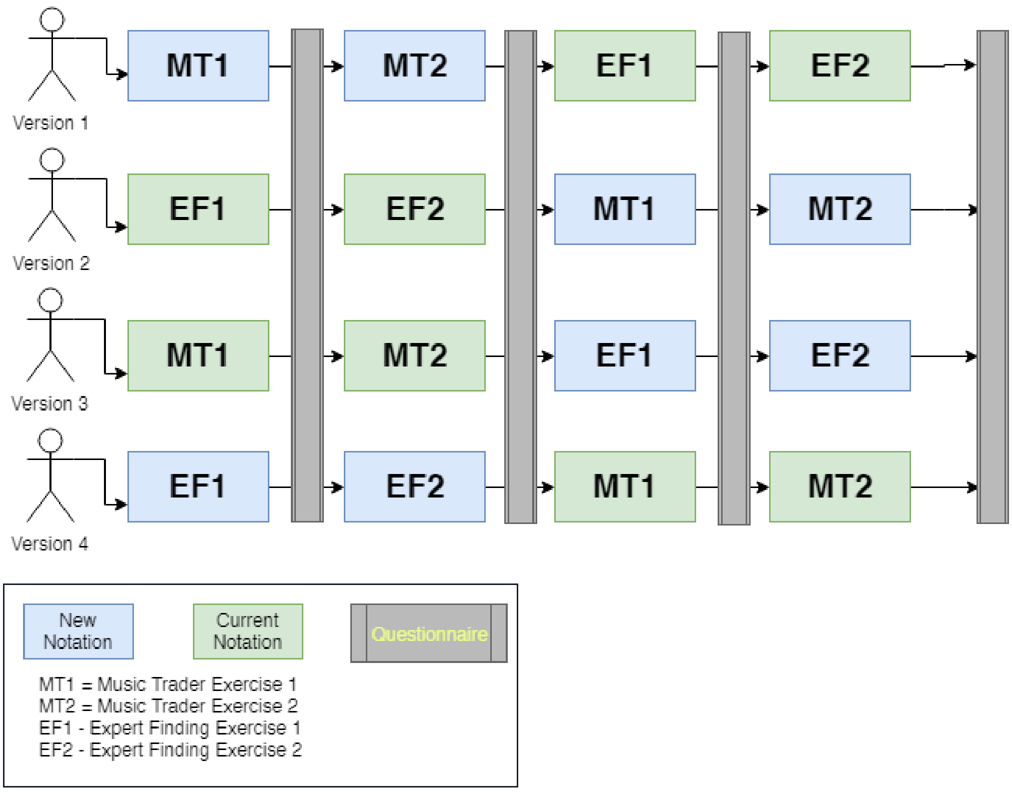


Figure 6.14: Synthesis of each version of the "SEA_ML current notation VS SEA_ML new notation" experiment

6.10 USE-ME methodology for the experiment

In this section we define the models for the experiment based on the USE-ME methodology as explained on chapter 3. These models synthesize the information of all of the execution phase steps described above.

6.10.1 Context Specification

The Context Specification activity defines the context of use for the *DSL*. Questions such as who will use the *DSL*, where will it be used and how is it expected to be used are added in this activity. This activity engages all stakeholders that are involved in the development of the tool. Domain experts and language engineers are involved on this activity. User profiling helps to decide which users should be engaged in which phases of the evaluation and how their input can be leveraged in the language evolution. Who will use the *DSL* and where it is going to be used will support the evaluator in creating a workflow for the respective testers. Conditions such as the technical and social environment where the experiment will take place are also modeled here.

The context specification presented in Fig.6.15 defines the technical aspects of SEA_ML, defined by the Domain Experts from EGE University. The type of user that uses SEA_ML is defined here, being composed by the Domain Expert, the Expert Evaluator, the Language Engineer, the *DSL* Stakeholder, the Architecture Programmer, the Plan Developer and the End User. Each of these users is defined by a template that defines a set of characteristic for them. As the majority of this specification was managed by the Domain Experts from EGE University, we have added a set of profile templates that are crucial for this experiment: four different End Users from the Computer Science (or similar) area, with a level degree that ranges from BSc, MSc and PhD. Users that have some knowledge from the area are also considered for this experiment. The Environment Specification was also defined by the Domain Experts from EGE University and it illustrates where SEA_ML should run (e.g. computer with a certain type of specification) correctly. The execution plan already defined above is also defined on the Context Specification through the Workflow Specification. As we have four different exercises for this experiment, four different workflows have been created, one for each exercise (two for the Music Trader case study and two for the Expert Finding case study).

6.10.2 Goal Specification

The goal modeling was defined by the Domain Experts from EGE University, as these goals refer to the objectives of the SEA_ML, which is beyond the scope of this dissertation.

6.10.3 Evaluation Specification

The Evaluation Specification expresses the purpose of the evaluation (objectives to be achieved) in a specific context.

Fig.6.16 presents the Evaluation Specification for the "SEA_ML current notation VS SEA_ML new notation" experiment. It defines the goals for this evaluation (the effectiveness and efficiency of the new notation, and the participants' satisfaction), the documentation that is provided in order to solve the experiment (case studies and exercises to

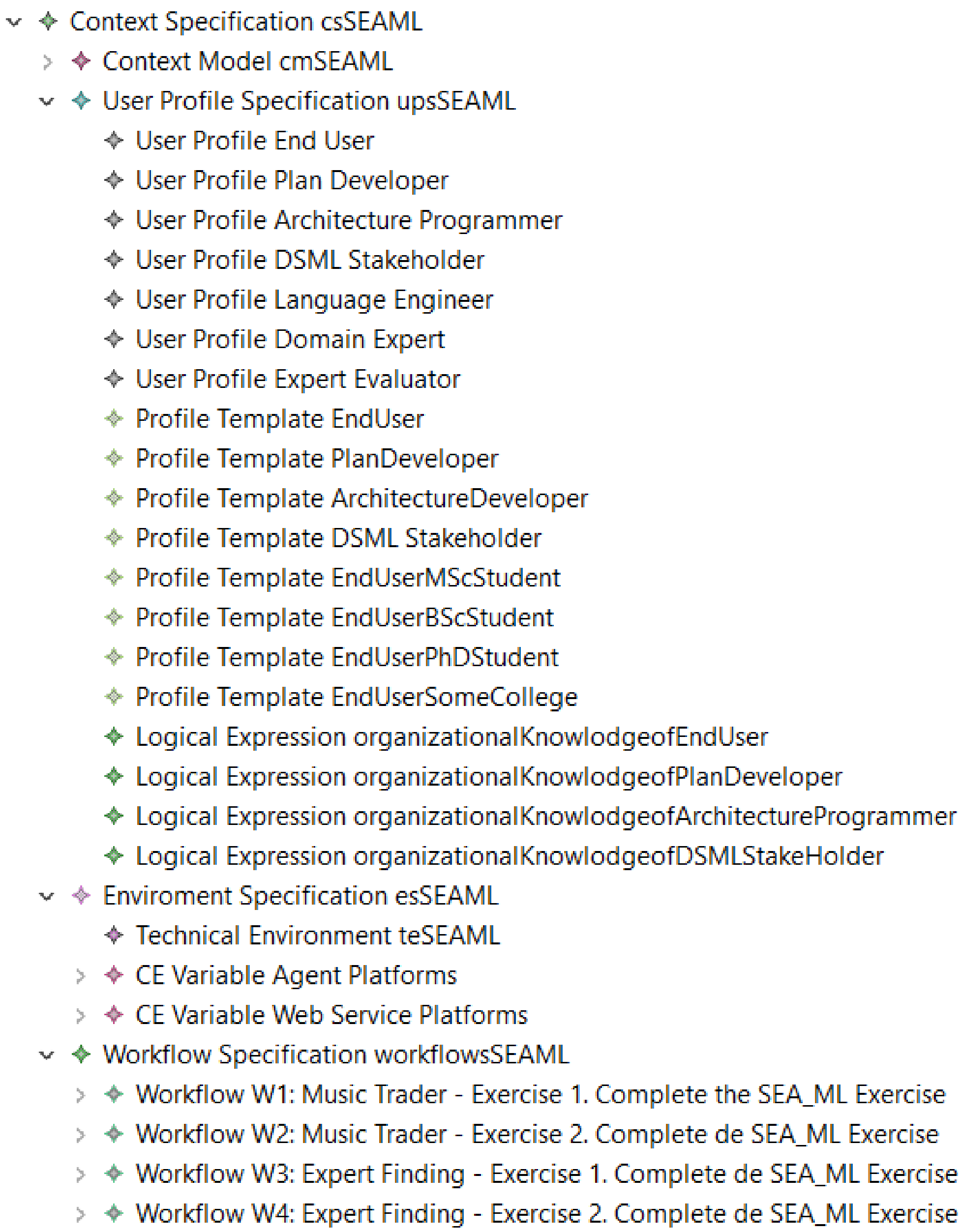


Figure 6.15: Context Specification model using the USE-ME methodology for the "SEA_ML current notation VS SEA_ML new notation" experiment.

solve) and the process of the evaluation for each version of the experiment (defined on the section "Execution Phase" above).

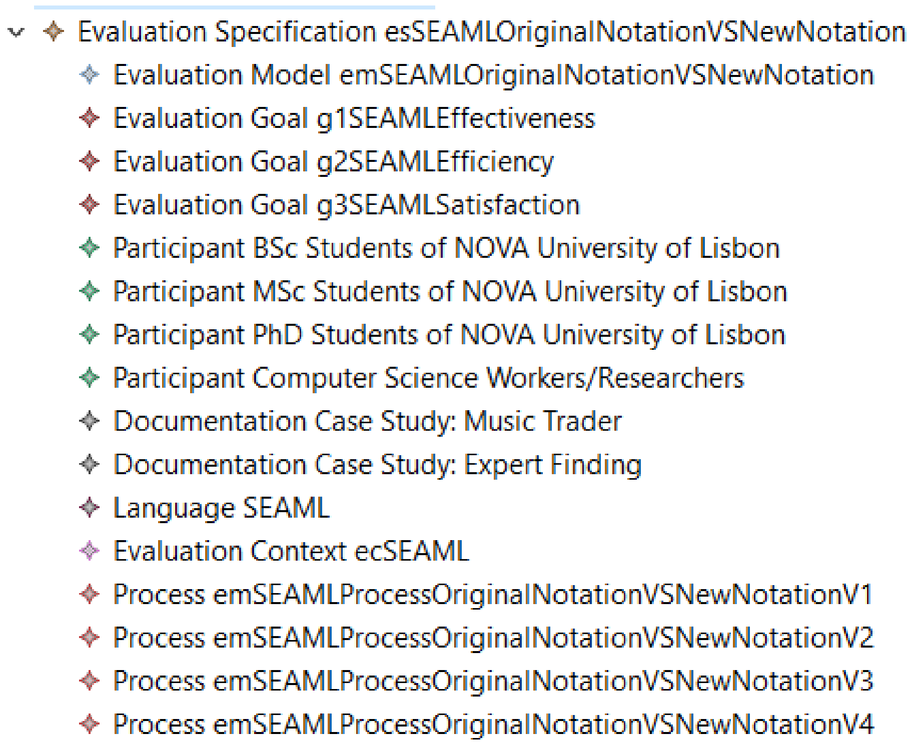


Figure 6.16: Evaluation Specification model using the USE-ME methodology for the "SEA_ML current notation VS SEA_ML new notation" experiment.

