Chapter 8

Conclusions and future work

This chapter sums up the contributions of our thesis, compares our contributions with some related works, and presents some open issues and shortcomings to be addressed by future work.

8.1 Introduction

As our work was advancing, we introduced and discussed many ideas, and put some of them in practice, but there are other issues deserving our attention that we could not afford in the context of this thesis. We have been faced with so many difficulties that we have had to focus on a limited set of issues, barely tackling other subject matters. This chapter sums up the outcomes of our efforts and recapitulate both the achievements and the still open issues.

We have stated our aim at the Introduction: to provide a framework for developing and deploying open Multi-Agent Systems supporting the automatic, on-demand configuration of agent teams according to stated problem requirements. As a result of our efforts, we have developed a multi-layered framework for MAS development and deployment that integrates Knowledge Modelling and Cooperative Multi-Agent Systems together. This framework is called ORCAS, which stands for Open, Reusable and Configurable multi-Agent Systems.

The ORCAS framework encompasses three separated layers, the Knowledge Modelling Framework, the Operational Framework, and the Institutional Framework, also called the ORCAS e-Institution. The purpose of separating the framework in layers is to bring developers an extra flexibility in adapting the framework to their own requirements, preferences and needs. Now we are going to review the main features of this framework, focusing on those issues we have made more contributions.

The ORCAS Knowledge Modelling Framework provides a conceptual framework for the design of open MAS that aims at maximizing capability reuse

[Gómez and Plaza, 2004a]. The main features of this framework are, in brief, the following:

- the components of a MAS are described at the knowledge-level [Gómez et al., 2003a], abstracted from any implementation details, and enriched with semantic information specified as shared ontologies;

- there is a clear separation of tasks and capabilities from the application domain, through the specification of domain-models using its own ontologies [Gómez et al., 2001];

- a compositional, bottom up approach to system design is proposed based on two types of matching relations: task-capability matching and capability-domain matching [Gómez and Plaza, 2004a];

- automated, on-demand design of agent teams according to the requirements of the problem at hand (Knowledge Configuration), implemented as a search process supporting three strategies [Gómez and Plaza, 2004b]: interactive, depth first (Search and Subsumes) and best first (Constructive Adaptation); and

- a clear separation of two levels in the description of a MAS: on the one hand, the Abstract Architecture states the type of components used to describe a system, the features characterizing each type of component, and the relations constraining the way components can be connected; on the other hand, the Object Language refers to the representation language used to specify component features and the inference mechanism used to reason about component specifications (to compare components in order to verify a matching relation).

The Operational Framework provides a link between the KMF and cooperative MAS; specifically, it extends the Abstract Architecture to become a full-fledged Agent Capability Description Language (ACDL), and describes an alternative model of the Cooperative Problem Solving (CPS) process based on a Knowledge Modelling approach [Gómez and Plaza, 2004b]. The main features of the ORCAS ACDL addressed by the Operational Framework are the following:

- the ORCAS ACDL includes both the communication aspects required to enact an agent capability, and the operational description of a task-decomposer, specifying the control flow among subtasks;

- the communication and the operational description of capabilities are decoupled from the functional aspects, thus maximizing the reuse of agent capabilities through different interaction protocols; and

- both the communication and the operational description of a capability are specified at the macro (social) level, using concepts from the electronic institutions formalism.

The main characteristics of the ORCAS model of the CPS process are the following:

- it includes a Problem Specification stage and a Knowledge Configuration stage to guide the Team Formation process according to the requirements of the problem;
- the different stages of the CPS process can be interleaved, allowing an agent society to react to runtime events and adapt to the dynamic nature of open environments; and
- the multiagent planning stage is partially substituted by the Knowledge Configuration process, and somehow by the team configuration stage carried out during the Team Formation process.

The Institutional Framework describes an implemented infrastructure for developing and deploying Multi-Agent Systems following the electronic institutions formalism. The ORCAS framework has been implemented as an electronic institution [Esteva et al., 2002b] in which institutional agents are responsible for mediating between providers and requesters of problem-solving capabilities [Gómez et al., 2003b].

Whilst the KMF and CPS processes provide the tools to design the competence and the social requirements of agents, the ORCAS e-Institution provides the protocols for registering services, configuring tasks and customizing agent teams to solve those tasks. It is a tool to deploy flexible, extensible and configurable Multi-Agent Systems. An application of this infrastructure has been successfully applied to build a configurable meta-search application [Gómez and Abasolo, 2002] in a medical domain [Gómez et al., 2002]. The ORCAS e-Institution has set a precedent as the first multiagent infrastructure based on the electronic institutions approach. Moreover, we are somehow contributing to the field of electronic institutions by providing a framework for configuring electronic institutions dynamically, out of scenes and performative structures describing the communication and operational aspects of capabilities.

We have summarized the main features of the ORCAS framework, and now we are going to discuss some of this features, comparing them with related work.

8.2 Discussion

With the introduction of the knowledge level [Newell, 1982] in the development of Knowledge Based Systems, the knowledge acquisition phase turned from a knowledge transfer approach to a model construction approach [Clancey, 1989, Studer et al., 1998]. Knowledge Modelling Frameworks propose methodologies, architectures and languages for analyzing, describing and developing knowledge systems [Steels, 1990, Chandrasekaran, 1986, McDermott, 1988, Schreiber et al., 1994a, Fensel et al., 1999]. The goal of a KMF is to provide a conceptual model of a system that describes the required knowledge and inferences at an implementation independent way. This model is intended to support the engineer in the knowledge acquisition phase [Van de Velde, 1993] and to facilitate reuse [Fensel, 1997a]. The reuse issue has received a lot of attention last years from the knowledge modelling community [Benjamins et al., 1996a, Motta, 1999, Fensel and Motta, 2001]; focusing on ontology-based reuse [John H. Gennari and Musen, 1998, Studer et al., 1996] and automated reuse mechanisms [Gaspari et al., 1999, Motta et al., 1999, Fensel and Benjamins, 1998b].

Surprisingly, Knowledge Modelling Frameworks have been rarely applied in the field of MAS to deal with the reuse and interoperation problems; two exceptions are found in [Iglesias et al., 1997, Glaser, 1996], which have adapted the CommonKADs methodology [Schreiber et al., 1994a] to provide a MAS development methodology.

However, instead of providing a methodology, we have provided a conceptual framework that aims to maximize the reuse of agent capabilities across different application domains. Furthermore, we have developed a MAS infrastructure supporting the on-demand configuration of agent teams according to the requirements of the problem at hand. An outstanding difference of our framework with other frameworks is the inclusion of domain-models as an independent entity, and the definition of a matching relation between domain-models and capabilities, so as to facilitate reuse of agent capabilities across different domains.

We adhere to the view of Internet as an open environment where providers and requesters of capabilities meet and interact to solve specific problems by using the resources at hand. This view of Internet as a distributed computational platform is in spirit the same of the Semantic Web initiative, in particular, our view of agent capabilities shows some aspects closer to issues from the Semantic Web. From the Semantic Web approach building an application is basically a process of composing, connecting and verifying the properties of Semantic Web Services (SWS) in a way that resembles our compositional approach to team design.

There are, however, two outstanding differences between the Semantic Web and ORCAS. On the one hand, ORCAS agents are autonomous entities that can decide to accept or to refuse a request, while services are reactive, passive entities which are directly invoked by the client; therefore, instead of a centralized composition of services, we view the composition of capabilities as a negotiation process among autonomous agents. On the other hand, our language for describing capabilities is domain independent, thus it is intended to maximize reuse, while Web Services frameworks ignore this issue, since they are domain dependent by nature (a Web service is associated to some concrete domain, like the weather of a specific country in a weather forecasting service).

8.2.1 On Agent Capability Description Languages

Some of the first languages for describing agents in open environments were based on logical deduction languages like Prolog; two well known examples are the Interface Communication Language (ICL) used in the Open Agent Architecture

[Martin et al., 1999, Cheyer and Martin, 2001] to describe agents as goals, and LDL++, used in the InfoSleuth infrastructure [Nodine et al., 1999].

Nonetheless, our way of describing the components of a Multi-Agent System is similar to LARKS [Sycara et al., 2002], a language used in the RETSINA infrastructure [Sycara et al., 2001] for describing agent capabilities and performing matchmaking. The major difference between these approaches and ORCAS lies up in the ORCAS KMF, which decouples the specification of tasks and capabilities from the application domain in order to maximize reuse. Moreover, while RETSINA relies on HTN (Hierarchical Task Network) planning and scheduling, the ORCAS framework substitutes plans by a more complex type of structure, task-configurations, and keeps scheduling out the framework due to its endo-deictic nature (belonging to the agent architecture and thus falling into the micro-level, whilst we are focused on the macro or social level). Moreover, while existing frameworks for MAS cooperation usually assume that plans are obtained beforehand (prior to engage in cooperative activities) or provided by the user, our proposal is to obtain the task-configuration on-demand, out of the capabilities and knowledge available at the moment of receiving a request to solve a problem.

8.2.2 On MAS Coordination and Cooperation

Despite the large research efforts done in the field of Cooperative Problem Solving (CPS), most of the work done falls into one or several stages of the CPS process as presented in [Wooldridge and Jennings, 1994], which has four stages: recognition, team formation, planning and execution. The problem solving process starts with an agent willing to solve a task and realizing the potential for cooperation. The process until the task to be solved is decided is usually skipped, assuming that it is already given [Wooldridge and Jennings, 1999]. However, task allocation among cooperating agents is typically based on a preplan that decomposes a task into subtasks [Shehory and Kraus, 1998], without specifying the algorithms to build such plan, neither the criteria to be taken into account.

In this thesis we have studied the feasibility and utility of a componential, bottom up design approach to build something similar to an initial plan, what we call a task-configuration. When addressing the problem of configuring a team according to problem requirements we agree with other researchers that users matter [Erickson, 1996b]; people may need to understand what happened and why a system alters it response, to have some control over the actions of a system, even when agents are still autonomous, and furthermore, users may need to predict what will happen.

Some frameworks have addressed the question of the user; we can mention for instance the Guided Team Selection approach described in [Tidhar et al., 1996], the top-down search approach proposed in [Clement and Durfee, 1999], and the case-based conversational broker described [Munoz-Avila et al., 1999].

The main difference of our approach is that a task-configuration contains more information than a Hierarchical Task Network, since it includes domain-models in addition to tasks and capabilities (equivalent to actions in an HTN).

We claim that our separation of the knowledge and the operational aspects involved in the CPS process helps understand the aspects underpinning agent cooperation, since it accounts for the static vs dynamic dimension of agent societies. The idea is to exploit the fact that the specification of agent capabilities remains stable for long periods of time, whereas there are dynamic aspects of the system or its environment that change very quickly, i.e. the agent workload or the network traffic. Therefore, it is useful to make a configuration in terms of the static description of capabilities, and thereafter the static configuration can be used to select the "best" candidate agents[1] according to dynamic and context-based information.

While other frameworks and infrastructures concentrate on the task allocation stage carried on during Team Formation, the ORCAS Knowledge Configuration process is situated before Team Formation in the Cooperative Problem-Solving model. This does not mean, however, that the Knowledge Configuration process should be completed prior to initiate Team Formation; in fact, we have proposed strategies to interleave both activities with the execution: distributed configuration, lazy configuration, and dynamic reconfiguration.

8.2.3 On Semantic Web Services

From the Semantic Web approach Internet is viewed as a network of distributed and heterogeneous services that must be composed and "orchestrated" to achieve complex tasks. From that view, a Service Description Language must support just the same type of activities we want to be supported by the ORCAS ACDL: discovery, execution, composition and interoperation. Actually, there is ongoing work to put services and agents together, for instance, the Web Service Modelling Framework (WSMF) [Fensel and Bussler, 2002] and the DAML-S ontology [The DAML-S Consortium, 2001] are being developed with a similar set of requirements in mind, though the same concepts are expressed using a different vocabulary: our way of describing the functional aspects of a capability is equivalent to a service profile in DAML-S, the communication supported by an agent over a capability corresponds to the grounding of a service, and the operational description of a task-decomposer plays a role similar to the process model of a service.

There are minor technical differences concerning the communication aspects of the two approaches. There is, however, an outstanding conceptual difference: agents are autonomous entities capable of refusing a request, whereas Web Services are passive entities that are directly invoked by the requester. Consequently, agents engage in cooperation following some kind of negotiation or communication activity in which they take an active role, whereas Web Services are composed and "orchestrated " under the baton of another entity (usually through the use of workflow-based specifications) without active participation of the Web Services.

[1]The notion of agent goodness is specified as a criteria to be optimized, i.e. cost, speed, reliability and so on.

DAML-S	Agent activities	ORCAS ACDL
Profile	Discovering (matchmaking)	Inputs, outputs and competence
Grounding	Invocation and Execution	Communication
Operational model	Composition and Interoperation	Subtasks and operational description

Table 8.1: Comparison of the ORCAS ACDL against DAML-S

Table 8.1 summarizes the relation between the features characterizing a capability in ORCAS, the features proposed to describe agent-enabled Semantic Web Services in the DAML-S ontology [The DAML-S Consortium, 2001], and the kind of activities these features are required for [Bansal and Vidal, 2003, Bryson et al., 2002, Park et al., 1998, Payne et al., 2001].

8.2.4 On the design of agent teams

Design is a fundamental aspect of engineering in general and software development in particular, and there are some efforts to provide design methodologies for agents and even multiagent systems, but idea of design has not been applied to coalition or team formation. Therefore, our idea of introducing a design perspective in the team formation process is new. Specifically, we have introduced a design stage, that we called the Knowledge Configuration process. The Knowledge Configuration process aims aims at deciding the competence required for a team of agents to achieve a global whilst satisfying global requirements. This process has been implemented as a search process over the space of possible configurations (designs). Such a search-based approach opens door to a large number of techniques to be applied; CBR is just an example of that (§4.5).

8.3 Future work

The objectives of this work were ambitious; the complexity of the problems faced and the wide, interdisciplinary scope of the challenges encountered make us concentrate on some specific aspects, while others issues have been disregarded or postponed. This section will introduce still open problems and will draw up some research lines to be followed by future work in order to advance forwards towards the realization of full-fledged open MAS.

Open systems allow the involvement of agents from diverse design teams, with diverse objectives that may be unknown at the time of design. Multiagent infrastructures are expected to provide a critical enabler for development of scalable interoperable systems, however, in order to successfully communicate in such an environment, agents need to overcome two fundamental problems: first, they must be able to *find* each other (since agents might appear or disappear at any time), and once they have done that, they must be able to *interact*

[Nwana and Woolridge, 1996, Jennings et al., 1998]. Although existing infrastructures are incorporating mechanisms for advertising, finding, using, combining and updating agent services and information [Decker et al., 1997b], most of them are still relying on homogeneity assumptions to achieve a successful interaction: a common communication language and protocol; a common format for the content of communication; and a shared ontology.

As stated in the Introduction (Chapter 1), full-fledged open MAS must support cooperative work spanning multiple application domains and assembly of teams out of heterogeneous agents (and legacy applications) developed by different teams, using different communication languages and ontologies. For this reason, there is a standing interest on semantic interoperability and semantic middleware.

In the ORCAS KMF, components can be described using its own, independent ontologies [Fensel et al., 1997]. Because of this conceptual decoupling, ontology mappings may be required to match capabilities to tasks and domain-models to capabilities when there is an ontology mismatch between two specifications. Nevertheless, we have focused here on the matching relations, assuming that the necessary ontology mappings are already built, or assuming that all the components share the same ontologies. This is a reasonable assumption, since it is feasible and convenient to built the mappings beforehand, previously to make a component available for its use. But we expect ontology engineering to become a very important ingredient of agent middleware, as well as other fields in which semantic interoperability may play a role (e.g. Semantic Web Services and Cooperative Information Systems).

Taking into account our general motivations, some lines of research deserving further attention are those concerned with ontology engineering, including the following topics:

- ontology alignment and mapping;
- languages for representing mappings; and
- metrics for measuring semantic similarity and semantic distance.

The ORCAS framework could highly benefit from technological advances supporting the automatization of ontology-related activities like reasoning with mappings and mapping discovery. The idea of introducing a new kind of reusable *connectors* to bridge the gap between semantically differing specifications seems considerably interesting. Some inspiring works concerning that subject are found in [Park et al., 1998, Gómez and Benjamins, 1999]. In ORCAS connectors could be inserted between capabilities and domain-models, and between tasks and capabilities as well. A connector in ORCAS involves two dimensions: a knowledge-level specification, which allows to match components specified with different ontologies; and the implemented counterpart, which allows semantically (or syntactically) heterogeneous agents to interoperate during the Teamwork process.

