# Agents negotiating in a Semantic Web Architecture (SWA)

**Rosa María Gil Iranzo**

Departament de Tecnologia
Universitat Pompeu Fabra

Doctorat en informàtica i Comunicació Digital
2004

Dipòsit legal:  B.14579-2005
ISBN: 84-689-1256-5

This thesis entitled:

Agents negotiating in a Semantic Web Architecture (SWA)
written by Rosa Maria Gil Iranzo
has been approved for the '*Departament de Tecnologia*'

*Directed by: Dr. Jaime Delgado*

Date: November 2004

A thesis submitted to the

'*Departament de Tecnologia
de la Universitat Pompeu Fabra*'

In partial fulfillment
Of the requirements for the degree of

*'Doctor per la Universitat Pompeu Fabra'*

Gil Iranzo, Rosa María (Doctorat en Informàtica i Comunicació Digital)
Agents negotiating in a Semantic Web Architecture (SWA)
Thesis directed by Dr. Jaime Delgado
Date: November 2004

# Abstract

The main issue of this work is to discover and face new challenges in negotiation over the World Wide Web network, concretely over the Semantic Web because it provides a new paradigm not only in language expression but also in its manipulation.

The first step has been to design an architecture allowing agents, which are modelled using Multi-Agent Systems, to negotiate in the Semantic Web. This architecture was tested inside the IPR (Intellectual Property Rights) context, concretely in the ***NewMars***[160] and ***AgentWeb***[161] projects.

As a result, a heterogeneous architecture is provided (with different Multi-agent Systems) and IPR knowledge is formalized in an IPR ontology (***IPROnto***). This ontology was submitted to MPEG-21 standardization process as a valuable tool to enable Multi-agent Systems IPR negotiation in the Semantic Web.

Nowadays, information must be available when it is required. This means that agents, in order to efficiently exploit information, have to know about the other agents and their environments. Ontologies have been used to model agents' knowledge. However, information is missing. What does it happen in the case of an agent is looking for something that is not related to any prior knowledge? How is it possible to create this relation and embrace new knowledge? In order to provide a model of the Semantic Web as real as possible, a deep statistical analysis of it has been made. It reveals that the Semantic Web behaves as a complex system and shares some properties with them. This can be used to study new ways of designing semantics-enabled applications. In this sense, Semantic Web can be modelled as a whole system and *macroscopic* behaviour can be established.

Thus, environment and agents can be analysed statistically and get new knowledge from their interactions. Then, what can we do with it? The quick answer is to build new ontologies and to store all this information in a database. However, concepts classification could not be so objective as we expect. To clarify this question a new research line, which goes deeper in the process of how an agent can learn in the Semantic Web, is explored. It was born from the effort that Semantic Web community is doing to establish a shared knowledge base for common understanding (***upper ontologies***). The contribution relates it to the physical domain (***space-time***), which is exemplified in the negotiation process.

# Dedication

*To Roberto*

# Acknowledgments

Thanks to my thesis director for his help. Zièd Chouckair’s advices were very useful at ENST-Bretagne. Every member of Technology Department at UPF has helped when I asked him/her: Jesús Ibañez ( *a fount of wisdom* ), Núria Garcia and Laura Igual ( for her *unconditional support* ), Nadjet Bouayad ( getting deeper in Zipf’s law ), Victor Dalmau ( ‘*emergence of a giant ’*), Jordi Faro ( *getting deeper in graphs* ), Ana Escudero ( ‘*wireless networks’ )* ,Vicen Gómez (*neural networks*), Daniel Giribert ( ‘*FlashMX2004’* ), Eva Rodriguez (*MPEG-21 issues*), Ruben Tous ( *critic point of view* ), Lluis Garrido ... I can say that list is too large…sorry if I forget someone.

Joseph Hilferty, R.V. Solé, R.C. i Ferrer helped me to find new ways to face learning from different points of view. Just to integrate methods and knowledge from different disciplines.

This thesis is dedicated to Roberto. It could not exist without him. The amount of greatest things he has done is too large to be put here. Just many thanks. I only hope not to disappoint you.

# Contents

# 1. Figure list

# 2. Table list

# 3. Introduction

When human beings deal with negotiation processes, many times the agreement is not achieved. The goal of this work is to perform it as good as possible and how it can be improved.

Unfortunately, this fact happens in legal domain. Abstract domain allows ambiguity in a degree that enables a common understanding sometimes.

Our researching group has a long experience working on standards, and our first attempt was to send a proposal to an international standard, the intention was clear: be unambiguous as we can.

However, we have gone further, trying to find something common not only in the abstract domain but also in the physical domain.

The meaning of negotiation concept and the identification it has with process concept and how it has been developed in different contexts as economics or technological is the first issue to be dealt in the State of the art. Moreover, it facilitates a guide and a short review of the technologies that are going to be used along this work for designing a distributed negotiation architecture.

The aim is to describe how negotiation process is performed by artificial agents which represent users. Agents are managed by Multi-Agent systems inside an architecture that involves inference, using Expert Systems (JESS), and semantics, using the IPROnto IPR ontology. IPROnto has been developed and submitted to MPEG-21 standardisation process. It is validated in the context of Intellectual Property Rights (IPR) negotiation, concretely in the NewMARS project. Section 5 describes it.

Section 5.9 focuses in customisation, it means that artificial agents adapt themselves to their environment as the customer device and profile detail. Moreover, peer to peer (P2P) technology was aggregated to this system to become an open distributed system.

In order to provide architecture interoperability inside Semantic Web, section 6.9 provides an exhaustive statistical analysis over the network knowledge represented by ontologies used to develop negotiation processes. Patterns that appear to be common in some networks are used to provide a way to deal huge amounts of information. The robustness architecture has been tested because of using networks that are scale free. Section 7.1 takes care about the reasons why this analysis can be done.

The last part, section 7, shows how is it possible to achieve semantics from scratch and tries to come some light to the fact of finding common knowledge.

# 4. State of the Art

This part is in charge of providing the survey around this work makes sense. It is remarkable that there are some disciplines that come from a wide scope: ***computer science technologies*** (see sections 4.2.1 to 4.2.5 and section 4.4.5), ***complex systems and statistical analysis*** (see sections 4.4.2 to 4.4.4), ***linguistics*** (see section 4.5 and 4.6) and ***mathematics*** (see sections 7.1 and 7.2). It is done in this way because '***negotiation***' is not only an abstract concept as it is going to be developed and it comes clear when this process has to be conceptualised and afterwards implemented in human real life. Legal and economic systems as IPR (Intellectual Property Rights) systems represent a challenge when a model of negotiation is necessary because too many parameters and variables are uncontrollable or even unknown, approximations coming from disciplines as economics try to model this process.

This is a review of relevant concepts that are going to be worked in the definition of a negotiation process. As an initial condition: ***negotiation***, the title concepts core has to be defined.

## 4.1. What is negotiation?

Usually, when there is an unknown concept or a list of unknown concepts, a dictionary or thesaurus comes to help. Thus, the first step to analyse the question is to look for them. For example, just looking for '*negotiation*':

"***A discussion intended to produce an agreement*** ".
WEBNOX CORP. Thesaurus

"***To confer (with another) with a view to agreement***".
Concise Oxford Dictionary

It is not specified who are the agreement actors and how is performed this consensus in the definitions however they are: actors and a goal (agreement). As a first approximation, negotiation definitions do not give an idea of its internal complexity. It seems that it is ***something dynamic created by some actors with an intention or goal called agreement.***

### 4.1.1. Is Negotiation a process?

The following question is putting boundaries about *what really it is* or to relate this concept to something known. Usually, it is related to the process concept, this section goes deeper in seeing why:

In WordNet 2.0 [1] the *noun process* has 6 senses:

1. (63) ***procedure, process*** -- (a particular course of action intended to achieve a result; "the procedure of obtaining a driver's license"; "it was a process of trial and error")
2. (42) ***process*** -- (a sustained phenomenon or one marked by gradual changes through a series of states; "events now in process"; "the process of calcification begins later for boys than for girls")
3. (3) ***process, cognitive process, mental process, operation, cognitive operation*** -- ((psychology) the performance of some composite cognitive activity; an operation that affects mental contents; "the process of thinking"; "the cognitive operation of remembering")
4. ***summons, process*** -- (a writ issued by authority of law; usually compels the defendant's attendance in a civil suit; failure to appear results in a default judgment against the defendant)
5. ***process, unconscious process*** -- (a mental process that you are not directly aware of; "the process of denial")
6. ***process, outgrowth, appendage*** -- (a natural prolongation or projection from a part of an organism either animal or plant; "a bony process")

While the verb process has 7 senses:

1. (268) ***process*** -- (deal with in a routine way; "I'll handle that one"; "process a loan"; "process the applicants")
2. (121) ***process, treat*** -- (subject to a process or treatment, with the aim of readying for some purpose, improving, or remedying a condition; "process cheese"; "process hair"; "treat the water so it can be drunk"; "treat the lawn with chemicals" ; "treat an oil spill")
3. (36) ***process*** -- (perform mathematical and logical operations on (data) according to programmed instructions in order to obtain the required information; "The results of the elections were still being processed when he gave his acceptance speech")
4. (4) ***action, sue, litigate, process*** -- (institute legal proceedings against; file a suit against; "He was warned that the district attorney would process him"; "She actioned the company for discrimination")
5. (2) ***work, work on, process*** -- (shape, form, or improve a material; "work stone into tools"; "process iron"; "work the metal")
6. (1) ***serve, process, swear out*** -- (deliver a warrant or summons to someone; "He was processed by the sheriff")

7***. march, process*** -- (march in a procession; "They processed into the dining room")

After looking at the results, the following features definitions claim attention:

The goal could be incorporated when there is a **purpose** (ex: a particular course of action intended to achieve a result).
**Dynamics** (ex: a sustained phenomenon or one marked by gradual changes through a series of states…)
Its definitions go from **abstract domain** (ex: mental, legal proceedings …) until **physical domain** (ex: particular course of action…)

Thus, a first premise is taken: negotiation will be interpreted as ***a process*** as a first approximation because provides ***dynamics*** plus a goal (***understood as agreement***). Section 7 will explore the last feature (***abstract to physical domain***)

Dynamics relates directly to time dimension; thus, is it possible to go beyond in the process definition, just looking at its relation to ***time?*** See Figure 1, this ontology (see section 4.5 for ontology definition) created by J. Sowa defines different types of processes and their first classification depending on **time**: ***ContinuousProcess*** vs ***DiscreteProcess*** were the first has one or more open time intervals while the second has closed time intervals. Curiously, time makes the difference again in the second classification of ***ContinuousProcess***. ***Initiation*** and ***Cessation*** differ in the close time interval position, at the beginning or at the end. On the other hand, ***Continuation*** has no specification about when time begins or ends. The case of ***DiscreteProcess*** is something different because time makes no difference between ***State*** and ***Event***, both have a closed time interval. However, there is nothing that changes in the first and something that does in the second.

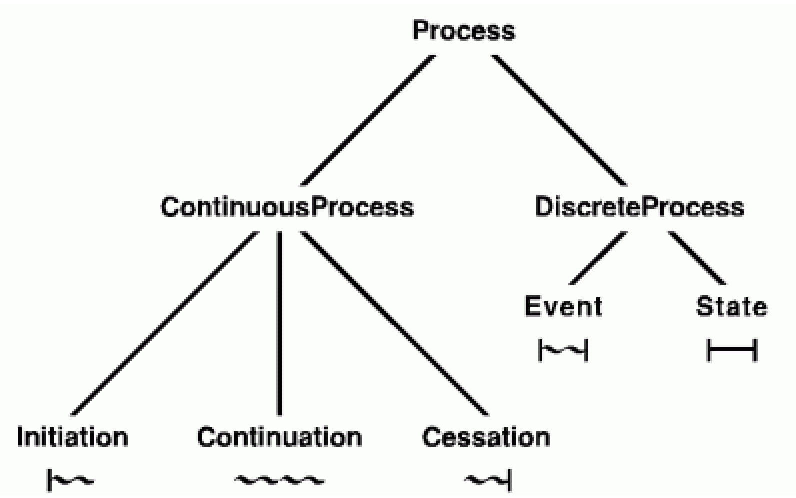



**Figure 1 Types of process (J. Sowa)**[2]

In addition to WordNet definitions, this ontology shows its relations to ***time and things that happen in time***.

Thus, negotiation can be understood as a process and the properties associated to it can be used to define the concept. In the following paragraphs the relation of negotiation with the different domains as telecom or economics is analysed.

### 4.1.2. Negotiation in Telecom domain

As a second approximation in order to discover what negotiation is, the telecom technology domain is studied. Some main components are present when it is modelled:

- ***Actors/roles***: Actors perform negotiation to achieve the agreement. Actors can have different roles. For example the same actor can have different roles as *buyer* or *seller* depending on the context. They exchange message whose elements are:
- ***Communication acts*** describes negotiation actions as *cancel, agree or inform*
- ***Content*** : message information
- ***Language*** defined used as encoding and ontology-semantics
- ***Protocol*** using id's as conversation-id, reply-with, in-reply-to or reply-by.
- ***Policy***: It is called also negotiation rules to be made depending on the negotiation events. Intelligence is currently being developed at this point with inference systems.
- ***Environment***: It is known also as context. It is the framework where the negotiation process is held.

Agents perform negotiation and usually they build communities called MAS (Multi-Agent Systems). The most representative systems are described in section 4.2. This study allows determining the capabilities of this kind of technology.

This is the view from telecom domain. However, it relates to other domains where negotiation is essential. The economy domain should be highlighted because there is an inherent economic value in negotiation. There are three main tendencies that are going to be analysed: **game-theory, heuristics and planning** and **ABN** (Argumentation based negotiation). All of them use tools taken from *science* as mathematics and *computer science*.

In the sections 4.1.3, 4.1.4 and 4.1.5 a state of the art overview about economics [3] is developed making a discussion of advantages and disadvantages of the current different analysis.

### 4.1.3. Negotiation in Economics: Game -Theoretic analysis

Game theory is the formal study of conflict and cooperation. The earliest example of a formal game-theoretic analysis is the study of a duopoly by Antoine Cournot in 1938; afterwards Emile Borel suggested a formal theory of games in 1921, which was furthered by the mathematician John von Neumann in 1928 in the “***theory of parlour games***”.

Game-theoretic analysis was widely developed in 1950 by John Nash, it was demonstrated that ***finite games have always have an equilibrium point*** and the optimal strategy is found by analysing the interaction as a game between ***identical participants***, and seeking its equilibrium[4]. The strategy determined by these methods can sometimes be made to be **optimal for a participant**, given the game rules, the assumed payoffs, and the goals of the participants, and assuming that the participants have no knowledge of one another not provided by introspection. Assuming further that participants behave according to the assumptions of **rational-choice theory**[5], this approach can guide the design of the interaction mechanism itself, and thus force such agents to behave in certain ways[6]. This means agents may be ***resource constrained***, altruistic, malicious, or simply badly-coded, so that participant behaviour may not conform to the assumptions of rational choice theory.

Classical game-theoretic approaches have some significant limitations from the computational perspective [7]: Specifically, most of these approaches assume that agents have **unbounded computational resources** and that the **space of outcomes is completely known**. In most realistic environments, however, these assumptions fail due to the limited processing and communication capabilities of the information systems. However it is applied in building efficient bidding rules for auction websites[8], or tamper-proof automated negotiations for purchasing communication bandwidth.

Frame problem is the key issue in this approximation because the solution is perfect; however the problem is how to fit real world:

Unbounded computational resources are totally impossible, because everything is finite, sometimes very big but finite after all.
The identical participants approximation is only true when no differences appear. However, it is not true for people.
The space of outcomes is only completely known when the problem is totally bounded, for instance in auction websites.
Finally, knowledge about other agents is key when some decisions have to be made about our acts.

### 4.1.4. Negotiation in Economics: Heuristic and planning

Heuristics are rules of thumb that produce good enough outcomes and are often produced in contexts with more relaxed assumptions about agents' rationality and resources. They act as a layer making better the game-theoretic studies using empirical testing and evaluation[9]. In general, these methods ***offer approximations to the decisions made according to game-theoretic studies*** where ***various heuristic decision functions are used for evaluating and generating offers or proposals*** in multi-attribute negotiation. A growing research area in economics that addresses some of the limitations of conventional models is ***evolutionary game theory*** in which the assumption of ***unbounded rationality*** is relaxed. In evolutionary models, games are played repeatedly, and strategies are tested through a ***trial-and-error*** learning process in which players gradually discover that some strategies work better than others. However, other assumptions, such as the availability of a preference valuation function, still hold. Another attempt is the modelling of '***bounded rationality***' by explicitly capturing elements of the process of choice, such as limited memory, limited knowledge, approximate preferences. It means that new constrains are added to the model.

A negotiation model with decision procedures based on ***distributed constraint satisfaction***[10], extended to allow for multiple concurrent negotiations and to accommodate ***fuzzy constraints***[11] and it continues further investigated. While heuristic methods do indeed overcome some of the shortcomings of game-theoretic approaches, they also have a number of disadvantages[12]:

Firstly, the ***problem of space of outcomes*** and ***behaviour*** is not solved yet. The models often lead to outcomes that are sub-optimal because they adopt an approximate notion of rationality and because they do not examine the full space of possible outcomes. And secondly, it is very difficult to predict precisely how the system and the constituent agents will behave.

Consequently, the models need extensive evaluation through simulations and empirical analysis.

In most game-theoretic and heuristic models, agents exchange proposals, i.e. potential agreements or potential deals. Agents ***are not allowed*** to exchange any additional information other than what is expressed in the proposal itself. This can be problematic, for example, in situations ***where agents have limited information about the environment***, or where their ***rational choices depend on those of other agents***. This is typically the case, for instance, with network goods such as printers. Here, the value of a printer to one agent depends on whether or not other agents have printers.

Another limitation of conventional approaches to automated negotiation is that agent's utilities or preferences are usually assumed to be completely characterised ***prior to*** the interaction. Thus an agent is assumed to have a mechanism by which it can assess and compare any two proposals. This may be easy, for example, when the utility of the negotiation object is defined in terms of an economic value, such as the price of a song in Internet. An agent can compare proposals of two service providers by simply comparing how much they charge per minute. However, there are more *complex negotiation* situations like charge per second, per computer, per software license or even per device as a CD. Agents may well have incomplete information which limits this capability. Thus, agents might:

- ***Lack some of the information*** relevant to making a comparison between two potential outcomes,
- Have ***limited resources*** preventing them from acquiring such information,
- Have the information, but ***lack the time*** needed to process it in order to make the comparison,
- Have inconsistent or uncertain beliefs about the environment,
- Have unformed or undetermined preferences (e.g., about products new to them), or have incoherent preferences.

Thus, to overcome these limitations, the process of acquiring information, resolving uncertainties, revising preferences, etc. often takes place as part of the negotiation process itself, which it means that it has to be solved 'on-line'

A further drawback of traditional models to automated negotiation is that agents' preferences over proposals are often assumed to be proper in the sense that they reflect the true benefit the agent receives from satisfying these preferences.

For example, an agent attempting to purchase a DVD device might assign a high value to a particular brand according to its belief that this brand makes good material than other brands. If this belief is false, then the preferences do not properly reflect the agent's actual gain if it was to purchased that DVD. Finally, game-theoretic and heuristic approaches assume that agents' utilities or preferences ***are fixed***. One agent ***cannot directly influence*** another agent's preference model, or any of ***its internal mental attitudes*** (e.g., beliefs, desires, goals, etc.) that generate its preference model. A rational agent would only modify its preferences upon receipt of new information. ***Traditional automated negotiation mechanisms do not facilitate the exchange of this information***.

Against this background, argumentation-based approaches to negotiation attempt to overcome the above limitations by allowing agents to exchange additional information, or to "argue" about their beliefs and other mental attitudes during the negotiation process. In the context of negotiation, it is needed an argument as a piece of information that may allow an agent to justify its negotiation stance or influence another agent's negotiation stance

Thus, in addition to accepting or rejecting a proposal, an agent can offer a critique of it. This can help make negotiations more efficient. By understanding why its counterpart cannot accept a particular deal, an agent may be in a better position to make an alternative offer that has a higher chance of being acceptable.

In a legal dispute, for example, an agent representing the customer might refuse an offer for a modified plan made by the organisation's management agent[13]. As a response, the management agent might offer a different plan and be concentrated on finding an arrangement for workload reduction.

Another type of information that can be exchanged is a ***justification of a proposal***, stating why an agent made such a proposal or why the counterpart should accept it, or the nature of the negotiation space itself. It means that perhaps something completely different can be a good deal if a justification of it is understood as a good deal by the agent. For example, a provider negotiating a price raise might propose a big increase that gets rejected by the multimedia distributor. After the provider justifies the proposal by denoting her significant achievements during the year, the multimedia distributor might accept. Agents may also exchange information that results in changing the negotiation object itself, by introducing new dimensions to the negotiation object. For example, the multimedia distributor might modify the negotiation object such that the negotiation involves not only the price, but also the number of copies.

### 4.1.5. Argumentation-Based Negotiation framework

From the discussion it should be clear that ***there is no universal approach to automated negotiation that suits every problem domain***. Rather, there is a set of approaches, each based on different assumptions about the environment and the agents involved in the interaction. Argumentation-based negotiation (ABN) frameworks, is gaining increasing popularity for its potential ability to overcome the limitations of more conventional approaches to automated negotiation. However, such models are typically more complex than their game-theoretic and heuristic counterparts.

Against this background, the aim of this analytical survey is to identify the main components of an abstract framework for ABN.

In ABN frameworks, agents need ***richer communication*** and domain languages to be able to exchange ***meta-level information*** (i.e., information other than that describing outcomes). Therefore, a major distinguishing factor of ABN frameworks is in the type of information that can be expressed and exchanged between agents, and consequently, in the specifications of the agents that generate and evaluate this information.

| | Non-ABN Frameworks | ABN Frameworks |
|---|---|---|
| Domain Language | Expresses proposals only (e.g., by describing products available for sale). | Expresses proposals as well as meta-information about the world, agent's beliefs, preferences, goals, etc. |
| Communication Language | Locutions allow agents to pass call for bids, proposals, acceptance and rejection, etc. | In addition, locutions allow agents to pass meta-information either separately or in conjunction with other locutions. |

**Figure 2 Differences between ABN and Non-ABN w.r.t Domain and Communication Languages**

For instance, Physical Agents' Agent Communication Language (FIPA ACL) offers 22 locutions (see section 4.2.1).

There are a number of challenges in the design of domain and communication (see Figure 2) languages for argumentation-based negotiation. First, there is a need to provide rich communication languages with ***clear semantics***. In multi-agent systems, ***dialogue games*** have been used to ***specify dialogue protocols*** for negotiation.

Given a communication and domain language, a negotiation framework (see Figure 3 and Figure 4 where the dashed lined boxes represent the additional components necessary for ABN agents) should also specify a negotiation protocol in order to constrain the use of the language.

The interaction protocol specifies, at each stage of the negotiation process, who ***is allowed to say what***. For example, after one agent makes a proposal, the other agent may be able to accept it, reject it or criticise it, but might not be allowed to ignore it by making a counterproposal. The protocol might be based solely on the last utterance made, or might depend on a more complex history of messages between agents.

The other rules that form part of the negotiation protocol may address the following issues[14]:

***Rules for admission***: specify when an agent can participate in a negotiation dialogue and under what conditions.
***Rules for participant withdrawal***: specify when a participant may withdraw from the negotiation.
***Termination rules***: specify when an encounter must end (e.g. if one agent utters an acceptance locution).
***Rules for proposal validity***: specify when a proposal is compliant with some conditions (e.g., an agent may not be allowed to make a proposal that has already been rejected).
***Rules for outcome determination***: specify the outcome of the interaction. In an auction-based framework, this would involve determining the winning bid(s). In argumentation-based frameworks, these rules might enforce some outcome based on the underlying theory of argumentation
***Commitment rules***: specify how agents' commitments should be managed, whether and when an agent can withdraw a commitment made previously in the dialogue, how inconsistencies between an utterance and a previous commitment are accounted for, and so on.

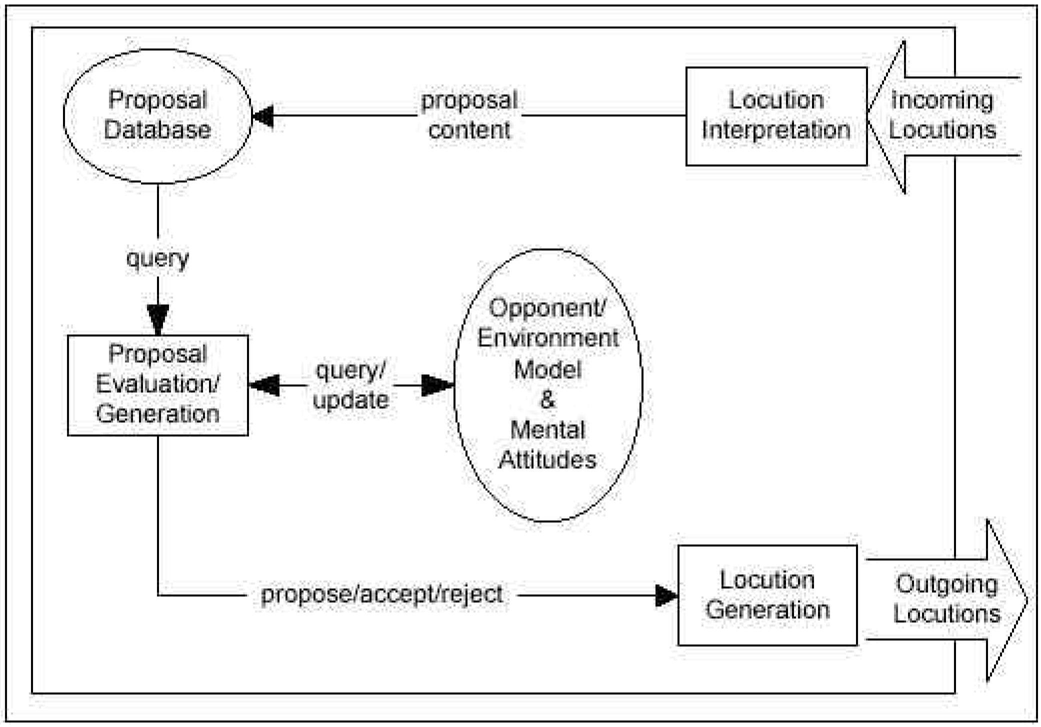



**Figure 3 Conceptual Elements of a Classical Negotiating Agent**

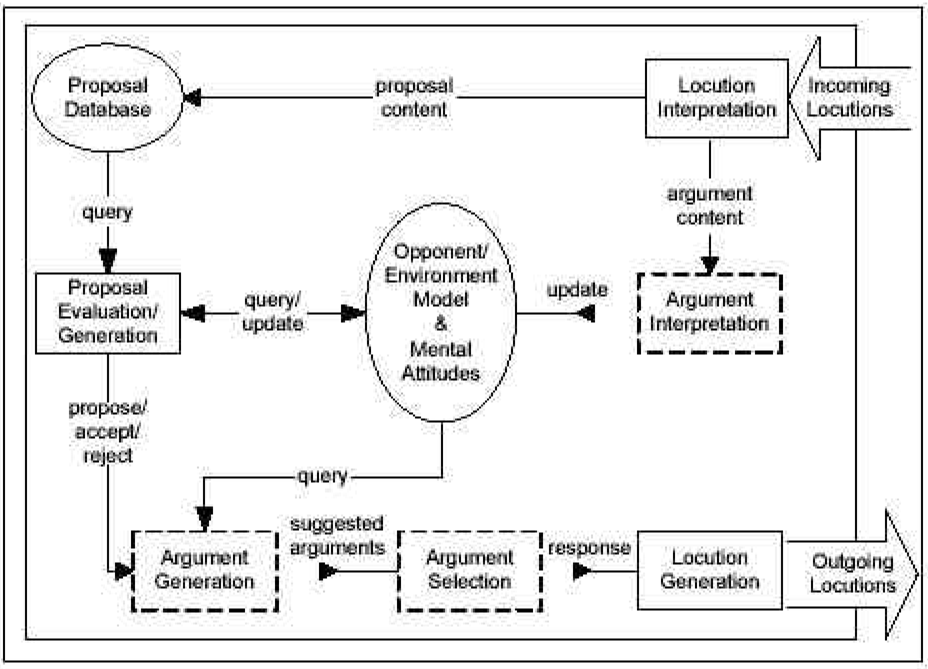



**Figure 4 Conceptual Elements of an ABN Agent**

Summarizing, heuristics and planning improve game theory analysis relaxing the unbounded computational resources restriction, one of its main disadvantages. Moreover, *the optimal for a participant, which is the game theory goal, could not be the optimal for everyone*. However heuristics and planning have problems with:

- Approximate notion of rationality
- The opportunity to examine the full space of possible outcomes
- Predict agents behaviour
- Manage information about the environment and the other agents
- Agents' utilities or preferences are usually assumed to be ***completely characterised PRIOR TO*** the interaction.

ABN provides a richer communication and meta-level information over this structure.

However, some questions remain open.

- Is it possible to define what is understood as *'rational'*?
- What is the full space of possible outcomes?
- Could the agent behaviour be *predictable*?
- Has '*Knowledge'* to be characterised PRIOR TO the interaction?

## 4.2. MAS (Multi-Agent Systems) and policies

Inside the architecture that it is going to be developed to interact to Semantic Web, agents play a key role (see section 9 for more information about ***agents***). They represent the dynamical part of the system. They have to perform tasks to achieve a goal. Usually, as in section 4.1 was explained some Multi-Agent Systems are non-cooperative. Here, this makes no sense, because there is a common goal: **an agreement for all participants formalized in a contract**. If there is no agreement, the goal is not achieved and there is no negotiation as it is understood in this work. In other words, in this context a negotiation takes sense when the agreement is achieved. This is the worst part to get. This is the main reason to research new methods in order to assure it accomplishes.

Thus, we are looking for an implementation of multiple programs (agents) Multi-Agent System (MAS) that cooperate to achieve a goal, that is, an agreement among the parts. They coordinate with each other to attempt to converge on the solution to one or more tasks. ***Agent negotiation is the convergence*** upon this solution through compromise and communication. Currently, the ***implementation of agents is highly dependent on the programming language***, and any perspective to the negotiation methods agents use to achieve goals and tasks are drawn after the implementation phase of the development. The MAS key features are:

- Internal Layer(s)
- Communication
- Coordination
- Legacy
- Integration

In the next sections, an overview of several MAS are shown to come clear to the reasons why one was chosen and why an Expert system should be integrated inside the architecture (see section 5).

### 4.2.1. FIPA

There are no many books related to specific MAS. The Foundation for Intelligent Physical Agents (FIPA) was formed in 1996 to produce software standards for heterogeneous and interacting agents and agent-based systems. It means, relatively, it is very new software and moreover it is an active area where changes happen every day. World Wide Web lists are the best tool to obtain the last news.

In the production of these standards, FIPA requires input and collaboration from its membership and from the agent's field in general to build specifications that can be used to achieve interoperability between agent-based systems developed by different companies and organisations.

The agent platform comes with a whole host of features:

It comes with a set of "Core agents"

- Directory Facilitator
- Agent Management System
- Database Agent
- Expert Knowledge Based Agents

And also Agent constructs

- Tasks, and a task manager that can be used and configured by an agent
- Messaging system allowing your agents to interact.

The Directory Facilitator (***DF***), Agent Management System (***AMS***) and Agent Communication Channel (***ACC***) are specific ***types of agents*** that support agent management. The AMS and ACC support inter-agent communication. The ACC supports interoperability within and across different platforms. The Internal Platform Message Transport (IPMT) provides a message routing service for agents on a particular platform which must be reliable, orderly and adhere to the requirements specified in FIPA Part 2. The ACC, AMS, IPMT and DF form will be termed as the Agent Platform (AP). These are mandatory, normative components of the model.

The system interfaces with the underlying network transparently (via HTTP[15], CORBA[16], sockets).  A Conversation manager is also provided to help keep track of conversations (which can get difficult if the agent is interacting with many other agents simultaneously).

FIPA works on enabling intelligent agents' interoperability via standardized agent communication and content languages. Beside the generic communication framework, FIPA is also specifying ontology and negotiation protocols to support interoperability in specific application areas (travel assistance, multimedia entertainment, network service provisioning, manufacturing etc.).

Intelligent agents refer to the classes of agents, most of which are static, that rely on high level, speech act agent communications (e.g. via KQML/KIF [17], FIPA-ACL/SL (Semantic language[18])).

The contents of the messages can be in any domain language. The locution inform(*a; b; '; lan*), for example, allows agent *a* to inform another agent *b* of statement *'* which is in language *lan*. Other locutions allow agents to express proposals for action, acceptance and rejection of proposals, make various queries about time and place, and so on. FIPA ACL has been given semantics in the form of pre- and post-conditions of each locution. This semantics is based on *speech act theory*, due to a philosopher of language, John Austin and his student John Searle in the 60's, in which a locution is seen as an action that affects the world in some way. While FIPA ACL offers the benefits of being a more or less standard agent communication language, it fails to capture all utterances needed in a negotiation interaction***. For example, FIPA ACL does not have locutions expressing the desire to enter or leave a negotiation interaction, to provide an explicit critique to a proposal or to request an argument for a claim***. While such locutions may be constructed by injecting particular domain language statements within locutions similar to those of FIPA ACL, the semantics of these statements fall outside the boundaries of the communication language[19]:

Within an agent communication paradigm, co-operation is realized via the Agent Communication Language (ACL), the content language and the ontology which identifies the set of basic concepts (taxonomy) used in the message content for co-operations. An ontology here is similar to an API in the RPC context. It identifies a specific co-operation interface of the intelligent agents. This part will be developed taking into account semantics in the next sections (see IPROnto in section 4.5.2).

Via message compositions and exchanging knowledge or concept definitions, we can achieve similar effects to those of mobile agents. For example, we can dynamically adapt the co-operation interfaces of the intelligent agents by modifying their knowledge about the ontology, we can delegate aggregations of actions (mobile agents) to reduce communication traffic and reduce dependency on the network connectivity, and also to distribute 'on-demand' the functions or intelligence.

Therefore, the intelligent agent approach also provides the means for dynamically adaptive, robust, flexible and effective co-operations and interoperability among distributed dynamic systems.

The major difference between mobile agents and intelligent agents and the corresponding OMG MASIF[20] and FIPA specifications is that a mobile agent usually uses a low level programming language, while the intelligent agent typically has a speech act alike communication language and a predicate logic based content language.

For instance, we consider ***Fipa-Request interaction protocol***, as we can see in the next Figure 5 and we ask ourselves about the suitable object structure of ***FipaRequestInitiatorBehavior*** protocol in order to implement it (see Figure 6).

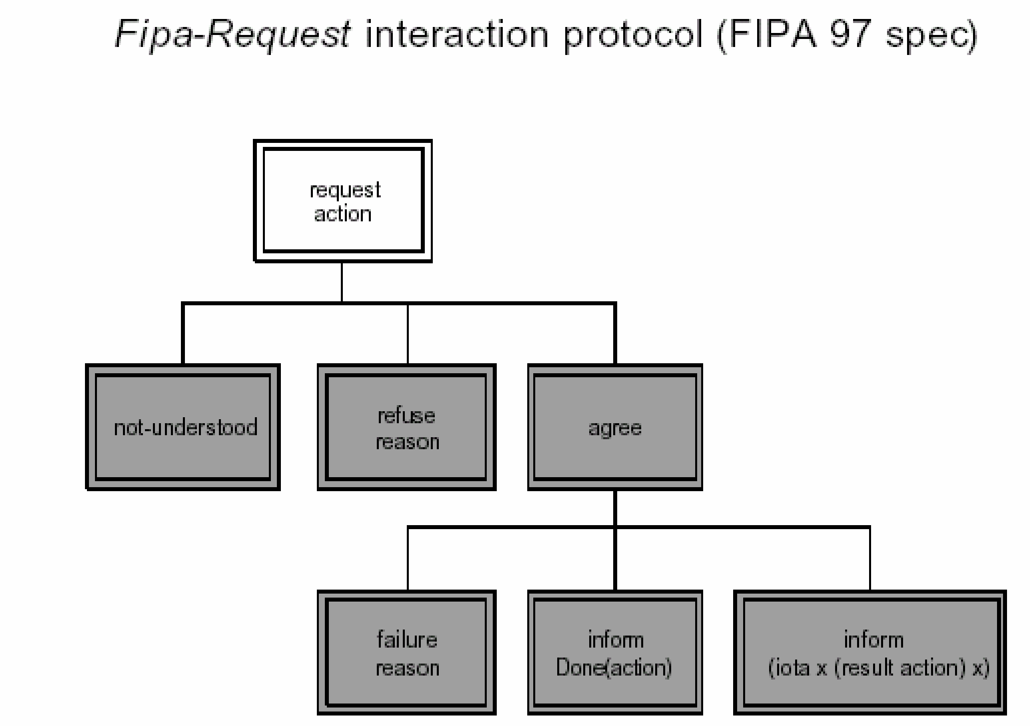



**Figure 5 Example of FIPA interaction protocol**

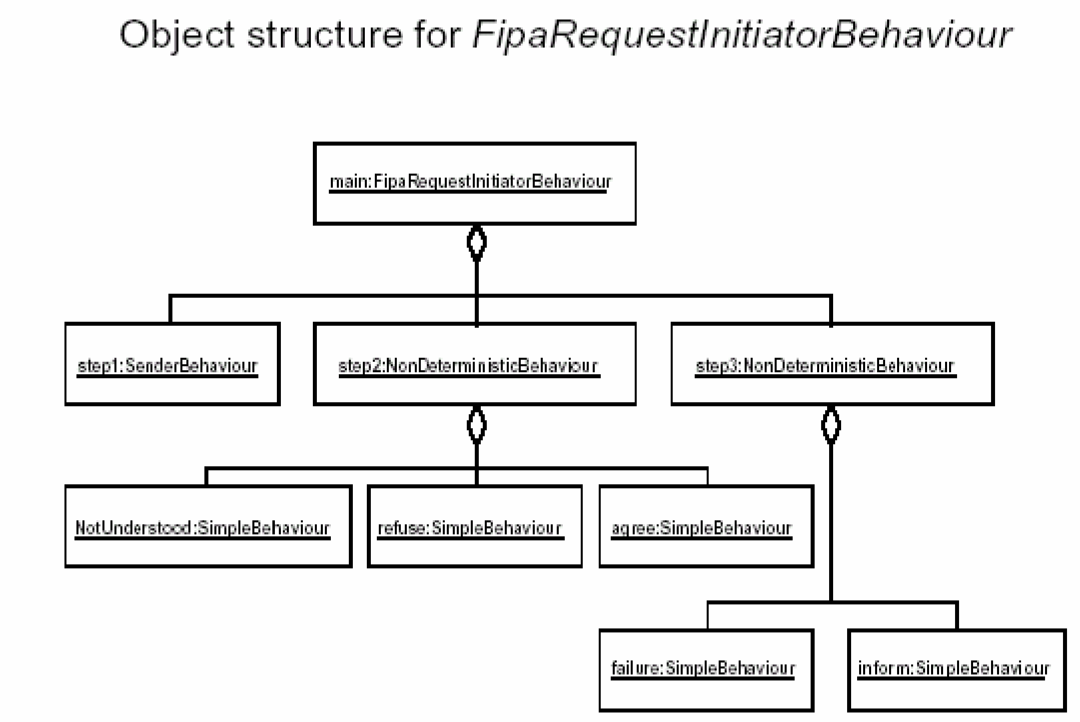



**Figure 6 Object structure of FipaRequestInitiatorBehavior protocol**

### 4.2.2. JADE-LEAP

JADE-LEAP [21] (Java Agent DEvelopment Framework) was designed as an abstract architecture over FIPA. There are other examples like ZEUS, (see Figure 12). Thus, JADE-LEAP is one Abstract Architecture over FIPA standards that covers:

- Agent Communication
- Interaction Protocols
- Agent Management
- Communicative Acts
- Agent Message Transport
- Content Languages.

Moreover, JADE-LEAP is a Java-based agent development framework. It is a combination of two products: A FIPA-Compliant Agent Platform and a package to develop Java agents. It includes the following agents (see Figure 7):

- AMS (Agent Management System)
- DF (Directory Facilitator)
- Sniffer: controls activity of the rest of agents in the platform
- RMA: represents de GUI (Graphical User Interface) where the other agents are shown.

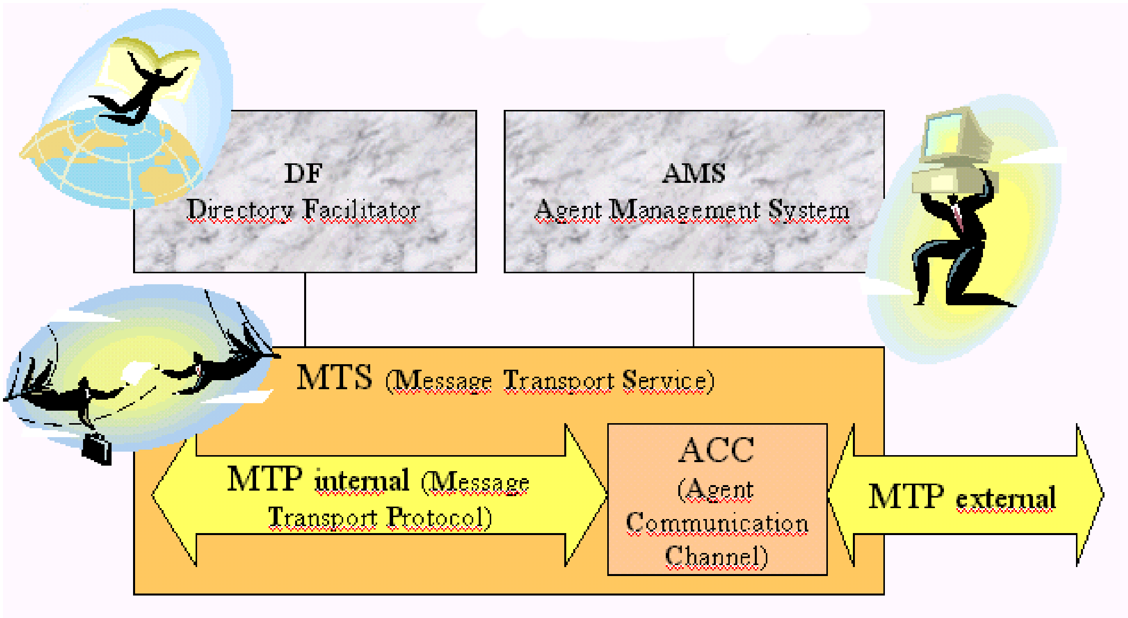



**Figure 7 JADE Front End Container**

It also includes features like:

- A library of FIPA interaction protocols
- Automatic registration of agents with AMS

- FIPA-compliant naming service
- FIPA-compliant IIOP to connect to different AP-s
- GUI to manage several agents and AP-s.

In the following lines we are going to explicit which are the FIPA standards (see Figure 8):

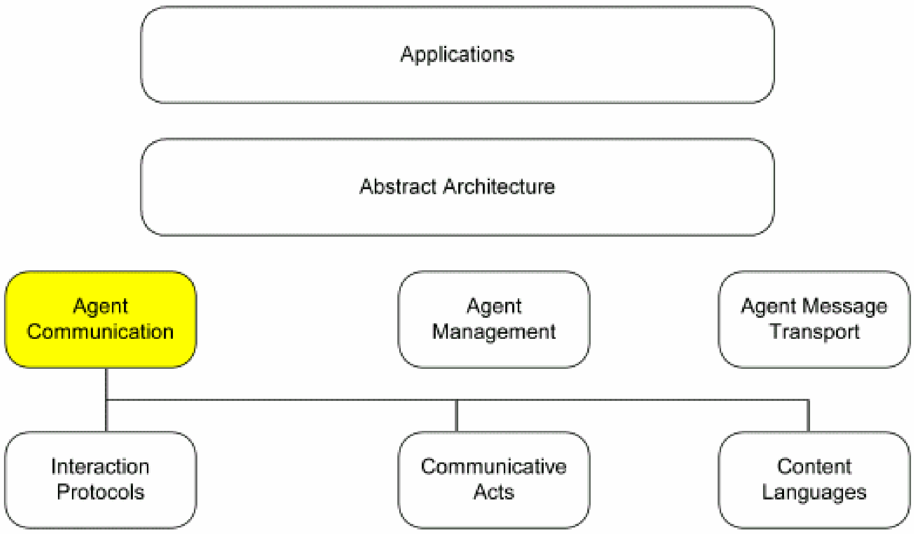



**Figure 8 JADE-LEAP as Abstract Architecture over FIPA standards**

Agent Communication: It deals with ACL (Agent Communication Language) Message Structure - specifies the elements of the FIPA-ACL message, the elements are

- Performative: ***FIPA message type***. Examples: INFORM, QUERY, PROPOSE… It answers the question: ***what is the purpose of the message?***
- Actor/role: sender, receiver, reply-to. It answers the question: ***who is manipulating*** the message or the ***role*** associate to the participant.
- Content: ACL Message Representation in: Bit-Efficient, String or XML/RDF. Agent Message Transport Envelope Representation in: Bit-Efficient or XML. In fact this part is where the ***meaning of the message*** is explained. Here, a compromise has to be made to choose what it is key in our system. We choose to send messages as ***semantically*** rich as we can so we use RDF.
- Language: FIPA Content Language (CL) specifications deal with different representations of the content of ACL messages. There are several Content Language specifications: ***FIPA-SL*** (Semantic Language), ***FIPA-CCL*** (Constraint Choice Language), ***FIPA-KIF*** (Knowledge Interchange Format) and ***FIPA-RDF*** (Resource Description Framework).
- It is also possible to explicit which ***ontology*** we are going to use. However, it has to be known by the agents prior to be analysed.

- Protocols based on ***FIPA Interaction Protocols Library specification***: Request, Query, Contract Net (see Figure 9) which we are going to develop, English Auction, Dutch Auction, Brokering, Recruiting, Propose.
- There is also a parameter to control the speech-act: conversation-id.
- ***FIPA Communicative Acts from the Communicative Act Library*** specification. Examples as Accept Proposal, Agree, Cancel, Call for Proposal, Confirm, Disconfirm, Failure, Inform, Inform If, Inform Ref, Not Understood, Propagate, Propose, Proxy, Query If, Query Ref, Refuse, Reject Proposal, Request, Request When, Request Whenever and Subscribe. They are the roots of the protocol, these communicative acts build them (see Figure 9).
- Moreover a control in the answered messages is done by the parameters: ***reply-with, in-reply-to*** and ***reply-by***.

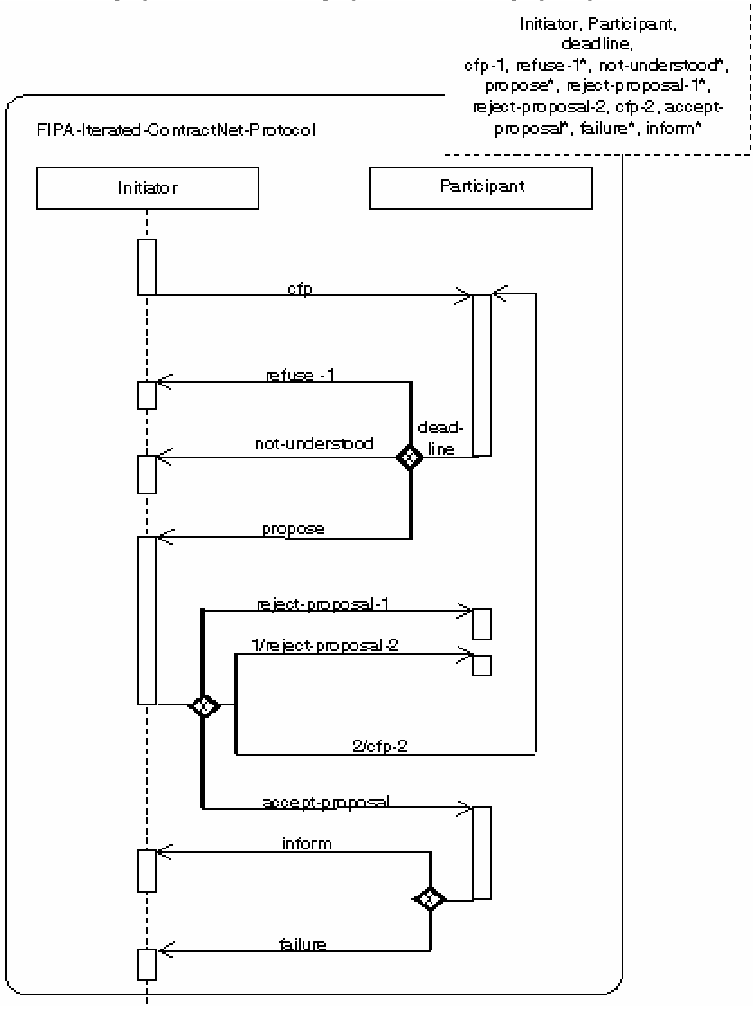



**Figure 9 FIPA-Contract-Net-Iterated protocol**

FIPA Agent Management specifications deal with the control and management of agents within and across agent platforms. Looking at figure Figure 10, it is possible to distinguish Directory Facilitator (***DF***), which provides a ***yellow pages directory*** service to agents. An Agent Platform (AP) can have more than one DF and the operations that DF must support are: ***register, deregister, modify and search***. There is also one Agent Management System (***AMS***) per

AP. The AMS maintains and controls agent life-cycle and authorization for agents to access the MTS. Moreover the operations that AMS must support are: register, deregister, modify, search and get-description. It can also instruct the AP to: suspend, terminate, create, execute agent, resume agent execution and resource management.

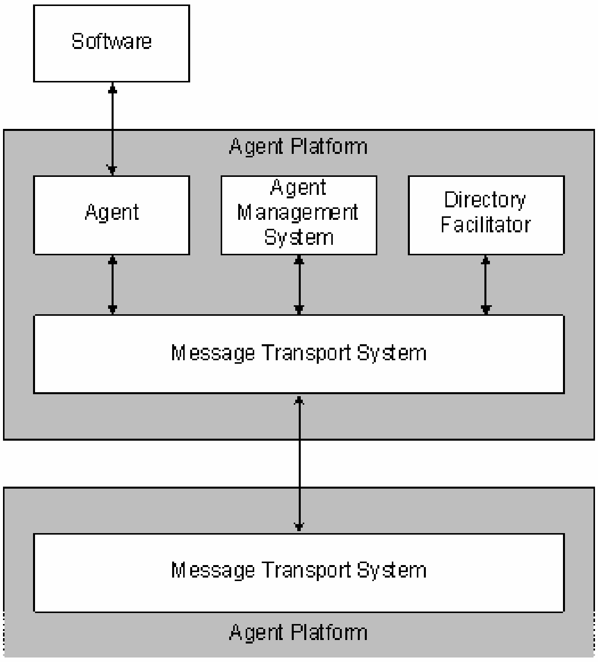



**Figure 10 FIPA Agent Message Transport system interaction**

FIPA Agent Message Transport (see Figure 10) specifications deal with the transport and representation of messages across different network transport protocols, including wired and wireless environments. Their specifications are:

- Agent Message Transport Service, where FIPA Agent Message Transport Protocol (MTP) specifications deal with different network transport protocols for delivering ACL messages: ***IIOP***[16]***,HTTP***[15] ***or WAP***[142].
- Messaging Interoperability Service.

Referring to JADE architecture, each agent ***lives inside a container*** which is a JVM and provides a complete runtime environment for agent execution. Moreover, it allows several agents to run concurrently and controls the ***life-cycle*** of agents (***create, suspend, resume, kill***). It deals also with communication (dispatches and routes messages). There is a light-weight container provided for agent execution within a web browser and a special Container - the front-end (FE) container. It runs the management agents and it represents the whole platform to the outside world. The GUI itself is implemented as an agent - Remote Management Agent (RMA).

The JADE-LEAP Agent Execution Model (see Figure 11) is based on behaviours, which are abstractions used to model the tasks that an agent is able to perform (for instance, multiple simultaneous conversations).

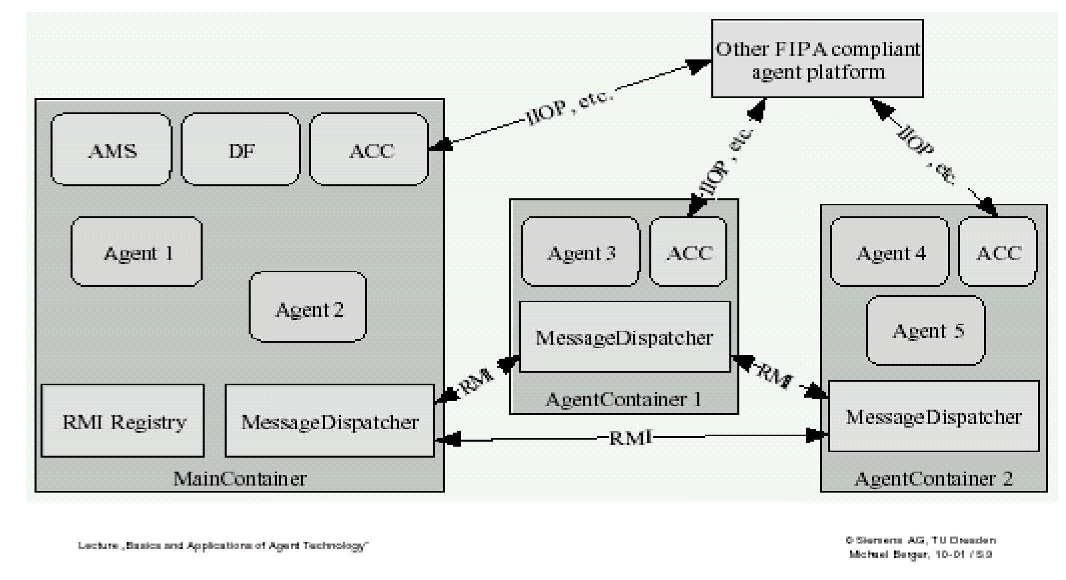



**Figure 11 JADE-LEAP Agent Execution Model**

Agents instantiate behaviours according to their needs and capabilities. JADE-LEAP uses the thread-per-agent concurrency model instead of a thread-per-behaviour model in order to keep the number of threads small.

Agents extend from the base Agent class which implements a scheduler. The scheduler carries out a round-robin non-pre-emptive policy among all behaviours available in the ready queue. The execution of a Behaviour derived class ends when the task itself relinquishes control.

Behaviours can be added/removed using Agent's methods ***addBehaviour*** and ***removeBehavior***. Developers can use existing behaviours provided by JADE or define their own classes, which need to extend Behaviour and implement the abstract methods *action()* and *done().*

Behaviour class can be defined as the *reification* of an agent task or plan. Reflection and reification are closely related concepts, since a reflective system requires the reification of some of its internals. That is, reflection is the capability of a system to ***reason about and act upon itself***. A reflective system is composed of a base-level, which is the part of the system reasoned about, and a meta-level, which has access to the reified information about the base-level.

Thus, representing tasks as first class objects allows extra flexibility. It must be noticed that higher-level agent architectures need to explicitly represent ***agent actions to reason about them***.

In JADE Programming Model, applications are made by one or more agents. A JADE agent is a user defined subclass of the Agent class in ***jade.core*** package and Agent tasks are mapped onto user defined subclasses of Behaviour class in ***jade.core.behaviours*** package. Respect to FIPA ACL messages and standard ontologies, they are mapped onto ACLMessage and ***AgentManagementOntology*** Java classes respectively. FIPA standard actions and interaction protocols are mapped onto suitable methods of the Agent class or to already made Behaviour subclasses.

There is another important issue as the concurrency control, where multi-threaded inter-agent scheduling is performed by threads (one agent is a thread), and Java VM does Thread-Switching. Also it is possible a single-threaded agent having and executing multiple tasks at the same time; these tasks are described in Behaviours, and each agent has non pre-emptive ***Round-Robin-Scheduler*** for Behaviours cooperating.

Referring to event dispatching and method calls, we could find two kind of platforms: intra and inter. The first uses Java RMI (Remote Method Invocation protocol) while the second uses OMG[20] IIOP. However, new protocols being supported as HTTP if we use the right plug-in. Also it is possible to distinguish two levels at the concurrency model: intra and inter, in the first case we find co-operative and behaviour classes and in the second case pre-emptive Java threads.

The relationship between Agent and Behaviour is like the Object and Method one in OOP (Object Oriented Programming).

Basic Behaviour-Schema can be described in terms of super-classes: SimpleBehaviour, OneShotBehaviour, CyclicBehaviour, SenderBehaviour, ReceiverBehaviour, ComplexBehaviour, SequentialBehaviour, NonDeterministicBehaviour and a user-defined Behaviour by sub-classing the previous ones.

Just in doors of the XXI century, it was clear that MAS have to arrive to light devices; each day there is a rising demand on services over light devices, this was the main reason for LEAP project (actually is completely integrated with JADE).

This project began in January 2000 with clear objectives as:

- Develop a reference Lightweight Extensible FIPA-compliant Agent Platform
- Provide an Integrated Agent Development Environment
- Develop Agent Services supporting a mobile enterprise workforce
- Evaluate platform and applications in Field Trials

In the LEAP project context can be found the next contractors: Motorola as prime contractor, ADAC, Broadcom, BT and Telecom Lab. It means that ZEUS and JADE platforms are working together as we will see: It seems that some Telecom initiatives in MAS systems seem to converge.

LEAP has already successfully (see Figure 13) been run on:

- J2SE devices (desktops, laptops);

- Palm OS (both Palm emulator POSE and real Palm devices IIIc and Vx);
- EPOC devices (Psion 5mx [22], Quartz emulator);
- Sun's and Zucotto's MIDP emulators;
- iPAQ (with wireless LAN or GSM connections).

Potentially, LEAP can run on any device that supports one of the following standardised versions of Java: J2SE, J2ME or pJava.

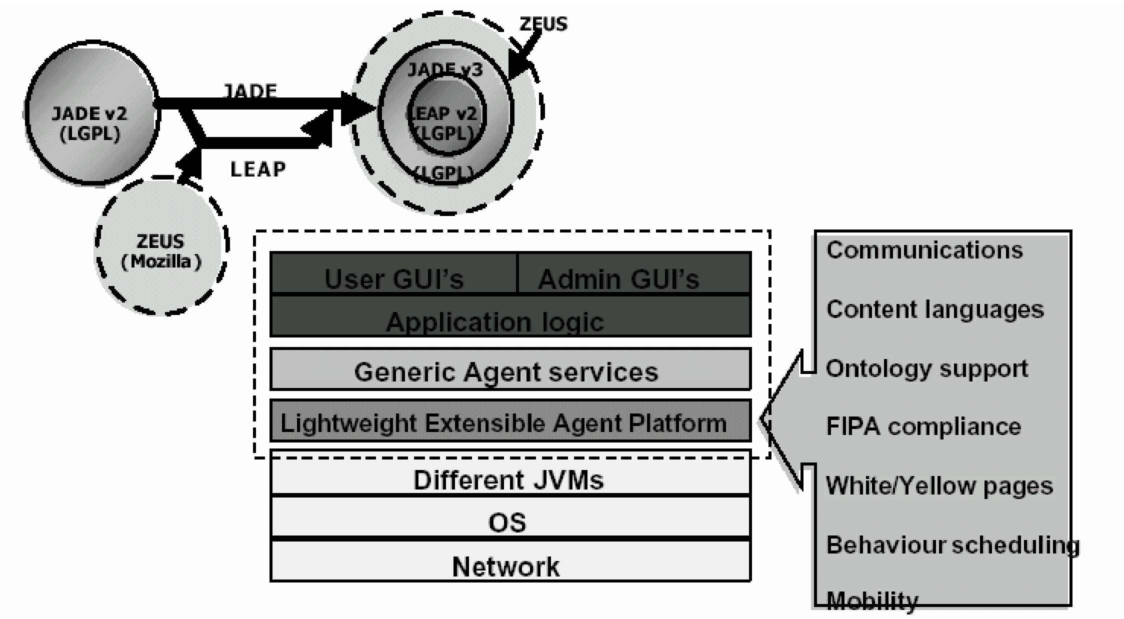



**Figure 12 LEAP development process**

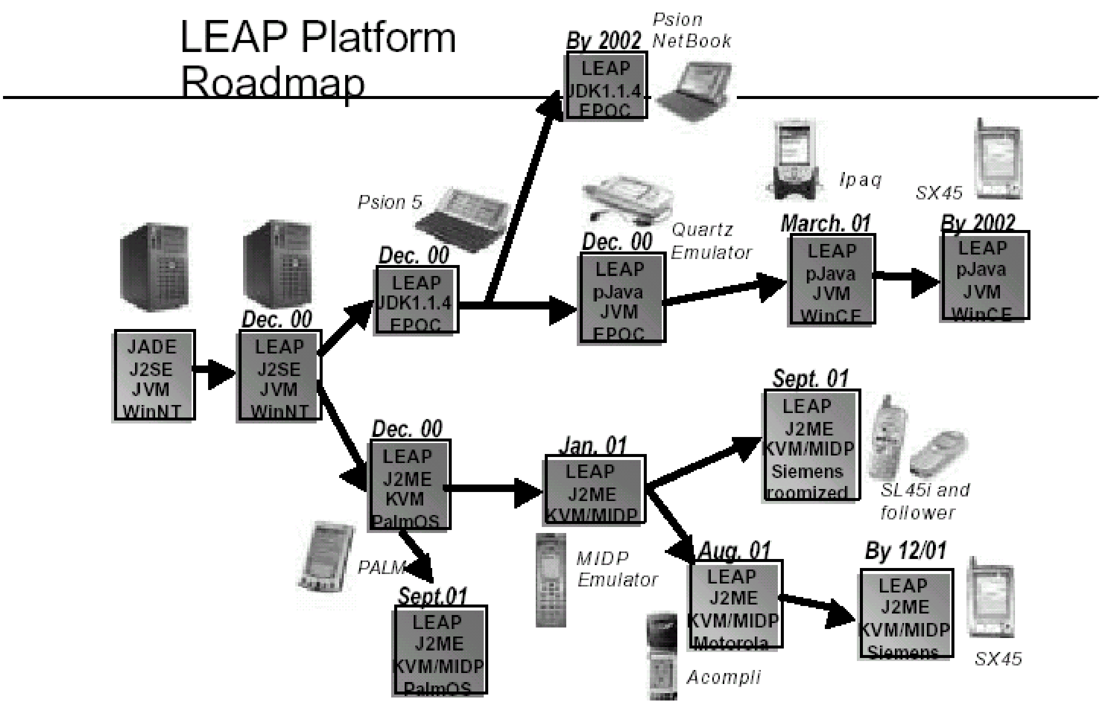



**Figure 13 LEAP platform in different wireless devices**

One of the advantages of LEAP structure is his modular (see Figure 12) origin is configurable to meet device and application, for instance, it has a Kernel module which is portable across systems and it is the common core of LEAP. Moreover has device dependent modules adapted to devices requirements as communications or encodings. Tools modules are optional as GUIs, sniffer, persistence or ontologies.

LEAP implements backward JADE Compatibility and J2ME CLDC Compliance with a Modified boot sequence and platform configuration that does not affect the developer, no change to the external API has been done and it means existing JADE services still run. But there is lack of ontology support on J2ME, in other words, there is minimal content language support and it generates a problem, some incompatibilities between J2SE and J2ME CLDC API classes appear. It is possible to find a solution, modifications of JADE code, trying to avoid using unsupported classes and methods, re-implementations of classes for J2ME and putting certain functionality out into device-dependent modules.

Programmers know that theory is not the same thing as design, it is the reason why it becomes necessary: *Mapping Theory into Design*

| | |
|---|---|
| Agents are autonomous | Agents are active objects |
| Agents are social entities, and can engage in many conversations | Intra-agent concurrency is Needed |
| Messages are speech acts, not invocations | Asynchronous messaging must be used |
| An agent can refuse to perform an action, he can say “no” and “I don’t care” | Peer-To-Peer messaging (built over distributed objects Client/Server interactions) |

Summarizing, JADE-LEAP provides a platform designed for a wide scope of devices plus knowledge management without forgetting security issues.

### 4.2.3. Grasshopper

Grasshopper[23] is an open Java-based mobile intelligent agent platform, which is in compliance with both available international agent standards, namely the OMG MASIF (Mobile Agent System Interoperability Facility) and FIPA (Foundation for Intelligent Physical Agents) specifications.

It is a mobile agent development and runtime platform that is built on top of a distributed processing environment. It is based on Java JDK and written entirely in Java. This way, an integration of the traditional client/server paradigm and mobile agent technology is achieved. Most importantly, Grasshopper is developed compliant to the OMG MASIF specifications. Two types of agents are distinguished in Grasshopper: mobile agents and stationary agents. The actual runtime environment for both mobile and stationary agents is an agency: on each host at least one agency has to run to support the execution of agents.

The grasshopper platform also handles persistence, security, communication and registration.

OMG MASIF aims at enabling mobile agents to migrate between Agent Systems of the same profile (language, agent system type, authentication type and serialization methods) via standardized CORBA IDL interfaces. The mobile agent's religion focuses on mobility of program codes together with their states among network sites. Via higher granularity of mobile agents (i.e. higher number of operations encapsulated within a mobile agent) and on-demand migrations, the mobile agents approach can help to dynamically adapt interfaces and services of remote systems, reduce dependency on the constant availability of underlying network connectivity, achieve dynamic load balance and enable dynamic distribution of functions. With these features, mobile agents provide a robust, flexible and effective design paradigm for the distributed dynamic environments.