6.10.4 Interaction Specification

The Interaction Specification expresses the tasks that users are going to manage on the experiment in study.

Participants interact with this experiment using the current and the notation of SEA_ML. As we intend to evaluate the effectiveness and the efficiency of the notations in SEA_ML, we intend to capture the actions that each participant does while interacting with the scenarios they are proposed to work on. As the participants have poor or no knowledge from with *DSLs* and *MAS*, users should create the missing parts that are specified on the documentation that is provided to them during the experiment. Through the results of the experiment we will understand if the participants understood and selected the correct semantic constructs for each exercise, while adding them to the exercises. Fig.6.17 presents the Interaction Specification for the "SEA_ML current notation VS SEA_ML new notation" experiment.

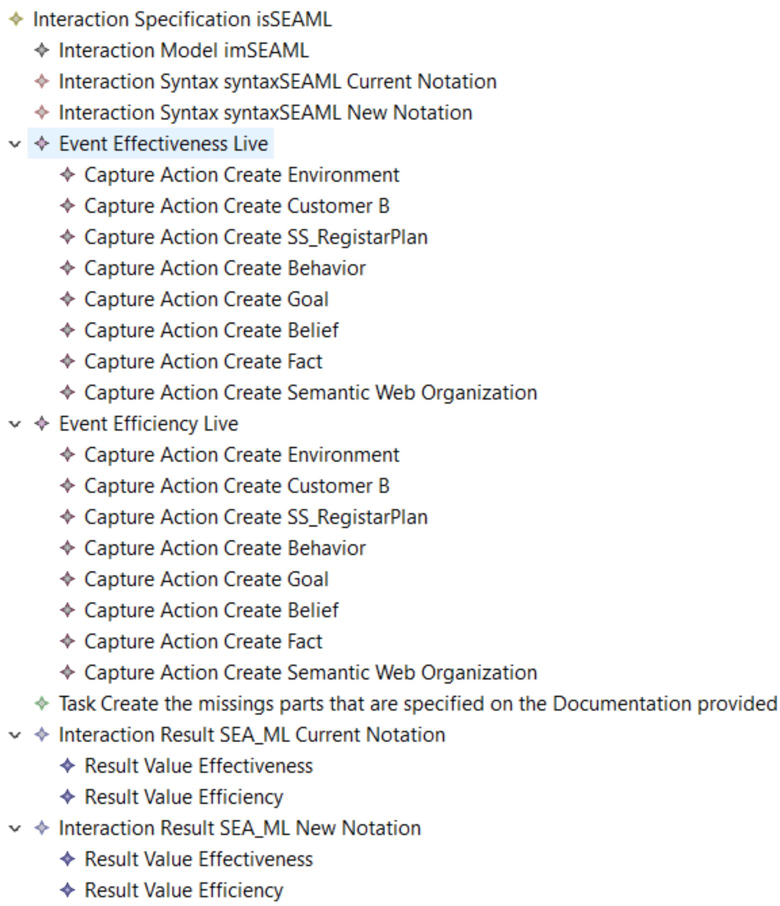


Figure 6.17: Interaction Specification model using the USE-ME methodology for the "SEA_ML current notation VS SEA_ML new notation" experiment.

6.10.5 Report Specification

For this experiment we want to understand which were the most selected visual notations by the participants and if these visual notations are the most adequate for the semantic constructs they were defined. Fig.5.6 presents the Report Modeling for the "Selecting a visual notation for SEA_ML" experiment.

The report specification helps on the construction of the final report for the experiment. The evaluation results are based on the analysis of the result models for the different tasks proposed for the experiment.

For this experiment we want to understand if the new notation for SEA_ML presents better results than the current visual notation. Participants should complete the exercises in less time with the new notation, with less errors and with a better success rate. Fig.5.6 presents the Report Specification for the "SEA_ML current notation VS SEA_ML new notation" experiment. It defines how the report of the results is made (verifying which notations were the most selected and presenting those results).

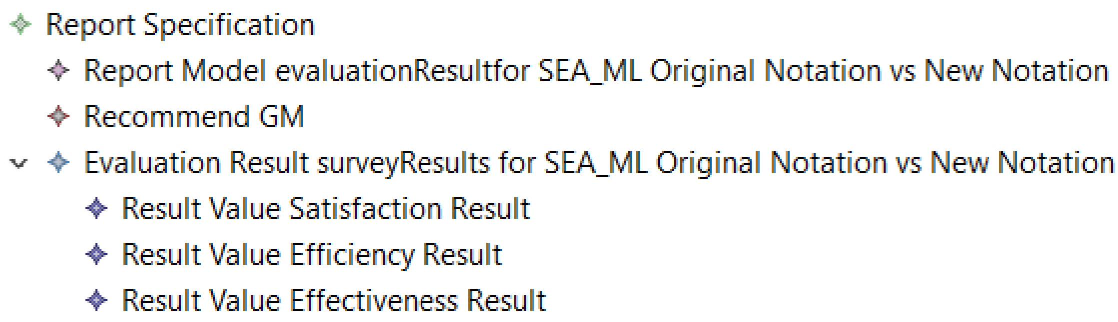


Figure 6.18: Report Specification model using the USE-ME methodology for the "SEA_ML current notation VS SEA_ML new notation" experiment.

6.10.6 Survey Specification

The Survey Specification is similar to the previous but with a set of questions that are important for the experiment.

For this experiment participants are presented to five different questionnaires: one questionnaire after the completion of each exercise (four in total) and a profile questionnaire to understand the type of participant that responded to the experiment. The first four questionnaires ask the participants about the exercise they just solved. A standard SUS questionnaire is asked on the first page, with four different questions being asked afterwards regarding their experience with the visual notations they just got to work with. For the profile questionnaire, participants are asked to define their gender, age, nationality, field of studies, completed education, current occupation and their experience with *MAS* and the *Semantic Web*. Fig.6.19 presents the Survey Modeling for the "SEA_ML current notation VS SEA_ML new notation" experiment. It defines all the questions that each participant responds during the experiment, for each version of the experiment.

6.10.7 Specification Modeling

Fig.6.20 defines the Specification Modeling for the "SEA_ML current notation VS SEA_ML new notation" experiment. It defines the goals of the experiment and what is SEA_ML and the technical specifications that define the language. As the experiment has 4 different versions, 4 different process models of evaluation were defined (one for each version).

6.11 Experiment Results

In this section the results of the "SEA_ML current notation VS SEA_ML new notation" experiment will be discussed.

6.11.1 Participants profile data

The following synthesizes the profile data of the participants in the experiment:

Figure 6.19: Survey Specification model using the USE-ME methodology for the "SEA_ML current notation VS SEA_ML new notation" experiment.

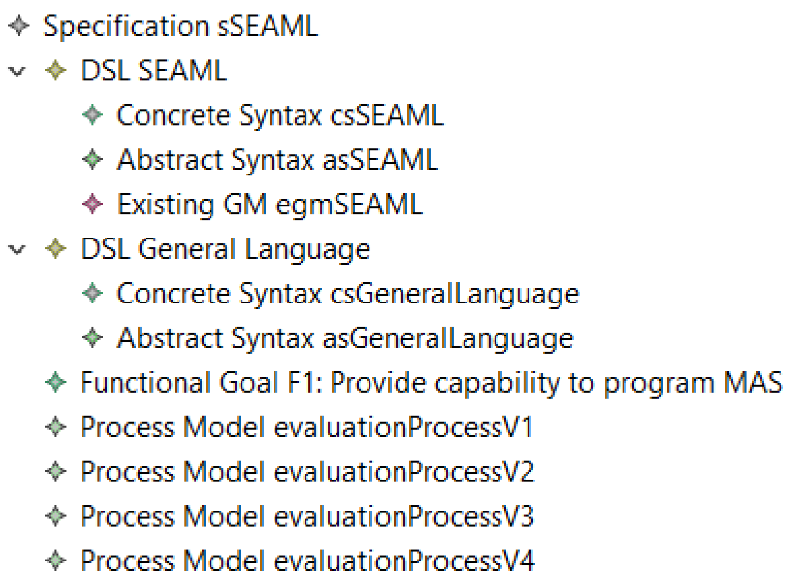


Figure 6.20: Specification model using the USE-ME methodology for the "SEA_ML current notation VS SEA_ML new notation" experiment.

1. We had 24 participants. 23 of these participants are Portuguese, while 1 is Spanish. 79,2% were male, while 20,8% were female;

2. 62,5% were Students (8,3% Work while Studying) and 37,5% were Workers (4,2% are Researchers while Working) (Fig.6.21). All of them studied/are studying Computer Science;

3. 66,7% of the participants is 22-25 years old, 25,0% are above 26 and 8,3% between 17-22 (Fig.6.22);

4. 62,5% of the participants have completed their BSc, 33,3% completed their MSc and 4,2% only completed High School (Fig.6.23);

5. 58,3% learned about the Semantic Web in the context of a course, 20,8% know what the Semantic Web is but never used it, 4,2% used it in a professional context and finally 16,7% never heard of the Semantic Web (Fig.6.24);

6. 50,0% of the participants learned about *M.A.S* in the context of a course, 4,2% know what *M.A.S* is but never used it and 45,8% never heard of it (Fig.6.25).