The idea of ontology agents specialized in discovering mappings fits well in the context of the ORCAS infrastructure, since ORCAS relies on institutional agents providing specialized services to other agents (e.g. the Librarian and the Team-Broker).

Other aspects of our work deserving a deeper study are those concerning the adaptation of agent teams to the dynamic nature of open environments to deal with unexpected events and handle errors. We have already draw the notions of reconfiguration, delayed configuration and lazy configuration as extensions to the core model of the ORCAS Operational Framework. We think a lot of work remains to be done yet in order to improve the adaptability of agent teams to changes in the environment, like introducing learning or incorporating some kind of meta-strategy to decide the better strategy at each particular moment.

Another point that can be addressed by future work is the extension of the ORCAS ACDL to provide fully declarative description for the operational model of a task-decomposer, including not only control flow but also intermediate data processing. Such a feature will allow task-decomposers to exist in a purely declarative form, thus any agent understanding that language would be able to follow a task-decomposer, rather than having specific agents implementing each new task-decomposer.

The ORCAS e-Institution is subject to large modifications and improvements; for instance, we think two interesting areas to work upon are security matters and extended interoperation.

Concerning security, we have not considered any security issues yet in the ORCAS framework, though we are aware of their critical importance to the field in order to develop industrial and commercial applications. An interesting line to be followed through is that introducing specialized agents to take care of supervision and security tasks, like sentinels [Dellarocas and Klein, 1999] and governors [Esteva et al., 2002b].

Concerning a greater support to interoperation, the idea of federated electronic institutions raises as a very interesting concept. The point is to allow configuring agent teams out of agent running in separate agent infrastructures. Although we have carried out some experiments involving several libraries and heterogenous agent platforms (we have connected the NOOS Agent Platform, that uses Lisp, and JADE, that uses Java), they have been conducted on a rather ad-hoc manner; therefore, we think the ORCAS framework could be improved by including a principled approach to form federations embracing several ORCAS compliant infrastructures.

Still another issue deserving work in the future is the inclusion of contractual mechanisms to support e-Commerce applications like supply chains, auctions and e-markets. Concerning this research line, we are thinking about the aspects required to implement what we like to call "terms of commitment", a mechanism to agree upon by team members when accepting a team-role. We envision agent societies negotiating the terms of service to which agents commit to partake in a team. An ontology of possible terms of commitment (e.g. exclusive vs. non exclusive, pulls vs. push, quality of service measures, cost, etc.) together with some interaction protocols has to be developed to deal with such a kind of negotiation.

Finally, we want to mention the advisability of a complete methodology for the design an development of MAS according to the ORCAS framework.

Appendix A

Specification of the Knowledge Modelling Ontology

```
(in-package noos)

(define-ontology KM-Ontology
  (creator "IIIA - CSIC")
  (description "KM-Ontology describes the elements of the \orcas\ KMF"))

(define-sort KM-Ontology)

(define-sort (KM-Ontology Concept))

(define-sort (KM-Ontology Binary-Relation)
  (argument1 Concept)
  (argument2 Concept))

(define-sort (Concept Knowledge-Component)
  (name String)
  (pragmatics Pragmatics)
  (ontologies Ontology Empty-Set))

(define-sort (Knowledge-Component Ontology))

(define-sort (Knowledge-Component Task)
  (uses Task Empty-Set)
  (input-roles Var Empty-Set)
  (output-roles Var Empty-Set)
  (competence Competence)
  (assumptions Formula))
```

```
(define-sort (Knowledge-Component Domain-Model)
  (uses Domain-Model Empty-Set)
  (properties Formula Empty-Set)
  (metaknowledge Formula Empty-Set)
  (knowledge Signature-Element Empty-Set))

(define-sort (Knowledge-Component Capability)
  (communication Communication)
  (input-roles Var Empty-Set)
  (output-roles Var Empty-Set)
  (competence Competence))

(define-sort (Capability Task-Decomposer)
  (subtasks Task Empty-Set)
  (operational-description Operational-Description))

(define-sort (Capability Skill)
  (knowledge-roles Signature-Element Empty-Set)
  (assumptions Formula Empty-Set))

(define-sort (Concept Pragmatics)
  (title string)
  (creator string)
  (subject string)
  (description string)
  (publisher string)
  (other-contributor string)
  (date string)
  (resource-type string)
  (format string)
  (resource-identifier string)
  (source string)
  (language string)
  (relation string)
  (rights-mangement string)
  (last-date string)
  (be-used string)
  (evaluation string)
  (application-descriptors Pragmatics-Descriptor Empty-Set))

(define-sort (Concept Pragmatics-Descriptor)
  (name string)
  (value string))

(define-sort (Concept Competence)
  (preconditions Formula)
  (postconditions Formula))

(define-sort (Concept Signature-Element)
```

```
  (name String))

(define-sort (Concept Formula)
  (name String))

(define-sort (Concept Operational-Description)
  (intermediate-roles Signature-Element Empty-Set)
  (programs string))

(define-sort (Concept Renaming)
  (in Signature-Element)
  (out Signature-Element))

(define-sort (Concept Communication)
  (communication string))

(define-sort (Binary-Relation Adapter)
  (argument1 Knowledge-Component)
  (argument2 Knowledge-Component)
  (pragmatics Pragmatics)
  (ontologies Application-Ontology Empty-Set)
  (renamings Renaming Empty-Set))

(define-sort (Adapter Bridge)
  (uses Bridge Empty-Set)
  (mapping-axioms Formula Empty-Set)
  (assumptions Formula Empty-Set))

(define-sort (Bridge Capability-Domain-Bridge)
  (argument1 Capability)
  (argument2 Domain-Model)
  (uses Capability-Domain-Bridge Empty-Set))

(define-sort (Bridge Capability-Task-Bridge)
  (argument1 Capability)
  (argument2 Task)
  (uses Capability-Task-Bridge Empty-Set))

(define-sort (Bridge Task-Domain-Bridge)
  (argument1 Task)
  (argument2 Domain-Model)
  (uses Task-Domain-Bridge Empty-Set))

(define-sort (Adapter Refiner)
  (in Knowledge-Component)
  (out Knowledge-Component))

(define-sort (Refiner Ontology-Refiner)
  (in Application-Ontology)
  (out Application-Ontology))
```

```
(define-sort (Refiner Domain-Refiner)
  (in Domain-Model)
  (out Domain-Model)
  (properties Formula Empty-Set)
  (metaknowledge Formula Empty-Set)
  (knowledge Formula Empty-Set))

(define-Sort (Refiner Task-Refiner)
  (in Task)
  (out Task)
  (input Input-Roles)
  (output Output-Roles)
  (competence Competence)
  (assumptions Formula Empty-Set))

(define-sort (Refiner Capability-Refiner)
  (in Capability)
  (out Capability)
  (communication Communication)
  (input Input-Roles)
  (output Output-Roles)
  (competence Competence))

(define-sort (Capability-Refiner Task-Decomposer-Refiner)
  (in Task-Decomposer)
  (out Task-Decomposer)
  (subtasks Task Empty-Set))

(define-sort (Capability-Refiner Skill-Refiner)
  (in Skill)
  (out Skill)
  (knowledge Signature-Element Empty-Set)
  (assumptions Formula Empty-Set))
```

Appendix B

Formalization of the Query Weighting Metasearch Approach

The capabilities used in the WIM application are based on a query weighting framework [Gómez and Abasolo, 2003] that is applied to transform queries during the query adaptation stage: to transform the user query into a collection of domain queries, and to transform each domain query into a collection of source queries. This framework relies on a keyword based representation of queries, plus the use of search filters, which are the common elements used by existing search engines in the Web.

A *query* Q is defined as a vector of non-repeated elements, which can be keywords, search filters or another element.

$$Q = \langle k_1 \dots k_n \rangle \qquad \forall i,j : 1 \leq i,j \leq n; k_i \neq k_j \tag{B.1}$$

A *weighted-query* QW is a pair composed of a query and a weight in the interval [0, 1].

$$WQ = \langle Q, w \rangle \qquad w \in [0,1] \tag{B.2}$$

A *query-transformation* τ is a relation between two queries (Q_1, Q_2) and a weight (w), defined as follows:

$$\tau(Q_1, Q_2, w) \Leftrightarrow (\exists! k | k \in Q_1 \wedge k \notin Q_2) \wedge (\exists! k' | k' \in Q_2 \wedge k' \notin Q_1) \wedge \sigma(k, k', w) \tag{B.3}$$

where k and *k'* are elements of queries, and $\sigma(k, k', w)$ is a relation between two elements and a weight in the interval [0,1].

Intuitively, this definition means that one query is exactly like the other but a single query element. In other words, there is a unique element k in Q_1 not in Q_2, and there exists an unique element k' in Q_2 not in Q_1, such that these elements are related by a weight $(\sigma(k, k', w))$. If there exists a query transformation between two queries and a weight w, one can transform one query into the other by replacing the element k by k', assigning the new query with a weight $= w$.

The query transformation relation is used to weight a query during a query adaptation process. In WIM, domain queries are generated from the original user consultation by using domain knowledge, such as a thesaurus or a collection of knowledge categories. The idea is that if one query is the result of a query transformation, then we can weight the new query with the weight relating both the original and the new query.

Notice that the weighted relationships between query elements (i.e. between two keywords) are encoded or can be derived from application domain knowledge. For instance, WIM uses a thesaurus to obtain semantic relationships between keywords, applying a mapping from the qualitative relationships (e.g. synonym) defined in the thesaurus to numeric values which are then used as weights (e.g. two synonyms are related with a weight $w = 1$). Moreover, during the query customization stage, domain queries are transformed into source queries by using a description of information sources. A description of a information source contains a mapping from concepts specified at the domain level to concepts used by a particular information source. The weight applied to a source query resulting of transforming a domain query into a source query (query customization) is obtained from the description of that source. The weighting values depend on the relation between the domain level concept and the source level concept, and have been decided during the knowledge acquisition phase with the help of an expert in medical bibliography.

We have not discussed yet how to assign weights when two or more query elements are changed between two queries, or how to assign a weigh to an already weighted query. Both problems are in fact the same, how to combine or synthesize weights.

Different functions can be used to combine weights. In the Query Weighting framework one can consider weights as membership values with respect to the user interest when posing a query, as well as logical values expressing the degree of relevance or utility of a query with respect to the user query. Therefore, weights can be combined by using numerical aggregation operators or multivalued logical operators (e.g. t-norms). The Query Weighting framework states a general rule that constrains the type of query synthesizing functions allowed. This rule states that the *weight of a query cannot be increased after applying a transformation*; the meaning is that query transformations move queries further away from the user request. In other words, if we transform a query q with a weight w into a new query *q'* with a weight *w'*, *w'* cannot be greater than w. Such class of operators includes -but is not reduced to- the family of t-norm operators.

The composition or synthesis of weights is defined from the notion of a *chain of query transformation.* A chain of query transformations T between two queries indicates that there exist a sequence of query-transformations between the two queries. A *chain of query transformations* T is defined as a relation between two queries and a weight, as follows:

$$T(Q_1, Q_2, w) \Leftrightarrow \begin{cases} (Q_1, Q_2, w) \wedge \\ \exists Q', \tau | \tau \wedge \tau(Q', Q_2, w') \wedge T(Q_1, Q', w'') \\ \wedge w = \Theta(w', w'') \end{cases} \tag{B.4}$$

where Θ is a *t-norm* operator.

Now we are going to define the functions used to obtain the weight for a query after applying a query transformation.

A *Query-weighting* function Γ is a function to obtain a weight between two queries according to the *chain of query-transformations* between those queries. This function is defined as follows:

$$\Gamma : \mathbb{Q} \times \mathbb{Q} \rightarrow [0, 1]$$

$$\Gamma(Q_1, Q_2) = \begin{cases} w & iff \quad T(Q_1, Q_2, w) \\ 0 & otherwise \end{cases} \tag{B.5}$$

where Q_1, $Q_2 \in \mathbb{Q}$, are queries and T is a *chain of query-transformations.*

A *weighted-query-weighting* function Ω is a function to calculate a weight according to the *chain of query-transformations* between a weighted query and a non-weighted query. Given a query Q_1 and a weighted query $\langle Q_2, w \rangle$, we define a *weighted-query-weighting* as follows:

$$\Omega : \mathbb{Q} \times \mathbb{Q} \times \mathbb{W} \rightarrow [0, 1]$$

$$\Omega(Q_1, Q_2, w) = \begin{cases} \Theta(w, w') & iff \quad T(Q_1, Q_2, w') \\ 0 & otherwise \end{cases} \tag{B.6}$$

where Q_1, Q_2, are queries, w is the weight assigned to one of the queries, T is a *chain of query-transformations* and Θ is a *t-norm* operator.

The *query-weighting* and *weighted-query-weighting* functions are used to obtain the weight for the query resulting of applying one or more query transformations. The former is used when the original query is not weighted, and the last when the original query is already weighted. In fact, both functions can be reduced to a unique function if we consider the non-weighted queries as having a weight equal to 1.

When query is weighted after applying a transformation, this expresses the relative importance or representativity of that query with respect to the original one. The meaning of a weight assigned to a query is logically inherited by the documents or items retrieved for that query, thus we can say that the weights associated to the items retrieved represent the membership of those elements to the topic requested by the user.