Typically, a mobile agent moves between two static software systems, the agent systems and their associated user or provider applications. In this sense, a mobile can be regarded as the means for adaptive, flexible co-operations or interoperability among software systems.

This difference results between intelligent agents (FIPA) and mobile agents (OMG) are. These differences have changed a lot and they continue changing. For instance, intelligent agents also have inter-platform mobility and persistence nowadays, but they were not designed for that preliminary. These differences are mentioned because they explain the two different origins: two conceptions of understanding world (intelligence *vs.* mobility).

**Table 1. Initial differences between FIPA and OMG agents.**

| | |
|---|---|
| Efficiency | Intelligent agent messages usually take less time and transport capacity to migrate between source and destination sites. However, it is generally more efficient to execute a mobile agent due to its lower level implementation. |
| Adaptability | The knowledge contained in the messages can be easily integrated into the knowledge of the receiving agent, making intelligent agent technology more appropriate for adapting intelligent agent interfaces and functionality. This is not so easy with mobile agents. However, mobile agents can sometimes be used to modify or replace the remote applications or their components (autonomous software downloading and configuration) if the mobile agents and the remote applications are implemented in the same language. Intelligent agents usually do not directly support this kind of modification or replacement. |
| Syntactical Interoperability | Mobile agents require homogeneous platforms for interoperability, while the intelligent agent paradigm supports the interoperability among heterogeneous environments. |
| Richness of Interaction Protocols | ACL can provide a richer set of semantically standardised interactions between static software systems than the mobile agent paradigm, where move and receive agent is the operation being standardised. |
| Semantic Interoperability | The intelligent agent approach supports not only syntax-based interoperability but also interoperability based on semantics. This feature will be very useful for complex and dynamic co-operation problems. |
| Binding AI alike Technologies | Agent communication, with its strong association to Artificial Intelligence (AI), can easily support the bindings of AI-like technologies into the individual static agents. This feature can further increase the flexibility, tolerance, robustness of the co-operation and negotiation among agents. |
| Security | It is easier to analyse the behaviour of an intelligent agent message. Therefore, the receiving intelligent agent can check the messages for subtle security and contract violations. Intelligent agents are therefore safer than mobile agents. |
| Reliability | Agent communication paradigm and its languages can be more easily associated to a formal theory for agent interactions. Such theory enables the formal analysis and verification of the global distributed systems and can further |

increase the reliability of agent-based applications.

The following mappings of MASIF and FIPA concepts may be identified:

- FIPA domain corresponds to MASIF Region
- FIPA Agent Platform corresponds to MASIF Agent System
- FIPA Directory Facilitator corresponds to MASIF MAF Finder
- FIPA Agent Management System corresponds to MASIF MAF Agent System
- FIPA ACL/CL corresponds to both MASIF mobile agent inter-agent communication and mobile agent migration
- FIPA Agent Resource Broker corresponds to MASIF MAF Finder
- FIPA Agent Wrapper corresponds to mobile agent Resource Interfaces

It has to be noted that at present there is no one-to-one mapping possible, and alignments of interfaces and components are necessary. However, the above mappings are used as stimuli for the following considerations. Subsequently, in order to analyze the applicability, advantages and disadvantages of different approaches for harmonizing OMG and FIPA mobile/intelligent agents paradigms and their impacts on the development of agent technologies, some examples are given in the Cordis web[24].

There are many M.A.S that can be compared to FIPA, however in telecom domain, this two worlds show the main tendencies.

## 4.2.4. Peer to peer (P2P)

P2P technology appeared as a new revolution. Its open nature as a distributed system offered new opportunities to provide multimedia content without centralised control. Users around the world share their computers to get new songs, films, trailers and so on.

This claimed our attention in order to establish the economical potential. Thus, an open source software was chosen to test it: Project JXTA[25]. It was started at Sun Microsystems[26] in 2001. JXTA technology is a set of simple, open P2P protocols that enable any device on the network to communicate, collaborate, and share resources.

JXTA peers create a virtual, ad hoc network on top of existing networks, hiding their underlying complexity (see Figure 14). In the JXTA virtual network, any peer can interact with other peers, regardless of location, type of device, or operating environment — even when some peers and resources are located behind firewalls or are on different network transports. Thus, access to the resources of the network is not limited by platform incompatibilities or the constraints of client-server architecture.

JXTA technology espouses the core technology objectives of ubiquity, platform independence, interoperability, and security. JXTA technology runs on any device, including cell phones, PDAs, two-way pagers, electronic sensors, desktop computers, and servers. Based on proven technologies and standards such as HTTP, TCP/IP and XML, JXTA technology is not dependent on any particular programming language, networking platform, or system platform and can work with any combination of these.

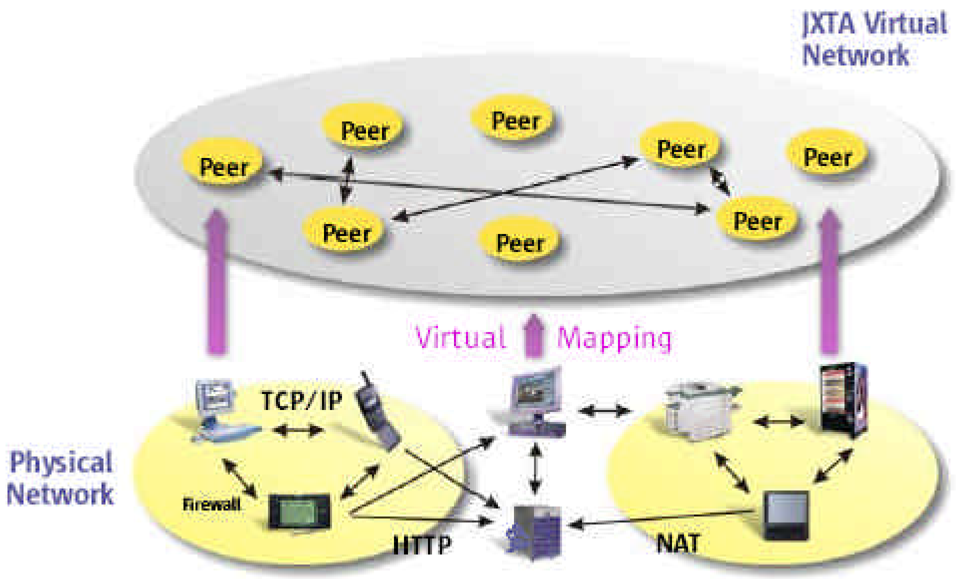



**Figure 14 JXTA Virtual Network**

JXTA defines a set of protocols that can be implemented by peers to communicate and collaborate with other peers implementing the JXTA protocols. It tries to standardize messaging systems, specifically peer-to-peer systems, by defining protocols, rather than implementations. Currently Java and C implementations of the JXTA protocols are available. There are 6 protocols in the JXTA specification:

- Peer Resolver Protocol (PRP)
- Peer Discovery Protocol (PDP)
- Peer Information Protocol (PIP)
- Pipe Binding Protocol (PBP)
- Endpoint Routing Protocol (ERP)
- Rendezvous Protocol (RVP)

Each of the JXTA protocols is independent of the others, and a peer is not required to implement all six protocols.

The idea of the JAL (JXTA Abstraction Layer) is to abstract out the functionality in a P2P environment, which makes P2P technology pluggable. Given the evolving nature of the JXTA API, and the steep learning curve, it makes sense to design a foundation/helper classes for all the important functionality. This API (Application Programmable Interface) abstraction tries to provide an immutable API for all the commonly used P2P primitives. JAL enables rapid development of P2P applications, and minimizes application complexity by abstracting out the P2P technology. Incorporating JXTA into the code can create a tie-in with JXTA technology. JAL Architecture is described in Figure 15.

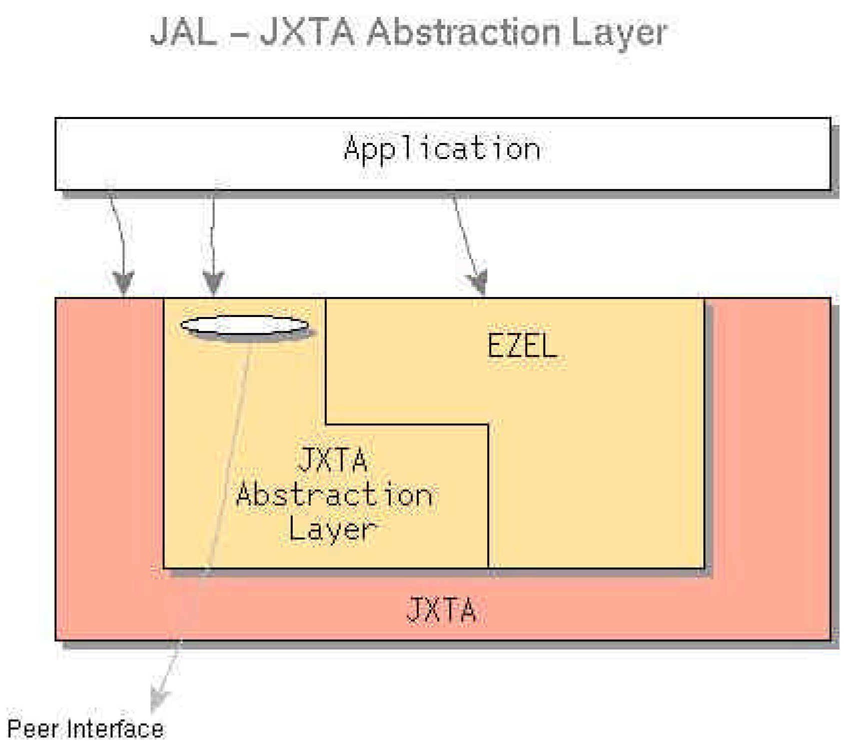



**Figure 15. Application interacting to JAL**

### 4.2.5. Expert Systems

When agents have to make decisions, a policy tells them how. Policy is designed as a set of rules and usually they are associated to a logic. This is the reason why expert systems are used in conjunction to M.A.S.

An expert system tool, or shell, is a software development environment containing the basic components of expert systems. Associated with a shell is a prescribed method for building applications by configuring and instantiating these components. The core components of expert systems are the knowledge base and the reasoning engine.

- **Knowledge base**: A store of factual and heuristic knowledge. An ES tool provides one or more knowledge representation schemes for expressing knowledge about the application domain. Some tools use both frames (objects) and IF-THEN rules. For instance, in PROLOG[27] the knowledge is represented as logical statements.

- **Reasoning engine**: Inference mechanisms for manipulating the symbolic information and knowledge in the knowledge base to form a line of reasoning in solving a problem. The inference mechanism can range from simple *'modus ponens'* backward chaining of IF-THEN rules to case-based reasoning.

- **Knowledge acquisition subsystem:** A subsystem to help experts build knowledge bases. Collecting knowledge needed to solve problems and build the knowledge base continues to be the biggest bottleneck in building expert systems.

- **Explanation subsystem**: A subsystem that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at to justifying the need for additional data.

- **User interface**: The means of communication with the user. The user interface is generally not a part of the ES technology, and was not given much attention in the past. However, it is now widely accepted that the user interface can make a critical difference in the perceived utility of a system regardless of the system's performance.

There are several options as JESS[28] (Java Expert System Shell) or PROLOG, the main difference is CLIPS is forward chaining while PROLOG is backwards chaining. CLIPS was chosen in the customer side in the Semantic Web architecture because it is optimised for speed at the cost of space and in applications which are near the user this condition is very restrictive, but PROLOG is optimised for space at the cost of speed; this characteristic makes it perfect for applications near the database. However software improvement is something variable and it can not be considered as an eternal statement.

JESS is a rule engine and scripting environment written entirely in Sun's Java language by Ernest Friedman-Hill at Sandia National Laboratories in Livermore, CA. JESS was originally inspired by the CLIPS expert system shell, but has grown into a complete, distinct Java-influenced environment of its own. Using JESS, it is possible to build Java applets and applications that have the capacity to "reason" using knowledge you supply in the form of declarative rules. JESS is surprisingly fast, and for some problems is faster than CLIPS itself (using a good JIT compiler, of course.)

JESS uses the Rete algorithm to process rules. Rete makes Jess much faster than a simple set of cascading if-then statements in a loop. Several decent expert systems are already written and widely available. Ernest Friedman-Hill wrote Jess entirely in Java at Sandia National Laboratories. It's fast, compact, and easily integrated into other systems. Jess is a clone of the CLIPS expert system shell (a shell is the expert system minus any rules). Jess provides their own notation for defining rules. The syntax of the Jess supports atoms, variables and list programming as in LISP, making the language easy to follow.

Two important constructs make up a rule-based expert system's knowledge base: facts and rules. A fact is a construct that defines a piece of information that is known to be true, rule is nothing more than an if/then statement that defines the set of facts that must be true (the if part) before a set of actions (the then part) can be executed. Rule-based expert systems are extremely powerful because actions themselves can assert new facts. When this happens additional rules apply and their actions are executed.

A rule-based system maintains a collection of knowledge nuggets called facts. This collection is known as the knowledge base. It is somewhat akin to a relational database, especially in that the facts must have a specific structure. In Jess, there are three kinds of facts: *ordered* facts, *unordered* facts, and *definstance* facts.

The knowledge base for any expert system relies heavily on the knowledge acquisition system. Jess does not provide any advanced level of knowledge acquisition methods. The knowledge can either be specified on the console using *deftemplate, deffacts* etc. But this is a tedious process. Also Jess does not automatically save the knowledge into any database. Hence the other option that Jess has for the gathering knowledge is use of the .CLP files. This is similar to programming languages in that the programs are stored in files and given to the compilers as input. The .CLP files can contain the facts and rules that the user wants to give to Jess as the knowledge for the system.

A *rule-based* expert system is an expert system that uses a set of rules as its knowledge base. Rules are simple (and not-so-simple) statements of the form: *If some condition is true, then do something.* A rule-based expert system continually tests a set of conditional statements (known as rules) against a database of data (known as facts) to see if any apply. If a rule applies, the expert system executes the associated actions.

This paragraph outlines some declarative features supported by JESS.

- **Negation** The Not Pattern allows the negation to in the rules.
- **Recursion**. The JESS language with its LISP like implementation supports recursion.
- **Declarative Database Updates**. The ability to call Beans within functions gives the ability to write new facts and rules as part of rule language using the I/O facilities of Java.
- **Schema Browsing**. Although essentially a debugging tool, JESS has a view command which allows the programmer to view the Rete Implementation of the rules in the database.
- **Integrity Constraints**. JESS does not have any built-in constraints since it is not a database. But rules can be written to check for the constraints.
- **Formal Semantics**. JESS uses a language derived from CLIPS and that have LISP like syntax. So it is very well defined.

The following lines outline the object-oriented features provided by JESS.

- **Object-Relational**. Jess allows the Bean Objects to be added as facts.
- **Modularity**. Jess does not support modules. But it does support functions. Also user defined Java classes can be used to define the structure of objects using the *deftemplate* construct to give rise to unordered facts.
- **Encapsulation**. The encapsulation in Jess is directly inherited from the encapsulation provided by Java.
- **Method Implementation**. Jess gives the ability to extend the Jess language by adding new commands to Jess. Also invocation of Java methods is possible in Jess and vice versa.

CLIPS is a multi-paradigm programming language that provides support for rule-based, object-oriented and procedural programming.

As a knowledge base is constructed and CLIPS reads characters from the keyboard or files, it groups them together into tokens. Tokens represent groups of characters that have special meaning to CLIPS.

CLIPS primitive data types:

- numeric fields (numbers):float, integer
- symbol
- string
- external address
- instance name
- instance address

Thus, the basic components of the rule-based Expert System are:

- Fact list: contains the data on which inferences are derived.

- Knowledge base: contains all the rules.
- Inference engine: controls overall execution.

### 4.2.6. Swarm systems

This MAS was chosen to test semantic and agent's behaviour boundaries. These systems provide simple rules however the agents show complex behaviour. The open question was: Should we know perfectly how the agent must interact to get agents that create a virtual market?

The expression "swarm intelligence" was first used by Beni, Hackwood, and Wang [29] in the context of cellular robotic systems where many simple agents occupy one or two dimensional environments to generate patterns and self-organize through nearest-neighbour interactions. Moreover the expression "swarm intelligence" is used to describe any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies.

Butrimenko [30] applied these ideas to the control of telecoms networks. The same type of approach has been used to design artificial neural networks that solve problems, or in the development of genetic algorithms for optimisation: if the brain and evolution, respectively, served as starting metaphors, most examples of neural networks and genetic algorithms in the context of engineering are strongly decoupled from their underlying metaphors. In these examples, some basic principles of brain function or of evolutionary processes are still present and are the most important, but, again, ultimately a good problem-solving device does not have to be biologically relevant.

The pattern detection and classification processes[31] executed by the proposed system emerge from the coordinated activities of agents of two populations in a shared computational environment. The result system-level behaviour is adaptive, robust, and scalable.

ACO [32](ant-based algorithms or Ant Colony Optimisation algorithms) have been applied to solve TSP (travel salesman problem) and other combinatorial optimisation problems such as the quadratic assignment problem, graph colouring, job-shop scheduling, sequential ordering, and vehicle routing[33].

As an emergent result of the actions of many ants, the shortest path between two or more locations is discovered and maintained. TSP was chosen for several reasons

- It is a shortest path problem to which the ant colony metaphor is easily adapted
- It is a very difficult problem (NP-hard)
- It has been studied a lot (it is considered to be "the benchmark problem" in combinatorial optimisation
- It is a didactic problem: it is very easy to understand and explanations of the algorithm behaviour are not obscured by too many technicalities.

- To find a closed tour of minimal length connecting n given cities. Each city must be visited once and only once.

The idea is to use a positive feedback mechanism, based on an analogy with ***the trail-laying trail-following behaviour of some species of ants and some other social insects***, to reinforce those portions of good solutions that contribute to the quality of these solutions, or to directly reinforce good solutions. A virtual pheromone, used as reinforcement, allows good solutions to be kept in memory, from where they can be used to make up better solutions. Of course, one needs to avoid some good, but not very good, solutions becoming reinforced to the point where they constrain the search too much, leading to a premature convergence (*stagnation*) of the algorithm. To avoid that, a form of negative feedback is implemented through pheromone evaporation, which includes a time scale into the algorithm. This time scale must not be too large, otherwise sub-optimal premature convergence behaviour can occur. But it must not be too short either. Otherwise no cooperative behaviour can emerge. Cooperative behaviour is the other important concept here: ant colony algorithms make use of the simultaneous exploration of different solutions by a collection of identical ants. Ants that perform well at a given iteration influence the exploration of ants in future iterations. Because ants explore different solutions, the resulting pheromone trail is the consequence of different perspectives on the space of solutions. There is a cooperative effect across time because ants in the next iteration use the pheromone trail to guide their exploration.

Ant System (AS) was the first ***ant colony optimisation algorithm***, had a performance similar to that of general purpose heuristics, such as simulated annealing or genetic algorithms, on small problems, but did not scale up well.

Let $d_{ij}$ be the distance between cities i and j. The problem can either be defined in Euclidean space, in which case

$$d_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2\right]^{1/2}$$

where $x_i$ and $y_i$ are the coordinates of city I, or can be more generally defined on a graph (N,E) where the cities are the nodes N and the connections between the cities are the edges of the graph E. The graph need not be fully connected. Note that the “distance” matrix need not be symmetric: if it is asymmetric the length of an edge connecting two cities i and j depends on whether one goes from i to j or from j to i. But whether the problem is symmetric or asymmetric does not change how it is being solved by Ant System (AS)

During an iteration of the AS algorithm each ant k, k=1, …, m builds a tour executing $n = |N|$ steps in which a probabilistic transition rule is applied. Iterations are indexed by t, $1 \le t \le t_{\max}$, where $t_{max}$ is the user defined maximum number of iterations allowed.

For each ant, **the transition** from city i to city j at iteration t of the algorithm depends on:

- Whether or not **the city has already been visited**. For each ant, a memory (also called tabu list) is maintained: it grows within a tour, and is then emptied between tours. The memory is used to define, for each ant k, the set $J_i^k$ of cities that the ant still has to visit when it is on city i (at the beginning $J_i^k$ contains all the cities but i). By exploiting $J_i^k$ an ant k can avoid visiting a city more than once.
- The inverse of the distance $h_{\mathrm{ij}} = 1/\mathrm{d}_{\mathrm{ij}}$, called *visibility*. **Visibility** is based on strictly **local information and represents the heuristic desirability** of choosing city j when in city i. Visibility can be used to direct ants' search, although a constructive method based on its sole use would produce very low quality solutions. The heuristic information is static, that it, it is not changed during problem solution.
- The **amount of virtual pheromone trail** $t_{ij}(t)$ on the edge connecting city i to city j. Pheromone trail is updated on-line and is intended to represent the **learned desirability** of choosing city j when in city i. As opposed to distance, a pheromone trail is a **more global type of information**. The pheromone trail information is changed during problem solution to **reflect the experience acquired by ants during problem solving**.

The **transition rule**, that is, the probability for ant k to go from city i to city j while building its t[th] tour is called **random proportional transition rule** and is given by:

$$p_{ij}^k(t) = \frac{[t_{ij}(t)]^a \cdot [h_{ij}]^b}{\sum_{l \in J_i^k} [t_{ij}(t)]^a \cdot [h_{ij}]^b}$$

if $j \in J_i^k$, and 0 if $j \notin J_i^k$, where $a$ and $b$ are two adjustable parameters that control the **relative weight of trail intensity,** $t_{ij}(t)$, and visibility $h_{\mathrm{ij}} = 1/\mathrm{d}_{\mathrm{ij}}$. If $a$ =0, the closest cities are more likely to be selected: this corresponds to a classical stochastic greedy algorithm (with multiple starting points since ants are initially randomly distributed on the nodes). If, on the contrary, $b$ =0, only pheromone amplification is at work: this method will lead to the rapid selection of tours that may not be optimal. A tradeoff between tour length and trail intensity therefore appears to be necessary. It is important to note that, although the form of the previous equation remains constant during an iteration, the value of the probability $p_{ij}^k(t)$ can be different for two ant on the same city i, since $p_{ij}^k(t)$ is a function of $J_i^k$, that is ,of the partial solution built by ant k.

After the completion of a tour, each ant k lays a quantity of pheromone $\Delta t_{ij}^k(t)$ on each edge (i,j) that it has used; the value $\Delta t_{ij}^k(t)$ depends on how well the ant has performed. At iteration t (the iteration counter is incremented by 1 when all ants have completed a tour), ant k lays $\Delta t_{ij}^k(t)$ on edge (i,j):

$$\Delta t_{ij}^{k}(t) = \begin{cases} Q / L^{k}(t) & \text{if (i, j)} \in \mathrm{T^{k}}(t) \\ 0 & \text{if (i, j)} \notin \mathrm{T^{k}}(t) \end{cases}$$

where $\mathrm{T^{k}}(t)$ is the tour done by ant k at iteration t, $L^{k}(t)$ is its length, and Q is a parameter (although the value of Q only weakly influences the final result, it should be set so that it has a value of the same order of magnitude as that of the optimal tour length, for example, found running a simple constructive heuristic like the nearest the nearest neighbor heuristic.

This method could not perform well without pheromone decay: in effect, it would lead to the amplification of the initial random fluctuations, which very probably would not be optimal. In order to ensure efficient solution space exploration, trail intensity must be allowed to decay, otherwise all ants will end up doing the same tour (stagnation): because of the addition of trail intensity, the probabilities of transitions between cities would be dominated by the pheromone term. Trail decay is implemented by introducing a coefficient decay $r, 0 \leq r \leq 1$. The resulting pheromone update rule, which is applied to all edges, is then:

$$t_{ij}(t) \leftarrow (1 - r) \cdot t_{ij}(t) + \Delta t_{ij}(t)$$

where $\Delta t_{ij}(t) = \sum_{k=1}^{m} \Delta t_{ij}^{k}(t)$, and m is the number of ants. The initial amount of pheromone on edges is assumed to be a small positive constant $t_0$ (that is, there is a homogeneous distribution of pheromone at time t=0).

The total number of ants m, assumed constant over time, is an important parameter: too many ants would quickly reinforce sub-optimal trails and lead to early convergence to bad solutions, whereas too few ants would no produce the expected synergistic effects of cooperation because of the (otherwise necessary) process of pheromone decay. Set m=n, that is, using as many ants as there are cities in the problem, provides a good tradeoff. At the beginning of each tour, ants are either placed randomly on the nodes (cities), or one ant is placed on each city (no significant difference in performance was observed between the two choices).

In an effort to improve AS performance, also “elitist ants” (as ***elitist strategy*** used in ***genetic algorithms***) were introduced. An elitist ant is an ant which reinforces the edges belonging to $T^+$, the best tour found from the beginning of the trial, by a quantity $Q/L^+$, where $L^+$ is the length of $T^+$.

At every iteration e elitist ants are added to the usual ants so that the edges belonging to $T^+$ get an extra reinforcement $e \cdot Q/L^+$. The idea is that the pheromone trail of $T^+$, so reinforced, will direct the search of all the other ants in probability toward a solution composed of some edges of the best tour itself. Experiments have shown that a small number of elitist ants can improve the algorithm’s performance.

Referring to the time complexity of AS is $O(t \cdot n^2 \cdot m)$, where t is the number of iterations done. If m=n, that is, if the number of ants is equal to the number of cities, the time complexity becomes $O(t \cdot n^3)$

```
/* Initialization*/
For every edge ( i,j ) do
    The initial amount of pheromone on edges
    t_ij(0) = t_0
End For

For k=1 to m do
    Place ant k on a randomly chosen city
End For

Let T+ be the shortest tour found from beginning and L+ its length
For t=1 to t_max do
    For k=1 to m do
        Build tour T^k(t) by applying n-1 times the following step:
        Choose the next city j with probability
```

$$p_{ij}^{k}(t) = \frac{[t_{ij}(t)]^{a} \cdot [h_{ij}]^{b}}{\sum_{l \in J_i^k} [t_{ij}(t)]^{a} \cdot [h_{ij}]^{b}}$$

```
        where i is the current city
    End For
For k=1 to m do
    Compute the length L^k(t) of the tour T^k(t) produced by ant k
End For
If an improved tour is found then
Update T+ and L+

For every edge ( i,j ) do
    Update pheromone trails by applying the rule:
```

$$t_{ij}(t) \leftarrow (1-r) \cdot t_{ij}(t) + \Delta t_{ij}(t) + e \cdot \Delta t_{ij}^{e}(t) \text{ where}$$

$$\Delta t_{ij}(t) = \sum_{k=1}^{m} \Delta t_{ij}^{k}(t)$$

$$\Delta t_{ij}^{k}(t) = \begin{cases} Q / L^{k}(t) & \text{if } (i, j) \in T^{k}(t) \\ 0 & \text{if } (i, j) \notin T^{k}(t) \end{cases}$$

and

$$\Delta t_{ij}^{e}(t) = \begin{cases} Q / L^{+} & \text{if } (i, j) \in T^{+} \\ 0 & \text{otherwise,} \end{cases}$$

```
End For

For every edge (i,j) do
t_ij(t+1) = t_ij(t)
```

```
End For

End For
Print the shortest tour T+ and its length L+
Stop

/*Initial condition*/
/*Values of parameters used in experiments*/
a =1,  b =5,  r =0.5, m=n, Q=100, t0 =10-6, e=5
```

**Figure 16 High-level description of AS-TSP**

ACS – Ant Colony System represents an improved algorithm. Four modifications of Ant System: ***different transition rule, a different pheromone trail update rule, the use of local updates of pheromone trail to favour exploration, and the use of a candidate list to restrict the choice of the next city to visit***.

The transition rule is modified to allow explicitly for exploration. An ant k on city i chooses the city j to move to following the rule:

$$j = \begin{cases} \arg\max_{u \in J_i^k} \{[t_{iu}(t)] \cdot [h_{iu}]^b\} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases}$$

where q is a random variable uniformly distributed over [0,1], $q_0$ is a tunable parameter $(0 \leq q_0 \leq 1)$, and $J \in J_i^k$ is a city that is randomly selected according to probability

$$p_{iJ}^k(t) = \frac{[t_{iJ}(t)] \cdot [h_{iJ}]^b}{\sum_{l \in J_i^k} [t_{il}(t)] \cdot [h_{il}]^b}$$

which is very similar to the transition probability used by Ant System. We see therefore that the ACS transition rule is identical to Ant System's when $q > q_0$, and is different when $q \leq q_0$. More precisely, $q \leq q_0$ corresponds to an exploitation of the knowledge available about the problem, that is, the heuristic knowledge about distances between cities and the learned knowledge memorized in the form of pheromone trails, whereas $q > q_0$ favours more exploration. Cutting exploration by tuning $q_0$ allows the activity of the system to concentrate on the best solutions instead of letting it explore constantly. It is clear that tuning $q_0$ is similar to tuning temperature in simulated annealing: when $q_0$ is close to 1, only the locally optimal solution is selected (but a combination of locally optimal solutions may not result in a globally optimal solution), whereas when $q_0$ is close to 0, all local solutions are examined, although a larger weight is given to locally optimal solutions (unlike simulated annealing, where all states have similar weights at high temperature). It is therefore possible in principle to progressively freeze the system by tuning $q_0$

from 0 to 1, in order to favour exploration in the initial part of the algorithm and then favour exploitation. This possibility has not been explored yet.

In ACS only the ant that generated the best tour since the beginning of the trial is allowed to globally update the concentrations of pheromone on the branches. The ants therefore are encouraged to search for paths in the vicinity of the best tour found so far. In other words, exploration is more directed. Another difference is that in Ant System the pheromone trail updating rule is applied only to the edges belonging to the best tour since the beginning of the trial. The updating rule is:

$$t_{ij}(t) \leftarrow (1-r)\cdot t_{ij}(t) + r\cdot \Delta t_{ij}(t)$$

where (i,j)'s are the edges belonging to $T^+$, the best tour since the beginning of the trial, $r$ is a parameter governing pheromone decay and

$$\Delta t_{ij}(t) = 1/L^+$$

where $L^+$ is the length of $T^+$. We see that this procedure allows only the best tour to be reinforced by a global update. However, local updates are also performed, so other solutions can emerge.

Repast [34] is a software framework for creating agent based simulations using the Java language (requires version Java 1.2 or greater). It provides a library of classes for creating, running, displaying and collecting data from an agent based simulation. In addition, Repast can take snapshots of running simulations, and create *Quicktime movies* of simulations.

Repast borrows much from the Swarm simulation toolkit and can properly be termed "Swarm-like." In addition, Repast includes such features as run-time model manipulation via GUI (Graphical User Interface) widgets first found in the Ascape [35] simulation toolkit.

On a more technical level, Repast envisions a simulation as a state machine whose state is constituted by the collective states of all its components. These components can be divided up into infrastructure and representation. The infrastructure is the various mechanisms that run the simulation, display and collect data and so forth. The representation is what the simulation modeller constructs, the simulation model itself. The state of the infrastructure is then the state of the display, the state of the data collection objects etc. The state of the representation is the state of what is being modelled, the current values of all the agents' variables, the current value of the space or spaces in which they operate, as well as the state of any other representation objects (e.g. aggregate quasi-independent "institution" objects). The history of the simulation as a software phenomenon is the history of both these states, while the history of the simulation as a simulation is the history of the representational states. In Repast as in Swarm, any changes to the states of the infra-structural components and the representational components occur through a Schedule[36] object. In short then, Repast allows a user to build a simulation as a state machine in which all the changes to the state occur through a schedule. This provides clarity and extensibility both for the simulation writer/user as well as the software designer seeking to extend the toolkit. [37]

Summarizing, we have a computational model which allows making sophisticated routes with simple rules. The system is complex however the agents are not.

## 4.3. MPEG-21

MPEG-21 describes a standard that defines the description of content and also processes for accessing, searching, storing and protecting the ***copyrights*** of content. It is a comprehensive standard framework for networked digital multimedia designed by the Moving Picture Experts Group. The intent is that the framework, once finished, will cover the entire multimedia content delivery chain encompassing content creation, production, delivery, trade and consumption. It means that is involved in IPR (Intellectual Property Rights) negotiation processes.

A novel paradigm is provided in which the user can create, modify, protect, adapt and consume digital items (DI) (which are a fundamental unit of distribution and transaction and it is the representation of a work, and as such, it is the thing that is acted upon (managed, described, exchanged, etc.) within the model). Such paradigm accepts all participants of the content value-chain as Users (operators, service providers, content creators, and end-users) while allowing each of them to have simultaneously different roles. Table 2 shows the current state of MPEG-21 standard. We have focused in MPEG-21 parts 5 and 6. They are explained in sections 4.3.1 and 4.3.2 respectively.

**Table 2. The *MPEG-21 standard* is currently formed by sixteen parts (Last update: 08/31/2004)**

| | **Stage (ballot due date)** | **WD** | **CD PDAM PDTR** | **FCD FPDAM DCOR** | | **IS AMD TR COR** |
|---|---|---|---|---|---|---|
| Part 1: Vision, Technologies and Strategy [38]. | TR | to be published | | | | |
| Part 2: Digital Item Declaration (DID) [39] | IS | Published (2003-03-15) | | | | |
| Part 2: Digital Item Declaration (DID) [40] | FCD (2004-12-01) | - | - | - | 2005-01 | 2005-04 |
| Part 3: Digital Item Identification (DII) [41] | IS | Published (2003-04-01) | | | | |
| Part 4: Intellectual Property Management and Protection (IPMP) [42] | WD | - | 2004-07 | 2005-01 | 2005-07 | 2005-09 |
| ***Part 5: Rights Expression Language (REL)*** | IS | Published (2004-04-01) | | | | |
| ***Part 6: Rights Data Dictionary (RDD) [43]*** | IS | Published (2004-05-15) | | | | |
| ***Part 6: Rights Data Dictionary (RDD) [44]*** TECHNICAL CORRIGENDUM 1 | DCOR (2004-10-22) | - | - | - | 2004-10 | 2005-01 |
| Part 7: Digital Item Adaptation (DIA) [45] AMENDMENT 1: DIA Conversions and Permissions | PDAM (2004-10-22) | - | - | 2004-10 | 2005-04 | 2005-07 |
| Part 8: Reference Software [46] | CD (2004-06-21) | - | - | 2004-07 | 2005-01 | 2005-04 |
| Part 9: File Format [47] | FCD (2004-12-13) | - | - | - | 2005-01 | 2005-04 |
| Part 10: Digital Item Processing (DIP) [48] | CD (2004-04-28) | - | - | 2004-07 | 2004-10 | 2005-01 |
| Part 11: Evaluation Methods for Persistent Association Technologies [49] | TR | to be published | | | | |
| Part 12: Test Bed for MPEG-21 Resource Delivery [50] | PDTR (2004-04-15) | - | - | - | 2004-07 | 2004-10 |
| Part 13: Scalable Video Coding [51] | WD | - | 2005-10 | 2006-03 | 2006-07 | 2006-10 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Part 14: Conformance Testing [52] | WD | - | 2004-07 | 2005-01 | 2005-07 | 2005-10 |
| Part 15: Event Reporting (ER) [53] | WD | - | 2004-10 | 2005-04 | 2005-10 | 2006-01 |
| Part 16: Binary Format [54] | CD (2004-10-22) | - | - | 2004-10 | 2005-04 | 2005-07 |

### 4.3.1. Rights Expression Language (REL)

This MPEG-21 standard specifies the syntax and semantics of a Rights Expression Language. MPEG chose XrML[55] as the basis for the development of the MPEG-21 Rights expression language.

MPEG-21 Rights Expression Language (REL) specifies the syntax and semantics of the language for issuing rights for Users to act on Digital Items, their Components, Fragments, and Containers.

The most important concept in REL is the license that conceptually is a container of grants, each one of which conveys to a principal the sanction to exercise a right against a resource. A license if formed by the following elements:

***Title***: It provides a descriptive phrase about the License that is intended for human consumption in user interfaces. Automated processors must not interpret semantically the contents of such title elements.

***Inventory***: It is used for defining variables within a License. In the Inventory element of a license can be defined LicensePart elements that in turn can have licensePartId attributes that can be referenced from elsewhere in the license.
Therefore, REL provides a syntactic mechanism for reducing redundancy and verbosity in Licenses that can be used throughout a License.

***Grant or GrantGroup***: The Grants and GrantGroups contained in a license are the means by which authorization policies are conveyed in the REL architecture. Each Grant or GrantGroup that is an immediate child of a license exists independently within that license, no collective semantic (having to do with their particular ordering or otherwise) is intrinsically associated with the presence of two or more of them within a certain license.

***Other information***: Using the wildcard construct from XML Schema, a License provides an extensibility hook within which license issuers may place additional content as they find appropriate and convenient. This can be useful for conveying information that is peripherally related to, for example, authentication and authorization, but is not part of the REL core infrastructure.