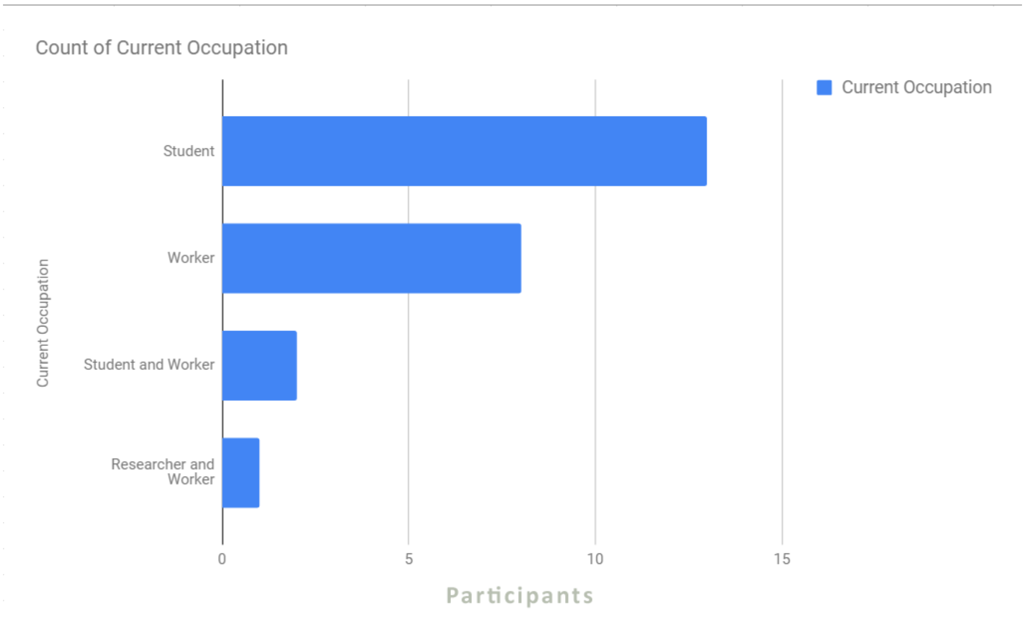


Figure 6.21: Current occupation of the participants on "SEA_ML current notation VS SEA_ML new notation" experiment

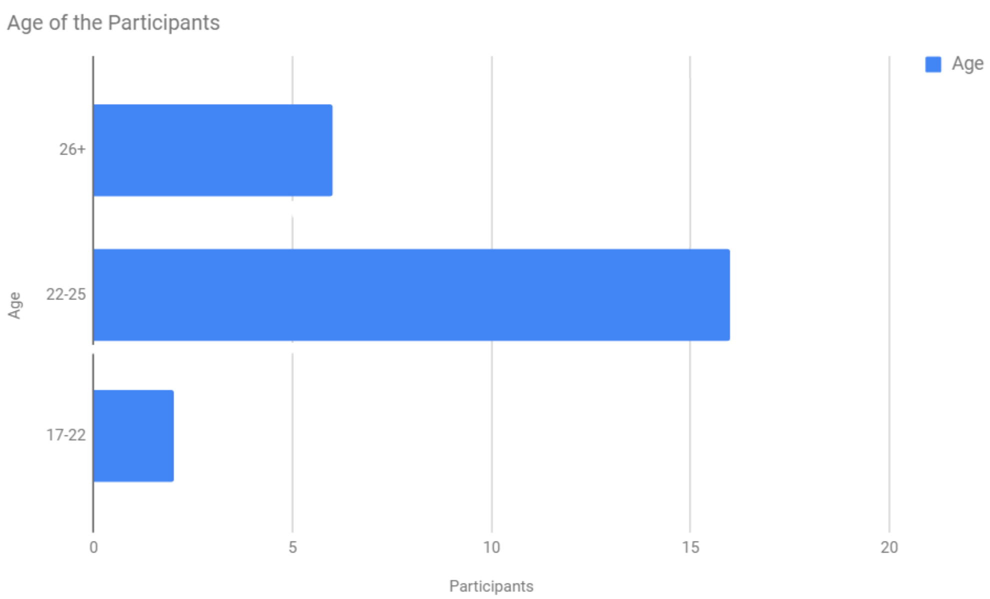


Figure 6.22: Age of the participants on the "SEA_ML current notation VS SEA_ML new notation" experiment

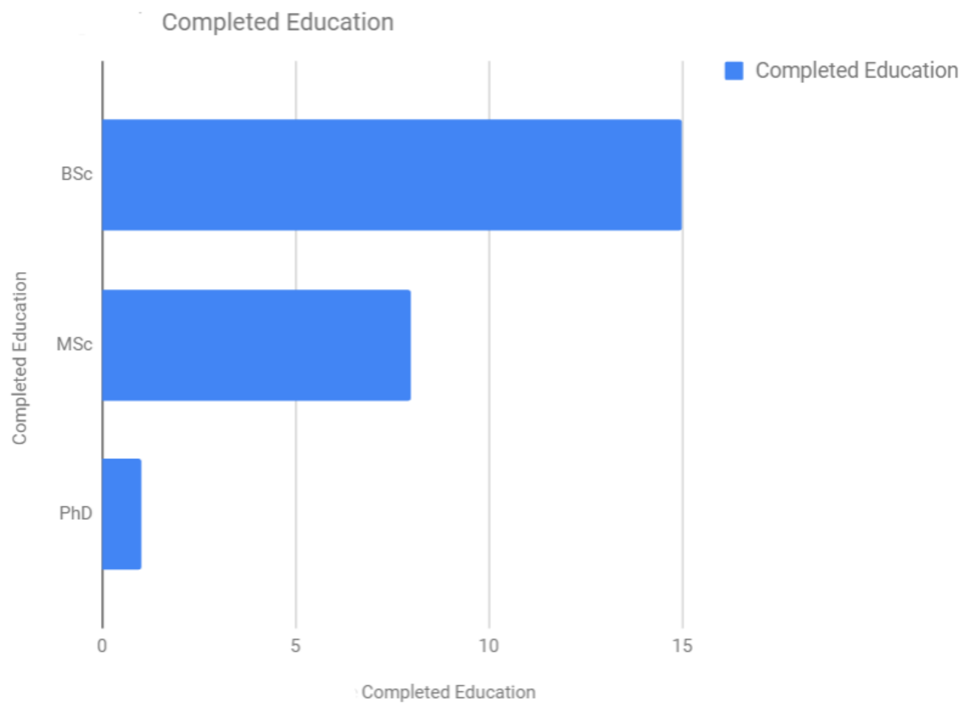


Figure 6.23: Completed education of the participants on "SEA_ML current notation VS SEA_ML new notation" experiment

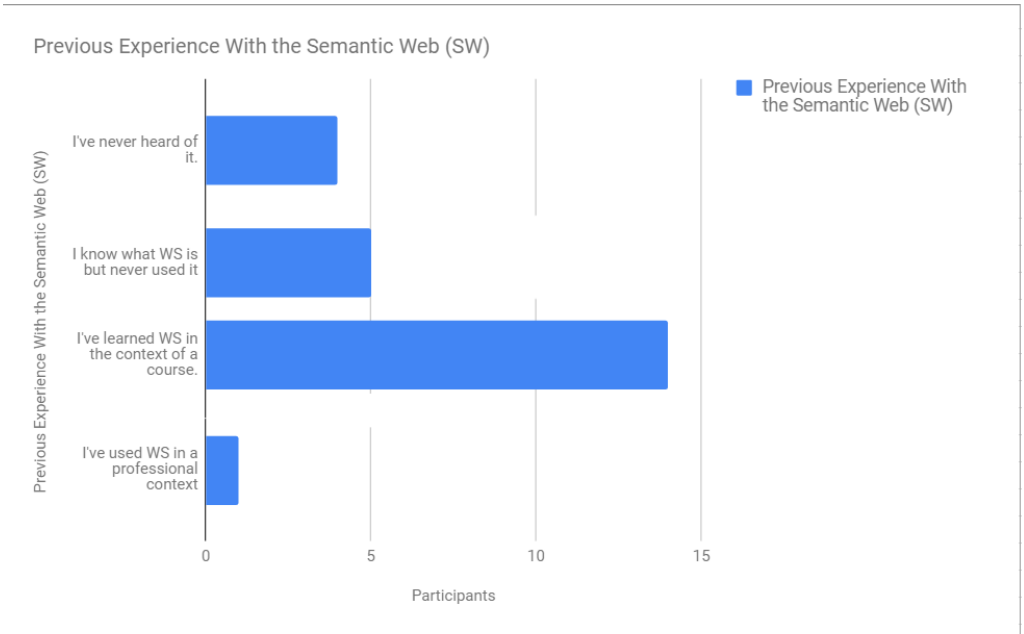


Figure 6.24: Previous experience with Semantic Web of the participants on the "SEA_ML current notation VS SEA_ML new notation" experiment

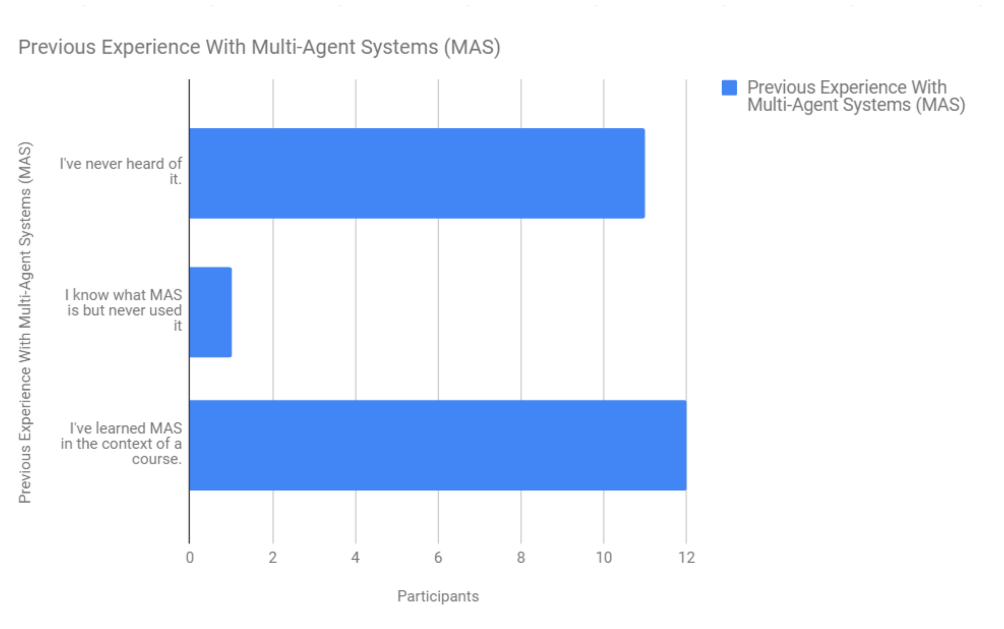


Figure 6.25: Previous Experience with *M.A.S* on the "SEA_ML current notation VS SEA_ML new notation" experiment

6.11.2 Correct answer rate and time spent on each exercise

RQ1: Are the results gathered from the experiment better using the current notation or the new notation of SEA_ML?

As already described above, version 1 is similar to version 3, with the same being produced with version 2 and version 4 (same order of exercises, different notations for each). Participants working on version 1 start with the new notation, ending on the current notation of SEA_ML. In contrast, version 2 participants start with the current notation, ending on the new notation. Both versions start with exercises from "Music Trader" case study, ending on the "Expert Finding" case study. Participants working on version 3 start with the current notation, ending on the new notation. In contrast, version 4 participants start with the new notation, ending with the current notation of SEA_ML. Versions 3 and 4 start with exercises from the "Expert Finding" case study, ending with exercises from the "Music Trader" case study.