Appendix C

Specification of the **ORCAS** e-Institution

```
(in-package noos)

(define-institution \orcas\_institution as
  dialogic-framework = \orcas\-dialogical-framework
  performative-structure = \orcas\_performative_structure
  norms = ())

(define-performative-structure \orcas\_performative_structure as
    scenes = (
        (Root Root-Scene)
        (Output Output-Scene)
        (Brokering Brokering-Scene)
        (Registering Registering-Scene)
        (Team-formation Team-Formation-Scene)
        (Problem-solving Problem-Solving-Scene)
        (Request-inform Request-Inform-Scene)
        (Request-wrapper Request-Wrapper-Scene)
        (Full-Problem-solving Full-Problem-Solving-Scene))
    transitions = ((T0 AND-AND)
              (T1 AND-AND)
                      (T2 AND-AND)
              (T3 AND-AND)
                      (T4 AND-AND)
                      (T5 AND-AND)
              (T6 AND-AND)
                      (T7 AND-AND))
    connections = (
        (Root T0((x PSA)(y Librarian)))
                (Root T1((x PA)(y Broker)))
                (Root T2 ((x Team-broker)))
```

```
                (Registering T1((z Librarian)))
        (Registering T2((x PSA)))
        (Brokering T2((x PA)))
                (Brokering T3((x Librarian)))
        (Team-formation T5((y Team-broker)))
                (Team-formation T4((x PA)(z PSA)))
        (Problem-solving T6 ((x PA) (y PSA)))
        (T0 Registering((x PSA)))
        (T0 Registering((y Librarian)))
        (T1 Brokering((x PA)))
                (T1 Brokering((y Broker)))
                (T1 Brokering((z Librarian)))
        (T3 Output((x Librarian)))
                (T3 Output((y Broker)))
                (T2 Team-formation((z PSA)))
                (T2 Team-formation((y Team-broker)))
                (T2 Team-formation((x PA)))
                (T5 Output((x Team-broker)))
                (T6 Output((x PA)))
        (T6 Output((y PSA)))
                (T4 Problem-solving((x PA)))
                (T4 Problem-solving((y PSA)))
                (Root T7 ((x Requester) (y PA)))
                (T7 Full-Problem-solving ((x Requester) (y PA))))
    initial-scene = Root
    final-scene = Output)

(define-scene Root-Scene as
    roles = (PSA PA Librarian Broker)
    scene-dialogic-framework = IBROW-Library
    states = (W0)
    initial-state = W0
    final-states = (W0)
    acces-states = ((PSA (W0)) (PA (W0)) (Librarian (W0)) (Broker (W0)) )
    exit-states = ((PSA (W0)) (PA (W0)) (Librarian (W0)) (Broker (W0)) )
    connections = (
    ))

(define-noos-scene Team-Formation-Scene
  :description "The Team Broker forms a team to solve a problem"
  :roles  (PA PSA Team-broker)
  :states  (W6 W5 W4 W3 W2 W1 W0)
  :initial-state  W0
  :final-states  (W6)
  :connections  ((W0 W1 (request (?x PA) (?y Team-broker) Task-Configuration))
                 (W1 W2 (inform (!y Team-broker) (All PSA) Start-Team-formation))
                 (W2 W3 (request (!y Team-broker) (All PSA) Team-role))
         (W3 W3 (accept (?z PSA) (!y Team-broker) Team-role))
                 (W3 W3 (refuse (?z PSA) (!y Team-broker) Team-role))
                 (W3 W2 (inform (!y Team-broker) (?z PSA) Team-role))
```

```
                (W2 W4 (inform (!y Team-broker) (All PSA) Start-Team-configuration))
        (W4 W4 (commit (!y Team-broker) (?z PSA) Team-role))
        (W4 W5 (inform (!y Team-broker) (All PSA) Finish-Team-configuration))
                (W5 W6 (inform (!y Team-broker) (!x PA) Finish-Team-formation))
                (W2 W5 (inform (!y Team-broker) (All PSA) Finish-Team-configuration))
                ))


(define-noos-scene Registering-Scene
  :description "PSAs register their capabilities to the Librarian"
  :roles  (PSA Librarian)
  :states  (W2 W1 W0)
  :initial-state  W0
  :final-states  (W2)
  :connections  ((W0 W1 (Register (?x PSA) (?y Librarian) Capability-Set))
                 (W1 W2 (Inform (!y Librarian) (!x PSA)  Capability-Set))
                 ))



(define-noos-scene Brokering-Scene
  :description "PA request Broker for a Constructive Adaptation or First-Depth
               Search configuration"
  :roles  (Librarian Broker PA)
  :states  (W6 W5 W4 W3 W2 W1 W0)
  :initial-state  W0
  :final-states  (W6)
  :connections  ((W0 W1 (request (?x PA) (?y Broker) Problem-Specification))
                 (W1 W2 (request (!y Broker) (?z Librarian) any))
                 (W2 W3 (inform (!z Librarian) (!y Broker) Library))
                 (W3 W4 (inform (!y Broker) (!x PA) Broker-Message))
                 (W4 W5 (request (!x PA) (!y Broker) GUI-Message))
                 (W4 W6 (accept (!x PA) (!y Broker) Broker-Message))
                 (W5 W6 (inform (!y Broker) (!x PA) Broker-Message)))
                 )

(define-noos-scene Problem-Solving-Scene
  :description "PA request a team of PSAs to solve a problem"
  :roles  (PSA PA)
  :states  (W2 W1 W0)
  :initial-state  W0
  :final-states  (W2)
  :connections  ((W0 W1 (Request (?x PA) (?y PSA) Start-Problem-solving))
                 (W1 W2 (Inform (!y PSA) (!x PA)  Finish-Problem-solving))))

(define-scene Output-Scene as
    roles = (Broker Librarian PA PSA)
    states = (W0)
    initial-state = W0
    final-states = (W0)
```

```
    acces-states = ((PA (W0)) (PSA (W0)) )
    exit-states = ((PA (W0)) (PSA (W0)) )
    connections = ())

(define-noos-scene Request-Inform-Scene
  :description "PSA requests another PSA in the team to solve some subtask"
  :roles  (PSA)
  :states  (W2 W1 W0)
  :initial-state  W0
  :final-states  (W2)
  :connections  ((W0 W1 (Request (?x Requester) (?y Informer) Start-Problem-solving))
                 (W1 W2 (Inform (!y Informer) (!x Requester)  Finish-Problem-solving))))


(define-noos-scene Request-Wrapper-Scene
  :description "PSA request a Wrapper to query some information source"
  :roles  (PSA)
  :states  (W2 W1 W0)
  :initial-state  W0
  :final-states  (W2)
  :connections  ((W0 W1 (Request (?x Requester) (?y Informer) any))
                 (W1 W2 (Inform (!y Informer) (!x Requester)  any))))

(define-noos-scene Full-Problem-Solving-Scene
  :description "External agent or GUI request PA to solve a problem using all
             the steps (Full mode)"
  :grid ((3 . 1)
        ((W0 0 0) (W1 1 0) (W2 2 0))
        ((W0 W1 :U) (W1 W2 :U)))
  :roles  (PA Requester)
  :states  (W2 W1 W0)
  :initial-state  W0
  :final-states  (W2)
  :connections  ((W0 W1 (Request (?x Requester) (?y PA) Full-Problem))
                 (W1 W2 (Inform (!y PA) (!x Requester)  any))))
```

Appendix D

Specification of the ISA-Ontology

```
(define-ontology ISA-Ontology
  (creator "IIIA - CSIC")
  (description "Information Search and Aggregation (ISA) Ontology")
  (uses KM-Ontology))

(define-sort Var
  (name any)
  (sort Symbol))

(define-sort FT-Signature-Element
  (name String))

(define-sort FT-Formula
  (name String))

(define-sort (FT-Signature-Element Query-Model)
  (name String "Query-Model")
  (query Query)
  (result  Scored-Item Empty-Set)
  (weight  Number 1)
  (source Source Empty-Set))

(define-sort (FT-Signature-Element Query-Models)
  (name String "Query-Models")
  (q-models Query-Model Empty-Set))

(define-sort (FT-Signature-Element Query)
  (name String "Query")
  (filters Filter Empty-Set))
```

```
(define-sort (Query Domain-Query)
  (name String "Domain-Query")
  (terms Term Empty-Set)
  (category Category))

(define-sort (Query Source-Query)
  (name String "Source-Query")
  (t-filters Filter Empty-Set)
  (source Source))

(define-sort (Source-Query PMID-Query)
  (name String "PMID-Query")
  (pmid String))

(define-sort (Query Source-Queries)
  (name String "Source-Queries")
  (s-queries Source-Query Empty-Set))

(define-sort (FT-Signature-Element Term)
  (name String "Term")
  (term-correlations Term-Correlation Empty-Set)
  (parent Term Empty-Set)
  (children Term Empty-Set))

(define-sort (FT-Signature-Element Category)
  (name String "Category")
  (terms Term-Correlation Empty-Set)
  (filters Filter-Weighting Empty-Set))

(define-sort (FT-Signature-Element Term-Correlation)
  (name String "Term-Correlation")
  (term Term)
  (weight Number 1))

(define-sort (FT-Signature-Element Filter)
  (name String "Filter")
  (attribute String)
  (value String))

(define-sort (FT-Signature-Element Filter-Weighting)
  (name String "Filter-Weighting")
  (filter Filter)
  (weight Number 1))

(define-sort (FT-Signature-Element Item)
  (name String "Item")
  (id String)
  (content any)
  (date number)
  (infoextra string))
```

```
(define-sort (Item Bibliographic-Item)
  (name String "Bibliographic-Item")
  (UID String)
  (Title String)
  (Author String)
  (Publication-date Date)
  (Languages String)
  (Publication-type String))

(define-sort (FT-Signature-Element Scored-Item)
  (name String "Scored-Item")
  (item Item)
  (score Number))

(define-sort (FT-Signature-Element Scored-Items)
  (name String "Scored-Items")
  (s-items Scored-Item Empty-Set))

(define-sort (Sources Source)
  (name String "Source")
  (weight Number 1)
  (search-attributes Attribute-Weighting Empty-Set)
  (basic-attribute Attribute-Weighting)
  (filter-attributes Attribute-Translation Empty-Set)
  (content domain-model))

(define-sort (FT-Signature-Element Sources)
  (name String "Sources")
  (sources Source Empty-Set))

(define-sort (FT-Signature-Element Attribute-Weighting)
  (name String "Attribute-Weighting")
  (attribute String)
  (weight Number))

(define-sort (FT-Signature-Element Attribute-Translation)
  (name String "Attribute-Translation")
  (domain-attribute String)
  (source-attribute String))

(define-sort (FT-Signature-Element Search-Assessment)
  (name String "Search-Assessment")
  (assessment FT-Formula)
  )

(define-sort (FT-Signature-Element Weighted-Pair)
  (name String "Weighted-Pair")
  (weight number)
  (value number))
```

```
(define-sort (FT-Signature-Element Weighted-Pairs)
  (name String "Weighted-Pairs")
  (w-pairs Weighted-Pair Empty-Set))

(define-sort (FT-Signature-Element Item-Info)
  (name String "Item-Info")
  (item Item)
  (pairs Weighted-Pair Empty-Set))

(define-sort (FT-Signature-Element Item-Infos)
  (name String "Item-Infos")
   (item-infos Item-Info Empty-Set))

(define-sort (FT-Signature-Element  Weighting-Function)
  (name "Weighting-Function"))
```

Appendix E

Specification of the ISA-Library

```
(define (Library Wim-Library
  (creator "IIA-CSIC")
  (description "WIM ISA Library with feature terms")
  (uses ORCAS-KM-Ontology ISA-Ontology))


(define (Task :id Information-Search)
  (name "Search")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      SATISFY-CONSULT
      ))))

(define (Task :id PCM-search)
  (name "PCM-Search")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
```

```
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      ASSESS-SEARCH-RESULT
      SATISFY-CONSULT
      ))))

(define (Task :id Modify-search)
  (name "Modify-Search")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      SATISFY-CONSULT
      CHANGE-SCOPE
      )))
  (configuration-options "Configurable On Runtime")
  )


(define (Task :id Critique-search)
  (name "Critique-Search")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (output-roles
   (define (var)
     (name 'assesment)
     (sort Assess-results)))
  (competence
   (define (Competence)
     (postconditions
      ASSESS-SEARCH-RESULT
      )))
  ;(configuration-options "Produces-new-competence")
  )
```

```
(define (Skill :id Search-Assessment)
  (name "Assess-results")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (output-roles
   (define (var)
     (name 'assesment)
     (sort Search-Assessment)))
  (competence
   (define (Competence)
     (postconditions
      ASSESS-SEARCH-RESULT
      )))
  )



(define (Task :id Adapt-query)
  (name "Adapt-query")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 'elab-queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
      ELABORATE-CONSULT
      CHANGE-SCOPE
      )))
  )

(define (Skill :id Query-generalization)
  (name "Query-generalization")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
```

```
     (name 'elab-queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
      GENERALIZE-QUERY
      ELABORATE-CONSULT
      )))
  (knowledge-roles
   Thesaurus)
  )

(define (Skill :id Query-specialization)
  (name "Query-specialization")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 'elab-queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
      SPECIALIZE-QUERY
      ELABORATE-CONSULT
      )))
  (knowledge-roles
   Thesaurus)
  )


\subsection{Elaborate-query task}\label{sec:elab-q}

(define (Task :id Elaborate-query)
  (name "Elaborate-query")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name '?elab-queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
```

```
      ELABORATE-CONSULT
      ))))

(define (Skill :id Query-expansion-with-thesaurus)
  (name "Query-expansion-with-thesaurus")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 'elab-queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
      ELABORATE-WITH-THESAURUS
      )))
  (knowledge-roles
   Thesaurus)
  )

(define (Skill :id Exhaustive-query-expansion-with-thesaurus)
  (name "Exhaustive-query-expansion-with-thesaurus")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 'elab-queries)
     (sort Query-models)))
  (competence
   (define (Competence)
     (postconditions
      ELABORATE-WITH-THESAURUS-EXHAUSTIVE
      )))
  (knowledge-roles
   Thesaurus)
  )

(define (Skill :id Query-expansion-with-categories)
  (name "Query-expansion-with-categories")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
```

```
    (sort Query-Model)))
  (output-roles
   (define (var)
     (name 'elab-queries)
     (sort Query-models)))
  (competence
   (define (Competence)
     (postconditions
      ELABORATE-WITH-CATEGORIES
      )))
  (knowledge-roles
   Categories)
  )


(define (Skill :id Exhaustive-query-expansion-with-categories)
  (name "Exhaustive-query-expansion-with-categories")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 'elab-queries)
     (sort Query-models)))
  (competence
   (define (Competence)
     (postconditions
      ELABORATE-WITH-CATEGORIES-EXHAUSTIVE
      )))
  (knowledge-roles
   Categories)
  )

(define (Task :id Customise-query)
  (name "Customise-query")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'query)
     (sort Query-Model))
   )
  (output-roles
   (define (var)
     (name 'queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
```

```
      CUSTOMISE-DOMAIN-QUERY
      )))
  )

(define (Skill :id Query-customisation)
  (name "Query-customisation")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'query)
     (sort Query-Model))
   )
  (output-roles
   (define (var)
     (name 'queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
      NON-EXHAUSTIVE-CUSTOMISATION
      )))
  (knowledge-roles
   Source-Descriptions)
  )


(define (Skill :id Exhaustive-query-customisation)
  (name "Exhaustive-query-customisation")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'query)
     (sort Query-Model))
   )
  (output-roles
   (define (var)
     (name 'queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
      EXHAUSTIVE-CUSTOMISATION
      )))
  (knowledge-roles
   Source-Descriptions)
  )

(define (Skill :id Basic-query-customisation)
  (name "Basic-query-customisation")
  (ontologies ISA-Ontology)
```

```
  (input-roles
   (define (var)
     (name 'query)
     (sort Query-Model))
   )
  (output-roles
   (define (var)
     (name 'queries)
     (sort Query-Models)))
  (competence
   (define (Competence)
     (postconditions
      BASIC-CUSTOMISATION
      )))
  (knowledge-roles
   Source-Descriptions)
  )

(define (Task :id Retrieve)
  (name "Retrieve")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'query)
     (sort Query-Model))
   )
  (output-roles
   (define (var)
     (name 'result)
     (sort Query-Model)))
  (competence
   (define (Competence)
     (postconditions
      SATISFY-QUERY
      )))
  )

(define (Skill :id Retrieval)
  (name "Retrieval")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'query)
     (sort Query-Model))
   )
  (output-roles
   (define (var)
     (name 'result)
     (sort Query-Model)))
  (competence
```

```
   (define (Competence)
     (postconditions
      SATISFY-QUERY
      )))
  )

(define (Task :id Retrieve-PMID)
  (name "Retrieve-PMID")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'pmid)
     (sort PMID-Query))
   )
  (output-roles
   (define (var)
     (name 'pubmedarticle)
     (sort String)))
  (competence
   (define (Competence)
     (postconditions
      SATISFY-PMID
      )))
  )

(define (Skill :id Retrieval-PMID)
  (name "Retrieval-PMID")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'pmid)
     (sort PMID-Query))
   )
  (output-roles
   (define (var)
     (name 'pubmedarticle)
     (sort String)))
  (competence
   (define (Competence)
     (postconditions
      SATISFY-PMID
      )))
  )

(define (Task :id Aggregate)
  (name "Aggregate")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'q-models)
```

```
    (sort Query-Models)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      AGGREGATE-ALL
      )))
  )


(define (Task :id Elaborate-items)
  (name "Elaborate-items")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'q-models)
     (sort Query-Models)))
  (output-roles
   (define (var)
     (name 'item-infos)
     (sort Item-Infos)))
  (competence
   (define (Competence)
     (postconditions
      ELABORATE-ITEM-INFOS
      )))
  )



(define (Skill :id Items-elaboration)
  (name "Items-elaboration")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'q-model)
     (sort Query-models)))
  (output-roles
   (define (var)
     (name 'item-infos)
     (sort Item-Infos)))
  (competence
   (define (Competence)
     (postconditions
      ELABORATE-ITEM-INFOS
      ))))
```

```
(define (Task :id Aggregate-items)
  (name "Aggregate-items")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'item-inf)
     (sort Item-Info)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Item)))
  (competence
   (define (Competence)
     (postconditions
      AGGREGATE-ITEM-INFOS
      )))
  )

(define (Skill :id Arithmetic-mean)
  (name "Arithmetic-mean")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'item-inf)
     (sort Item-Info)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Item)))
  (competence
   (define (Competence)
     (postconditions
      AGGREGATE-WITH-ARITHMETIC-MEAN
      )))
  )

(define (Skill :id Weighted-mean)
  (name "Weighted-mean")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'item-inf)
     (sort Item-Info)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Item)))
  (competence
```

```
   (define (Competence)
     (postconditions
      AGGREGATE-WITH-WEIGHTED-MEAN
      )))
  )

(define (Skill :id OWA)
  (name "OWA")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'item-inf)
     (sort Item-Info)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Item)))
  (competence
   (define (Competence)
     (postconditions
      AGGREGATE-WITH-OWA
      )))
  (knowledge-roles
   Weighting-Function)
  )

(define (Skill :id WOWA)
  (name "WOWA")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'item-inf)
     (sort Item-Info)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Item)))
  (competence
   (define (Competence)
     (postconditions
      AGGREGATE-WITH-WOWA
      )))
  (knowledge-roles
   Weighting-Function))



(define (Task-Decomposer :id Metasearch)
  (name "Metasearch")
  (ontologies ISA-Ontology)
```

```
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      SATISFY-CONSULT
      )))
  (subtasks
   Elaborate-query
   Customise-query
   Retrieve
   Aggregate
   ))

(define (Task-Decomposer :id Metasearch-without-elaboration)
  (name "Metasearch-without-elaboration")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      SATISFY-CONSULT
      )))
  (subtasks
   Customise-query
   Retrieve
   Aggregate
   ))


(define (Task-Decomposer :id Aggregation)
  (name "Aggregation")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'q-models)
     (sort Query-models)))
```

```
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      AGGREGATE-ALL
      )))
  (subtasks
   Elaborate-items
   Aggregate-items
   ))

(define (Task-Decomposer :id PCM-metasearch)
  (name "PCM-Metasearch")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
   (define (Competence)
     (postconditions
      ASSESS-SEARCH-RESULT
      SATISFY-CONSULT
      )))
  (subtasks
   Search
   Critique-search
   Modify-search
   )
  )

(define (Task-Decomposer :id Modify-metasearch)
  (name "Modify-metasearch")
  (ontologies ISA-Ontology)
  (input-roles
   (define (var)
     (name 'consult)
     (sort Query-Model)))
  (output-roles
   (define (var)
     (name 's-items)
     (sort Scored-Items)))
  (competence
```

```
 (define (Competence)
   (postconditions
    SATISFY-CONSULT
    CHANGE-SCOPE
    )))
(subtasks
 Adapt-query
 Customise-query
 Retrieve
 Aggregate
 )
)
```

Appendix F

ORCAS Services

This appendix describes the interaction protocol and the format of the messages to communicate between agents in the ORCAS e-Institution and external agents requesting some ORCAS service.