It should, however, be carefully understood that not all processors of REL licenses will understand the semantics intended by any particular use of this extensibility hook. Processors of the license may choose wholly at their own discretion to completely ignore any such content that might be present therein.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <r:grant>
        <r:keyHolder licensePartID="Alice">
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
        <mx:play/>
        <r:digitalResource>
            <r:nonSecureIndirect URI="http://www.webmusic.com/theEnd.mp3"/>
        </r:digitalResource>
        <r:validityInterval>
            <r:notBefore>2004-01-01T00:00:00</r:notBefore>
            <r:notAfter>2004-12-31T12:59:59</r:notAfter>
        </r:validityInterval>
    </r:grant>
    <r:issuer>
        <r:keyHolder licensePartID="MusicDist">
            <r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>
                        <dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
                        <dsig:Exponent>AQABAA==</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </r:info>
        </r:keyHolder>
    </r:issuer>
</r:license>
```

**Figure 17. REL License Example**

### 4.3.2. Rights Data Dictionary (RDD)

The goal of the MPEG Rights Data Dictionary standard (ISO/IEC 21000-6) is to meet the needs rights owners and consumers by providing a consistent vocabulary for digital rights management.

The core of the RDD is a set of clear, consistent, structured, integrated and uniquely identified terms to express the rights that content owners may wish to grant to users, in this case REL, another part of the MPEG-21 standard. These rights are expressed as verbs (such as *Play, Print and Adapt*). While the dictionary contained in the standard is fairly basic, it is recognized that terms may in future be required to be extremely specialized and granular. As the dictionary is founded on a logical data model, it will be possible to extend it to meet the needs of content owners and other participants in the value chain. For instance, it could be used to express a very specific set of semantic requirements from a rights holder or a technology provider. In this way, it could provide semantics expressing the parameters of a small device which can then be mapped to rights semantics in a complex grant made by a content owner.

In addition to providing semantics to support the MPEG Rights Expression Language, the RDD is designed to support the mapping of terms from different namespaces. For instance, there may be a need to create a rights expression that requires semantics from two content domains. While the semantics in each domain will be interpretable within the domain, cross domain interpretation would be difficult without some kind of translation process. The mapping process will be the responsibility of the Registration Authority, to be set up by ISO in due course.

The RDD Dictionary has the characteristics of a structured ontology, in which meaning, once it has been defined, can be passed on from one term to another by logical rules of association such as inheritance and opposition. An ontology, in this context, is a structured catalogue of entities in which meaning, once defined, can be passed on from one term to another by logical rules of association such as inheritance and opposition. The structure of this ontology is designed to provide a set of well-defined terms for use in rights expressions governing the use of Digital Items. In recognition of the great diversity and complexity associated with multimedia content, it is also designed to represent as many different specializations of meaning as its users require, and to show their relationships in a structured way in order to support the mapping and transformation of terms between different schemas and systems.

The Standardized Terms in the RDD Dictionary are therefore not a closed list, but the foundations of a widely extensible Rights Data Dictionary.

The RDD System is comprised of the following three elements:
- The Specifications contained in the RDD Standard.
- A Dictionary, the Terms and their TermAttributes defined according to this specification.

- A Database, the tool containing the RDD Dictionary and supporting its maintenance.

The use of the RDD System will facilitate the accurate exchange and processing of information between interested parties involved in the administration of rights, the use of Digital Items and the Rights Expression Language.

The RDD System is designed to support the mapping of Terms from different namespaces. Such mapping will enable the transformation of metadata from the terminology of one namespace (or Authority) into that of another namespace (or Authority). Mapping, to ensure minimum ambiguity or loss of semantic integrity, will be the responsibility of the Registration Authority.

The RDD Dictionary is a *prescriptive* Dictionary, in the sense that it defines a single meaning for a Term represented by a particular RddAuthorized TermName, but it is also *inclusive* in that it can recognize the prescription of other Headwords and definitions by other Authorities and incorporates them through mappings. The RDD Dictionary also supports the circumstance that the same name may have different meanings under different Authorities. Therefore Terms that are directly authorized by the RDD Registration Authority neither define nor prescribe intellectual property rights or other legal entities.

The RDD defines the meaning for the terms defined in the REL. Table 3 summarizes the ***ActTypes*** in this part of the MPEG-21 standard that have been defined in response to requirements identified in the process of developing the REL and RDD Standards, particularly focussed on common processes in the use and adaptation of Digital Resources.

**Table 3. RDD** *ActType* **and its parent and definition**

| **ActType** | **Parent** | **Definition** |
|---|---|---|
| Adapt | Derive, ChangeTransiently | To ChangeTransiently an existing Resource to Derive a new Resource. |
| Delete | Destroy | To Destroy a DigitalResource. |
| Diminish | Adapt | To Derive a new Resource which is smaller than its Source. |
| Embed | Relate | To put a Resource into another Resource. |
| Enhance | Adapt | To Derive a new Resource which is larger than its Source. |
| Enlarge | Modify | To Modify a Resource by adding to it. |
| Execute | Activate | To execute a DigitalResource. |
| Install | UseTool | To follow the instructions provided by an InstallingResource. |
| Modify | Change | To Change a Resource, preserving the alterations made. |
| Move | Modify | To relocate a Resource from one Place to another. |
| Play | Render, Perform | To Derive a Transient and directly Perceivable representation of a Resource. |
| Print | Render, Fix | To Derive a Fixed and directly Perceivable representation of a Resource. |
| Reduce | Modify | To Modify a Resource by taking away from it. |
| Move | Modify | To relocate a Resource from one Place to another. |
| Play | Render, Perform | To Derive a Transient and directly Perceivable |

| | | |
|---|---|---|
| | | representation of a Resource. |
| Uninstall | UseTool | To follow the instructions provided by an UninstallingResource. |

When using RDD ***actTypes*** in REL is not only important the meaning of the act, it is also important the RDD Hierarchy of the term, for instance, an authorization can be erroneous if the hierarchy of the right is not taken into account.

## 4.4. Graphs models

The section goal is to provide the basis for understanding how systems can be modelled and which tools as statistical analysis ( see sections 4.4.2 to 4.4.4 ) or visualization tools can be applied ( see section 4.4.5 ).

Sections 6.6 to 6.9 make use of this theoretic basis in order to justify that the architecture is represented by graphs, therefore it can be feasible scalable and for developing a new method designed for a special kind of graph to be analysed statistically.

Environment as context or even agents can be conceptualised as nodes representing entities relating to each other, the differences among these relations make the distinction in the graphs classification. It reassures the idea of observer and observed can not be isolated because there is always a relation.

In mathematical terms a network is represented by a graph. A **graph** is a pair of sets *G = {P,E}*, where ***P*** is a set of ***N*** **nodes** (or vertices or points) $P_1$, $P_2$, … $P_N$ and ***E*** is a set of **edges** (or links or lines) that connect two elements of *P*. Graphs are usually represented as a set of dots, each corresponding to a node, two of these dots being joined by a line if the corresponding nodes are connected.

Note the edges are just pairs of vertices, i.e. pairs of elements of $\forall e \in e, e \in P \times P$. Thus the ends of edges in a graph always end on vertices of the graph. The opposite is not true, that is, a vertex need not have any edges connected to it. Whenever people talk about removing vertices from a graph, they mean implicitly that you also remove all edges connected to that vertex; otherwise you would not have a valid graph.

The number of edges between vertices $(i, j) \in P$ is the total **number of vertices** and the **order** of a graph: $|P(G)|$ , while the **size** of a graph is the total number of edges and here will be denoted by $|E(G)|$

Table 4 and Table 5 summarize **classification and topics** of graphs.

**Table 4 Graphs classification**

| | |
|---|---|
| Directed graphs | Have a direction or arrow associated with their edges so that the edge (i,j) is not the same as an edge (j,i) and we can have $e_{ij} \neq e_{ji}$. |
| Coloured graph | They have vertices carrying labels, sometimes also called |

| | |
|---|---|
| | colours even if we do not use colour as the actual label. In an unlabelled graph all vertices are of the same type. For instance for the London tube system, se could let stations be vertices and we could include a label indicating which lines the stations lie on. |
| Simple graphs | It has no more than one link between any pair of vertices, and no edges from a vertex back to the same vertex. $e_{ii} = 0 \;\; \mathrm{i} \in \mathrm{P}, \mathrm{e}_{\mathrm{ij}} \in \{0,1\} \; i, j \in P$ All the vertices are identical, i.e., they are uncoloured. Thus $E \leq N(N-1)/2$ |
| Pure graphs | It is the simple unweighted unlabelled graph. For instance it can be obtained by dropping all weights and labels, all edges connecting the same vertex (a small loop) and all edges apart from one which connect the same pair of vertices. This is my definition which I use below. |
| Complete graph | They have all vertices connected to all other vertices by an edge. |
| Sparse graph | They have many fewer connections than is possible. For a simple graph, the maximum number of connections is clearly N-1 for each vertex, thus we require that: K<<N. If a graph is too sparse, $K \sim 1$, then it may be hard to distinguish different types. |

**Table 5. Graphs topics**

| | |
|---|---|
| Components | They are individual connected pieces of a graph. So a component of a graph is a subgraph, i.e. a part of a graph. It is a connected graph and every component is maximal – no other edge or vertex from the original graph can be added to the subgraph which keeps the subgraph connected. |
| Weighted edges | They have a number associated with them. If there are multiple edges between the same pair of vertices, this might be equivalent to a single edge with a weight. It could represent a physical quantity, some sort of capacity for flow along the edge (cost of creating a road), or the physical distance associated wit the two vertices at the |

| | |
|---|---|
| | end of the edge (distance of road between two towns), or whatever. Thus $e_{ij}$ can be any number (real, integer or whatever) as well as zero.<br>Again, unless noted, edges and graphs will be assumed to be unweighted, so all edges carry an equal weight, and $e_{ij} \in Z$ |
| Clustering coefficient | Complex networks exhibit a large degree of clustering. If we consider a node in a random graph and its first neighbours, the probability that two of these neighbours are connected is equal with the probability that two randomly selected nodes are connected. Consequently the clustering coefficient of a random graph is<br>$C_{rand} = p = \frac{\langle k \rangle}{N}$ |
| Characteristic path length | For random graphs, we have very short distances if we analyse path length (d), defined as the average minimum distance between any pair of nodes. It can be shown that in random graphs:<br>$d_{random} \sim \log(n) / \log(z)$.<br>Graphs where $d \sim d_{random}$ are said to be '*small-world*' networks that propagate information very efficiently. |

| | |
|---|---|
| Diameter and connectivity | The **diameter** of a graph is the **maximal distance** between any pair of its nodes. Strictly speaking, the diameter of a disconnected graph (i.e. made up of several isolated clusters) is infinite, but it can be defined as the **maximum diameter** of its clusters. Random graphs tend to have small diameters, provided p is not too small. The reason for this is that a random graph is likely to be spreading: with large probability the number of nodes at a distance l from a given node is not much smaller than $\langle k \rangle^l$. Equating $\langle k \rangle^l$ with N we find that the diameter is proportional with ln(N)/ ln($\langle k \rangle$), thus it depends only logarithmically on the number of nodes.<br>The diameter of a random graph has been studied by many authors [56]. A general conclusion is that for most values of p, almost all graphs have precisely the same diameter. This means that when we consider all graphs with N nodes and connection probability p, the range of values in which the diameters of these graphs can vary is very small, usually concentrated around<br>$d = \frac{\ln(N)}{\ln(pN)} = \frac{\ln(N)}{\ln(\langle k \rangle)}$<br>In the following we summarize a few important results:<br>If $\langle k \rangle$ = pN < 1 the graph is composed of isolated trees and its diameter equals the diameter of a tree.<br>If $\langle k \rangle$ > 1 a giant cluster appears. The diameter of the graph equals the diameter of the giant cluster if $\langle k \rangle$ ≥ 3.5, and is proportional to ln(N)/ ln($\langle k \rangle$ ).<br>If $\langle k \rangle$ ≥ ln(N) the graph is totally connected. Its diameter is concentrated on a few values around ln(N)/ ln($\langle k \rangle$ ). |

## 4.4.1. Random graphs: Erdös and Rényi model

Graph theory has its origins in the 18th century in the work of Leonard Euler, the early work concentrating on small graphs with a high degree of regularity. In the 20th century graph theory has become more statistical and algorithmic. A particularly rich source of ideas has been the study of random graphs, graphs in which the edges are distributed randomly. Networks with a complex topology and unknown organizing principles often appear random, thus random graph theory is regularly used in the study of complex networks, networks which behaviour is 'at the edge of chaos' .

The theory of random graphs was founded by Paul Erdös and Alfréd Rényi[57],[58], after Erdös discovered that probabilistic methods were often useful in tackling problems in graph theory. In addition, there is an insightful review of the parallels between phase transitions[59] and random graph theory. In the following we will briefly describe the most important results of random graph theory, focusing on the aspects that are of direct relevance to complex networks.

In their classic first article on random graphs, Erdös and Rényi define a **random graph** as *N* labeled nodes connected by *n* edges which are chosen randomly from the $\frac{N(N-1)}{2}$ possible edges. In total there are $C^{n}_{\frac{N(N-1)}{2}}$ graphs with *N* nodes and *n* edges, forming a probability space in which every realization is equiprobable. Thus if $G_0$ is a graph with *N* nodes and *n* edges, the probability of obtaining it as a result of a random graph generating process is $P(G_0) = 1/C^{n}_{\frac{N(N-1)}{2}}$.

Although the number of nodes and edges seem to be independent parameters, in most cases n has to depend on N in order to satisfy certain basic requirements of a random graph (for example, N - 1 = n = N(N - 1)/2 assures that the graph is not disconnected and there are no duplicate edges). An alternative and equivalent definition of a random graph is called the binomial model. In this model we start with N nodes P1, P2, … PN, every pair of nodes being connected with probability p. Consequently, the total number of edges is a random variable with the expectation value

$$E(n) = p\frac{N(N-1)}{2}.$$

If G0 is a graph with nodes P1, P2, … PN and n edges (see Figure 18), start with N = 10 isolated nodes (upper panel), then connect every pair of nodes with probability p. The lower panel of the figure shows two different stages in the graph's development, corresponding to p = 0.1 and p = 0.15. We can notice the emergence of trees (drawn with dashed lines) and cycles (drawn with dotted lines) in the graph, and a connected cluster which unites half of the nodes at p = 0.15. The probability of obtaining it by this graph construction process is

$$P(G_0) = p^n (1 - p) \frac{N(N-1)}{2}$$

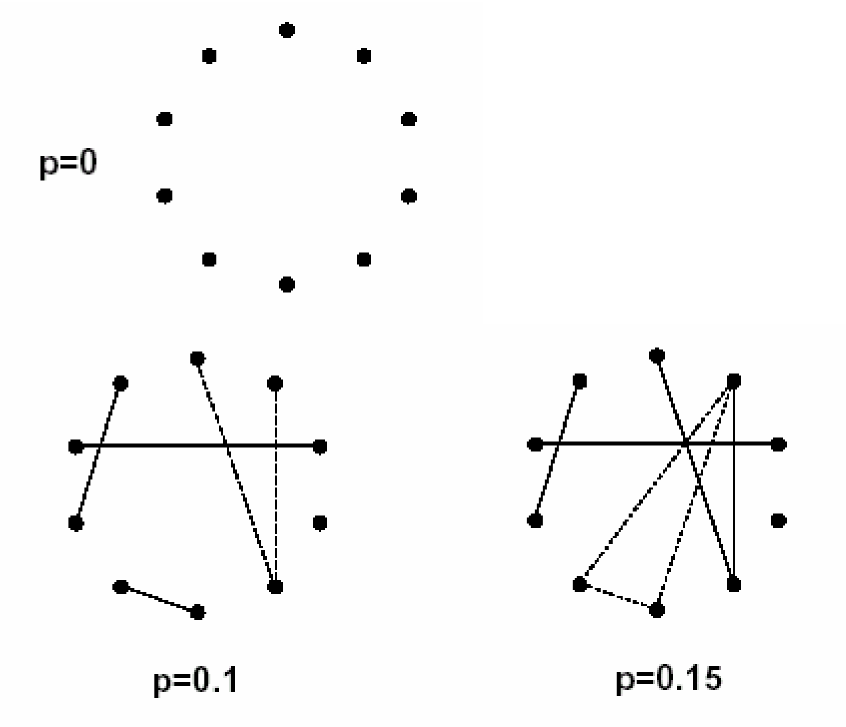



**Figure 18 Illustration of the graph evolution process for the Erdös-Rényi model.**

Random graph theory studies the properties of the probability space associated with graphs with *N* nodes as *N ? 8*. Many properties of such random graphs can be determined using probabilistic arguments. In this respect Erdös-Rényi used the definition that almost every graph has a property *Q* if the probability of having *Q* approaches 1 as *N ? 8*. Among the questions addressed by Erdös-Rényi some have direct relevance to understanding networks as well, such as:

- Is a typical graph connected?
- Does it contain a triangle of connected nodes?
- How does its diameter depend on its size?

The construction of a random graph is often called in the mathematical literature an evolution: starting with a set of *N* isolated vertices, the **graph develops by the successive addition of random edges**. The graphs obtained at different stages of this process correspond to larger and larger connection probabilities *p*, eventually obtaining a fully connected graph (having the maximum number of edges *n* =*N*(*N* - 1)/2) for *p ?* 1. The **main goal** of **random graph theory** is to determine at **what connection probability *p* a particular property** of a graph will most likely arise.

The greatest discovery of Erdös and Rényi was that many important properties of random graphs appear quite suddenly. That is, at a given probability either almost every graph has the property *Q* (e.g. every pair of nodes is connected by a path of consecutive edges) or on the contrary, almost no graph has it. The transition from a property being very unlikely to being very likely is usually swift. For many such properties there is a unique critical probability $p_c(N)$.

If $p(N)$ **grows slower** than $p_c(N)$ as *N ? 8*, then almost every graph with connection probability $p(N)$ fails to have $Q$.
If $p(N)$ **grows somewhat faster** than $p_c(N)$, then almost every graph has the property $Q$. Thus the probability that a graph with $N$ nodes and connection probability $p = p(N)$ has property $Q$ satisfies

$$\lim_{N\to\infty} P_{N,p}(Q) = \begin{cases} 0 \text{ if } \dfrac{\mathrm{p(N)}}{\mathrm{p_c}(N)} \to 0 \\ 1 \text{ if } \dfrac{\mathrm{p(N)}}{\mathrm{p_c}(N)} \to \infty \end{cases}$$

An important note is in order here. Physicists trained in critical phenomena will recognize in $p_c(N)$ the critical probability familiar in **percolation**. In the physics literature usually the system is viewed at a fixed system size $N$ and then the different regimes reduce to the question whether $p$ is smaller or larger than $p_c$. The proper value of $p_c$, that is, the limit $p_c = p_c$(*N ? 8*) is obtained by finite size scaling. The basis of this procedure is the assumption that this limit exists, reflecting **the fact** that ultimately **the percolation threshold does not depend on the system size**. This is usually the case in finite dimensional systems which include most physical systems of interest for percolation theory and critical phenomena. In contrast, **networks** are, by definition**, infinite dimensional**: the number of neighbours a node can have increases linearly with the **system size**. Consequently, in random graph theory the occupation probability is defined as a function of the **system size**: $p$ represents the fraction of the edges which are present from the possible $N(N-1)/2$.

Larger graphs with the same $p$ will contain more edges, and consequently properties like the appearance of **cycles** could occur earlier (at smaller $p$) in **large graphs** than in smaller ones. This means that for many properties $Q$ in random graphs there is no unique, $N$-independent threshold, but **we have to define a threshold function which depends on the system size**, and for the "percolation threshold" we have $p_c$(*N ? 8*) *?* 0. On the other hand, we will see that the average number of edges per node (also called the *average degree of the graph*)

$$\langle k \rangle = 2n/N = p(N-1) \simeq \mathrm{pN}$$

does have a **critical value** which is **independent of the system size**. In the coming lines these ideas are illustrated by looking at the **emergence** of various sub-graphs in random graphs.

The first property of random graphs studied by Erdös and Rényi in 1959 was the appearance of sub-graphs. A graph $G_2$ consisting of a set $P_2$ of nodes and a set $E_2$ of edges is a sub-graph of a graph $G_1 = \{P_1, E_1\}$ if all nodes in $P_2$ are also nodes of $P_1$ and all edges in $E_2$ are also edges of $E_1$. The simplest examples of sub-graphs are cycles, trees and complete sub-graphs. A cycle of order *k* is a closed loop of *k* edges such that every two consecutive edges and only those have a common node. One way to graphically imagine a cycle is by drawing a polygon: a triangle is a cycle of order 3, while a rectangle is a cycle of order 4. The average degree of a cycle is equal to 2, since every node has two edges. The opposite of closed loops are the trees, which cannot form closed loops. More precisely, **a graph** is called a **tree** of order ***k*** if it has ***k* nodes** and ***k*-1 edges**, and none of its sub-graphs is a cycle. The average degree of a tree of order *k* is $\langle k \rangle = 2 - 2/k$ , approaching 2 for large trees. Complete sub-graphs of order *k* contain *k* nodes and all the possible *k*(*k* - 1)/2 edges, in other words being completely connected.

If the evolution process is considered to be described above for a graph $G = G_{N,p}$. *N* isolated nodes start, then connect every pair of nodes with probability *p*. For small connection probabilities the edges are isolated, but as *p*, and with it the number of edges, increases, two edges can attach at a common node, forming a tree of order 3. An interesting problem is to determine the critical probability $p_c(N)$ at which almost every graph *G* contains a tree of order 3. Most generally, it is possible to ask whether there is a critical probability which marks the appearance of arbitrary sub-graphs consisting of *k* nodes and *l* edges.

In random graph theory there is a rigorously proven answer to this question[58]. Consider a random graph $G = G_{N,p}$. In addition, consider a small graph F consisting of k nodes and l edges. In principle, the random graph G can contain several such subgraphs F. Our first goal is to determine how many such subgraphs exist.

The k nodes can be chosen from the total number of nodes N in $C_N^k$ ways and the l edges are formed with probability $p^l$. In addition, it is possible to permute the k nodes and potentially obtain k! new graphs *(the correct value is k!/a, where a is the number of graphs which are isomorphic to each other)*. Thus the expected number of Sub-graphs F contained in G is

$$E(X) = C_N^k \frac{k!}{a} p^l \simeq \frac{N^K p^l}{a}$$

This notation suggests that the actual number of such sub-graphs, X, can be different from E(X), but in the majority of the cases it will be close to it. Note that the sub-graphs do not have to be isolated, i.e. there can exist edges with one of their nodes inside the sub-graph, but the other outside of it.

If p(N) is such that $p(N)= N^{k/l} \rightarrow 0$ as $N \rightarrow 0$, the expected number of sub-graphs $E(X) \rightarrow 0$, i.e. almost none of the random graphs contains a sub-graph F. On the other hand, if $p(N) = cN^{k/l}$, the mean number of sub-graphs is a finite number, denoted by $l = c^l / a$, indicating that this function might be the critical probability. The validity of this finding can be tested by calculating the distribution of sub-graph numbers, Pp(X = r), obtaining[60]

$$\lim_{N\to\infty} P_P(X = r) = e^{-l}\ \frac{l^r}{r!}$$

The probability that G contains at least one sub-graph F is then

$$P_P(G \supset F) = \sum_{r=1}^{\infty} P_P(X = r) = 1 - e^{-l}$$

which converges to 1 as c increases. For p values satisfying $pN^{k/l} \to \infty$ the probability $Pp(G \supset F)$ converges to 1, thus, indeed, the critical probability at which almost every graph contains a sub-graph with k nodes and l edges is $p_c(N) = cN$ where k=l.

A few important special cases:

(a) The critical probability of having a **tree** of order k is $p_c(N) = cN^{-k/(k\_1)}$;
(b) The critical probability of having a **cycle** of order k is $p_c(N) = cN^{-1}$;
(c) The critical probability of having a **complete** sub-graph of order k is $p_c(N) = cN^{-2/(k-1)}$

Consider a random graph with N nodes and assume that the connection probability p(N) scales as $N^z$ where z is a tuneable parameter that can take any value between $-\infty$ and 0. For z less than –3/2 almost all graphs contain only isolated nodes and edges. When z passes through –3/2, trees of order 3 suddenly appear. When z reaches –4/3, trees of order 4 appear, and as z approaches -1, the graph contains trees of larger and larger order. However, as long as z < -1, such that the average degree of the graph $\langle k \rangle = pN \to 0$ as $N \to \infty$, the graph is a union of disjoint trees, and cycles are absent. Exactly when z passes through -1, corresponding to $\langle k \rangle$ = const, even though z is changing smoothly, the asymptotic probability of cycles of all orders jumps from 0 to 1. Further results can be derived for z = -1, i.e. when we have $p \propto N^{-1}$ and the average degree of the nodes is $\langle k \rangle$ =const. For $p \propto N^{-1}$a random graph contains trees and cycles of all order, but so far we have not discussed the size and structure of a typical graph component. A component of a graph is by definition a connected, isolated subgraph, also called a cluster in network research and percolation theory.

Figure 19 illustrates this concept. For p~N3/2 $\to$ 0 the graph consists of isolated nodes and edges. For p~N-3/2 trees of order 3 appear, at p~N-4=3 trees of order 4. At p~N-1 trees of all orders are present, and at the same time cycles of all orders appear. The probability p~N-2=3 marks the appearance of complete subgraphs of order 4 and p~N-1=2 corresponds to complete subgraphs of order 5. As z approaches 0, the graph contains complete subgraphs of increasing order.

As Erdös and Rényi show, *there is an abrupt change in the cluster structure of a random graph as <k> approaches 1*. If $0 < \langle k \rangle < 1$, almost surely all clusters are either trees or clusters containing exactly one cycle. Although cycles are present, almost all nodes belong to trees. The mean number of clusters is of order N - n, where n is the number of edges, i.e. in this range by adding a new

edge the number of clusters decreases by 1. The largest cluster is a tree, and its size is proportional to ln N.
When $\langle k \rangle$ passes the threshold $\langle k \rangle$c = 1, the structure of the graph changes abruptly. While for $\langle k \rangle$ < 1 the greatest cluster is a tree, for $\langle k \rangle$c = 1 it has approximately $N^{2/3}$ nodes and has a rather complex structure. Moreover for $\langle k \rangle$ > 1 the greatest (giant) cluster has [1 - f($\langle k \rangle$)]N nodes, where f(x) is a function that decreases exponentially from 1 to 0 for x $\rightarrow \infty$. Thus a finite fraction S = 1 - f($\langle k \rangle$) of the nodes belongs to the largest cluster. Except for this giant cluster, all other clusters are relatively small, most of them being trees, the total number of nodes belonging to trees being Nf($\langle k \rangle$). As $\langle k \rangle$ increases, the small clusters coalesce and join the giant cluster, the smaller clusters having the higher chance of survival.

Thus at $p_c \simeq 1/N$ the random graph changes its topology abruptly from a loose collection of small clusters to being dominated by a single giant cluster. The results of beginning of the supercritical phase show that in this region the largest cluster clearly separates from the rest of the clusters, its size S increasing proportionally with the separation from the critical probability, $S \propto (p - p_c)$

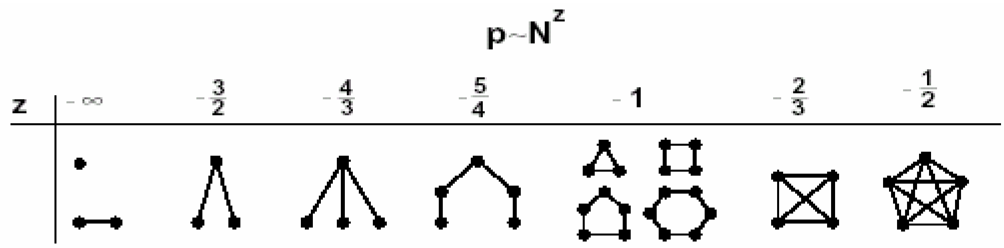



**Figure 19. The threshold probabilities at which different subgraphs appear in a random graph.**

Since the pioneering paper of Erdös and Rényi, much work has concentrated on the existence and uniqueness of the minimum and maximum degree of a random graph.

The results indicate that for a large range of p values both the maximum and the minimum degrees are determined and finite. For example, if P(N)~$N^{-1-1/K}$ (thus the graph is a set of isolated trees of order at most k+1) almost no graph has nodes with degree higher than k. On the other extreme, if p = [ln(N)+k ln(ln(N))+c]/N, almost every random graph has minimum degree of at least k. Furthermore, for a sufficiently high p, respectively if pN= ln(N) $\rightarrow\infty$, the maximum degree of almost all random graphs has the same order of magnitude as the average degree. Thus, despite the fact that the position of the edges is random, a typical random graph is rather homogeneous. The majority of the nodes have the same number of edges.

Another way to characterize the spread of a random graph is to calculate the average distance between any pair of nodes, or the average path length. One expects that the average path length scales with the number of nodes in the same way as the diameter

$$l_{rand} \sim \frac{\ln(N)}{\ln(\langle k \rangle)}$$

It is showed that the average path length of real networks is close to the average path length of random graphs with the **same size**. The product $l_{rand}\ln(<k>)$ is equal to ln(N), so plotting $l_{rand}\ln(<k>)$ as a function of ln(N) for random graphs of different sizes gives a straight line of slope 1.

Any graph G with N nodes can be represented by its adjacency matrix A(G) with N×N elements $A_{ij}$, whose value is $A_{ij} = A_{ji} = 1$ if nodes i and j are connected, and 0 otherwise. The spectrum of graph G is the set of eigenvalues of its adjacency matrix A(G). A graph with N nodes has N eigenvalues $?_j$, and it is useful to define its spectral density as

$$r(l) = \frac{1}{N}\sum_{j=1}^{N} d(l - l_j),$$

which approaches a continuous function if N ? 8 1. The interest in spectral properties is related to the fact that the spectral density can be directly related to the graph's topological features.

One of the most interesting findings of **random graph theory is the existence of a critical probability at which a giant cluster forms**. Translated into network language, it indicates the **existence of a critical probability $p_c$ such that below $p_c$ the network is composed of isolated clusters but above $p_c$ a giant cluster spans the entire network.**

This phenomenon is markedly similar to a percolation transition, a topic much studied both in mathematics and in statistical mechanics. Indeed, the ***percolation transition*** and the ***emergence of the giant cluster*** *are the same phenomenon expressed in different languages*. Percolation theory, however, does not simply reproduce the predictions of random network theory. Asking questions from a different perspective, it addresses several issues that are crucial for understanding real networks, but are not discussed by random graph theory. Consequently, it is important to review the predictions of percolation theory relevant to networks, as they are crucial to understand important aspects of the network topology.

It is curious to observe that **star topology is not only an artificial network representation, also represents a Bose-Einstein quantum level** (a matter state very difficult to find it in nature) as an enterprise system known as '***monopoly***'[61].

It is a hierarchical network and moreover it is decentralized and distributed. There are special nodes called **hubs,** for instance, if "World Wide Web" is analysed (WWW), portals as "www.google.com", "www.yahoo.com" are well known for everyone. In other words, nodes are highly connected and represent these hubs. However, comparatively there are many nodes poorly related. Network history does not give any special material to explain such behaviour as it is going to be explained.

Reviewing history, Tim Berners-Lee was blamed [62] because he proposed a system where URL (Uniform Resource Locations) can disappear! Nowadays, it is not a serious problem. However it means that nodes are going on and off continuously. And it is a grave problem for copyright issues. Software as "*Kazaa*" or "*Emule*" take advantatge of this.

There is a legal void about multimedia content copyright because the traditional environment and the transactional processes have changed. Now, URLs going nowhere are not a problem. However, changes always awake fears. For instance, this was the case with the massive reproduction of 'copy' from 'originals' highlighted by Adorno [63] at the beginning of 20th century. Then society did not accept that 'art', an original object, could be copied and massively distributed.

A new step has to be done in the direction of accepting this network origin and build a natural strategy to face new technologies. To achieve that it is necessary to understand this kind of network nature. "***Everything has value only in so far as it can be exchanged, not in so far as it is something in itself. For consumers the use value of art, its essence, is a fetish, and the fetish --the social valuation which they mistake for the merit of works of art--becomes its only use value, the only quality they enjoy***". Hence the culture industry dissolves the "***genuine commodity character***" that artworks once possessed when exchange value presupposed use value.

Lacking a background in Marxist theory, and desiring to secure legitimacy for "mass art" or "popular culture," too many of Adorno's anglophone critics simply ignore the main point to his critique of the culture industry. His main point is that the culture-industrial replacement of use value by exchange value evidences a fateful shift in the structure of all commodities and therefore in the structure of capitalism itself.

### 4.4.2. Small-World

Real systems do not behave as perfect systems (periodically) nor chaotic. Their behaviour is just in the middle and not predictable.

Recent researching over large networks among scientists have focused on a number of distinctive statistical properties that large networks seem to share as: **Multi-Agent Collaboration networks** [64], technological networks such as **Peer to Peer systems** [65], the **World Wide Web**[66], **power grids**[67], **biological networks** such as neural networks among others disciplines[68] .These features are going to be described in the next paragraphs.

Scale free networks[69] have appeared to be accurate descriptions of real networks as Internet or the Web. Using graphs theory it is possible to make a representation of concepts and their relationships among them, the graph is enormous when web is analysed, for instance, or it can be only a partial analysis but the great it is that it doesn't matter, it's scale free!, it means that it has a power-law distribution, so you can find the same properties a different scales: as a fractal.

Following these lines, small world phenomenon is described [70], showing that "***there are many connections between near neighbours and few with far nodes***".

### 4.4.3. WS (Watts-Strogatz) model

They proposed a one-parameter model that interpolates between an ordered finite-dimensional lattice and a random graph. The algorithm behind the model is the following:

- ***Start with order***: Start with a ring lattice with n nodes in which every node is connected to its first K neighbours (K/2 on either side). In order to have a sparse but connected network at all times, consider n >> K >> ln(n) >>1.
- ***Randomize***: Randomly rewire each edge of the lattice with probability p such that self-connections and duplicate edges are excluded. This process introduces pnK / 2 long-range edges, which connect nodes that otherwise, would be part of different neighbourhoods. By varying p one can closely monitor the transition between order (p=0) and chaos (p=1).

This model has its roots in social systems in which most people are friends with their immediate neighbours [71]. However, everybody has one or two friends who are a long way away, people in other countries, old acquaintances, who are represented by the long-range edges obtained by rewiring in the WS model.

Small world systems exhibit properties as Average path length, clustering coefficient, degree distribution and spectral properties [72]. Defining properties, the small world definition will appear.

- ***Path length***: for random graphs, we have very short distances if we analyse path length (d), defined as the average minimum distance between any pair of nodes. It can be shown that in random graphs, $d_{random} \sim \log(n) / \log(z)$. Graphs where $d \sim d_{random}$ are said to be 'small-world' networks that propagate information very efficiently.

- ***Clustering coefficient***: it measures the probability that two neighbours of a given node (z) are also neighbours of one another. For a random graph, CR ˜ z/n and is thus a very small quantity. Also, it is noticed that in real networks, $C \gg C_{random}$. High clustering favours small-worldness.

Therefore, for a given network, if it is observed a small path length, i.e. $d \sim d_{random}$, but a big clustering coefficient, i.e. $C \gg C_{random}$, it can be said that very likely it is a small world.

The main contribution was to show that many real-world networks have properties of random graphs and properties of regular low dimensional lattices. A model explaining the behaviour was missing in the proposed "small-world" model which gives rise to a connectivity distribution function with an exponential form, whereas many real world networks show a highly skewed degree distribution, usually with a power-law tail.

$$P(k) \propto k^{-g}$$

### 4.4.4. BA (Barabási -Albert) model

The BA algorithm has the following[73] graph construction steps:

- **Growth**: starting with a small number ($m_0$) of nodes, at every time step, we add a new node with m ($<m_0$) edges that link the new node to m different nodes already present in the system.
- **Preferential attachment**: when choosing the nodes to which the new node connects, we assume that the probability $\Pi$ that a new node will be connected to node I depends on the degree $k_i$ of node i (see equation 2).

$$\Pi(k_i) = \frac{k_i}{\sum_j k_i} \qquad (1)$$

After *t* time steps this procedure results in a network with $N = t+m_0$ nodes and $E = mt$ edges. And, what is more important, the probability *p(k)* that a vertex has a degree *k* follows a power-law distribution (3), not a Poisson one like in the ER model.

$$P(k) \sim k^{-g} \qquad (2)$$

There are networks that exhibit power-law degree distribution and were addressed by Barabási and Albert and are captured by the BA graph model. They showed that the scale-free nature of real networks is rooted in two generic mechanisms shared by many real networks: growth and preferential attachment.

- ***Growth*:** as an example the Web grows exponentially in time by the addition of new web pages, and the research literature constantly grows by the publication of new papers.
- ***Preferential attachment***: we can consider a web page will more likely include hyperlinks to popular documents with already high degrees, because such highly connected documents are easy to find and thus well known, or a new manuscript is more likely to cite well-known and thus much-cited publications than less-cited and consequently less-known papers.

This kind of distributions are characterised by the $\gamma$ exponent and are called scale-free networks [74], using the connectivity probability in (2) the degree distribution has $\gamma_{BA} = 3$. Moreover, while the goal of ER and WS models is to construct a graph with correct topological features, the modelling of scale–free networks, as this last model is, puts the emphasis on capturing the **network dynamics**.

In order to capture network dynamics is necessary to be able to describe behaviour system, thus it is not possible to think in something periodic because it will not be a real approximation and even worst considering a chaotic approximation, so the behaviour we want to, its just in the middle of these two behaviours, in other words we are looking for something self-organizing [75], it is a system that manages in such way that it is not necessary to control it and reacts to external perceptions without loosing control of itself.

In this way there is an important researching field, that it is called SOC (Self-Organized Criticality). SOC was proposed by Bak et al.[76] as an explanation for the behaviour of a simple cellular-automata model that they developed. In this model, there is a square grid of boxes and at each time-step a particle is dropped into a randomly selected box.

When a box accumulates four particles, the particles are redistributed to the four neighbouring boxes, or in the case of edge boxes, lost from the grid. Redistributions can lead to further instabilities, with “avalanches” of particles lost from the edge of the grid. Due to this “avalanche” behaviour, this was called the “sand pile” model.

The no cumulative frequency-area distribution of model “avalanches” was found to satisfy a power-law (fractal) distribution. This is a stochastic model and also there are two models plus this: a deterministic model, slider-block model [77], and a deterministic chaos model, forest-fire model [78].

For example, SOC models are used to study from human agents' behaviour [79] to packet-based communications networks. In this case the model is implemented as a Monte-Carlo cellular automata simulation due to the way the system evolves through time [80].

Topology is also relevant, i.e. the way components are arranged is a valuable information. If we recapitulate we have systems that exhibit some properties (scale-free, small world) and behaviours (SOC) in a self-organizing way. Therefore, if we look for topologies we are looking for patterns inside our systems, i.e. the way structure appears. At first paths, roads and afterwards highways that will change dynamically and will emerge from SOC models in a natural way.

To have an ordered situation as an ad hoc application can be achieve not designing one by one the components as classical point of view, for instance, a good job has been done in this sense with agents that have artificial pheromones [81] (see section 4.2.6)

To summarize, graphs (ER) and statistical properties conceptualized in the two algorithms (WS and BA) are used because :

Open distributed systems, for instance the Internet and World Wide Web artificial networks, present dynamicity over space and time, an enormous size and unstable relations. Other characteristic features are:

- Hierarchical topology (scale-free): '***there are highly connected and poorly connected nodes'***.
- Fractal behaviour (as the sand pile model) makes the system self-organized (SOC).
- It is relative easy to connect to the farthest network node (WS model).

## 4.4.5. Petri Nets

The basic Petri net is a weighted, bipartite graph that consists of places (drawn as circles) and transitions (drawn as rectangles) connected by directed input and output arcs. The state of a net is denned by a marking that species a distribution of tokens (black dot or a number) over the places of the net. The real-time execution semantics of Petri nets models the production and consumption of resources: a transition is enabled when all its input places are marked such that it can re by moving tokens (the number spaced by the weight of the arc) from input to output places.

A Petri net [82] can be considered a merge of a flow chart and a finite-state machine. The circles of the Petri net, which are called *places*, correspond to the states of the finite-state machine; the bars, called *transitions*, correspond to the events of the flow chart.

The most relevant features of Petri nets for our purposes are their ability to model events and states in a distributed system and cleanly capture sequentially, concurrency and event-based asynchronous control. Our extensions to the basic Petri net formalism include typed arcs, hierarchical control, durative transitions, parameterisation, typed (individual) tokens and stochastic.

Processes are represented by a-cyclic graphs because no path can ever loop back to an earlier point in time. Procedures, however, often contain loops. Those loops do not cycle back to an earlier time, but to another instance of the same type of state or event. In the diagrams of Figure 20, the state types are labelled *p*, *q*, *r*, *s*, and *t*; the events types are labelled *a*, *b*, *d*, *e*, and *f*. Any of the three notations in Figure 20 can be used to specify an infinite family of discrete processes. Next, there are the sequences of state and event types for the first three processes in the family:

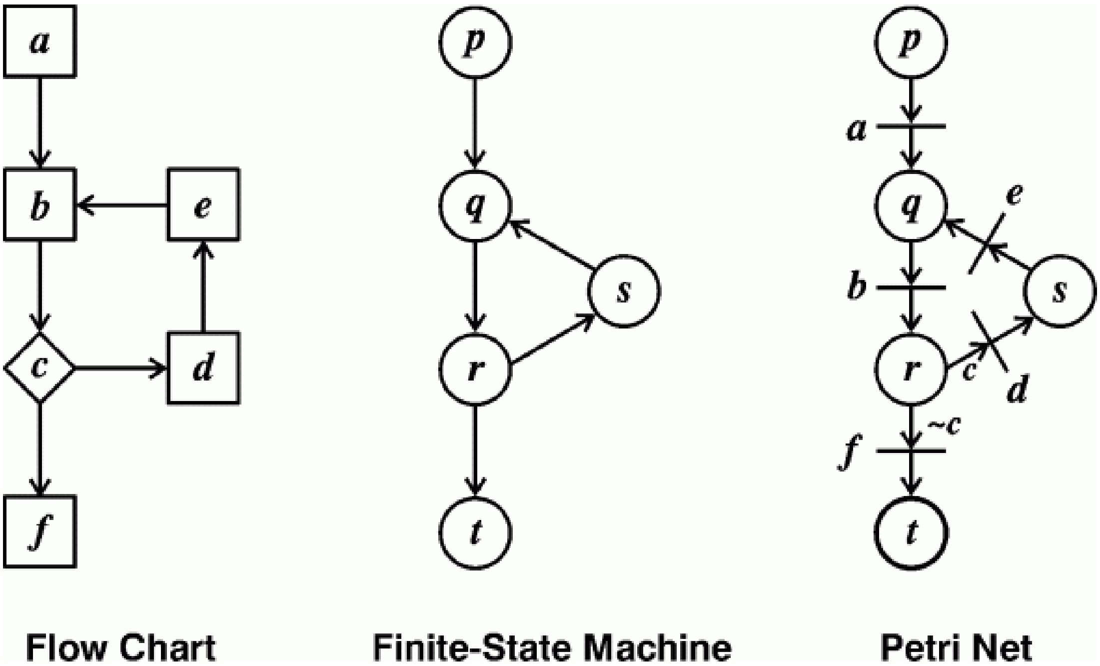



**Figure 20** Three graphical notations for specifying procedures

For every place *x* in *P*, there exists exactly one state type *p* in the generalization hierarchy specified in Axiom 3.2. The predicate *p* is called the *type* of the place *x*.
*Transition types*. For every transition *y* in *T*, there exists exactly one event type *q* in the generalization hierarchy specified in Axiom 3.2. The predicate *q* is called the *type* of the transition *y*.
*Input compatibility*. For every transition *y* in *T*, each precondition of the type *q* of *y* must be a super-type of the corresponding place type.
*Output compatibility*. For every transition *y* in *T*, each post-condition of the type *q* of *y* must be a subtype of the corresponding place type.

Procedures can be specified in many different notations, but Petri nets are especially convenient because their graph structure has a direct mapping to discrete processes represented by directed a-cyclic graphs. The fundamental structure of processes and procedures, however, is independent of the details of the notation. As an example, *linear logic* is a notation that is formally equivalent to Petri nets although it has a very different appearance. The proof procedures of linear logic mimic the firing of Petri net transitions.

# 4.5. Knowledge representation: Semantic Web

Systems as agents or even architectures can be expressed as graphs (see section 4.4.) and this section will show that knowledge representation too.

Graphs properties (see sections 4.4.3 to 4.4.5) help when many nodes are involved and there is information about the main nodes relations.

When language is involved, the relations among concepts are known and they are crucial which becomes a high complex framework, however all human languages present the same distribution law of words frequency called ***Zip's law*** and it can be found that ***simultaneous minimization in the effort of both hearer and speaker is formalized with a simple optimization process operating on a binary matrix of signal-object associations. Zipf's law is found in the transition between referentially useless systems and indexical reference systems*** [83]

Thus, there are evidences for generic laws in language. However, the point will be not to work only syntactically whether semantically. It means, that concepts are related by their meanings: this was the Semantic Web origin.

*Section 4.5.1 describes semantic web languages and explains how the ontology concept was born and the contribution to syntactic approximation, section 4.5.2 goes deeper into the ontology concept and section 4.5.3 explains the ontologies goal inside the semantic web. The aim of this part is to understand if it is possible to know about what the others know* because this was expressed as a first challenge).

## 4.5.1. Languages

Semantics is a key issue when negotiation is performed. The meaning has to be clarified if the negotiation parts want to make an agreement. There are many situations where the same term, when it is syntactically parsed, can be interpreted in different ways. This is the reason why it is not only necessary to parse it but also to analyse it semantically (see Figure 21).

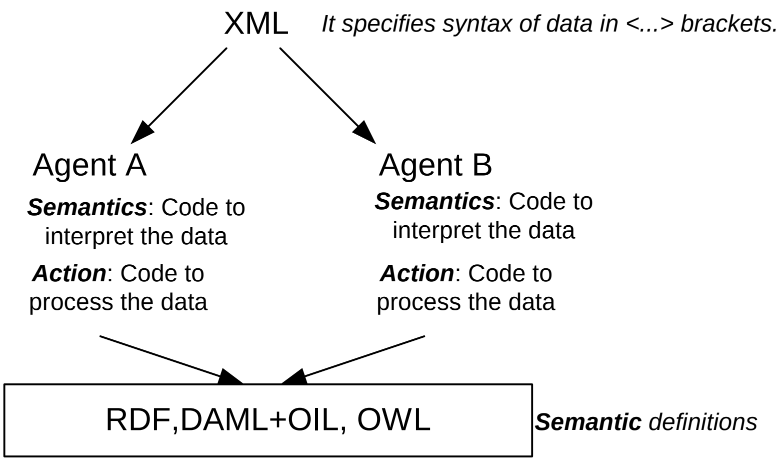



**Figure 21 Syntax vs Semantics.**

Several mark-up languages are designed to represent knowledge. For instance, web pages are based on HTML and its evolutions as DHTML. This was a revolution that took its greatest expression in the web. However, something relevant was still missing. We refer to semantics, because the amount of information that can be found in the web needed a way to capture its associated meaning. A good metadata representation tool was necessary.
XML (eXtensive Markup Language [84]) was born to accomplish this, with XML schemas aid. Some initiatives have arisen in enterprise world as ebXML (Electronic Business using eXtensible Markup Language [85]). Nowadays it is the most widespread metadata representation used in enterprise software.

However, RDF (Resource Description Language [118]) represents an evolution that uses XML syntax but it is not constrained by that. RDF is a foundation for metadata modelling and it provides interoperability between applications that exchange machine-understandable information on the Web. RDF emphasizes facilities to enable automated processing of Web resources.

RDF constitutes the basis; on top of it, other more sophisticated tools are being developed. OWL (Ontology Web language [86]) is the latest Semantic Web tool for ontology modelling. It continues the work started with DAML+OIL (Darpa Agent Markup Language+Ontology Inference Layer). Ontologies are a knowledge representation formalism taken from the philosophy tradition.

It is crucial to understand the differences between XML and RDF. First, how is the file interpreted when it is being processed? Both formats use API's (Application Programming Interfaces), e.g. Xerces [87] for XML and Jena for RDF.

Both are defined by schemas, XMLSchema and RDFSchema respectively. Schemas define vocabulary and relationships constraints over the XML tree and the RDF graph as it is going to be shown.

RDF is an abstraction layer over XML, it is only used for RDF graph serialization. Therefore, RDF/XML and XML files can seem similar. However, their interpretations differ because XML is interpreted as a ***tree*** while RDF models a ***directed graph***. Another great difference is order. XML enforces a serialisation order while RDF does not. For instance, in the description of place, it does not matter the order but in a XML file we have to put it correctly, if not, the application will not success interpreting the file. However, RDF allows specifying if order is important or not using list or sets, for example. Finally, RDF interpretation can be partial, but it is not the case of XML, i.e. all the document must be understood. The consequence is that their structure is known 'a priori'. On the other hand, with RDF it is only enough to know some elements and to follow their relationships.

DAML-OIL and OWL allow providing more detailed constrains over the RDF graph. For instance, it can specify elements cardinality, it means that it is possible in an application to express how many days are needed. Even, classify elements by their properties, i.e. define implicit classes. For example, a task that has to be performed during a period as *every week* or *every day*, this information can be formalized inside *every time* things because they have a characteristic period and it is done without 'a priori' information. It is performed automatically as they were rules.

DAML-S (Darpa Agent Markup Language – Services, Ontology[88]), and the newer but similar OWL-S, seems to be in a near future a veritable standard in services world. DAML-S supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. DAML-S markup of Web services will facilitate the automation of Web service tasks including automated Web service discovery, execution, interoperation, composition and execution monitoring. Following the layered approach to markup language development, the current version of DAML-S builds on top of DAML+OIL, in the case of OWL-S on top of OWL.

In addition, initiatives as UDDI (Universal Description, Discovery Integration[89]) which claims to be a point of reference from industry with WDSL (Web Services Description Language[90]) as a web service definition language. Thus, WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

Both initiatives, DAML-S and UDDI will surely coexist and they will complement each other because the first is focused in knowledge management while UDDI is understood as a widespread services discovery all over the world.

OWL builds on top and extends RDF Schema. At the basis, the purpose of OWL is identical to RDF Schemas *(to provide an XML vocabulary to define classes, their properties and their relations among classes)*.

## 4.5.2. Ontologies

Generally, an ontology can be said that it is a formal explicit description of concepts in a domain of discourse (***classes*** (sometimes called "*concepts"*)), properties of each concept describing various features and attributes of the concept (***slots*** (sometimes called "*roles"* or "*properties"*)), and restrictions on slots (***facets*** (sometimes called "*role restrictions"*)). An ontology together with a set of individual ***instances of classes*** constitutes a ***knowledge base.*** In reality, there is a fine line where the ontology ends and the knowledge base begins.

Classes are the focus of most ontologies. Classes describe concepts in the domain. For example, a class of licenses represents all licenses. Specific classes are instances of this class. A class can have ***subclasses*** that represent concepts that are more specific than the ***superclass***.

In practical terms, developing an ontology includes:

- Defining classes in the ontology.
- Arranging the classes in a taxonomic (subclass-superclass) hierarchy.
- Defining slots and describing allowed values for these slots.
- Filling in the values for slots for instances.

In *computer science*, an ontology is the attempt to formulate an exhaustive and rigorous conceptual schema within a given domain, a typically hierarchical data structure containing all the relevant entities and their relationships and rules (theorems, regulations) within that domain.

It is true that everybody can design its own ontology, however if knowledge has to be connected to access it, it is worth to relate the designed ontology to others initiatives. Following these lines, there are some that can suggest how to connect the new ontology.

For instance, the Suggested Upper Merged Ontology (SUMO) is an upper ontology (see Table 6), 'upper' understood as high abstraction level, and intended as a foundation ontology for a variety of computer information processing systems. It was developed by Teknowledge Corporation and it is a candidate for the "Standard Upper Ontology" that IEEE working group 1600.1 is working on. It can be downloaded and used freely. SUMO was first released in December 2000.

WordNet [1] is a semantic lexicon for the English language (it has been opened to other languages too). It groups English words into sets of synonyms called ***synsets***, provides short definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of *dictionary and thesaurus* that is more intuitively usable, and to support *automatic text analysis and artificial intelligence* applications. The database and software tools have been released under a BSD style licence (open source) and can be downloaded and used freely. The database can also be browsed online. ***Synsets*** for nouns and verbs to SUMO classes has also been defined.

Table 6 summarizes the ontologies classification into upper, mid-level and lower-level ontologies, it means from highest abstract level to the lowest.

**Table 6. Ontologies classification**

| | |
|---|---|
| Upper ontologies | It defines very broad, universal Classes and properties<br>Example: Cyc Upper ontology [91] |
| Mid-level ontology | An upper ontology for a specific domain |
| Lower-level ontology | It is an ontology for a specific domain, with specific Classes and properties |

### 4.5.3. DOLCE, D&S and Web services: DAML-S, OWL-S

This section explores ontologies goals and how they can be specified to provide connectivity to other ontologies. An example of how ontologies are growing for providing services inside the Semantic Web will be presented.

Ontologies are used for knowledge representation as it has been seen in the previous section. Moreover, they can help modelling new knowledge. DOLCE and D&S have been designed for this task, see more ontologies in Figure 25. Thus, different layers can be distinguished, see Figure 22.

At the top, DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) belongs to the WonderWeb Project Foundational Ontology Library (WFOL). It is designed to be minimal in that it includes only the reusable and widely applicable upper-level categories, rigorous in terms of axiomatization and extensively researched and documented [92]. It has four categories: ***endurant*** (including object and substance-like entities), ***perdurant*** (event- and state- like entities), ***quality*** (individual attributes), and ***abstracts*** (mainly conceptual “regions” for structuring attributes). DOLCE is based on D&S (See Figure 24).

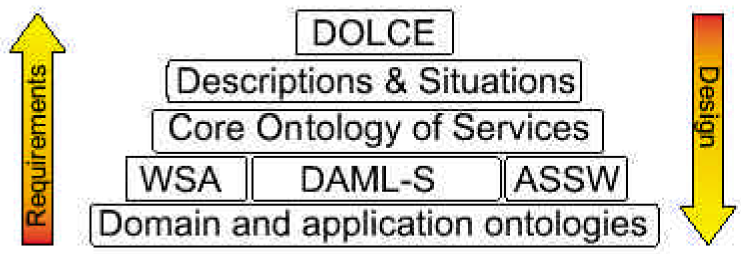



**Figure 22 Ontology stacking in WonderWeb** [93]

The Descriptions and Situations ontology (D&S), see Figure 22, is an attempt to define a theory that supports a first-order manipulation of theories and models, independently from the particular foundational ontology it is plugged in.

D&S commits only to a very ancient ontological distinction between flux, or an unstructured world or context, and logos, or an intentionality. D&S is neutral with respect to a realism issues. Hence, a flux can have as many inherent structures (parts, boundaries, qualities, etc.) as one might want to believe in or might claim to have discovered, but without a logos, a flux would have no description of that structure.

When logos is applied to the description of the flux, some structure emerges. The emerging structure is not necessarily equivalent to the actual structure.

D&S implements reification rules for any logos-like theory (either formal or capable of being at least partly formalised) called ***description***, a basic framework for any logical structure (either formal or capable of being at least partly formalised) called ***situation*** and for their elements (see Figure 23).

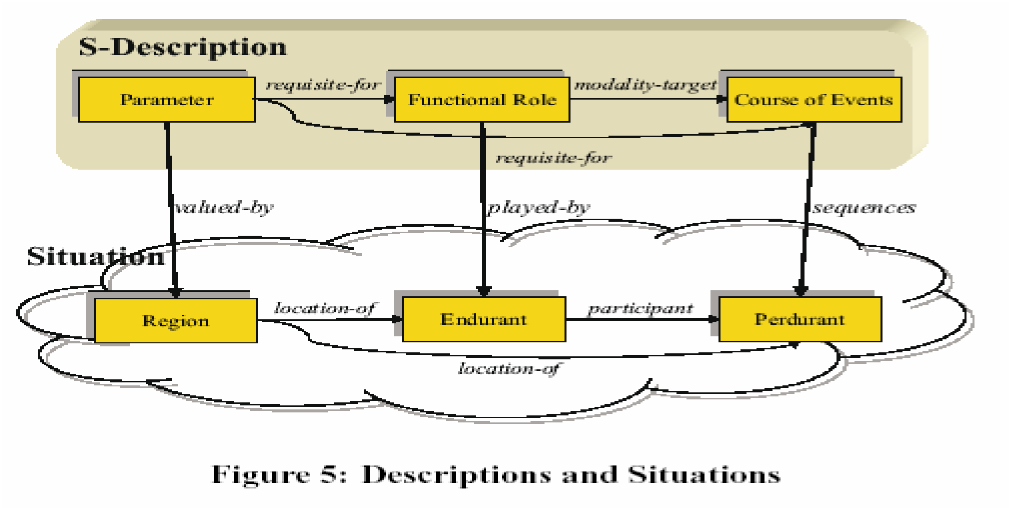



**Figure 23 Descriptions and situations**

Flux-like structures are not reified in D&S, but they result to be the structures that include all the (ground) logical dependencies of the components of a situation S classified within an ontology O, plus any additional elements that could be part of the ground context of S according to some encoder of O, but that are not inside O. A flux-like structure is called a ***state of affairs*** (SOA) in D&S.

D&S results to be a theory of ontological contexts because it is capable to describe various notions of context (physical and non-physical situations, topics, provisions, plans, assessments, beliefs, etc.) as first-order entities. Figure 24 shows the D&S ontology and the connections to DOLCE in UML.

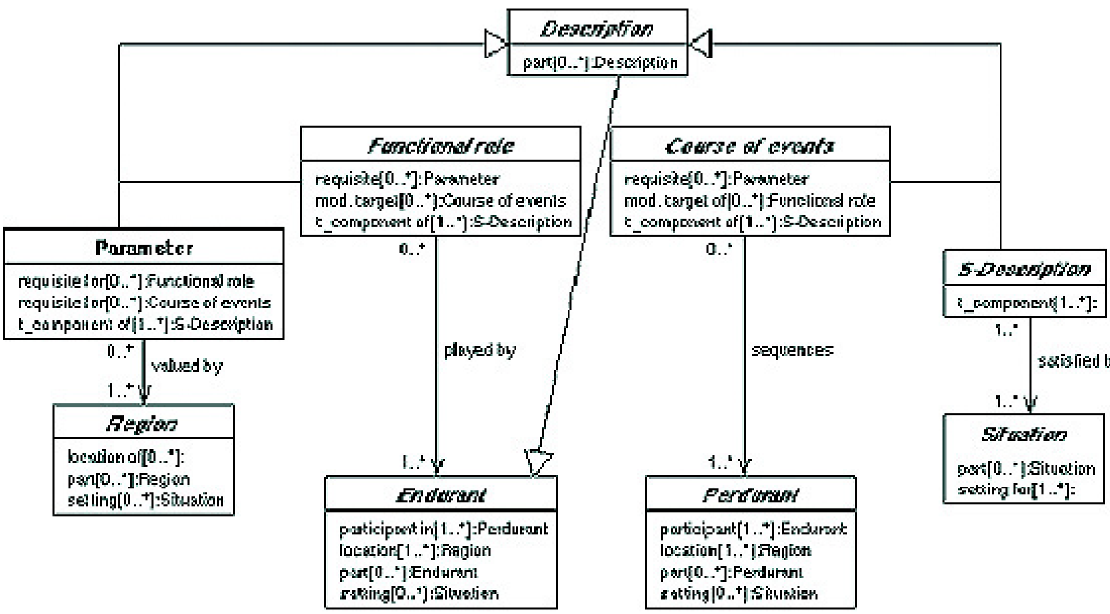



**Figure 24 UML overview of the D&S ontology of descriptions**

Following these lines (see Figure 25), eight popular ontologies were surveyed[94] for their expressiveness considering only the Part-of-Speech (POS) noun and verb. For nouns, the hypernym and meronym relations were identified while, for verbs, the hypernym and domain relations were considered. A noun is said to be in the domain of a verb if the verb describes either a capability of the noun or a transformation on the noun. In DOLCE and OpenCyc, the noun form often doubles as a predicate for use in expert system reasoning applications. In cognitive linguistics FrameNet (see section 7.5) is analysed and it will be discussed in conjunction to DOLCE and D&S because it is the only that takes care of the domain of the verb.

| *Ontology* | *Noun* | | *Verb* | |
|---|---|---|---|---|
| | *kind-of* | *part-of* | *kind-of* | *domain* |
| DOLCE | ✓ | ✓ | | |
| GUM | ✓ | | | |
| EDR | ✓ | | ✓ | |
| FrameNet | | | ✓ | ✓ |
| OpenCyc | ✓ | ✓ | | |
| SENSUS | ✓ | ✓ | ✓ | |
| SUMO | ✓ | ✓ | | |
| WordNet | ✓ | ✓ | ✓ | |

**Figure 25 Survey of Upper Ontologies[94]**

Following the layers shown in Figure 22, Web Services ontologies are the next.

Web Services provide not merely static information but allow one to effect some action or change in the world, such as the sale of a product or the control of a physical device. The semantic web should enable users to locate, select, employ, compose, and monitor Web-based services automatically.

DAML-S is an ontology of web services and it provides a new language as it has been described [88]. There are three main parts: ***service profile*** for advertising and discovering services; ***process model***, which gives a detailed description of a service's operation; and ***grounding***, which provides details on how to interoperate with a service, via messages.

From OWL-S [86] emerges a more effective way of describing services. It will be easier to map concepts from different ontologies. Every concept will have not only a meaning, but also a well-defined context and relationships with the other concepts. This makes the difference with nowadays technologies, but it can be reduced to a graph with many connections among the nodes.

## 4.6. A new approach to conceptualise the concept of ‘Negotiation’

This section discusses two issues that are inherent problems in negotiation processes: ***Causal model*** and ***Domain knowledge***. A brief explanation is provided in the following paragraphs.

### 4.6.1. Causal model

Trying going deeper in the definition of what it is a negotiation process, the idea of causality was an obstacle that even in our days has not been solved yet and it is the origin of reasoning inside our culture. Inference rules are constrained by causality. In the ‘process ontology’ (remember Figure 1) a directed dependence of time was explained. This section outlines what is inherent to it and to the consequent model: classical CAUSAL model. A discussion and answers to the open questions will be solved in section 7.

David Hume (1711-1776) was the first philosopher to make a sharp distinction between analytical and empirical claims. He thought that the former are product of thoughts, the latter matter of fact.

Afterwards, he classified causal claims as empirical, rather than analytical. He identified the source of all empirical claims with human experience, namely sensory input. He established the basis for the movement called ***empiricism***.

In the XVIII century, another paradigm and questions about the mechanism of how the things happen were discussed. The essential mechanism of causation is elevated to be the main issue so as the questions related to it and its relation to empirical basis:

- What empirical evidence legitimizes a cause-effect connection?
- What inferences can be drawn from causal information?

Currently, the idea of building machines that make sense of what goes on in their environment has appeared, so they can realize when things do not turn out to be exactly as expected.

Moreover, universal understanding and how to teach what we know about the world come to be important questions, because the way we communicate about the world is through this strange language called causation.

This pressure to build machines that both learn about and reason with cause and effect, something that David Hume did not experience, now casts new light on the riddles of causation, coloured with engineering flavour:

- How should an artificial agent acquire causal information from the environment?
- How should an artificial agent process causal information received from its creator-programmer?

An approximation to these questions has the following premises in ***classical causality*** approach:

- The central theme is to view causality a **computational scheme** devised to facilitate prediction of the effects of actions.

- It seems to be best understood if we view actions as **external entities**, originating from outside our theory, not as a model of behaviour within the theory.

The assumption that the world is organized in the form of stable mechanisms, or physical laws, which are sufficient for determining all events that are of interest to the modeller, has the origin in the XVIII century. Mechanisms are autonomous and compare to ***mechanical linkages in a machine***, or logic gates in electronic circuits.

In these systems, element can be interchanged without affecting the others. This is the core of the autonomy concept.

Thus, causality tells us which mechanism is to be surgically modified by any given action.

These principles can be encapsulated neatly and organized in a mathematical object called a ***causal model***. They are:

- Causation as encoding of behaviour under interventions.
- Interventions in the model as ***surgeries*** on mechanisms.
- Mechanisms as stable functional relationships (***equations*** + graphs).

The purpose of a model is to assign **truth values** to sentences in a **given language**. If models in standard logic assign truth values to logical formulas, causal models embrace a wider class of sentences, including those that we normally classify as causal. The kinds of sentences that are analysed in this system are:

- **Actions**: B will be true (B) if we do A.
- **Counterfactuals**: B would be different (not B) if A were true.
- **Explanation**: B because of A.

Action is understood as decision, counterfactuals as to infer the opposite fact and explanation is the connexion to the origin of from A.

The difference between action and counterfactuals is merely that the class between the antecedent and the current state of affairs is **explicit**.

For instance, if a circuit diagram is studied it is possible to distinguish some interesting points to notice in this example: It qualifies as a causal model because it contains the information to confirm or refute all action, counterfactual and explanatory sentences concerned with the operation of the circuit.

Anyone can figure out what the output would be like if a gate sets Y to zero, or if there is a change of the OR gate to a NOR gate or if we perform any of the billions combinations of such actions.

A logical function (Boolean input-output relation) would not be sufficient for answering such queries. These actions were not specified in advance, they do not have special names and they do not show up in the diagram.

In fact, the great majority of the action queries that this circuit can answer have never been considered by the designer of this circuit.

The circuit encodes this extra information through two encoding tricks:

- There is an ***implicit connection*** between symbolic and physical mechanisms. The symbolic units correspond to stable physical mechanisms (i.e., the logical gates).

- The variable ***behaviour*** is ***known***: each variable has precisely one mechanism that determines its value.

Now that we are on familiar grounds, let us observe more closely the way a causal model encodes the information needed for answering causal queries.

An example that it is used faces the fact that sometimes the logical time events don't happen as it is expected. For instance, a firing squad with two shooters are waiting for an order to shoot. However, one of them shoots and there was no order. The result is a person dies.

It can be showed that in the case of firing squad, the question that arises is that sometimes ***some object does not behaviour in the way that it is expected to do***.

Summarizing, there are some open questions that come over after these paragraphs:

- Is it reasonable to expect the model behaviour to be the same for everybody in '*time development'* for example? (see section 7.2).
- The truth values are always 'true' and if they can change, it is possible to say how? (see section 7.2).
- Given a language, how can the mappings from it to another knowledge representation be done? Is it possible to understand symbolic knowledge representation from scratch? It could be possible to relate to stable physical mechanisms?(see section 7.1)

## 4.6.2. Domain knowledge

These paragraphs describe the efforts to explain the way human beings process reality with their minds, a way to come clear the way reality is conceptualised from physical to abstract domain. The challenge is to find a way to connect two domains in an appropriated format for humans and machines: just to connect symbolic to physical domain.

A little review of history is provided from a discipline that has arisen at the end of the 20th century at the intersection of a number of existing disciplines, including *psychology, linguistics, computer science, philosophy, and physiology*. It is known as Cognitive Science [95]. The shared interest that has produced this coalition is to understand the nature of the mind. This quest is an old one, dating back to antiquity in the case of philosophy, but new ideas are emerging from the fresh approach of Cognitive Science. Previously, each discipline sought to understand the mind from its own perspective. They benefited little from progress in other fields because they employed different methods. With the advent of Cognitive Science, however, common interests and theoretical ideas have overcome methodological differences, and interdisciplinary interaction has become the hallmark of this field.

The intellectual developments that paved the way for Cognitive Science began in the 1940s and 1950s. The most significant events were outgrowths of the conceptual invention (*via mathematical description*) of computer machines by the British mathematician, Alan Turing, in 1950. The first digital computers -- also known as "***universal Turing machines***" -- were built shortly thereafter. Turing and others soon realized that these computers could be programmed to perform complex "intellectual" tasks previously performed only by humans, tasks such as playing chess, proving mathematical theorems, and understanding language.

Pioneers in this new field of computer science began to make progress toward these goals by programming computers to simulate mental processes. For example, Allen Newell and Herbert Simon's famous program, the ***General Problem Solver*** (GPS), was able to play chess and to prove theorems remarkably well for a program written in the early 1960s. Understanding natural language has proven to be a more difficult task, but progress is also being made in that domain. Surprisingly, the supposedly "simple" process of perceiving the visual world, which is not a uniquely human capability at all, has turned out to be among the hardest capabilities to simulate in computers. Current research in computer science is aimed at further progress in all of these domains.

In response to AI (Artificial Intelligence), philosophers began to formulate a new approach to the age-old problem of the relation between mind and brain. Their idea was to explore a particular **analogy** suggested by the work in artificial intelligence: that **mind is to brain** as **program is to computer**. Thus was born the notion that minds are essentially "program like" entities that "run" on brains instead of computers. This proposal spawned a major philosophical debate about the nature of mental events. It centred on new issues, such as whether a computer could really ***"understand***" language or really ***"have"*** conscious experiences as the result of running the right program, as some believed and others disputed. This debate has not finished yet.

The closely related idea that mental activity could be described as information processing emerged in psychology at about the same time. This was partly due to the direct influence of work in artificial intelligence, especially via Newell and Simon's proposal that a **computer program** was a *psychological theory of how people performed the task it simulated*. Other psychologists were also exploring information processing as a way to break the grip of Behaviourism on psychology. The behaviourists, with B. F. Skinner as leader, dominated psychology for decades. They claimed that the only proper object of study for scientific psychology was *behaviour*. Thus, they ruled out any reference to internal mental states, as a state machine in computer science.
Then, the information processing approach stated that mental events could be described as a structure of operations for constructing and transforming internal representations and gave a principled way in which internal events could be specified rigorously and tested scientifically. Because of the paradigm shift, often referred to as the "cognitive revolution", information processing has now replaced Behaviourism as the dominant force in psychology. The key was children learning. Children learn language very quickly in the first years although they have not been exposed to many stimuli. This ‘stimulus poorness’ was the reason of the paradigm change.

Related ideas were also revolutionizing the linguistics field at about the same time [96]. For the first time, it is proposed a ***transformational approach to grammar*** in which the ***surface structure*** of sentences was derived from an underlying ***deep structure*** of primitive linguistic units by a series of rules or transformations. The formal structure of these transformations was closely related to ***finite state automata*** in computational theory and to the information processing approach in psychology.

During the same period, in the 60’s, new techniques were being pioneered in neurophysiology. These techniques allowed scientists to begin to understand the workings of the brain as an information-processing device. For example, new methods of staining individual neurons showed how they projected from one area of the brain to another, allowing anatomists to map out the ***large-scale wiring diagram*** of certain brain regions. Even more importantly, neurophysiologists developed methods for recording the activity of individual brain cells. As an example, this technique allowed Nobel laureates David Hubel and Thorsten Wiesel to determine the patterns of retinal stimulation that caused cells in visual cortex to fire.

More recent advances in physiology have come from various brain scanning and imaging techniques, such as Computer-Assisted Tomography (CAT), Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET) methods. For example, scientists can now identify specific regions of brain damage in neurological patients so that ***symptoms*** can be correlated with ***anatomical location***. Using these methods in conjunction with those of cognitive psychology, cognitive neuroscientists are beginning to map out the function of major areas of the human brain. Even alternative medicine developed in other cultures, such as acupuncture relates symptoms to a distributed anatomical location.

From the efforts of trying to understand the way language is learned and understood, NTL (Neural Theory of Language [97]) was born. This theory attempts to explain how many brain functions (including **emotion and social cognition**) work together to understand and learn language. The NTL assumption is that people understand narratives by subconsciously imaging (or simulating) the situation being described. There is both linguistic evidence (from classifier languages) and imaging data [98] supporting the idea that the **meaning of a noun** depends on the uses of the **underlying thing**.

Animals neurons have been correlated, for instance, mirror neurons in monkeys and their homologues in people [99] suggest an overlapping substrate for the execution of actions and for the perception of the same action. Moreover, language and physiology in human beings has been correlated [100]

In summary, over two decades of work in neuroscience to suggests that cortical pre-motor areas contain neurons that respond to multi-modal (visual, somato-sensory, and auditory) sensory stimuli.

Cognitive linguistics is based in "*Image Schemas*", which are regularities in our perceptual, motor and cognitive systems. They structure our experiences and interactions with the world and may be grounded in a specific cognitive system, but are not situation-specific in their application. This means that they can apply to many domains of experience. They can be conceptualised as patterns.

Image-schemas, that is, universal primitives of spatial relations, such as containment, contact, centre-periphery, paths, and so on. There are models many of these in terms of structured connectionist neural networks using models of such visual cortex structures as topographic maps of the visual field, orientation-sensitive cell assemblies, and so on[101]. Image-Schemas are conceptual or perceptual and they represent a link between language and spatial perception. Mental imagery experiments, using fmRI techniques have shown that much of the visual system, down to the primary visual cortex, is active when mental imagery is created without visual input. The brain's visual system is also active when a person dreams[102]. Moreover, congenitally blind people, most of whom have the visual system of the brain intact, can perform visual imagery experiments perfectly well, with basically the same results as sighted subjects, but a bit slower [103]. In short, one should not think of the visual system as operating purely on visual input. Thus, it makes neurological sense that structures in the visual system can be used for conceptual purposes, even by the congenitally blind people.

Moreover, the visual system is linked to the motor system via the prefrontal cortex. Via this connection, Image-Schemas are related to body parts [104]. A remarkable feature is that even the motor control is presented as if it was a states machine. However, these motor systems do not present ***the 'stop'***

***machine problem*** associated to Turing machines [105]. They have embedded time out mechanisms.

Static Image-Schemas could be divided into three major groups:

- Above-schema which is ***orientational***.
- Contact-schema which is a ***topological schema***.
- Support-schema which is related to ***force-dynamics***.

*"Spatial schemas"* are image schemas where there are two roles sharing a location (at):

- A *"Trajector"* (TR) which is an object being located.
- A *"Landmark Schema"* (LM) that represents the reference object.

Sometimes is possible to find that there is an ***asymmetry*** from the spatial point of view, if the TR is *'on'* LM, it means that LM is *'under'* TR.

The basic question about the role of Image Schemas is if the spatial representations associated with certain verbs are merely vestigial and only accessible meta-cognitively, or perhaps they are automatically activated by the process of comprehending those verbs.

It seems there is a related connection between **language** and **Spatial Schemas**, here there are some examples connected by ***metaphors***:

- ***Social relations***: people say that they look up to some people, but look down on others because those we deem worthy of respect are somehow "above" us, and those we deem unworthy are somehow "beneath" us (it is an effort going against gravity force, someone who climbs is respected because of his/her effort).
- ***Economy***: economy behaviour is plenty of these metaphors, the economy increases or goes down.
- ***Negotiation***: an agreement was "reached".
- ***Discussion issues***: arguments can go "back and forth," and hopes can get "too high".

It is important to differentiate boundaries and bounded region, see Figure 26 and Figure 27. Boundary Schema is the first operation when an object wants to be distinguished. Three roles can be distinguished in both figures, the difference is the boundary. It is closed in the bounded region while it is not in the Boundary Schema.

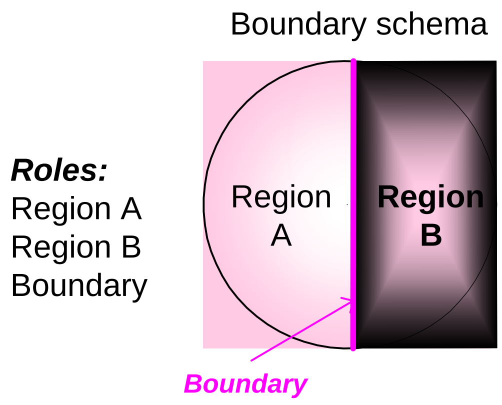



**Figure 26  Boundary Schema.**

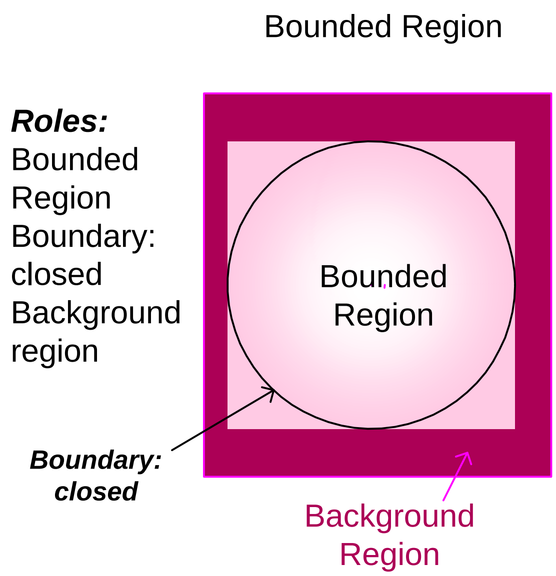



**Figure 27  Bounded region.**

Topological relations as separation, contact, coincidence, overlap, inclusion or encircle/surround can be studied using bounded regions.

However the controversial situation appears when a context, in cognitive linguistics called 'frame' because how the axes which people is related are going to be taken. It is not an easy question. There are as ***minimum three*** prototypical frames of reference[106] in languages:

- Intrinsic
- Relative
- Absolute

Figure 28 shows an intrinsic frame of reference, the object itself determines the orientation: *Front, Left, Right, Back* of the object respectively. When other object is next to reference object, then the four orientations can change depending on the relative position of them.

Everyone knows an absolute frame of reference, for instance our Earth magnetic field determines where there are four points of reference: *North, East, South* and *West*, see Figure 29. Some Australian aborigines only use this system. They are supposed to do it in the way because their lives are always related to nature and with a wide and open environment. Astronomers use fixed stars as ‘an absolute reference frame” because the distance between the earth and they is so huge that the relative movement can be depreciated.

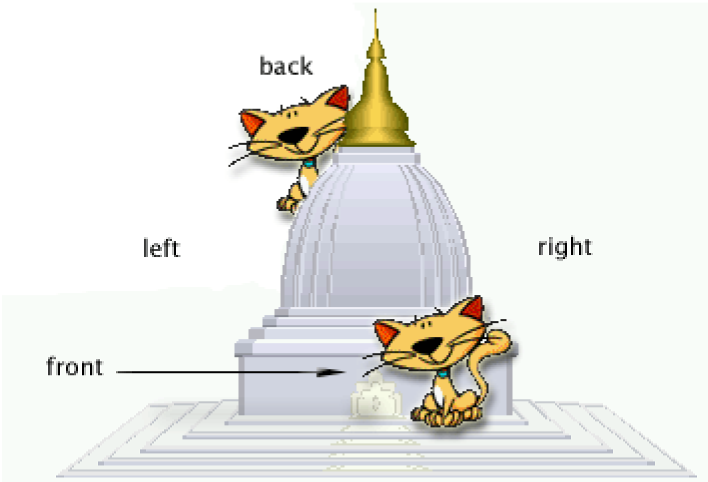



**Figure 28 An object determines an *intrinsic* frame of reference**

**Figure 29 Earth determines our absolute Frame of Reference**

Depending on how the objects are orientated ones respect to others, it is time to explore another schema called “*Container-Schema*”. This structure forms a gelstat, in the sense that the parts make no sense without the whole. This structure is topological in the sense that the boundary can be made larger, smaller, or distorted and the boundary of a Container Schema remains. It is composed by (see Figure 30, and remember Figure 27):

- An interior
- A boundary
- An exterior

Container schema

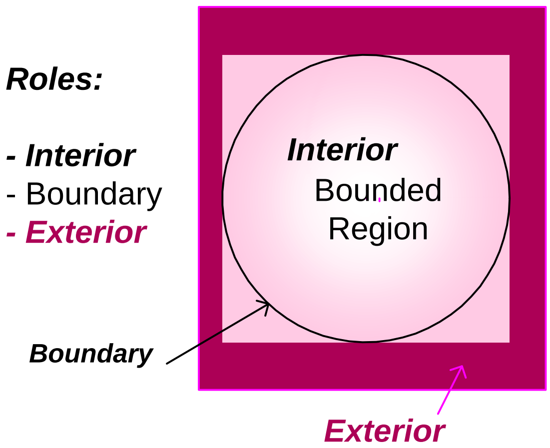



**Figure 30 Container schema with the roles associated to it.**

In Figure 31, two states, “*out”* and “*in”*, come from combining *Container-Schema* with *Trajector* and *Landmark*. The latter two represent objects that are related to each other.

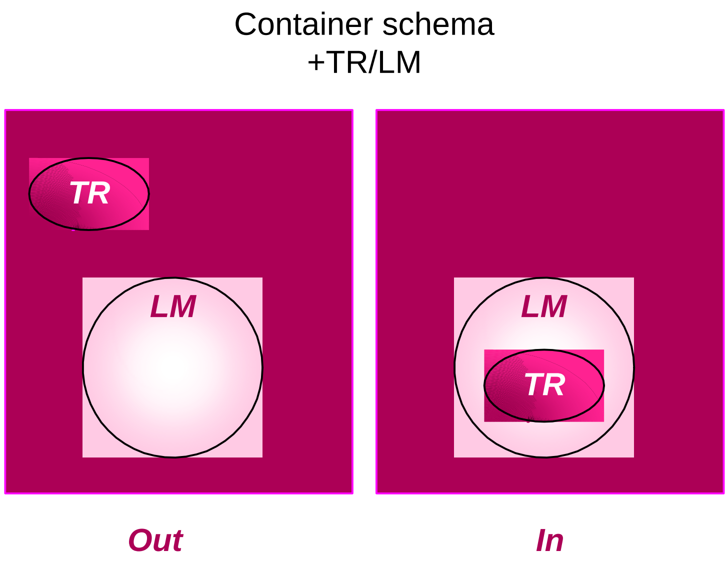



**Figure 31 "In" and "out" states arise from combining TR/LM with *"Container-Schema"***

Container Schema can be elaborated further. Its complexity grows with more roles and specifications, for instance boundary properties: ***Strength or Porosity***

Moreover, from an anomaly in the boundary called "portal", the Source-Path-Goal-Schema arises.

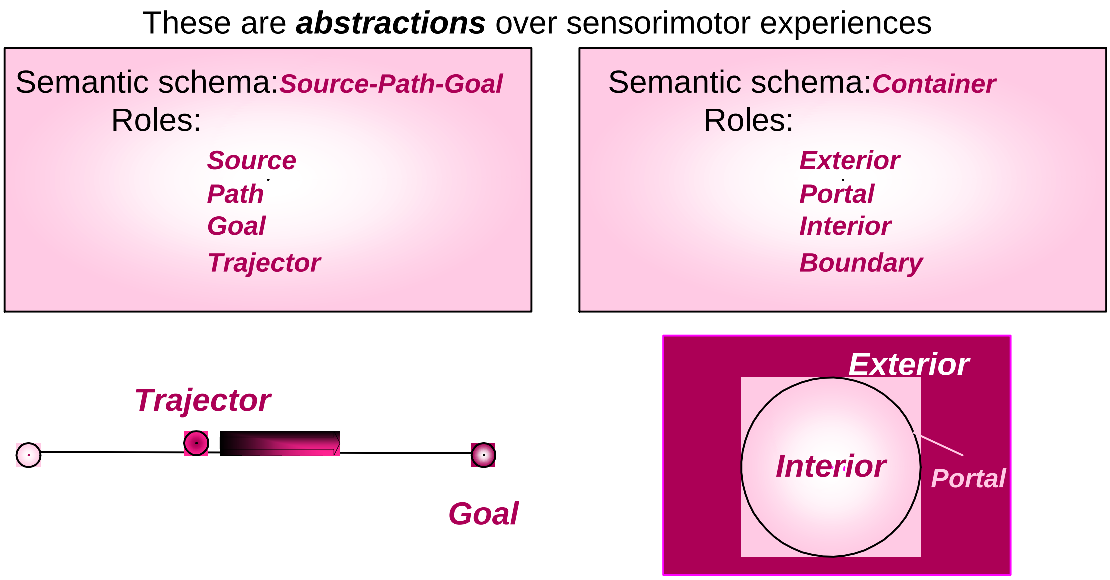



**Figure 32 Source-Path-Goal-Schema connected to Container-Schema**

If the following sentences a comparison between language symbols, containers and the *source path goal schema* (SPGS) will be made:

"She drove from the store to the gas station."

SPGS-Trajector (TR) is the actor she
SPGS-Source is the store
SPGS-Goal is the gas station

“She ran into the room.”

SPGS-Trajector (TR) is the actor she
SPGS-Source is the Container.Exterior (out of the room)
***SPGS-Path*** is the ***Container.Portal*** (in the boundary of the room)
SPGS-Goal is the Container.Interior (in the room)

# 5. Negotiation in a SWA: Agents aspects

In order to work in the Semantic Web, an architecture has to be provided. There are several components inside it. This section concentrates on the most active and nearer to the customer: agents (more information in section 9). These programs represent users inside the system and they negotiate in the name of final users.

*Moreover, this part addresses the question of what it can be understood as '**rational**' and if the agent behaviour can be predictable*). The key issue in this part is using Semantic Web resources plus JESS because it allows a dynamic negotiation that can be modified every time as well as behaviour agents that it is called ***policy***.

The message content as well as formalization knowledge is explained in section 6.

Then, a Semantic Web Architecture is going to be developed, see Figure 33. One of the components is a Multi-agent System that it is the *active part* that negotiates; in this context: *"negotiation is carried out on a **contract** to obtain **common resources** and on the **request** of an agent called **the initiator**. It brings together an initiator and a set of agents to whom the contract is proposed by the initiator agent, these agents are called the participants, and runs until an **agreement** satisfying a percentage of participants is reached. Participants equally try to obtain the best possible solution for themselves while giving a **minimum set of information** to the others."*[107]. It will be applied inside IPR (Intellectual Property Rights) business model.

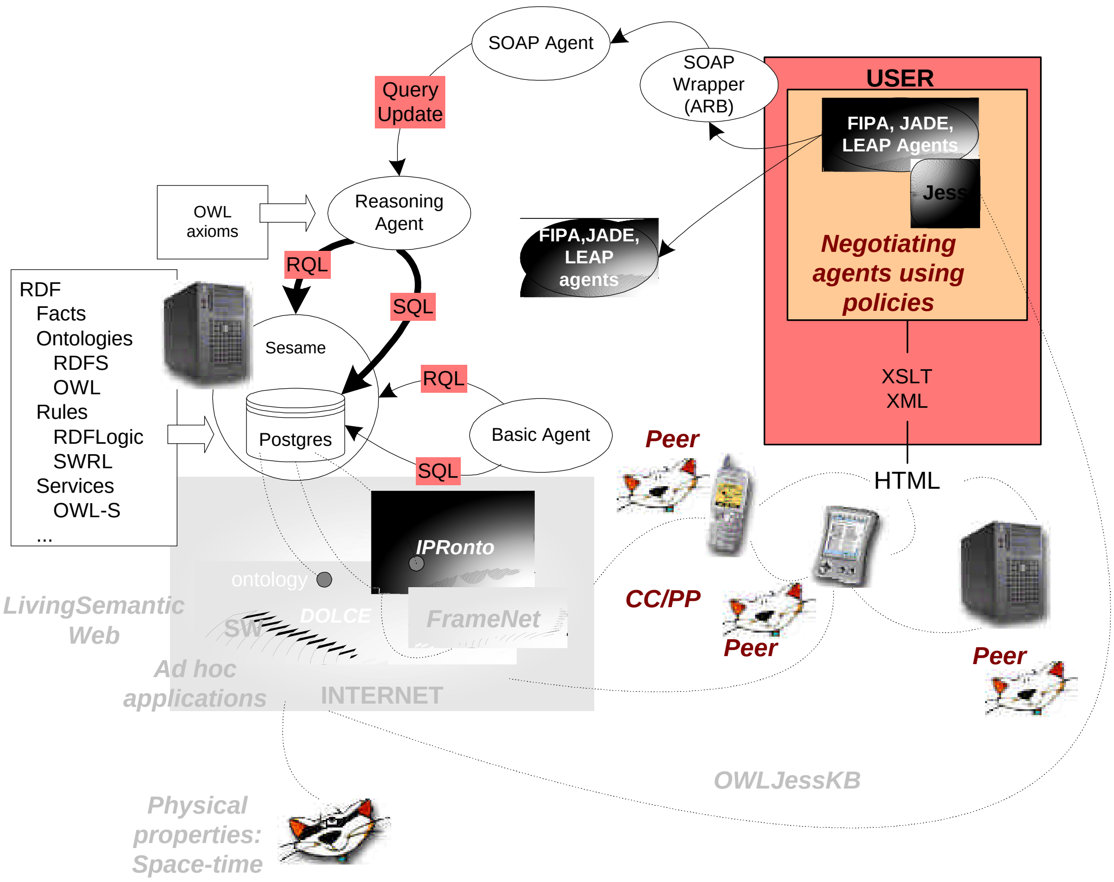



**Figure 33 Semantic Web Agents Architecture**

## 5.1. Negotiation components in a MAS

In the introduction some statements about the components of the negotiation process have been defined, the question is how they are interpreted in the definition for MAS.

There are some actors performing negotiation process, for instance, in this definition is ***the initiator*** and ***the participants***. The ***minimum set of information*** is controlled by the policy or negotiation rules. ***The contract*** represents this negotiation because it is going to be built dynamically, changing from initial parameters as the negotiation goes on. This contract is represented by the messages content exchanged by the agents.

It is possible to find a great variety of MAS: Aglets, Grasshopper, FIPA, JADE-LEAP, Swarm…[108] every platform has its advantages and disadvantages. The question is justify why one is chosen in front of the others and this is only possible when there is a problem to solve and one MAS seems the most appropriate in that case. In every case negotiation process can be developed. Thus the question is to justify the election of one MAS.

JADE-LEAP (see 4.2.2), a MAS coming from telecom domain, was the initial system to be tested in order to perform agents' negotiation. This platform was chosen although Grasshopper presented many desirable features, as persistence or mobility, and it was also based on FIPA standard.
Moreover, JADE-LEAP dealt with semantics. At first, it was a key advantage because the contracts to be negotiated were very complex because the goal being negotiated was IPR (Intellectual Property Rights). In order to deal with semantics, an Expert System was used to reason about them and JADE-LEAP was integrated into JESS (see section 4.2.5). It seemed a suitable platform for the initial problem (see Figure 38).

## 5.2. Policy: negotiation rules definition

In the previous section it was shown that a MAS as JADE-LEAP needed other components to develop an architecture where negotiation process about multimedia content can be done.

It has been explained that FIPA-Contract-Net Protocol controls negotiation in JADE-LEAP, however it does not allow to verify when and in which conditions every protocol step has to be realised. Policy or negotiation rules appear to be this missing piece. However, it was an item contemplated by JADE-LEAP platform. As it was explained in the previous section one of the reason for choosing it was the fact that it was highly connected to JESS, an expert system.

JESS, (Java Expert System Shell) or PROLOG, this kind of software allow to contain negotiation rules that can be specified in semantic languages as RDF or own languages as JESS language known as CLIPS (see section 4.2.5). Moreover, the rules and the facts in these systems can be removed dynamically. It means that changes can be made automatically.

From this perspective, a Semantic Web Agents Architecture was designed as it can be seen in Figure 34. The user interaction is represented with a triangle where some technologies are presented as WML[139] for telephones, HTML for browsers, VoiceXML for computers, centralized by a MAS, i.e. JADE-LEAP with JESS running. These agents can send information using Wrapper depending on where it is going to be the search: in a database as Postgres [160] or in a registry as UDDI [89].

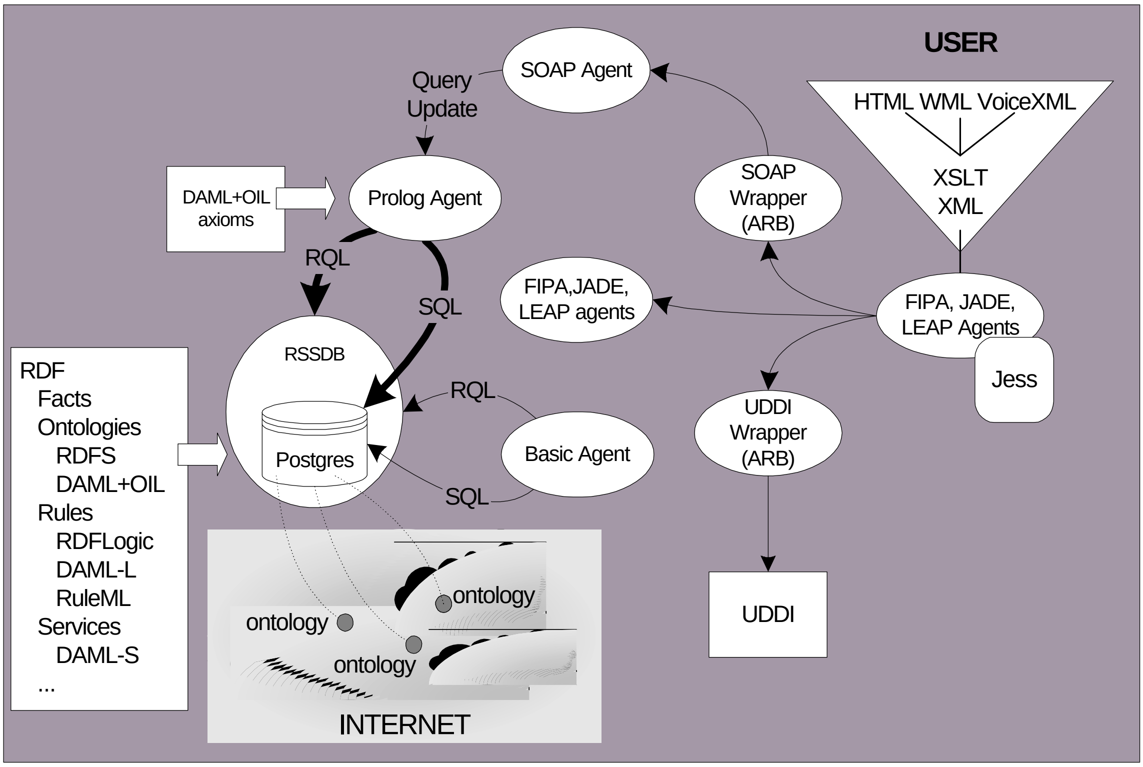



**Figure 34 Semantic Web Agents Architecture v.1.0**

Thus, JADE-LEAP [115, 116] was tested together with the SOAP agent (see Figure 35 and final reference [109]) and the queries to the ontologies-enabled persistence layer [160]. Semantics was the challenge for the architecture and an unavoidable feature in order to perform the negotiation process about Digital Rights Management, more concretely the copyright negotiation process of Intellectual Property Rights.

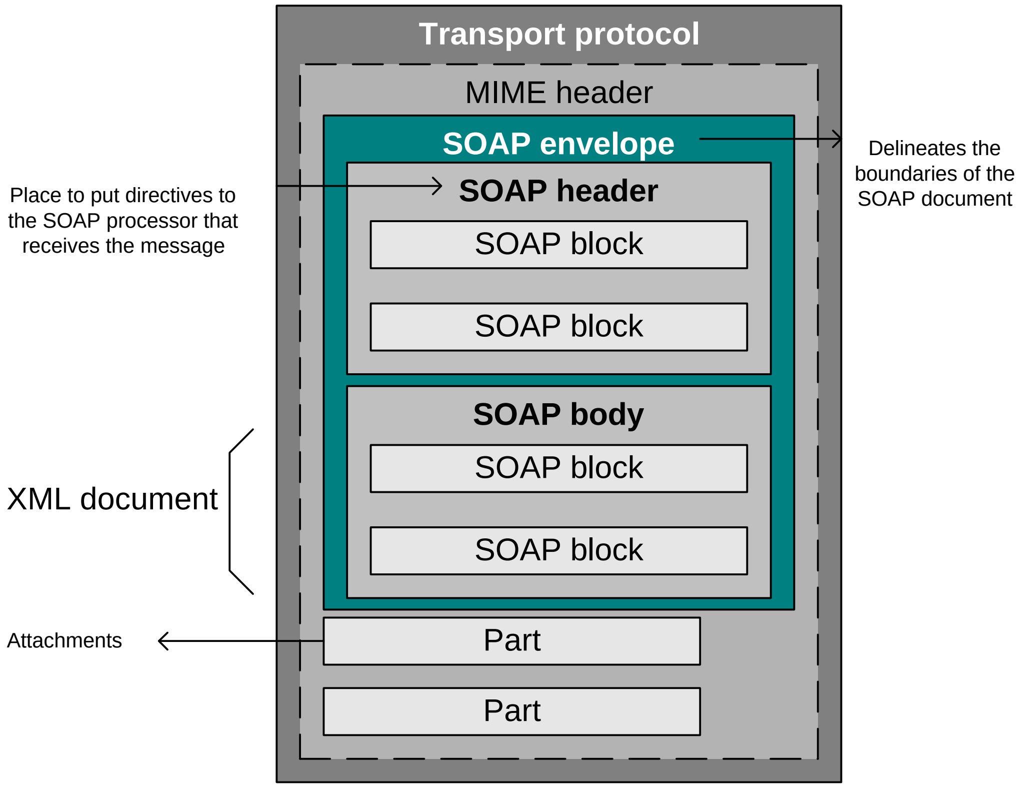



**Figure 35 SOAP message structure**

To design an architecture able to negotiate copyright or Intellectual Property Rights was not a simple task because the different laws for every country. It was necessary to begin with something simple and increasing complexity for solve the multiple questions that arose.

When trading with multimedia content, one of the key issues is what does it happen with the copyright or Intellectual Property Rights (IPR) that are normally associated to that content, being a picture, a song, a video clip or any other audiovisual content.

When managing those IPR, what is normally called Digital Rights Management (DRM), we have to face with many issues, most of them not having currently a really optimal solution.

In the first stage a system developed for e-commerce of video content (MARS project), initially focused on data localisation aspects following different metadata techniques for its implementation was chosen. Then, copyright information to that metadata was added, including watermarks in the content itself. The results of this project were the main starting point.

Apart from metadata, in relation to representation of the information, there are other aspects to consider when dealing with DRM, such as the issues associated to the Rights Expression Languages (REL) and Rights Data Dictionaries (RDD). For this, the current standardisation efforts in MPEG-21 [110], that try to integrate previously existing initiatives, is a clear example.

A different approach to that problem consists on specifying an IPR ontology able to add semantic information for the management of digital rights; this allows the interoperability of different applications. The architecture presented in Figure 34 shows how this idea was considered in addition to the MAS platform and the expert system [111]. It is going to be explained further in section 5.10.

## 5.3. Case of study: IPR Business Models

In order to work on DRM over the designed architecture, it is necessary to select a base model for IPR representation and negotiation. The IMPRIMATUR Business Model [112] was the first step in order to identify the roles that participating entities may take: *Creator, Provider, Rights Holder, Distributor, IPR Data Base or Watermarking & Fingerprint marking.*

A simplified and specific model, the one it is going to be implemented, consists on the use of a Broker (with the role of Distributor) in charge of being an intermediary between providers of multimedia material (content providers) and customers interested in buying that material and the corresponding rights for use and/or commercial exploitation. From a functional point of view, these copyrighted multimedia material providers may also assume the roles of Creator and Rights Holder in the same entity.

Furthermore, the broker stores and keeps up to date information about the multimedia material for sale in the system (from all content providers associated to the broker). This includes information about the terms and conditions under which commercial electronic transactions are done. Figure 36 illustrates this Broker Based IPR General Model.

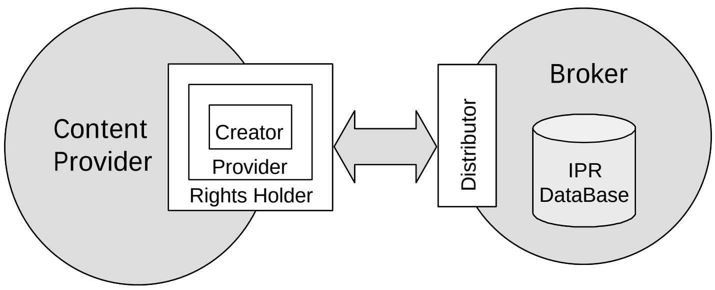



**Figure 36** Broker Based IPR General Model

## 5.4. Negotiation of IPR conditions in the IPR General Model

A negotiation process has always a framework; in this case it is IPR General Model (see Figure 36). The following paragraphs explain how these parts exchange information.

Based on the IPR attributes set, when a buyer requests, to the broker, a purchase of audiovisual material subject to copyright, the broker extracts IPR information from its database and presents an initial offer to the buyer. This information allows the buyer to take a decision on how to buy IPR, i.e., to know what are the copyright rules associated to the asset, to decide if to re-sell it, etc. To facilitate this process, a negotiation mechanism has developed, and it is described later.

The negotiation protocol, that it is part of the “Service Request” phase in an e-commerce model [113], has three sub-phases:

1. Initial offer,
2. Co-operative contract production, and
3. Payment.

In the Contract production sub-phase, the most complex and important one, there are several alternatives over which to work. First, the selling entity initiates the protocol with an initial proposal of digital rights conditions, normally taken from a pre-defined subset. After that, the buying entity has three alternatives:

1. Accepting the offer,
2. Making a counter- proposal and
3. Rejecting the proposal.

After the initial proposal, the negotiating entities elaborate the contract, using the negotiation protocol, from the sequence of proposals and counter-proposals until a final agreement is reached, forming then the final electronic contract.

## 5.5. An implementation with MAS: NewMARS

The Semantic Web architecture (see Figure 34) is going to be developed in the context of the NewMARS[160] project, see Figure 37, integrated as a subproject of the AREA2000 project [161]. Results from previous work [114] in our research group are going to be used.


**Figure 37 NewMARS presentation**

As an example, a specific Digital Rights Management negotiation scenario, taken from MPEG-21 (see section 4.3 for more details) work is going to be used to develop the model. In the scenario, *the user is a web designer that has decided to use a specific image in her current web work*. *She wants to locate a specific version of this image and acquire the necessary digital rights to use it*.

The phases of this scenario, according to the NewMARS implementation, are: *user interaction, search, negotiation, outcome presentation and control*. A sequence diagram of them, except for the background control phase, is shown in Figure 38.

In the first approximation, the context was the multimedia content negotiation, where four phases were present: *user interaction, search, negotiation protocol and results outcome* (see Figure 38).

In the negotiation process, the agents exchanged messages. If an agreement is the goal, it seems reasonable that every part (every agent) has to understand what the other part (other agent) is negotiating. This argument seems to be a plausible one to choose a MAS, as it was done with JADE-LEAP. It was tested with mobile and non-mobile agents [115,116] in the multimedia content copyright domain.

In Figure 38 there is the negotiation process, where technological agent is represented by a square headed agent while the user agent is circle headed. Moreover, four parts can be distinguished :

1. *User interaction*: Agent represent user in the negotiation process. The agent processes user profile and performs negotiation in user's name.
2. *Search*: agent interacts to a meta-search agent to find multimedia document in a repository or content directory. This meta-search is specialised on that [117].
3. *Protocol negotiation*: The core of negotiation process is holding a protocol, exactly a *FIPA contract-net protocol* that can be iterated or not. And there are basically two roles, in this case: the user's agent and the licensing agent by the other. The licensing agent has the authority to make license or contracts in the name of the Rights holder, which is at last, the receptor's payment.
4. *Results:* A contract or a license is generated from the negotiation process

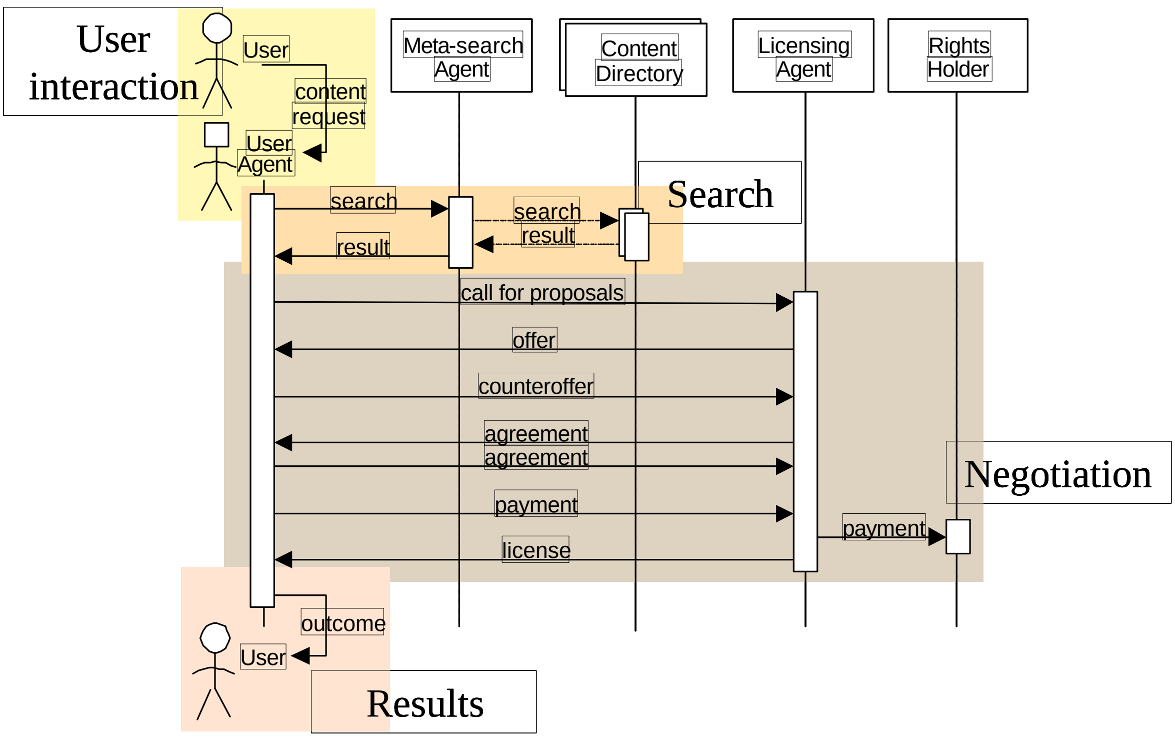



**Figure 38. Negotiation process**

To fulfil this scenario, the web designer of this example uses the NewMARS facilities, so she interacts locally with her NewMARS user agent residing in her mobile device. She uses agent's GUI to determine all the criteria required to select the image she is interested in. Then, the agent enters the search phase and it migrates to another agent-platform container. This new container is one of the server kind, where it has better Internet connectivity and processing power. Thanks to these greater resources, it can carry on less constrained searches and a more accurate negotiation process of the required image.

The search is performed through the NewMARS meta-search engine [117] that looks for the required image in some Internet image directories. When the content is found, the meta-search agent returns a reference to the agent managing the image rights and the image identifier.

Once the user agent owns enough information, it can start the automatic negotiation process. The retrieved licensing agent is contacted and a call for proposals is issued. An initial offer is received, if the licensing agent really has the requested image. From this point, some counter-offers may be interchanged till the negotiation ends due to a reject or an agreement.

The negotiation results are then communicated to the user. To facilitate user interaction, the user agent returns to the agent-platform container at the users mobile device. In parallel, a post-agreement background phase is initiated in the NewMARS system (see Figure 39 NewMARS Architecture). This control phase is conducted to guarantee a fair use of the negotiated product.

A more detailed description of these phases is given in the following sub-sections.

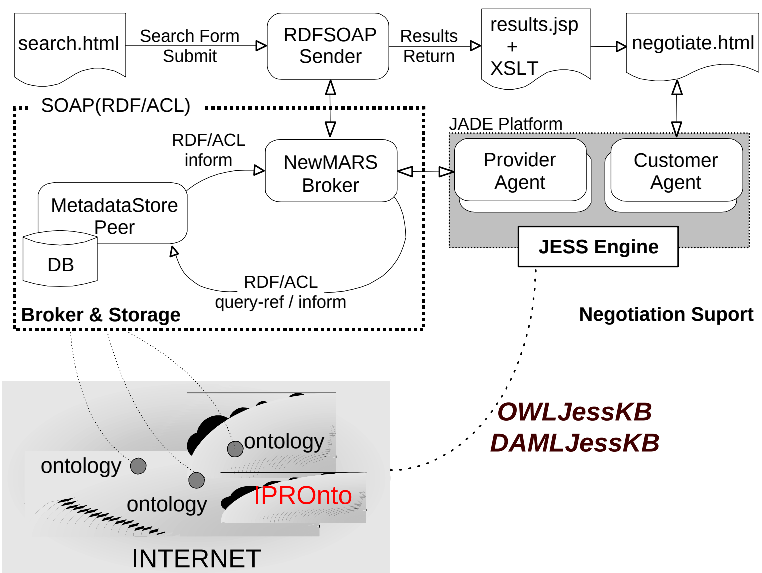



**Figure 39 NewMARS Architecture**

### 5.5.1. User interaction

The user initiates her user agent for images search and negotiation. This can be done in her desktop or, in our scenario, in her mobile device. The GUI shows up and the user interacts with the forms shown to specify the image she is interested in and in what conditions.

The GUI allows defining image characteristics, like image format, title, subject, size, dimensions, etc. The user can also determine the negotiation conditions, like price, allowed uses, period of time, etc. During the interaction, the user agent checks the defined image properties and values against the used ontologies, one for images and another one for digital rights. Thus, only valid constraints can be defined and the agent can assist the user during their definition. The constraints are modelled using RDF Model and Syntax [118] and the ontologies against which they are checked are modelled using RDF Schema [147].

Finally, the user submits the checked search (see Figure 40), and negotiation conditions. The user agent internally stores them and enters the search phase, detailed in the next section.

### 5.5.2. Search

When the user agent enters the *search phase*, it moves from the lightweight container in the user wireless device to a server container in a wired and more powerful machine. From this new location the user agent contacts a directory of services to locate the meta-search engine. This directory is initially implemented using the FIPA [119] Directory Facilitator (DF) provided with the FIPA-compliant agent platform that is used.

The DF returns a pointer to the agent implementing the meta-search engine, which previously registered itself in the DF. JADE-LEAP GUI represented by RMA agent takes into account.

Once the user agent has located the meta-search agent, it can use its predefined interaction protocol. A FIPA ACL message containing the image characteristics and negotiation constraints is sent. The meta-search agent processes it and, after performing the necessary searches, returns a set of licensing agents' locations. They have registered themselves as negotiators of the requested image. One of them is selected and the negotiation process starts.

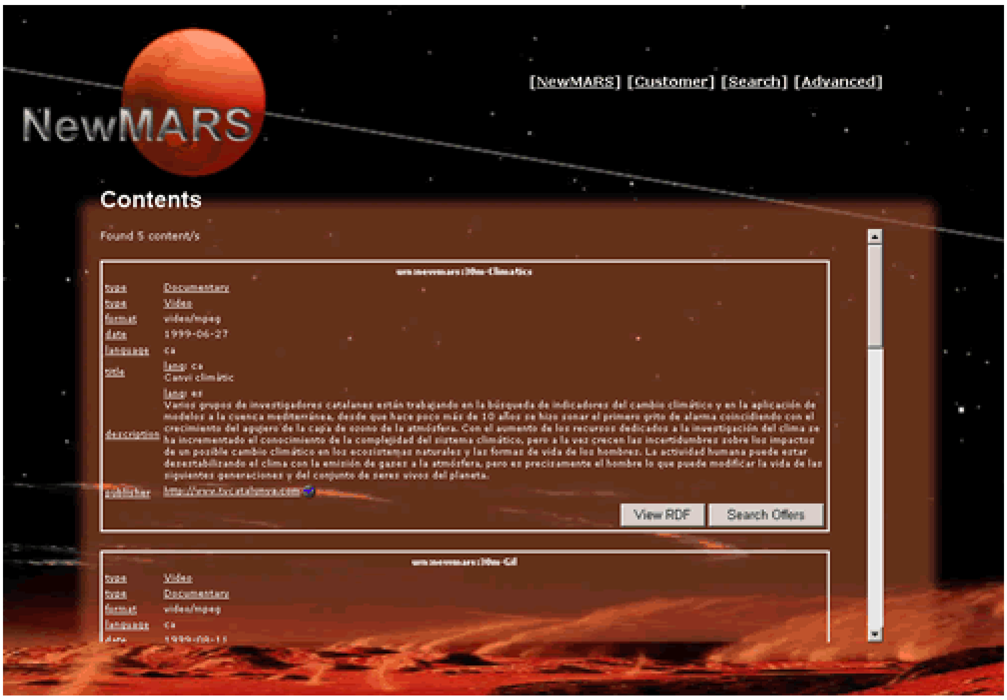



**Figure 40 NewMARS contents**

### 5.5.3. Negotiation protocol

Once the user agent has selected a reference to a provider of the image it is looking for, see Figure 41, the negotiation to obtain it begins. The negotiation protocol is obtained from the agent platform, where it has been previously registered.

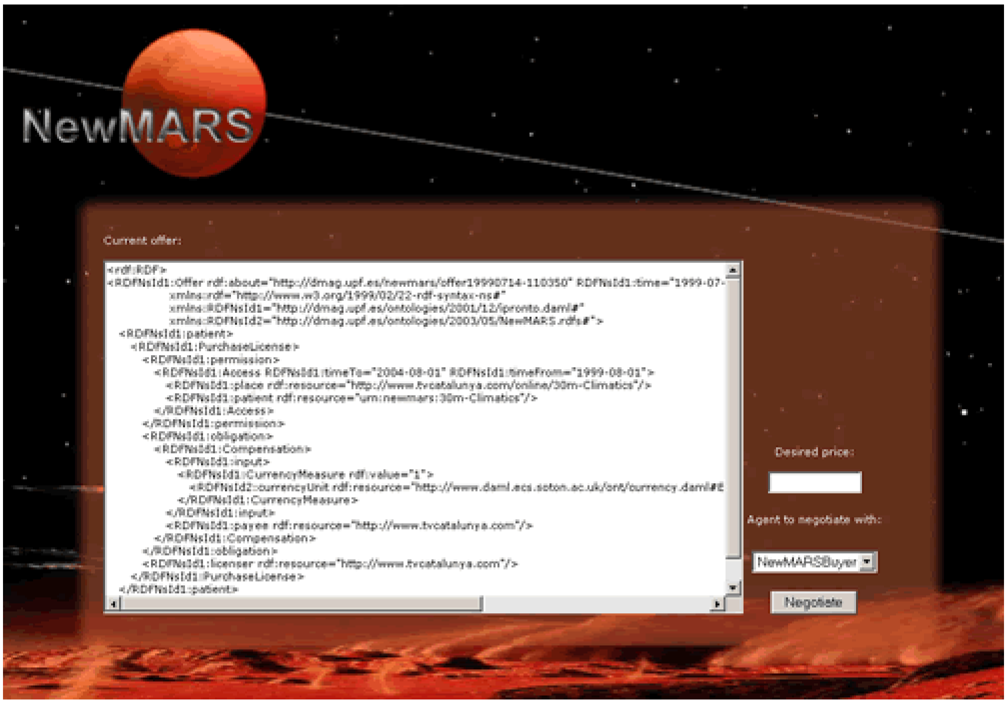



**Figure 41 Negotiation NewMARS offer and agents to choose**

First, the customer agent issues a request for proposals referred to the desired image. Then, the licensing agent responds with an initial offer if it has the requested content, a refusal otherwise. An example of offer is shown in Figure 41. Given that we are considering a totally automatic scenario, the user agent analyses this offer and decides what to do afterwards. If it does not accept the offer conditions, it can formulate a counter-offer.

**Table 7. Example of offer serialised as RDF/XML**