On table 6.1 we present the answer rate for each exercise on each version. Comparing the results for similar versions, we verify that, for version 1 and version 3, participants have better results on the exercise 1 from version 3 (current notation), exercise 2 from version 1 (new notation), the same result on exercise 3 and the best result on exercise 4 from version 3 (new notation). For these versions, participants have the most correct answer rate on exercises from the new notation rather than on the current notation. For versions 2 and 4, participants have better results on the exercise 1 from version 2 (current notation), exercise 2 from version 2 (current notation), exercise 3 from version 4 (current notation) and exercise 4 from version 2 (new notation). For these versions, participants have the most correct answer rate on exercises from the current notation rather than on the current notation.

On table 6.2 we present the time spent by the participants for each exercise and for each version. It is presented the minimum, the maximum and the average time the participants spent on each exercise for each version. Comparing version 1 and version 3, participants spent less time, on average, on exercise 1 from version 3 (current notation), exercise 2 from version 3 (current notation), exercise 3 from version 3 (new notation) and exercise 4 from version 3 (new notation). As we can verify, on four exercises, participants spent less time on 2 exercises from the current notation and two exercises from the new notation. Comparing version 2 and version 4, participants spent less time, on average, on exercise 1 from version 4 (new notation), exercise 2 from version 4 (new notation), exercise 3 from version 4 (current notation) and exercise 4 from version 3 (new notation). Similar to the comparison from version 1 and version 3, participants spent almost the same time using the current and the new notations of SEA_ML. As we can see, participants spent less time on the exercises from version 3 and version 4. Comparing both versions, participants from version 3 started on the current notation, ending on the new notation. In contrast, participants from version 4 started on the new notation, ending on the current notation. With this information we can verify that the order of the notations presented to the participants is not relevant to the matter, as participants from these versions started with different notations.

Correlating this information with the correct answer rate discussed above, we verify that participants spent more time using the new notation of SEA_ML, having equivalent results when compared to the current notation.

In conclusion, the results for this experiment are similar using the current and the new visual notation of SEA_ML. The participants success rate was almost equivalent for each notation, spending more time using the new visual notation instead of the current notation.

Version \Exercise	Exercise 1			Exercise 2			Exercise 3			Exercise 4		
	CAR (%)	Case Study	Notation	CAR (%)	Case Study	Notation	CAR (%)	Case Study	Notation	CAR (%)	Case Study	Notation
Version 1	83,33%	MT	NN	87,50%	MT	NN	83,33%	EF	CN	83,33%	EF	CN
Version 3	100,00%	MT	CN	83,33%	MT	CN	83,33%	EF	NN	91,66%	EF	NN
Version 2	91,66%	EF	CN	91,66%	EF	CN	79,15%	MT	NN	58,33%	MT	NN
Version 4	83,33%	EF	NN	83,33%	EF	NN	83,33%	MT	CN	33,33%	MT	CN
	CAR (%) = Correct Answer Rate | MT = Music Trader | EF = Expert Finding | CN = Current Notation | NN = New Notation											

Table 6.1: Correct answer rate for each exercise on the "SEA_ML current notation VS SEA_ML new notation" experiment. Similar colors represent versions with the same order of exercises but with different notations

Version\Exercise	Exercise 1					Exercise 2					Exercise 3					Exercise 4					Total		
	Min	Max	Avg	Case Study	Notation	Min	Max	Avg	Case Study	Notation	Min	Max	Avg	Case Study	Notation	Min	Max	Avg	Case Study	Notation	Min	Max	Avg
Version 1	00:06:05	00:16:00	00:11:20	MT	NN	00:12:03	00:25:11	00:18:37	MT	NN	00:08:50	00:17:32	00:14:18	EF	CN	00:08:24	00:16:01	00:11:18	EF	CN	00:35:22	01:14:44	00:55:03
Version 3	00:05:16	00:15:19	00:10:44	MT	CN	00:06:19	00:17:27	00:11:53	MT	CN	00:06:40	00:16:52	00:11:54	EF	NN	00:08:15	00:16:13	00:11:07	EF	NN	00:26:30	01:06:10	00:46:20
Version 2	00:08:46	00:14:56	00:11:40	EF	CN	00:07:10	00:15:44	00:12:29	EF	CN	00:06:26	00:17:38	00:12:11	MT	NN	00:09:19	00:13:01	00:11:27	MT	NN	00:31:41	01:01:19	00:46:30
Version 4	00:06:57	00:11:31	00:09:28	EF	NN	00:05:42	00:15:08	00:10:53	EF	NN	00:06:02	00:15:08	00:11:14	MT	CN	00:09:03	00:14:17	00:11:30	MT	CN	00:27:44	00:56:04	00:41:54

MT = Music Trader | EF = Expert Finding | CN = Current Notation | NN = New Notation

Table 6.2: Times spent by the participants on the "SEA_ML current notation VS SEA_ML new notation" experiment. Similar colors represent versions with the same order of exercises but with different notations

6.11.3 Questionnaire results

RQ2: Do participants find it easier to interact and solve problems using the new visual notation of SEA_ML?

6.11.3.1 SUS questionnaire

To determine SEA_ML's usability we analyzed the results gathered from the SUS questionnaires from the usability experiments.

As referred above, the experiment presents four different versions orders of the exercises to be demanded to the participants to understand if the notation is significantly important to the success of the participants on the exercises.

Table 6.3 represents the SUS Questionnaire score for each exercise on the Version 1 of the experiment. The only exercise from the new notation that has a better mean score than the results from the current notation scores is exercise 1.

Table 6.4 represents the SUS Questionnaire score for each exercise on the Version 2 of the experiment. We can verify that the exercises from the current notation clearly have better results than the exercises from the new notation.

Table 6.5 represents the SUS Questionnaire score for each exercise on the Version 3 of the experiment. The exercises from the new notation have better results than the exercises from the current notation.

Table 6.6 represents the SUS Questionnaire score for each exercise on the Version 4 of the experiment. The exercises from the new notation have better results than the exercises from the current notation.

Crossing versions that have the same exercises appearing at the same time but with different notations (version 1 and version 3, version 2 and version 4), we can compare that the results give slightly better results to the new notation rather than the current notation (5 exercises have better results on the new notation, while 3 exercises have better results on the current notation).

6.11.4 Hypothesis testing

To validate the results that the SUS questionnaires provided us on the subsection above, the Welch's t-test was tested using the results gathered from the current notation and the new proposed notation of SEA_ML. This is made due to the fact that the size of the

Version 1	2*Notation	N	Minimum	Maximum	Mean		Std. Deviation
		Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic
Music Trader: Exercise 1	NN	6	57,50	80,00	66,67	3,46	8,47
Music Trader: Exercise 2	NN	6	30,00	70,00	44,17	6,86	16,78
Expert Finding: Exercise 1	CN	6	47,50	70,00	57,50	3,82	9,35
Expert Finding: Exercise 2	CN	6	47,50	75,00	60,00	4,43	10,84
CN = Current Notation | NN = New Notation							

Table 6.3: SUS Questionnaire Results for Version 1

Version 2	2*Notation	N	Minimum	Maximum	Mean		Std. Deviation
		Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic
Expert Finding: Exercise 1	CN	6	30,00	95,00	62,50	10,99	26,93
Expert Finding: Exercise 2	CN	6	42,50	100,00	70,42	9,95	24,36
Music Trader: Exercise 1	NN	6	35,00	100,00	61,25	9,57	23,44
Music Trader: Exercise 2	NN	6	10,00	97,50	44,58	13,39	32,80
CN = Current Notation | NN = New Notation							

Table 6.4: SUS Questionnaire Results for Version 2

Version 3	2*Notation	N	Minimum	Maximum	Mean		Std. Deviation
		Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic
Music Trader: Exercise 1	CN	6	32,50	65,00	52,08	4,93	12,09
Music Trader: Exercise 2	CN	6	17,50	65,00	40,83	6,57	16,09
Expert Finding: Exercise 1	NN	6	40,00	85,00	56,25	7,55	18,49
Expert Finding: Exercise 2	NN	6	15,00	82,50	56,67	9,97	24,43
CN = Current Notation | NN = New Notation							

Table 6.5: SUS Questionnaire Results for Version 3

Version 4	2*Notation	N	Minimum	Maximum	Mean		Std. Deviation
		Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic
Expert Finding: Exercise 1	NN	6	47,50	90,00	72,50	6,55	16,05
Expert Finding: Exercise 2	NN	6	35,00	100,00	69,58	10,05	24,62
Music Trader: Exercise 1	CN	6	35,00	72,50	49,17	5,23	12,81
Music Trader: Exercise 2	CN	6	5,00	30,00	15,42	4,30	10,54
CN = Current Notation | NN = New Notation							

Table 6.6: SUS Questionnaire Results for Version 4

samples is short for each version of the experiment [Kit+17]. Alongside, the Levene's and the Brown-Forsythe were also tested, trying to check the equality of variance for a variable calculated for two or more groups (on the case of this study, two groups - current and new notation of SEA_ML) [Rao61].

It was hypothesized that the results would be better using the new notation instead of using the current notation of SEA_ML.

Using a level of significance of 5% (0,05) for the Levene's, Welch's and Brown-Forsythe test, we verify that if the probability is below 5% (0,05) for each test, that represents that the sample gathered from the experiment are statistically significant to support the claim that the new proposed notation is different than the current notation.

As each participant answered to four questionnaires about the visual notations (two for each notation), a total of 96 questionnaires were gathered (48 per notation). Table 6.7 presents the results for the three tests defined above. The mean results for the new notation were better than the result from the current notation for the SUS questionnaires (58,9583 for the new notation and 50,9896 for the current notation).

For the Levene's Test, the difference between the new and the current notation of SEA_ML resulted in $F(1{,}00, 94{,}00) = 0{,}446$, $p = 0{,}506$, $p > 0{,}05$, determine that the samples gathered from the experiment are not representative to conclude that the obtained results are significant for the experiment.

For the Welch's T-Test, the difference between the populations from the new and the current notation of SEA_ML resulted in $F(1, 93{,}982) = 3{,}064$, $p = 0{,}08$, $p > 0{,}05$,

	N	Minimum	Maximum	Mean	Std. Deviation
	Statistic	Statistic	Statistic	Statistic	Statistic
New Notation	48,00	10,00	100,00	58,9583	22,45464
Current Notation	48,00	5,00	100,00	50,9896	22,14999
Total	96,00	5,00	100,00	54,9740	22,54380
Levene's Test	Difference for the New Notation	Difference for the current notation	Significance		
0,446	1,00	94,00	0,506		
	Statistic	Difference for the New Notation	Difference for the current notation	Significance	
Welch's T-Test	3,064	1	93,982	0,083	
Brown-Forsythe Test	3,064	1	93,982	0,083	

Table 6.7: SUS Questionnaire results using the Levene's, Welch's and Brown-Forsythe test

which determines that the samples gathered from the experiment are not representative to conclude that the obtained results are significant for the experiment.