We distinguish between two types of services:

- information services are used to provide external agents information on the components registered in a library;
- operation services are used by external agents to request institutional agents to perform some action, like configuring a team.

These are the ORCAS operation services:

1. *Brokering*: to obtain a task-configuration satisfying a specification of problem requirements.
2. *Team formation*: to form and instruct a team of agents with the capabilities required by a task-configuration.
3. *Teamwork*: solving a problem by a recently configured team, given a team-identifier.
4. *Cooperative Problem Solving*: this service comprises all the previous services within a single request-inform protocol.

The ORCAS institutional services are accessed through a Personal Assistant (PA) agent, except the informational services, which are provided by the Librarian and can be accessed directly by an external agent. The PA is the mediator between the user and the system, but also between the ORCAS institution and external agents willing to request some of the ORCAS services. The PA agent understands both the ORCAS ontology and the specific application ontology (e.g. the WIM library), freeing the user of knowing them.

In ORCAS the PA role is defined as an external role, since a PA is responsible for interacting with a human user, and needs application specific knowledge in order to support the user during the problem specification stage. Nonetheless, the ORCAS agent platform provides a generic model of PA that is equipped with the social knowledge required to participate in the ORCAS e-Institution. This PA acts as a broker with respect to other agent willing to use the ORCAS services. In spite of learning the different scenes of the ORCAS institution and communicating other agents directly, an external requester have to communicate only with the PA.

Since the language used by an external agent may differ from the local communication language, there is another kind of agent that mediates between the external agent, and the PA running locally: the FIPA-Mediator. The ORCAS-proxy is responsible for adapting the language used by the external agent to the local language, and viceversa; specifically, it is able to translate messages from the FIPA-ACL to the NOOS ACL, and viceversa. The content is encoded using the FIPA Specification Language (SL) and either XML or RDF to serialize the data.

We focus now on the technical aspects required by external agents to use the ORCAS services through the FIPA-Mediator agent. §F.1 describes the data and protocol for the different services. §F.2 deals with the ontologies and format of the data required by the interaction protocols. §F.4 three contains examples of the FIPA-ACL messages to be interchanged.

F.1 Interaction protocols for the ORCAS services

Figure F.1 shows the interaction protocols for the different ORCAS services using the FIPA style. These diagrams are called Message Sequence Charts (MSC). Each vertical line represents the time running, and horizontal lines represent messages, the vertical rectangles represent agent processing operations, and the rhombuses represent choice points.

Figure F.1.a) shows the MSC for the *Brokering* sevice: The MAS participating in the ORCAS e-institution is configured at the knowledge-level, using the ORCAS-KMF as the Agent Capability Description Language. The external, FIPA-compliant agent, sends a request message with a problem-specification and receives a task-configuration.

Figure F.1.b) shows the MSC for the *Team Formation* service: a team of problem-solving agents is formed and instructed to solve problems according to a task configuration. The external client sends a request message with a task-configuration and receives the identifier of the already formed team.

Figure F.1.c) shows the MSC for the Teamwork service: a problem is solved by a team of problem-solving agents. The team should be previously formed, ensuring that team members have committed to solve the tasks required by the task-configuration in cooperation with the other team-mates. The external FIPA compliant client sends a request message with a problem instance and the identifier of a previously formed team. Data structures for this protocol are

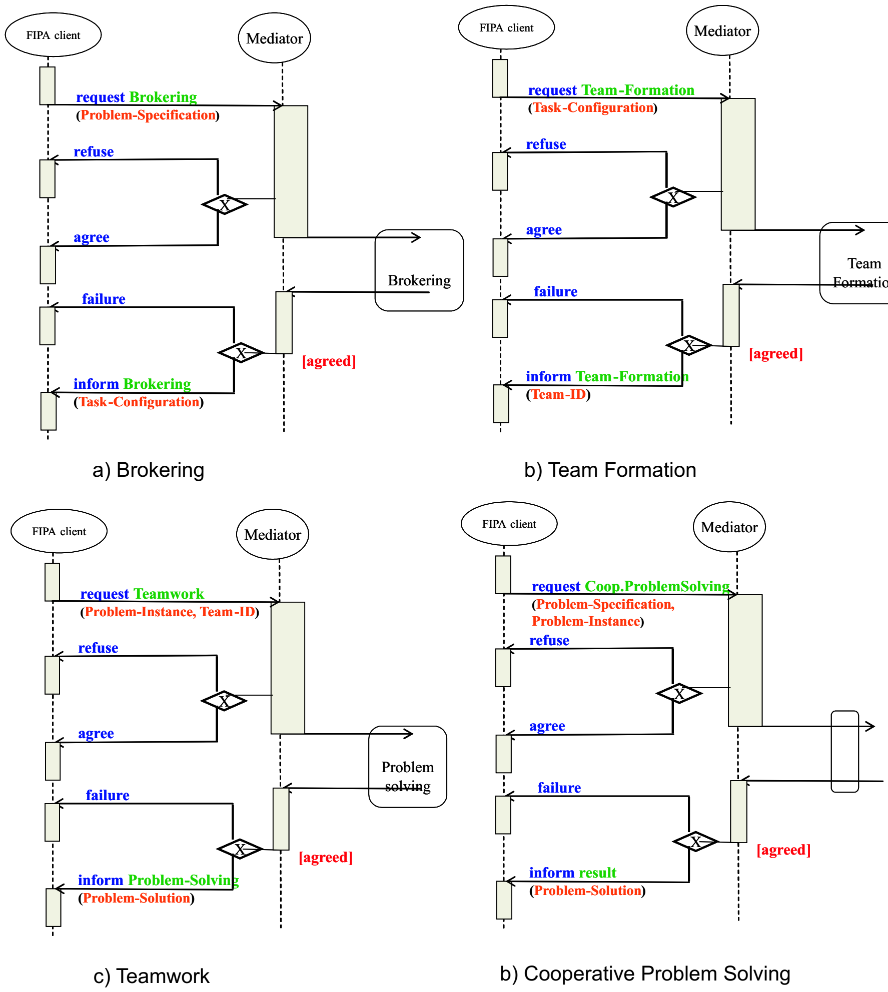

Figure F.1: FIPA Message Sequence Charts for the ORCAS services described in §F.2.

Figure F.1.d) shows the MSC for the Cooperative Problem-Solving service: The requester sends a request message to the FIPA-Mediator containing a problem specification that is made up of the problem requirements and the problem data. If all goes right with the protocol, at the end the PA sends the result to the FIPA-Mediator, which serializes the data and sends an inform with the result, a set of scored items.

F.2 Data structures and XML format

Problem-Specification			
Parameter	**Description**	**Type**	**Cardinality**
task-name	Name of the task to be configured	String	Single
preconditions	Constraints over the inputs of the task	Signature-element	Multiple
postconditions	Constraints over the output of the task, relations between ouput and input	Signature-element	Multiple
input-roles	Inputs of task	Formula	Multiple
knowledge-roles	Domain models of the knowledge to be used	Signature-element	Multiple

Table F.1: Problem specification

Task-Configuration			
Parameter	**Description**	**Type**	**Cardinality**
task-name	Name of the task being configured	Symbol	Single
input-roles	Input roles of the task	Signature-element	Multiple
capability-configuration	A configuration for a capability bound to the task.	Capability-configuration	Single

Table F.2: Task configuration

This is XML grammar for the content language to be used when accessing ORCAS services.

```
Problem-Specification:=
   <problem-specification>
```

TD-Configuration (is a Capability-Configuration)			
Parameter	**Description**	**Type**	**Cardinality**
capability-name	Name of the capability being configured	Symbol	Multiple
input-roles	Input roles of the Task Decomposer	Signature-Element	Multiple
subtasks-configuration	Configuration for the subtasks	Task-configuration	Multiple
operational-description	Intermediate roles and program description	Operational-Description	Single

Table F.3: Task-Decomposer configuration

Skill-Configuration (is a Capability-Configuration)			
Parameter	**Description**	**Type**	**Cardinality**
capability-name	Name of the capability being configured	Symbol	Multiple
input-roles	Input roles of the Skill	Signature-Element	Multiple
domain-models	Names of the domain-models to be used	Symbol	Multiple
knowledge-roles	Domain models to be used by the capability	Signature-element	Multiple

Table F.4: Skill configuration

```
        <task-name> Symbol</task-name>
        <preconditions>Formula*</preconditions>
        <postconditions>Formula*</postconditions>
        <inputs>Signature-Element*</preconditions>
        <knowledge-roles>Signature-Element*</ knowledge-roles >
    </problem-specification>

Task-Configuration :=
    <task-configuration>
        <task-name> Symbol </task-name>
        <capability-configuration>Capability-Configuration</capability-configuration>
    </task-configuration>

Capability-Configuration:= Task-Decomposer-Configuration |
Skill-Configuration

Task-Decomposer-Configuration :=
    <task-decomposer-configuration>
        <capability-name> Symbol </capability-name>
        <subtasks-configuration>Task-Configuration</subtasks-configuration>
    </Task-Decomposer-configuration>

Skill-Configuration :=
    <skill-configuration>
        <capability-name> Symbol </task-id>
        <domain-models>Symbol*</domain-models>
    </skill-configuration>

Formula:= <formula>String</formula>

Signature-Element:= <signature-element>String</signature-element>

Symbol:= String
```

F.3 ORCAS services in the WIM application

This section describes the aspects of the WIM ontology to be used by external clients requesting for some of the ORCAS services in the WIM application. Only those concepts used to communicate with the FIPA-Mediator are considered, which are basically keyword-based queries and scored-items, while other concepts concepts not required by an external agent to communicate with WIM are omitted..

```
User-Consult:=
        <user-consult>
            <query>Domain-Query </query>
            <sources>Source*</source>
        </user-consult>

Domain-Query :=
        <query>
            <keywords> Keyword* </keywords>
            <filters>Filter*</filters>
            <category>Category</category>
        </query>

Filter:=
        <filter>
            <attribute>Attribute</attribute>
            <value>String</value>
        </filter >
```

```
Source:=    <source>Symbol</source>

Keyword:=   <keyword>Symbol</keyword>

Scored-Item :=
        <scored-item>
            <item>Item</item>
            <score>Number</score>
        </scored-item>

Item :=
        <item>
            <id>String </id>
            <title>String</title>
            <author>String</author>
        </item>

Source := Pubmed | Medline-IGM | Healthstar-IGM | ISOCO

Category:=  Good-Evidence | Medium-Evidence |
            Poor-Evidence | Evidence

Attribute:= Author Name | Begin Year | End Year |
            Publication Type | Language | Journal
```

F.4 FIPA examples

F.4.1 Brokering

```
(request
    :sender     (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE)
    :receiver   (set (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS))
    :reply-with     configuration-request18236
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
       (action
        (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE)
        (Brokering
        :problem-specification
        (Problem-Specification
        :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?> :value
        <problem-specification>
            <task-name> Search </task-name>
            <postconditions>
                <formula>Satisfy-Consult</formula>
                <formula>Non-Exhaustive-Customization</formula>
                <formula>Aggregate-With-Arithmetic-mean</formula>
            </postconditions>
        <input-roles>
            <signature-element>Query-Model</signature-element>
        </input-roles>
        <knowledge-roles>
             <signature-element>Source-Descriptions</signature-element>
        </knowledge-roles>
        </problem-specification>))))

(agree ...)

(inform
    :sender     (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS)
    :receiver   (set (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE))
```

```
    :reply-with     configuration-request18236
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
       (result
          (action )
          (Task-configuration
        :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?>
        :value  <task-configuration></task-configuration>))))
```

F.4.2 Team formation

```
 (request
    :sender     (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE)
    :receiver   (set (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS))
    :reply-with     team-request 18237
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
    (action
        (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS)
        (Team-Formation
        :task-configuration
        (Task-Configuration
        :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?> :value
        <task-configuration>...</task-configuration>))))))

(agree ...)

(inform
    :sender     (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS)
    :receiver   (set (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE))
    :reply-with  team-request 18237
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
       (result
          (action ... )
            (Team-ID
            :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?> :value
            <team-id> Team-18237 </team-id>))))
```

F.4.3 Problem-Solving

```
(request
    :sender     (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE)
    :receiver   (set (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS))
    :reply-with     problem-solving-request18236
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
       (action
        (agent-identifier :name \wim\-Proxy@wim.iiia.csic.es:7778/NOOS)
```

```
    (Problem-Solving
       :problem-instance
       (Problem-Instance
       :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?> :value
       <user-consult>
           <query>
               <keywords>
                   <keyword>Ofloxacin</keyword>
                   <keyword>Ofloxacin</keyword>
                   <keyword>Guidelines</keyword>
               </keywords>
           </query>
           <sources>
               <source>Pubmed</source>
               <source>ISOCO<source>
           </sources>
       </user-consul
      :team-ID
      (Team-ID:
         :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?> :value
         <team-id> Team-18237 </team-id>))))

(agree ...)

(inform
    :sender     (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS)
    :receiver   (set (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE))
    :reply-with     problem-solving-request18236
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
       (result
        (action ...)
        (Problem-Solution
        :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?>
        :value
        <scored-items>...</scored-items>))))
```

F.4.4 Cooperative Problem-Solving

```
(request
    :sender     (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE)
    :receiver   (set (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS))
    :reply-with     configuration-request18236
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
       (action
        (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE)
        (FullProblemSolving
        :problem-specification
           (ProblemSpecification
           :encoding  <xml? version="1.0" encoding ="ISO-8859-1"> :value
           <problem-specification>
               <task-name> Search </task-name>
               <postconditions>
                   <formula>Satisfy-Consult</formula>
                   <formula>Non-Exhaustive-Customization</formula>
                   <formula>Aggregate-With-Arithmetic-mean</formula>
               </postconditions>
           <input-roles>
```

```
            <signature-element>Query-Model</signature-element>
        </input-roles>
        <knowledge-roles>
             <signature-element>Source-Descriptions</signature-element>
        </knowledge-roles>
        </problem-specification>)
      :problem-instance
         (Problem-Instance
        :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?> :value
        <user-consult>
            <query>
                <keywords>
                    <keyword>Ofloxacin</keyword>
                    <keyword>Ofloxacin</keyword>
                    <keyword>Guidelines</keyword>
                </keywords>
            </query>
            <sources>
                <source>Pubmed</source>
                <source>ISOCO<source>
            </sources>
        </user-consul
       :team-ID
       (Team-ID:
          :encoding  <xml? version='1.0' encoding ='ISO-8859-1'?> :value
          <team-id> Team-18237 </team-id>))))

(agree ...)


(inform
    :sender     (agent-identifier :name WIM-Proxy@wim.iiia.csic.es:7778/NOOS)
    :receiver   (set (agent-identifier :name uva-agent@a1136.fmg-uva-nl:1099/JADE))
    :reply-with     problem-solving-request18236
    :encoding   String
    :language   FIPA-SL0
    :ontology   WIM-Ontology
    :protocol   FIPA-request
    :conversation-id    configuration18236
    :content
       (result
         (action ... )
         (Problem Solution: :encoding  <xml? version="1.0"
            encoding ="ISO-8859-1"?> :value
            <scored-items>
                <scored-item>
                    <item>
                       <Identifier>PMID1756688</Identifier>
                       <Title>Treatment of lower respiratory infections in outpatients
                       with ofloxacin compared with erythromycin.</title>
                       <Author>Peugeot RL, Lipsky BA, Hooton TM, Pecoraro RE.</Author>
                    </item>
                    <score>0.02</score>
                </scored-item>
                <scored-item>
                    <item>
                       <Identifier>PMID1864291</Identifier>
                       <Title>Role of quinolones in the treatment of bronchopulmonary infections,
                        particularly pneumococcal and community-acquired pneumonia..</Title>
                       <Author>Thys JP, Jacobs F, Byl B.</Author>
                    </item>
                    <score>0.02</score>
                </scored-item>
            </scored-items>))))
```

F.5 The Personal Assistant

The Personal Assistant (PA) mediates between the human user or an external agent and the other agents (institutional and PSAs). The PA helps the user specifying a problem by using the user's domain ontology, and avoids him knowing technical details like the agent communication language and the interaction protocols (scenes in the ORCAS e-Institution underlying the WIM application). Specifically, the PA is able to transform the user specification of the problem into a problem specification using the ORCAS Agent Capability Description Language (ACDL). This problem specification contains the problem requirements to be used by the institutional agents to form a new team of agents that is able to solve the problem at hand according to the requirements specified by the user.