```
<rdf:RDF     xmlns:rdf  = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:ipr  = "http://dmag.upf.es/ontologies/ipronto#"
        xmlns:xsd  = "http://www.w3.org/2000/10/XMLSchema#">
<ipr:Offer rdf:about="http://dmag.upf.es/mars/offer20020211183424">
<ipr:patient>
    <ipr:PurchaseLicense>
        <ipr:licenser rdf:resource="http://www.howlinwolf.com"/>
        <ipr:licensee rdf:resource="http://chiTouristGuide.org"/>
        <ipr:permision>
        <ipr:Access>
            <ipr:place rdf:resource="http://chiTouristGuide.com/issues/march02"/>
            <ipr:patient rdf:resource="http://www.howlinwolf.com/imgs/0973.jpg"/>
            <ipr:user rdf:resource="http://chiTouristGuide.com/members"/>
            <ipr:timeFrom><xsd:date rdf:value="2002-03-01"/></ipr:timeFrom>
            <ipr:timeTo><xsd:date rdf:value="2003-03-01"/> </ipr:timeTo>
```

```
            </ipr:Access>
            </ipr:permission>
            <ipr:obligation>
            <ipr:Compensation>
                  <ipr:payer rdf:resource="http://chiTouristGuide.com "/>
                  <ipr:payee rdf:resource="http://www.howlinwolf.com"/>
                  <ipr:input><ipr:DollarQuantity rdf:value="100"/></ipr:input>
            </ipr:Compensation>
            </ipr:obligation>
            <ipr:time><xsd:date rdf:value="2002-02-11"/></ipr:time>
      </ipr:PurchaseLicense>
</ipr:patient>
</ipr:Offer>
</rdf:RDF>
```

The same applies for the licensing agent when it receives the counter-offer. This interchange of counter-offers continues till any of the parties abandons the negotiation or an agreement arises.

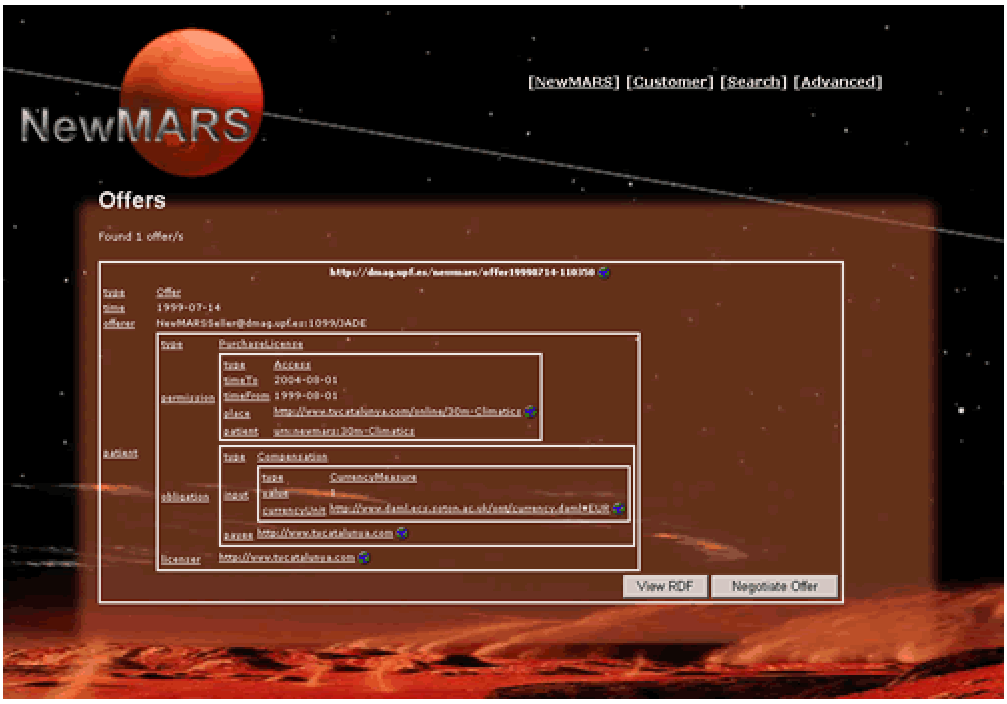



**Figure 42 NewMARS Offers**

Finally, when any of the parties agrees with the last offer (see Figure 42), the other can also agree and an agreement is reached. An electronic contract is produced, i.e. a RDF/XML document quite similar to that shown in Table 7. It contains the agreed conditions and two extra elements pointing to both license consenters. Both parties digitally sign it with XML Signature [120] and the result is a license that authorises the costumer to use the negotiated content under the stated conditions.

## 5.5.4. Outcome presentation and control

When the negotiation has ended, the results are communicated to the user. To facilitate user interaction, the user agent returns to the agent-platform container at the users wireless device.

If the user agent has succeeded, a RDF document representing the achieved agreement is presented to the user through the agent GUI. It is formatted in a user-friendlier manner and it contains a URL pointing to a location from where the licensed image can be retrieved.

All the involved parts have digitally signed the RDF agreement; really their representative agents have performed this action. Therefore, it can be used as a licensing contract that proves that the requested uses have been authorised.

If not agreement has been achieved the user agent presents the last outcomes of the negotiation process, the offer, counteroffer or refusal where the negotiation broke.

Finally, the control phase occurs in parallel and continues in the background. Its purpose is to monitor that the customer fulfils the conditions established in the agreement license[114]. We outlined different approaches to this issue using mobile agents.

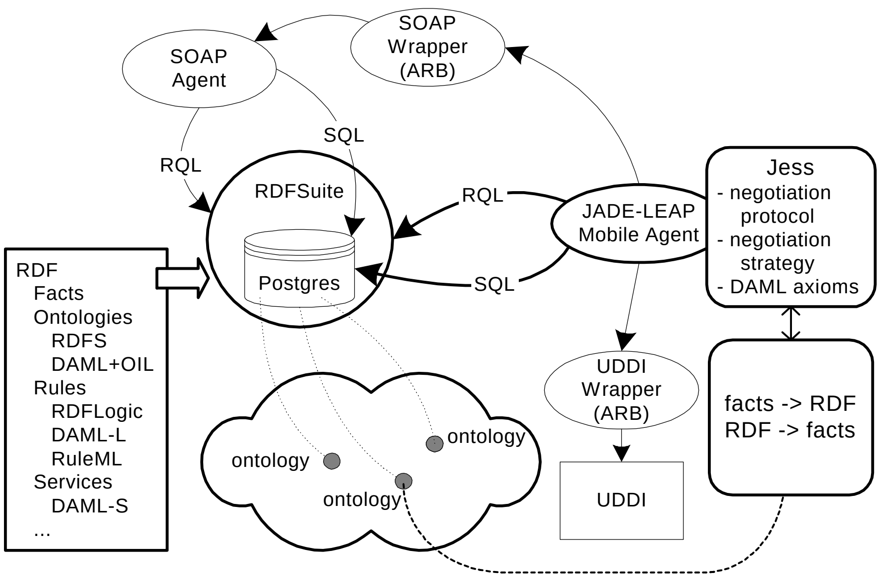



**Figure 43 Semantic Web Agents Architecture v.2.0**

Summarizing the technologies that have been used:

The managed data, the facts, are stored in a database using ICS-FORTH RDFSuite [121] that provides persistence and a RDF query language, RQL.

The RDF facts are structured using different ontologies, allowing to model processes and rules, like DAML-S [122], RDFLogic [123] or RuleML [124]. DAML+OIL [125] capabilities are planned using Prolog and DAML+OIL axioms [126].

Agents are modelled using JADE-LEAP. They are coordinated by an inference engine, in our case JESS, that also cares about the policy of the negotiation protocol.

Moreover, a second version (see Figure 43) of the Semantic Web Agents Architecture was developed to improve v. 1.0 (see Figure 34) because a connection between JADE-LEAP, JESS and ontologies was needed. For that reason CLIPS files (the Jess expert system format) store rules that govern coordination and negotiation protocol besides rules that allow understanding DAML/OWL ontologies.

The coordination role consists on controlling invocations of JADE agents and their association to negotiation protocols by loading the corresponding CLIPS file that contains them. This file manages negotiation protocols, so it takes cares of message interchanging and translating the RDF content to Jess facts. Translation is performed by a DAMLJessKB/OWLJessKB module [127]. This can be done because Jess has DAML axioms as Jess rules, which represent an ontology inside this database. Jess knowledge base interprets these facts firing rules and generating a result that will be converted back to RDF or DAML (see Figure 41).

## 5.6. A first step towards user customisation

Finally the connection between agents belonging to a JADE-LEAP platform and final users, who are represented by mobile agents inside a wireless device, is going to be presented.

In February 2002, JADE-LEAP was ready to integrate wireless devices with a minimum of quality of service and it was possible because the wireless devices increased their capabilities as store data or processor quality.

On the other hand, Sun had developed J2ME, intended to wireless devices too (see Figure 44), where there are MIDP profiles [128], which are a subset of PersonalJava [129], allowing to build more sophisticated applications because it has been thought for PDAs.

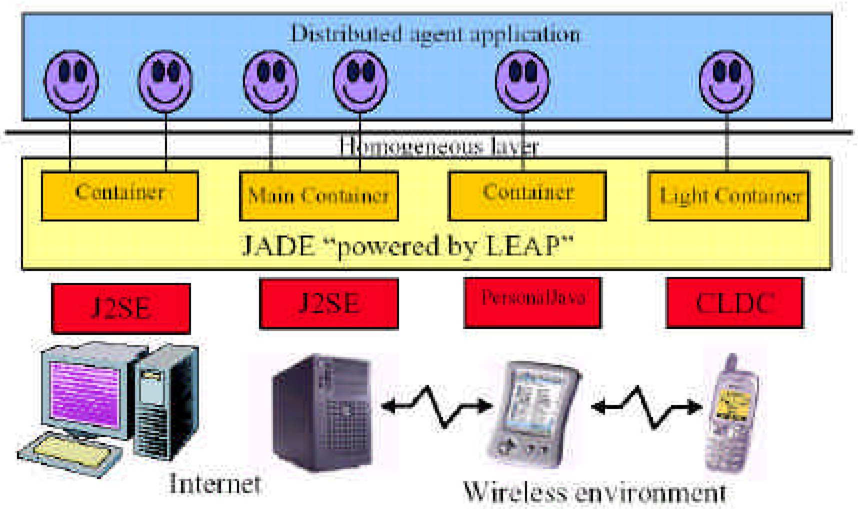



**Figure 44. Agents and containers in JADE platform**

At this moment, there were some restrictions, as GUI on PersonalJava. It must be developed with the standard Java AWT (Abstract Windowing Toolkit) and not Java Swing interface is possible. In every wireless device there is an AWT GUI (Graphical User Interface) associated to our agent that shows an agent list and its behavior policies.

Agents and policies are mobile (see Figure 45) and they are moved when the user decides it, so the agents send a call to Jess in the main-container. In the next section we will see how Jess engine with the policy manages mobility.

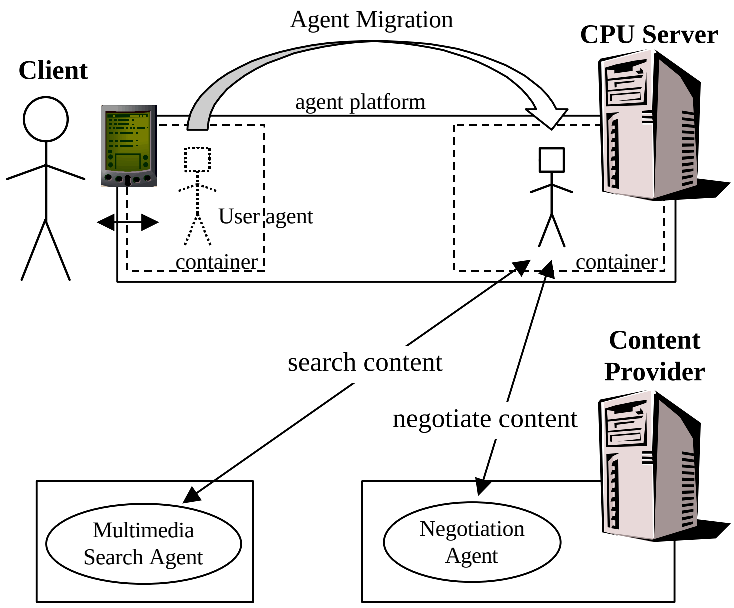



**Figure 45 Negotiation mobile-agent scenario**

## 5.7. Mobility policy

Mobility is managed from the Jess engine in the main container, a J2SE [130] one (see Figure 44). It contains a set of rules that move agents when it is convenient and balance the load of the different containers.

As we can see in Figure 45, a mobile agent has capability for moving from a wireless device, as iPAQ (a model of Compaq company) or PersonalJava (it means the software container), to another resource in the same platform (intra-platform mobility[131]) to carry out a mission. It would be valuable when, for instance, a negotiation must be done, because it means a lot of messages going up and down. The process takes place in a J2SE container, in a local server, so Jess rules govern the negotiation protocol and its policy.

The Jess inference engine governs agents' behaviour, what includes their mobility patterns. User agents reside initially in the client hardware device, but these devices can have limited connectivity and processing power. When an agent wants to search and negotiate some content in the Internet, it is more efficient to do that from a wired and powerful device, what we have called a CPU Server.

Under these circumstances, the mobility behaviour rules are put into action, see Figure 46. The files list show how is possible to implement mobility policy in the next page. Moreover, when an agent is operating from a mobile location and willing to start a connection intensive interaction, if there is an available server location, it is moved from the mobile location to the server.

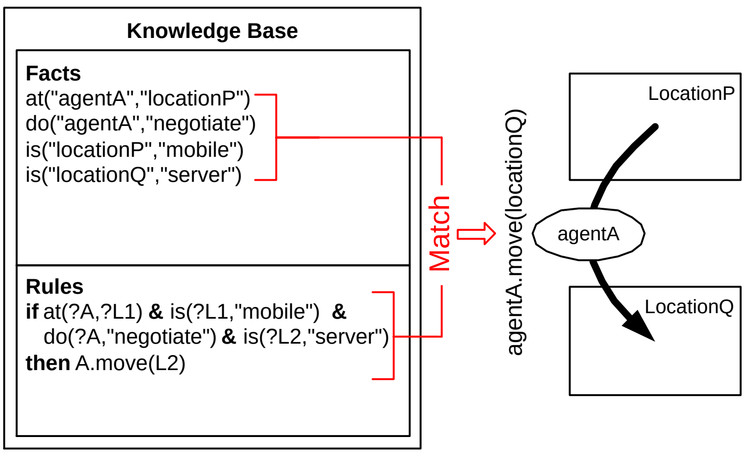



**Figure 46 Example of rule governing mobility**

The architecture presented in the previous section is only the first step towards a mobile and transparent content management platform.

In this section the new features that have been planned are introduced. They are basically necessary because delivery contexts are becoming more and more heterogeneous. However, users do not want to have to deal with different particularised interfaces.

Mobile agents can become this homogeneous content management system and provide the means to make content negotiation and customisation particularities totally transparent to final users (see Figure 47, there is an example how to control mobility from different containers, it is possible to move an agent from a light container as Container-7 or Container-8 to the Main-Container or vice-versa. Virtually it was possible to create with personal Java (LEAP) a container that simulated a PDA. See Figure 44 for containers classes. Mobility is thought in this example as to be intra-platform, however nowadays it is possible to build inter-platform mobility.

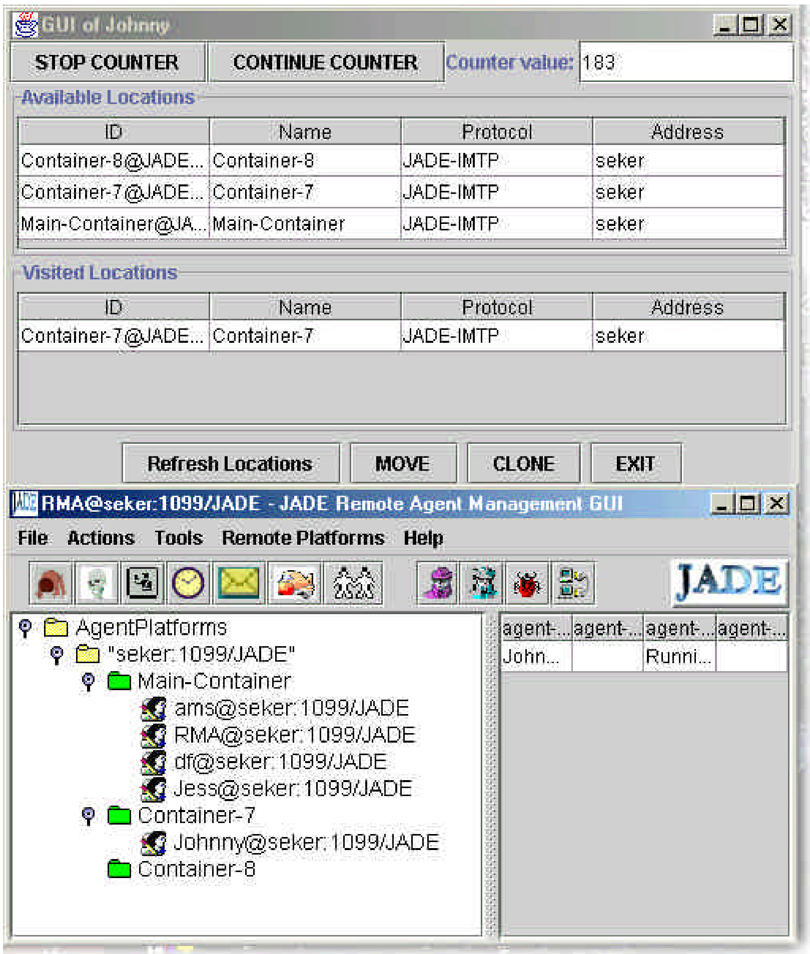



**Figure 47 RMA agent showing mobility between containers.**

Moreover, at the same time that delivery contexts diversity increases, new specialised technologies appear. They are related to delivery features but also to other accessory ones with which our mobile agents solution is required to interact. Therefore, we also present our preliminary explorations of an interoperability framework for mobile agents solutions. The following example files will describe more accurately the process.

## 5.7.1. MobilityDaml.clp file

This file tries to integrate quickly the ontology *(mobilityOnto.daml)* into JESS. For that reason, the ontology is stored in other format *(mobilityOntoJess.sav)* using DamlJessKB for translating to JESS language. It is the same process for OWLJessKB. (see Table 8)

**Table 8. Ontology integration into JESS**

```
; -*- clips -*-

;; ------------------------------------------------------------
;; Generate Rete save-file with DAML+OIL plus ontology
;; ------------------------------------------------------------

;(set-reset-globals nil)
;(defglobal ?*damlJess* = nil)
;(defglobal ?*damlRete* = nil)

;(bind ?*damlJess* (new edu.drexel.itcsl.daml.DAMLJessKB))
;(call ?*damlJess* loadDAMLFilenameOrURL "mobilityOnto.daml")
;(bind ?*damlRete* (get-member ?*damlJess* rete))
;(call ?*damlRete* executeCommand "(bsave mobilityOntoJess.sav)")
```

The Rete algorithm is the basis for JESS structure, so the next step is to create an object Rete from *mobilityOntoJess.sav*, in other words, from the ontology stored in the format created by DamlJessKB. (see Table 9)

**Table 9. Rete Creation**

```
;; ------------------------------------------------------------
;; Load Rete with DAML+OIL plus ontology ready to work
;; ------------------------------------------------------------
(bload mobilityOntoJess.sav)
(set-reset-globals nil)
(defglobal ?*damlJess* = nil)

;-- Create DamlJessKB object and assign it loaded Rete
(bind ?*damlJess* (new edu.drexel.itcsl.daml.DAMLJessKB))
(set-member ?*damlJess* rete (engine))
```

Once the mobility ontology is loaded as Rete object, facts are needed, they can be taken from other ontology , for example *(mobilityScenario.daml)* using again DamlJessKB for translating into JESS structure (see Table 10). The same process is done with OWLJessKB.

**Table 10. Ontology loaded**

```
;-- Load DAML+OIL facts using loaded ontology
(call ?*damlJess* loadDAMLFilenameOrURL "mobilityScenario.daml")
```

The mobility policy is described by the mobility behaviour rules. It is possible to describe semantically the actions that the agent has to perform in the negotiation process (see Table 11).

**Table 11. Mobility behaviour rules**

```
;; ---------------------------------------------------------
;; MOBILITY BEHAVIOUR RULES
;; ---------------------------------------------------------
(deffunction print-migrate-message (?agent ?from ?to)
  "Prints message when agent migrates"
  (printout t crlf ?agent " moved from " ?from " to " ?to "." crlf)
)
;-- Construct full URI from namespace URI and fragmente id
(deffunction NS (?ns ?fragment)
   (sym-cat ?ns ?fragment)
)
;-- Namespaces defined
(defglobal ?*rdf* = http://www.w3.org/1999/02/22-rdf-syntax-ns#)
(defglobal ?*rdfs* =  http://www.w3.org/2000/01/rdf-schema#)
(defglobal ?*daml* = http://www.daml.org/2001/03/daml+oil#)
(defglobal ?*moby* = file:mobilityOnto.daml#)

(defrule negotiate-in-server
  (PropertyValue =(NS ?*moby* do) ?agent =(NS ?*moby* negotiate))
  (PropertyValue =(NS ?*rdf* type) ?agent =(NS ?*moby* Agent))
  ?currentLocation <- (PropertyValue =(NS ?*moby* at) ?agent ?locationAt)
  (PropertyValue =(NS ?*rdf* type) ?locationAt =(NS ?*moby* MobileLocation))
  (PropertyValue =(NS ?*rdf* type) ?locationTo =(NS ?*moby* ServerLocation))
  =>
  (retract ?currentLocation)
  (assert (PropertyValue =(NS ?*moby* at) ?agent ?locationTo))
  (print-migrate-message ?agent ?locationAt ?locationTo)
)
(defrule start-negotiation
  (PropertyValue =(NS ?*moby* do) ?agent =(NS ?*moby* negotiate))
  (PropertyValue =(NS ?*rdf* type) ?agent =(NS ?*moby* Agent))
  (PropertyValue =(NS ?*moby* at) ?agent ?locationAt)
  (PropertyValue =(NS ?*rdf* type) ?locationAt =(NS ?*moby* ServerLocation))
  =>
  (printout t crlf ?agent " negotiating at " ?locationAt crlf)
)
(run)
```

## 5.7.2. Mobility.clp file

Heading the CLIPS file, there is templates definition of roles negotiation and functions invoked for agents moving (see Table 12), moreover the mobility behaviour rules are integrated in the server side (see Table 13).

**Table 12. Templates and functions definition**

```
; -*- clips -*-

;; ---------------------------------------------------------
;; DEFTEMPLATES
;; ---------------------------------------------------------

(deftemplate Location
  (slot name)
)
(deftemplate MobileLocation extends Location)
(deftemplate ServerLocation extends Location)

(deftemplate Agent
  (slot name)
)

;; ---------------------------------------------------------
;; Deffunctions
;; ---------------------------------------------------------

(deffunction print-migrate-message (?agent ?from ?to)
  "Prints message when agent migrates"
  (printout t ?agent " moved from " ?from " to " ?to "." crlf)
)
```

**Table 13. Mobility behaviour rules**

```
;; ---------------------------------------------------------
;; MOBILITY BEHAVIOUR RULES
;; ---------------------------------------------------------

(defrule negotiate-in-server
  (do ?agentName negotiate)
  (Agent (name ?agentName))
  ?currentLocation <- (at ?agentId ?location1Name)
  (MobileLocation (name ?location1Name))
  (ServerLocation (name ?location2Name))
  =>
  (retract ?currentLocation)
  (assert  (at ?agentName ?location2Name))
  (print-migrate-message ?agentName ?location1Name ?location2Name)
)
```

In this part the initial conditions are related to define :facts that initially are true (see Table 14) in the negotiation process as which devices are present and moreover statements about them, as their name. Also, the initial actors and the actions involved in the negotiation process.

**Table 14. Initial conditions: facts definition**

```
;; -----------------------------------------------------------
;; Initial conditions
;; -----------------------------------------------------------

(deffacts initial-data
  (MobileLocation (name iPaq1))
  (ServerLocation (name server1))
  (Agent (name agent1))
  (at agent1 iPaq1)
  (do agent1 negotiate)
)
```

These lines always put in order the expert system, resetting and performing the facts included as running it for seeing which rules have fired and which are the resulting facts (see Table 15)

**Table 15. Rules running**

```
(reset)
(facts)
(run)
(facts)
```

## 5.7.3. MobilityOnto.daml file

This file is not representing a CLIPS file, it is a mobility ontology. The same that is integrated in JESS structure as it has been seeing in the first file: `MobilityDaml.clp`

In this file there are some sections: URI's and namespaces definition, see Table 16, Classes ,Properties and Instances definition, see Table 17

**Table 16. URI's and namespaces definition**

```
<?xml version='1.0' encoding='ISO-8859-1'?>

<!DOCTYPE uridef[
  <!ENTITY rdf    "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY rdfs   "http://www.w3.org/2000/01/rdf-schema#">
  <!ENTITY daml   "http://www.daml.org/2001/03/daml+oil#">
  <!ENTITY moby   "file:mobilityOnto.daml#">
]>

<rdf:RDF
  xmlns:rdf="&rdf;"
  xmlns:rdfs="&rdfs;"
  xmlns:daml="&daml;"
  xmlns:moby="&moby;"
>
```

**Table 17. Classes ,Properties and Instances definition**

```
<!-- Classes -->

<daml:Class rdf:ID="Location"/>

<daml:Class rdf:ID="MobileLocation">
   <daml:subClassOf rdf:resource="#Location" />
</daml:Class>
<daml:Class rdf:ID="ServerLocation">
   <daml:subClassOf rdf:resource="#Location" />
</daml:Class>

<daml:Class rdf:ID="Action"/>

<daml:Class rdf:ID="Agent">
   <daml:subClassOf>
      <daml:Restriction>
         <daml:onProperty rdf:resource="#at" />
         <daml:hasValue rdf:resource="#Location" />
      </daml:Restriction>
   </daml:subClassOf>
   <daml:subClassOf>
      <daml:Restriction>
```

```
        <daml:onProperty rdf:resource="#do" />
        <daml:hasValue rdf:resource="#Action" />
      </daml:Restriction>
   </daml:subClassOf>
</daml:Class>

<!-- Properties -->

<daml:ObjectProperty rdf:ID="at" />
<daml:ObjectProperty rdf:ID="do" />

!-- Instances -->

<moby:Action rdf:about="negotiate" />
<moby:Action rdf:about="interactWithUser" />
<moby:Action rdf:about="moveTo" />

</rdf:RDF>
```

## 5.7.4. MobilityScenario.daml file

This file represents the Scenario ontology. The mobile locations of the wireless devices are described by URN's in the instances section, see Table 18.

**Table 18. Scenario ontology file**

```
<?xml version='1.0' encoding='ISO-8859-1'?>

<!DOCTYPE uridef[
  <!ENTITY rdf    "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY rdfs   "http://www.w3.org/2000/01/rdf-schema#">
  <!ENTITY daml   "http://www.daml.org/2001/03/daml+oil#">
  <!ENTITY moby   "file:mobilityOnto.daml#">
]>

<rdf:RDF
  xmlns:rdf="&rdf;"
  xmlns:rdfs="&rdfs;"
  xmlns:daml="&daml;"
  xmlns:moby="&moby;"
>
```

```
<!-- Instances -->

<moby:MobileLocation
      rdf:about="urn:JADE-IPMT:dmag.upf.es:1099/JADE.iPaq1-Container" />
<moby:ServerLocation
       rdf:about="urn:JADE-IPMT:dmag.upf.es:1099/JADE.server1-Container" />

<moby:Agent rdf:about="agent1">
   <moby:do rdf:resource="&moby;negotiate"/>
   <moby:at rdf:resource="urn:JADE-IPMT:dmag.upf.es:1099/JADE.iPaq1-Container"/>
</moby:Agent>

</rdf:RDF>
```

More details about all these ideas are presented in the next subsections and summarised in the new architecture picture as it can be seen in Figure 48.

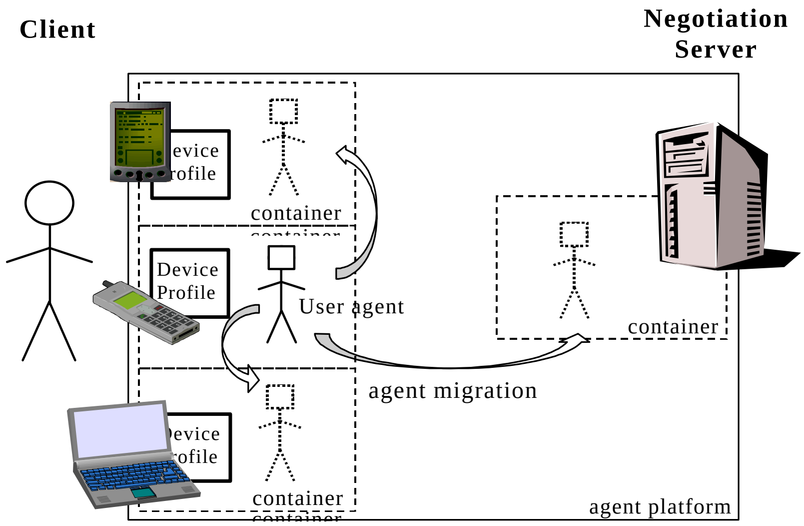



**Figure 48 Negotiation mobile-agent platform supporting multiple user devices**

## 5.8. A further step to user customisation

As it has been already pointed out, there is an increasing range of final users devices that cannot be translated to an even more heterogeneous set of interfaces for them.

Mobile agents can help users deal with this diversity but, in order to do that, mobile agents must be aware of the kind of devices they must deal with. Therefore, mobile agents need device descriptions where their characteristics are explicitly stated. There is a W3C [132] initiative that is specifically conceived for this task, the Composite Capability/Preference Profile CC/PP [133] one. A more detailed description of CC/PP is presented in the next sections.

The mobile agent knows the devices it is in charge of through the corresponding CC/PP profiles. They are explicitly stored in agent's knowledge base or implicitly by references, URL, that point to the WWW location where they are stored. Moreover, there can be particular modifications to these profiles, for instance a device with an upgraded amount of memory.

Once the mobile agent is aware of the different delivery context it manages for its human user, it must take them into account when it is negotiating content.

The mobile agent starts negotiating with a content provider and informs it about the delivery context of the content. This can be done sending explicitly the content of the corresponding CC/PP profile or a reference of it, eventually augmented by the concrete device particularities.

Using the device profiles, the content provider can adjust its offer to the requested delivery context. On one hand, it can adjust context characteristics, for instance image size, colour depth, streaming bandwidth, etc. On the other hand, it can adjust economical and utilisation conditions accordingly.

Besides that, the existing negotiation protocol stays unaffected. The user mobile agent can directly accept the proposed conditions or start a negotiation cycle that approaches them to those it thinks its user may require.

A detailed view of how customisation is managed and how device profiles are used during negotiations is going to be the main issue.

Trying to integrate the platform with future tendencies, the final part provides a way to integrate mobile agents platforms with other initiatives. For instance, P2P, Web Services, Grid… as it is going to be discussed.

## 5.9. Describing delivery contexts

A delivery context could be defined as a set of attributes that characterises the capabilities of the access mechanism and the preferences of the user. An access mechanism is a combination of hardware (including one or more devices and network connections) and software (including one or more user-agents) that allows a user to perceive and interact with the Web using one or more interaction modalities (sight, sound, keyboard, voice, etc.). Web servers and applications that adapt content for multiple access mechanisms must be sensitive to delivery context. Composite Capability/Preference Profile (CC/PP) corresponds to the W3C initiative in this line.

CC/PP provides a standard way for devices to transmit their capabilities and user preferences. It was originally designed to be used when a device requests web content via a browser so that servers and proxies can customize content to the target device, i.e. support device independence. It is a proposed industry standard for describing delivery context.

Moreover, the World Wide Web Consortium (W3C) hosts two activities that are relevant to this specification. The first of these is the Device Independence Working Group, which is part of the W3C Interaction domain. This group has produced a document describing device independence principles [134] and two informal drafts, one discussing delivery context [135], i.e. mechanisms like CC/PP, and the other one discussing authoring [136].

The second relevant activity hosted by the W3C is the CC/PP Working Group, which is also part of the W3C Interaction domain [137].

UAProf[138], User Agent Profile, is a concrete implementation of CC/PP developed by the Open Mobile Alliance (OMA) that like the W3C is organised into areas or groups and UAProf is part of the Mobile Application Group. UAProf is an implementation of CC/PP aimed at WAP-enabled [139] mobile terminals.

CC/PP profiles contain capability and preference information sent from a client to a server. In order to validate CC/PP profiles, there must be a set of rules that determine what constitutes a valid profile. According to the CC/PP structure and Vocabularies Working Draft a CC/PP profile must first meet the constraint of being a valid XML and Resource Description Framework (RDF) document. The W3C's RDF validation service [140] can be used to validate a profile in this way.

Currently there are two protocols developed for CC/PP exchange based on HTTP, CC/PP-ex [141] and W-HTTP, the last one based on WAP [142]. These protocols have many common features. They send CC/PP information in two forms: references profiles and profile diffs. A reference profile is sent as a URI between the client and the server. The server then uses this URI to retrieve the profile from a third source known as a profile repository. Profile diffs on the other hand are sent as in-line XML fragments and may or may not be present in the headers. Profile diffs are associated with a sequence number that indicates processing order.

A first step in processing CC/PP is to make the current generation presentation-oriented Web technology interoperable with the next-generation Semantic Web technology [143]. For example, CSS [144] style sheets are currently not able to take CC/PP profiles into account. CSS has, however, a feature that is closely related to CC/PP, and allows the specification of device dependent style rules. Figure 49 shows an example of a style sheet that uses bigger fonts on mobile devices screens than desktops screens of the same document.

```
@media screen { min-width:32px
body { font-size: 8 pt }
}
@media screen { min-width:640px
body{font-size: 12pt}
}
```

**Figure 49 Device dependent style rules as CSS3 extension**

Style engines need to be able to deal with these features in order to take full advantage of the information specified in CC/PP delivery contexts.

Note that the need to take CC/PP information into account also applies to XSLT [145] transformation engines. One could, for example, imagine an extension of XSLT's mode concept. For example, transformation rules could be selected in a way similar to that of the media rules in CSS. In such a hypothetical extension, see Figure 50, one could, for instance, define a rule for creating a two column layout only if the output medium is a desktop and the screen is wider than 1024 pixels.

```
<xsl:templatematch="body"
mode="screen and ( min-
width:1024px)">
...
<fo:region-bodycolumn-count="2"/
>
...
</xsl:template>

@media screen { font-size: 12pt}
}
```

**Figure 50 Using XSLT transformation engines**

In addition to taking information about delivery contexts into account, style sheets also need to take into account the semantic information that is contained in the metadata associated with the content. Currently, style selector mechanisms only match on the syntactic properties of the underlying XML document hierarchy. This applies both to the selector mechanism used by CSS and to the XPath [146] selectors used by XSLT.

In all examples above, the rules were intended to match on the <body> element of an HTML document. Using the current generation CSS and XSLT engines process general metadata it is, however, not practical to match on the semantic properties of metadata: for CSS and XSLT processors, RDF is just XML. As a result, it is very hard to write, for example, a rule that matches on all alternative XML serializations that are allowed for RDF.

A more serious problem, however, is that it is impossible to write CSS or XSLT rules that make use of the semantic features of RDF Schema [147] (RDFS). For instance, a style rule that applies to all objects that are instances of a specific RDFS class.

Future semantics-aware selector mechanisms would allow specification of rules in terms of the RDF semantics expressed in the metadata. This would extend the currently used CSS and XPath selectors, which are based on the XML syntax encoding the semantics. Consider the extended XSLT example rule in Figure 51, which uses the RDF-aware query language RQL [148] for its selector, instead of XPath.