For the Brown-Forsythe test, the difference between the populations from the new and the current notation of SEA_ML resulted in F(1, 93,982) = 3,064, p = 0,08, p> 0,05 which determines that the samples gathered from the experiment are not representative to conclude that the obtained results are significant for the experiment.

Taking into account the short number of participants for this experiment, we can verify that the presented means are around the same values, where we can conclude that we cannot see a significant difference when comparing the results from each SUS Questionnaire for the current and the new notation of SEA_ML. Both results are below 68, value that is considered to be when the SUS scores are above average [Ban+09].

Table 6.8 and table 6.9 present the learnability results for each visual notation as presented for the participants. Since each participant used both notations on the experiment, we wanted to understand the impact participants had when facing each notation (e.g: starting on the new notation and then passing to the current notation or vice-versa). When verifying the results we can determine that participants did not notice any difference between the current notation and the new notation. To verify that the obtained results are significant, the Levene's, Welch's and Brown-Forsythe Test were applied for the gathered results.

For the Levene's Test, the difference between the new and the current notation of SEA_ML resulted in F(1,00, 46,00) = 1,179, p = 0,28, p > 0,05, determine that the samples gathered from the experiment are not representative to conclude that the obtained results are significant for the experiment.

For the Welch's T-Test, the difference between the populations from the new and the current notation of SEA_ML resulted in F(1, 44,178) = 1,765, p = 0,191, p > 0,05, which determines that the samples gathered from the experiment are not representative to conclude that the obtained results are significant for the experiment.

For the Brown-Forsythe test, the difference between the populations from the new and the current notation of SEA_ML resulted in F(1, 44,178) = 1,765, p = 0,191, p> 0,05 which determines that the samples gathered from the experiment are not representative to conclude that the obtained results are significant for the experiment.

Levene's Test	Difference for participants that started first with the new notation	Difference for participants that started first with the current notation	Significance
1,179	1,00	46,00	0,283
Welch's T-Test	Difference for participants that started first with the new notation	Difference for participants that started first with the current notation	Significance
1,765	1	44,178	0,191
Brown-Forsythe Test	Difference for participants that started first with the new notation	Difference for participants that started first with the current notation	Significance
1,765	1	44,178	0,191

Table 6.8: Learnability significance using the new notation of SEA_ML

Levene's Test	**Difference for participants that started first with the current notation**	**Difference for participants that started first with the new notation**	**Significance**
0,097	1,00	46,00	0,757
Welch's T-Test	**Difference for participants that started first with the new notation**	**Difference for participants that started first with the current notation**	**Significance**
3,054	1	45,765	0,087
Brown-Forsythe Test	**Difference for participants that started first with the new notation**	**Difference for participants that started first with the current notation**	**Significance**
3,054	1	45,765	0,087

Table 6.9: Learnability significance using the current notation of SEA_ML

6.11.5 Visual notations questionnaire

Alongside the SUS questionnaire, three questions were also proposed to the participants for each experiment in order to understand if they think that the visual notations and the user interface (*UI*) influenced there interaction with the system. The questions were the following:

1. The symbols on the user interface (UI) were easy to understand;
2. The symbols on the UI are adequate to the constructions they are linked to;
3. The symbols on the UI helped me solve the exercise in less time.

Each question had a Likert scale, where 1 meant "Strongly Disagree" and 5 "Strongly Agree".

For these questions we used the Paired Samples t-Test, which is a statistical test used to determine whether the mean difference between two sets of observations is zero.

Using the same variance that was applied on the SUS questionnaire, a 5% significance level is applied, which means that if we have values below this percentage the results are relevant, while above are not significant to conclude that the results are relevant.

Table 6.10 presents the results for the question "The symbols on the user interface (UI) were easy to understand". Verifying its means, we verify that the new notation had better results than the current notation (3,958 for the new notation and 2,916 for the current notation). These means refer that the participants of this experiment find the visual notations from the new notation easier to understand than the current notation of SEA_ML. Applying the Paired Samples T-Test we verify that the significance of the gathered results is 98,6%, above the 5% level of significance defined above, which means that the results gathered from the experiment are not relevant enough to conclude that the new notation results are better than the results for the current notation.

Table 6.11 presents the results for the question "The symbols on the UI are adequate to the constructions they are linked to". Verifying its means, we verify that the new notation had better results than the current notation (3,645 for the new notation and 2,958 for the current notation). These means refer that the participants of this experiment find the visual notations from the new notation easier to understand than the current notation of SEA_ML. Applying the Paired Samples T-Test we verify that the significance of the gathered results is 29,0%, above the 5% level of significance defined above, which means that the results gathered from the experiment are not relevant enough to conclude that the new notation results are better than the results for the current notation.

Table 6.12 presents the results for the question "The symbols on the UI helped me solve the exercise in less time". Verifying its means, we verify that the new notation had better results than the current notation (3,833 for the new notation and 2,854 for the current notation). These means refer that the participants of this experiment find the visual notations from the new notation easier to understand than the current notation of SEA_ML. Applying the Paired Samples T-Test we verify that the significance of the gathered results is 46,6%, above the 5% level of significance defined above, which means that the results gathered from the experiment are not relevant enough to conclude that the new notation results are better than the results for the current notation.

Applying the Paired Samples T-Test for each version, we verify that the results are not statistically significant for each of these questions.

	Paired Samples Statistics			
	Mean	N	Std. Deviation	Std. Error Mean
New Notation	3,958	48	1,090	0,157
Current Notation	2,916	48	1,251	0,180
	Paired Samples Correlations			
	N	Correlation	Significance	
New Notation & Current Notation	48	-0,003	0,986	

Table 6.10: Paired Samples t-Test results for the question "The symbols on the user interface (UI) were easy to understand"

	Paired Samples Statistics			
	Mean	N	Std. Deviation	Std. Error Mean
New Notation	3,645	48	1,020	0,147
Current Notation	2,958	48	1,030	0,148
	Paired Samples Correlations			
	N	Correlation	Significance	
New Notation & Current Notation	48	-0,156	0,290	

Table 6.11: Paired Samples t-Test results for the question "The symbols on the UI are adequate to the constructions they are linked to"

	Paired Samples Statistics			
	Mean	N	Std. Deviation	Std. Error Mean
New Notation	3,833	48	1,098	0,158
Current Notation	2,854	48	1,288	0,185
	Paired Samples Correlations			
	N	Correlation	Significance	
New Notation & Current Notation	48	-0,108	0,466	

Table 6.12: Paired Samples t-Test results for the question "The symbols on the UI helped me solve the exercise in less time"

6.11.6 Discussion of the results

A new notation was proposed to SEA_ML in order to be in accordance to the principles of "The "Physics" of Notations".

We hypothesised that the new visual notation of SEA_ML would have a higher usability rating when compared to the current visual notation. Verifying the results through its time and success rate on the proposed exercises we can conclude that the results are not statistically relevant to prove that the new notation is better than the current notation.

When comparing the SUS scores without the Levene's, Welch's and Brown-Forsythe test we verify that the results are not significant to prove that the new notation is better than the current notation. Applying the Levene's, Welch's and Brown-Forsythe Test we verify the same results, proving that the samples gathered are not significant enough to conclude that the new notation is really better than the current notation.

Alongside the results from the SUS questionnaire, the visual notations questionnaire also provided results that are not significant enough to support that the UI and the visual notations affect the interaction of the participants in the system.

As reported above, the new notation that was proposed to SEA_ML intended to improve the learnability of the *DSL* using better metaphors for its semantic constructs. As users interact with SEA_ML using a visual workbench, this would ideally improve the *DSL*. The gathered results show that the participants did not find clear differences between the current and the new visual notation of SEA_ML, where we can imply that even though users interact with these visual notations, users will still look-up for the names of the semantic constructs when interacting with *SEA_ML*.

Although the gathered results are not significant to prove that the new proposed visual notation is better than the current notation of SEA_ML, the current visual notation of SEA_ML has room for improvement and the new proposed notation find ways to improve these visual notation. As we apply the principles of "The "Physics" of Notations" in order to normalize the visual notation of SEA_ML, we could verify that the current visual notation is not according to these principles, which directly improve the visual communication of the *DSL*. The new proposed visual notation considers these principles and applies them accordingly, resulting on a new visual notation based on a standard for visual notations.

6.11.7 Threats to Validity

[Woh+12] presents some threats to the validity Experimentation in Software Engineering. The population selection for this experiment is one of the concerning threats. Participants that test the experiment should provide a representative collection of the population. Due to timing and resource constraints, all testers of the experiment were students/former students of the Computer Science course in NOVA University of Lisbon. Ideally the experiment should have more participants from different colleges and countries, from the Computer Science area, because SEA_ML is for participants from Computer Science.

More participants, from different countries and different colleges, but from the same course, would increase the confidence level in the experimental evaluation results for the experiment and concrete results.

The experience of the participants should also be taken in consideration, as mixing different types of users from the same course can lead to results different than the expected ones.

A construction validity to the experiment should also be taken in consideration. The chosen perspective behind this experience may not be representative or the best for the presented scenarios. It is being tested that the participants will have better results when using the new visual notation instead of the current notation of SEA_ML, but it is not clear that the visual notations proposed are the more suitable for the semantic constructs of SEA_ML. As the participants of this experiment only interact with SEA_ML using its current and new visual notation on controlled exercises, participants are restricted to a small sample and cannot add a different visual notation for some of the semantic constructs.

The exercises that are being proposed for the participants to deal with only define a part of SEA_ML. Ideally, participants would interact with every section of the *DSL*. On developing this experiment it was agreed with the Domain Experts from EGE University that the presented scenarios would represent the principal features of SEA_ML.

The time spent by the participants on the experiment when reading the details of the scenarios and comparing them to the exercises they are provided to may lead to imprecise data, as participants may not be focused enough when making the whole experience (each exercise can take up to 20 minutes to completion and is dealing with many specific details of SEA_ML).

As the complexity of each exercise presented on this experiment is similar, it would be interesting to produce a similar experience using exercises with different complexities.

6.12 Summary

In this chapter we presented the experiment "SEA_ML current notation VS SEA_ML new notation". This experiment allows participants to interact with SEA_ML's current notation and the new proposed notation on two different case studies: the "Music Trader" and the "Expert Finding". The experiment has four different exercises (two for each case study), representing four different viewpoints from SEA_ML. The results of the experiment conclude that the new proposed notation has slightly better results than the current notation. Applying three different statistic tests, we verify that the results are not relevant to conclude that one notation is better than the other, with participants not understanding the difference between each presented notation.

Chapter 7

Conclusions and Future Work

This thesis intended to evaluate the usability of the *SEA_ML* language in order to improve it. To do so, the [Bar16] methodology was used and "The "Physics" of Notations" [Moo09] principles were applied.