The PA brings an added value to the WIM services, for it is able to organize the user tasks as a collection of interests and goals and schedule them to update the results periodically. An interest refers to a topic or a subject the user is interested in while goals are specific issues the user wants to search information on and are represented as specific queries to look up on bibliographic databases. An interest is specified as a collection of goals together with a set of preferences (problem requirements, configuration strategy, scheduling options, etc.), and each goal is specified as a consultation (keywords, filters, category, information sources, etc.), plus a set of preferences. The preferences of a goal are inherited from the interest, but they can be refined for any particular goal. We will show some examples of the functionality offered by the PA to human users through a Web interfaced defined for the WIM application. In order to provide a Web interface to the application, we have connected the ORCAS e-Institution to an *http server* through a pseudo agent called the *www-mediator*, as showed in Figure F.2.

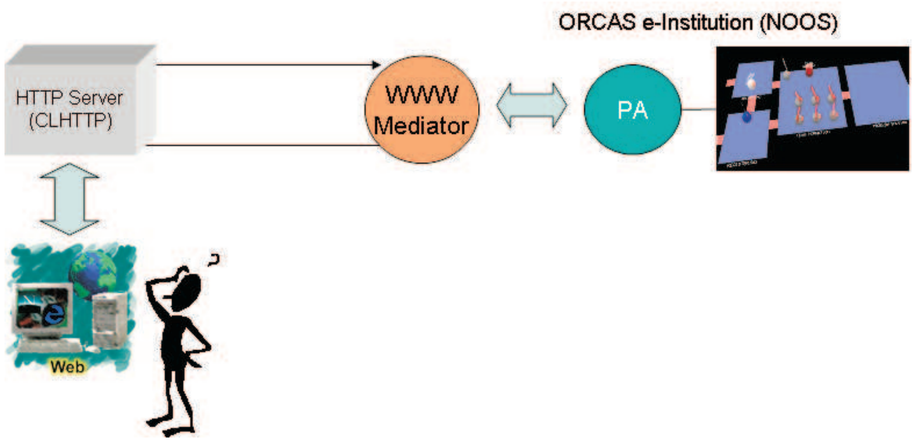

Figure F.2: Web interface to WIM

The http-server calls a function of the www-mediator using data obtained from a Web form, and builds a results http page using the data returned by the www-mediator after communicating with the PA. Therefore, the www-mediator can be seen as an adaptor or wrapper that agentifies the http-server so as to allow the PA to interact with it. The www-mediator operates by accepting function calls from the http-server, translating the Web data format to the data format used in the NOOS agent platform, and communicating with the PA using the agent communication language.

Figure F.3 shows an example of a user interested in three topics: *Cannabis*, *AIDS*, and *Pneumonia prognosis and therapy*. On the right side, the figure shows the specification of the interest called *AIDS*, which has two goals: *AIDS classification* and *AIDS therapy*.

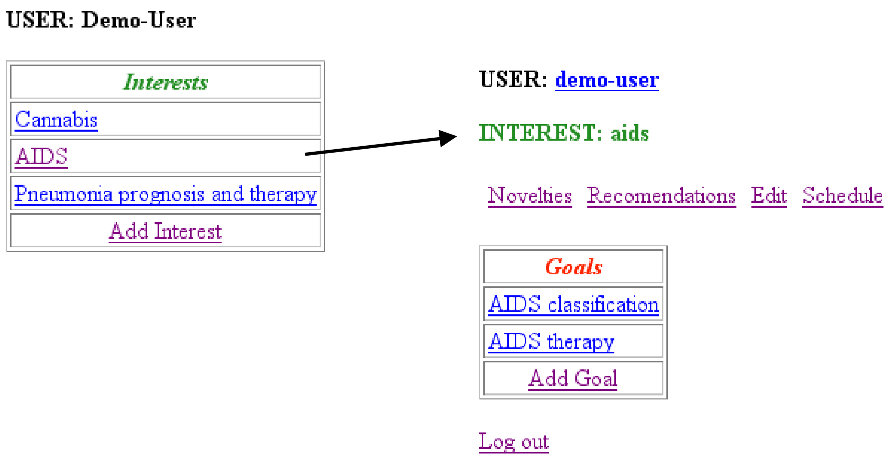

Figure F.3: Managing Interests and goals

Figure F.4 shows the interface used to edit one goal. The user can specify up to three keywords to characterize the subject of the information search, together with a category from the knowledge base on Evidence Based Medicine (EBM). Categories can be used to enrich the user's query and rank the information retrieved in form of scored items according to some of the subjects typically required by EBM practitioners (e.g. references on diagnosis and therapy, or clinical protocols), or by a desired degree of evidence for the references to be retrieved (this is defined by a three-levels ordinal scale: Good, Medium, and Poor Evidence Quality). The user may also specify requirements, such as the type of query elaboration desired and the aggregation operator preferred. Search filters like publication date periods (Begin-Year and End-Year), author name, publication type, language, and journal.

In addition, goals can be scheduled by the PA to repeat the consultation periodically, allowing the user to automatically update the results for each goal. Figure F.5 shows the interface used to schedule the execution of one goal. A goal

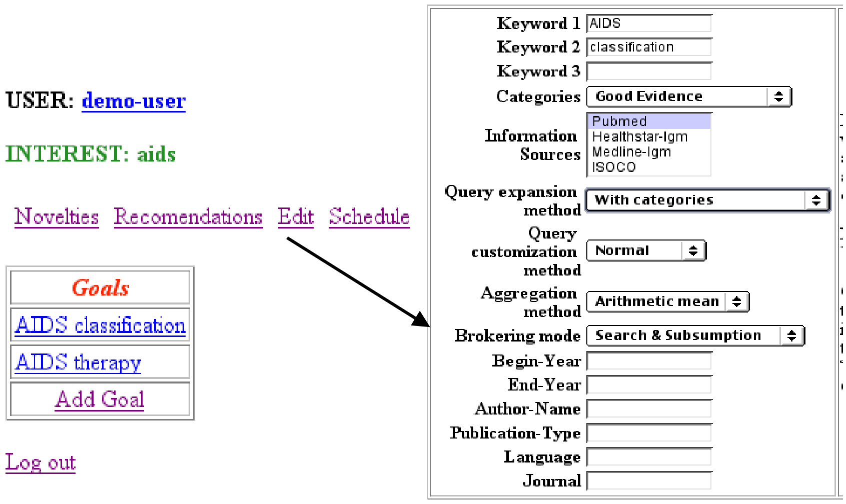

Figure F.4: Goal editing

can be defined as permanent or volatile (executed only once). When it is stated as permanent, the user can specify the typical schedulling options for periodic execution of tasks, such as time, week-day and month-day.

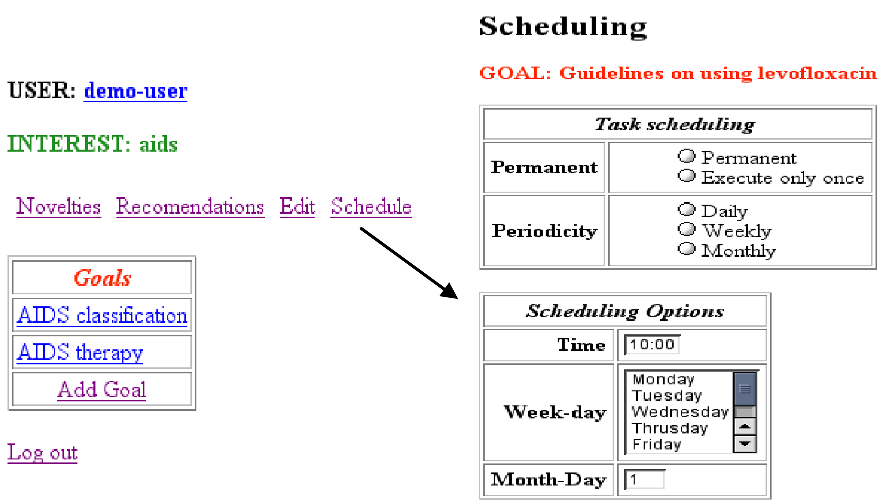

Figure F.5: Scheduling

Appendix G

Glossary of abbreviations

ACDL Agent Capability Description Language

CPS Cooperative Problem Solving (a process involving agents)

DAML DARPA Agent Markup Language

EBM Evidence-Based Medicine

HTN Hierarchical Task Network (a type of planning)

IGM Internet Grateful Med (a Web-based search-engine)

ISA Information Search Library

KB Knowledge-Broker (an ORCAS agent role)

KMF Knowledge Modelling Framework

KMO Knowledge Modelling Ontology

MAS Multi Agent Systems

MSC Message Chart Diagram

MeSH Medical Subject Headings

ORCAS Open, Reusable and Configurable multi-Agent Systems

OWA Ordered Weighted Average

PA Personal Assistant (an ORCAS agent role)

PSA Problem-Solving Agent (an ORCAS agent role)

PSM Problem Solving Method

Pubmed Public Medline (a Web-based search-engine)

SWS Semantic Web Services

TB Team-Broker (an ORCAS agent role)

TMD Task-Method-Domain (a model used in Knowledge Modelling)

WIM Web Information Mediator (an ORCAS based application to search information in the Internet)

WOWA Weighted OWA

Bibliography

[Abasolo and Gómez, 2000] Abasolo, C. and Gómez, M. (2000). Melisa: An ontology-based agent for information retrieval in medicine. In *Proceedings of the First International Workshop on the Semantic Web (SemWeb2000)*, pages 73–82.

[Aitken et al., 1998] Aitken, S., Filby, I., Kingston, J., and Tate, A. (1998). Capability descriptions for problem-solving methods. Submitted to the European Conference on AI.

[Ankolekar et al., 2002] Ankolekar, A., Huch, F., and Sycara, K. P. (2002). Concurrent semantics for the web services specification language DAML-s. In *Coordination Models and Languages*, pages 14–21.

[Arcos, 1997] Arcos, J. L. (1997). *The Noos representation language.* Monografies del iiia, Universitat Politècnica de Catalunya.

[Arcos, 2001] Arcos, J. L. (2001). T-air: A case-based reasoning system for designing chemical absorption plants. In Aha, D. W. and Watson, I., editors, *Case-Based Reasoning Research and Development*, number 2080 in Lecture Notes in Artificial Intelligence, pages 576–588. Springer-Verlag.

[Arcos and López de Mántaras, 1997] Arcos, J. L. and López de Mántaras, R. (1997). Perspectives: a declarative bias mechanism for case retrieval. In Leake, D. and Plaza, E., editors, *Case-Based Reasoning. Research and Development*, number 1266 in Lecture Notes in Artificial Intelligence, pages 279–290. Springer-Verlag.

[Arcos et al., 1998] Arcos, J. L., López de Mántaras, R., and Serra, X. (1998). Saxex : a case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27 (3):194–210.

[Arens et al., 1993] Arens, Y., Chee, C. Y., Hsu, C.-N., and Knoblock, C. A. (1993). Retrieving and integrating data from multiple information sources. *International Journal of Cooperative Information Systems*, 2(2):127–158.

[Armengol and Plaza, 1997] Armengol, E. and Plaza, E. (1997). Induction of feature terms with INDIE. In van Someren, M. and Widmer, G., editors, *Euro-*

pean Conference on Machine Learning, Lecture Notes in Artificial Intelligence. Springer-Verlag.

[Armengol and Plaza, 2001] Armengol, E. and Plaza, E. (2001). Similarity assessment for relational CBR. In *Case-Based Reasoning Research and Development*, volume 2080 of *Lecture Notes in Computer Science*, pages 44–58.

[Atkinson, 1997] Atkinson, S. (1997). *Engineering Software Library Systems*. PhD thesis, School of Information Technology, University of Queensland, 1997. To appear.

[Bansal and Vidal, 2003] Bansal, S. and Vidal, J. M. (2003). Matchmaking of web services based on the DAML-S service model. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*.

[Barbuceanu and Fox, 1995] Barbuceanu, M. and Fox, M. (1995). COOL: a language for describing coordination in multi-agent systems. In *Proceedings of the First International Conference in Multi-Agent Systems ICMAS'95*, pages 17–24. AAAI Press.

[Bastide et al., 1999] Bastide, R., Sy, O., and Palanque, P. (1999). Formal specification and prototyping of corba systems. In *Proceedings of 13th European Conference on Object-Oriented Programming*, volume 1628 of *Lecture Notes in Computer Science*, pages 474–494. Springer-Verlag.

[Beckers et al., 1994] Beckers, R., Holland, O., and Deneubourg, J. (1994). From local actions to global tasks: Stigmergy and collective robotics. In *Artificial Life IV. 4th International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press.

[Benjamins, 1993] Benjamins, R. (1993). *Problem Solving Methods for Diagnosis*. PhD thesis, University of Amsterdam.

[Benjamins, 1997] Benjamins, R. (1997). Problem-solving methods in cyberspace. In *Proceedingsof the Workshop on Problem-Solving Methods for Knowledge-based Systems of IJCAI*.

[Benjamins et al., 1996a] Benjamins, R., Fensel, D., and Chandrasekaran, B. (1996a). PSMs do IT. Summary of track on Sharable and Reusable Problem-Solving Methods. KAW'96.

[Benjamins et al., 1999] Benjamins, R., Wielinga, B., Wielemaker, J., and Fensel, D. (1999). Towards brokering problem-solving knowledge on the internet. In *Knowledge Acquisition, Modeling and Management*, pages 33–48.

[Benjamins et al., 1996b] Benjamins, V., de Barros, L., and Andre, V. (1996b). Constructing planners through problem-solving methods. In *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*.

[Benjamins et al., 1998] Benjamins, V., Plaza, E., Motta, E., Fensel, D., Studer, R., Wielinga, B., Schreiber, G., Zdrahal, Z., and Decker, S. (1998). Ibrow3: An intelligent brokering service for knowledge-component reuse on the world-wide web.

[Benjamins et al., 1996c] Benjamins, V. R., Fensel, D., and Straatman, R. (1996c). Assumptions of problem-solving methods and their role in knowledge engineering. In *European Conference on Artificial Intelligence*, pages 408–412.

[Berners-Lee et al., 1999] Berners-Lee, T., Fischetti, M., and Dertouzos, M. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor.* Harper, San Francisco.

[Biggerstaff and Perlis, 1989] Biggerstaff, T. J. and Perlis, A. J., editors (1989). *Software Reusability.* ACM press.