```
<xsl:template match="RQL(http://dmag.upf.es/schema.rdf#VideoMatrix)">
...
</xsl:template>
```

**Figure 51 Semantic matching of XSLT rules using RQL selectors**

It matches on all resources that are instances of (subclasses of) the RDF class VideoMatrix. Given the fact that our RDF Schema would define "Matrix Reloaded" as a subclass of VideoMatrix, the rule would also match on the other HTML fragments. Such rules that employ the semantic relations defined in the metadata are currently impossible to write in XSLT.

Some API's as DELI (A Delivery context Library for CC/PP and UAProf) and Intel ® CC/PP SDK [149] have been implemented in the latest times to solve some of the previous problems and content negotiation has been implemented using CC/PP and WAP UAProf [150]. However, there is not a standard way of doing all this, but it is under development [151]. Its release was about June 2003.

Meanwhile, we have preferred to use the available mobile agents communication facilities, i.e. FIPA-ACL messages, following our model. The mobile agent knows the devices it is in charge of through the corresponding CC/PP profiles. They are explicitly stored in agent's knowledge base or implicitly by references, URL, that point to the WWW location where they are stored. Moreover, there can be particular modifications to these profiles, for instance a device with an upgraded amount of memory.
In the next lines, it is shown how agents interchange CC/PP profiles and use them during negotiation.

At the beginning, the mobile agent resides at one of the user devices, which is mobile agents capable, i.e. it has a mobile agents container installed. The user interacts through the agent's GUI to determine all the criteria required to select the content he is interested in. Then, the agent enters the search phase and it migrates to another agent-platform container. This new container is one of the server kind, where it has better Internet connectivity and processing power. Thanks to these greater resources, it can carry on less constrained searches and a more accurate negotiation process of the required content.

When the user agent owns enough information, it can start the automatic negotiation process through a protocol defined by rules. The retrieved licensing agent is contacted and a call for proposals is issued. An initial offer is received, if the licensing agent really has the requested content. From this point, some counter-offers may be interchanged till the negotiation ends due to a reject or an agreement.

The negotiation results are then communicated to the user. To facilitate user interaction, the user agent returns to the agent-platform container at the users mobile device as it has been seen with the mobility policy.

The first change to allow negotiating customisation is to make the content provider agent aware of the delivery context conditions. ACL messages with the “inform” communicative act are used for this [152]. An example is shown in Table 19.

More specifically, the “inform” communicative act is used to communicate explicit delivery context information, i.e. FIPA-ACL message content is the CC/PP profile serialised in its RDF/XML form [153], see Table 19. To communicate it implicitly, the “inform-ref” communicative act is used and the ACL message content is the URL reference pointing to the CC/PP online version. Although the content is in RDF/XML form, we have also inspired ourselves on the FIPA device ontology [154].

The use of RDF/XML profile encoding allows a direct integration of all this information in the negotiation process. DAMLJessKB [155] allows adding and extracting facts from Jess engine. In other words, RDF (including CC/PP delivery context profiles) is incorporated into the Jess engine. Recently OWLJessKB has appeared. It makes the same from OWL statements.

Then, the negotiation policies are implemented by Jess rules, which can be exported to a common rules interchange format, RuleML[156], and also imported. The rules take into account delivery context information to steer the negotiation. For an example see Table 20.

**Table 19. Delivery context exchange by a inform-ref preformative ACL message[157]**

```
(inform
  :sender User1MA
  :receiver ContentProviderA
  :content(
    <?xml version="1.0" encoding="UTF-8" ?>
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20010330#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <rdf:Description ID="S55_Profile">
        <prf:component>
```

```
            <rdf:Description ID="HardwarePlatform"> …
        <prf:component>…
    </rdf:Description>
    </rdf:RDF>)
:language fipa-rdf0 )
```

**Table 20. Jess rule that uses RDF metadata imported by DAMLJessKB to detect image support**

```
(defrule are-images-supported
(PropertyValue ?resource
    http://www.w3.org/1999/02/22-rdf-syntax-ns#type
    http://www.wapforum.org/…/UAPROF/ccppschema-20010330#HardwarePlatform)
(PropertyValue ?resource
    http://www.wapforum.org/…/UAPROF/ccppschema-20010330#imageCapable "Yes")
=>
(assert (include-images)))
```

## 5.10. Integrating customer and devices profiles

Thus, a great variety of mobile devices are spreading over the world, but they are not the only ones, they share communications environment with desktops and laptops among other devices. An end user can have many of them; customisation is the only way to get desired content almost everywhere. So if we want to design a true service negotiating multimedia content, it is a goal to achieve. CC/PP seems to be a veritable tool and we have reviewed how it could be useful.

However, there is still a lot of work to do in order to integrate the semantic capabilities of CC/PP in the customisation process. We have outlined a server side specific option based on the use off the Jess rule engine with RDF semantics capabilities. However, in order to see a spread adoption of CC/PP, semantic capabilities should be integrated in style sheet technologies, i.e. CSS or XSLT.

Moreover, thinking about the nearest future it is important to have in mind a global idea about the network, because customisation implies to be capable of working in many technologies, as P2P or Grid, see section 4.2.4. Thus, The goal is to propose a complete business solution based on mobile agents. They can travel around Mobile Agent Platforms and negotiate in users' name.

During the development of the Mobile Agents Negotiation Framework, it has been found some tasks that are not well faced by mobile agents solutions. For instance, negotiation decision tasks that are computation intensive. In this case, it is more convenient to encapsulate all these intensive computations as independent services and implemented by other technologies, for instance Grid networks [158].

At this point it is needed to face interoperability problems between this two different kinds of mobile computation networks. The first one, which was used till now, seems more appropriate for user mobility environments as the one resolved by the previous content negotiation architecture. The second one, the Grid, appears as the best choice in the "server" side. When intensive computations are required the best choice is to transparently integrate many computational resource into a Grid. Inside this Grid, the required computations can be distributed to attain the best affordable computational throughput.

Therefore, it seems that there are two solutions available. However, it is not necessary to choose one, both can be integrated using an interoperability layer.

Using interoperability layers to connect different solutions as Grid or P2P it is possible to get the best of the two approaches. Peer groups are configured dynamically from global UDDI repositories using Web Services tools. Once the required services have been found and configured, the ad-hoc peer group is established.

This new proposed architecture model, presented in Figure 52**,** shows how the different technological pieces are combined to meet the requirements. These pieces are Grid Computing, P2P, Semantic Web Services and Agents. These technologies are organized into a layer cake design.

At the bottom layer, there are the computational and storage resources managed by the Grid. This layer contains a set of grids that conform resource-sharing spaces with a unique logical access point. The problem is that users see these spaces like isolated islands, and, altogether, they behave like static groups of resources that must be put together by hand.

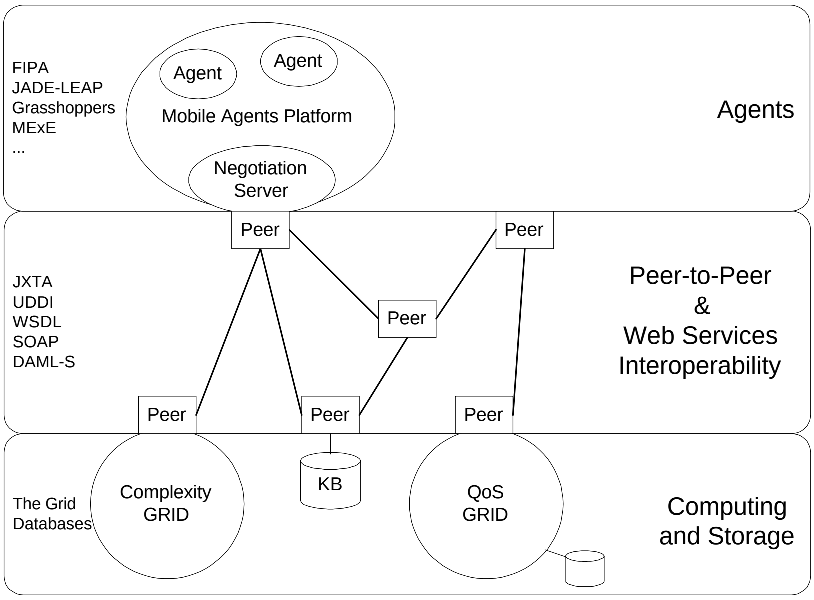



**Figure 52 Architecture integrated by interoperability layers: Agents-P2P-Grid**

The next step was how to connect P2P architecture to MAS.

The first P2P technologies emerged more than a decade ago to facilitate communication and resource utilization within the enterprise. Nowadays P2P describes the general model of using direct communication between all devices on the network. P2P is supposed to bring connectivity to the edge of the network, enabling any connected device on the network to communicate and collaborate.

If it is possible, it means applications can be more collaborative and communication-focused, and information can be more timely and accurate. While P2P is not a specific architecture or technology, it should enable a number of innovative applications, including:

- Sharing multimedia files
- New forms of content distribution and delivery
- Instant messaging and pervasive devices communicating
- Collaborative work and play such as Web-based meetings and interactive gaming

- Distributed search and indexing to enable ‘deep’ searches of Internet content that quickly yield up-to-the-minute results
- Sharing CPU and storage resources to better utilize capital investments

This fact represents a challenge for the negotiation process because a new paradigm was arisen: distributed negotiation among a huge number of users. Thus, it was necessary to determine the domain of this technology and discriminate if it was so important as it claimed to be.

Interoperability is a central goal in the negotiation process because communication is a basic requirement. Moreover, JXTA technology is designed to enable interconnected peers to easily locate and communicate with each other, participate in community-based activities, and offer services to each other seamlessly across different platforms and networks. Integrated security mechanisms such as Transport Layer Security (TLS), digital certificates, and certificate authorities help ensure security while facilitating free-flowing communication.

The main benefits of P2P technology (interoperability, distributed search and collaborative work) represent a great contribution to the characteristic of the negotiation agents. Therefore, P2P technology becomes automatically an ideal complement to the negotiation process.

An application called *“instantp2p”* was chosen to test P2P technology capabilities[159]. The goal was to prove the reliability for multimedia document exchange. However many technical questions arose related to:

***Communication***: the application was very sensitive to P2P software version. There were many bugs associated to different versions of peers' software that enables the communication between them.
***Network topology***: proxies and routers in the network topology caused the P2P application went down.

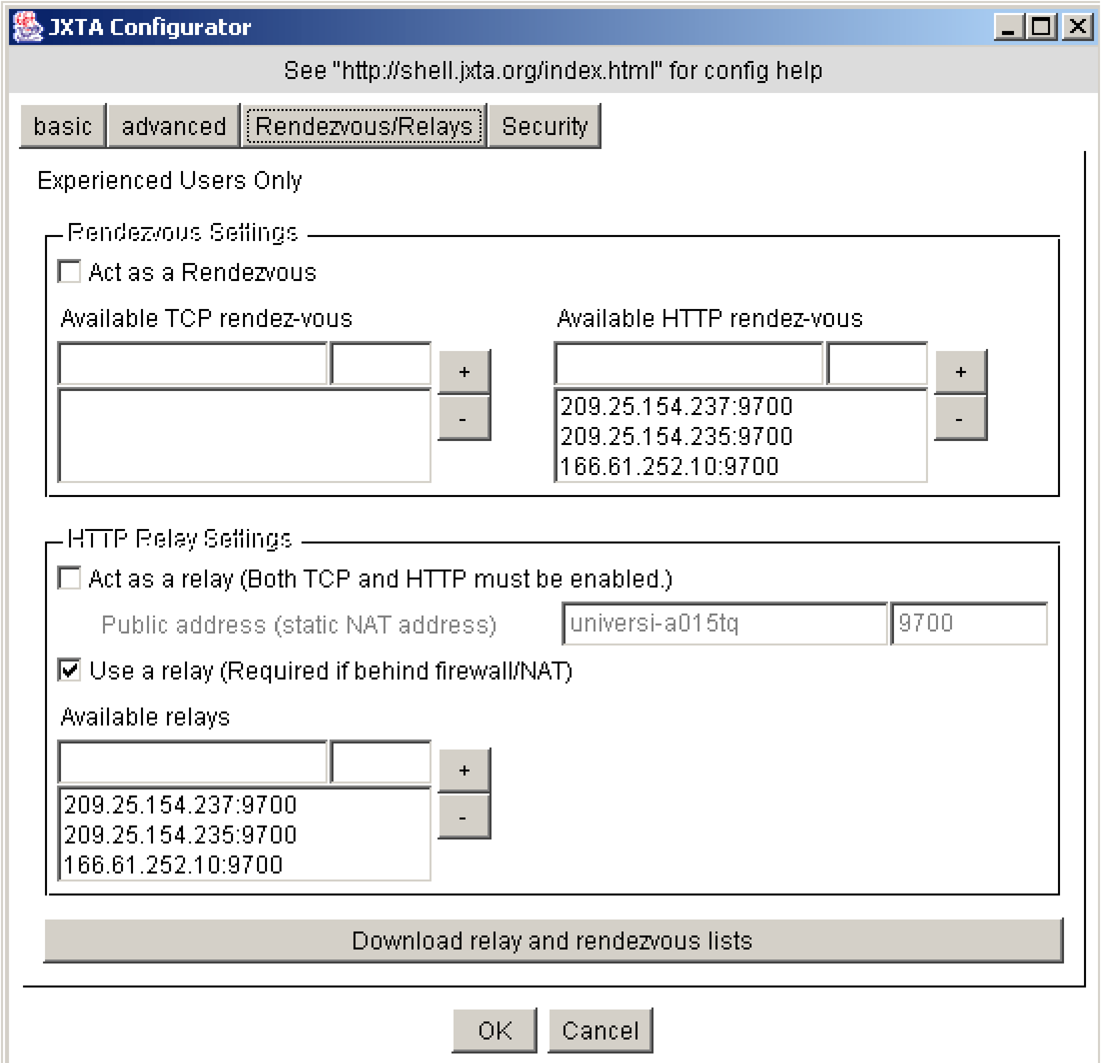



**Figure 53 Peer configuration by JXTA Configurator**

It seems that if was possible to bound these problems and provide concrete conditions to solve the questions exposed as:

- To provided  a controlled software version of P2P technology.
- To configure the network in a known way.

It was time to build the interoperability layer between P2P and MAS then, inside the NewMARS[160] and AgentWeb[161] projects, some proposals were tested to provide the interoperability between JADE and JAL/JXTA.

The architecture was improved because in P2P technology documents were exchanged in the negotiation process among peers. JAL (see section 1.1.5) provided an abstract layer to connect JXTA to the application (represented in this case by the negotiation agents in a MAS as JADE).

The goal was to design a JADE agent that could behaviour as a peer while it was controlled by an expert system as JESS. The below code shows briefly how this process can be implemented: Table 21, Table 22 show the first and the second steps.

**Table 21. First step: Defining JADE agent creation**

```
/*
 * JadeRoro.java
 *
 * Created on 15 de enero de 2003, 17:05
 */

import net.pkg.jal.*;
import jade.wrapper.*;
import jade.core.Profile;
import jade.core.ProfileImpl;


public class JadeRoro extends jade.core.Agent {
```

**Table 22. Second step: Inside the JADE agent, using JAL, a peer is created**

```
EZMinimalPeer me = new EZMinimalPeer();
```

A cyclic behaviour is associated to this peer (see Table 23), consisting on testing which peers and groups there are and being ready for receiving messages. In this stage is possible to define which group we want the peer to belong to (in this case NewMARS group).

**Table 23. Peer Cyclic behaviour**

```
class CyclicOne extends jade.core.behaviours.CyclicBehaviour
{
    public CyclicOne(jade.core.Agent a)
    {
        super(a);
    }
    public void action()
    {
        try {
            me.displayPeers();
            me.displayGroups();
            me.receiveMessage();
            doWait(10000);
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}

/** Creates a new instance of JadeRoro */
```

```
public JadeRoro() throws java.lang.Exception
{
    super();
    me.boot("JadeRoro");
    me.createAndOrJoinGroup("NewMars");
    me.publish();
}

protected void setup()
{
    CyclicOne myCyclicOne= new CyclicOne( this);
    addBehaviour(myCyclicOne);
}

public static void main(String args[])
{
    try {
        // Get a hold on JADE runtime
        jade.core.Runtime rt = jade.core.Runtime.instance();
        // Exit the JVM when there are no more containers around
        rt.setCloseVM(true);
        …
```

In the ***third step***, JADE platform is launched on port 8888, see Table 24. The agent containers (the main container and a light container) are launched also. The main container contains RMA agent, which is an interface for controlling the rest of agents. The agent-peer called "*roroAgent*" is launched in the light container.

**Table 24. Peer integration in the platform**

```
// Launch a complete platform on the 8888 port
// create a default Profile
Profile pMain = new ProfileImpl(null, 8888, null);

System.out.println("Launching a whole in-process platform..."+pMain);
MainContainer mc = rt.createMainContainer(pMain);

System.out.println("Launching the rma agent on the main container ...");
AgentController rma =
    mc.createNewAgent("rma", "jade.tools.rma.rma", new Object[0]);
rma.start();

// Launch our agent
Agent custom = (Agent)mc.createNewAgent("roroAgent",
    JadeRoro.class.getName(), new Object[0]);
custom.start();

// set now the default Profile to start a container
ProfileImpl pContainer = new ProfileImpl(null, 8888, null);

System.out.println("Launching the agent container ..."+pContainer);
AgentContainer cont = rt.createAgentContainer(pContainer);
```

Negotiation begins and a new agent is created, called *JessAgent*, which takes care about the negotiation rules or policy. Moreover two agents are launched also, representing the ***buyer*** and the ***distributor***, see Table 25.

**Table 25. Negotating agent inside JESS**

```
    //Launch negotiating agent inside JESS
    //Agent negotiate = (Agent)mc.createNewAgent("jessAgent",
        JessAgent.class.getName(), new Object[0]);
    //negotiate.start();

    //Launch  negotiator buyer
    Agent buyer = (Agent)mc.createNewAgent("comprador",
        es.upf.dmag.newmars.agents.agente.Comprador.class.getName(),
        new Object[0]);
    buyer.start();

    //Launch  negotiator distributor
    Agent distributor = (Agent)cont.createNewAgent("distribuidor",
        es.upf.dmag.newmars.agents.agente.Distribuidor.class.getName(),
        new Object[0]);
    distributor.start();
   }
    catch(Exception e) {
    e.printStackTrace();
   }
  }
}
```

In the following lines, see Table 26, the code shows how it is possible from JAL to build a peer.

**Table 26. Peer creation process**

```
import net.jxta.endpoint.Message;
import net.jxta.peergroup.*;
import net.pkg.jal.*;
import java.io.*;

public class TestPeer extends net.pkg.jal.EZMinimalPeer
{
    public TestPeer() {}
    public static void main (String args[])
    {
        EZMinimalPeer me = new EZMinimalPeer();
        try {
            me.boot("MDE" + args[0]);
            me.displayPeers();
            me.displayGroups();
            me.createGroup("MDE");
            me.displayGroups();
            me.joinGroup("MDE");

            // finally try out getJoinedGroups
            String a[] = me.getGroups();
            System.out.println("getGroups:");
            for(int i = 0; i < a.length; i++)
                System.out.println(a[i])
            System.exit(0);
```

```
          }catch (Exception e){
               e.printStackTrace();
               System.exit(0);
          }
     }
}
```

*JessAgent* represents an agent that performs negotiation rules from JESS. The negotiation policy is expressed in clips format, the JESS native language.

In Table 27, bellow, it is showed what happens when a *"Call for Proposals"*, abbreviated *'cfp'*, arrives. The negotiation continues and a message with a *'propose'* is sent.

**Table 27. Negotiation communicative acts sequence I**

```
(defrule proposal
     "When a 'cfp' message arrives from an agent ?s, this rule asserts a
     'propose' message to the same sender and retract the just arrived
     message"
?m<-(ACLMessage (communicative-act CFP) (sender ?s)
               (content ?c) (receiver ?r))
 =>
(send (assert (ACLMessage (communicative-act PROPOSE)
     (receiver ?s) (content ?c) )))
(assert (ACLMessage (communicative-act PROPOSE) (sender ?r)
     (receiver ?s) (content ?c) ))
(retract ?m)
)
```

**Table 28. Negotiation communicative acts sequence II**

```
(defrule send-a-message
     "When a message is asserted whose sender is this agent, the message
     is sent and then retracted from the knowledge base."
(MyAgent (name ?n))
?m <- (ACLMessage (sender ?n))
=>
(send ?m)
(retract ?m) )
```

Once the protocol, in this case represented by the negotiation rules, has been performed, see Table 28, the expert system looks for *'facts'* and the system initiates with *'run'*, see Table 29.

**Table 29. System resets**

```
(watch facts)
(watch all)
(reset)
(run)
```

## 5.11. Challenges

SOAP technology was tested also [109] as it was mentioned. The capabilities of SOAP communication arise to be a reliable way to exchange messages among negotiators. However, it didn't provide any remarkable novelty.

Going deeper into the environment and the policy, the navigation in the environment depends on the user. The system intelligence is in the Expert System rules and these rules and how they fire are programmed by a human being. Thus, how is possible to transform MAS and become intelligent without the human interaction is an open question. It is needed a mechanism for developing rules by themselves. In order to built this mechanism it is important to establish where the agents are, in other words, their environment. Some questions have been arisen:

It has been supposed that the environment is completely unrelated to agents and it can be separated into two. However P2P system showed that it was not possible. The value of something changes in periods of time shorter than ever. The idea of *OBSERVER and OBSERVERED THING* was more related to ever. The negotiation process is so dynamical that a new characterization was needed to help it. Is it possible to describe agent-environment accurately?

This open question was very complex. Thus, in a first step and forgetting semantics, another MAS answered some questions: swarm systems (see section 4.2.6). These systems exhibit emergent behaviour, self-organization and sometimes self-organized criticality [162]. In other words, they are systems that are on the edge of chaos [163]. The key question is how it was possible to model them.

It was a requirement to understand what kind of relations or connectors appear between agents and environment. Moreover some models were examined coming from other disciplines (artificial life (*Alife* [164]) and NK models [165] among others). Once self-organization was shown to be related to animated matter, new discoveries related it also to unanimated matter [166].

At this point some questions appear to be clear:

- Real systems behaviour, as it can be for real process, can be modelled on the edge of chaos because they are neither completely ordered nor chaotic.
- Real systems exhibit properties that can be extracted from statistical analysis.

However the agreement that represents the goal for the negotiation process was not considered, in that systems there is no a clear individual target with a clear relation to a global target.

The next section objective is to estimate the advantage of using this MAS for understanding semantics range in the negotiation process.

## 5.12. Semantics edge: Swarm systems

One question that came quickly was if a complex behaviour was necessary to make complex systems. In other words, if simple agents were able to create a real system as a real negotiation.

In the previous section, agents could manage semantic information and with an Expert System they could make decisions. Only this kind of agents could make complex negotiations?

The answer was no. Sometimes complex rules were not necessary in some cases as it is going to be explained.

For instance, taking an example, an agent A wants to negotiate something with agent B, or wants B to do something, a typical situation in MAS, see Figure 54. A and B negotiate using a negotiation protocol. It means, that they must to understand the protocol or the language they are using. It generates that an effort in communication and resources have to be made. An easy way of economize resources would be: modifying the environment in a way that agent B can recognize, see Figure 55. A or B modifies the environment of each other. There is only one thing common in A or B and when it is found it, each agent reacts. ***The idea is there is an agreement to do something (the goal) and the agents react to something external modifying the environment***.

There are many cases in biology where this phenomenon occurs, and it was the inspiration for Swarm Systems [see section 4.2.6].

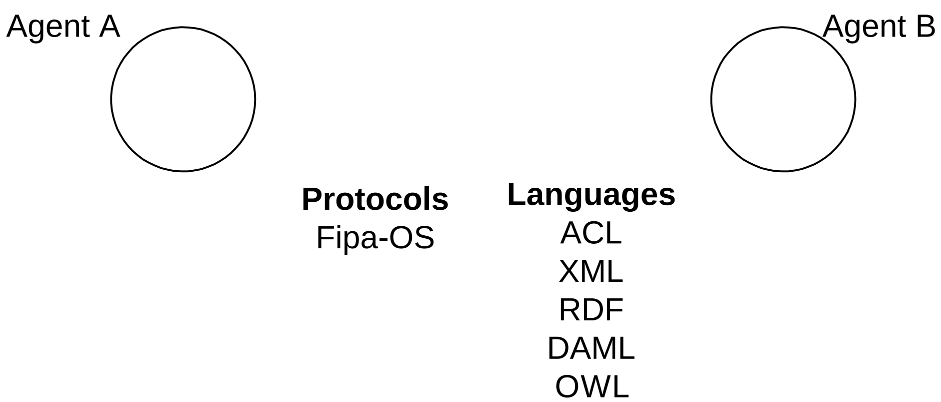



**Figure 54. Interaction in a MAS**

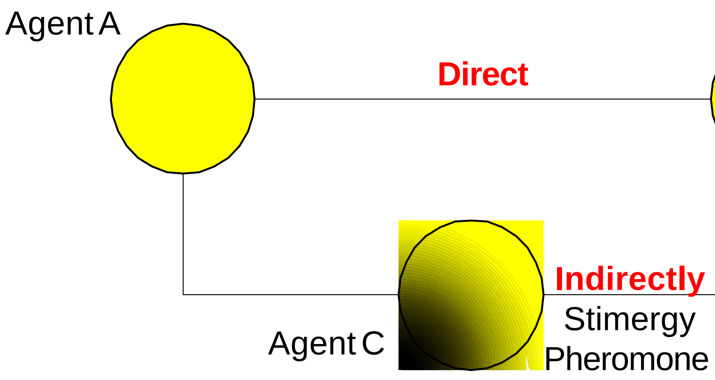



**Figure 55. Direct and undirect interactions between agents.**

Stimergy is produced when individual behaviour modifies the environment, which in turn modifies the behaviour of other individuals. It means that not always is necessary to understand the other to establish communication if there is something subjected to be modified and this result understood for every part.

It is not exactly a broker [167] because it is not another agent, is a passive element, only subjected to be modified. ***Mathematically speaking: a function that it can be the representation of natural phenomenon: Pheromone trail***.

It is easy to think in something like this, for instance, when someone travels around the world, usually find shops that have exactly the same appearance and products, and it is easy to get them even if we don't understand the country language. Branding is a key issue in commerce.

Thus, swarm systems seem to be very attractive because they take care of a global property: Self-organization.

Self-organization is a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components. The rules specifying the interactions among the system's constituent units are executed on the basis of purely local information, without reference to the global pattern, which is an emergent property of the system rather than a property imposed upon the system by an external ordering influence. It relies on four basic ingredients: positive and negative feedback, agreement, multiple interactions and amplification of fluctuations.

A self-organized phenomenon can be characterized by a few key properties: creation of spatial-temporal structures in an initially homogeneous medium, multi-stability (possible coexistence of several stable states) and bifurcations (the behaviour of a self-organized system changes dramatically at bifurcations).

In psychology, Gestalt is the analogy to the stimergy phenomenon.

### 5.12.1. Case of study: agents travelling

In this case, the goal was to achieve a complex system as vehicle coordination when customers want to use this service in an efficient way.

This problem shows too many parameters to control:

- location of every vehicle
- traffic jams because of :
    - spontaneous accidents
    - periodic events
- day/night traffic routines
- streets model
- cartography
- behaviour of every vehicle

The idea was to solve this problem isolating the parts that interact, building independent layers such as: street model or traffic information. Over the whole system an ACO algorithm was launched to prove suitable vehicle routes.

There are four basic files to implement the ACO model (see section 4.2.6) and Figure 56):

- ACOAgent: Represents the agent in the ACO model
- ACOModel: It is the model that implements the ACO algorithm
- ACONode: Represents the location where each agent can be.
- ACOEdge: Represents the distance among nodes.

The parameters and the performance of the model were redesigned to assure that there was an effective way for building solutions in a short time.

The following case was tested:

“Many agents moved in a known constrained space, for instance, a region of Barcelona (Spain) which is called ‘Eixample’. These agents do not directly communicate among them; they do it against a distributed global system that has information about all the agents. They trade in the city and can take the optimal route between the source and the destination. Moreover, the distributed global system also distributes the information.”

This case shows two relevant issues:

With simple rules is possible to achieve a system which **behaviour is complex and self-organized**. For instance, many vehicles can organize themselves in the city with simple rules. It is always possible to decide an optimal route for trading.

This system compared to economical systems and MAS represent the opposite side of 'selfish agents'. The agents cooperate using rules known by the community. Semantics are simple however the behaviour of the community is not. Thus, it is not necessary to have a greatest detailed semantics to achieve a complex model.

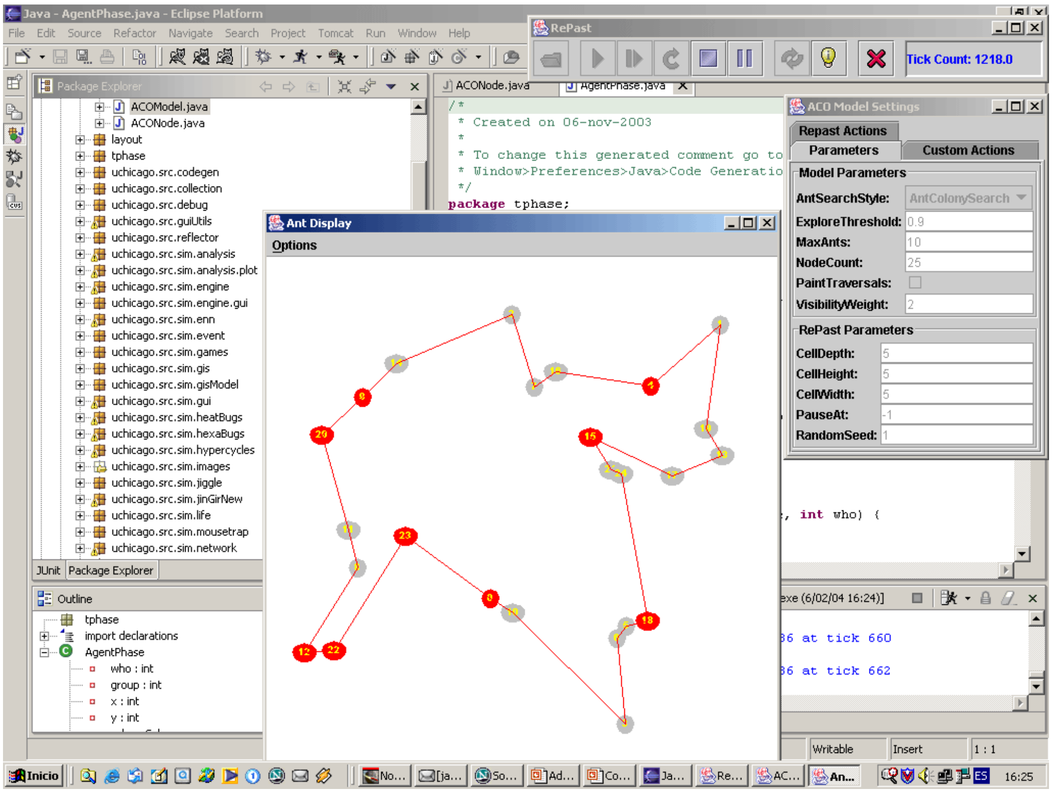



**Figure 56 ACO model implementation using Repast and Eclipse**

Negotiation processes cannot always be described for many reasons using Swarm Systems because agents don't use the same shared rules. Only is possible when agents react in the same way when environment is present (stimergy).

In the opposite side, macroeconomics models show similar features to Swarm Systems as: openness, long-term, short-term, fluctuations or aggregated demand [168], however it is not possible sometimes to assure that the reaction of the agents in front of an stimuli from the environment it will be the same. This model seems to be appropriate when many agents want to communicate using a consensus language and they know exactly what they are looking for. At that point two ways of researching were opened:

Try to model the situation when there are many agents and the environment is huge compare to agent number and the agents don't know 'prior to' what they are looking for.

How do artificial agents learn new things if they are not supplied by human beings?

## 5.13. Conclusions

Firstly, negotiation is conceptualised as a process where it is created by agents (Multi-Agent Systems-MAS) transferring each other material (multimedia content) in an environment (Intellectual Property Rights (IPR) context) following a protocol (based on a standard as FIPA).

An architecture to manage Intellectual Property Rights (IPR) is provided, at first, two versions were proposed (see Figure 34 and Figure 43) based in Semantic Web issues where every aspect is included, from customer to database design. This architecture was developed inside NewMARS [160] and AgentWeb [161] projects where negotiation of multimedia content is performed.

A negotiation model is developed inside these projects using the explained architecture. Contracts and licences have a semantic representation that has improved the negotiation process. In the context of IPR management is necessary to relate information ***dynamically***. It is possible because Expert System rules are used on-line in conjunction to Semantic Web infrastructure.

To come clear if it was necessary to perform complex rules to achieve complex system, Swarm Systems (another MAS) were analysed. From Swarm Systems, where the knowledge is formalized with simple rules is possible to create autonomous systems. The key in those systems is to be sure that every agent will behave in the same way when faced with the same event. For instance, these systems work very well in Telecom domain where packets follow deterministic rules.

Thus, we have artificial agents representing the user, and they have to make decision in the name of him/her. Then, Is complex Semantics necessary for solving negotiation process? Then answer is that with simple rules is possible to achieve a system *which behaviour is complex and self-organized*. Various systems have been analysed from 'selfish agents' coming from economics (see Introduction) and MAS (section 5) to Swarm Systems (also MAS but inspired in biology) agents that cooperate using rules known by the community. Semantics are simple however the behaviour of the community is not. Thus, it is not necessary to have a greatest detailed semantics to achieve a complex model. Swarm Systems provides a different approach nearer to database side than customer because users desires, beliefs … change quickly in time.

The agent also has to face a great amount of information using a tool for reasoning, e.g. an Expert System (see 4.2.5). However, they exhibit some issues that are not solved yet. In this case, management systems, as IPR management systems, present some problems when Expert Systems are used. For instance:

- It is difficult to manage a ***huge amount of rules (size problem)***.

Every country has its own IPR rules and cases. This means that the amount of information is enormous.

- Laws ***ambiguity*** when they are interpreted ***(ambiguity problem)***

  A standard or common point has to be found in order to establish a process as negotiation. Nowadays our research group is working very hard on this issue working at MPEG-21 standardization. There is a long experience working on standards.

- *Inference is not easy when it has to manage causality and completeness.*

  For instance, here there is an example of causality: ***(causality problem)***:

  INPUT

  "If Queen plays music, then Freddy Mercury sang"
  "If I put the Queen CD honouring Freddy Mercury, Queen will play music"

  OUTPUT

  "If I put the Queen CD honouring Freddy Mercury, then Freddy Mercury sang"

  It is not exactly true, because Freddy Mercury died between the two statements. So, the first is true in a different context than the second.

Section 6 faces the size and ambiguity problem while section 7 the causality problem.

# 6. Negotiation in a SWA: Ontology aspects

The goal is answering to section 5.13 questions about **size** and **ambiguity** problems in the SWA where we are working. In order to achieve this, there are several stages inside this section. To solve **size problem** it was necessary to understand the architecture of knowledge as a graph (see section 6.1). The base graph was built from the Semantic Web. Then, the graph was statistically analysed. The analysis produced a set of properties and its graph model (section 4.4). All this work is detailed in section 6.9.

**Ambiguity problem** has faced taking advantage of Semantic Web properties. The knowledge formalization has been made in such a a way that ambiguity problem could be minimized. That is, it has employed open knowledge representation formats taken from the Semantic Web initiative and already presented in section 4.5.

The IPR business model is the core and it is taken from the IPROnto formalisation (see section 6.3). As it has been already shown, IPROnto is an ontology designed to manage together parts 5 and 6 of MPEG-21 standard (see section 4.3). Before going into detail, it is necessary to recall the answer to the question what an ontology is.

Ontologies are the basic infrastructure of the Semantic Web. They are used to share vocabularies for describing resources content and capabilities. Their semantics are described in a (reasonably) unambiguous and machine-processable form. The ontologies task is describing this semantics, i.e. what is sometimes called the *intended meaning* of vocabulary terms.

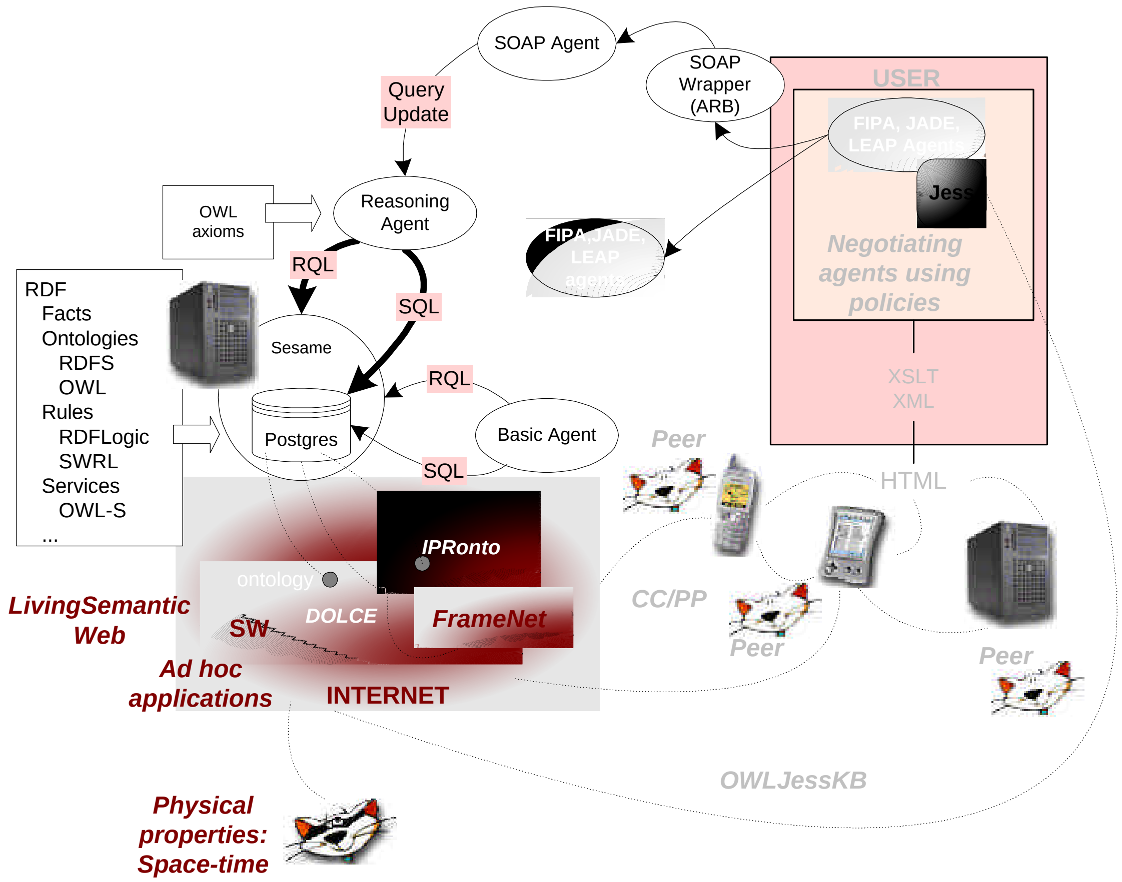



**Figure 57 Semantic Web Ontologies Architecture**

In most practical applications, ontologies appear as simple taxonomic structures of primitive or composite terms together with associated definitions. These are the so-called *lightweight* ontologies, used to represent semantic relationships among terms in order to facilitate *content-based access* to the World Wide Web data produced by a given community. In this case, the *intended meaning* of primitive terms is more or less known in advance by the members of such community. Hence, in this case, the role of ontologies is more than supporting *terminological services*, which are inferences based on relationships among terms. These relationships are usually just taxonomic rather and they do not explain or define the intended meaning.

On the other hand, however, the necessity to establishing precise agreements as the meaning of terms becomes crucial as soon as a community of users evolves, or multicultural and multilingual communities need to exchange data and services. This problem may have been "one of the main reasons that so many online market makers have foundered". The transactions they had viewed as simple and routine actually involved *many subtle distinctions in terminology and meaning*". To capture such subtle distinctions it is needed an explicit representation of the so-called *ontological commitments* about the meaning of terms, in order to remove terminological and conceptual ambiguities. A rigorous logical axiomatisation seems to be unavoidable in this case, as it accounts not only for the relationships between terms, but also for the formal structure of the domain to be represented. This allows one to use axiomatic ontologies not only to facilitate ***meaning negotiation*** *(see section 7, for more information)* among agents, but also to clarify and model the negotiation process itself, and in general the structure of interaction.

The quality of meaning negotiation may drastically affect the *trust* in a service offered by the Semantic Web, but not the computational performance of the service itself. Thus, for example, a product procurement process involving multiple agents with distributed lightweight ontologies may be carried out in an efficient way by using simple terminological services. However, the risk of *semantic mismatch* can be minimized only if the agents rely on explicit, axiomatic ontologies, which serve to ensure mutual compatibility of the respective models in such a way as to check the extent of real agreement.

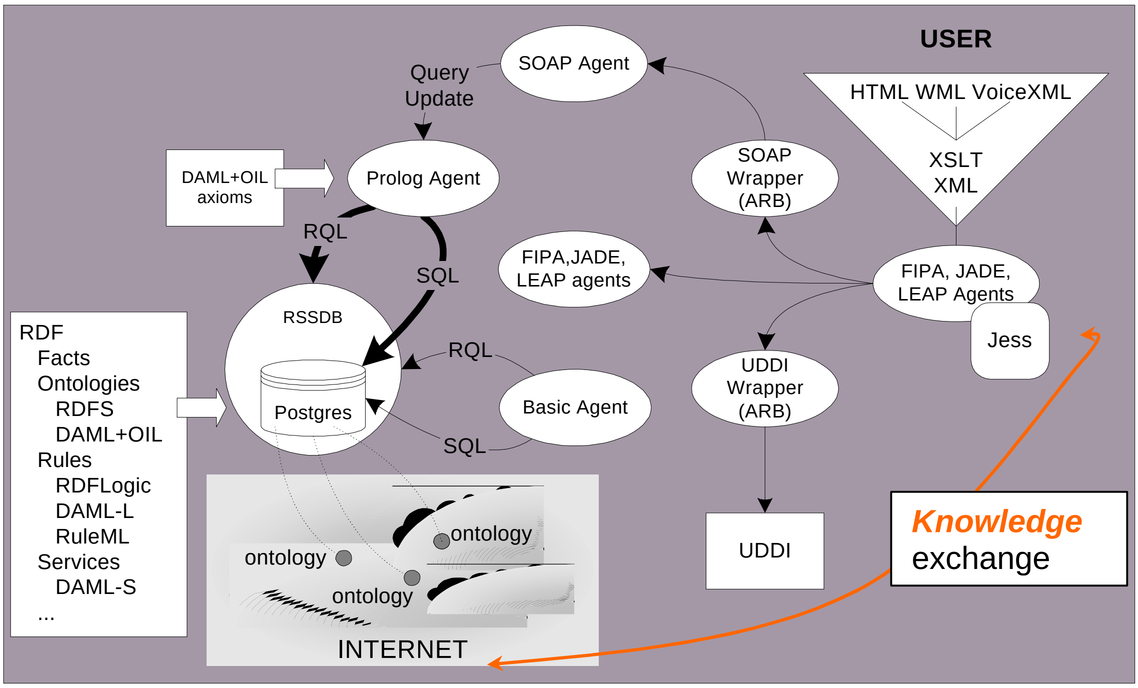



**Figure 58. Adding connections to Semantic Web Agents Architecture**

## 6.1. Semantic Web Advantages: network behaviour

When different physical entities begin processes, there are some transfers between them. In business affairs, these transfers are known as transactions. They are not specified, they can be whatever that can be related to an economic value.

Moreover, if an abstraction is made, entities could be represented as nodes and these transfers as relationships among the entities/nodes. This is a network. Networks are represented mathematically by graphs. This is the reason why they are used to make models and it seems a suitable way if the problem of knowledge size has to be solved.

However, the question is if there is a way to visualize the properties of a real system, in other words, a real process and extract information, even if it is not possible to know 'prior to' the conditions to be met. This is the reason why it is necessary to review something about the mathematical models.

For instance, trying to formalise an agent's policy as a set of negotion rules was a great challenge. Just, looking into the Expert System experience, the rules had to be defined before the negotiation performs and it was necessary to control every situation at last. This fact generated a problem of knowledge about what it is going to happen. Was it possible to know everything it was going to happen?

In order to answer this question, another question came before: Happen where?

As it is explained in the previous section, the negotiation process could be seen as agents communicating each other. In a highest degree of abstraction, it could be seen as nodes and relations. Semantics (*messages meaning*) and negotiation rules (*policy*) were provided by human beings. It seems that there is not a 'true' relation between agents and their environment. At last, so true as human being desires.

Then, how agents can take information about their environment?

One option is environment could be semantically noted, an example is semantic web where their components are semantically noted. However, as in WWW, information is not semantically noted. It seems that discover information is a difficult task. Somebody has to decide if data is knowledge or not. Usually, this job is done by human beings so the system will be so intelligent as human beings are!

Agents (technological agents) should have a mechanism to discover data, verify that this data contains knowledge in their criterion in a concrete environment.

The environment, as agents themselves, can be seen as a network. Internet or WWW are some examples. Thus, environment and agents can be seen as networks. Networks are analysed in mathematical terms. It is done because in scientific models always maths is a common language for communicating world vision.

A real **model to describe environment agents**, in other words, a real network that exhibits real properties has to be ***open and distributed*** because it is the only way to generate novelty. Closed systems where there is no communication among agents do not allow creativeness (as radical political systems where communication and expression is censured). The Semantic Web architecture pretends to allow both properties. In other words, to make a basis how a reliable agent **learning** could exist.

To accomplish the first point, an open and distributed environment needs to have an environment, i.e. Semantic Web must adapt to new challenges. It should have elements able to adapt. We call them agents.

This section goes deeper in the first point and explains how it is possible to achieve it. The second has to manage with the difficult question of how to make a consensus or agreement among agents and their internal knowledge representation (see section 6). How can I be sure that a concept has exactly the same meaning for the other agent?. Semantic Web tries to solve it, just allow anyone to put their own representation (his/her ontology) and to point it. However, language also has another component that is the internal model of the agent. The knowledge is interpreted in a different way depending on the agent. How this knowledge can be shared? The aim of this section is trying to go deeper in the way mathematics as a common basic language among human beings is learned to evaluate a way for provide an interface to a common understanding concepts in future sections. It can be found some paradoxes as: Everything or anything is almost the same [169]. Just thinking in the unity concept and their representation as '1': it can be interpreted as the whole ('God as the one') or as something more prosaic like a 1€ is!

## 6.2. Ontology creation: IPROnto

This ontology was build with the aim to improve the management of the IPR (Intellectual Property Rights). The goal was to provide a reference framework for IPR representation in the open and global framework provided by the Web where intellectual property rights can be managed in an open, global and adaptable form, so people can share, sell, buy, etc… content subject to intellectual property rights, depending on their needs. Concretely it was designed to solve Part 5 and Part 6 of MPEG-21 standard (see section 4.3) together.

A semantic approach seemed a flexible and efficient way of achieving these activities than a syntactic because IPROnto endowed agents with a background knowledge, which allows them to work quite autonomously.

The quality of meaning negotiation was accomplished because axiomatised ontologies were used, in this case, an upper ontology was chosen, concretely, IEEE SUO initiative was incorporated as the upper level and <indecs> framework as the core IPR specific part. WIPO [170] recommendations have been followed for specific legal aspects.

There are a lot of contributions trying to solve Digital Rights Management (DRM). Starting from isolated and proprietary initiatives but lately clearly moving to a web-broad application domain. An international approach that has already entered the standardisation phase is MPEG-21, our group is working on it [171] and moreover comparing to other languages as ODRL [172].

One of our research group lines have determined that, the use of a Semantic Web approach, and particularly web ontologies, is one optimal way to solve this problem in the WWW context.

The goal is to facilitate the automation and interoperability of IPR frameworks integrating both parts, called Rights Expression Language and Rights Data Dictionary in the context of the corresponding MPEG-21 call for proposals. This approach can be accomplished using ontologies. They can provide the required definitions of the rights expression language terms in a machine-readable form. Thus, from the automatic processing point of view, a more complete vision of the application domain is available and more sophisticated processes can be carried out.

Moreover, the modularity of web ontologies, constituted by concept and relation definitions openly referenceable, allows its free extension and adaptation without loosing the connection to previous roots. Once decided the methodology, the web-ontologies approach, the ontology creation process was initiated. However, it did not start from scratch.

Firstly, a clear definition of the IPR domain was searched in order to get a starting point. We decided *<indecs>*[173] was a suitable beginning because of its accuracy in describing this domain. We adopted the terms definitions and their structure.

However, this was not enough. If we want to provide a robust ontology basis we needed a more generic framework. SUMO [174] achieves this: it was born as an attempt to establish a standard upper ontology inside the Standard Upper Ontology (SUO) IEEE Working Group [175]. This characteristic was crucial to connect our work to other independently developed ontologies with the same basis.

SUMO has a top-most “Entity” class with two disjoint subtypes, “Abstract” and “Physical”. From these two initial distinctions, further refinements are done to conform the set of concepts that constitute the upper level ontology. Some of them are shown in Figure 59

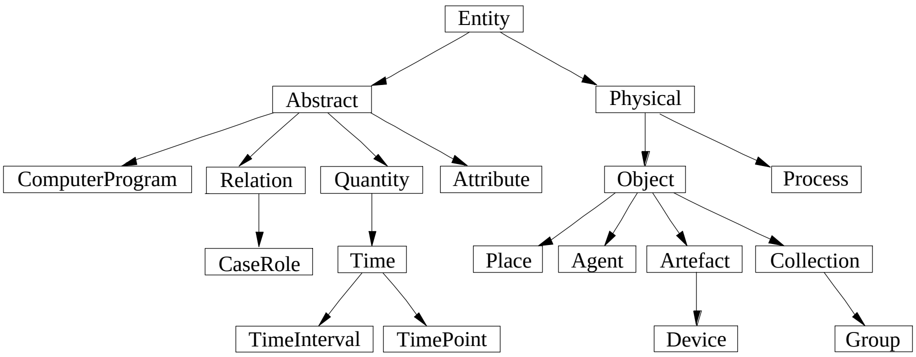



**Figure 59 SUMO Upper level (arrows relate superclasses to subclasses)**

World Intellectual Property Organisation (WIPO) is defining a common legal framework for IPR. It has been used to complete legal part of the ontology.

IPROnto [176] is a group contribution, and it is composed of the parts described in the previous lines plus other work that is described in the next section. This proposal has been developed from previous IPR related work of our group, the DMAG [177]. It ranges from security [178] to automatic negotiation using agents [179], [180], [181] and the application of a semantic approach [182].

**Table 30. Rights hierarchy**

| | |
|---|---|
| Abstract | |
| LegalConcept | * Detail: |
| ExclusiveRight | Copyright |
| IntellectualPropertyRight |     MoralRight |
| NeighbouringRight |         DisseminationRight |
| SuiGenerisRight |         PaternityRight |
| ExceptionsRight |         RespectRight |
| CiteRight |         WithdrawalRight |
| LibrariesRight |     ExploitationRight |
| PrivateCopyRight |         TransformationRight |
| Copyright* |             AdaptationRight |
| |             TranslationRight |
| |                 SubtitlingRight |
| |         CommunicationRight |

| | |
|---|---|
| | BroadcastRight |
| | PublicPerformanceRight |
| | DistributionRight |
| | RentalRight |
| | ReproductionRight |

Besides these legal aspects, other extensions have been faced to provide a really operative ontology. First of all, the intellectual property life cycle and the different roles of the parties involved have been modelled. This generic business model has been inspired in the one defined by the Imprimatur project [183]. The different roles, represented as ellipsis, and the actions they perform, the transitions, are shown in

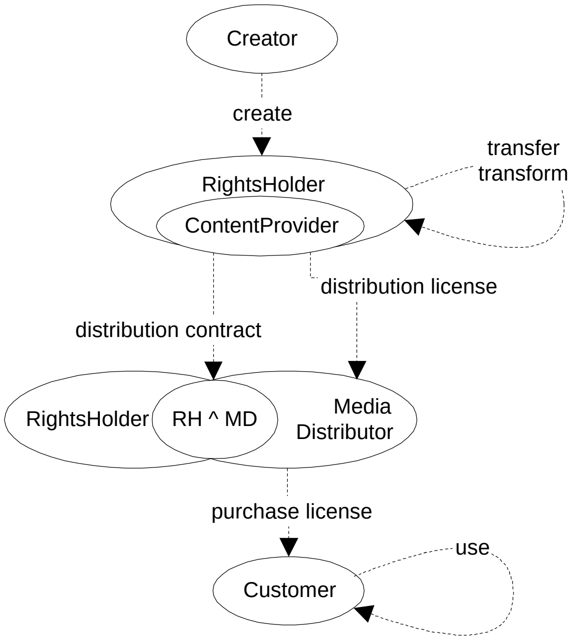



**Figure 60 Content life cycle**

In Figure 60, the first involved role is the ***Creator***, who creates a creation. This event corresponds to the arrow labelled with “create”. Automatically, there is someone that becomes the ***RightsHolder***, which could be the creator or not. From this point the creation can be transferred or transformed changing the rights owner. The ***ContentProvider***, who is a ***RightsHolder*** and also maintains the contents, initiated the distribution of the latter. This is performed by the ***MediaDistributor***, who can hold rights or not depending on the kind of contract agreed with the ***ContentProvider***, a distribution contract in the first case and a distribution license in the latter. Finally, the Customer receives the permission to use the content formalised in a purchase license given by the ***MediaDistributor***.

The role’s actions are modelled as *events*. Thus, we take an event-centred approach, already considered by <indecs>, as the way the ontology is used. The complete set of events is detailed in Figure 61.

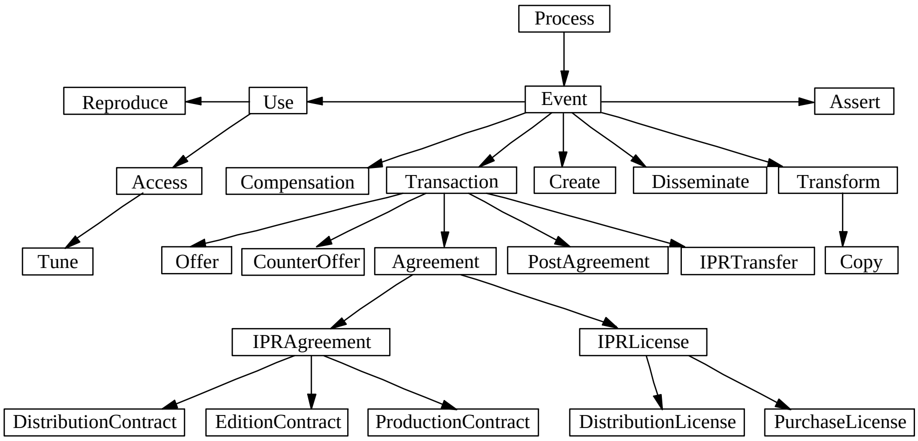



**Figure 61 Events hierarchy (arrows relate super classes to subclasses)**

Events are the main concepts when using the ontology to make IPR statements. They are connected to the other involved concepts using properties, as defined in RDF specification. Currently, these event properties, the roles in Figure 62, are structured following the <indecs> basis and extended to cope with the whole range of events incorporated into IPROnto.

The semantics of these events is represented with restrictions associated to the properties that relate them to the rest of the action pieces.

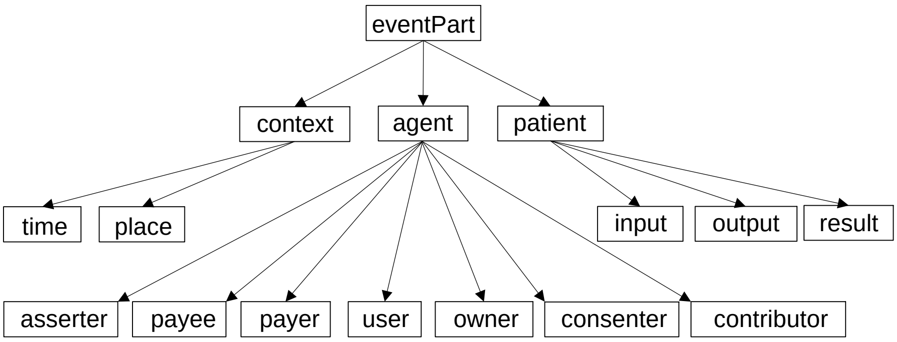



**Figure 62 Roles hierarchy (arrows relate superproperties to subproperties)**

IPR management requires an exhaustive exam of involved processes concerning their legal aspects. Sometimes, technical issues are very similar, but behind them we could find legal differences. Ontology is a tool that enables us to represent and solve them when you are working at an automatic environment. In the next subsection we are going to explain an example, where two situations are presented with technical points of view that are very similar, but not their legal aspects. This is an example of ontology accuracy.

To show a broader view of ontology functionality, a negotiation scenario will be solved using our ontology as a basis. In addition, we would see how to integrate it with agents' technology.

Technology issues could be considered the same or almost the same in some cases, as we will see in the next lines, listening a CD or tuning a radio program.

However, behind this, there is a legal part that can differ from the technological point of view. Consequently, a great event detail becomes necessary as well as adaptability and flexibility. These are aspects present in our model as it is going to be shown.
Imagine the following situation: we are running a media player program and we press the “Play” button. However, it is not a real button, so, are we doing a play? Are we running a program?

We can combine ontology elements to get the desired level of detail. In this case, we have the following meaning of the concept “use”:

- Reproduce: the user employs a physical manifestation of a creation, or one of its copies.
- Access: when one makes use of a creation during a specific time, i.e. communication to the public such as a performance or a broadcast.

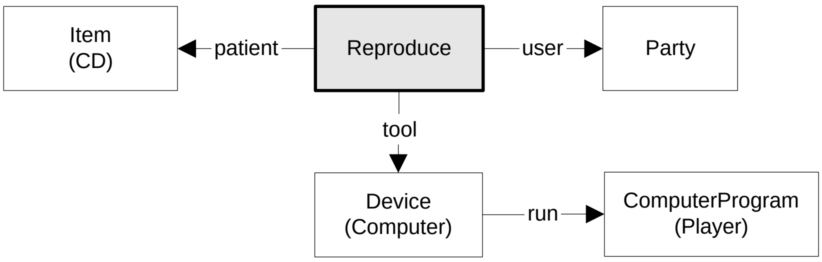



**Figure 63 Playing a CD in a computer**

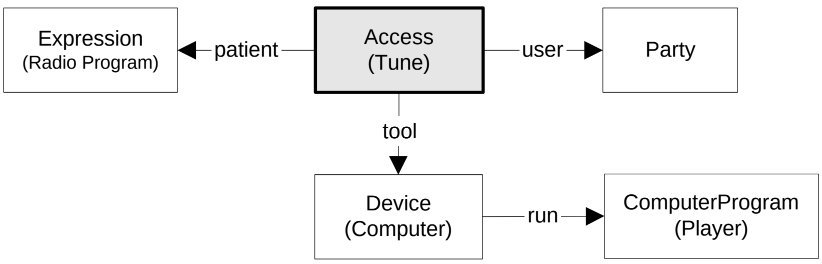



**Figure 64 Streaming, diffusion on-line on-demand (or Internet Radio, tune)**

An example of the first type of use could be the one shown in Figure 63, where a physical form of a creation is involved. The second type of use can be exemplified by a computer program acting as a tuner of an Internet radio or reproducing on-line on demand, i.e. an audio streaming, see Figure 64.

We can see both actions as a play, but they involve different kinds of creations. Therefore, we must distinguish both cases, as different intellectual property rights might be involved.

# 6.3. Relation of IPROnto to other initiatives

IPROnto was developed following a completely different approach from other initiatives. As it has been mentioned in section 6.2, this ontology was born to solve two MPEG-21 standard parts: Part 5 (REL- Rights Expression Language) and Part 6 (RDD- Rights Data Dictionary) (see section 4.3).

Its difference was not only that was expressed in a different language as RDF and DAML (see section 4.5.1) from the other initiatives that were presented using XML moreover it gives solution to two parts at once.

Summarizing, it was done in this way to accomplish a solution that is:

- ***Powerful***. This was the reason for choosing a web ontology and Semantic Web standard languages (see section 4.5)
- ***Open*** to other initiatives. Ontologies allow relating the connections more efficiently than schemas (see section 4.5.1 to see XML and RDF differences).
- ***Minimal***. Perhaps it is the main feature, because Part 5 and Part 6 are the sides of the same coin. A language is completely related to its dictionary, they are not independent things. They need each other. IPROnto allows to extract REL and RDD independently if it is the case.

Following this lines there is an explanation of how RDD has to be managed as well as the structure summarized in Table 31.

**Table 31 MPEG-21 standard: RDD (Rights Data Dictionary)**

| ActType | Parent(s) | Definition | Comments (Informative) |
|---|---|---|---|
| Adapt | Derive, ChangeTransiently | To ChangeTransiently an existing Resource to Derive a new Resource. | With **Adapt,** two distinct **Resources** will exist as a result of the process, one of which is the original **Resource** in unchanged form, and one of which is newly made. Changes can include the addition to and removal of elements of the original **Resource**, including the **Embedding** of other **Resources**. Changes can be made temporarily to the original resource in the course of the **Adapt** process, but such changes are not saved in the original **Resource** at the end of the process. |

| ActType | Parent(s) | Definition | Comments (Informative) |
|---|---|---|---|
| | | | Specializations of **Adapt** can be differentiated by specific attributes of the **Resource** which are preserved or changed. The specific attributes can be on a list or can be called out by using a list. Lists can be inclusive (for example, "Attributes a and b must be changed") or exclusive (for example, "Everything except attributes c and d must be changed"). Attributes that are not constrained in specializations can be changed. Most ActTypes that are generally known as "copying" may be represented in the RDD Dictionary as children of Adapt. In most domains "copy" typically means to Derive a new Resource which has the same set of specified or implied attributes as its Source, a common example being the "copying" of a Digital Object. However, the concept of "sameness" is not to be confused with that of identity, as two things cannot technically be "identical" because at the very least they will have different spatial or temporal attributes (that is, they will be located in a different place, or created at a different time), and so a "copy" with absolutely identical attributes to the original cannot logically exist. Particular interpretations of "copy" can be defined as specializations of Adapt [for further explanation see Annex D]. |
| Delete | Destroy | **To** Destroy **a** DigitalResource**.** | Delete applies only to DigitalResources. Delete is not capable of reversal. After a Delete process, an |

| ActType | Parent(s) | Definition | Comments (Informative) |
|---|---|---|---|
| | | | “undelete” action is impossible. |
| Diminish | Adapt | To Derive a new Resource which is smaller than its Source. | With **Diminish**, two distinct **Resources** will exist at the end of the process, one of which is the original **Resource** in unchanged form, and one of which is newly made, whose content is **Adapted** from the original **Resource,** and a Measure of which is smaller than that of the original while no Measures of it are larger. Changes can include the removal of elements of the original **Resource**. Changes can be made temporarily to the original **Resource** in the course of the **Diminish** process, but such changes are not saved in the original **Resource** at the end of the process. |
| Embed | Relate | To put a Resource into another Resource. | The **Resource** into which a **Resource** is **Embedded** can be pre-existing or can be created by the act of combining the **EmbeddedResource** with one or more others. **Embed** refers only to the embedding of an existing **Resource** in another: if a “copy” of an existing **Resource** is to be created and **Embedded** in another, then both **Adapt** and **Embed** would be used. |
| Enhance | Adapt | To Derive a new Resource which is larger than its Source. | With **Enhance**, two distinct **Resources** will exist at the end of the process, one of which is the original **Resource** in unchanged form, and one of which is newly made, whose content is **Adapted** from the original **Resource,** and a Measure of which is larger than that of the original while no Measures of it are smaller. Changes can include the addition of elements to the |

| ActType | Parent(s) | Definition | Comments (Informative) |
|---|---|---|---|
| | | | original **Resource**, including the **Embedding** of other **Resources**. Changes can be made temporarily to the original **Resource** in the course of the **Enhance** process, but such changes are not saved in the original **Resource** at the end of the process. |
| Enlarge | Modify | To Modify a Resource by adding to it. | With Enlarge, a single Resource is preserved at the end of the process. Changes can include the addition of new material, including the Embedding of other Resources, but not the changing or removal of existing elements of the original Resource. |
| Execute | Activate | To execute a DigitalResource. | Execute refers to the primitive computing process of executing. Execute applies only to a DigitalResource. |
| Install | UseTool | To follow the instructions provided by an InstallingResource. | An InstallingResource is a Resource that provides instructions which when followed result in one or more Resources that are new, or Enabled, or both new and Enabled. |
| Modify | Change | To Change a Resource, preserving the alterations made. | With Modify, a single Resource is preserved at the end of the process (that is, no additional Resource(s) come into existence). Changes can include the addition to and removal of elements of the original Resource, including the Embedding of other Resources within it. Specializations of Modify can be differentiated by specific attributes of the Resource being preserved or changed. The specific attributes can be on a list or can be called out by using a list. Lists can be inclusive (for example, “Attributes a and b must be |

| ActType | Parent(s) | Definition | Comments (Informative) |
|---|---|---|---|
| | | | changed") or exclusive (for example, "Everything except attributes c and d must be changed"). Attributes that are not constrained in specializations can be changed. |
| Move | Modify | To relocate a Resource from one Place to another. | With Move, at least the location of the Resource is Changed. |
| Play | Render, Perform | To Derive a Transient and directly Perceivable representation of a Resource. | Play covers the making of any forms of Transient representation that can be Perceived directly (that is, without any intermediary process) with at least one of the five human senses. Play includes playing a video or audio clip, displaying an image or text document, or creating Transient representations that can be touched, or Perceived to be touched. When Play is applied to a DigitalResource, content can be rendered in any order or sequence according to the technical constraints of the DigitalResource and renderer. |
| Print | Render, Fix | To Derive a Fixed and directly Perceivable representation of a Resource. | Print refers to the making of a Fixed physical representation, such as a hard-copy print of an image or text, that can be Perceived directly (that is, without any intermediary process) with one or more of the five human senses. |
| Reduce | Modify | To Modify a Resource by taking away from it. | With Reduce, a single Resource is preserved at the end of the process. Changes can include only the removal of existing elements of the original Resource. |
| Uninstall | UseTool | To follow the instructions provided by an UninstallingResource. | An UninstallingResource is a Resource that provides instructions which when followed result in one or more Resources that had previously been Installed |

| ActType | Parent(s) | Definition | Comments (Informative) |
|---|---|---|---|
| | | | being Disabled or Destroyed. |

The Standardized Terms in Table 31 are specifically defined to support the REL as defined in ISO/IEC 21000-5 and provide the foundation of the RDD Dictionary. New terms, developed specifically to support REL requirements independently from mappings from other schemes, can be added to the RDD Dictionary through the registration of such Terms with the Registration Authority. Once new Terms have been added to the RDD Dictionary, they may be used explicitly in REL expressions, or they may be translated into appropriate REL expressions. The process is therefore flexible, capable both of supporting the REL directly and of providing a means by which it can be supported in future by the addition of Terms from external schemes, thus providing for interoperability between different Authorities.

Great care should be taken in the use of RDD Dictionary Terms in any specific environment or application in order to avoid unintended consequences. As a closed ontology, all RDD Dictionary Terms are defined with reference to other RDD Dictionary Terms. This has two main consequences for the understanding of an RDD Dictionary term when it is used in an REL license. The first is that no assumptions should be made about the meaning of a Term based on the coincidence that it bears the same name as something in an application domain. For example, the words “Play” and “Print” are common in applications and terminals, and they have many shades of meaning. The RDD Standardized Terms “Play” and “Print” mean only what they are defined to mean in this part of ISO/IEC 21000. The RDD Dictionary meanings of “Play” and “Print” may or may not correspond to the meanings attached to the words “play” and “print” in other domains. Words used as the names of Terms are only convenient labels: mapping is achieved by analysis of the defined meanings of Terms, irrespective of their names.

The second consequence concerns the inheritance of meaning. As the RDD Dictionary is a hierarchical ontology, most of the meaning of a Term is inherited from its parent(s) (in RDD Dictionary terminology, its “Archetypes”). Because of this, if an REL license contains a Right to a ***StandardizedActType*** (for example, “Modify”), then the holder of the license will also have all Rights for which Modify is the sole parent – that is, “Move”, “Enlarge” and “Reduce” – even though these are not explicit in the license. On the other hand, if a term has more than one parent, it is not wholly included in each. So, for example, if an REL license contains a Right to “Adapt”, it does not include the Right to “Play” or “Print”, because Adapt is only one of the parents of these Terms.

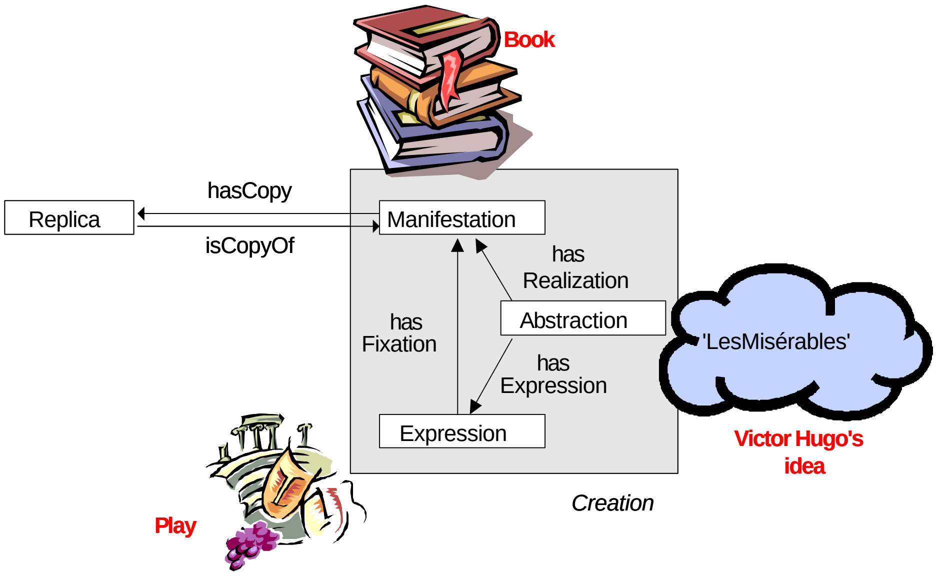



**Figure 65. IPROnto Creation concept**

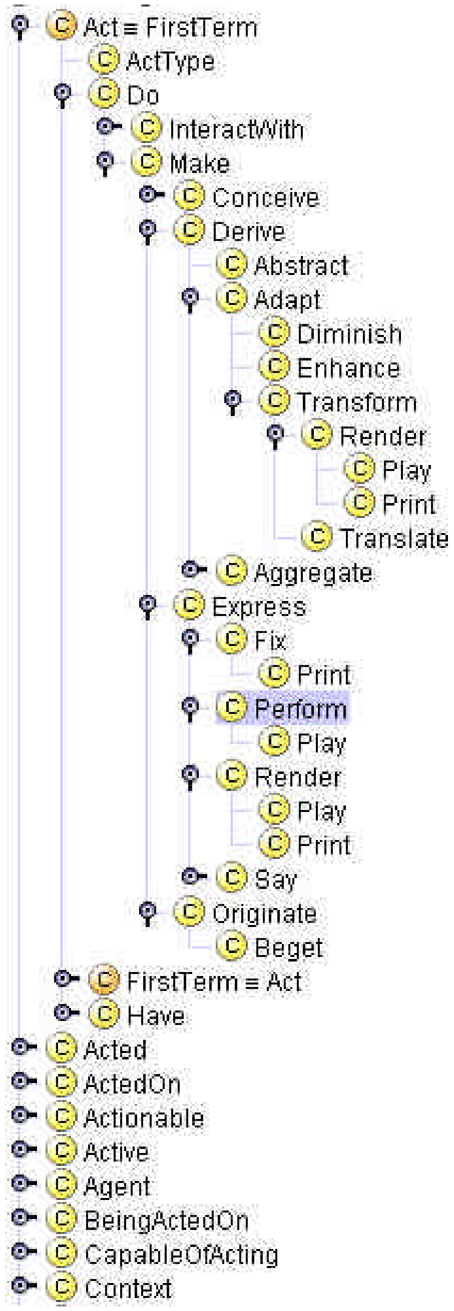



**Figure 66 Partial REL structure in Protégé**

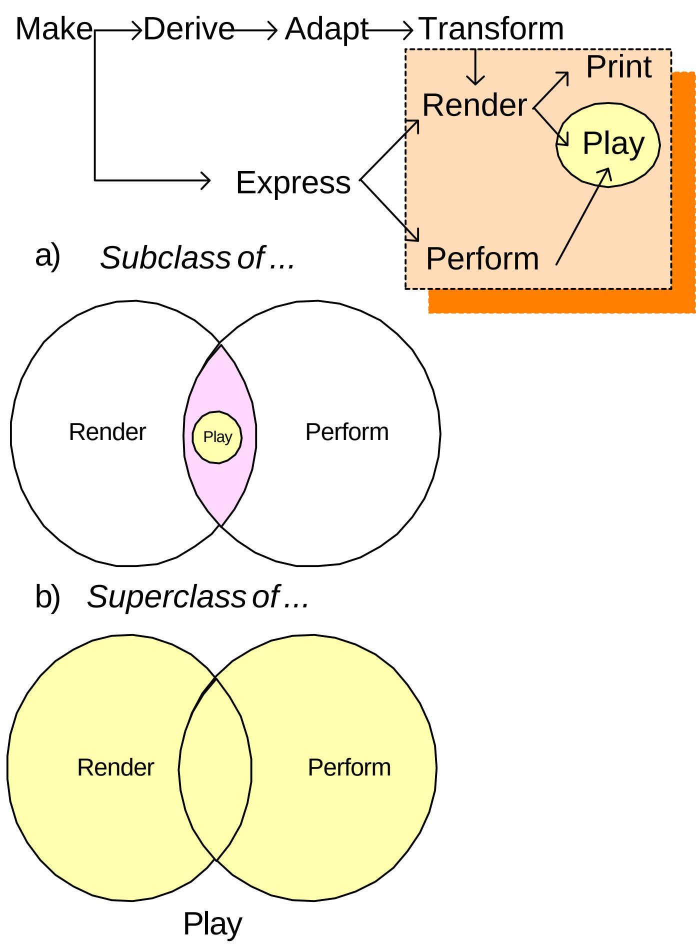



**Figure 67 'Play' interpretation inside RDD compared to REL**

As an example of how useful can be a minimal representation Figure 66 show REL structure while Figure 67 show the difference between the RDD definition and the same difference in REL structure. In other words, there is an inconsistency inside this standard because RDD and REL are independent structures. If a standard is developed as a graph, the structure and the relations are maintained and it is easier to avoid inconsistencies.

## 6.4. Ontologies meet Ad hoc applications

IPR issues need applications to be able to manage them. One of the first goals was to establish how ontologies could help ad hoc applications. Ad hoc applications need an open framework to become real applications for service providers. It is easier to provide them when the problem is constrained and the environment is closed.

This previous feature implies that sometimes, there is ***no prior*** information about new ad hoc application components going into our environment. However, if new application components are described semantically and related to an ontology, this problem can be solved. Even if the new component uses an unknown ontology, there is an opportunity to find information as we are going to see. Well-defined ontologies are always related to 'upper ontologies', ontologies of high abstraction level. They can be viewed as something common between the ontology that describes our application and the unknown ontology.
For example, a concept named "pressureEvent" does not say too much. However if it is related to "Event" concept of an upper ontology, it is described in a high level and can be connected to another concept depending on "Event" such "temperatureEvent" how it is useful to use semantics and which representations and languages are more suitable to describe component applications. Following, it is explained how statistical analysis is going to be used. A model and an example, ***weather forecast***, are used to validate it. Semantics and complex systems analysis are the pillars of this example telecom application. In order to build services the CoCTelS[194] platform is used as a pilot prove to IPR management.

## 6.5. Improving Ad hoc applications using ontologies

Applications in open environments where there is no prior knowledge are developed based on existing components. These components define the application. When the components are usually renewed because they are required or the execution environment evolves, it is necessary to find a dynamic way to get valuable aspects as autonomy and mobility. Thus, we are describing just the situation that will be held on the next telecom generation.

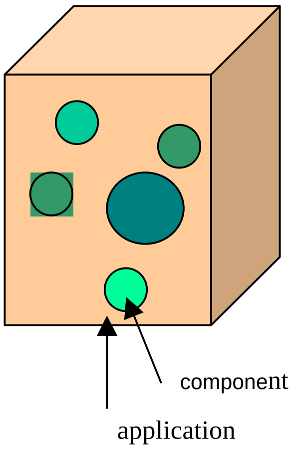



**Figure 68 Classical application design constrained by predefined components**

First, a review of the current situation is needed. There is an application based on known components, as shown in Figure 68, where it is necessary to know not only the components but also their operators and functions. This is the required basis in order to be able to connect them and keep our application going on. This is also the basis of systems like EAI and BDI, which are going to be explained in section 6.6.

This approach does not fit with next telecom requirements, so a new model must be developed to accomplish them, we can see an overview at Figure 69, where the application is built in a dynamical and self-organizing way, moreover there is also a negotiation with service providers to get the most appropriate component every time is needed.

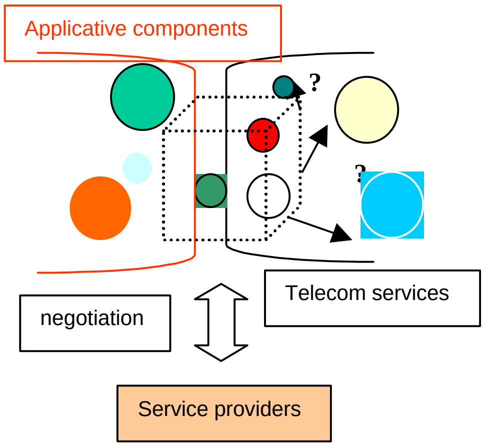



**Figure 69 Ad hoc application design**

Indeed, each current configuration management schemes consist of two phases: a configuration phase and a transmission phase. In wireless ad hoc networks, the **topology** is so unstable that by the time the transmission phase can utilize the established configuration, it is already obsolete due to node mobility. Hence, existing approaches to configuration management do not work for ad hoc networks.[184]

Another relevant question was where is the information?

- Is it hard coded and buried in each application?
- Is it in a common place where everybody can go and look for?
- Is it distributed along the network?

The next challenge is to establish how to manage information in ad hoc applications. Ontologies come to help us in this issue.

## 6.6. Without 'prior' knowledge

There are some initiatives proposed as EAI (Enterprise application integration)[185]. BPI (Business Process Improvement [186]) is at the top of the EAI abstract level. It is thought to integrate components as it is shown in Figure 68. In order words, it creates a limited environment where collaboration is made without hierarchy and components know each other. Therefore, these models are limited by a priori knowledge and thus they are not really dynamic.

EAI market originated with the installation of enterprise resource planning (ERP) systems on a wide scale in the early 1990s. Companies once used client/server technology to build departmental applications, but later realized the gains in linking multiple business processes. Enterprises built distributed computing environments; only to find a competitive advantage in expanding those applications to include external business partners but nowadays with Web projects, a new strategy has to be developed. The size and the continuous change in the web topology are the most important issues to be solved.

EAI Market Segmentation could be divided into five hierarchical categories of integration.

*Abstract level*

**Business Process Integration**
Process of workflow integration
**Application integration**
Data translation and transformation; rules base routing; pre-programmed adapters; monitoring and management tools
**Component Integration**
Application Servers
**Data Integration**
Tools for extracting, transforming and loading data
**Platform Integration**
Messaging

Nowadays, the solution taken in EAI is to design global packages and to customize depending of the user necessities. They use as background technologies great applications servers, which usually are multi-component, cross-platform application servers and are designed for building applications and web services with J2EE, PowerBuilder, XML or CORBA.

Agents platforms have been integrated in some applications servers as BlueJADE project in HP (BlueJADE project[187], where JADE-LEAP (JADE-LEAP platform, see section 1.) agent platform has been added to application server (JBoss[188]).

Other interesting purpose is XMI (OMG management group web site[189]), which enables easy interchange of metadata between modelling tools (based on the

OMG UML) and between tools and metadata repositories (OMG MOF based) in distributed heterogeneous environments. XMI integrates three key industry standards:

XML - eXtensible Markup Language, a W3C standard
UML - Unified Modelling Language, an OMG modelling standard
MOF - Meta Object Facility and OMG modelling and metadata repository standard.

Thus, there is a tendency to :

- integrate services and applications over networks (as the Web is)
- use semantic representation for metadata, for instance XML is widespread over applications servers.
- integrate different technologies as web services and applications using agents' technologies.

These concepts have to face to the fact that every day the amount of components grows not only in number if not in quality. Thus, these existing technologies provide a great basis to build a new ways to provide a more specific analysis as we are going to see, first of all adding semantics and complex systems analysis as a second phase.

Enterprises are using XML technology as a basis but it is not the only one and it is not the most effective, many propositions are being made in the last times, see section 4.5.

Web is an important content and service provider. Telecom service providers should not forget it [190].

## 6.7. Applications components discovery and negotiation

Agents as a technological concept seem to be the natural approach to realize components discovery and negotiation. It is a crucial point to achieve dynamical behaviour because we need something that can take decisions, act autonomously and of course making a deal, negotiating at last in the most effective way.

Some work will be showed in the model description, where agents negotiate multimedia content with service providers using expert systems and semantics to achieve this in a dynamical and effective way. However, a good mechanism to perform discovery was missing. It is the reason why we look at complex systems. They provide a quick way of getting information from huge networks. Moreover, properties can be extracted and analyzed in order to take decisions that maintain feedback in the system.

Many enterprises are working to provide web services platforms. Some of them are proprietary but others provide open environments, for instance JWSDP[191].

APIs are being developed to integrate services, not only web services but also EJB (Enterprise Java Beans) with XML. They provide interaction interfaces between clients, containers, components and concentrators.

# 6.8. Ad hoc applications integrate into the Semantic architecture

The previous sections have showed the difference between XML and RDF. Moreover, it has been explained why using graphs can be a great advantage. However, this is only the beginning. Figure 70 shows a model schema that is going to be described.

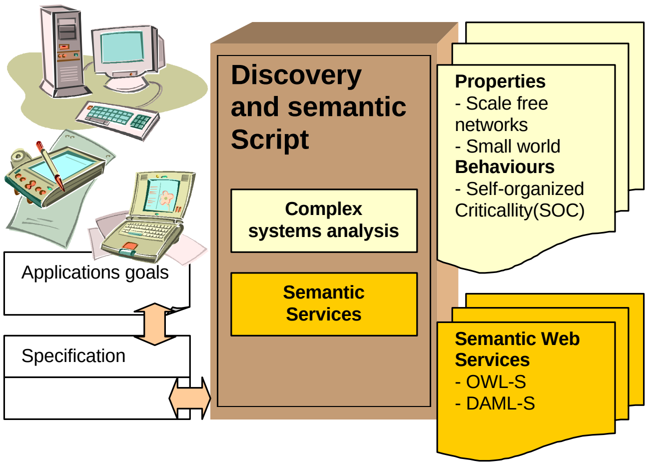



**Figure 70. New application model based on complex systems analysis and semantics**

It is supposed to show some components that are distributed all over a large network. The objective is to discover the required components that will build distributed applications.

The premises are:

- The network is huge. However, we have seen that there is an interesting behavior that can be characterized, see section 4.4.
- We know application components semantics but we do not know prior information about exactly where they are.

The goal is to build an application that makes a ***weather forecast***. At first, as we have explained in the premises, we do not know where we can find the components to build this application, perhaps because a resource has disappeared or we have lost the information we had in the past.

Software engineering provides interesting solutions to assemble software components according to proven methodologies and technologies [192,193]. We are going to work over a composition platform for telecom services. They are implemented by applications built from distributed components. The platform CoCTelS [194] offers to each user an environment which allows him to select services he is interested in.

Bouquets are composed of elementary components and the connector-factory generates the connector which is an abstraction for component interactions, see Figure 71. Each user has his own connector being able to call shared services and aspects. Hence, the connector-factory should create connectors as small as possible to ease their storage, their activation induce minimal resource consumption when executed. The synchronization of the accesses to the elementary services has to be managed as many connectors may invoke the same service at the same time. They are still managed as before.

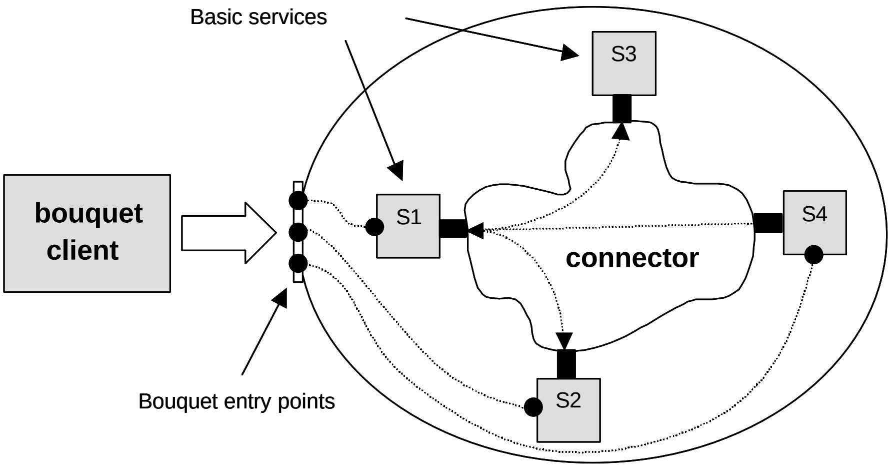



**Figure 71. The bouquet and its entry points**

The connector-factory will base his generation according to the description of the elementary services and aspects which are selected by the client. Two points of views have to be considered: the static point of view and the dynamic point of view. The static point of view deals with the connection links between the elementary components of the bouquet of services. The dynamic point of view deals with the management of the dynamic aspects of the collaboration between the components participating in the virtual bouquet. For example, when a client asks his bouquet to load a video mail, the connector has to call the mail service, get the reference of the video and its description and then, according to network resources availability propose, to the caller the video or to select an alternative among rendering degradation, or getting notified later when the network will offer enough resources to assure enough quality.

Adding values to the services is a factor of satisfaction to the users and of income to the operators. When helping the client to configure his virtual bouquet of services, the bunch manager has to propose, when appropriate, the aspects which may potentially be composed with the services already retained by the user. For this purpose, the bunch manager has to roam the entry points of the description of the service and whenever available, propose potential aspects to the user. If the user retains an aspect, the composition of the aspect is done the same way as between primary level services and secondary or system level services.

The bunch manager now has all the elements to generate the description file of the connector: a path in the ontology. It has at its disposal the services and the aspects which will compose the virtual bouquet, as well as the invocation sequence for each potential invocation. The connector description file generated by the bunch manager specifies trees whose roots are the high level services. The first leaves are the high level methods. Each high level method is associated to one or many aspects and has an execution tree to help the connector factory generate the right sequence of calls. They are the paths that we are going to build in the ontologies.

The first stage of connector-factory generation consists in getting the information concerning the high level services, the high level methods, the aspects and the processed calls in the execution tree. Then the factory has to get the associated technical information: localization of the service, the invocation protocol, the invocation parameters, pre and post conditions. Then, it generates the source code of the connector directly based on the execution tree. Afterwards, the connector-factory will look up in the connector description file the rules that have one of the methods in their left side. These rules will be applied to the virtual bouquet of service that is being considered.

### 6.8.1. Example of Ad hoc application: Weather forecast

If we use semantics, we can fix an entry point: a weather ontology – for example we can take this one [195], described in RDF, DAML or OWL, in other words: a graph. In this case it is useful to go to an ontology library [196]

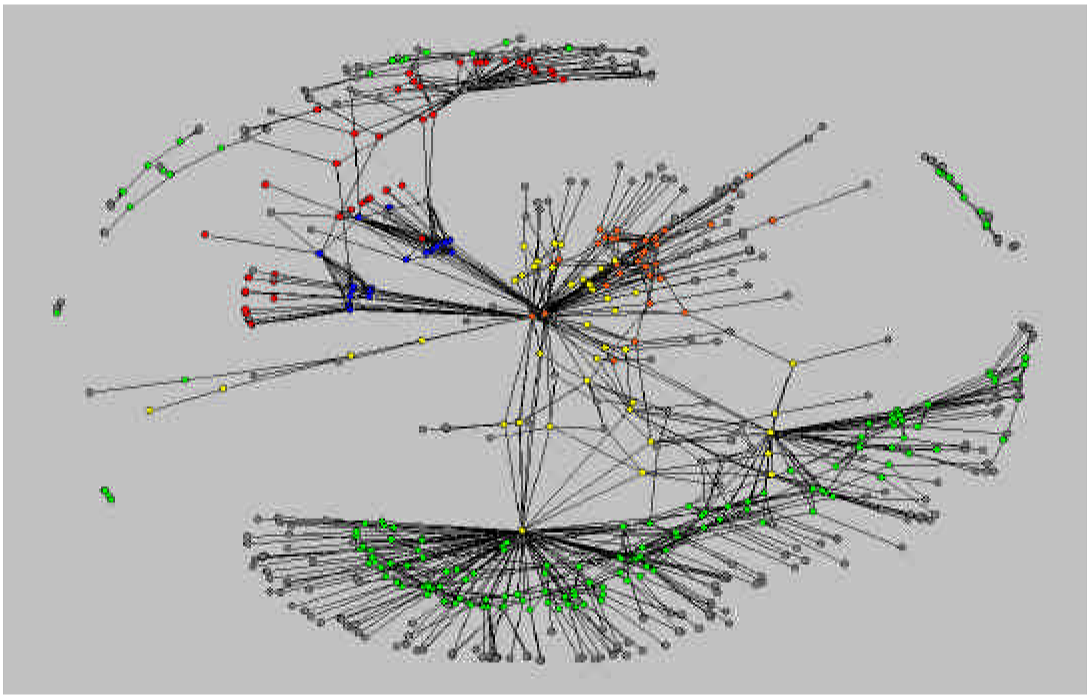


**Figure 72. Network analysis: Green Weather Ontology, Yellow DAML, Red Dublin Core, Orange RDF/S, Grey anonymous resources and literals**

The key point is to make ontologies related to others or to an upper ontology, and this is relevant when we represent it as we can see in Figure 72.

Pajek [197], a large network analysis tool, was selected. The RDF N-Triples serialization was translated to a '.net' Pajek network file. The triples subjects and objects became network nodes connected by directed edges from subject to object. Nodes are identified by their original URIs to allow network construction and the edges are unnamed so duplicated edges are ignored.

An analysis is performed over the graph to locate concepts that can describe applications components, for instance, time. There are some ontologies related to time, we have taken one [198] and a GIS ontology [199]. In fact, ontologies are chosen dynamically depending on user profile (it is a graph and it describes customer preferences). The graph is showed in Figure 73.

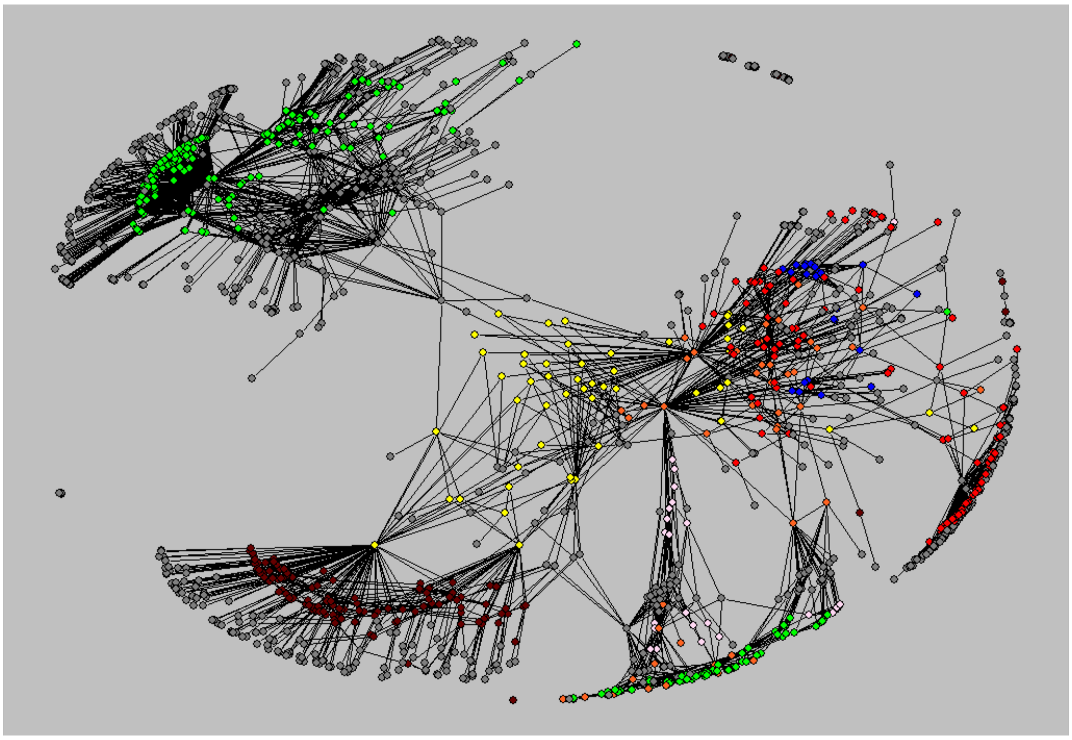


**Figure 73. Network drawing: Green ISI, Red Dublin Core, Blue purl, Yellow DAML+OIL, Orange RDF/S, Brown Mindswap, Pink Map-ont**

In this way, we can discover from an ontology, which is describing a service, connections to other components in other ontologies and discover a path to find new information. Data extracted is summarized in the final part of this section.

Moreover, it is possible to discover new information from the topology. The network analysis allows us to locate 'hubs', in other words nodes over the network that have many relations. They are good candidates to be important resources. Also it is possible to see clustering belonging to small world phenomenon as it was explained in section 4.4.2.)

For each bouquet of service being composed, new paths are generated according to the elementary services which already compose the bouquet and its entry points. In order to simplify proof obligations, we assume that we should only verify the correctness all along execution paths and then consider states to be correct only if they belong to a valid execution trace. A trace ends when it reaches a stable state or an illegal state (component's input precondition is not satisfied when it is activated).

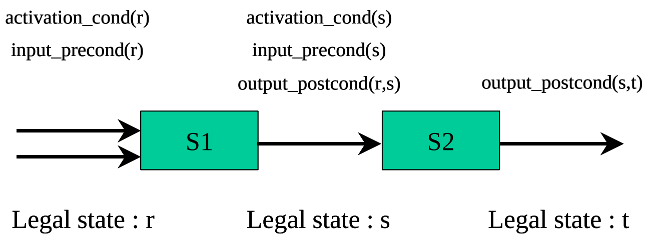



**Figure 74. Activation obligation example**

Semantics are done using DAML-S or OWL-S, to represent telecom services and design our connector because as we can see in Figure 74 and Figure 75, DAML-S has been used to describe services and the structure. In order to check properties in the calling paths, constraints in DAML-S allow inferring new properties.

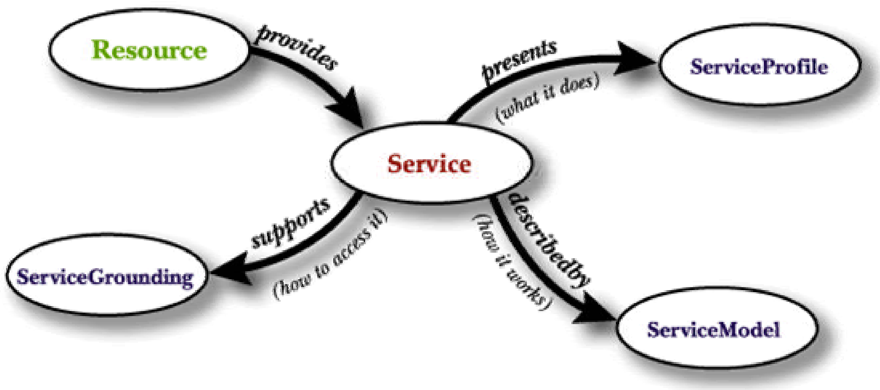



**Figure 75. DAML-S Upper Ontology (image from Terry Paine)**

Going beyond, in our example, for instance, at first describing the service, in DAML-S, it can be seen in Figure 75, an upper ontology for describing services. A service presents what it does in the Service Profile where are defined: input types, output types, pre-conditions and post-conditions as we have seen to be necessary for telecom platform in Figure 74. For example see Table 32:

**Table 32. Weather Project Service Profile**