The experiment "Selecting a visual notation for SEA_ML" was made to a selected group of students, in accordance with the Domain Experts. From the 33 visual notations we verified that only 4 of the current visual notations were selected by the participants, while 29 semantic constructs presented different visual notations from the current SEA_ML visual notation. From the 33 semantic constructs that were proposed to be modified, only 1 visual notation did not correlate with the correct semantic construct. Some of the semantic constructs needed other experience from the participants, as shown on the "ODMOWLCLASS" semantic construct, where the most selected icon had a 16% of selection and 16 different icons were selected to be the most suitable visual notation for this semantic construct.

For the experiment "SEA_ML current notation VS SEA_ML new notation", we applied the Levene's, Welch's and Brown-Forsythe Test and verified that the results were not statistically relevant, which means that the population that was tested on the experiment was not sufficient enough to proof that the new is better than the current notation.

Both experiences were made in parallel and there was no connection between them when executing them. This was due to the fact that the experiences were made on two different moments in time. As the "SEA_ML current notation VS SEA_ML new notation" does not correlate the results gathered from the "Selecting a visual notation for SEA_ML", participants from the first experiment used the current and the new visual notation described on chapter 4. Correlating the results from both experiments, we verify that the participants select more notations from the new visual notations and have similar results when interacting with both notations, verifying that we do not have significant results to proof that the new visual notation is better than the current notation of SEA_ML. After both experiments and the application of "The "Physics" of Notations" to the SEA_ML notation, we have detected that there is room for improvement on the visual notation of SEA_ML.

7.1 Contributions

We detected that there was room for improvement for the SEA_ML language and for that reason a new notation was developed for the *DSL*.

Two experiments were developed and can be executed on other experiments that try to understand which is the most suitable notation for SEA_ML. The "SEA_ML current notation VS SEA_ML new notation" experiment compared both notations with the same type of exercises and the "Selecting a visual notation for SEA_ML" experiment had a mix of the current notation and the new proposed notation in order for the users to select the most suitable notation for SEA_ML.

7.2 Future Work

The experiments should be applied to different and more users in order to have more concrete results. The new proposed notation is a basis that can be used to improve SEA_ML's visual notation, improving the usability of the *DSL*.

The remaining viewpoints that were not tested on this thesis should also be evaluated in order to understand if there is room for improvement for those viewpoints, as discussed on chapter 6.

Bibliography

[Ban+09] A. Bangor, P. Kortum, and J. Miller. "Determining what individual SUS scores mean: Adding an adjective rating scale". In: *Journal of usability studies* 4.3 (2009), pp. 114–123.

[Bar16] A. Barisic. *Usability Software Engineering - Modeling Environment (USE-ME 1.0)*. Nov. 2016. DOI: 10.5281/zenodo.189245. URL: https://doi.org/10.5281/zenodo.189245.

[Bel+05] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi. "JADE—a java agent development framework". In: *Multi-Agent Programming*. Springer, 2005, pp. 125–147.

[BL+01] T. Berners-Lee, J. Hendler, O. Lassila, et al. "The semantic web". In: *Scientific american* 284.5 (2001), pp. 28–37.

[Ber+02] C. Bernon, M.-P. Gleizes, S. Peyruqueou, and G. Picard. "ADELFE: a methodology for adaptive multi-agent systems engineering". In: *International Workshop on Engineering Societies in the Agents World*. Springer. 2002, pp. 156–169.

[Bey+09] G. Beydoun, G. Low, B. Henderson-Sellers, H. Mouratidis, J. J. Gomez-Sanz, J. Pavon, and C. Gonzalez-Perez. "FAML: a generic metamodel for MAS development". In: *IEEE Transactions on Software Engineering* 35.6 (2009), pp. 841–863.

[Bra+12] M. Brambilla, J. Cabot, and M. Wimmer. "Model-driven software engineering in practice". In: *Synthesis Lectures on Software Engineering* 1.1 (2012), pp. 1–182.

[Bus+99] P. Busetta, R. Rönnquist, A. Hodgson, and A. Lucas. "Jack intelligent agents-components for intelligent agents in java". In: *AgentLink News Letter* 2.1 (1999), pp. 2–5.

[Cam+10] J. P. Campos, A. de Paiva Oliveira, and J. L. Braga. "Modeling Languages for Multiagent Systems: A case study using AUML and MAS-ML". In: *INFOCOMP Journal of Computer Science* 9.2 (2010), pp. 57–64.

[Cha+14] M. Challenger, S. Demirkol, S. Getir, M. Mernik, G. Kardas, and T. Kosar. "On the use of a domain-specific modeling language in the development of multiagent systems". In: *Engineering Applications of Artificial Intelligence* 28 (2014), pp. 111–141.

[Cha+16] M. Challenger, G. Kardas, and B. Tekinerdogan. "A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems". In: *Software Quality Journal* 24.3 (2016), pp. 755–795.

[Dav09] D. A. David. "Considering non-functional requirements in model-driven engineering". In: *Universidad Politécnica de Catalunya* (2009).

[EE+11] M. Elammari, R. Elsaeti, et al. "Multi-Agent System Meta-Models". In: (2011).

[FG97] J. Ferber and O. Gutknecht. *Aalaadin: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems*. 1997.

[FG98] J. Ferber and O. Gutknecht. "A meta-model for the analysis and design of organizations in multi-agent systems". In: *Multi Agent Systems, 1998. Proceedings. International Conference on*. IEEE. 1998, pp. 128–135.

[GS14] J. J. Gomez-Sanz. "Ten years of the ingenias methodology". In: *Agent-oriented software engineering*. Springer. 2014, pp. 193–209.

[GSP05] J. J. Gómez-Sanz and J. Pavón. "Implementing multi-agent systems organizations with INGENIAS". In: *International Workshop on Programming Multi-Agent Systems*. Springer. 2005, pp. 236–251.

[Gon+15] E. J. T. Gonçalves, M. I. Cortés, G. A. L. Campos, Y. S. Lopes, E. S. Freire, V. T. da Silva, K. S. F. de Oliveira, and M. A. de Oliveira. "MAS-ML 2.0: Supporting the modelling of multi-agent systems with different agent architectures". In: *Journal of Systems and Software* 108 (2015), pp. 77–109.

[Hah08] C. Hahn. "A domain specific modeling language for multiagent systems". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2008, pp. 233–240.

[Hah+09] C. Hahn, C. Madrigal-Mora, and K. Fischer. "A platform-independent metamodel for multiagent systems". In: *Autonomous Agents and Multi-Agent Systems* 18.2 (2009), pp. 239–266.

[Häs+16] F. Häser, M. Felderer, and R. Breu. "An integrated tool environment for experimentation in domain specific language engineering". In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. ACM. 2016, p. 20.

[Jok+03] T. Jokela, N. Iivari, J. Matero, and M. Karukka. "The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11". In: *Proceedings of the Latin American conference on Human-computer interaction*. ACM. 2003, pp. 53–60.

[Kit+17] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs, and A. Pohthong. "Robust statistical methods for empirical software engineering". In: *Empirical Software Engineering* 22.2 (2017), pp. 579–630.

[Küh06] T. Kühne. "Matters of (meta-) modeling". In: *Software & Systems Modeling* 5.4 (2006), pp. 369–385.

[Mer+05] M. Mernik, J. Heering, and A. M. Sloane. "When and how to develop domain-specific languages". In: *ACM computing surveys (CSUR)* 37.4 (2005), pp. 316–344.

[Moo09] D. Moody. "The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering". In: *IEEE Transactions on Software Engineering* 35.6 (2009), pp. 756–779.

[Nie03] J. Nielsen. *Usability 101: Introduction to usability*. 2003.

[NW06] D. G. Novick and K. Ward. "Why don't people read the manual?" In: *Proceedings of the 24th annual ACM international conference on Design of communication*. ACM. 2006, pp. 11–18.

[Pav+05] J. Pavón, J. J. Gómez-Sanz, and R. Fuentes. "The INGENIAS methodology and tools". In: *Agent-oriented methodologies*. IGI Global, 2005, pp. 236–276.

[Puv+08] M. Puviani, G. Cabri, and L. Leonardi. "Agent roles: From methodologies to infrastructures". In: *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*. IEEE. 2008, pp. 416–423.

[Rao61] C. Rao. *Contributions to Probability and Statistics. Essays in Honor of Harold Hotelling*. 1961.

[Rum+04] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified modeling language reference manual, the*. Pearson Higher Education, 2004.

[Sei+10] V. Seidita, M. Cossentino, V. Hilaire, N. Gaud, S. Galland, A. Koukam, and S. Gaglio. "The metamodel: a starting point for design processes construction". In: *International Journal of Software Engineering and Knowledge Engineering* 20.04 (2010), pp. 575–608.

[Sil+02] V. Silva, A. Garcia, A. Brandão, C. Chavez, C. Lucena, and P. Alencar. "Taming agents and objects in software engineering". In: *International Workshop on Software Engineering for Large-Scale Multi-agent Systems*. Springer. 2002, pp. 1–26.

[Voe+13] M. Voelter, S. Benz, C. Dietrich, B. Engelmann, M. Helander, L. C. Kats, E. Visser, and G. Wachsmuth. *DSL engineering: Designing, implementing and using domain-specific languages*. dslbook. org, 2013.

[War13] S. Warwas. *DSML4MAS Sourceforge Wiki*. https://sourceforge.net/p/dsml4mas/wiki/Home/. 2013.

[Woh+12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[Woo09] M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[Woo+00] M. Wooldridge, N. R. Jennings, and D. Kinny. "The Gaia methodology for agent-oriented analysis and design". In: *Autonomous Agents and multi-agent systems* 3.3 (2000), pp. 285–312.

Appendix A

Case Studies for the "SEA_ML original notation VS SEA_ML new notation" experiment

In this appendix are presented in detail the case studies for the "SEA_ML original notation VS SEA_ML new notation" experiment. It presents two different case studies: the Music Trader and the Expert Finding. Each case study has two different exercises. The Music Trader exercise uses the *M.A.S* and *Agent-SWS* viewpoints, while the Expert Finding uses the *Agent Internal* and *Ontology* viewpoints.

Each case study has a different notation (two exercises for each notation). The exercises ask users to assess if a certain viewpoint is correct according to the text provided to them. If the viewpoint is wrong or incomplete, users should complete it in order to have it correctly done.

As we have four different versions of the experiment, the exercises are presented in a different order for each version:

1. — **Version 1** — Participants start on the case study "Music Trader", finishing on the "Expert Finding" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled with the new notation, while the second case study is modeled with the original notation;

2. — **Version 2** — Participants start on the case study "Expert Finding", finishing on the "Music Trader" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled with the original notation, while the second case study is modeled with the new notation;

3. — **Version 3** — Participants start on the case study "Music Trader", finishing on the "Expert Finding" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled with the original notation, while the second case study is modeled with the new notation;

4. — **Version 4** — Participants start on the case study "Expert Finding", finishing on the "Music Trader" case study. Each case study has 2 exercises and different viewpoints (4 exercises and 4 viewpoints in total). The first case study is modeled with the new notation, while the second case study is modeled with the original notation;

In this case study, users are requested to develop a system that allows agents to trade their music albums without using any currency. Agents want to trade their music albums for other albums, with this trade being made on an N to N basis (Agent A wants to trade the album A1 for the album B1).