[Birmingham et al., 1995] Birmingham, W. P., Durfee, E. H., Mullen, T., and Wellman, M. P. (1995). The distributed agent architecture of the university of michigan digital library (extended abstract). In *AAAI Spring Symposium on Information Gathering.*

[Bond and Gasser, 1988a] Bond, A. and Gasser, L., editors (1988a). *Readings in distributed Artificial Intelligence.* Morgan Kaufmann Publishers.

[Bond and Gasser, 1988b] Bond, A. H. and Gasser, L. (1988b). An analysis of problems and research in dai. In Bond, A. H. and Gasser, L., editors, *Readings in Distributed Artificial Intelligence*, pages 61–70. Kaufmann, San Mateo, CA.

[Börstler, 1995] Börstler, J. (1995). Feature-oriented classification for software reuse. In *Proceedings Seventh International Conference of Software Engineering and Knowledge Engineering.*

[Bratman, 1988] Bratman, M. E. (1988). Plans and resource bounded practical reasoning. *Computational Intelligence*, 4:249–355.

[Bratman, 1990] Bratman, M. E. (1990). What is intention? In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, pages 15–31. MIT Press, Cambridge, MA.

[Bratman, 1992] Bratman, M. E. (1992). Shared cooperative activity. *Philosophical Review*, 101:327–341.

[Bratman et al., 1991] Bratman, M. E., Israel, D., and Pollack, M. (1991). Plans and resource-bounded practical reasoning. In Cummins, R. and Pollock, J. L., editors, *Philosophy and AI: Essays at the Interface*, pages 1–22. The MIT Press, Cambridge, Massachusetts.

[Brazier et al., 1997] Brazier, F. M. T., Dunin-Keplicz, B. M., Jennings, N. R., and Treur, J. (1997). DESIRE: Modelling multi-agent systems in a compositional formal framework. *International Journal of Cooperative Information Systems*, 6(1):67–94.

[Brazier et al., 2002] Brazier, F. M. T., Jonker, C. M., and Treur, J. (2002). Principles of component-based design of intelligent agents. *Data Knowledge Engineering*, 41(1):1–27.

[Breuker, 1994] Breuker, J. (1994). A suite of problem types. In Breuker, J. and Van de Velde, W. W., editors, *CommonKADS Library for Expertise Modeling*, volume 21 of *Frontiers in Artificial Intelligence and Applications*, pages 57–88. IOS-Press.

[Breuker and Van de Velde, 1994] Breuker, J. and Van de Velde, W., editors (1994). *COMMONKADS Library for Expertise Modelling*. IOS Press.

[Brown and Wallnau, 1996] Brown, A. W. and Wallnau, K. C. (1996). Engineering of component-based systems. In *Component-Based Software Engineering: Selected Papers from the Software Engineering Institute*, pages 7–15. IEEE Computer Society Press.

[Bryson et al., 2002] Bryson, J. J., Martin, D., McIlraith, S., and Stein, L. A. (2002). Agent-based composite services in daml-s: The behavior-oriented design of an intelligent semantic web. In Zhong, N., Liu, J., and Yao, Y., editors, *Web Intelligence*. Springer-Verlag.

[Buchanan et al., 1983] Buchanan, B. et al. (1983). Constructing an expert system. In Hayes-Roth, F., Waterman, D., and D.Lenat, editors, *Building Expert Systems*. Addison-Wesley.

[Burmeister, 1996] Burmeister, B. (1996). Models and methodology for agent-oriented analysis and design. In Fischer, K., editor, *Proceedings of the Workshop on Agent-Oriented Programming and Distributed Systems*. DFKI Document D-96-06.

[Bussler et al., 2002] Bussler, C., Maedche, A., and Fensel, D. (2002). A conceptual architecture for semantic web enabled web services. *SIGMOD Record*, 31(4):24–29.

[Butler and Duke, 1998] Butler, S. and Duke, R. (1998). Defining composition operators for object interaction. *Object Oriented Systems*, 5(1):1–16.

[Bylander and Chandrasekaran, 1988] Bylander, T. and Chandrasekaran, B. (1988). Generic tasks in knowledge-based reasoning: The right level of abstraction for knowledge acquisition. In Gaines, B. and Boose, J., editors, *Knowledge Acquisition for Knowledge Based Systems*, volume 1, pages 65–77. Academic Press.

[Cammarata et al., 1983] Cammarata, S., MacArthur, D., and Steeb, R. (1983). Strategies of cooperation in distributed problem solving. In *Proceedins of the Eighth International Joint Conference on Artificial Intelligence.*

[Canal et al., 2001] Canal, C., Fuentes, L., Pimentel, E., Troya, J. M., and Vallecillo, A. (2001). Extending CORBA interfaces with protocols. *The Computer Journal*, 44(5):448–462.

[Carbonell, 2000] Carbonell, J. (2000). ISMIS invited talk.

[Cardoso and Sheth, 2002] Cardoso, J. and Sheth, A. P. (2002). Semantic e-workflow composition. Technical report, LSDIS Lab, Department of Computer Science, University of Georgia.

[Chandrasekaran, 1986] Chandrasekaran, B. (1986). Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. *IEEE Expert*, 1:23–30.

[Chandrasekaran, 1987] Chandrasekaran, B. (1987). Towards a functional architecture for intelligence based on generic information processing tasks. In *International Joint Conference on Artificial Intelligence*, pages 1183–1192.

[Chandrasekaran, 1990] Chandrasekaran, B. (1990). Design problem solving: A task analysis. *AI Magazine*, 11(4):59–71.

[Chandrasekaran and Johnson, 1993] Chandrasekaran, B. and Johnson, T. (1993). Generic tasks and task structures: History, critique and new directions. In David, J., Krivine, J., and Simmons, R., editors, *Second Generation Expert Systems*, pages 239–280. Springer-Verlag.

[Chandrasekaran et al., 1992] Chandrasekaran, B., Johnson, T., and Smith, J. (1992). Task structure analysis for knowledge modeling. *Communications of the ACM*, 33(9):124–136.

[Chandrasekaran et al., 1998] Chandrasekaran, B., Josephson, J., and Benjamins, R. (1998). The ontology of tasks and methods. In *In Proceedings of the 11th Knowledge Acquisition Modeling and Management Workshop, KAW'98, Banff, Canada, April 1998.*

[Chawathe et al., 1994] Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom, J. (1994). The TSIMMIS project: Integration of heterogeneous information sources. In *16th Meeting of the Information Processing Society of Japan*, pages 7–18.

[Cheyer and Martin, 2001] Cheyer, A. and Martin, D. (2001). The open agent architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1):143–148. OAA.

[Clancey, 1989] Clancey, W. (1989). The knowlede level reinterpreted. *Machine Learning*, 4:285–291.

[Clement and Durfee, 1999] Clement, B. J. and Durfee, E. H. (1999). Top-down search for coordinating the hierarchical plans or multiple agents. In Etzioni, O., Muller, J. P., and Bradshaw, J. M., editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 252–259, Seattle, WA, USA. ACM Press.

[Clements, 1996] Clements, P. C. (1996). From subroutines to subsystems: Component-based software development. In *Component-Based Software Engineering: Selected Papers from the Software Engineering Institute*, pages 3–6. IEEE Computer Society Press.

[Cohen and Levesque, 1990] Cohen, P. R. and Levesque, H. J. (1990). Persistence, intention, and commitment. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, pages 33–69. MIT Press, Cambridge, MA.

[Cohen and Levesque, 1991] Cohen, P. R. and Levesque, H. J. (1991). Teamwork. *Nous*, 25(4):487–512.

[Decker, 1996] Decker, K. (1996). TAEMS: A Framework for Environment Centered Analysis and Design of Coordination Mechanisms. In *Foundations of Distributed Artificial Intelligence*, pages 429–448. G. O'Hare and N. Jennings (eds.), Wiley Inter-Science.

[Decker and Lesser, 1995] Decker, K. and Lesser, V. R. (1995). Designing a family of coordination algorithms. In Lesser, V., editor, *Proceedings of the First International Conference on Multi-Agent Systems*, pages 73–80, San Francisco, CA, USA. The MIT Press: Cambridge, MA, USA.

[Decker et al., 1997a] Decker, K., Pannu, A., Sycara, K., and Williamson, M. (1997a). Designing behaviors for information agents. In *Proceedings of the 1st International Conference on Autonomous Agents*, pages 404–412. ACM Press.

[Decker et al., 1997b] Decker, K., Sycara, K., and Williamson, M. (1997b). Middle-agents for the internet. In *Proceedings the 15th International Joint Conference on Artificial Intelligence*, pages 578–583.

[Decker et al., 1996] Decker, K., Williamson, M., and Sycara, K. (1996). Matchmaking and brokering. In *Proceedings of the 2nd International Conference in Multi-Agent Systems*.

[Decker and Lesser, 1992] Decker, K. S. and Lesser, V. R. (1992). Generalizing the partial global planning algorithm. *International Journal of Intelligent and Cooperative Information Systems*, 1(2):319–346.

[Dellarocas, 2000] Dellarocas, C. (2000). Contractual agent societies negotiated shared connote and social control in open multi-agent systems. In *Proceedings of the Workshop on Norms and Institutions in Multi-Agent Systems, ICMAS'02*.

[Dellarocas and Klein, 1999] Dellarocas, C. and Klein, M. (1999). Civil agent societies: Tools for inventing open agent-mediated electronic marketplaces. In *Proceedings ACM Conference on Electronic Commerce (EC-99)*.

[Dignum et al., 2001] Dignum, F., Dunin-Keplicz, B., and Verbrugge, R. (2001). Agent theory for team formation by dialogue. In *Intelligent Agents VII, Agent Theories Architectures and Languages*, volume 1986 of *Lecture Notes in Artificial Intelligence*, pages 150–166. Springer-Verlag.

[Dignum et al., 2002] Dignum, V., Meyer, J.-J., Weigand, H., and Dignum, F. (2002). An organization-oriented model for agent societies. In *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications*.

[d'Inverno et al., 1997] d'Inverno, M., Fisher, M., Lomuscio, A., Luck, M., de Rijke, M., Ryan, M., and Wooldridge, M. (1997). Formalisms for multi-agent systems. *Knowledge Engineering Review*, 12(3).

[d'Inverno et al., 1998] d'Inverno, M., Kinny, D., and M.Luck (1998). Interaction protocols in agentis. In *Proceedings of the Third International Conference on Multi-Agent Systems ICMAS'98*, pages 112–119.

[Doran and Palmer, 1995] Doran, J. and Palmer, M. (1995). The eos project: modelling prehistoric sociocultural trajectories. In *Aplicaciones informaticas en Arqueologia: Teoria y Sistemas. Proceedings of First International Symposium on Computing and Archaeology (1991)*, volume 1.

[Doran et al., 1997] Doran, J., S.Franklin, Jennings, N., and T.J.Norman (1997). On cooperation in multi-agent systems. *The Knowledge Engineering Review*, 12(3):1–6. Panel Discussion at the First UK Workshop on Foundations of Multi-Agent Systems.

[Duke et al., 1991] Duke, R., King, P., Rose, G., and Smith, G. (1991). The object-z specification language. Technical report, Department of Computer Science, University of Queensland.

[Durfee, 1988] Durfee, E. H. (1988). *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers.

[Durfee and Lesser, 1989] Durfee, E. H. and Lesser, V. (1989). Negotiating task decomposition and allocation using partial global planning. In Gasser, L. and Huhn, M., editors, *Distributed Artificial Intelligence Volume II*, pages 229–244. Pitman Publishing.

[Durfee et al., 1998] Durfee, E. H., Mullen, T., Park, S., Vidal, J. M., and Weinstein, P. (1998). The dynamics of the UMDL service market society. In Klusch, M. and Weiss, G., editors, *Cooperative Information Agents II*, Lecture Notes in Artificial Intelligence, pages 55–78. Springer.

[Ephrati and Rosenschein, 1996] Ephrati, E. and Rosenschein, J. S. (1996). Deriving consensus in multiagent systems. *Artificial Intelligence*, 87(1-2):21–74.

[Erickson, 1996a] Erickson, T. (1996a). An agent-based framework for interoperability. In Bradshaw, J. M., editor, *Software Agents*. AAAI Press.

[Erickson, 1996b] Erickson, T. (1996b). Designing agents as if people mattered. In Bradshaw, J. M., editor, *Software Agents*. AAAI Press.

[Eriksson et al., 1995] Eriksson, H., Shahar, Y., Tu, S. W., Puerta, A. R., and Musen, M. A. (1995). Task modeling with reusable problem-solving methods. *Artificial Intelligence*, 79(2):293–326.

[Erol, 1995] Erol, K. (1995). *Hierarchical Task Network Planning: Formalization, Analysis and Implementation.* PhD thesis, University of Maryland.

[Erol et al., 1994] Erol, K., Hendler, J., and Nau, D. S. (1994). HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1123–1128, Seattle, Washington, USA. AAAI Press/MIT Press.

[Esteva, 1997] Esteva, M. (1997). *Electronic Institutions: From Specification to Development*. PhD thesis, Universitat Autnoma de Barcelona.

[Esteva et al., 2002a] Esteva, M., de la Cruz, D., and Sierra, C. (2002a). Islander: an electronic institutions editor. In *Proceedings 1th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1045–1052.

[Esteva et al., 2002b] Esteva, M., Padget, J., and Sierra, C. (2002b). Formalizing a language for institutions and norms. In *Intelligent Agents VIII: Lecture Notes in Artificial Intelligence*, volume 2333 of *Lecture Notes in Artificial Intelligence*, pages 348–366. Springer-Verlag.

[Esteva et al., 2001] Esteva, M., Rodriguez, J. A., Sierra, C., Garcia, P., and Arcos, J. L. (2001). On the formal specifications of electronic institutions. In *Agent-mediated Electronic commerce. The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Artificial Intelligence*, pages 126–147.

[Euzenat, 2001] Euzenat, J. (2001). An infrastructure for formally ensuring interoperability in a heterogeneous semantic web. In *Proc. 1st internationnal on semantic web working symposium (SWWS), Stanford (CA US)*, pages 345–360.

[Feinstein and Horwitz, 1997] Feinstein, A. and Horwitz, R. (1997). Problems in the evidence of evidence-based medicine. *American Journal of Medicine*, 103:529–535.

[Fensel, 1997a] Fensel, D. (1997a). An ontology-based broker: Making problem-solving method reuse work. In *Proceedings Workshop on Problem-solving Methods for Knowledge-based Systems at IJCAI'97.*

[Fensel, 1997b] Fensel, D. (1997b). The tower-of-adapter method for developing and reusing problem-solving methods. In *Knowledge Acquisition, Modeling and Management*, pages 97–112.

[Fensel et al., 1998a] Fensel, D., Angele, J., and Studer, R. (1998a). The knowledge acquisition and representation language karl. *IEEE Transcactions on Knowledge and Data Engineering*, 10(4):527–550.

[Fensel and Benjamins, 1998a] Fensel, D. and Benjamins, R. (1998a). The role of assumptions in knowledge engineering. *International Journal of Intelligent Systems*.

[Fensel and Benjamins, 1998b] Fensel, D. and Benjamins, V. (1998b). Key issues for automated problem-solving methods reuse. In *Proceedings 13th European Conference on Artificial Intelligence*.

[Fensel et al., 1999] Fensel, D., Benjamins, V., Motta, E., and Wielinga, B. (1999). UPML: A framework for knowledge system reuse. In *International Joint Conference on AI*, pages 16–23.

[Fensel and Bussler, 2002] Fensel, D. and Bussler, C. (2002). The web service modelling framework. Technical report, Vrijer Universiteit Amsterdam.

[Fensel et al., 1997] Fensel, D., Decker, S., Motta, E., and Zdrahal, Z. (1997). Using ontologies for defining task, problem-solving methods and their mappings. In *Proceedings European Knowledge Acquisition Workshop*, Lecture Notes in Artificial Intelligence.

[Fensel et al., 1998b] Fensel, D., Groenboom, R., and de Lavalette, G. (1998b). Modal change logic (mcl): Specifying the reasoning of knowledge-based systems. *Data and Knowledge Engineering*, 26(3):243–269.

[Fensel et al., 2000] Fensel, D., Horrocks, I., van Harmelen, F., Decker, S., Erdmann, M., and Klein, M. C. A. (2000). OIL in a nutshell. In *Proceedings of the European Knowledge Acquisition, Modeling and Management Conference (EKAW)*, Lecture Notes in Artificial Intelligence, pages 1–16. Springer-Verlag.