```
<profile:input>
     <profile:ParameterDescription rdf:ID=”Date”>
               <profile:parameterName>Date</profile:parameterName>
     <profile:restrictedTo rdf:resource=” http://www.w3.org/2000/10/XMLSchema#Date ”>
</profile:ParameterDescription >
</profile:input>
<profile:input>
     <profile:ParameterDescription rdf:ID=”Location”>
               <profile:parameterName>Location</profile:parameterName>
     <profile:restrictedTo rdf:resource=” http://site.uottawa.ca/Context#Location”>
              </profile:ParameterDescription >
</profile:input>
<profile:input>
     <profile:ParameterDescription rdf:ID=”Device”>
               <profile:parameterName>Device</profile:parameterName>
     <profile:restrictedTo rdf:resource=”“http://www.w3.org/2002/11/08-ccpp-
client#deviceIdentifier”>
</profile:ParameterDescription >
</profile:input>
```

Resource is the URI who provides the service, Service Grounding explains how it is possible to access: communication protocol as (RPC, HTTP,), port number, marshalling/serialization and finally the Service Model describes how it works: process flow, composition hierarchy and process definitions.

Once, we find what we are looking for, we negotiate the component. This is done in a transparent way for the user. We employ the same methodology and technologies that we have employed in previous projects about multimedia content negotiation [200, 201].

Thus, we are going to make deals based on offers (see Table 33), and counteroffers, then we have been developing an architecture for negotiation with mobile agents which at last requires to make at the end a negotiation with the content provider or services provider, where an agent representing user searches for multimedia content, it can be done in a automatic way using techniques that comes from artificial intelligence as expert systems as Jess developed as rules connected to semantic tools also for describing content and components. Device profiles for customization to different devices are also modelled using semantic tools.

**Table 33. Example of offer serialized as RDF/XML**


```
<rdf:RDFxmlns:rdf  = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:ipr  = "http://dmag.upf.es/ontologies/ipronto#"
    xmlns:xsd  = "http://www.w3.org/2000/10/XMLSchema#">
<ipr:Offer rdf:about="http://dmag.upf.es/mars/offer20020211183424">
    <ipr:patient>
        <ipr:PurchaseLicense>
            <ipr:licenser rdf:resource="http://www.howlinwolf.com"/>
            <ipr:licensee rdf:resource="http://chiTouristGuide.org"/>
            <ipr:permision> <ipr:Access>
                <ipr:place rdf:resource="http://chiTouristGuide.com/issues/march02"/>
                <ipr:patient rdf:resource="http://www.howlinwolf.com/imgs/0973.jpg"/>
                <ipr:user rdf:resource="http://chiTouristGuide.com/members"/>
                <ipr:timeFrom><xsd:date rdf:value="2002-03-01"/></ipr:timeFrom>
                <ipr:timeTo><xsd:date rdf:value="2003-03-01"/></ipr:timeTo>
            </ipr:Access> </ipr:permission>
            <ipr:obligation> <ipr:Compensation>
                <ipr:payer rdf:resource="http://chiTouristGuide.com "/>
                <ipr:payee rdf:resource="http://www.howlinwolf.com"/>
                <ipr:input><ipr:DollarQuantity rdf:value="100"/></ipr:input>
            </ipr:Compensation> </ipr:obligation>
            <ipr:time><xsd:date rdf:value="2002-02-11"/></ipr:time>
        </ipr:PurchaseLicense>
    </ipr:patient>
</ipr:Offer>
</rdf:RDF>
```


Analyzing the networks in Figure 72 and Figure 73, we extract the following results.

- It has been noticed some ***errors***, nodes that are not connected, it means that they are not well defined and they can not be well accessed. Ex: *weather-ont.daml#HeavyIntensity, weather-ont.damlWeatherDescriptor or weather-ont.daml#Partial*.
- Ontologies not always are related to other ontologies as upper ontologies. If it is not done, no other connections will be allowed and no new information can be discovered. This must be a parameter design.

- Nodes are related semantically and they build clusters related each other with paths and belonging to 'small worlds'. Ex: we have discovered a kind of weather report used in the aviation domain, METAR reports. METAR is the international standard code format for hourly surface weather observations which is analogous to the SA coding currently used in the US. The acronym roughly translates from French as Aviation Routine Weather Report. Therefore, we have the MetarReport ontology class that describes METAR formatted reports.
- Nodes degree can be normalized and represented, so the hierarchical semantics is showed graphically. This information shows the relevance in connections of every component. For security reasons is important to see where these components are situated in the graph and how important they are. It is the information the degree gives. The size of the node is a measure of that phenomenon.

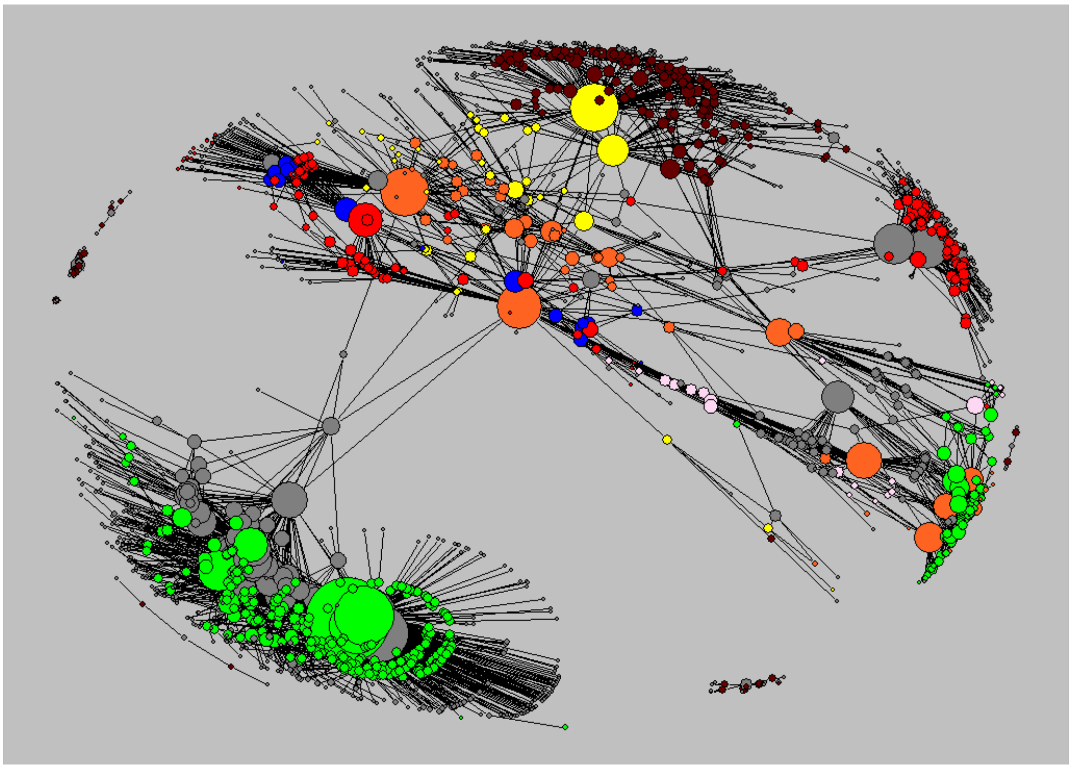


**Figure 76. Normalized degree network. Green isi, Red dublincore, Blue purl, Yellow daml+oil, Orange W3C, Brown mindswap, Pink map-ont**

Moreover, a flash interface was used to query web services and customize to a user using semantic information (profile), (see Figure 77).

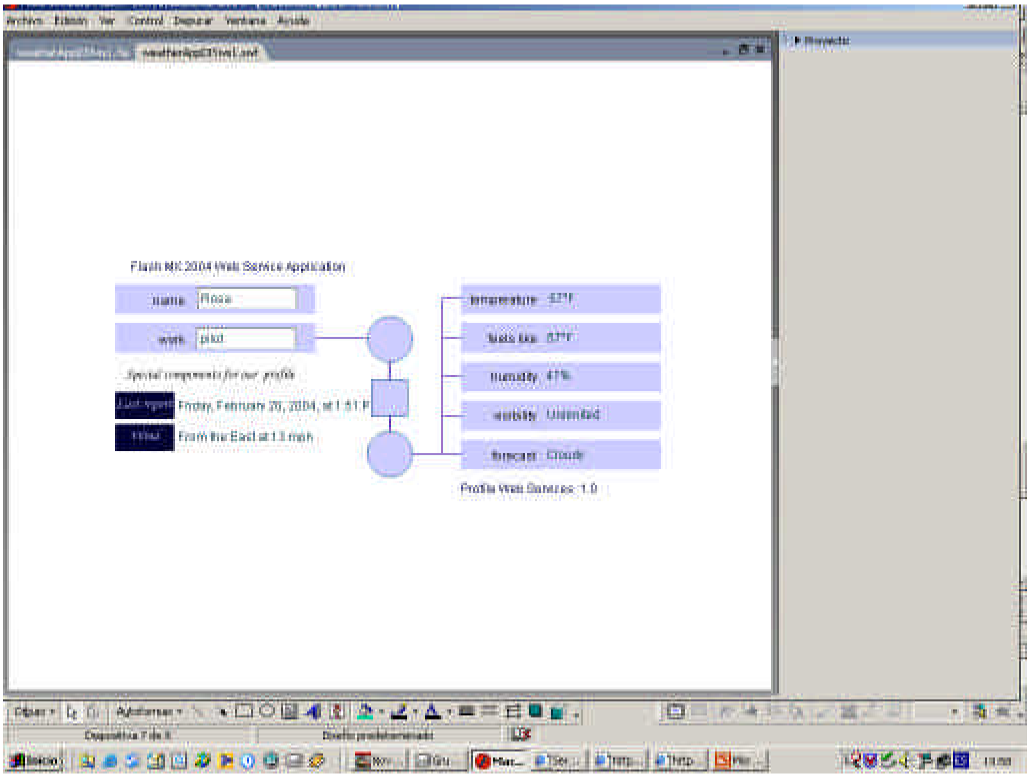



**Figure 77. Web Service Flash MX 2004 Weather interface**

## 6.9. Living Semantic Web

The goal of this section[202] is to show how to model and analyse the **Semantic Web** as a whole, i.e. as a **complex system to establish if there are global patterns (see sections 4.4.2 to 4.4.4** )

To explain the difference between simple and complex systems, the terms "interconnected" or "interwoven" are somehow essential. Qualitatively, to understand the behaviour of a complex system, we must understand not only how the parts interact but also how they act together to form the behaviour of the whole. This is because we cannot describe the whole without describing each part, and because each part must be described in relation to other parts.

Recent research on networks among scientists has focused on a number of distinctive statistical properties that most networks seem to share as acquaintance networks [203], Multi-Agent Collaboration networks [204], technological networks such as Peer to Peer systems [205], the World Wide Web [206], power grids [207], biological networks such as neural networks [208] and food webs [209] among others disciplines.

We have modeled a meaningful portion of the Semantic Web showing that it satisfies Complex Systems properties:

The following methodology has been applied:

First, we have "crawled" a set of ontologies building the RDF graph model that they define. The first step towards analyzing the Semantic Web as a Complex System is to build an appropriate graph model. Due to self-similarity of complex systems, we have selected a significant portion of the Semantic Web to perform the study. It comprises the ontologies available from the DAML Ontology Library.[210]. A modification of the RDFCrawler [211] using the Jena [212], a RDF parser (NewRDFCrawler [213]) was launched over all the DAML Library ontology URIs and all the others that were referenced from them, for instance RDF Schema or DAML+OIL. Some of them were unavailable and others not processable by Jena. The 196 processed URIs were combined in a RDF graph and serialised in N-Triples form. The RDF graph was the result of combining **160,000** triples at the starting point of the study. Summarizing:

*NewRDFCrawler* —> DAML Ontology Library (processed URIs [214], crawler log) —> RDF graph (N-Triples serialization[215])

The same has been applied to a smaller portion of the Semantic Web, starting from an individual ontology, IPROnto [216], and the ontologies referenced from it. IPROnto is an intellectual property rights ontology developed by our research group. This parallel analysis will allow comparing different scales results. The graph for IPROnto is much smaller, only 971 nodes, and it is shown in

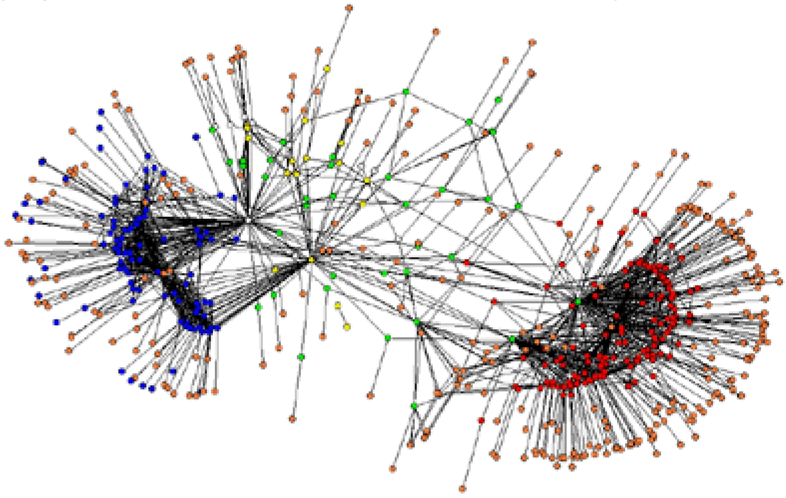


Figure 78 (Red nodes correspond to IPROnto concepts, blue ones come from different Dublin Core schemas and, on the center, a combination of resources from RDF/S (white/yellow) and DAML+OIL( light green ). Finally, literals and anonymous resources are the orange nodes).

*NewRDFCrawler* —> IPROnto (processed URIs [217], crawler log) —> RDF graph (RDF serialization [218]).

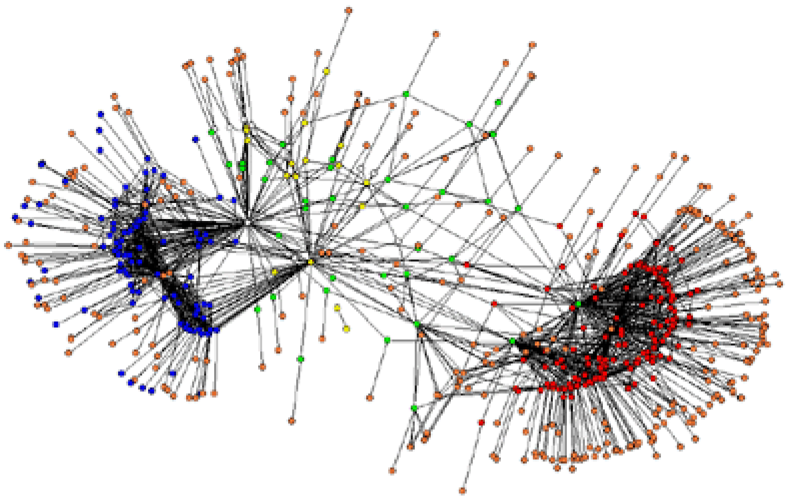


**Figure 78. Graph model for IPROnto ontology and the schemas referenced from it, directly or indirectly.**

Second, we have applied statistical tools to extract graph properties in order to compare to other complex systems.

In order to analyse the Semantic Web graph we obtained, Pajek, a large networks analysis tool, was selected. The *RDF N-Triples serialisation* was translated to a '.net' Pajek network file. The triples subjects and objects became network nodes connected by directed edges from subject to object. Nodes are identified by their original URIs to allow network construction and the edges are unnamed so duplicated edges are ignored. This is so because we do not need all this information. We are only interested in the network structure.

The Pajek network has **56,592 nodes** and **131,130 arcs**. Once loaded in Pajek, the available tools were used to obtain the required information about the graph:

- *Average degree* and *degree distribution*: using the **Net/Partitions/Degree** command.

- *Clustering factor*: using the **Net/Vector/ClusteringCoefficients** command.

- *Average minimum path length*: average over a random selection of 20 nodes (using Partition/CreateRandomPartitions and Partition/MakeCluster of size 20) and the averages of their k-neighbours vectors (using the Net/k-Neighbours with the Net/k-Neighbours/FromCluster option).

- Power-law tails exponent: linear regression from the degree distribution using GNUPlot [219].

The numeric results [220,221] of the graph analysis are shown in Table 34. The first line, the DAMLOntos network, shows the analysis of the network of ontologies at DAML Ontologies Library. The second one, IPROnto, the parameters for our intellectual property rights ontology. They can be compared with the same parameters from other complex systems networks: the results from some *WWW studies* and human *language words networks* can be seen in Table 34. The table shows for each networks, its number of nodes, the **average degree <k>**, the **clustering factor C**, the average minimum **path length <d>** and the **power-law** tails exponents. For directed graphs it is detailed input, output and all arcs parameters.

**Table 34. Comparison of results**

| Network | Nodes | <k> | C | <d> | $\gamma$ |
|---|---|---|---|---|---|
| DAMLOntos<br>(input arcs)<br>(output arcs) | 56592 | 4.63 | 0.151 | 4.37<br>5.21<br>2.95 | -1.04<br>-1.83<br>-0.86 |
| IPROnto<br>(input arcs)<br>(output arcs) | 971 | 3.71 | 0.071 | 3.99<br>2.08<br>3.03 | -3.29 |
| WWW<br>(input arcs)<br>(output arcs) | ~200 M | | 0.108 | 3.10 | -2.10<br>-2.38 |
| WordsNetwork<br>(undirected) | 500000 | | 0.687 | 2.63 | -1.50 |

From the previous data, we can deduce that the Semantic Web is a **Small World** comparing its graph to the corresponding random graph, with the same size and average degree. The clustering factor **C** = 0.152 is much greater than for the random graph **C**$_{rand}$ = 0.0000895 while the average path length is similar, **<d>** = 4.37 and **<d>**$_{rand}$ = 7.23. For IPROnto the same holds, **C**$_{rand}$ = 0.0034272 and **<d>**$_{rand}$ = 5.38.

On the other hand, studying the degree distributions, their scale-free nature has been detected and the power-law exponents have been calculated.

The distribution for DAMLOntos, shown in Table 34, is particularly interesting. It seems that there are two areas with different $\gamma$, or at least a differentiated one on the right. The first one goes from degree 1 to degree 4 with $\gamma$ = -0.43. The second one goes from degree 4 to the maximum degree, i.e. 6380, with $\gamma$ = -3.17. The second one is clear because when it is considered on its own, the error coefficient of the linear regression is only about $\varepsilon_{\%}$ = 0.75 while for the whole distribution it is $\varepsilon_{\%}$ = 4.25.

Therefore, we can deduce, as in the BA model, that preferential attachment is taking place in the more connected part of the Semantic Web, those nodes with degree greater than 3. On the other hand, it can be seen that nodes with degree smaller than 4 are basically anonymous resources and literals so it can be said that preferential attachment is taking place in the Semantic Web as a whole.

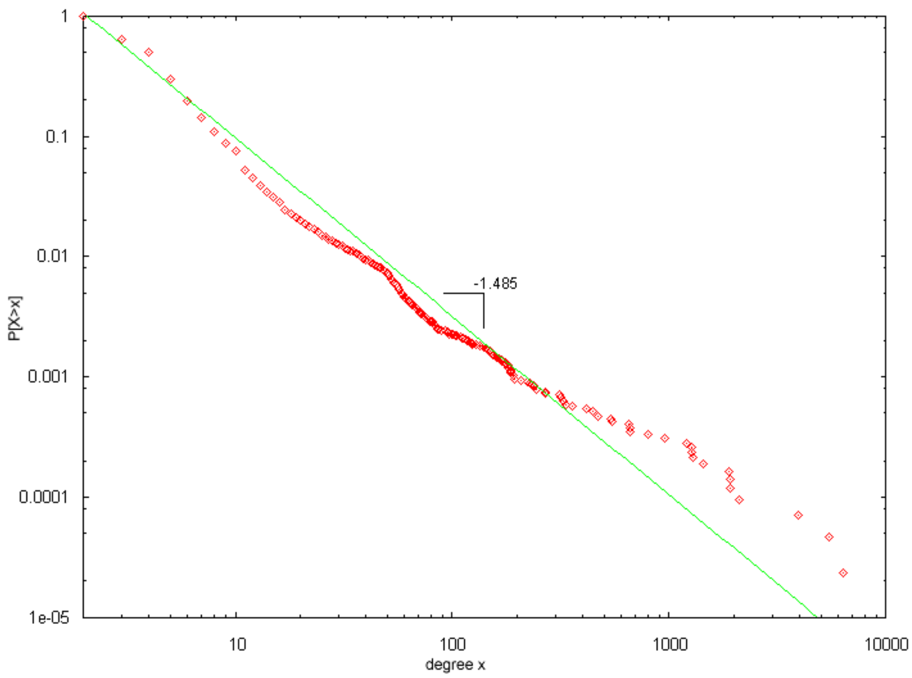



**Figure 79. Logscale Degree CDF (Cumulative Distribution Function) for the set of studied DAML library ontologies (DAMLOntos) plus linear regression and computed exponent**

## 6.10. Conclusions

The basic idea is to model real systems as agents and their environment behaviour as networks. These can be formalised using mathematical tools and graphs. Thus, it is possible to explore and analyse a great quantity of data using statistical information.

Networks that represent real systems are not always periodic and chaotic either. Their behaviour is in the middle, just moving from one to the other behaviour. It doesn't mean that any knowledge can arise because there are some patterns that repeat themselves: scale-free networks, small-world phenomenon or clustering.

Changes in network behaviour are not always infinitesimals, sometimes they are sharp, they are called 'transition phases' in physics and they are characteristic in networks that behaviours in this region 'at the edge of chaos'. They are not exactly free, Bak [76] discovered that they are attached to these kind of systems, this model is know as 'sand pile model' and it follows a power law distribution.

Knowledge is also represented by networks, i.e. graphs. These graphs are known as ontologies and they possess syntax plus semantics.

Ontologies as knowledge representation represents a suitable tool for achieving a shared knowledge. It is accessible and moreover provides a framework where it is easy to modify contents.

Ad hoc applications can take profit of it because it has to accomplish the same goals as ontologies provides. They are distributed and for their own nature can discover new knowledge.

- Application components that are services or part of services have to be semantically described to achieve ad hoc applications with no information about where they are ( a path directly related to them)
- Semantically related, it means that application components are members of an ontology. Well-defined ontologies are related to 'upper ontologies', that are the *common point* to establish new relations and discover new knowledge.
- Semantic relations build a graph, if there are many, it becomes a large graph or large network. Statistical analysis can be applied. Many properties arise such "small world", degree or clustering. This information shows changes in topology, community phenomenon, clustering and even security analysis can be performed. For example, when a component is highly connected, it becomes a vulnerable component, so it is necessary to control how many there are and where they are situated in the graph.

Another connection is mentioned above: ontologies are a complex system. If they are and the results seem to assure it. The properties of complex systems can apply to ontologies.

Thus, a model that can be used to model the environment as the agents themselves: a network conceptualised as a graph, that behaviours as a complex system is identified as Semantic Web in this case.

Semantic Web is a technological network as WWW is. The challenge was to analyse the huge amount data as a whole and perform an analysis to prove that it behaviours following a pattern and how it could be useful. This analysis has used mathematical tools as statistics ,graphs and semantic representation.

The results have shown that it is very near from a power-law and it's scale free. Moreover it's not the same as language network, however phenomenon as clustering or small world can help to analyse which are the most connected parts or how they are connected to improve Semantic Web complexity to be similar to human languages. A methodology for exploring a huge network and translate to semantic representation has been developed.

It implies that a tool for working with huge amount of network data, being capable of extract semantic information has been developed.

A first step to manage the complexity of IPR laws has been achieved.

# 7. A different approach to negotiation representation in a SWA

At last, negotiation basis are abstract terms only understood by human beings. In our case described in ontologies in natural language. If artificial agents that reason as well as human beings have to be developed. It is necessary to understand how persons use abstract terms to give the first step. Mathematical language and Semantic Web languages are abstract languages, however the first is better related to essential concepts as space and time. The aim of this chapter is trying to make a way to arrive to the roots of mathematical language and make a connection to Semantic Web languages.

Then, how can agents learn new concepts in mathematics? For answering, just explain some points about mathematical operators that perhaps they are not public domain.

It is possible to add, subtract, multiply and divide objects as primary operations. However this was not the right order in human beings understanding [222]. Human beings could add, multiply and divide before they can understand subtract. In the XVI century negative numbers were accepted and they are referenced as 'false' numbers. In fact, when human beings operate with numbers another level of abstraction is done. As an example Table 35 shows that each number has a different representation depending what it is going to be counted, in different cultures as tribal Indian people in British Columbia in Canada (They spoke 'tsimshian' language. This information was gathered by north-american anthropologist Franz Boas and published for the first time in 1881), or even in Japanese culture.

**Table 35. Numeral numbers used by Indian tribal people in British Columbia in Canada.**

| N. | Oral numeration | Plane objects | Round objects | People | Long objects | Canoe | Measures |
|---|---|---|---|---|---|---|---|
| 1 | Gyak | Gak | G'erel | K'al | K'awutskan | K'amaet | K'ai |
| 2 | T'epqat | T'epqat | Goupel | T'epqadal | Gaopskan | G'alpeeltk | Gulbel |
| 3 | Guant | Guant | Gutle | Gulal | Galtskan | Galtskantk | Guleont |
| 4 | Tqalpq | Tqalpq | Tqalpq | Tqalpqdal | Tqaapskan | Tqalpqsk | Tqalpqalont |
| 5 | Ketone | Ketone | Ketone | Keenecal | K'etoentskan | Tetoonsk | Ketonsilont |
| 6 | K'alt | K'alt | K'alt | K'aldal | K'aoltskan | K'altk | K'aldelont |
| 7 | T'epqalt | T'epqalt | T'epqalt | T'epqaldal | T'epqaltskan | T'epqaltk | T'epqaldelont |
| 8 | Guandalt | Yuktalt | Yuktalt | Yuktleadal | Ek'tlaedskan | Yuktaltk | Yuktaldelont |
| 9 | Ketemac | ketemac | ketemac | Ketemacal | Ketemaetskan | Ketemack | Ketemasilont |
| 10 | Gy'ap | Gy'ap | Kpeel | Kpal | Kpeetskan | Gy'apsk | Kpeont |

Counting itself was an important step in abstraction, in Mesopotamian culture, people has to relate stone-to-sheep to count. A baby needs his/her own collection of fingers to count and even some adults do it. In other words, fingers represent a collection that it is always with us. It seems that mapping from our perceptions as objects to numbers allow human beings to count. The branch of mathematics that takes care of relation of space time is geometry.

Geometry is the common language for human beings to understand the relation to space-time. Points, lines, vectors, matrix…. represent the perception of real world or sometimes called the objective world. It conceptualises the basic common language known as mathematics in an efficient way.

Languages are symbolic, they are built with symbols. Symbols are part of signs, it is true that they are languages built over signs, however human beings usually prefer symbols. Also there is no waste in energy if it possible to express more with less.

Language symbols built words. They relate to each other syntactically. Parsers are used to extract these relations, which are expressing in the context of this language how words are related.

Mathematics is a language too. Geometry represents the meaning for this language, in other words, the nearest relation to the describe real world. By the other side mathematics is often conceptualised by Algebra, which are symbols. It will be worth to unify in the same representation Algebra and Geometry. It means that meaning understood as the nearest relation to real world (***geometry***) and a symbol(***algebra***) identifying it will be together. This next section explains it.

Summarizing, if the agents can represent their knowledge related to symbols that have a geometric meaning it could be a common language understood for all human beings and for any system able to capture spatial and time information.

## 7.1. Adding a new representation

*"... for geometry, you know, is the gate of science, and the gate is so low and small that one can only enter it as a little child".*

William Kingdon Clifford (1845-1879), fellow of the Royal Society

*" the hardest task in mathematics is the study of the elements of algebra, and yet this stage must precede the comparative simplicity of the differential calculus".*

Alfred North Whitehead, Philosopher.

Geometry helps us to compare objects, from ancient times classical Euclidean geometry was taken to be an absolute framework until Einstein formulate General Relativity. Human beings refer space and time objects properties to a framework.

Leibniz (characteristica geometrica)[223] dreamt with a geometric algebra, he thought that when algebraic equations were visible to us as a geometric forms everyone will understand them. Some work was done by Grassmann and Clifford in the XIX century however was partially lost in the vector analysis[224] that it is taught in vectorial spaces.

The next paragraphs explain this part and show what was missing
A story[225] began in the 19$^{th}$ century when representing rotations in 3-D was a problem that occupied mathematicians a long time. Hamilton produced the ***quaternions***, which were a generalization of the complex numbers to 3-D (1844). This algebra contains four elements:

$$\{1, i, j, k\}$$

which satisfy

$$i^2 = j^2 = k^2 = ijk = -1$$

however this elements had not vector properties.

While Hamilton was developing his quaternionic algebra, Grassmann was formulating his own algebra in 1844 and 1877, the key to which was the introduction of the exterior or outer product (denoted by '^') and has the associativity and anticommutativity features, however the inner product is commutative

Here are the foundation of much of modern supersymmetry and superstring theory.

William Kingdom Clifford in an attempt to unite the algebra of Hamilton and Grassmann into a single structure, he introduced his own *geometric algebra.* In this algebra we have a single geometric product formed by uniting the inner and outer products – this is associative like Grassmann's product but also invertible ,like products in Hamilton's algebra. In Clifford's geometric algebra an equation of the type ab=C has the solution $b=a^{-1}C$, where $a^{-1}$ exists and is known as the inverse of a.

In *Appendix B: Geometric Algebra* a little explanation is detailed about the difference with classical vector approach.

The relevant features for geometric algebra of the plane are the following:

Suppose two orthonormal basis vectors: $\{e_1,e_2\}$, when a vector is multiplied by itself the result is 1, if the first multiplies the second and vice versa the result is 0.

The outer product $e_1$^$e_2$ represents the directed area element of the plane and $e_1$ and $e_2$ have the conventional right-handed orientation

| | | |
|---|---|---|
| $1$ | $\{e_1, e_2\}$ | $\{e_1 \wedge e_2\}$ |
| scalar | vectors | bivector |

In a Clifford algebra, an element is expressed as a multivector:

$$A \equiv a_0 1 + a_1 e_1 + a_2 e_2 + a_3 e_1 \wedge e_2$$

and the geometric product is expressed as:

$$ab = a \cdot b + a \wedge b$$

Summarizing:

The geometric product of two parallel vectors is a scalar number: the product of their lengths

The geometric product of two perpendicular vectors is a bivector: the directed area formed by the vectors.

Parallel vectors commute under the geometric product however perpendicular vectors anticommute.

The bivector $e_1$^$e_2$ has the geometric effect of rotating the vectors e1 and e2 in its own plane by 90 degrees clockwise when multiplying them on their left. It rotates vectors by 90 degrees anticlockwise when multiplying on their right. This can be used to define the orientation of $e_1$ and $e_2$

The square of the bivector area is a scalar =-1. A subalgebra is equivalent to the complex numbers.($z=x+ye_1e_2=x+yi$)

Geometric Algebra seems to be a powerful to describe a symbolic system (algebraic) without loosing meaning (geometric). What it is interesting is that geometric properties are closer to be universal when they are defined appropriately.

## 7.2. Logic and axioms

This section outlines which are the mathematical roots when a theory is postulated. This reflexion is necessary to understand the innate model contraints. Thus, what about the pillars of mathematics as logic and axioms? (see Figure 80).

Unfortunately, both are relative (for instance, Euclidean Geometry and Aristotle classic logic have been true statements during many centuries) however it doesn't mean that something is true, as Thomas Khun [226] explains everyone has to situate his/her point of view in context. Thus, once it is achieved, a statement can be said.

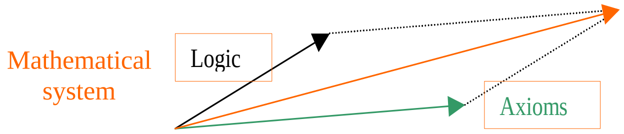



**Figure 80 Logic are system rules while axioms are true statements, their combination makes mathematical systems.**

These lines show that only it is possible a 'general framework' in a bounded region. Although it seems paradoxically, it is a statement to help us that when something comes, it must be accepted that it comes with assumptions, bounding the 'general framework'. Kuhn has made this point most strongly in considering theory and research in science. His "Structure of Scientific Revolution" provided the grounding for a major paradigm shift in science toward a "best description" view of theory rather than an approximation to the "truth". In essence, he argued that the meaning of our vocabulary resides in our theory rather than outside of it. Thus ***there is no meta-vocabulary that sits independent of theory*** and as such, it is impossible to translate between theories. That is, theories provide their own lens into the world, with each theory providing a different lens (or perspective). For example, Kuhn argues that there is no theory independent way to reconstruct phrases like "really there". All "facts" are theory laden.

From the before discussion:

The architecture can be conceptualised by a model. It can be represented by concepts and their relations, i.e. mathematically as a graph. As part of a mathematical language, it is constituted by symbols (algebra) and meanings (geometric relations) which are interpreted by agents
The models are constrained by axioms and logic which determines a domain understood as framework where situations can be more or less bounded.

### 7.2.1. Network statistical properties

The questions to be addressed first will be the graph properties and graph configurations. Afterwards a review of how these properties have been found in different disciplines. The key is discover more about the knowledge domain and fit the context to provide a framework. This section shows the discrepancy about this issue and how it is not possible to an universal network behaviour however some patterns appear.

Königsberg (Prussia) was the city where Euler offered in 1736 a rigorous mathematical proof stating that with the seven bridges such aha path does not exist to connect them. He not only solved the Königsberg problem, but in his brief paper inadvertently started an immense branch of mathematics known as graph theory.

The origin of the study of searching is in sociology since the seminal experiment of Travers and Milgram, however Barabasi has another version about the origin (Karinthy's) [61]. Surprisingly, it was found that the average length of acquaintance chains was about six. This means not only that short chains exist in social networks as reported, for example, in the "small world" as WS model, but even more striking that these short chains can be found using local strategies, that is without knowing exactly the whole structure of the social network.

WS most important discovery is that clustering does not stop at the boundary of social networks. The model offered an elegant compromise between the completely random world of Erdös and Rényi, which is a 'small world' but hostile to circles of friends, and a regular lattice, which displays high clustering but in which nodes are far from each other. The connectors were missing yet.

A random web had been the ultimate carrier of egalitarianism, since the Erdös and Rényi theory guarantees that all nodes are very similar to each other, each having roughly the same number of incoming links

Three types of semantic networks: ***word associations, WordNet, and Roger's thesaurus*** [227] present Small-world structure and it is characterized by sparse connectivity, short average path-lengths between words, and strong local clustering. In addition, the distributions of the number of connections follow power laws that indicate a scale-free pattern of connectivity, with most nodes having relatively few connections joined together though a small number of hubs with many connections. These regularities have also been found in certain other complex natural networks, such as the www.

Natural and artificial nets display a surprisingly widespread feature: **the presence of highly heterogeneous distributions of links, providing an extraordinary source of robustness against perturbations**. A simple optimisation process can also account for the observed regularities displayed by most complex nets. Using an evolutionary algorithm involving minimization of link density and average distance [228].

However how is it possible to search in a graph or even navigate ? Is it perhaps, a random process, is there any knowledge ? The search algorithm proposed by Kleinberg is the following [242]. A packet standing at one node will be sent to the neighbour of the node that is closer to the destination in terms of the distance ?. The algorithm is local because, as shown in the next figure, the heuristics of minimizing ? does no warrant that the packet will follow the shortest path between is current position and its destination. Therefore, the underlying two-dimensional lattice has an imprecise global information content.

Kleinberg (see Figure 81, Consider nodes A and B. The distance between them is ?AB=6 although the shortest path is only 3. A search process to get from A to B would proceed as follows. From A, we would jump with equal probability to D or F, since ?DB= ?FB = 5 : suppose we chose F. The next jump would then be to G or C with equal probability since ?CB = ?GB =4, although from C it is possible to jump directly to B. This is a consequence of the local knowledge of the network assumed by Kleinberg.) showed with a local algorithm over a two-dimensional regular lattice that in essentially local scenario short paths cannot be found in general, unless the parameter r is fixed to r=2.Where r is the parameter from the expression of the probability that the link is established with node j is:

$$\prod_{ij} \propto (\Delta_{ij})^{-r}$$

The problem of search and congestion simultaneously it is solved for hierarchical lattice by exploiting the symmetry properties of the network. For complex networks is related to the structure of the network where short-range, long-range, random and preferential connections are mixed , the network performs well for very low load and it becomes easily congested when the load is increased. When searching for optimal structures in a general scenario there is a clear transition from star-like centralized structures to homogeneous decentralized ones.

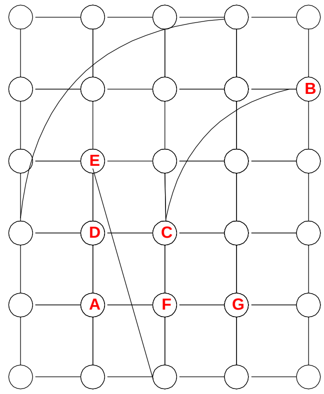



**Figure 81 Network topology and search in Kleinberg’s scenario.**

These results show that ***even with local knowledge is not possible to predict behaviour***. It could be very interesting if the networks will hold sometimes patterns suitable for inference their behaviour.

*One question that arises is if it is possible to compare Internet-like networks and classical random graph based network.*

Trying to inference the scaling of the traffic load with the nodal degree was established, and confirmed in a numerical simulation of the TCP traffic. The scaling allows to estimate the link capacity upgrade required making and extra connection to an existing node [229].

Anomalous phase transition of the emergence of the giant connected in scale-free networks growing under mechanism of preferential linking [230]. There are exact results for the size of the giant connected component and the distribution of vertices among connected components. Moreover in the entire phase without the giant connected component such growing networks are in a “critical state”, in other words, the probability that a randomly chosen vertex belongs to a connected component of the size k is of a power-law form.

How do local breakdowns due unbalances or congestion propagate in real dynamical networks ?. The model [231] converges to a self -organized critical stationary state in which the network shapes itself as a consequence of avalanches of rewiring processes. Depending on the model’s specification, it is possible to obtain either single scale or scale-free networks. The statistical properties of the network and the nature of the critical state is computed by critical exponents.

A slow dynamics where links are added to the network and a fast relaxation dynamics of avalanche events.

Thus, statistical properties of the network provides an extra information. However, is it possible to extract new information applying conservation laws?

**Ilya Prigogine** distinguishes between “conservative” systems (which are governed by the 3 conservations laws for energy: translational momentum and angular momentum (which give rise to reversible processes) and “dissipative” systems (subject to fluxes of energy and/or matter)

He proved that, under special circumstances, the distance from equilibrium and the no linearity of a system drive the system to ordered configurations.

No linearity express the fact that a perturbation of the system may reverberate and have disproportionate effects. Moreover, non-equilibrium and non-linear systems favour the spontaneous development of self-organizing systems. **This fact reveals that sometimes a controller is not the best option to design self-organizing systems**.

He defines complexity as “the ability to switch between different modes of behaviour as the environmental conditions are varied”.

**Mathematical networks are also an inspiration in biology. Stuart Kauffman**'s first discovery was that cells behave like mathematical networks. He proved that, in any organism, the number of cell types must be approximately the square root of the number of genes.

It have been seen that the rules population goes to a space region of every rule that it is on the edge of chaotic rules and not chaotic rules [232].

After the exhaustive analysis, the conclusion is that networks are generally very far from random [233]. They have highly distinctive statistical signatures, some of which, such high clustering coefficients and highly skewed degree distributions, are common to networks of a wide variety of types.

The high connectivity nodes play the important role of hubs in communication and networking, a fact that can be exploited when designing efficient search algorithms. They introduce a number of local search strategies that utilize high degree nodes in power-law graphs and that have costs scaling sub-linearly with the size of the graph [234].

Tools taken from statistical mechanics have been used to understand not only the topological properties of these communication networks, but also their dynamical properties. The main focus has been in the problem of search ability, although when the number of search problems that the network is trying to solve increases it raises the problem of congestion at some central nodes [235].

In real networks and in model communication networks, they collapse when the load is above a certain threshold and the observed transition can be related to the appearance of the 1/f spectrum of the fluctuations in Internet flow data.

Agents are nodes of a network and can interchange information packets along the network links. Each agent has a certain capability that decreases as the number of packets to deliver increases. The transition from a free phase to a congested phase has been studied for different network architectures and the problem of network optimisation for fixed number of links and nodes has been attacked.

The idea of network arises even at LQG (*loop quantum gravity*), which is a theory that tries to unify the M-theory that will explain the hole universe.
In LQG, the fabric of space-time is a foamy network of interacting loops mathematically described by spin networks. These loops are about $10^{-35}$ meters in size, called the Plank scale. The loops knot together forming edges, surfaces, an vertices, much as do SOAP BUBBLES joined together. ***Space-time is quantized***. In LQG, spin networks represent the quantum states of the geometry of relative space-time.

Networks are around us, their dynamics based only on **the preferential attachment of new nodes do not lead to a sufficiently heavy-tailed degree distribution in ad hoc networks** [236]. In fact, it is important to introduce a local dynamics because even in the limit of equal insertion and deletion rates true scale-free structures emerge, where the degree distributions obey a power-law with a tunable exponent, which can be made arbitrarily close to 2. It means there is an evidence of emergence of scale-free degree distributions purely due to dynamics, i.e., in networks of almost constant average size. Dynamics can be used to craft protocols for designing highly dynamic P2P networks.

It is proposed a new role-based self organization algorithm that extends the hierarchical connected dominated set (CDS) architecture for scalable operation of the network [237]. They assign routing and sensing roles to sensor nodes depending upon their connectivity and sensing capabilities, respectively. These sensing zones individually act as an aggregate consisting of sensor nodes collaborating to achieve a common sensing objective with a certain sensing quality of service (sQoS). They elect sensing coordinators that act as leaders for their respective sensing zones.

They presented the foundations of a random graph modelling approach for the scalable analysis/design of the topological properties of large ad hoc networks [238], [239].

An important observation in nature is that systems of interest are often open, and out of thermodynamic equilibrium state. Their simplest condition is that of a non-equilibrium steady state. Non-equilibrium instabilities are attended by ordering phenomena so analogous to those of equilibrium statistical mechanics that one may speak of *non-equilibrium phase transitions*.[240]

For instance, the neural network of the worm C. Elegans, see Figure 82 (where Symbols: black down triangle, Ythan estuary; red triangle, Silwood park food web; star, C. elegans neural network; red square, E. coli substrate graph; green square, E. coli reaction graph; black square, E. coli metabolic network; red circle, Internet, router level, 1995; green circle, Internet, domain level; X, power grid; black triangle, NCSTRL coauthorship; green triangle, LANL coauthorship; blue triangle, SPIRES coauthorship; yellow triangle, math coauthorship; black circle, Internet, router level, 2000; red diamond,WWW, site level, undirected; brown triangle, actor collaboration; black diamond,WWW, nd.edu; purple triangle, MEDLINE coauthorship; green diamond, WWW)., has been shown to have a small-world structure, but is definitely not scale-free in its connectivity. The degree distribution falls off according to a very clear exponential law with a single characteristic scale[241] .Likewise, the connectivity of neurons within a cortical area may have some small-world features, due to the existence of excitatory connections within a local two-dimensional neighbourhood and long-range inhibitory connections between neighbourhoods [242].

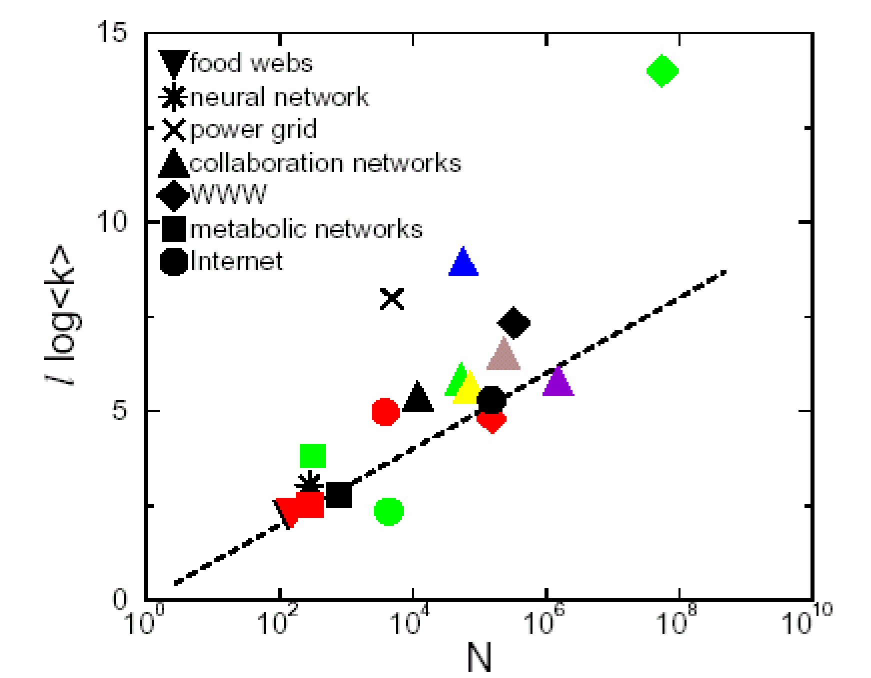



**Figure 82 Comparison between the average path length of real networks and the prediction of random graph theory (dashed line).**

In Economics, networks are strong when they are dense and the cluster k nodes build which is called ("closely knit" by Peyton Young ). It increases with agents invasion, it no has relation to connectivity and for that reason it is independent of the way the connectivity arises. The question when dynamics over the network has to be solved is if it is possible to design agents which behaviours allow to coordinate themselves[243]. The set of states that are stable strategies is a subset of states that performs Nash equilibrium. Sub-optimum Nash equilibrium can arose by stable evolutionary behaviour and no phase-transition is allowed if some agents change their behaviour[244].

Thus, these examples show that networks are used in many domains, it is done because statistical properties networks allow to describe systems with a great amount of particles/nodes.

## 7.2.2. Meaning

If observer is included, nodes and relations have a subjective meaning depending on the observer. Only when this meaning is shared over a community arises the idea of *'objective meaning'*. However it is not an easy question to solve, the classical example of colour (see Figure 83 for conceptual spaces of colour) shows that it is impossible to isolate the object from the observer when meaning is implied. What is ***blue colour***?. It is not only the physical wave. It is the whole process between the being and the light wave [253].

It has been seen that every ontology does not reflect an objective knowledge. It is distributed and sometimes difficult to fix in the same space-time because links are broken or never exist and an ontology doesn't represent a truth for a determinate agent, it is need to reach a consensus or agreement among some agents. At last, the model it is very similar to molecularism and not holism (see Figure 84 ) because there are several ontologies generated by different communities that are related to each other. At last, standards reflect this kind of common connections. ***The effort has to be made not to create a 'perfect model of the world' if not a model with high connections to relevant models accepted by the community.***

Two things are present: it is not possible to separate environment from the observer and it does not exist absolute frameworks. However, how can it be managed by our cognitive model?

An assumption has to be made: Inferences rely on internal cognitive model (geometrical representations) and facts about graded structures are not independent

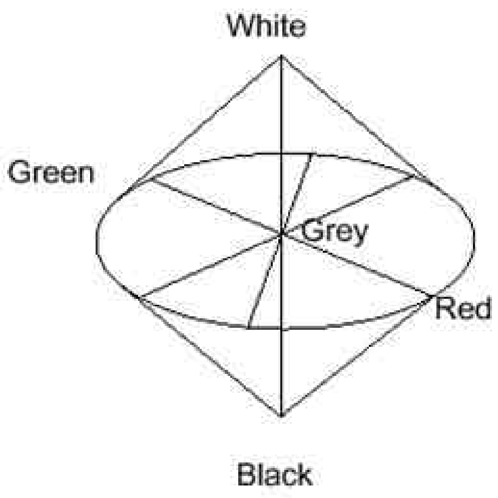



**Figure 83 Conceptual spaces**

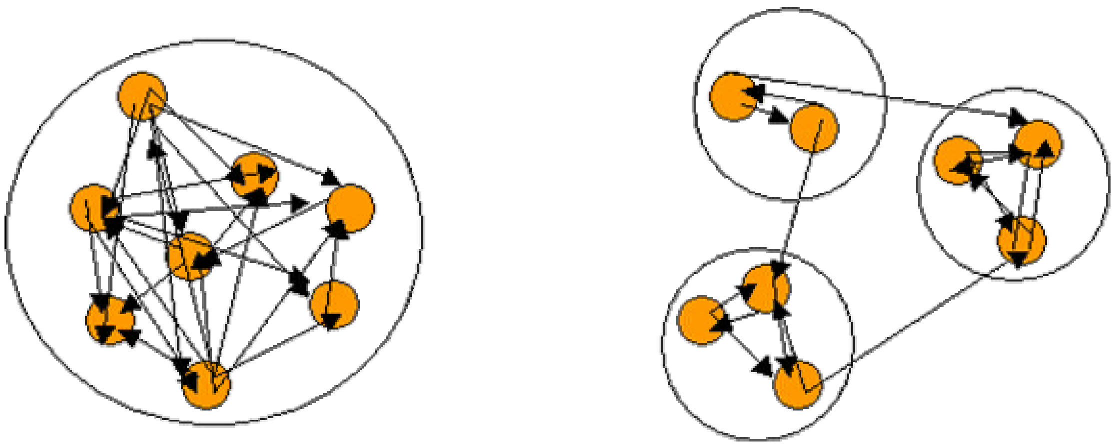


**Figure 84 Holism vs. molecularism**

When linguists look for features which are "universal," they can't look at every language in the world, so they look at several languages from different language families. If they find the feature in these unrelated languages, it is likely to be universal.

The next question is related about how beings and in particular human beings perceive and structure the physical information from environment.

Image schemas are an attempt to solve it (see section 4.6). Moreover it seems difficult to say that there is an evidence for image schemas in lower primates because the main evidence for them is linguistic. It is known that animals have spatial abilities, so the question is whether or not the animal uses spatial schemas to conceptualise other things such eating utensils [245].

After many cases of networks, framework problem or domain problem is unavoidable. Then, the question has become as: it is possible to model the domain in the context of negotiation?

In Europe in the 40s and 50s, there was a theory of semantics having to do with words and their relationships which is called lexical field theory. These theorists studied certain related sets of words, such as day of the weeks. They noticed that there were certain relationships between these words in these sets. Other lexical fields include kinship terms, eating utensils, etc. In the mid-70s, it was observed that to really understand the relationships between these words, it is necessary to understand the structure underlying the fields. For instance, to understand the relationships between words such as "***buy***," "***sell***," etc., it is necessary to understand the ***commercial event frame***. In any frame there are participants. In the commercial events frame, the participants are '***buyer***' and '***seller***'. Other entities are '***money***' and '***goods***'. In frames, there are also scenarios *which have three states*. In the commercial event frame, the initial state is:

Buyer has money and wants goods.
Seller has goods and wants money.
The middle state is an exchange:
The Buyer gives money to the Seller.
The Seller gives goods to the Buyer.
In the final state:
The Buyer has goods.
The Seller has money.

The predicates in the frame, such as '***have***' and '***want***', are simpler than '***buy***' and '***sell***', which are defined relative to the frame. Notice that there is a ***causal structure***, a linear order structure and other types of structure in the frame. There are also inferences you can make given the structure. For instance, if you say "John bought a book from Harry," you can infer that John owns the book now and that Harry has more money now. *The idea is that lexical items get their meanings from frames, from the overall structure of the frames*. The frames relate all the *lexical items* in a *lexical field*. Frames also define semantic roles of the participants and entities, such as **agent**, **patient**, **source**, etc. In the phrase, "John bought the book from Henry", John is the agent and goal; Harry is the source; the book is the patient.

Frame problem in AI is translating to no criterion for assigning pertinence and stopping combinatorial explosion. One of the proposed solutions was to make it ***domain-specific architecture*** [246]

A "schema" is a way of describing conceptual categories such as containers, mammals, or commercial transactions. Schemas are defined through their relationships to other schemas and through their "roles." Roles are simply the constituents that are required to define a category – for example, the "commercial transaction" schema might have roles for "buyer," "seller," "payment," and "goods."

Minsky in the 70's expressed the necessity to design frames based on schemas. For instance:

Ex: TIGER
Slots (default values as prototypic values)
Is a : mammal
Has legs : 4
Eats : meat
Has fur :yellow, black, white
Lives in: jungle, zoo, circus

Frames, or concepts, are limited clusters or properties with paradigmatic cases (prototypes).

There are schemas that conceptualise physical information (cognitive grounding) and frames that give a context to theses schemas, the next question is how is it possible to relate different frames? The concept of ***metaphor*** arises when some schemas are the same in different knowledge domains.

If the concept of metaphor is applied to a great deal of everyday language not typically considered metaphorical some examples give the key for trying to understand the way that perhaps works for human being understanding. For example, ECG (Embodied Cognitive Grounding) treats the statement, ***"Prices rose," as a metaphor***. If you don't see how this could be the case, ask yourself this: When the prices in question increased, did they, in any literal sense, move? The verb "rose" belongs to the domain of movement through physical space. In this example, spatial movement is the "source domain" of the metaphor – the conceptual category in which the literal meaning of the statement belongs. "Prices," on the other hand, belong in a domain we might call commerce. In this example, commerce is the "target domain" of the metaphor – the conceptual category in which the subject of the metaphorical statement belongs.

In addition to defining schemas in isolation, it is important to define ***relationships between schemas and between metaphors***. There are two ways that entities of the same type can be related in ECG: entities can be "subcases" of other entities of the same type, or they can be "evoked" by other entities of the same type.

A “subcase” of an entity is a special or more specific case of that entity. For example, a “vessel” is a specific type of “container;” accordingly, the “vessel” schema would be a “subcase” of the “container” schema. When we say that an entity “evokes” another entity, we are stating that the evoked entity is somehow required in order to define the entity that evokes it, though they may not in any sense be “the same.”

FrameNet [247] is a project that attempts to conceptualise all this information. They provide a database in XML, we have translated to OWL, using partially an initial work [248]. The database is FrameNet property. The review of the stylesheets can be found on the web[249] and can be manipulated by Protègè. The stylesheets have been used to translate XML to OWL and extract ‘process’, ‘event’ and ‘state’ concepts, see Figure 85.


**Figure 85 FrameNet representation for ‘Process’, ‘State’ and ‘Event’.**

## 7.3. Conceptualising cross-domain knowledge

Usually, language, knowledge is associated to mind and brain. The following section wants to show that perhaps if knowledge and language is associated to space-time perceptions, it is also associated to all the systems involved in this process: the body and how is this information manipulated to reach abstract mathematics.

Conceptual metaphors, which cognitively are cross-domain mappings preserving inferential structures. Srini Narayanan, in his dissertation, models these (also in a structured connectionist model) using *neural connections* from *sensory-motor areas* to other areas. Narayanan's startling result is that the *same neural network structures that can carry out high-level motor programs can also carry out abstract inferences about event structure under metaphorical projections*. Since metaphorical projections preserve inferential structure, they are a natural mechanism for expanding upon our inborn numericizing abilities.

Nuñez and G.Lakoff [250] have found that metaphorical projections are implicated in two types metaphorical conceptualization. First, there are **grounding metaphors** that allow us to expand on simple numeration using the structure of everyday experiences, such as forming collections, taking steps in a given direction, and so on. They find, not surprisingly, that basic arithmetic operations are metaphorically conceptualized in those terms: ***adding is putting things*** in a pile; ***subtracting is taking away***. Second, there are **linking metaphors**, which allow us to link distinct conceptual domains in mathematics. For example, we **metaphorically conceptualize numbers as points on a line**. In set-theoretical treatments, numbers are metaphorized as sets. Sets are, in turn, metaphorically conceptualized as containers.

They have looked in detail at the conceptual structure of cartesian coordinates, exponentials and logarithms, trigonometry, infinitesimals (the Robinson hyperreals), imaginary numbers, and fractals. They have worked out the conceptual structure of ***e*** to the power ***pi*** times **i**. It is NOT ***e*** multiplied by itself ***pi*** times and the result multiplied by itself i times-whatever that could mean! Rather it is a complex composition of basic mathematical metaphors.

Simple numeration is expanded to "abstract" mathematics by metaphorical projections from human beings sensory-motor experience. Persons do not just have mathematical brains; they have mathematical bodies! The everyday functioning in the world with the brains and bodies gives rise to forms of mathematics. *Mathematics is not "abstract", but rather metaphorical*, based on projections from sensory-motor areas that make use of "inferences" performed in those areas. The metaphors are not arbitrary, but based on common experiences: putting things into piles, taking steps, turning around, coming close to objects so they appear larger, and so on.

Simple numeration appears to be located in a confined region of the brain. But mathematics [251] - all of it, from set theory to analytic geometry to topology to fractals to probability theory to recursive function theory - goes well beyond simple numeration. Mathematics as a whole engages many parts of our brains and grows out of a wide variety of experiences in the world. What Nunez and Lakoff have found is that mathematics uses conceptual mechanisms from our everyday conceptual systems and language, especially image-schemas and conceptual metaphorical mappings than span distinct conceptual domains. *When a person is **thinking of points inside a circle** or elements in a group or members of set, he/she is using the **same image-schema of containment** that he/she uses in thinking of the chairs in a room*.

There appears to be a part of the brain that is relatively small and localized for numeration. Given the subitizing capacity of animals, this would appear to be genetically based, it means being able to distinguish easily collections of one to four objects. But the same cannot be said for mathematics as a whole. There are no genes for cartesian coordinates or imaginary numbers or fractional dimensions. These are imaginative constructions of human beings. And if Nunez and Lakoff are right in their analyses, they involve a complex composition of metaphors and conceptual blends [252]

Moreover, Dehaene thinks that this requires a non-platonic philosophy of mathematics that is also not socially constructivist. Indeed, what is required is a special case of experientialist philosophy (or "embodied realism") [253], [254], [255]. Such a philosophy of mathematics is not relativist or socially constructivist, since it is embodied, that is, based on the shared characteristics of human brains and bodies as well as the shared aspects of our physical and interpersonal environments. As Dehaene said, ***pi*** is not an arbitrary social construction that could have been constructed in some other way. Neither is e, despite the argument that Nuñez and Lakoff give that human being understanding of ***e*** requires quite a bit of metaphorical structure. The metaphors are not arbitrary; they too are based on the characteristics of human bodies and brains.

Mathematics is not platonist or objectivist from this point of view. As Dehaene says, it is not a feature of the universe. But this has drastic consequences outside the philosophy of mathematics itself. The reason is that the correspondence theory of truth does not work for mathematics. Mathematical truth is not a matter of matching up symbols with the external world. Mathematical truth comes out of human beings, out of the physical structures of our brains and bodies, out of our metaphorical capacity to link up domains of our minds (and brains) so as to preserve inference, and out of the non-arbitrary way we have adapted to the external world. Mathematics has been, after all, the paradigm example of objectivist truth.

## 7.4. Building a common resource

In the labour of building a common resource, in this case a domain knowledge, human beings share some capabilities. For instance, to characterize arithmetic operations and their properties, human beings exhibit:

- Capacity for subtitizing: it means be able instantly to recognizing small numbers of items, one to four
- Capacity for estimating how many objects there are in a collection. There are some cognitive capacities needed to count from one to four:
  - grouping capacity
  - ordering capacity
  - pairing capacity
  - memory capacity
  - exhaustion-detection capacity
  - cardinal-number assignment
  - independent-order capacity.
- Capacity to count beyond four:
- Combinatorial-grouping capacity
- Symbolizing capacity

Following cognitive linguistics, arithmetic is object collection [250], a mapping is performed between two different domains, it means that it is a metaphor.(see Figure 86)

| Souce domain OBJECT COLLECTION | | Target domain ARITHMETIC |
|---|---|---|
| Collections of objects of the same size | → | Numbers |
| The size of the collection | → | Size of the number |
| Bigger | → | Greater |
| Smaller | → | Less |
| The smallest collection | → | The unit (one) |
| Putting collections together | → | Addition |
| Taking a smaller from a larger collection | → | Subtraction |

**Figure 86 Object Collection-Arithmetic metaphor.**

Metaphor provides operations are performed in the same way in every domain: add, difference, product, division.

# 7.5. Negotiation from physical to abstract domain

Human beings build abstract domain from physical domain. It means that abstract domain represents a short-cut in communication. However, when concepts appear, human beings relate to what it is known for them: physical domain.

This section is the description of this journey, concretely for negotiation process. Where the domain knowledge is conditioned by human capabilities of learning. In the next paragraphs is explained how reification rules are needed. They establish the connection between physical and non-physical entities. Afterwards, the negotiation process is developed from physical to abstract domain.

## 7.5.1. D&S Reification rules

Important fields of research have negated an ontological primitiveness to non-physical objects [256], because they are taken to have meaning only in combination with some other entity, i.e. their intended meaning results from a statement. For example, a norm, a plan, or a social role are to be represented as a (set of) statement(s), not as a concepts. This position is documented by the almost exclusive attention dedicated by many important theoretical frameworks (BDI agent model [257], theory of trust [258], situation calculus [259], formal context analysis [260]), to states of affairs, facts, beliefs, viewpoints, contexts, whose logical representation is set at the level of theories or models, not at the level of concepts or relations.

Moreover, recent work [256] addresses non-physical objects as first-order entities that can change, or that can be manipulated similarly to physical entities. This means that **many relations and axioms that are valid for physical entities can be used for non-physical ones as well**.

It is supported [261] the position by which non-physical entities can be represented both as theories/models and as concepts with explicit reification rules, and it is shared the following motivations:

There is an intrinsic desire to provide ***reifications*** in society and its differents domains, moreover there is a ***significant amount of terms convey concepts related to non-physical entities***, and such concepts seem to be tightly interrelated.

***Interrelations between theories*** are notoriously difficult to be manipulated because it is difficult to make an consensus about meaning concepts.

For many domains of application, it is necessary to face with partial theories and partial models that are explicated and/or used at various detail levels.

***Partiality and granularity*** are two more reasons to have some theories and models manipulated as first-order entities.

Natural languages are able to reify whatever fragment of (usually informal) theories and models by simply creating or reusing a noun. Once linguistically reified, a theory or a model (either formal or informal) enters a **life-cycle that allows agents to communicate even in presence of partial (or even no) information about the reified theory or model**. The Web contains plenty of examples of such creatures: catalogue subjects or topics, references to distributed resources, unstructured or semi-structured (but explicitly referenced) contents, such as plans, methods, regulations, formats, profiles, etc., and even linguistic elements and texts (taken independently from a particular physical encoding) can be considered a further example.

They feel entitled to say that representing ontological (reified) contexts is a difficult alternative to avoid, when so much domain-oriented and linguistic categorisations involve reification. However, they provide an explicit account of the contextual nature of non-physical entities and thus aim for a reification that accounts to some extent for the partial and hybrid structure of such entities.

From the logical viewpoint, any reification of theories and models provides a first order representation. From the ontological engineering view point, a straightforward reification is not enough, since the elements resulting from reification must be framed within an ontology, possibly built according to a **foundational ontology**.

Moreover, from a practical viewpoint, the actual import of theories and models (when they are used as concepts) into an ontology requires not only reification rules, but also mapping and inheritance rules. This partial and hybrid transformation allows an easy grasp and manipulation of reified theories and models.

D&S (Descriptions and Situations Ontology) (see section 4.5.3), in Figure 24 provides reification rules for the three basic categories of DOLCE ***(region, endurants and perdurants***), which are called parameters, roles and courses. D&S also defines a template, called S-Description ***(Situation Description)*** for modelling non-physical contexts such as ***views, theories, beliefs, norms***, etc. An important distinction is made in D&S between (the elements of) descriptions and (elements of) a particular model, also called state-of-affairs (SOA): elements of a ***SOA (regions, endurants and perdurants)*** may play the ***parameters, roles and courses of a description***, in which case the SOA is understood as a situation (case) for a particular description. However, the same SOA may be interpreted according to other, alternative descriptions. This captures an important feature of contexts, namely that multiple overlapping (or alternative) contexts may match the same world or model.

**Service descriptions** as **non-physical contexts** are ideally suited as *applications of D&S*. Descriptions of services can be considered as views from various perspectives on a series of activities that constitute the service for the various parties involved. In other words, service descriptions exhibit the same distinction between what is offered, expected or planned (descriptions, theories) and the elements that consist a particular model of the world.

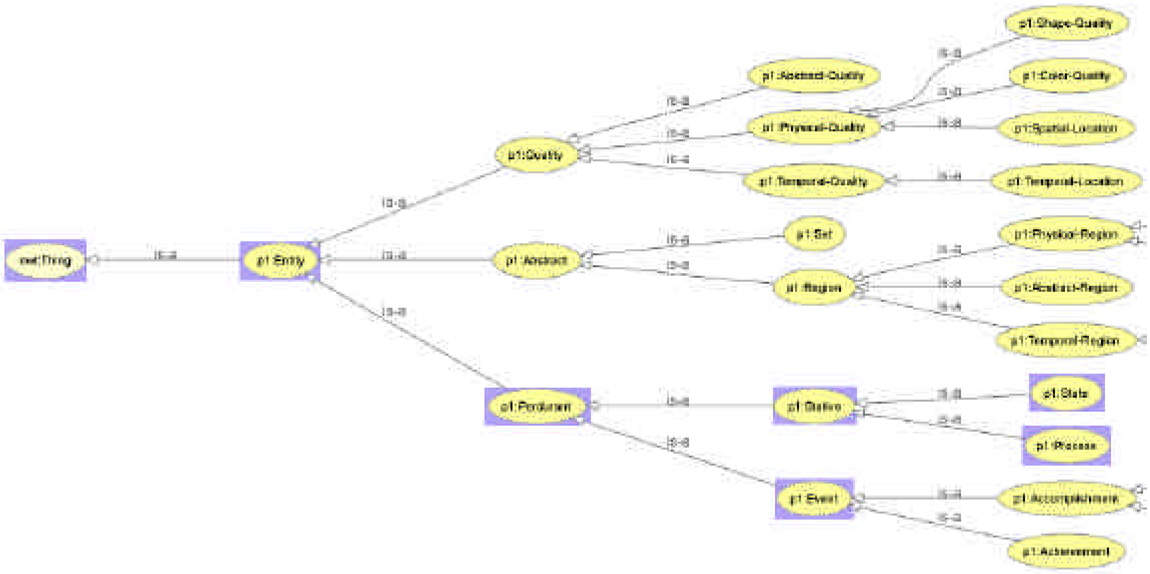



**Figure 87 DOLCE representation for ‘Event’, ‘State’ and ‘Process’**

### 7.5.2. Extending DOLCE and FrameNet

Human beings can measure spatial-time environment properties and it is possible to do by an artificial agent, too. It can measure space – time, i.e. wide, length, high and time and these tasks can be performed using for example:

- devices as microscopes (as tunnelling effect) to telescopes (as Hubble) depending on the scale where the measure is done for ***space***.
- atomic clocks for time which they have good precision for ***time***.

Variations in space along time are conceptualise as movement. Depending on how is this change: it means the variation space with time is known as: ***'celerity'***, ***'speed'*** or ***'velocity'*** depending on the exactly time point or interval, while the second derivative comes ***'acceleration'.*** Moreover, it can be understood as a space-time deformation.

In the framework provided by space-time also it is possible to define, apart of relations between space-time, entities that can be found in this framework, however before going deeper in what it is an entity, it is valuable to remember the relations existence among objects: operators.

Using operators as: addition, subtraction, product and division it is possible to show these relations (calculus as branch mathematics takes care of that). This was the main work from people as Newton or Leibniz, they were able to relate new concepts to old ones, for instance: defined integral as a sum of products, in other words, as a metaphor it can be seen as a sum of ***areas*** in physical domain.

Thus, the physical framework (space-time) plus variations on it as derivatives plus objects relations can be formalised using the same representation?

At this stage, concepts are expressed in scientific community using mathematical language. It allows interoperability in human beings. It is not only a symbolic language moreover has the capability to express also geometric features. Geometric algebra allows to formalize not only the symbols moreover their geometric meanings (see section 7.1.)

Moreover geometric algebra can be expressed in mathematical languages as Matlab, even from here it can be found in XML format.

However, going beyond this and trying to make it accessible to anyone and adding the semantic part. An expression in N3 format (it is an abbreviated form of RDF) has been used to call functions that perform operations inside this algebra with geometric meaning.

A previous version in 2-D for the operators has been developed in N3 format, see Table 36, Table 37, Table 38.

**Table 36. Namespaces definition**

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.

@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix list: <http://www.w3.org/2000/10/swap/list#>.
@prefix string: <http://www.w3.org/2000/10/swap/string#>.
@prefix v: <http://geoalg.org/vector#>.
```