EXERCISE 1:
Based on the description of the system proposed below, a representation of the system has been built in order to fulfill every necessary step. Please review the following diagram and, if you think it is needed, complete it, in order to be satisfying according to the system's description.

In order for the system to work properly, the system has an Exchange Manager (represented in the system as a ***Semantic Web Agent***), responsible for every trade that is made in the system. The Exchange Manager receives every trade proposal that is submitted in to the system by the hand of the Managers, represented by a ***Semantic Web Organization*** in the system. After receiving every trade proposal, the Exchange Manager has to match trade proposals and follow all of the trade process between users.

Each trade has at least 2 items (one item for each user), and they are directly connected to the Managers and the Exchange Manager. Alongside these items, each trade has a Data Manager that cooperates directly with the Exchange Manager and Managers to manage correctly the information to be traded.

All of the exchanges are inserted in an Environment (represented in the system as an ***Environment***), with this Environment connecting (represented in the system as an ***Interact_with***) to every ***Semantic Web Organization*** existing in the system.

An example can be made that illustrates the Music Trader system:

- Pedro (Customer A) has the latest album from the British band Muse. He listened carefully to the album but he does not seem to find any connection to the band nowadays. As such, he would like to trade this album for the most recent album of the French band Phoenix.
- Sara (Customer B), on the other hand, bought the latest album from the French band Phoenix and felt disappointed with what she heard. Since she is a fan of the British band Muse and doesn't have the latest album, she would like to trade the Phoenix disc for the latest Muse disc.

EXERCISE 2:
Based on the description of the system proposed below, a representation of the system has been built in order to fulfill every necessary step. Please review the following diagram and, if you think it is needed, complete it, in order to be satisfying according to the system's description.

In order for the system to work properly, the system has a Music Trader System (represented by a ***SWS***), including two Web Services: Exchanging and Items (each represented by a ***Web Service***). Items calls the service Exchange Call (represented by a ***Grounding***), being called on both ways (from Exchange Call to Items and Items to Exchange Call). Music Trader System is also connected to a ***Process*** named Exchange Process, with the ***Link*** being made between them called "described by", connecting from the Exchange Process to the Music Trader System. Alongside both connections, a third connection is made using an ***Interface*** called Trade Flyer, connecting from Trade Flyer to Music Trader System.

The Music Trader System has a ***Role*** Trades interacting with him, that is played by an Exchange Manager (represented by a ***Semantic Web Agent***).

The Exchange Manager is connected to a Music Trading Finder (represented by a ***SS_FinderPlan***), with that connection being described with a ***Link*** called "applies". The Exchange Manager is also connected to Trading (represented by ***SS_ExecutorPlan***) with a ***Link*** called "applies". Alongside these connections, a connection called Exchanging is made (represented by a ***SS_AgreementPlan***) with that connection being described with a ***Link*** called "applies".

The Exchange Call previously described also connects to Trading using a ***Link*** called "uses".

The Music Trading Finder previously described also connects to a Music Trading Matcher (represented by ***SSMatchmaker Agent***). This Music Trading Matcher connects to a Music Trader Registration (represented by a ***SS_RegisterPlan***). A ***Link*** called "applies" is made between them (from the Music Trader Matcher to Music Trader Registration). Music Trading Finder also connects to the previous described TradeFlyer, using a ***Link*** called "discovers".

Exchanging (that was previously described) also connects to Trade Flyer.

Trading (that was previously described) also connects to Exchange Process using the ***Link*** called "executes".

The ExchangeProcess is described by 5 characteristics: two ***Inputs*** (Request and Parameters), one ***Output*** (Result), one ***Precondition*** (Check the Availability) and one ***Effect*** (Add the Request Result).

The TradeFlyer is described by 5 characteristics: two ***Inputs*** (Exchange Request and Type), one ***Output*** (Exchange Result), one ***precondition*** (Exchange Availability) and one ***Effect*** (Check if the trade is Possible).

In this case study, users are requested to develop a system that allow agents to find information about other agents that they are searching for in order to communicate with them. Agents have some information about the other agent they are looking for (they are family related or were friends at the past), which is crucial in order to find the correct Semantic Web Service to search the right person. The communication between agents can be made through Social Networks, E-Mail, VoIP or Phone Call.

An example can be made that illustrates the Expert Finding system:

- Ana (User A) is trying to find her old friend Tobias. She doesn't have any contact with him on the last ten years. She has one phone number that belonged to him and his old e-mail. She also knows his full name, date of birth and the address where he lived with his parents. Ana expects that with this information she can find Tobias with the help of the Expert Finding system.

EXERCISE 3:
Based on the description of the system proposed below, a representation of the system has been built in order to fulfill every necessary step. Please review the following diagram and, if you think it is needed, complete it, in order to be satisfying according to the system's description.

In order for the system to work properly, some communication capabilities are pre-defined, that include ***Goals***, ***Beliefs*** and ***Plans***. The system has two ***Goals*** pre-defined: Find the correct Agent and find the appropriate Semantic Web Services (SWSs). ***Beliefs*** are determined by Family or Friendly Knowledge, based on previous experiences.

For these ***Goals*** to be correctly accomplished some ***Plans*** have to be executed. Services that allow searching for other agents should be correctly registered, the connections between services should be correctly defined and there should be a detailed and defined plan to find an agent (two ***Roles***: Search Adequate Service and Search Ordered Person). Each of these plans have two different ***Behaviors*** on the system (one Behavior connects to one Plan – Finding a Service for Establishing Connection and Finding an Agent to Finding a Person): to find the correct service for the search **AND** to find the agent that is being searched **(two different Behaviors**). Each of these behaviors also connects to one ***Role*** (Finding a Service to Search Adequate Service and Finding an Agent to Search Ordered Person).

User A (that wants to search for some other User) has Agents working in order to find the person he is looking for. Agents may be on an Active state (they are looking for the person) – represented by an ***Agent State*** - and should be from the Personnel type (internal agents, working only for this system) – represented by an ***Agent Type***. On an active state, Agents are addressing the previously defined goals. Each Agent type may play two different ***Roles***: Search Adequate Service and Search Ordered Person.

EXERCISE 4:
Based on the description of the system proposed below, a representation of the system has been built in order to fulfill every necessary step. Please review the following diagram and, if you think it is needed, complete it, in order to be satisfying according to the system's description.

In order for the system to work properly, the system needs Communications (represented by a ***SWS***), that has a ***Link*** described as "depends_on" connecting to Search Service Ontology (represented by a ***Service Ontology***).

The Search Service Ontology is also connected to the Ontology Manager (represented by an ***Ontology Mediator Role***), with a connection "knows service" being linked between them (from Ontology Manager to Search Service Ontology). Alongside, Search Service Ontology also connects to General Knowledge (represented by a ***Belief***) and Family Facts (represented by a ***Fact***), being linked through "includes".

The General Knowledge referred above also connect to Search Role Ontology (represented by a ***RoleOntology***) and a Communication Organization Ontology (represented by an ***Organization Ontology***), being linked through "includes".

The Family Facts referred above also connect to Search Role Ontology and Communication Organization Ontology.

The Search Role Ontology referred above also connects to Person Finder (represented by a ***Role***). Person Finder also connects to Communication Organization Ontology. The connection between them is represented by "knowsOrgOnt".

The Communication Organization Ontology also connects to Communication Web Organization (represented by a ***Semantic Web Organization***), with the connection between them being represented by "knowsOrg".

Appendix B

Letter of Consent and Profile Data Inquiry

In this appendix we presented the Letter of Consent that users should accept in order to be part of any of the experiments and the Profile Data questionnaire that users solve after each experiment is presented.

Letter of Consent

This experimental work is conducted within the NOVA Laboratory for Computer Science and Informatics (NOVA LINCS). NOVA LINCS is a new unit of the national Science & Technology network in the area of Computer Science and Engineering, launched in 2014/2015, and hosted at the Departamento de Informática of Faculdade de Ciências e Tecnologia of Universidade Nova de Lisboa (DI-NOVA), a leading academic department in Portugal.

All information stated as part of this experiment is confidential and will be kept as such.
Prof. Vasco Amaral and Prof. Miguel Goulão are responsible for this experiment and can be contacted at:

Prof. Vasco Amaral: vasco.amaral@fct.unl.pt; +351 212 948 300 (ext. 10712); Office P2/3
Prof. Miguel Goulão: mgoul@fct.unl.pt; +351 212 948 536 (ext. 10731); Office P2/17

We would like to emphasize that:
- Your participation is entirely voluntary;
- You are free to refuse to answer any question;
- You are free to withdraw at any time.

The experiment will be kept strictly confidential and will be made available only to members of the research team of the study or, in case external quality assessment takes place, to assessors under the same confidentiality conditions. Data collected in this experiment may be part of the final researc
h
report, but under no circumstances will your name or any identifying characteristic be included in the report.

*Obrigatório

1. **I accept the terms addressed above:** *
Marcar apenas uma oval.

( ) Yes
( ) No

Profile Data

*Obrigatório

1. **Gender** *

Marcar apenas uma oval.

- ( ) Female
- ( ) Male

2. **Age** *

Marcar apenas uma oval.

- ( ) 17-22
- ( ) 22-25
- ( ) 26+

3. **Nationality** *

____

4. **Field of Studies** *

Please select your field of studies. If none of the below defines it correctly, please choose the option "other" and write the correct description.

Marcar apenas uma oval.

- ( ) Computer Science
- ( ) Electronic Engineering
- ( ) Outra: ______________________________

5. **Completed Education** *

Marcar apenas uma oval.

- ( ) High School
- ( ) BSc
- ( ) MSc
- ( ) PhD
- ( ) Post-Doc

6. **Current Occupation** *

Marcar tudo o que for aplicável.

- [ ] Student
- [ ] Researcher
- [ ] Worker

7. **Previous Experience With Multi-Agent Systems (MAS)** *
Marcar apenas uma oval.

- I've learned MAS in the context of a course.
- I've used MAS in a professional context
- I know what MAS is but never used it
- I've never heard of it.

8. **Previous Experience With the Semantic Web (SW)** *
Marcar apenas uma oval.

- I've learned WS in the context of a course.
- I've used WS in a professional context
- I know what WS is but never used it
- I've never heard of it.

9. **If you want to receive the final results, please provide us your email:**

____

Appendix C

"SEA_ML original notation VS SEA_ML new notation" experiment inquiries

In this appendix are presented the questionnaires for the "SEA_ML original notation VS SEA_ML new notation" experiment.