[Fensel and Motta, 2001] Fensel, D. and Motta, E. (2001). Structured development of problem solving methods. *Knowledge and Data Engineering*, 13(6):913–932.

[Fensel and Straatman, 1996] Fensel, D. and Straatman, R. (1996). Problem-solving methods: Making assumptions for efficiency reasons. *Lecture Notes in Computer Science*, 1076:17–??

[Finin et al., 1994] Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an Agent Communication Language. In Adam, N., Bhargava, B., and Yesha, Y., editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463, Gaithersburg, MD, USA. ACM Press.

[Fink, 1998] Fink, E. (1998). How to solve it automatically: Selection among problem-solving methods. In *Proceedings of AIPS*, pages 128–136.

[FIPA, 2002] FIPA (2002). FIPA contract net interaction protocol specification.

[FIPA, 2003] FIPA (2003). Agent Communication Language Specifications. http://www.fipa.org/repository/aclspecs.html.

[Fischer et al., 1995] Fischer, B., Kievernagel, M., and Struckmann, W. (1995). VCR: A VDM-based software component retrieval tool. In *Proc. ICSE-17 Workshop on Formal Methods Application in Software Engineering Practice.*

[Fisher et al., 1997] Fisher, M., Muller, J., Schroeder, M., Staniford, G., and Wagner, G. (1997). Methodological foundations for agentbased systems. *Knowledge Engineering Review*, 12(3):323–329.

[Franklin,] Franklin, S. Coordination without communication. http://www.msci.memphis.edu/ franklin/coord.html.

[Franklin and Graesser, 1996] Franklin, S. and Graesser, A. (1996). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Intelligent Agents III. Agent Theories, Architectures and Languages (ATAL'96)*, volume 1193, Berlin, Germany. Springer-Verlag.

[Garland and Perry, 1995] Garland, D. and Perry, D. (1995). Special issue on software architectures. *IEEE Transactions on Software Engineering.*

[Garland et al., 1993] Garland, S. J., Guttag, J. V., and Horning, J. J. (1993). An overview of Larch. In *Functional Programming, Concurrency, Simulation and Automated Reasoning*, volume 693 of *Lecture Notes in Computer Science*, pages 329–348. Springer-Verlag.

[Gaspari et al., 1998] Gaspari, M., Motta, E., and Fensel, D. (1998). Exploiting automated theorem proving in UPML: Automatic configuration of PSM from PSMs libraries. Technical report, Robotics Institute, Carnegie Mellon University.

[Gaspari et al., 1999] Gaspari, M., Motta, E., and Fensel, D. (1999). Automatic selection of problem solving libraries based on competence matching. In *ECOOP Workshops*, pages 10–11.

[Gasser and Briot, 1992] Gasser, L. and Briot, J.-P. (1992). Object-based concurrent programming and distributed artificial intelligence. In Avouris, N. M. and Gasser, L., editors, *Distributed Artificial Intelligence: Theory and Praxis*, pages 81–107. Kluwer, Dordrecht.

[Genesereth et al., 1997] Genesereth, M., Keller, A., and Duschka, O. (1997). Infomaster: an information integration system. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 539–542.

[Genesereth and Ketchpel, 1997] Genesereth, M. R. and Ketchpel, S. P. (1997). Software agents. *Communications of the ACM*, 37(7).

[Gennari and Tu, 1994] Gennari, J. and Tu, S. (1994). Mapping domains to methods in support of reuse. *International Journal on Human Computer Studies*, 41:399–42.

[Georgeff, 1983] Georgeff, M. (1983). Communication and interaction in multi-agent planning. In *Proceedings of the Third National Conference on Artificial Intelligence*.

[Georgeff and Lansky, 1987] Georgeff, M. and Lansky, A. (1987). Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, volume 2, pages 677–682.

[Giampapa and Sycara, 2001] Giampapa, J. A. and Sycara, K. (2001). Conversational case-based planning for agent team coordination. *Lecture Notes in Computer Science*, 2080.

[Giampapa and Sycara, 2002] Giampapa, J. A. and Sycara, K. (2002). Team-oriented agent coordination in the retsina multi-agent system. Technical Report CMU-RI-TR-02-34, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. Presented at AAMAS 2002 Workshop on Teamwork and Coalition Formation.

[Giunchiglia and Shvaiko, 2003] Giunchiglia, F. and Shvaiko, P. (2003). Semantic matching. Technical Report DIT-03-013, Informatica e Telecomunicazioni, University of Trento.

[Glaser, 1996] Glaser, N. (1996). *Contribution to Knowledge Modelling in a Multi-Agent Framework*. PhD thesis, L'Université Henri Poincaré, Nancy I, France.

[Goguen et al., 1996] Goguen, J., Nguyen, D., Meseguer, J., Luqi, Zhang, D., and Berzins, V. (1996). Software component search. journal of systems integration. *Journal of Systems Integration,*, 6:93–134.

[Gómez and Benjamins, 1999] Gómez, A. and Benjamins, V. (1999). Overview of knowledge sharing and reuse of components: Ontologies and problem-solving methods. In *Proceedingsof Workshop on Ontologies and Problem-Solving Methods of IJCAI*.

[Gómez and Abasolo, 2002] Gómez, M. and Abasolo, C. (2002). Improving meta-search by using query-weighting and numerical aggregation operators. In *Proceedings 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*.

[Gómez et al., 2001] Gómez, M., Abasolo, C., and Plaza, E. (2001). Domain-independent ontologies for cooperative information agents. In *Proceedings* *of the Fifth International Workshop on Cooperative Information Agents*, volume 2128 of *Lecture Notes in Artificial Intelligence*, pages 118–129. Springer-Verlag.

[Gómez et al., 2002] Gómez, M., Abasolo, C., and Plaza, E. (2002). Problem-solving methods and cooperative information agents. *International Journal on Cooperative Information Systems*, 11(3-4):329–354.

[Gómez and Abasolo, 2003] Gómez, M. and Abasolo, J. M. (2003). A general framework for meta-search based on query weighting and numerical aggregation operators. In *Intelligent Systems for Information Processing: From Representation to Applications*, pages 129–140. Elsevier Science.

[Gómez et al., 2003a] Gómez, M., Abasolo, J. M., and Plaza, E. (2003a). Description and configuration of multi-agent systems at the knowledge level. In Aguiló, I., Valverde, L., and Escrig, M., editors, *Artificial Intelligence Research and Development*, volume 100 of *Frontiers in Artifical Intelligence and Applications*, pages 221–232. IOS Press.

[Gómez et al., 2003b] Gómez, M., Abasolo, J. M., and Plaza, E. (2003b). Orcas: Open, reusable and configurable multi-agent systems. Third price in the Infrastructures Category.

[Gómez and Plaza, 2004a] Gómez, M. and Plaza, E. (2004a). Extending matchmaking to maximize capability reuse (to appear). In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems.*

[Gómez and Plaza, 2004b] Gómez, M. and Plaza, E. (2004b). A knowledge based framework for configuring agent teams on-demand. In *Proceedings of the Workshop on Semantic Intelligent Middleware for Interoperable Systems.*

[Grosz et al., 1999] Grosz, B. J., Hunsberger, L., and Kraus, S. (1999). Planning and acting together. *The AI Magazine*, 20(4):23–34.

[Grosz and Kraus, 1993] Grosz, B. J. and Kraus, S. (1993). Collaborative plans for group activities. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 367–373.

[Grosz and Kraus, 1996] Grosz, B. J. and Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357.

[Grosz and Sidner, 1990] Grosz, B. J. and Sidner, C. L. (1990). Plans for discourse. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, pages 417–444. MIT Press, Cambridge, MA.

[Gruber, 1993a] Gruber, T. R. (1993a). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands. Kluwer Academic Publishers.

[Gruber, 1993b] Gruber, T. R. (1993b). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

[Guarino, 1997a] Guarino, N. (1997a). Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In Pazienza, M., editor, *Summer School on Information Extraction*, pages 139–170. Springer Verlag.

[Guarino, 1997b] Guarino, N. (1997b). Understanding, building, and using ontologies: A commentary to using explicit ontologies in kbs development, by van heijst, schreiber, and wielinga. *International Journal of Human and Computer Studies*, 46:293–310.

[Hall, 1993] Hall, R. J. (1993). Generalized behavior-based retrieval. In *Proceedings of 15th International Conference on Software Engineering*, pages 371–380.

[Han, 1999] Han, J. (1999). Semantic and usage packaging for software components. In *Workshops of 13th European Conference on Object-Oriented Programming*, pages 7–8.

[Haustein and Ludecke, 2000] Haustein, S. and Ludecke, S. (2000). Towards information agent interoperability. In Klusch, M. and Kerschberg, L., editors, *Proceedings Cooperative Information Agents*, Lecture Notes in Computer Science, pages 208–219. Springer.

[Herlea et al., 1999] Herlea, D. E., Jonker, C. M., Treur, J., and Wijngaards, N. J. E. (1999). Specification of behavioural requirements within compositional multi-agent system design. In Garijo, F. J. and Boman, M., editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, volume 1647, pages 8–27. Springer-Verlag: Heidelberg, Germany.

[Horrocks, 2002] Horrocks, I. (2002). DAML+OIL: A reasonable web ontology language. In *Proceedings the 7th Conference on Extending Database Technology*, volume 2287 of *Lecture Notes in Computer Science*, pages 2–13. Springer-Verlag.

[Huguet et al., 2002] Huguet, M.-P., Esteva, M., Parsons, S., Sierra, C., and Wooldridge, M. (2002). Model checking electronic institutions. In *Proceedings of the ECAI Workshop on Model Checking Artificial Intelligence.*

[Iglesias et al., 1998] Iglesias, C., Garrijo, M., and Gonzalez, J. (1998). A survey of agent-oriented methodologies. In *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages*, volume 1555 of *Lecture Notes in Artificial Intelligence*, pages 317–330.

[Iglesias et al., 1997] Iglesias, C. A., Garijo, M., Centeno-Gonzalez, J., and Velasco, J. R. (1997). Analysis and design of multiagent systems using MAS-common KADS. In *Agent Theories, Architectures, and Languages*, pages 313–327.

[IIIA, 2003] IIIA (2003). Dialogical institutions. http://e-institutor.iiia.csic.es/.

[International Foundation on Cooperative Information Systems, 1994] International Foundation on Cooperative Information Systems (1994). Second international conference on cooperative information systems.

[Iribarne et al., 2002] Iribarne, L., Troya, J., and Vallecillo, A. (2002). Selecting software components with multiple interfaces. In *Proceeding of the 28th EUROMICRO Conference – Component-Based Software Engineering*, pages 26–32.

[Jennings, 1993] Jennings, N. R. (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250.

[Jennings, 2000] Jennings, N. R. (2000). On-agent-based software engineering. *Artificial Intelligence*, 117:227–296.

[Jennings and Campos, 1997] Jennings, N. R. and Campos, J. (1997). Towards a social level characterisation of socially responsible agents. *IEEE Proc. Software Engineering*, 144(1):11–25.

[Jennings et al., 1992] Jennings, N. R., Mamdani, E., Laresgoiti, I., Perez, J., and Corera, J. (1992). GRATE: a general framework for cooperative problem solving. *IEEE-BCS Journal of Intelligent Systems Engineering*, 1(2):102–104.

[Jennings et al., 1998] Jennings, N. R., Sycara, K., and Woolridge, M. (1998). A roadmap of agent-research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):227–296.

[Jeusfeld and Papazoglou, 1996] Jeusfeld, M. A. and Papazoglou, M. (1996). Information brokering: Design, search and transformation. *Aachener Informatik-Bericht*, 96(18).

[John H. Gennari and Musen, 1998] John H. Gennari, W. G. and Musen, M. (1998). A method-description language: an initial ontology with examples. In *Proceedings 11th Workshop on Knowledge Acquisition, Modelling and Management*.

[Jonathan Lee and Chiang, 2002] Jonathan Lee, K. F. L. and Chiang, W. (2002). Possibilistic petri nets as a basis for agent service description language. Submitted to Fuzzy Sets and Systems.

[Juhasz and Paul, 2002] Juhasz, Z. and Paul, P. (2002). Scalability analysis of the contract net protocol. In *Proceedings of the 2nd International Workshop on Agent based Cluster and Grid Computing at IEEE International Symposium on Cluster Computing and the Grid*, Berlin, Germany.

[Kendall et al., 1995] Kendall, E. A., Malkoun, M. T., and Jiang, C. H. (1995). A methodology for developing agent based systems for enterprise integration. In *First Australian Workshop on Distributed Artificial Intelligence*.

[Kiepuszewski, 2002] Kiepuszewski, B. (2002). *Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows.* PhD thesis, Queensland University of Technology, Brisbane, Australia.

[Kinny and Georgeff, 1996] Kinny, D. and Georgeff, M. (1996). Modelling and design of multi-agent systems. In *Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Lecture Notes in Artificial Intelligence.

[Kinny et al., 1992] Kinny, D., Ljungberg, M., Rao, A. S., Sonenberg, E., Tidhar, G., and Werner, E. (1992). Planned team activity. In Castelfranchi, C. and Werner, E., editors, *Artificial Social Systems. Fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, volume 830 of *Lecture Notes in Artificial Intelligence*, pages 226–256. Springer-Verlag: Heidelberg, Germany.

[Kirk et al., 1995] Kirk, T., Levy, A. Y., Sagiv, Y., and Srivastava, D. (1995). The Information Manifold. In Knoblock, C. and Levy, A., editors, *Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, Stanford, California.

[Klein, 2000] Klein, M. (2000). The Challenge: Enabling Robust Open Multi-Agent Systems.

[Klinker et al., 1991] Klinker, G., Bhola, C., Dallemagne, G., Marques, D., and McDermott, J. (1991). Usable and reusable programming constructs. *Knowledge Acquisition*, 3:117–135.

[Knoblock et al., 1994] Knoblock, C. A., Arens, Y., and Hsu, C.-N. (1994). Cooperating agents for information retrieval. In *Proceedings of the 2nd International Conference on Cooperative Information Systems*, Toronto, Ontario, Canada. University of Toronto Press.

[Labrou and Finin, 1997] Labrou, Y. and Finin, T. (1997). Semantics and conversations for an agent communication language. In Pollack, M. E., editor, *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence*, pages 584–591, Nagoya, Japan. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[Larson and Sandholm, 2000] Larson, K. and Sandholm, T. (2000). Anytime coalition structure generation: An average case study. *Journal of Experimental and Theoretical AI*, 11:1–20.

[Lemahieu, 2001] Lemahieu, W. (2001). Web service description, advertising and discovery: Wsdl and beyond. In Vandenbulcke, J. and Snoeck, M., editors, *New Directions In Software Engineering.* Leuven University Press.

[Lesser, 1991] Lesser, V. R. (1991). A retrospective view of distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics, Special Issue on Distributed Artificial Intelligence*, 1(1):63–83.

[Lesser et al., 1989] Lesser, V. R., Durfee, E., and Corkill, D. (1989). Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):63–83.

[Levesque, 1990] Levesque, H. J. (1990). On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 94–99.

[Levy et al., 1996] Levy, A. Y., Rajaraman, A., and Ordille, J. J. (1996). Query-answering algorithms for information agents. In *Proceedings of the AAAI*, pages 40–47.

[Liskov and Wing, 1993] Liskov, B. and Wing, J. M. (1993). A new definition of the subtype relation. In Nierstrasz, O. M., editor, *Proceedings 7th European Conference on Object-Oriented Programming*, volume 707 of *Lecture Notes in Computer Science*, pages 118–141. Springer-Verlag.

[Luck et al., 1997] Luck, M., Griffiths, N., and d'Inverno, M. (1997). From agent theory to agent construction: A case study. In Müller, J. P., Wooldridge, M. J., and Jennings, N. R., editors, *Proceedings of the ECAI'96 Workshop on Agent Theories, Architectures, and Languages: Intelligent Agents III*, volume 1193, pages 49–64. Springer-Verlag: Heidelberg, Germany.

[Luck et al., 2003] Luck, M., McBurney, P., and Preist, C. (2003). *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. AgentLink.

[Maarek et al., 1991] Maarek, Y., Berry, D., and Kaiser, G. (1991). An information retrieval approach for automatically constructing software libraries. *IEEE Transactions on Software Engineering*, 17(8):800–813.

[Martin et al., 1999] Martin, D. L., Cheyer, A. J., and Moran, D. B. (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1-2):91–128. OAA.