**Table 37. 2D vectors geometric product**

```
# 2D vectors geometric product
#-----------------------------------------
{
    ?geoprod a v:GeometricProduct.
    ?geoprod.v:firstOperand a v:Vector.
    ?geoprod.v:secondOperand a v:Vector.
    ((?geoprod.v:firstOperand.v:components.rdf:first
        ?geoprod.v:secondOperand.v:components.rdf:first).math:product
     (?geoprod.v:firstOperand.v:components.rdf:rest.rdf:first
        ?geoprod.v:secondOperand.v:components.rdf:rest.rdf:first).math:product)
        math:sum ?inner.
    ((?geoprod.v:firstOperand.v:components.rdf:first
        ?geoprod.v:secondOperand.v:components.rdf:rest.rdf:first).math:product
     (?geoprod.v:firstOperand.v:components.rdf:rest.rdf:first
        ?geoprod.v:secondOperand.v:components.rdf:first).math:product)
        math:difference ?outer.
} =>
{
    ?geoprod v:innerProduct ?inner;
        v:outerProduct [a v:Bivector;
            rdf:value ?outer].
}.
```

**Table 38. Geometric algebra elements classes defintion**

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix v: <http://geoalg.org/vector#>.
@prefix : <#>.

v:Vector a owl:Class;
    rdfs:subClassOf [a owl:Restriction;
        owl:maxCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:components];
    rdfs:subClassOf [a owl:Restriction;
        owl:minCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:components].
```

```
v:Bivector a owl:Class;
    rdfs:subClassOf [a owl:Restriction;
        owl:maxCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty rdf:value];
    rdfs:subClassOf [a owl:Restriction;
        owl:minCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty rdf:value];
    rdfs:subClassOf [a owl:Restriction;
        owl:allValueFrom xsd:float;
        owl:onProperty rdf:value].
v:GeometricProduct a owl:Class;
    rdfs:subClassOf [a owl:Restriction;
        owl:minCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:firstOperand];
    rdfs:subClassOf [a owl:Restriction;
        owl:maxCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:firstOperand];
    rdfs:subClassOf [a owl:Restriction;
        owl:minCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:secondOperand];
    rdfs:subClassOf [a owl:Restriction;
        owl:maxCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:secondOperand];
    rdfs:subClassOf [a owl:Restriction;
        owl:minCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:innerProduct];
    rdfs:subClassOf [a owl:Restriction;
        owl:maxCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:innerProduct];
    rdfs:subClassOf [a owl:Restriction;
        owl:minCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:outerProduct];
    rdfs:subClassOf [a owl:Restriction;
        owl:maxCardinality "1"^^xsd:nonNegativeInteger;
        owl:onProperty v:outerProduct].

v:components a owl:ObjectProperty;
    rdfs:range math:List.
v:firstOperand a owl:ObjectProperty;
    rdfs:range v:Vector.
v:secondOperand a owl:ObjectProperty;
    rdfs:range v:Vector.
v:innerProduct a owl:DatatypeProperty;
    rdfs:range xsd:float.
v:outerProduct a owl:ObjectProperty;
    rdfs:range v:Bivector.

:geoprod a v:GeometricProduct;
    v:firstOperand [a v:Vector;
        v:components ("1" "1")];
    v:secondOperand [a v:Vector;
        v:components ("1" "0")].
```

**Table 39. Operation result**

```
@prefix : <http://geoalg.org/vector#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<#geoprod>    :innerProduct 1;
```