Participants reply to a questionnaire for each of the exercises that involve the experiment (4 questionnaires in total).

As this experiment has four different versions, each version presents a different order for this questionnaires:

1. — **Version 1** — Participants start on the case study "Music Trader", finishing on the "Expert Finding" case study. The first case study is modeled with the new notation, while the second case study is modeled with the original notation;

2. — **Version 2** — Participants start on the case study "Expert Finding", finishing on the "Music Trader" case study. The first case study is modeled with the original notation, while the second case study is modeled with the new notation;

3. — **Version 3** — Participants start on the case study "Music Trader", finishing on the "Expert Finding" case study. The first case study is modeled with the original notation, while the second case study is modeled with the new notation;

4. — **Version 4** — Participants start on the case study "Expert Finding", finishing on the "Music Trader" case study. The first case study is modeled with the new notation, while the second case study is modeled with the original notation;

CS1: Music Trader

Exercise 1

*Obrigatório

1. **I think that I would like to use this system frequently.** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

2. **I found the system unnecessarily complex.** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

3. **I thought the system was easy to use.** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

4. **I think I would need the support of a technical person to be able to use this system.** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

5. **I found the various functions in this system were well integrated.** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

6. **I thought there was too much inconsistency in this system.** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

7. **I would imagine that most people would learn to use this system very quickly.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

8. **I found the system very cumbersome (i.e. difficult) to use.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

9. **I felt very confident using the system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

10. **I needed to learn a lot of things before I could get going with this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

11. **Suggestions**

____
____
____
____
____

CS1: Music Trader

12. **The symbols on the user interface (UI) were easy to understand** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

13. **The symbols on the UI are adequate to the constructions they are linked to** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

14. **The symbols on the UI helped me solve the exercise in less time** *

Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

CS1: Music Trader

Exercise 2

*Obrigatório

1. **I think that I would like to use this system frequently.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

2. **I found the system unnecessarily complex.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

3. **I thought the system was easy to use.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

4. **I think I would need the support of a technical person to be able to use this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

5. **I found the various functions in this system were well integrated.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

6. **I thought there was too much inconsistency in this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

7. **I would imagine that most people would learn to use this system very quickly.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

8. **I found the system very cumbersome (i.e. difficult) to use.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

9. **I felt very confident using the system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **I needed to learn a lot of things before I could get going with this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

11. **Suggestions**

CS1: Music Trader

12. **The symbols on the user interface (UI) were easy to understand** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

13. **The symbols on the UI are adequate to the constructions they are linked to** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

14. **The symbols on the UI helped me solve the exercise in less time** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

CS2: Expert Finding

Exercise 3

*Obrigatório

1. **I think that I would like to use this system frequently.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

2. **I found the system unnecessarily complex.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

3. **I thought the system was easy to use.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

4. **I think I would need the support of a technical person to be able to use this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

5. **I found the various functions in this system were well integrated.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

6. **I thought there was too much inconsistency in this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

7. **I would imagine that most people would learn to use this system very quickly.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

8. **I found the system very cumbersome (i.e. difficult) to use.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

9. **I felt very confident using the system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **I needed to learn a lot of things before I could get going with this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

11. **Suggestions**

CS2: Expert Finding

12. **The symbols on the user interface (UI) were easy to understand** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

13. **The symbols on the UI are adequate to the constructions they are linked to** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

14. **The symbols on the UI helped me solve the exercise in less time** *

Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

CS2: Expert Finding

Exercise 4

*Obrigatório

1. **I think that I would like to use this system frequently.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	( )	( )	( )	( )	( )	Strongly Agree

2. **I found the system unnecessarily complex.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	( )	( )	( )	( )	( )	Strongly Agree

3. **I thought the system was easy to use.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	( )	( )	( )	( )	( )	Strongly Agree

4. **I think I would need the support of a technical person to be able to use this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	( )	( )	( )	( )	( )	Strongly Agree

5. **I found the various functions in this system were well integrated.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	( )	( )	( )	( )	( )	Strongly Agree

6. **I thought there was too much inconsistency in this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	( )	( )	( )	( )	( )	Strongly Agree

7. **I would imagine that most people would learn to use this system very quickly.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

8. **I found the system very cumbersome (i.e. difficult) to use.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

9. **I felt very confident using the system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **I needed to learn a lot of things before I could get going with this system.** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

11. **Suggestions**

____
____
____
____
____

CS2: Expert Finding

12. **The symbols on the user interface (UI) were easy to understand** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

13. **The symbols on the UI are adequate to the constructions they are linked to** *
Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

14. **The symbols on the UI helped me solve the exercise in less time** *

Marcar apenas uma oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

Appendix D

"Selecting a visual notation for SEA_ML" experiment questionnaire

In this appendix we present the questionnaire for the "Selecting the best visual notation for SEA_ML" experiment.

Since SEA_ML has 43 (44 including the arrows that relate each entity) different constructions, users are only defining symbols for the constructions that have been modified on chapter 4 (33 different constructions in total).

Matching Concepts and Figures

Match the figures in the table of figures (AVAILABLE HERE: https://goo.gl/QTps8C) with the concepts described below.

Insert the number of the figure you find more suitable for each of the concepts described below (Please choose only one figure per concept).

Each figure can match one or more concepts.

*Obrigatório

1. **GOAL: A goal is a desire that has been adopted for active pursuit by the agent.** *

2. **CAPABILITY: Taking BDI agents into consideration, there is an entity called Capability which includes each agent's Goals, Plans and Beliefs about the surroundings.** *

3. **FACT: The statement about the agent's environment which can be true. Agents can decide based on these facts.** *

4. **PLAN: Plans are sequences of actions that an agent can perform to achieve one or more of its intentions.** *

5. **SEMANTIC SERVICE REGISTER PLAN: The Semantic Service Register Plan (SS_RegisterPlan) is the plan used to register a new SWS by SSMatchmakerAgent.** *

6. **SEMANTIC SERVICE FINDER PLAN: Semantic Service Finder Plan (SS_FinderPlan) is a Plan in which automatic discovery of the candidate semantic web services take place with the help of the SSMatchmakerAgent.** *

7. **SEMANTIC SERVICE AGREEMENT PLAN: Semantic Service Agreement Plan (SS_AgreementPlan) is a concept that deals with negotiations on quality of service (QoS) metrics (eg, service execution cost, duration and position) and contract negotiation.** *

8. **SEMANTIC SERVICE EXECUTOR PLAN: After service discovery and negotiation, the agent applies the Semantic Service Executor Plan (SS_ExecutorPlan) to invoke appropriate semantic web services.** *

9. **SEND: An action to transmit a message from an agent to another. This can be based on some standard such as FIPA_Contract_Net** *

10. **RECEIVE: An action to collect a message from an agent. This can be based on some standard such as FIPA_Contract_Net** *

11. **TASK: Tasks are groups of actions which are constructing a plan in an agent.** *

12. **ACTION: An action is an atomic instruction which constitutes a task.** *

13. **MESSAGE: A package of information to be send from an agent to another; possibly to deliver some information or instructions. Two special types of actions, namely Send and Receive, are used to handle these messages.** *

14. **MESSAGE SEQUENCE: A series of message to be applied to realize a role.** *

15. **ODMOWLCLASS: A class of ontology to be used in the multi agent system.** *

16. **DOMAIN ROLE: A type of agent role which is dedicated to a specific domain, such as buyer or seller roles.** *

17. **AGENT STATE: This concept refers to certain conditions in which agents are present at certain times. An agent can only have one state (Agent State) at a time, e.g. waiting state in which the agent is passive and waiting for another agent or resource.** *

18. **RESOURCE: It refers to the system resources that the MAS is interacting with. For example, the database.** *

Matching Concepts and Figures

Match the figures in the table of figures (AVAILABLE HERE: https://goo.gl/QTps8C) with the concepts described below.

Insert the number of the figure you find more suitable for each of the concepts described below.

Each figure can match one or more concepts.

19. **WEB SERVICE: Type of service which is presented via web.** *

20. **PROCESS: It describes how the SWS is used by defining a process model. Instances of the SWS use the process via described_by to refer to the service's ServiceModel.** *

21. **INTERFACE: This document describes what the service provide for prospective clients. This is used to advertise the service, and to capture this perspective, each instance of the class Service presents a Service Interface.** *

22. **GROUNDING: In this document, it is described how an agent interact with the SWS. A grounding provides the needed details about transport protocols. Instances of the class Service have a supports property referring to a Service Grounding.** *

23. **PRECONDITION: Defines the pre-conditions for processes and interfaces of a SWS.** *

24. **EFFECT: Defines the post-conditions or effects for processes and interfaces of a SWS.** *

25. **ARCHITECTURE ROLE: The roles may be used in the architectural aspect of the multi-agent systems.** *

26. **ONTOLOGY MEDIATOR ROLE: This role is mediating between different ontologies.** *

27. **SEMANTIC WEB ORGANIZATION: Refers to an organized group of semantic web agents (SWAs).** *

28. **ROLE ONTOLOGY: Demonstrates the ontology of roles in the MAS. Proximity relationships of roles in organizations can be created with this concept.** *

29. **ORGANIZATION ONTOLOGY: Demonstrates the ontology of organizations in the MAS. The association of the organizations in MAS can be shown with this ontology.** *

30. **SERVICE ONTOLOGY: It refers to the ontology of the services in the MAS. The semantic relationship between the services is specified by this ontology.** *

31. **INTERACTION: For communication and collaboration of agents, they can use series of messages via a message sequence which results to an agent interaction.** *

32. **BEHAVIOR: In re-active agents, a behavior is a re-action of an agent towards an external or internal stimuli.** *

33. **AGENT TYPE: The agents in a multi-agent system can have different types taking various responsivities and representing various stakeholders.** *

Profile Data

34. **Gender** *
Marcar apenas uma oval.

- Female
- Male

35. **Age** *
Marcar apenas uma oval.

- 17-22
- 22-25
- 26+

36. **Nationality** *

37. **Field of Studies** *
Marcar apenas uma oval.

- Computer Science
- Electronic Engineering
- Outra: ____________

38. **Completed Education** *
Marcar apenas uma oval.

- High School
- BSc
- MSc
- PhD
- Post-Doc

39. **Current Occupation** *
Marcar tudo o que for aplicável.

- Student
- Researcher
- Worker

40. **Experience With Multi-Agent Systems (MAS)** *

Marcar apenas uma oval.

- ( ) I've learned MAS in the context of a course.
- ( ) I've used MAS in a professional context
- ( ) I know what MAS is but never used it
- ( ) I've never heard of it.

41. **Previous Experience With the Semantic Web (SW)** *

Marcar apenas uma oval.

- ( ) I've learned WS in the context of a course.
- ( ) I've used WS in a professional context
- ( ) I know what WS is but never used it
- ( ) I've never heard of it.

42. **If you want to receive the final results, please provide us your email:**

____