```
:outerProduct  [
   a :Bivector;
   rdf:value -1 ] .
```

Summarizing, from values that are got measuring coordinates (for instance, (1,1) and (1,0) with origin (0,0)), a space-time framework is build and the valuable point is that operators such as '***geometric product'*** can be expressed in terms from the previous measures without loosing the geometric meaning. As detachable example is the bivector, which encapsulates not only the information handed by a vector, moreover it has an area information about other two vectors. With this formalisation, operators are the same for any dimension as 'rotors' and can be applied to any object. Table 39 shows the operation result

Relations can be conceptualised as operations in the physical world using the geometric algebra. Objects, or in a generic case, entities also are able to use this representation. Neural nets are able to distinguish boundaries, and discriminate two states as '***in***' and '***out***', fuzzy logic can take care about the boundary itself [262], just putting the basis to distinguish entities, see Figure 30 and Figure 31. It can be the input to metaphor to containers and other Schemas as Figure 32.

Before going into negotiation process and detailed schemas, imagine a bullet that impacts on a pendulum , see Figure 88.

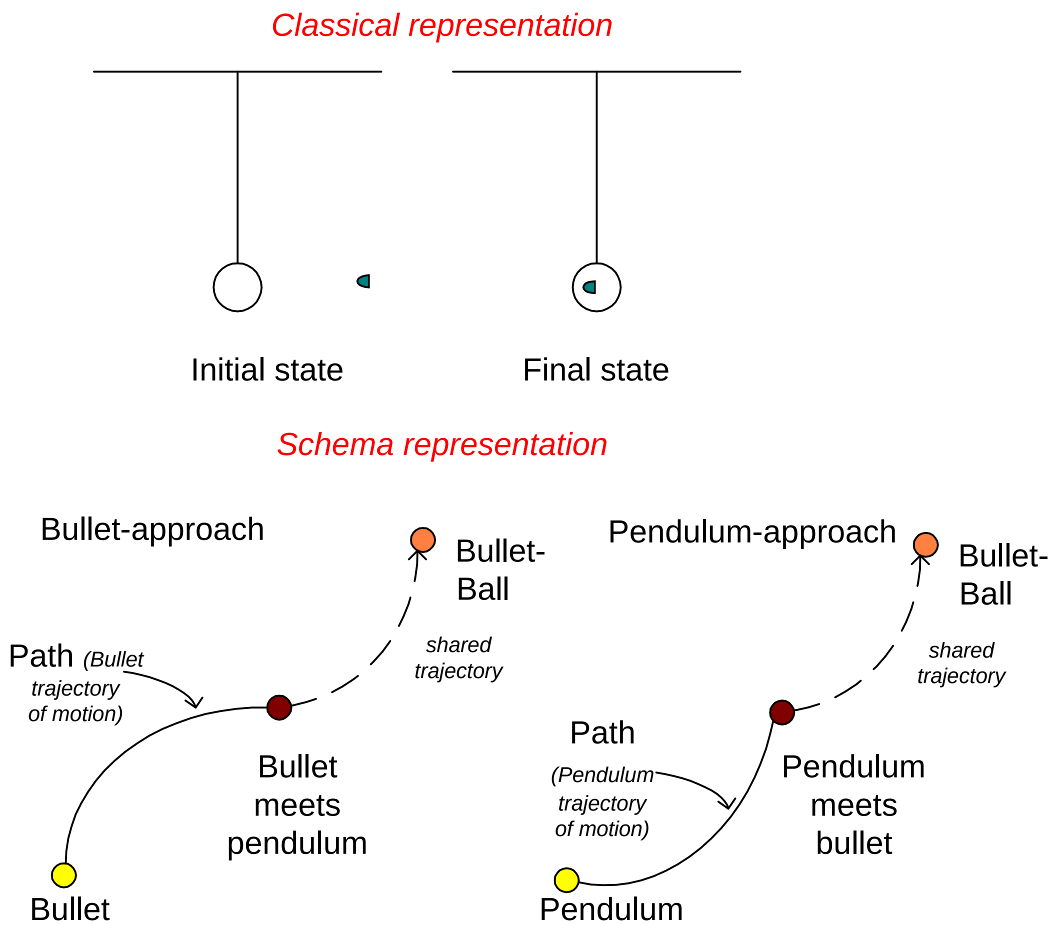



**Figure 88 Bullet meets pendulum**

When this problem it is conceptualised from Physics, there are the position of the objects related to an absolute frame of reference and moreover two conservative laws: energy law, mass*velocity law.

Moreover, if the problem is conceptualised from bullet or pendulum point of view, there are many different physical details such as speed, position … however a ***structure***, a ***schema*** comes clear.

A source (bullet or pendulum) makes its own way until something happens (a meeting) and both (bullet-pendulum) arrives until a final state. This schema is known as SPGS (Source-Path Goal Schema).

The ***Source-Path Goal Schema*** fits effectively to negotiation process. From cognitive linguistics there is a principal schema concerned with motion, and it has the following elements (or roles):

- Trajector that moves.
- A source location (the starting point).
- A goal – that is, an intended destination of the trajectory.
- A route.
- Actual trajectory of motion.
- Position of the trajectory at a given time.
- Direction of the trajectory at that time.
- Actual final location.

Figure 89 represents a generic geometrical expression for the **Source-Path-Goal-Schema** (SPGS), while Figure 90 is the representation in 1-D however SPGS is a topological schema (see Figure 91). It means that it is not constrained by geometry.

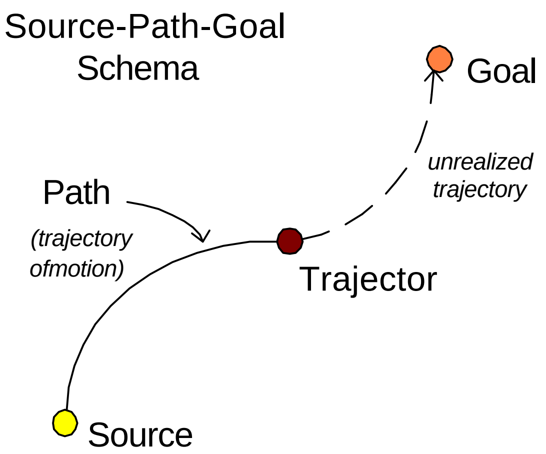



**Figure 89. Source-Path-Goal Schema.**

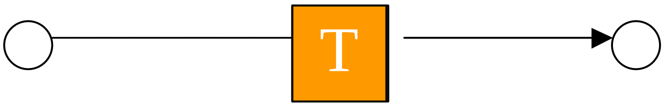



**Figure 90 SPGS in 1-D**

As it has been told, mathematics rely on axioms and logic. The logic is related to causality, there is a sequence on time where different steps follow: ***a Starting point, a Goal, a Trajectory, a Trajector and an Actual position.***

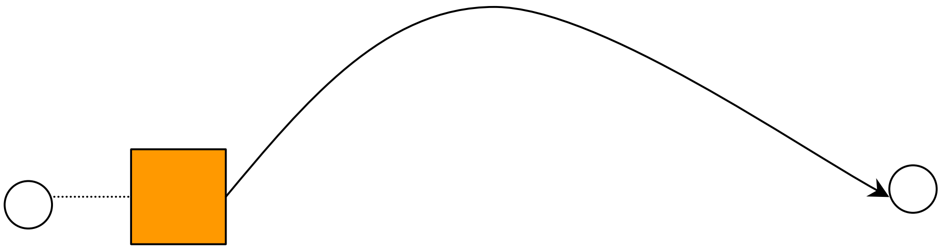


**Figure 91 SPGS is a topological schema**

Once, the elements and the process are identified as the operations themselves (algebra-geometric operations).This schema becomes very relevant because it is plausible, in fact, to map all these features to negotiation features.

Every object movement can be expressed in terms of geometric algebra and be understood as near as possible in terms of physical perceptions as space-time. The schema structure when applicable in different domains, allows understanding in one domain in terms of other totally different.

Another significant example is '*Romeo and Juliet*'[263]. **It shares the same structure as the bullet-pendulum but in an abstract domain**: it is a novel. However novels explain usually mixed physical world stories.

In fact, the title is "Romeo and Juliet: The Star crossed lovers".
It means that if they met, they have to remain together, as when the bullet met the pendulum they continue their motion together. Thus, when Romeo dies Juliet has to die too. Human beings often call it 'destiny'.

In natural language, verbs are always as the engine of the car. It provides the energy and behaves as a nucleus, agglutinating different roles around it. The following question is how to relate SPGS to natural language. Table 40 and Table 41 formalizes the mapping from verbs until the SPGS.

**Table 40. Thematic roles as subtypes the four types of participants combined with six kinds of verbs**

| | Initiator | Resource | Goal | Essence |
|---|---|---|---|---|
| Action | Agent (+volitional) Effect or (- | Instrument | Result, Recipient | Patient (changed) |

| | | | | |
|---|---|---|---|---|
| | volitional) | | | Theme (unchanged) |
| Process | Origin, Agent | Matter | Result, Recipient | Patient, Theme |
| Transfer | Agent, Origin | Instrument, Medium | Recipient, Experiencer | Theme |
| Spatial | Origin | Path | Destination | Location |
| Temporal | Start | Duration | Completion | Point InTime |
| Ambient | Reason | Manner | Aim (+volitional) Consequence (-volitional) | Condition |

**Table 41. Mapping SPGS ( Source Path Goal Schema ) to Negotiation Process.**

| | Initiator | Resource | Goal | Essence |
|---|---|---|---|---|
| Negotiation process | Event, Agent | Contract | Agreement | Agent as Actor/role |
| SPGS | Source | Trajector | Goal | Trajector |

Image Schemas are conceptualised on FrameNetI and FrameNetII (FNI, FNII)[247] as it was mentioned earlier. This project has thus far produced two databases: a collection of approximately 80 frames with frame descriptions, chosen to cover a broad range of semantic domains; and a hand-annotated dataset of about 50,000 sentences from the British National Corpus[264]. The databases document both syntactic and semantic behaviour of a wide variety of lexical items

These projects are based in a computational formalism that captures structural relationships among participants in a dynamic scenario. It represents the internal structure of FNI and FNII in terms of parameters for event simulations.

The current release of the FrameNet database defines a COMMERCE frame with frame elements including the familiar ***Buyer***, ***Seller***, ***Payment*** and ***Goods***, along with several other frame elements (FEs) needed to cover the data. The frame includes 10 verbs relevant to commercial transactions, for a total of 575 annotated sentences.

The COMMERCE frame is implicitly associated with a complex, dynamic network of interrelated ***events***, ***actions*** and ***participants***. It allows to distinguish a perspective-neutral description of a commercial transaction from the perspectivized situations described by particular verbs. The resulting event representation can be integrated with a simulation-based inference engine to account for differences in the interpretation of sentences like those in the annotated FrameNet data.

ECG is a constraint-based formalism similar in many respects to other unification-based linguistic formalism, includes formalisms for both schemas (conceptual representations) and constructions (conventionalised pairings of form and meaning[265]). It differs from other linguistically motivated proposals in that it is designed to support a model of language understanding in which utterances evoke a complex network of conceptual schemas that are then mental simulated in context to produce a rich set of inferences.

**schema** name
[**subcase of** name

**evokes**

$$\left[ \{ schema\ as\ local-name \} \right.$$

**roles**

$$\left\{ \begin{array}{l} local-role \\ local-role{:}restriction \\ local-role \leftrightarrow role \\ local-role \leftrightarrow role{:}restriction \end{array} \right\}$$

**constraints**

$$\left[ \left\{ \begin{array}{c} role \leftrightarrow role \\ phase :: condition \end{array} \right\} \right.$$

**Figure 92 Frame schema pattern**

It fits to Commercial Transactions very well [266] and it is an exceptional methodology for conceptualise the frame problem.

A relation between D&S, DOLCE and FrameNet is being studying. The idea is to build a bridge between to knowledge representations where the difference is that the first is purely symbolic and FrameNet tries to model something physical.

If the concepts such as '*event*', '*process*' or '*state*' are compared seeing Figure 85 and Figure 87, it can be established a similar classification in DOLCE and FrameNet, however when something more complex as different negotiation processes are involved, the amount information generated by several agents is difficult to manage.

Initiatives as UML or Petri Nets represent processes conceptually. When this knowledge arrives to the user several languages are useful for represent as: UML or Petri Nets (see Figure 94). Existing multi-criteria decision making algorithms applied to already scored Petri-Net models would provide a significant stepping stone towards design of efficient negotiation protocols.

What it is necessary is to add something more, a layer for schemas going from physical domain to abstract domain and vice versa using as a basis the maximum information taken from physical world.

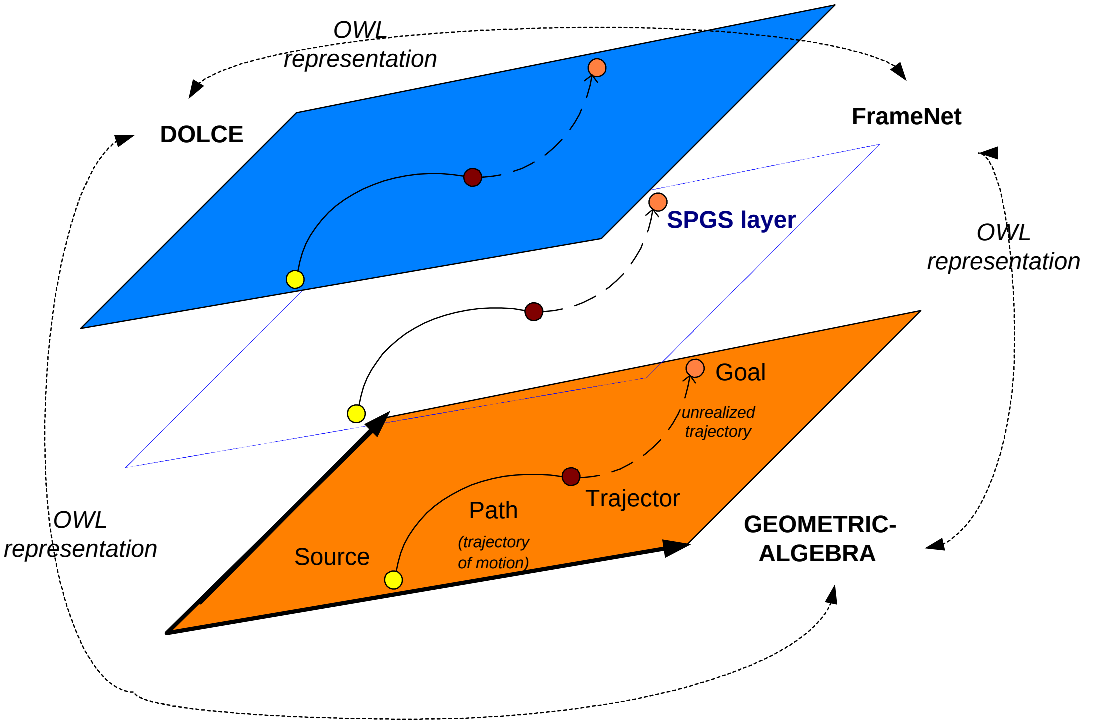



**Figure 93 SPGS layer in relation to other domains as DOLCE, FrameNet and Geometric-Algebra.**

The SGPS simplicity allows to show how DOLCE and FrameNet can work together formalized in OWL and using relations to physical world with Geometric Algebra.

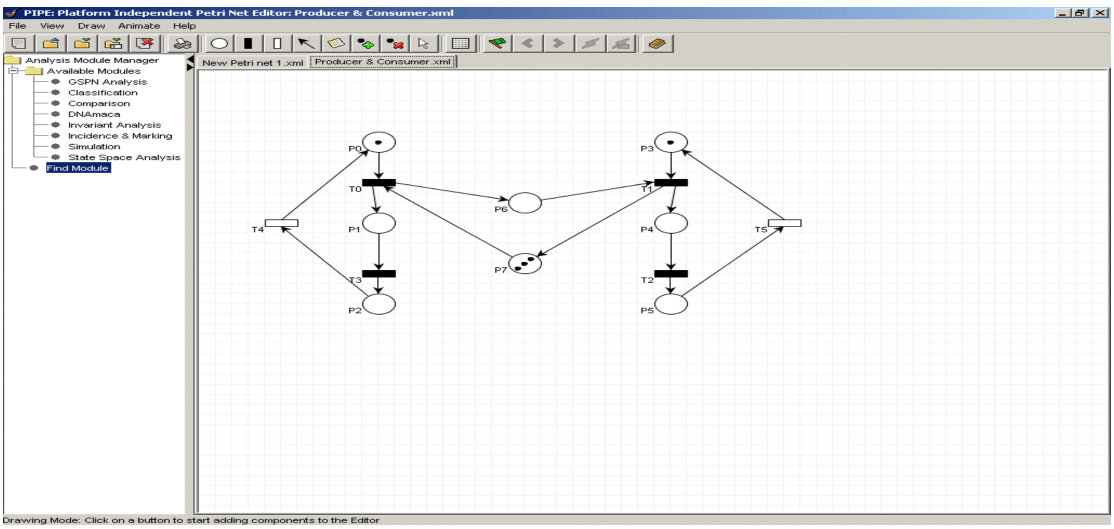



**Figure 94 Producer and Consumer model in XML represented by a Petri Net.**

Figure 95 shows the relation SGPS to OWL-S and geometric algebra while Figure 96 shows the relation SGPS to DOLCE and geometric algebra. In both cases geometric algebra provides a subtract to relate physical properties over a pattern formalised in SGPS that extends itself in our cognitive level.

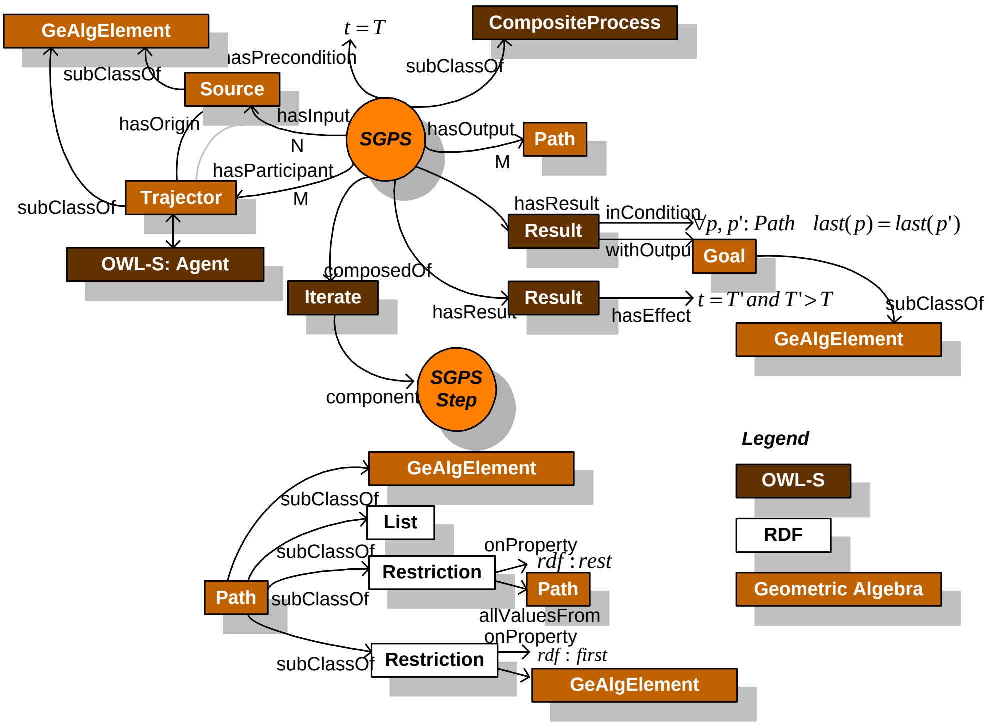



**Figure 95 Relating SGPS to OWL-S and Geometric Algebra**

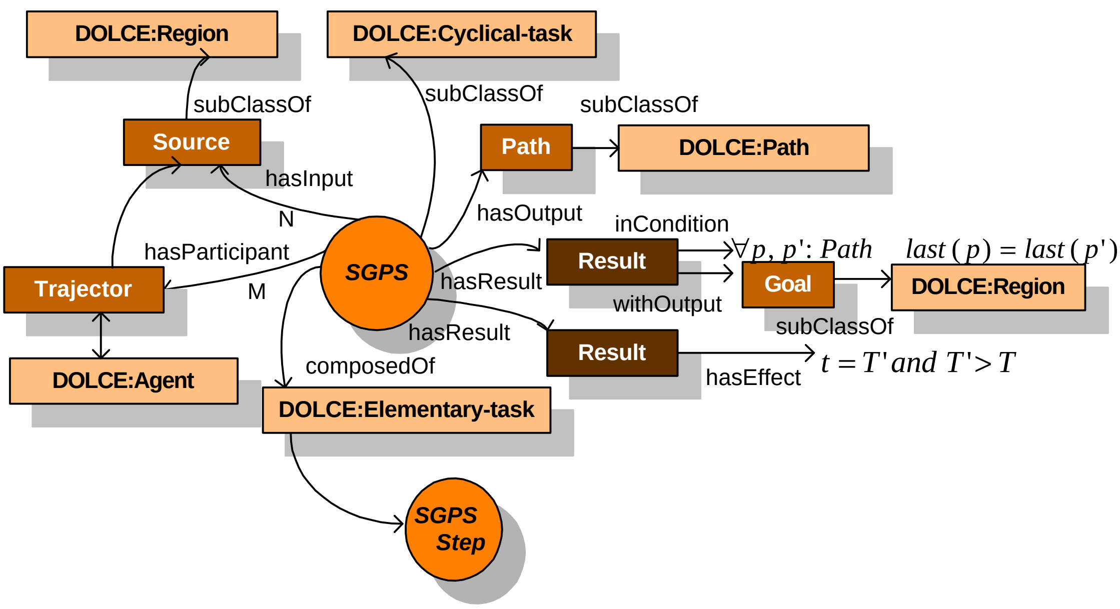



**Figure 96 Relating SGPS to DOLCE**

## 7.6. Conclusions

Upper ontologies describe concepts in abstract domain. Two different projects as FrameNet and DOLCE are presented. The consequence is that it is not evident how to map concepts from one project to another.

A new methodology was designed using some aspects taken from FrameNet and connecting also to DOLCE.

FrameNet has a cognition model that connects physical aspects to abstract aspects. Physical aspects were not so developed as abstract aspects. Thus, the labour has done describing physical aspects. The language that was choosen was Geometric Algebra because it is **coordinate-free** and relates **geometric to symbolic properties**.

Thus, a methodology has been developed connecting physical properties (Geometric Algebra) to abstract domain, i.e. Semantic Web.

The result is that artificial agents have tools with the purpose of making new knowledge with themselves from physical experiences using a cognition model based on human beings.

# 8. Final Conclusions and future work:

Conclusions have been developed in every chapter. Contributions in different topics have been specified:

***Semantic Web Architecture (Chapters 5,6 and 7)***

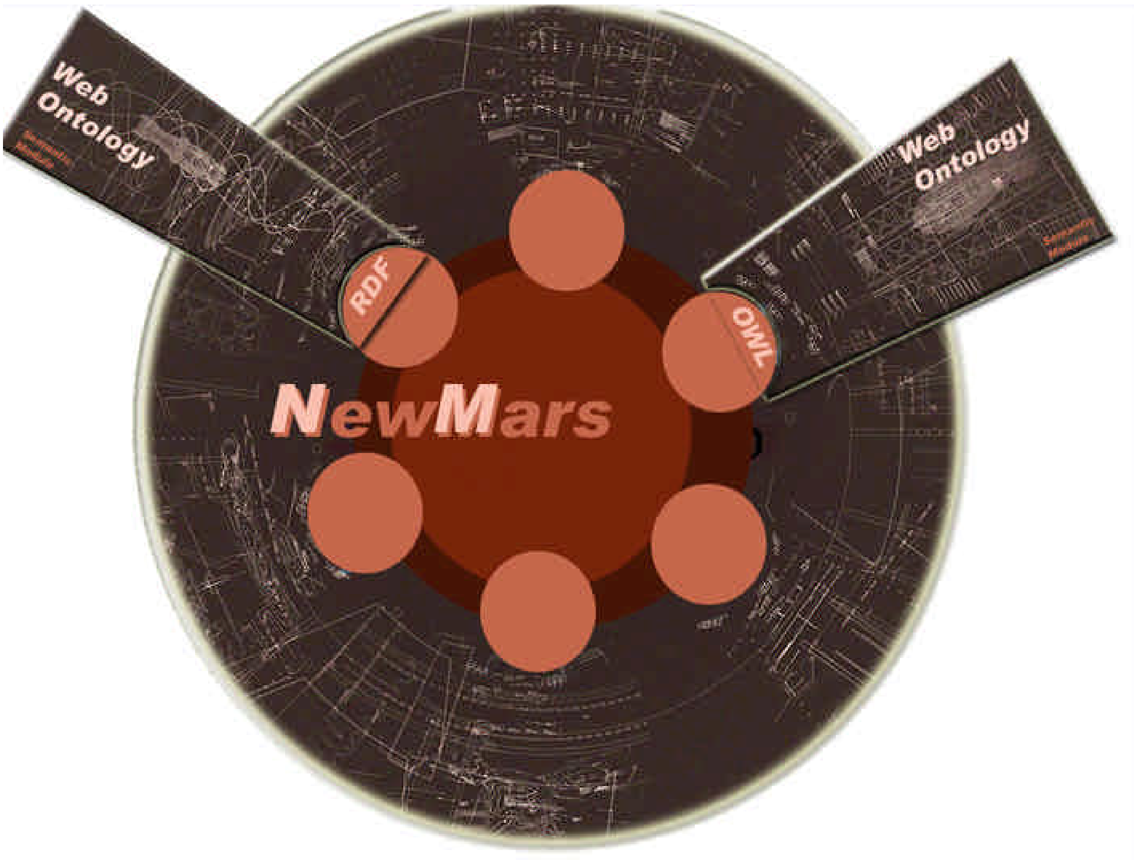



**Figure 97. Semantic Web Architecture (SWA) schema**

A Semantic Web Architecture has been designed, developed and tested in the context of IPR management, concretely in NewMars [160] and AgentWeb[161] projects. It has been accepted in an international conference (ODBASE 04).

> García, R., Gil, R. and Delgado, J. ***Intellectual Property Rights Management using a Semantic Web Information System***
> 3th International Conference on Ontologies, Databases and Applications of Semantics, ODBASE 2004
> Lecture Notes in Computer Science, Vol. 3291, pp 689 - 704. Springer-Verlarg, 2004. ISBN 3-540-23662-7

***Multi-Agent Systems (Chapter 5)***

An exhaustive analysis has been made in Multi-Agent Systems and two papers were accepted, one focusing on the mobile agents architecture and the other in the mobile agents customization in the process of negotiation.

Gil R., García R., Delgado J. “***Delivery context negotiated by mobile agents using CC/PP”***. Mobile Agents for Telecom Applications, (MATA’03).
Lecture Notes in Computer Science, Vol. 2881, pp 99 - 110. Springer-Verlarg, 2003

Delgado, J.; Gallego, I.; Garcia, R. and Gil, R.
“***An architecture for negotiation with mobile agents***”.
Mobile Agents for Telecom Applications (MATA'02).
Lecture Notes in Computer Science, Vol. 2521, pp 21-31. Springer-Verlag, 2002

***Ontologies and ad hoc applications (Chapter 6)***

Ontologies have been used to model agents’ knowledge. In order to provide a model of the Semantic Web as real as possible, a deep statistical analysis of it has been made. It reveals that the Semantic Web behaves as a complex system and shares some properties with them. This can be used to study new ways of designing semantics-enabled applications. In this sense, Semantic Web can be modelled as a whole system and *macroscopic* behaviour can be established. The **First International Semantic Web Conference** was chosen to test the ontology designed for the Semantic Web Architecture.

Delgado, J.; Gallego, I.; García, R. and Gil, R.
“***IPROnto - Intellectual Property Rights Ontology***”.
Poster in the First International Semantic Web Conference (ISWC), 2002

Also, the DMAG is involved in activities related to Standardization in MPEG-21

For instance, DMAG made a proposal for MPEG-21 Rights Data Dictionary and Rights Expression Language call proposals. The group keeps a web page with the material and tools developed in MPEG-21 framework[267].

In order to validate the scalability and applicability of the architecture, several activities related to Ad-hoc Applications in Semantic Web and complexity, were done in 2003 , at ENST-Bretagne (Brittany) collaborating with Dr. Zièd Choukair. As a result the work was send to the biggest conference **relating to World Wide Web.**

Rosa Gil, Jaime Delgado, Zièd Choukair. ***Towards the composition of Ad hoc B2B Applications: Semantics, Properties and Complexity management***. The 13th World Wide Web Conference, Poster at Workshop on Application Design, Development and Implementation Issues in the Semantic Web, May 2004

Gil, R., García, R. and Delgado, J.   ***Measuring the Semantic Web***
"Semantic Web Challenges for Knowledge Management: towards the Knowledge Web",SIGSEMIS Bulletin Vol 1, Issue 2, pp 69 - 72. July 2004

### *A new negotiation representation (Chapter 7)*

The last part manages the possibility to give the agents the opportunity of '*learn to learn*'. In order to achieve it, some approaches have been discussed: DOLCE, D&S and cognitive linguistics using 'image-schemas' to connect them to the semantic web, geometric algebra allow to connect symbolic (algebra) and semantic (geometry understood as physical meaning) as a last step. Finally, this part outcomes how artificial agents using this methodology can discover new knowledge by themselves from physical experiences using a cognition model based on human beings. Although this work is still in progress, some results are presented applied to negotiation process.

# 9. Appendix A: Agents

What is an agent?

As a general definition, an *agent* is an animate entity that is capable of doing something on purpose. That definition is broad enough to include humans and other animals, the subjects of verbs that express actions, and the computerized robots and softbots. Linguistically, an agent is represented by the subject of an active verb. Socially, an agent is an animate being that takes responsibility for its actions in the world. But it depends on other words whose meanings are just as problematical: *animate, capable, doing,* and *purpose*. The task of defining those words raises questions that involve almost every other aspect of ontology [268].

*Animate*. Literally, an animate entity is one that has an *anima* or soul. But *anima* is the Latin translation of Aristotle's word *psychê*, which had a much broader meaning than the English word *soul*. Aristotle defined a hierarchy ranging from a vegetative psyche for plants to a rational psyche for humans. The first question is whether Aristotle's hierarchy of psyches can accommodate the modern robots and softbots.
*Capable*. The agent of a verb plays that role only as long as the action persists, but an entity can also be considered an agent if it has the power to perform some action whether or not it actually does. Formalizing that notion of power raises questions about modality, potentiality, dispositions, and counterfactuals that have been discussed in philosophy for centuries.
*Doing*. The verb *do* sounds as simple as two other little verbs *be* and *have*. But like those verbs, its dictionary entry has one of the largest number of senses of any word in the English language. A common feature of all those senses is causality and purpose: some agent for some purpose causes some process to occur. This feature not only creates a cyclic dependency of *doing* on *agent*, it also introduces the notions of causality, process, and occurrence.
*Purpose*. In the top-level ontology, purpose is defined as an intention of some agent that determines the interaction of entities in a situation. That is consistent with the definition of an agent as an entity that does something on purpose, but the circularity makes it impossible to give a closed-form definition of either term.

In practice, agents are on the top of programming, before it, in 1974 there was structured programming and in 1982 object oriented programming was born.

- Biology:
    - John von Neumann: self-reproducing automata (‘50s)
    - John Conway: game of Life (‘60s)
    - Chris Langton: artificial life (late ‘80s)
- Social science:
    - Simon, March and Cyert: the ‘behavioral school’ and simulation of few agent systems (‘50s and ‘60s)
    - Tom Schelling: tipping model of segregation (late ‘60s)
- Computer science:

- o artificial intelligence (AI)
- o robotics
- o distributed AI (DAI)
- o object-oriented programming (OOP)

The adjectives in relation to the environment that describe agents are:

*Autonomy*: agents encapsulate some state (that is n ot accessible to other agents), and make decisions about what to do based on this state, without the direct intervention of humans and others.
*Reactivity*: agents are situated in an environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps many of these combined), are able to perceive this environment (through the use of potentially imperfect sensors), and are able to respond in a timely fashion to changes that occur in it.
*Pro-activeness*: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.
*Social ability*: agents interact with other agents (and possibly humans) via some kind of agent-communication language, and typically have the ability to engage in social activities (such as cooperative problem solving or negotiation) in order to achieve goals.

Moreover they could be intelligent, mobile and adaptable depending on their duties.

So, if we trace a line showing the degree of symbolic representations, in one extreme there will be purely cognitive agents where only symbolic representations are present, in the other hand, purely reactive agents with no representations at all.

Two great behaviors models describe agents: *teleonomic* where their behaviors is towards specific objects from other agents and *reflex* when the tendencies come from the environment. [269].

Agents have…:

Internal data representations (*memory* or *state*)
Means for modifying their internal data representations
(perceptions)
Means for modifying their environment (*behaviors*)

Depending on the degree of abstraction or realization there are different approaches represented by diverse languages. As communication languages there are KQML (Knowledge Query and Manipulation Language) [270] or FIPA-ACL[271], behavior description languages as Petri Nets [272], knowledge representation language as Semantic Nets (as DAML,OWL). Above them there are the specification languages and below implementation languages managed by expert systems.

# 10. Appendix B: Geometric Algebra applications

Clifford algebras are well-known to pure mathematicians. Geometric algebra is a coordinate-free approach to geometry. The elements are coordinate-independent objects called multi-vectors which can be multiplied together using a geometric product. The system deals with rotations in n-dimensional space very efficiently. Since geometric algebra has already been successfully applied to many areas of mathematical physics and engineering:

- Computer Vision:

Binocular and tri-nocular geometry. Affine and projective reconstruction. Trifocal tensor and invariants for matching, object recognition and image coding.

- Lie Groups and Lie Algebras:

In the geometric algebra frame for the computation of differential invariants, affine structure of image sequences and visual symmetries for visual guided robot navigation.

- Robotics:

The use the 4D algebra of the motors for 3D kinematics. The motor algebra together with fuzzy logic are being used for geometric reasoning useful for object avoidance and navigation. In terms of motors we represent points, lines and planes and their motion. These entities and their spatial invariants are being used for manoeuvre.
Hand-eye calibration using motors for a binocular head on a mobile robot. The control of a binocular head is being formulated as a problem of multi-vector control. Related controllers, filters and estimators are extended for multidimensional control.

- Neural Computing:

Neural learning is an issue of geometric learning. Standard MLP and RBF neural nets and the back-propagation training rule in the geometric algebra Framework have been modelled. The nets show a much reasonably performance during learning and in the generalization due to the geometric product, the avoidance of redundant components and the coordinate independence of the data coding. Thus the learning in these architectures are improving using the sub-manifold intrinsic dimensionality.

Following this lines, this is a briefly explanation of 2D geometric algebra, for our purposes it is not necessary to wide the scope to 3D. However, it is direct to do it[273].

VECTOR PLANE AND THE COMPLEX NUMBERS

A ***point*** is conceived (but not defined) as a geometric element without extension, infinitely small, that has position an is located at a certain place on the plane.

(capabilities: measuring distances, distinguishing objects)

A ***vector*** is defined as an oriented segment, that is, a piece of a straight line having length and direction. A vector has no position and can be translated anywhere. Usually it is called a free vector. Also a vector represents a translation from one point to other because if a vector is placed, the end of it at a point determines another point.

(capabilities: measuring distances, distinguishing objects, sequence)

Initial properties: It is not relevant the order in which the addition operation is performed

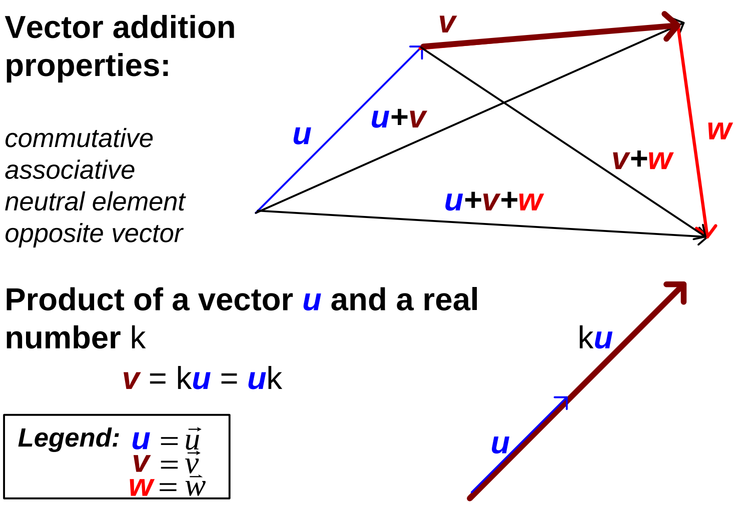



**Figure 98. Vector addition properties and product of a vector and a real number**

Product of two vectors (***geometric product***)

Properties:

Associative:

$$\vec{u}(\vec{v}+\vec{w}) = \vec{u}\vec{v}+\vec{u}\vec{w}$$

The square of a vector must be equal to the square of its length

$$\vec{u}^{\,2} = |\vec{u}|^2$$

Mixed associative property: (k,l *real numbers*), ($\vec{u}\vec{v}$ vectors)

$$k(\vec{u}\vec{v}) = (k\vec{u})\vec{v} = k\vec{u}\vec{w}$$
$$k(l\vec{u}) = (kl)\vec{u} = kl\vec{u}$$

The product of three vectors is associative:

$$\vec{u}(\vec{v}\vec{w}) = (\vec{u}\vec{v})\vec{w} = \vec{u}\vec{v}\vec{w}$$

From this property follows the **permutative property**: every vector can be permuted with a vector located two positions farther in a product, although it does not commute with the neighbouring vectors.

The permutative property implies that any pair of vectors in a product separated by an odd number of vectors can be permuted. The permutative property is characteristic of the plane and it is also valid for the space whenever the three vectors are coplanar. This property is related with the fact that the product of complex numbers is commutative.

$$\vec{a}\vec{b}\vec{c}\vec{d} = \vec{a}\vec{d}\vec{c}\vec{b} = \vec{c}\vec{d}\vec{a}\vec{b} = \vec{c}\vec{b}\vec{a}\vec{d}$$

Curiously, it follows:

| | |
|---|---|
| **Case A**: $\vec{a} \perp \vec{b}$<br><br>The product of two ***perpendicular*** vectors is ANTICOMMUTATIVE | Demonstration:<br><br>If vectors are orthogonal then Pythagorean theorem applies:<br>$if\ \vec{a} \perp \vec{b} \Rightarrow \mathrm{c}^2 = \mathrm{a}^2 + \mathrm{b}^2 \Rightarrow ab + ba = 0 \Rightarrow ab = -ba$ |
| **Case B**: $\vec{a} \parallel \vec{b}$<br><br>The product of two ***proportional*** vectors is COMMUTATIVE | Demonstration:<br><br>One vector is the other multiplied by a real number, k.<br>$if\ \vec{a} \parallel \vec{b} \Rightarrow \vec{b} = k\vec{a}, k\ \mathrm{real} \Rightarrow ab = aka = kaa = ba$ |

If it is not case A neither B, one vector can be discomposed in the perpendicular direction of the other and in the same as it.

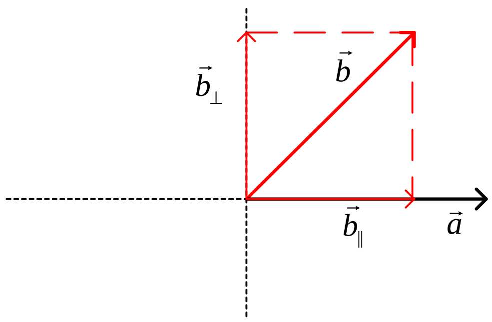



**Figure 99. Vector $\vec{b}$ discomposes in two vectors, one perpendicular to $\vec{a}$ that it is called $\vec{b}_\perp$ and other in the same direction as $\vec{a}$, $\vec{b}_\parallel$.**

$$\vec{a}\vec{b} = \vec{a}(\vec{b}_\parallel + \vec{b}_\perp) = \vec{a}\vec{b}_\parallel + \vec{a}\vec{b}_\perp$$

Term $\vec{a}\vec{b}_{\parallel}$:

The product of one vector by the proportional component of the other is called the ***inner product*** (also *scalar product*) and noted by a point '·'. Taking into account that the projection of $\vec{b}$ onto $\vec{a}$ is proportional to the cosine of the angle between both vectors:

$$\vec{a}\cdot\vec{b} = \vec{a}\vec{b}_{\parallel} = |\vec{a}||\vec{b}|\cos a = \text{real number}$$

The result is a ***scalar***

**Physical example:** "Work made by a force acting on a body is the **inner product** of the force and the walked space"

Term $\vec{a}\vec{b}_{\perp}$:

The product of one vector by the orthogonal component of the other is called the ***outer product*** (also *exterior product*) and it is noted with the symbol ^.

$$\vec{a}\wedge\vec{b} = \vec{a}\vec{b}_{\perp} \Rightarrow |\vec{a}\wedge\vec{b}| = |\vec{a}\vec{b}_{\perp}| = |\vec{a}||\vec{b}||\sin a|$$

$$\vec{a}\wedge\vec{b} = -\vec{b}\wedge\vec{a}$$

The result is not a vector, it is a ***bivector*** which is interpreted geometrically as a ***directed area element in the plane*** spanned by:
$\vec{a}$ and $\vec{b}$

**Physical example:** "Angular momentum is an outer product"

Geometrical algebra of the vectorial plane:

The set of all the vectors on the plane together with the operations of vector addition and product of vectors by real numbers is a two-dimensional space usually called the vector plane $V_2$. The geometric product generates new elements (the complex numbers) not included in the vector plane. So, the geometric (or Clifford) algebra of a vectorial space is defined as the set of all the elements generated by products of vectors for which the geometric product is an inner operation. The geometric algebra of the Euclidean vector plane is usually noted as $Cl_{2,0}(\Re)$ or simply as $Cl_2$. Making a parallelism with probability, the same sample space O is the set of elemental results of a certain random experiment. From the sample space O , the union $\bigcup$ an the intersection $\bigcap$ generate the Boole algebra *A(O)*

Inner and outer products can be written using the geometric product:

$$\vec{a}\cdot\vec{b} = \frac{\vec{a}\vec{b} + \vec{b}\vec{a}}{2}$$

$$\vec{a}\wedge\vec{b} = \frac{\vec{a}\vec{b} - \vec{b}\vec{a}}{2}$$

BASE OF VECTORS FOR THE PLANE

Linear combination of two vectors

Every vector on the plane $\vec{w}$ is always a linear combination of two independent vectors $\vec{u}$ and $\vec{v}$:

$\vec{w} = k\vec{u} + l\vec{v}$ k,l real

Because of this, the plane has dimension equal to 2. In order to calculate the coefficients of linear combination k and l, the vector $\vec{w}$ is multiplied by $\vec{u}$ and $\vec{v}$, two equations are obtained and the following results arise:

$$k = \frac{\vec{w}\wedge\vec{v}}{\vec{u}\wedge\vec{v}} \quad , \quad l = \frac{\vec{u}\wedge\vec{w}}{\vec{u}\wedge\vec{v}}$$

The resolution of a vector as a linear combination of two independent vectors is a very frequent operation and also the foundation of the coordinates method.

Base and components

Any set of two independent vectors $\{\hat{e}_1, \hat{e}_2\}$ can be taken as a base of the vector plane. Every vector u can be written as linear combination of the base vectors:

$$\vec{u} = u_1\hat{e}_1 + u_2\hat{e}_2$$

$u_1$ y $u_2$ are the coefficients of this linear combination, also called components of the vector in this base.

$$\vec{u}\vec{v} = (u_1\hat{e}_1 + u_2\hat{e}_2)(v_1\hat{e}_1 + v_2\hat{e}_2) = u_1v_1\hat{e}_1^{\,2} + u_2v_2\hat{e}_2^{\,2} + u_1v_2\hat{e}_1\hat{e}_2 + u_2v_1\hat{e}_2\hat{e}_1$$

$$\vec{u}\vec{v} = u_1v_1|\hat{e}_1|^2 + u_2v_2|\hat{e}_2|^2 + u_1v_2\hat{e}_1\hat{e}_2 - u_2v_1\hat{e}_1\hat{e}_2$$

Orthonormal bases

Any base is valid to describe vectors using components, although the orthonormal bases, for which both $\hat{e}_1$ and $\hat{e}_2$ are unitary and perpendicular are the more convenient and suitable:

$$\hat{e}_1 \perp \hat{e}_2, \qquad |\hat{e}_1| = |\hat{e}_2| = 1$$

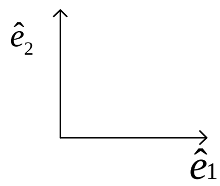



**Figure 100 Canonical vector base**

For every orthonormal base:

$$\hat{e}_1\hat{e}_2 = -\hat{e}_2\hat{e}_1, \qquad \hat{e}_1^{\,2} = \hat{e}_2^{\,2} = 1$$

The product $\hat{e}_1\hat{e}_2$ represents a square of unity area. The square power of this product is equal to –1, formally is represented by '***i***' in mathematics and '***j***' in Physics:

$$(\hat{e}_1\hat{e}_2)^2 = \hat{e}_1\hat{e}_2\hat{e}_1\hat{e}_2 = -\hat{e}_1\hat{e}_1\hat{e}_2\hat{e}_2 = -1$$

For an orthonormal base, the ***geometric product*** of two vectors becomes:

$$\vec{u}\vec{v} = u_1v_1 + u_2v_2 + (u_1v_2 - u_2v_1)\hat{e}_1\hat{e}_2$$

If $\{\hat{e}_1, \hat{e}_2\}$ is the canonical base of the vector plane *$V_2$, its geometric algebra* is defined as the vector space generated by the elements $\{1, \hat{e}_1, \hat{e}_2, \hat{e}_1\hat{e}_2\}$ together with the geometric product, so that the *geometric algebra* $Cl_2$ has dimension four. The unitary area $\hat{e}_1\hat{e}_2$ is usually noted as $\hat{e}_{12}$. Due to the associative character of the geometric product, the geometric algebra is an associative algebra with identity. The complete table for the geometric product is the following.

| | 1 | $\hat{e}_1$ | $\hat{e}_2$ | $\hat{e}_{12}$ |
|---|---|---|---|---|
| 1 | 1 | $\hat{e}_1$ | $\hat{e}_2$ | $\hat{e}_{12}$ |
| $\hat{e}_1$ | $\hat{e}_1$ | 1 | $\hat{e}_{12}$ | $\hat{e}_2$ |
| $\hat{e}_2$ | $\hat{e}_2$ | $-\hat{e}_{12}$ | 1 | $-\hat{e}_1$ |
| $\hat{e}_{12}$ | $\hat{e}_{12}$ | $-\hat{e}_2$ | $\hat{e}_1$ | -1 |

**Figure 101 Table for the geometric product**

In the complex plane, the complex numbers are represented taking the real component as the abscissa and the imaginary component as the ordinate. The vectorial plane differs from the complex plane in the fact that the vectorial plane is a plane of absolute directions whereas the complex plane is a plane of relative directions with respect to the real axis, to which we may assign any direction. As explained in more detail in the following chapter, the unitary complex numbers are rotation operators applied to vectors. The following equality shows the ambivalence of the ***Cartesian coordinates*** in the ***Euclidean plane***:

$$\hat{e}_1(x + y\hat{e}_{12}) = x\hat{e}_1 + y\hat{e}_2$$

Due to careless use, often the complex numbers have been improperly thought as vectors on the plane, furnishing the confusion between the complex and vector. It will argued that this has been very fruitful, but this argument cannot satisfy geometers, who search the fundamentals of the geometry. On the other hand, some physical magnitudes of a clearly vectorial kind have been taken improperly as complex numbers, specially in quantum mechanics. The relation between vectors and complex numbers is stated in the following way: If $\vec{u}$ is a fixed unitary vector, then every vector $\vec{a}$ is mapped to a unique complex $z = z_R + z_I$ fulfilling

$\vec{a} = \vec{u}z$ with $\vec{u}^2 = 1$

Also other vector is mapped to a complex number $t = t_R + t_I$:

$$\vec{b} = \vec{u}t$$

The ***outer*** and ***inner product*** of the vectors $\vec{a}$ and $\vec{b}$ can be written now using the complex numbers z and t:

$$\vec{a}\cdot\vec{b} = \frac{\vec{a}\vec{b} + \vec{b}\vec{a}}{2} = (z_R t_I + z_I t_R)$$
$$\vec{a}\wedge\vec{b} = \frac{\vec{a}\vec{b} - \vec{b}\vec{a}}{2} = (z_R t_I - z_I t_R)\ \hat{e}_{12}$$

Complex quantities must be distinguished from vectorial quantities, and relative directions (complex numbers) from absolute directions (vectors). An example is the alternating current. The voltage V and intensity I in an electric circuit are continuously rotating vectors. The energy E dissipated by the circuit is the inner product of both vectors, $E = V \,.\, I$. The intensity vector can be calculated as the geometric product of the voltage vector multiplied by the inverse of the impedance $I=V\cdot Z^{-1}$

Moreover, Geometric Algebra is a mathematical formalism that affords a direct representation of quantum mechanics in ordinary space-time: ***Geometric-Algebraic Quantum Mechanics (GAQM)***. The move backwards from Hilbert space to space-time representation, of course, has a price. The representing objects are of a more general kind than vectors and the product between them is non-commutative. Nevertheless all these objects are directly *interpretable geometrically*, hence GAQM might be called a *geometrization* of quantum mechanics. This label is meant to highlight the fact that the formalism, being mathematically equivalent to the orthodox Hilbert space version, makes visible an additional geometrical structure of the theory that might be exploited for interpretational purposes. For instance, as example two features are enunciated:

In GAQM the Schröedinger equation is a limit of the Dirac equation for small velocities and vanishing magnetic fields which is not the case in the orthodox formalism (***due to the fact that there is no smooth transition between the Pauli and Schröedinger equations***). It has been claimed that this shows the hidden presence of spin in the Schröedinger theory and forces an entirely new interpretation of that theory.

GAQM represents the quantum mechanics of spin in a way that is both geometrical and elegant (coordinate-free). It thus comes as a surprise that the GAQM representation of a two-particle spin system introduces much additional complexity: The geometric stand-ins for the complex unit are system-relative and must be identified by an explicit assumption missing in the orthodox formalism. This, however, should be seen as a virtue, not a vice of the approach. It is argued that here a hidden assumption about the geometric relation of two quantum-mechanical systems can be disclosed.

# IPROnto - Intellectual Property Rights Ontology

http://dmag.upf.es/ontologies/ipronto

**Jaime Delgado (1) , Isabel Gallego (2) , Roberto García (1) , Rosa Gil (1)**
(1) Universitat Pompeu Fabra (UPF), Departament de Tecnologia,
Pg. Circumval.lació 8, E-08003 Barcelona, Spain

(2) Universitat Politècnica de Catalunya, Departament d'Arquitectura de Computadors,
Campus Nord, Mòdul D6, Jordi Girona 1-3, E-08034 Barcelona, Spain

# Towards the composition of Ad Hoc B2B Applications : Semantics, Properties and Complexity Management


Rosa Gil(*)(**), Jaime Delgado(**) & Zièd Choukair(*)

(**)Universitat Pompeu Fabra (UPF)
Departament de Tecnologia, Pg. Circumval.lacio 8, E 08003
Barcelona, Spain
rosa.gil@upf.edu, jaime.delgado@upf.edu

(*)Ecole Nationale Supérieure des Télécom-munications de Bretagne (ENSTB)
Laboratoire d'Informatique des TéléCommuni-cations (LIT)
Technopole Brest- Iroise- BP 832-29285 - Brest Cedex- France
zied.choukair@enst-bretagne.fr



WWW2004

Workshop on Application Design, Development and Implementation Issues in the Semantic Web


*" A fisherman goes from USA to Canada and he wants an ad hoc application that builts itself depending on the environment. Weather forecast could be one application "*

# Semantics

## Step 1: Semantic application components

- Application components that are services or part of services have to be ***semantically described*** to achieve ad hoc applications. We consider there is no knowledge 'a priori' about where the components are.

- Semantically related: it means that application components are members of an **ontology**. Well-defined ontologies are related to 'upper ontologies', that are the common point to establish new relations and discover new knowledge, other ontologies describing new application components.

Languages: DAML, RDF & OWL
Web Services: DAML-S
Ontologies: TIME, WEATHER, GIS
http://www.isi.edu/~pan/damltime/time-entry.owl
http://www.mindswap.org/~mhgrove/weather-ont.daml
http://www.daml.org/2001/06/map/map-ont

## Step 2: Customer and device profile

Depending on the customer and device profile (*), an ontology is chosen, a bouquet fo services is ready (platform COCTELS)(**)

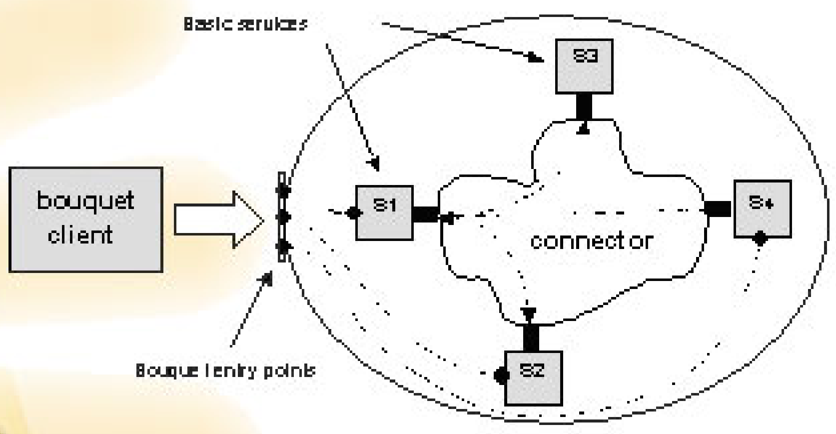
### Tools in:

http://dmag.upf.edu/livingsw/

### Device profile-CC/PP (*)

Rosa Gil, Roberto Garcia, Jaime Delgado
*5th International Workshop on Mobile Agents for Telecommunication Applications (MATA'03), October 2003*
*http://hayek.upf.es/dmag/papers/rgrgjdmata03.pdf*

### Platform CoCTelS (**)




*Choukair, Z. "COCTELS, A Model to Compose Private Adaptive Bouquets of Services in Telecom Networks"., (JOIN) Special Issue on Advanced Information Networking: Architectures and Algorithms, sept 2003*

# Ad-hoc Applications

## Step 3: Statistical analysis

Semantic relations build a graph, if there are many, it becomes a large graph or large network. Statistical analysis can be applied. Many properties arise such 'small world', 'degree' or 'clustering'. This information shows changes in topology, community phenomenon, clustering and even security analysis can be performed. For example, when a component is highly connected, it becomes a vulnerable component, so it is necessary to control how many there are and where they are situated in the graph.

*Green http://www.isi.edu*
*Red dublincore.org*
*Blue http://purl.org*
*Yellow http://www.daml.org/2001/03/daml+oil*
*Orange http://www.w3.org*
*Brown weather-ont.daml*
*Pink http://www.daml.org/2001/06/map/map-ont*
*Cyan http://www.daml.ri.cmu.edu/ont*

## Step 4: Conclusions

1) Defining an ontology it is not an easy task, sometimes nodes are isolated, it means anybody can reach them, as it is showed in the graph.

2) New ontologies can be discovered, as it is showed in the graph, there are more relationships if they are connected to 'upper ontologies':

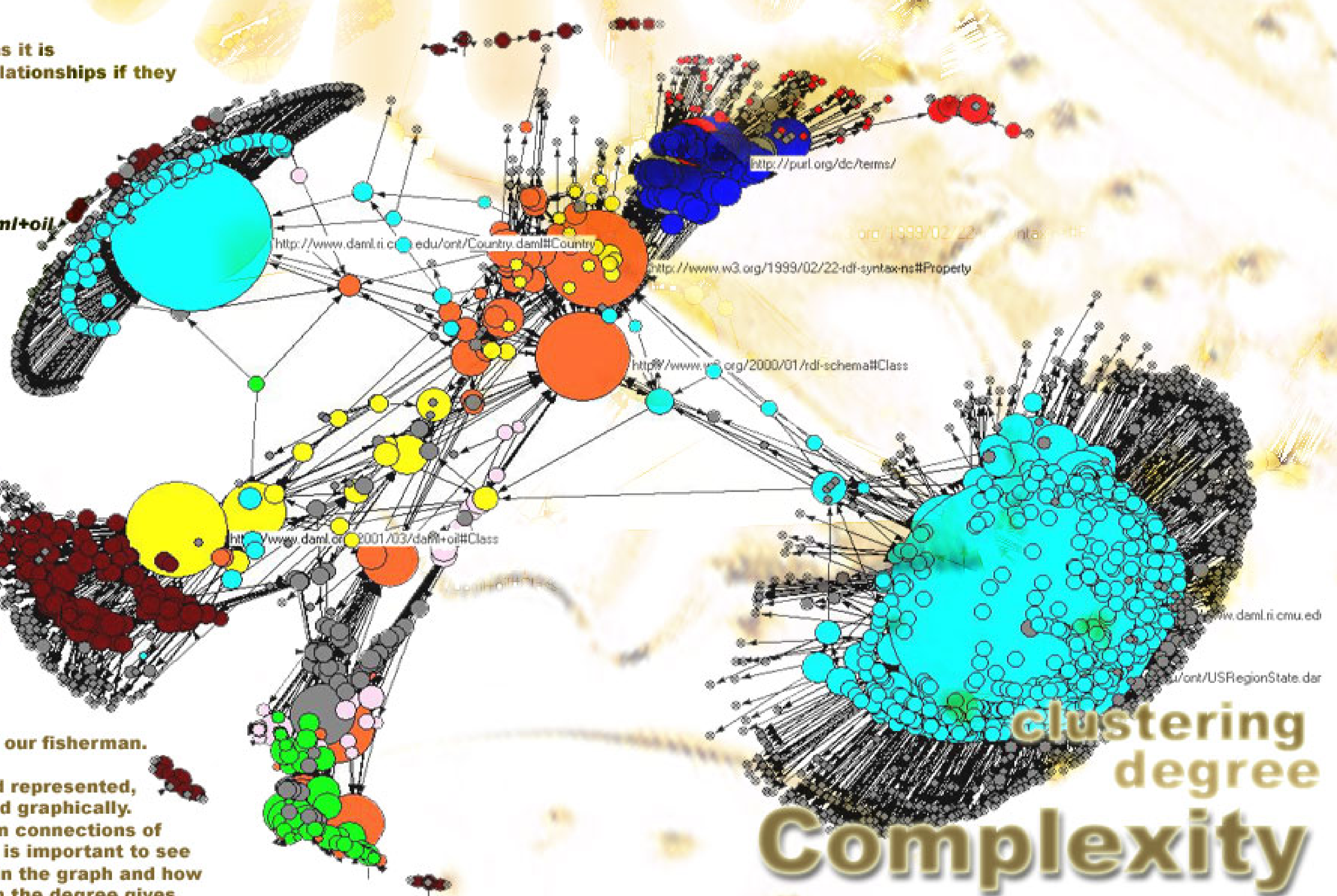
Expected
*Red dublincore.org*
*Blue http://purl.org*
*Orange http://www.w3.org*
*Yellow http://www.daml.org/2001/03/daml+oil*

Not expected
*Cyan http://www.daml.ri.cmu.edu/ont*
(countries)

The discovery depends on resources as time or depth searching.

3) Nodes are related semantically and they build clusters related each other with paths and belonging to 'small worlds'. Ex: we have discovered a kind of weather report used in the aviation domain, METAR reports. METAR is the international standard code format for hourly surface weather observations which is analogous to the SA coding currently used in the US. The acronym roughly translates from French as Aviation Routine Weather Report. Therefore, we have the MetarReport ontology class that describes METAR formatted reports. It could be useful to our fisherman.

4) Nodes degree can be normalized and represented, so the hierarchical semantics is showed graphically. This information shows the relevance in connections of every component. For security reasons is important to see where these components are situated in the graph and how important they are. It is the information the degree gives. The size of the node is a measure of that phenomenon




clustering
degree
# Complexity

# 13. Bibliography


[1] [Web] WordNet web page. http://www.cogsci.princeton.edu/~wn/

[2] [Web] Process ontology definition: http://www.jfsowa.com/ontology/process.htm

[3] [Paper] Rahwan, I. , Ramchurn, S. D. , Jennings ,N. R., McBurney, P. Parsons, S. and Sonenberg L. "Argumentation-based negotiation" The Knowledge Engineering Review. (to appear) (2004)

[4] [Book] von Stengel, B. Computing equilibria for two-person games. Chapter 45, Handbook of Game Theory, Vol. 3. eds. R.J. Aumann and S. Hart, North-Holland, Amsterdam, 1723-1759. 2002

[5] [Book] S. Coleman, James. The Foundations of Social Theory.Cambridge, Mass.: Harvard University Press, 1990

[6] [Proceedings] Larson, K., Sandholm, T., An Alternating Offers Bargaining Model for Computationally Limited Agents. In the proceedings of the First International Joint Conference on Autonomous Agents and Multiagents Systems (AAMAS 2002),Bologna, Italy, July 2002.

[7] [Paper] Dash, R. K., Parkes, D. and Jennings, N.R. Computational Mechanism Design: A call to Arms. IEEE Intelligent Systems 18 (6):pp. 40-47. 2003

[8] [Paper] Binmore and Vulkan, Applying game theory to automated negotiation, Netnomics Vol.1, 1999, pages 1-9

[9] [Book] Kraus, S., Strategic Negotiation in MultiAgent Environments. The MIT Press, 280 pp. September 1, 2001. ISBN:0-262-11264-7

[10] [Proceedings] Yokoo, M. An Approach to Over-constrained Distributed Constraint Satisfaction Problems: Distributed Hierarchical Constraint Satisfaction. Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS-2000), 2000

[11] [Paper] Luo, X., Jennings, N. R., Shadbolt, N., Leung, H. and Lee, J. H. A fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments. Artificial Intelligence Journal 148(1-2):pp. 53-102. 2003

[12] [Paper] Jennings, N. R., Faratin, P., Lomuscio, A. R , Parsons, S. ,Sierra C. and Wooldridge M. Automated negotiation: prospects, methods and challenges Int. J. of Group Decision and Negotiation 10 (2) 199-215. 2001

[13] [Paper] Sycara, K. Coordination of Multiple Intelligent Software agents, International Journal of Cooperative Information Systems 5(2 - 3): 181 - 211. 1996

[14] [Paper] Esteva, M., Padget J.A. Auctions without auctioneers: distributed auction protocols. Lecture Notes in Artificial Intelligence, Vol. 1778, Springer Verlag, pp. 20-38. 2001

[15] [Web] HTTP. http://www.w3.org/Protocols/

[16] [Web] CORBA Web site. http://www.corba.org/

[17] [Paper] Finin ,T., Labrou ,Y., and Mayfield, J., KQML as an Agent Communication Language. In Software Agents, J. M. Bradshaw (ed.), AAAI Press/The MIT Press, 291-316. 1977

[18] [Web] SL (Semantic Language). http://www.fipa.org/specs/fipa00008/XC00008G.html

[19] [Paper] Morreau, M., Kraus, S., Syntactical Treatments of Propositional Attitudes, Artificial Intelligence Journal, 106: 161-177, 1998.

[20] [Web] OMG MASIF. http://www.det.ua.pt/Projects/difference/work/D7/d7chap4.html

[21] [Web] JADE's web site. http://sharon.cselt.it/projects/jade/

[22] [Web] http://leap.crm-paris.com/public/docs/FIPA-London-Demo.pdf

[23] [Web] Grasshoper web site. http://www.grasshopper.de

[24] [Web] http://www.cordis.lu/infowin/acts/analysys/products/thematic/agents/ch2/ch2.htm#3

[25] [Web] JAL (Java Abstract Layer) belonging JXTA http://ezel.jxta.org/jal.html

[26] [Web] Sun company: http://www.sun.com

[27] [Web] PROLOG. http://www.swi-prolog.org/

[28] [Web] JESS web page: http://herzberg.ca.sandia.gov/jess/

[29] [Paper] Reif, J. and Wang. H. Social potential fields: A distributed behavioral control for autonomous robots. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, International Workshop on Algorithmic Foundations of Robotics (WAFR), pages 431--459. A. K. Peters, Wellesley, MA, 1995.

[30] [Paper] Arato, M. Butrimenko, A. and Gelenbe, E., Some extensions to multiclass queueing network analysis. In Performance of Computer Systems., Eds. North Holland, Amsterdam, 1979.

[31] [Paper] Brueckner, S.A. Parunak. H. Van Dyke. Swarming Agents for Distributed Pattern Detection and Classification. ACM 1-58113-000-0/00/0000

[32] [Web] ACO public software. http://iridia.ulb.ac.be/~mdorigo/ACO/aco-code/public-software.html

[33] [Paper] Doerner, K., Hartl, R.F. Kiechle, G., Lucka, M. and Reimann, M.: Parallel Ant Systems for the Capacitated Vehicle Routing Problem, to appear in: Raidl et al. (Eds.): Applications of Evolutionary Computing, Springer LNCS. (2004)

[34] [Web] Repast web site. http://repast.sourceforge.net

[35] [Web] Ascape model. http://www.brook.edu/es/dynamics/models/ascape

[36] [Web] SimBuilder for Repast. http:docs/api/uchicago/src/sim/engine/Schedule.html

[37] [Paper] Tobias, R., Hofmann, C., Evaluation of Free Java-Libraries for Social-Scientific Agent Based Simulation (html,31pp), Journal of Artificial Societies and Social Simulation, Vol. 7, No. 1. (2004)

[38] [ISO document] “(21000-1) ISO/IEC TR 21000-1”. ISO/IEC JTC 1/SC 29/WG 11N6149. First edition in 2001

[39] [ISO document] “(21000-2) ISO/IEC 21000-2:2003”. ISO/IEC JTC 1/SC 29/WG 11N6409. March 2004.

[40] [ISO document] “ISO/IEC 21000-2 DID 2nd Edition CD”. ISO/IEC JTC 1/SC 29/WG 11N6409. March 2004.

[41] [ISO document] “ISO/IEC 21000-3 FDIS Part 3: Digital Item Identification”. ISO/IEC JTC 1/SC 29/WG 11N4939. July 2002

[42] [ISO document] “ISO/IEC 21000-4 WD v.1 IPMP”. ISO/IEC JTC 1/SC 29/WG 11N 6644. July 2004.

[43] [ISO document] “ISO/IEC FDIS 21000-6 - Rights Data Dictionary”. ISO/IEC JTC 1/SC 29/WG 11N6172. April 2004.

[44] [ISO document] “ISO/IEC 21000-6:2004/ Rights Data Dictionary TECHNICAL CORRIGENDUM 1”. ISO/IEC JTC 1/ SC 29 N 6172. October 2004.

[45] [ISO document] “ISO/IEC 21000-7 Digital Item Adaptation AMENDMENT 1: DIA Conversions and Permissions”. ISO/IEC JTC 1/SC 29/WG 11N6173. October 2004.

[46] [ISO document] “Text of ISO/IEC 21000-8 CD MPEG-21 Reference Software”. ISO/IEC JTC 1/SC 29/ N 5953. June 2004.

[47] [ISO document] “Text of ISO/IEC 21000-9 CD”. ISO/IEC JTC 1/SC 29/ 6234. December 2004.

[48] [ISO document] “ISO/IEC 21000-10 Study of CD Digital Item Processing”. ISO/IEC JTC 1/SC 29/WG 11N5796. April 2004.

[49] [ISO document] “DoC on PDTR of ISO/IEC 21000-11”. ISO/IEC JTC 1/SC 29/WG 11N6391. March 2004.

[50] [ISO document] “Study of ISO/IEC 21000-12/PDTR Test Bed for MPEG-21 Resource Delivery”. ISO/IEC JTC 1/SC 29/WG 11N5783. April 2004.

[51] [ISO document] “Request for Subdivision of ISO/IEC 21000: Part 13 Scalable Video Coding”. ISO/IEC JTC 1/SC 29/WG 11N6192. December 2003.

[52] [ISO document] “WD3.0 of ISO/IEC 21000-14 MPEG-21 Conformance”. ISO/IEC JTC 1/SC 29/WG 11N6472. March 2004.

[53] [ISO document] “ISO/IEC 21000-15 WD v.1 Event Reporting”. ISO/IEC JTC 1/SC 29/WG 11N6419. March 2004.

[54] [ISO document] “WD1.0 of Text of ISO/IEC 21000-16”. ISO/IEC JTC 1/SC 29/WG 11N6333. March 2004.

[55] [Web] XrML web page. http://www.xrml.org/

[56] [Paper] Linyuan L. The diameter of random massive graphs. Symposium on Discrete Algorithms. Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms table of contents. Washington, D.C., United States .Pages: 912 – 921. ( 2001 ) ISBN:0-89871-490-7

[57] [Papers] Erdos P., and A. Renyi (1959): "On Random Graphs," . Publicationes Mathematicae 6, pp. 290-297. Erdos P., and A. Renyi (1960): "On the Evolution of Random Graphs," Hungarian Academy of Sciences 5, pp. 17-61.

Erdos P., and A. Renyi (1961): "On the Strength of Connectedness of a Random graph," Acta Mathematica Scientia 12, pp. 261-267

[58] [Book] Bollobas B.: Graph Theory: An Introductory Course. New York: Springer Verlag. 1979

[59] [Book] M. L. Cohen, Theoretical calculations of phase transitions in the megabar regime, Scripta Metal. 22, 127 .1988.

[60] [Book] Bollobas B. :Random Graphs. Academic Press. New York. 1985

[61] [Book] Barabási, A.L. Linked. April, 2003 Plume Books. ISBN: 0452284392 1999

[62] [Book]Berners-Lee. T, Fischetti. M. Weaving the Web .Harper San Francisco; Hardback: ISBN:0062515861

[63] [Book] Horkheimer, M. and Adorno, T. W. Dialectic of Enlightenment: Philosophical Fragments (1947), t, Stanford, ed. G. S. Noerr, trans. E. Jephcot Calif.: Stanford University Press, 2002.

[64] [Paper] Aguirre, C., Martínez, J., Huerta, R., Corbacho, F.: A New Topology for Multi-Agent Collaboration. 1st First Technical Workshop of the Computer Engineering Department, Universidad Autónoma de Madrid .2000

[65] [Proceedings] Zhang, H., Goel, A., Govindan, R.: Using the Small-World Model to Improve Freenet Performance. Proc. IEEE Infocom, 14. 2002

[66] [Proceedings] Adamic, L.A.: The Small World Web. Proceedings of ECDL'99, LNCS 1696, Springer-Verlag Pages 443-452. 1999

[67] [Paper] Watts, D.J., Strogatz, S. H.: Collective dynamics of small-world networks. Nature 393 .Pages 440-442. 1998

[68] [Paper] Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics,74 Pages 47-97. 2002

[69] [Paper] Barabási, A.L and Bonabeau, E.. Scale-Free Networks. Scientific American, May 2003

[70] [Paper] Milgram, S.: The small world problem. Psychology Today 2 (1967) 60-67

[71] [Paper] Travers, J., Milgram, S. ,Sociometry 32, 425. 1969

[72] [Paper] Solé, R.V., Ferrer, R., Montoya, J.M., Valverde, S.: Selection, tinkering and emergence in Complex Systems. Complexity, 8(1) Pages 20-33. 2002

[73] [Paper] Barabási, A.L., Albert, R.: Science, 286, 509. 1999

[74] [Proceedings] Barabási, A.L., Dezso, Z., Ravasz, E., Yook, S.H., Oltvai, Z.: Scale-free and hierarchical structures in complex networks. Proc. on Complex Networks. 2002

[75] [Book] Kauffman S., At Home in the Universe: The Search for Laws of Self-Organization and Complexity. Oxford University Press; Reprint edition (October 1996) . 336 pages. ISBN: 0195111303

[76] [Paper] Bak, P., Tang, C., and Wiesenfeld, K.: Self-organized criticality, Phys. Rev., A3 Pages 364–374. 1988

[77] [Paper] Carlson, J. M. and Langer, J. S.: Mechanical model of an earthquake fault, Phys. Rev., A4 Pages 6470–6484. 1989

[78] [Paper] Drossel, B., Schwabl, F.: Self-organized critical forest-fire model, Phys. Rev. Let., Pages 61629–1632. 1999

[79] [Paper] Ishii, H., Page, S.E., Wang, N.: A day at the beach: human agents self-organizing on the sand pile. Advances in Complex Systems, Vol. 2, No. 1 Pages 37-63. 1999

[80] [Web] SOC bibliography: http://www.pitt.edu/~tedt/theory2.PDF

[81] [Paper] Odell, J., Parunak, H. Van, Fleischer, Ml, Dyke, B. S, Modeling Agents and their Environment: The Physical Environment. Jounal of Object Technology. http://www.jot.fm/issues/issue_2003_03/column5

[82] [Web] PetriNets tutorial: http://www.daimi.au.dk/PetriNets/

[83] [Proceedings] Ferrer-i-Cancho, R. and Sole, R. V. Least effort and the origins of scaling in human language. Proceedings of the National Academy of Sciences of the United States of America, 100:788--791. (2003)

[84] [Web] XML specification: http://www.w3c.org/TR/REC-xml

[85] [Web] Ebxml web page: http://www.ebxml.org

[86] [Web] Mindswap web page. http://owl.mindswap.org/

[87] [Web] Apache web page.http://xml.apache.org

[88] [Web] DAML services. http://www.daml.org/services

[89] [Web] UDDI specification http://www.uddi.org/specification.html

[90] [Web] WSDL specification http://www.w3.org/TR/wsdl.html

[91] [Web] Opencyc ontology http://www.opencyc.org

[92] [Proceedings] A.Oltramari, A. Gangemi, N.Guarino, and C. Masolo. Sweetening ontologies with DOLCE. In A. Gómez-Pérez and V.R. Benjamins, editors. Knowledge Engineering and Knowledge Management, Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza. Spain, October 1-4, 2002, Proceedings, volume 2473 of Lecture Notes in Computer Science, Springer 2002.

[93] [Web] Wonder Web page. http://wonderweb.semanticweb.org/owl/

[94] [Web] Survey of upper ontologies. http://www.cs.uoregon.edu/~tdbreaux/poster/index.html

[95] [Book] De Bustos, E. *'La Metáfora'* . Fondo de Cultura Económica de España, S.L. UNED. ISBN: 84-375-0498-8. (2000)

[96] [Book] Chomsky, N. Syntactic Structures, (1957)

[97] [Web] Feldman, J. and Narayanan, S. Embodiment in a Neural Theory of Language. Brain and Language. http://www.icsi.berkeley.edu/NTL/papers/B+L.doc (2003).

[98] [Web] M. Tettamanti, G. Buccino , M. C.Saccuman , V.Gallese , M. Danna , D. Perani , S. F. Cappa , F. Fazio, G. Rizzolatti Sentences describing actions activate visual motor execution and observation systems. http://www2.hawaii.edu/~bergen/ling750M/papers/tettamantitextfigs.pdf ,(2004)

[99] [Paper] Buccino, G., Binkofski, F., Fink, G.R., Fadiga, L., Fogassi, L., Gallese, V., Seitz, R.J., Zilles, K., Rizzolatti, G., & Freund, H.-J. (2001). Action observation activates premotor and parietal areas in a somatotopic manner: an fMRI study. European Journal of Neuroscience, 13, 400-404.

[100] [Book] Wilson, Frank R. "La Mano: De cómo su uso configura el cerebro, el lenguaje y la cultura humana". Tusquets Editores. ISBN: 84-8310-828-3. (2002)

[101] [Book] Regier, T. The Human Semantic Potential: spatial language and constrained connectionism, MIT Press. ISBN: 0-262-18173-8. (1996)

[102] [Book] Hobson, J. A. (1988). The dreaming brain. London: Penguin Books.

[103] [Paper] Marmor, G., & Zaback, L. "Mental rotation by the blind: Does mental rotation depend on visual imagery?" Journal of Experimental Psychology: Human Perception and Performance, 2, 515-521 (1976).

[Paper] Kerr, N. H. The role of vision in "visual imagery" experiments: Evidence from the congenitally blind. Journal of Experimental Psychology: General, 112, 265-277. (1983).

[104] [Proceedings] Bailey, D., Feldman, J., Narayanan, S. & Lakoff, G. Modelling Embodied Lexical Development. In Proceedings of the 19th Annual Conference of the Cognitive Science Society, (Stanford University, 7-10 August 1997), Lawrence Earlbaum. (1997)

[105] [Book] Penrose. R. Emperor's new mind. Publisher: Penguin USA (Paper); Reprint edition (February 27, 1994) .ISBN: 0099771705

[106] [Paper] Levinson, S.C. Frames of reference and Molyneux's question: Crosslinguistic evidence. In P. Bloom, M.A. Peterson, L. Nadel & M.F. Garrett (eds.), Language and Space (pp. 109-169). Cambridge, MA: MIT Press. (1996).

[107] [Web] A Generic Negotiation Model for MAS Using XML. Paper ABA2003.Belonging to ANTS project. Équipe SMAC. Systèmes Multi-Agents et Coopération. Laboratoire de Recherche en Informatique de l'Université des Sciences et Technologies de Lille. http://www.lifl.fr/

[108] [Web] Agents web page http://www.umbc.com

[109] [Final Career Project] Amat Jiménez, R. Automatic agent negotiation through SOAP. Enginyeria Informática. (FIB-UPC). October 2002.

[110] [Web] MPEG-21 web page.http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm

[111] [Web] IPROnto, DMAG contribution to MPEG-21 Rights Data Dictionary and Rights Expression Language CfP. http://dmag.upf.es/ontologies/ipronto/mpeg21/m7675(DMAG).zip

[112] [Web] IMPRIMATUR Project, http://www.imprimatur.net

[113] [Paper] Gallego, I; Delgado, J and Acebron, J.J.: Distributed models for brokerage on electronic commerce. In Trends in distributed systems for electronic commerce, Springer, Germany.1998. ISBN: 3-540-64564-0. 129-140.

[114] [Paper] Gallego, I., Delgado, J. and García, R.: Use of Mobile Agents for IPR Management and Negotiation. In Mobile Agents for Telecom Applications, MATA'00. LNCS 1931, ISBN: 3 540 41069 4. Springer, Germany, 2000, pp. 205-213.

[115] [Final Career Project] Gallego, A. "Stationary Agents for IPR-Contract Development". (FIB-UPC). June 2002

[116] [Final Career Project] Sirvent, D. "Agentes para el control del copyright en Internet". (FIB-UPC). January 2002

[117] [Proceedings] Tous, R.: Interoperability Adaptors for Distributed Information Search on the Web. From Information to knowledge, Electronic Publishing 2003 (ElPub2003) Proceedings, Pag. 334 - 340, June 2003. ISBN 972-98921-2-1.

[118] [Web] Lassila, O., Swick, R.R. (editors): Resource Description Framework (RDF), Model and Syntax Specification. W3C Recommendation 22 February 1999 http://www.w3.org/TR/REC-rdf-syntax

[119] [Web] FIPA web page, http://www.fipa.org

[120] [Web] Eastlake, D.; Reagle, J.; Solo, D. (editors): XML-Signature Syntax and Processing . W3C Recommendation 12 February 2002 . http://www.w3.org/TR/xmldsig-core

[121] [Web] ICS-FORTH RDFSuite, http://www.ics.forth.gr/proj/isst/RDF

[122] [Web] DAML-S Coalition: DAML-S: Semantic markup for Web services. In Proc. Semantic Web Working Symposium (SWWS), 411–430, 2001.

[123] [Web] RDFLogic, http://www.w3.org/2000/10/swap/log

[124] [Web] RuleML, www.dfki.de/ruleml

[125] [Web] Darpa Agent Markup Language + Ontology Interchange Language (DAML+OIL), http://www.daml.org

[126] [Web] Zou, Y.: DAML XSB Rule version 0.3. http://www.cs.umbc.edu/~yzou1/daml/damlxsbrule.pdf

[127] [Web] DAMLJessKB web page. http: plan.mcs.drexel.edu/projects/legorobots/design/software/DAMLJessKB. [Paper] OWLJessKB. Engineering Design Knowledge Management for Conceptual Design. American Association for Artificial Intelligence, 2004

[128] [Web] MIDP Mobile Information Device Profile, http://java.sun.com/products/midp

[129] [Web] Pjava, Personal Java. http://java.sun.com/products/personaljava/pj-emulation.html

[130] [Web] Java 2 Platform Standard Edition, http://java.sun.com/j2se

[131] [Web] JADE inter-platform mobility developed by UAB http://mia.ece.uic.edu/~papers/MediaBot/jade/doc/tutorials/HTTP_UAB.html.

[132] [Web] World Wide Web Consortium, http://www.w3c.org

[133] [Web] CC/PP information, http://www.w3.org/Mobile/CCPP

[134] [Web] Device independence principles, http://www.w3.org/TR/2001/WD-di-princ-20010918

[135] [Web] Discussing delivery context, http://www.w3.org/TR/2002/WD-di-dco-20021213

[136] [Web] Discussing authoring, http://www.w3.org/TR/2002/WD-acdi-20021018

[137] [Web] Composite Capabilities/Preferences Profile Working Group, http://www.w3.org/Mobile/CCPP

[138] [Web] DELI: A Delivery context Library for CC/PP and UAProf. http://www-uk.hpl.hp.com/people/marbut/DeliUserGuideWEB.htm

[139] [Web] WAP forum. http://www.wapforum.org

[140] [Web] RDF validation service. http://delicon.sourceforge.net/

[141] [Web] Ohto ,H. and Hjelm, J.: CC/PP exchange protocol based on HTTP Extension Framework.W3C Note 24 June 1999, http://www.w3.org/TR/NOTE-CCPPexchange

[142] [Web] WAP Forum Releases, Technical Specification, http://www.wapforum.org/what/technical.htm

[143] [Poster] Delgado, J., Gallego, I., Garcia, R., Gil, R.: An ontology for Intellectual Property Rights: IPROnto. Extended poster abstract, International Semantic Web Conference (2002) http://dmag.upf.es/papers/ExtAbstractPosterISWC.pdf

[144] [Web] Wugofski, T., Dominiak, D., Stark, P. and Roy, T.: CSS Mobile Profile 1.0. W3C Candidate Recommendation 25 July 2002, http://www.w3.org/TR/css-mobile

[145] [Web] Clark, J. (ed.): XSL Transformations (XSLT) Version 1.0 W3C Recommendation 16 November 1999, http://www.w3.org/TR/xslt

[146] [Web] XML Path Language, http://www.w3.org/TR/xpath

[147] [Web] Brickley, D. and Guha, R.V. (eds): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004.http://www.w3.org/TR/rdf-schema/

[148] [Paper] Karvounarakis, G.; Alexaki, S.; Christophides, V.; Plexousakis, D.; Scholl, M.: RQL: A Declarative Query Language for RDF. The Eleventh International World Wide Web Conference (2002)

[149] [Technical Final Draft] Bowman, M., Chandler, R.D., Keskar. D V. Intel Labs. Delivering Customized Content to Mobile Device Using CC/PP and the Intel ®CC/PP SDK. http://www.intel.com/pca/developernetwork/download/bsp/ccpp_WP_finaldraft.pdf

[150] [Technical Report] Butler M. H.: Implementing Content Negotiation using CC/PP and WAP UAProf. Information Infrastructure Laboratory HP Laboratories Bristol HPL-2001-190

[151] [Web] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/

[152] [Web] FIPA Communicative Act Library Specification, http://www.fipa.org/specs/fipa00037/SC00037J.html

[153] [Web] Adapted Content Delivery for Different Contexts http://opera.inrialpes.fr/people/Tayeb.Lemlouma/Papers/saint2003.pdf

[154] [Web] FIPA Device Ontology Specification. http://www.fipa.org/specs/fipa00091/XC00091C.pdf

[155] [Web] DAMLJessKB documentation. http://edge.mcs.drexel.edu/assemblies/software/damljesskb/javadocs/edu/drexel/gicl/daml/damljesskb/DAMLJessKB.html

[156] [Web] RuleML http://www.semanticweb.org/SWWS/program/full/paper20.pdf

[157] [Web] Example fragment extracted from the Siemens S55 mobile phone CC/PP profile available at http://communication-market.siemens.de/UAProf/S55_00.xml

[158] [Paper] Foster, I.; Kesselman, C.; Tuecke, S.: The Anatomy of the Grid, Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15:3 (2001)

[159] [Final Career Project] Zoyo Vázquez, R. “Servicio de Edición Conjunta”. Ingeniería Informática. (UPF). September 2003

[160] [Project] NewMARS is part of the Agent Web project, partly supported by the Spanish government research project. TIC-2002-01336. http://dmag.upf.edu/newmars

[161] [Project] Agent Web project supported by the Spanish government research project. TIC-2002-01336. http://dmag.upf.es. Inside Projects/AgentWeb

[162] [Book] Mandelbrot B. “Los objetos fractales”. Metatemas. Tusquets Editores, S.A. ISBN:84-7223-458-4. 5th edition. May 2000.

[163] [Book] Lewin, R. “Complejidad”. Tusquets editores, S.A. ISBN 84-7223-922-5. 2nd edition. May 2002

[164] [Web] Alife project. http://www.alife.org
[165] [Book] Kauffman. S. “Investigaciones”. Tusquets Editores, S.A. ISBN: 84-8310-851-8.1st edition. January 2003
[166] [Book] Strogatz, S. Sync. Hyperion; 1st edition (March 5, 2003) ISBN: 0786868449
[167] [Thesis] Gallego, I. “Modelos para Comercio Electrónico basados en Sistemas Intermediarios”. Thesis. May 2001
[168] [Book] Mankiw N.G., Macroeconomics, Fourth Edition.. 2000 Worth Publishers, Inc. ISBN:84-95348-12-8
[169] [Book] Jay Gould, S. The book of nothing. Published by Jonathan Cape. Random House, 20 Vauxhall Bridge, London SW1V 2SA. (2000)
[170] [Web] World International Properties Organization - http://www.wipo.int/
[171] [Paper] Delgado, J., Gallego, I., Rodriguez, E. Use of the MPEG-21 Rights Expression Language for Music Distribution . 3rd IEEE International Conference on Web Delivering of Music 2003 (WEDELMUSIC 2003), Setember 2003, IEEE Computer Society.
[172] [Paper] Polo, J., Prados, J., Delgado, J. Interoperability between ODRL and MPEG-21 REL. ODRL International Workshop, April 2004. ISBN 1-74064-500-6. 2004
[173] [Web] <indecs> web page. http://www.indecs.org/
[174] [Web] Suggested Upper Merged Ontology (SUMO), http://ontology.teknowledge.com
[175] [Web] Standard Upper Ontology (SUO) IEEE Working Group, http://suo.ieee.org
[176] [Web] IPROnto, DMAG contribution to MPEG-21 Rights Data Dictionary and Rights Expression Language call for proposals. http://dmag.upf.es/mpeg21/m7675(DMAG).zip
[177] [Web] Distributed Multimedia Applications Group web home page, http://dmag.upf.edu
[178] [Paper] Delgado, J.; Gallego, I.; Perramon, X.: Broker-based secure negotiation of Intellectual Property Rights. In: Information Security, 4th International Conference, ISC2001. Lecture Notes in Computer Science, Vol. 2200. Springer-Verlag (2001) 486-496
[179] [Book] Gallego, I., Delgado, J., García, R.: Use of Mobile Agents for IPR Management and Negotiation. In: Mobile Agents for Telecom Applications, MATA 2000. Lecture Notes in Computer Science, Vol. 1931. isbn: 3-540-41069-4. Springer-Verlarg (2000) 205–213. isbn 3-540-41069-4
[180] [Paper] Delgado, J. and Gallego, I.: Negotiation of Copyright in E-Commerce of Multimedia Publishing Material. 5th International ICCC/IFIP Conference on Electronic Publishing. IOS Press 298-306. 2001
[181] [Paper] Delgado, J. and Gallego, I.: Distributed Negotiation of Digital Rights. International Conference on Media Futures. Italy 2001
[182] [Paper] Garcia, R.; Delgado, J.: Brokerage of Intellectual Property Rights in the Semantic Web. 1st Semantic Web Working Symposium (SWWS-1), Stanford, CA. 2001
[183] [Web] IMPRIMATUR Project, http://www.imprimatur.net
[184] [Web] iMAQ: An Integrated Mobile Ad hoc QoS Framework. http://cairo.cs.uiuc.edu/adhoc/
[185] [Book] Linthicum, D. S. Enterprise Integration: An Architecture for Enterprise Application and Systems Integration. John Wiley & Sons; 1st edition. February 1, 2002
[186] [Web] BPM http://www.ebizq.net/topics/bpm
[187] [Web] BlueJade project: http://sourceforge.net/projects/bluejade/
[188] [Web] HP Server: http://www.jboss.org/index.html, developed in Java and also HP-AS
[189] [Web] XMI: http://www.omg.org/technology/xml/)
[190] [Paper] Choukair, Z. and Sfar, S. Run Time Adaptation of UMTS Services to Available Resources.; Aina March 2003; China.
[191] [Web] Java Web Services Developer Pack, http://java.sun.com/webservices/
[192] [Proceedings] Batory, D., Geraci, B.J; Validating component compositions in software system generators, Proceedings of Fourth International Conference on Software Reuse, 23-26 Page(s): 72 –81. April 1996

[193] [Proceedings] Succi, G., Wong,.R , Liu,E., Smith, M., Supporting dynamic composition of components, Proceedings of the 2000 International Conference on Software Engineering, 4-11 Page(s): 787. June 2000

[194] [Paper] Choukair, Z. COCTELS, A Model to Compose Private Adaptive Bouquets of Services in Telecom Networks., (JOIN) Special Issue on Advanced Information Networking: Architectures and Algorithms, September 2003

[195] [Web] http://www.mindswap.org/~mhgrove/weather-ont.daml

[196] [Web] Daml ontologies: http://www.daml.org/ontologies/

[197] [Web] Pajek software: http://vlado.fmf.uni-lj.si/pub/networks/pajek

[198] [Web] Time ontology: http://www.isi.edu/~pan/damltime/time-entry.owl

[199] [Web] Map ontology: http://www.daml.org/2001/06/map/map-ont

[200] [Paper] Delgado, J.; Gallego, I.; García, R. and Gil, R. An Architecture for Negotiation with Mobile Agents Mobile Agents for Telecom Applications, MATA 2002. Lecture Notes in Computer Science, Vol. 2521, pp 21-31. Springer-Verlarg 2002

[201] [Paper] Gil R., García R., Delgado J. Delivery context negotiated by mobile agents using CC/PP. Mobile Agents for Telecom Applications, Lecture Notes in Computer Science, Vol. 2881, pp 99 - 110. Springer-Verlarg, 2003

[202] [Web] Living Semantic Web page. http://dmag.upf.edu/livingsw

[203] [Proceedings] Amaral, L.A.N., Scala, A., Barthélémy, M., Stanley, H.E.: Classes of small-world networks. Proc. Natl. Acad. Sci, USA 97, Pages 11149-11152. 2000

[204] [Paper] Aguirre, C., Martínez, J., Huerta, R., Corbacho, F.: A New Topology for Multi-Agent Collaboration. 1st First Technical Workshop of the Computer Engineering Department, Universidad Autónoma de Madrid. 2000

[205] [Paper] Zhang, H., Goel, A., Govindan, R.: Using the Small-World Model to Improve Freenet Performance. Proc. IEEE Infocom, 14. 2002

[206] [Proceedings] Adamic, L.A.: The Small World Web. Proceedings of ECDL'99, LNCS 1696, Springer-Verlag Pages 443-452. 1999

[207] [Paper] Watts D.J., Strogatz, S. H.: Collective dynamics of small-world networks. Nature 393. Pages 440-442. 1998

[208] [Paper] Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics, 74. Pages 47-97. 2002

[209] [Paper] Montoya, J.M., Solé, R.V.: Topological properties of food webs: from real data to community assembly models. Santa Fe Institute Working Paper, 01-11-069 .2001

[210] [Web] http://www.daml.org/ontologies

[211] [Web] http://ontobroker.semanticweb.org/rdfcrawl

[212] [Web] http://www.hpl.hp.com/semweb/jena.htm

[213] [Web] http://dmag.upf.es/livingsw/nrdfcrawler.htm

[214] [Web] http://dmag.upf.es/livingsw/resources/damlontos.uris

[215] [Web] http://dmag.upf.es/livingsw/resources/damlontos.nt.zip

[216] [Web] http://dmag.upf.es/ontologies/ipronto

[217] [Web] http://dmag.upf.es/livingsw/resources/ipronto.uris

[218] [Web] http://dmag.upf.es/livingsw/resources/iprontomodel.rdf

[219] [Web] GNUPlot http://www.gnuplot.info/

[220] [Web] http://dmag.upf.es/livingsw/results1.htm,

[221] [Web] http://dmag.upf.es/livingsw/results2.htm

[222] [Book] Barrow ,J.D. Pi in the Sky. Counting, Thinking and Being. Oxford University Press, Oxford. (1992)

[223] [Book] C.I.Gerhardt, G.W.Leibniz. Mathematical Schiften V, RH and Der Briefwechsel von Gottfried Wilhelm Leibniz mit Mathematiker, 570"

[224] [Manuscript] Gibbs created vector analysis (the first "Vector Analysis" was a private edition of 1881).

[225] [Paper] Lasenby, J., Lasenby, A.N., Doran C.J.L. A Unified mathematical language for physics and engineering in the 21st century. Phil. Trans. R.Soc.London. A. 358, 21-39. (2000)

[226] [Book] Kuhn. T. S. What are Scientific Revolutions?. University of Chicago Press, Chicago, 1962 – ISBN 0-226-45808-3

[227] [Paper] M.Steyvers, J.B. Tenenbaum. The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. Submitted to Cognitive Science.

[228] [Paper] Ferrer i Cancho, R. Solé. R.V. Optimisation in complex networks. SFI Working paper. 2001

[229] [Paper] Péli, G., Papp, G., Are Scale Free Networks Better?. Econophysics. January 2004

[230] [Paper] Dorogovtsev, S.N. Mendes, J.F.F. and Samukhin. A.N. Anomalous percolating properties of growing networks. Phys. Rev. E 64, 066110 1-11 (2001)

[231] [Web] Bianconi, G., Marsili, M., Self-organized critical network dynamics. oai:arXiv.org:cond-mat/0312537. December 2003

[232] [Paper] Packard, N. H. Adaptation toward the edge of chaos. In: Kelso, J. A. S., Mandell, A. J., and Shlesinger, M. F. (Eds.), Dynamic Patterns in Complex Systems. World Scientific, Singapore. 1988.

[233] [Paper] Newman M.E.J. The Structure and function of complex networks. SIAM Review 45, 167-256 (2003)

[234] [Paper] Adamic, L.A. Lukose, R.M..Puniyani, A.R Huberman, B.A.. Search in power-law networks. Physical Review E, Volume 64, 046135. 2001.

[235] [Proceedings] Arenas, A. Cabrales, A. Díaz-Guilera, A. Guimerà, R.and Vega-Redondo, F. Search and Congestion in Complex Networks. Proceedings of the Conference "Statistical Mechanics of Complex Networks", Sitges, Spain, June 2002

[236] [Paper] Sarshar, N., Roychowdhury, V. Scale-free and Stable structures in Complex Ad hoc networks. Phys. Rev. E 69, 026101 .2004.

[237] [Paper] Kochhal, M., Schwiebert, L.,Gupta, S.. Role-based Hierarchical Self Organization for Wireless Ad hoc Sensor Networks. ACM 1-58113-764-8/03/0009. 2003

[238] [Paper] Faragó, A., Scalable Analysis and Design of Ad hoc networks via random graph theory. ACM 1-58113-587-4/02/0009. 2002

[239] [Project] Gupta, P. Gray,.R. Kumar. An Experimental Scaling Law for Ad Hoc Networks. (Material supported by the Air Force Office of Scientific Research. C AF-DC-5-36128 and F49620-93-10266 and U.S. Army Research Office under Contract DAAD19-00-1-0466.

[240] [Paper] Marro, J., Cortes, J.M. and Hurtado, P.I. Modeling Non equilibrium Phase Transitions and Critical Behaviour in Complex Systems. Comp. Phys. Comm. 147, 115 2002

[241] [Proceedings] Amaral, L.A.N, Scala, A., Barthélémy, M., & Stanley, H.E.. Classes of small-world networks. Proceedings of the National Academy of Sciences, 97,Pages 11149-11152. 2000

[242] [Paper] Kleinberg, J., Navigation in a Small World. Nature 406, 845. 2000

[243] [Paper] Friedman D., Evolutionary Games in Economics, Econometrica, 59,3,1991

[244] [Paper] Chwe, Suk-Young M. Communication and Coordination in Social Networks, Review of Economic Studies, 67, pp, 1-16, 2000

[245] [Book] Hauser, M. D. The Evolution of Communication. ISBN 0-262-58155-8. 776 pp., 214 illus. The MIT Press May 1997

[246] [Book] Fodor, J. The Mind Doesn't Work That Way: The Scope and Limits of Computational Psychology. Cambridge, MA. ISBN 0-262-56146-8 MIT Press. September 2001

[247] [Web] FrameNet I,II web site. http://www.icsi.berkeley.edu/framenet/

[248] [Web] Work from Aljoscha Burchardt, Saarland University. Translating FrameNet XML to OWL. http://www.coli.uni-sb.de/~albu/wd.php3

[249] [Web] Definitive stylesheets.http://www.tecn.upf.es/~rgil

[250] [Book] Nuñez R, Lakoff. G, Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being . New York: Basic Books. Hardcover, November, 2000, 493 pages. ISBN: 0-465-03770-4

[251] [Web] Dehaene, S. What Are Numbers, Really? A Cerebral Basis For Number Sense. The Third Culture. http://www.edge.org/3rd_culture/dehaene/index.html

[252] [Book] Turner, M., Fauconnier, G., The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic Books; ISBN: 0465087868, March 2003

[253] [Book] Lakoff G. and M. Johnson. Metaphors We Live By. Chicago: University of Chicago Press. ISBN: 0226468011. 1980.

[254] [Book] Lakoff. G. Women, Fire and Dangerous Things. University of Chicago Press; Reprint edition (January 1, 1990). ISBN: 0226468046 1987

[255] [Book] Johnson. M. The Body In The Mind. ISBN: 0226403181.1987

[256] [Paper] Moore, M.S. Legal Reality: A Naturalist Approach to Legal Ontology. Law and Philosophy, 21(6):619-705, International Journal for Jurisprudence and Legal Philosophy 2002.

[257] [Web] JADEX - http://vsis-www.informatik.uni-hamburg.de/projects/jadex/features.php

[258] [Paper] Castelfranchi, C. , Tan, Y-H. The Role of Trust and Deception in Virtual Societies. International Journal of Electronic Commerce, Volume 6, Number 3, Spring 2002, pp. 55.

[259] [Proceedings] Shapiro, S., Pagnucco, Lespérance, M. Y. and Levesque H. J. Iterated Belief Change in the Situation Calculus. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000), pages 527-538. Morgan Kaufmann Publishers, San Francisco, CA.

[260] [Web] http://www.fbmn.fh-darmstadt.de/home/wolff/Publikationen/A_First_Course_in_Formal_Concept_Analysis.pdf

[261] [Book] Oberle, D., Volz, R., Motik, B., Staab, S., An extensible open software environment. In Handbook on Ontologies, International Handbooks on Information Systems, chapter III, pages 311-333. Steffen Staab and Rudi Studer, Eds., Springer, 2004.

[262] [Book] Principe, Euliano, and Lefebvre. Chapter 3 Multilayer Perceptrons, Chapter 5. Support Vector Machines and Chapter 10 RBF from Neural and Adaptive Systems: Fundamentals Through Simulations. Published by John Wiley and Sons. ISBN: 0471351679.

[263] [Thesis] Hilferty,J. In Defense of Grammatical Constructions. ‘Dpt. Filologia Anglesa I Alemanya’. UB .

[264] [Proceedings] British National Corpus (Collin F.Baker, J.Fillmore, and John B.Lowe. 1998. The Berkeley FrameNet project. In Proc. COLING-ACL, Montreal, Canada.).

[265] [Technical Report] Bergen, B. K., and Chang N.C. Embodied Construction Grammar in simulation-based language understanding. Technical report TR-02-004, International Computer Science Institute.

[266] [Paper] Chang, N., Narayanan, S., Petruck, M.R.L. Putting frames in Perspective

[267] [Web] DMAG MPEG-21 Tools. http://hayek.upf.es/dmag/DMAGMPEG21Tools

[268] [Web] Ontologies in J. Sowa web page. http://www.jfsowa.com/ontology/toplevel.htm

[269] [Book] J. Feber, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Addison-Wesley, 1999.

[270] [Web] KQML .http://www.cs.umbc.edu/kqml/whats-kqml.html

[271] [Web] FIPA-ACL. http://www.engr.uconn.edu/~ibrahim/publications/acl.htm#3

[272] [Book] Reisig, W. Elements of Distributed Algorithms: Modeling and Analysis With Petri Nets. Publisher: Springer-Verlag; ISBN: 3540627529. (December 1, 1998)

[273] [Book] González Calvet, R. *Treatise of plane geometry through the geometric algebra* ISBN: 84-699-3197-0 (June 2000-July 2001).