[Martin et al., 1997] Martin, D. L., Oohama, H., Moran, D., and Cheyer, A. (1997). Information brokering in an agent architecture. In *Proceedings of the 2nd International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*.

[McDermott, 1988] McDermott, J. (1988). Toward a taxonomy of problem-solving methods. In Marcus, S., editor, *Automating Knowledge Acquisition for Expert Systems*, pages 225–256. Kluwer Academic.

[McIlraith and Son, 2001] McIlraith, S. and Son, T. C. (2001). Adapting golog for programming the semantic web. In *Working Notes of the 5th International Symposium on Logical Formalizations of Commonsense Reasoning*.

[McIlraith et al., 2001] McIlraith, S. A., Son, T. C., and Zeng, H. (2001). Semantic web services. *IEEE Intelligent Systems (Special Issue on the Semantic Web)*, 16(2):46–53.

[Mikhajlova, 1999] Mikhajlova, A. (1999). *Ensuring Correctness of Object and Component Systems*. PhD thesis, Abo Akademi University.

[Mili et al., 1997] Mili, A., Mili, R., and Mittermeir, R. (1997). Storing and retrieving software components: A refinement based system. *IEEE Transactions on Software Engineering*, 23(7):445–460.

[Mili et al., 1995] Mili, H., Mili, F., and Mili, A. (1995). Reusing software: Issues and research directions. *Software Engineering*, 21(6):528–562.

[Monica Crubezy and Musen, 2001] Monica Crubezy, W. L. and Musen, M. A. (2001). The internet reasoning service: Delivering configurable problem-solving components to web users.

[Motta, 1999] Motta, E. (1999). *Reusable Components for Knowledge Modelling*, volume 53 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

[Motta et al., 1999] Motta, E., Fensel, D., Gaspari, M., and Benjamins, A. (1999). Specifications of knowledge components for reuse. In *Proceedings of SEKE '99, 1999.*

[Moulin and Brassard, 1996] Moulin, B. and Brassard, M. (1996). A scenario-based design method and an environment for the development of multiagent systems. In *Proceedings of the 1th Australian Workshop on Distributed Artificial Intelligence*, volume 1087 of *Lecture Notes in Artificial Intelligence*, pages 216–231.

[Müller, 1996] Müller, H. J. (1996). Towards agent systems engineering. *International Journal on Data and Knowledge Engineering. Special Issue on Distributed Expertise*, 23:217–245.

[Munoz-Avila et al., 1999] Munoz-Avila, H., Aha, D. W., Breslow, L., and Nau, D. S. (1999). HICAP: An interactive case-based planning architecture and its application to noncombatant evacuation operations. In *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence*, pages 879–885. AAAI Press.

[Musen, 1998] Musen, M. (1998). Modern architectures for intelligent systems: Reusable ontologies and problem-solving methods. In *Proceedings of AMIA Fall Symposium.*

[Narayan and McIlraith, 2002] Narayan, S. and McIlraith, S. (2002). Simulation, verification and automated composition of web services. In *Proceedings of the 11th International World Wide Web Conference*, pages 77–88. ACM Press.

[Narendra, 2003] Narendra, N. (2003). An agent-oriented architectural framework for enacting and managing web services. Agenticies Working Group on Service Description and Composition.

[Newell, 1982] Newell, A. (1982). The knowledge level. *Artificial Intelligence*, 28(2):87–127.

[Nii, 1989] Nii, H. (1989). Blackboard systems. In Bar, A., P.R.Cohen, and Feigenbaum, E., editors, *The Handbook of Artificial Intelligence IV*, pages 1–82. Addison-Wesley.

[Nodine et al., 1999] Nodine, M., Bohrer, W., and Ngu, A. (1999). Semantic brokering over dynamic heterogeneous data sources in infosleuth. In *ICDE*, pages 358–365.

[Nodine and Unruh, 1999] Nodine, M. and Unruh, A. (1999). Constructing robust conversation policies in dynamic agent communities. In *Proceedings of the AGENTS'99 Workshop on Specifying and Implementing Conversation Policies*.

[Noriega, 1997] Noriega, P. (1997). *Agent-Mediated Auctions: The Fish-Market Metaphor*. PhD thesis, Universitat Autnoma de Barcelona.

[Norman, 1994] Norman, T. (1994). Motivated goal and action selection. In *Working Notes of the AISB worskhop, Models or Behaviours, which way forward for robotics?*

[Nwana and Woolridge, 1996] Nwana, H. S. and Woolridge, M. (1996). Software agents: An overview. *Knowledge Engineering Review*, 11(3):205–244.

[O'Hare and Woolridge, 1992] O'Hare, G. and Woolridge, M. (1992). A software engineering perspective on multi-agent system design. In Avouris, N. M. and Gasser, L., editors, *Distributed Artificial Intelligence: Theory and Practice*, pages 109–127. Kluwer Academic Publishers.

[Orsvarn, 1996] Orsvarn, K. (1996). Principles for libraries of task decomposition methods – conclusions from a case-study. In *Proceedings of 9th European Knowledge Acquisition Workshop*, volume 1076 of *Lecture Notes in Artificial Intelligence*, pages 48–65. Springer-Verlag.

[Ousterhout, 1990] Ousterhout, J. K. (1990). Tcl: An embeddable command language. In *Proceedings of the USENIX Winter 1990 Technical Conference*, pages 133–146, Berkeley, CA. USENIX Association.

[Panzarasa and Jennings, 2001] Panzarasa, P. and Jennings, N. R. (2001). The organisation of sociality: A manifesto for a new science of multiagent systems. In *In Proceedings of the Tenth European Workshop on Multi-Agent Systems*.

[Paolucci et al., 2002] Paolucci, M., Kawmura, T., Payne, T., and Sycara, K. (2002). Semantic matching of web services capabilities. In *Proceedings of the 1st International Semantic Web Conference*.

[Park et al., 1998] Park, J. Y., Gennari, J. H., and Musen, M. A. (1998). Mappings for reuse in knowledge-based systems. In *Proceedings 11th Workshop on Knowledge Acquisition, Modelling and Management*.

[Payne et al., 2001] Payne, T. R., Paolucci, M., and Sycara, K. (2001). Advertising and matching daml-s service descriptions. In *Semantic Web Working Symposium (SWWS)*.

[Pednault, 1989] Pednault, E. P. (1989). ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 324–332. Morgan Kaufmann.

[Peltz, 2003] Peltz, C. (2003). Web services orchestration. a review of emerging technologies, tools and standards. Technical report, Hewlett Packard, Co.

[Penix, 1998] Penix, J. (1998). *Automated Component Retrieval and Adaptation Using Formal Specifications*. PhD thesis, University of Cincinnati.

[Penix and Alexander, 1997] Penix, J. and Alexander, P. (1997). Component reuse and adaptation at the specification level. In *Proceedings of the 8th Annual Workshop on Institutionalizing Software Reuse*.

[Penix and Alexander, 1999] Penix, J. and Alexander, P. (1999). Efficient specification-based component retrieval. *Automated Software Engineering: An International Journal*, 6(2):139–170.

[Penix et al., 1995] Penix, J., Baraona, P., and Alexander, P. (1995). Classification and retrieval of reusable components using semantic features. In *Proceedings of the 10th Knowledge-Based Software Engineering Conference*, pages 131–138.

[Piresa et al., 2003] Piresa, P. F., Benevides, M. R. F., and Mattoso:, M. (2003). Building reliable web services compositions. In *LNCS 2593*, volume 2593 of *Lecture Notes in Computer Science*, pages 59–72. Springer-Verlag.

[Plaza and Arcos, 2002] Plaza, E. and Arcos, J. L. (2002). Constructive adaptation. In Craw, S. and Preece, A., editors, *Advances in Case-Based Reasoning. Proceedings 6th ECCBR*, volume 2416 of *Lecture Notes in Artificial Intelligence*, pages 306–320.

[Poek and Gappa, 1993] Poek, K. and Gappa, U. (1993). Making role-limiting shells more flexible. In Aussenac, N. et al., editors, *Knowledge Acquisition for Knowledge-Based Systems. Proceedings of EKAW'93*.

[Prieto-Daz, 1987] Prieto-Daz, R. (1987). Classifying software for reusability. *IEEE Software*, 4(1).

[Puerta et al., 1992] Puerta, A., Egar, J., Tu, S. W., and Musen, M. A. (1992). A multiple-method knowledge acquisition shell for the automatic generation of knowledge acquisition tools. *Knowledge Acquisition*, 4:171–196.

[Pynadath et al., 1999] Pynadath, D. V., Tambe, M., Chauvat, N., and Cavedon, L. (1999). Toward team-oriented programming. In *Agent Theories, Architectures, and Languages*, pages 233–247.

[Rao, 1994] Rao, A. S. (1994). Means-end plan recognition : Towards a theory of reactive recognition. In Torasso, P. and Jon Doyle, a. E. S., editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 497–508. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[Rao and Georgeff, 1991] Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. In Allen, J., Fikes, R., and Sandewall, E., editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[Rao and Georgeff, 1995] Rao, A. S. and Georgeff, M. P. (1995). BDI-agents: from theory to practice. In *Proceedings of the First Intl. Conference on Multiagent Systems*, San Francisco.

[Rao et al., 1992] Rao, A. S., Georgeff, M. P., and Sonenberg, E. A. (1992). Social plans: A preliminary report. In Werner, E. and Demazeau, Y., editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, pages 57–76, Kaiserslautern, Germany. Elsevier Science B.V.: Amsterdam, Netherland.

[Rising, 2000] Rising, L. (2000). *The Pattern Almanac 2000*. Software PAtterns Series. Addison Wesley.

[Rodríguez-Aguilar, 1997] Rodríguez-Aguilar, J. A. (1997). *On the Design and Construction of Agent-mediated Electronic Institutions*. PhD thesis, Universitat Autnoma de Barcelona.

[Rollins and Wing, 1991] Rollins, E. J. and Wing, J. M. (1991). Specifications as search keys for software libraries. In Furukawa, K., editor, *Proceedings of the Eighth International Conference on Logic Programming*, pages 173–187, Paris, France. The MIT Press.

[Rosario and Ibrahim, 1994] Rosario, M. and Ibrahim, G. (1994). A similarity measure for retrieving software artifacts. In *Proceedings of the Sixth International Conference on Software Engineering and Knowledge Engineering*, pages 478–485.

[Sandholm et al., 1998] Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohme, F. (1998). Anytime coalition structure generation with worst case guarantees. In *Proceedings AAAI Innovative Applications of Artificial Intelligence*, pages 46–53.

[Sandholm, 1993] Sandholm, T. W. (1993). An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, pages 295–308, Hidden Valley, Pennsylvania.

[Schreiber et al., 1994a] Schreiber, A., Wielinga, B. J., Ackermans, J., Van de Velde, W., and Hoog, R. D. (1994a). CommonKADS: A comprehensive methodology for kbs development. *IEEE Expert*, 9(6):28–37.

[Schreiber et al., 1994b] Schreiber, G., Wielinga, B., Akkermans, H., Van de Velde, W., and Anjewierden, A. (1994b). CML: The CommonKADS conceptual modelling language. In Steels, L., Schreiber, G., and Van de Velde, W., editors, *A Future for Knowledge Acquisition: Proceedings of EKAW'94*, pages 1–25, Berlin, Heidelberg. Springer.

[Schreiber et al., 1993] Schreiber, G., Wielinga, B. J., and Breuker, J. A. (1993). *KADS: a Principled Approach to Knowledge-Based System Development*, volume 11 of *Knowledge-Based Systems*. Academic Press.

[Searle, 1969] Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. PhD thesis, Cambridge University Press.

[Shadbolt et al., 1993] Shadbolt, N., Motta, E., and Rouge, A. (1993). Constructing knowledge-based systems. *IEEE Software*, 10(6):34–39.

[Shaw and Garlan, 1996] Shaw, M. and Garlan, D. (1996). *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall.

[Shehory and Kraus, 1998] Shehory, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200.

[Shehory et al., 1997] Shehory, O., Sycara, S. K., and Jha, S. (1997). Multi-agent coordination through coalition formation. In *Intelligent Agents IV: Agent Theories, Architectures and Languages*, volume 1365, pages 143–154. Springer.

[Singh, 1994] Singh, M. P. (1994). *Multiagent Systems: A theoretical framework for intentions, know-how and communication*, volume 799 of *Lecture Notes in Artificial Intelligence*. Springer Verlag.

[Singh, 1998] Singh, M. P. (1998). The intentions of teams: Team structure, endodeixis, and exodeixis.

[Singh et al., 1993] Singh, M. P., Huhns, M. N., and M.Stephens, L. (1993). Declarative representations of multi-agent systems. *IEEE Transactions on Knowledge and Data Engineering*, 5(5):721–739.

[Smith, 1994] Smith, G. (1994). Formal definitions of behavioural compatibility for active and passive objects. In *Proceedings 1st Asia-Pacific Software Engineering Conference*, pages 336–344. IEEE Computer Society Press.

[Smith, 1940] Smith, R. G. (1940). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113.

[Sonenberg et al., 1994] Sonenberg, E., Tidhar, G., Werner, E., Kinny, D., Ljungberg, M., and Rao, A. (1994). Planned team activity. In *Articial Social Systems*, Lecture Notes in Articial Intelligence, pages 227–256.

[Splunter et al., 2003] Splunter, S. V., Wijngaards, N. J., and Brazier, F. M. (2003). Structuring agents for adaptation. In Alonso, E., D., K., and Kazakov, D., editors, *Adaptive Agents and Multi-Agent Systems*, Lecture Notes in Artificial Intelligence (LNAI) 2636. Springer-Verlag Berlin.

[Stab et al., 2003] Stab, S. et al. (2003). Web service: Been there, done what. *IEEE Intelligent Systems.*, 18(1).

[Stader and Macintosh, 1999] Stader, J. and Macintosh, A. (1999). Capability modelling and knowledge management. In *Applications and Innovations in Expert Systems VII*, pages 33–50. Springer Verlag.

[Steels, 1988] Steels, L. (1988). The deepening of expert systems. *AI Communications*, 1(1):9–17.

[Steels, 1990] Steels, L. (1990). Components of expertise. *AI Magazine*, 11(2):28–49.

[Steels, 1993] Steels, L. (1993). The componential framework and its role in reusability. In David, J., Krivine, J., and Simmons, R., editors, *Second generation expert systems*, pages 273–298. Springer Verlag.

[Studer et al., 1998] Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25(1-2):161–197.

[Studer et al., 1996] Studer, R., Eriksson, H., Gennari, J., Tu, S., Fensel, D., and Musen, M. (1996). Ontologies and the configuration of problem-solving methods. In *Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop.*

[Sycara et al., 1996] Sycara, K. P., Decker, K., Pannu, A., Williamson, M., and Zeng, D. (1996). Distributed intelligent agents. *IEEE Expert*, 11(6):36–46.

[Sycara et al., 1999a] Sycara, K. P., Klusch, M., Widoff, S., and Lu, J. (1999a). Dynamic service matchmaking among agents in open information environments. *SIGMOD*, 28(1):47–53.

[Sycara et al., 1999b] Sycara, K. P., Lu, J., Klusch, M., and Widoff, S. (1999b). Matchmaking among heterogeneous agents on the internet. In *Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace.*

[Sycara et al., 2001] Sycara, K. P., Paolucci, M., Velsen, M. V., and Giampapa, J. A. (2001). The RETSINA MAS infrastructure. Technical report, Robotics Institute, Carnegie Mellon University.

[Sycara et al., 2002] Sycara, K. P., Widoff, S., Klusch, M., and Lu, J. (2002). Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5:173–203.

[Szyperski, 1996] Szyperski, C. (1996). Independently extensible systems – software engineering potential and challenges. In *Proceedings of the 19th Australian Computer Science Conference*, Melbourne, Australia.

[Tambe, 1997] Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124.

[Tambe et al., 1999] Tambe, M., Adibi, J., Alonaizon, Y., Erdem, A., Kaminka, G. A., Marsella, S., and Muslea, I. (1999). Building agent teams using an explicit teamwork model and learning. *Artificial Intelligence*, 110(2):215–239.

[Tambe et al., 2000] Tambe, M., Pynadath, D. V., Chauvat, N., Das, A., and Kaminka, G. A. (2000). Adaptive agent integration architectures for heterogeneous team members. In *Proceedings of the International Conference on Multiagent Systems*, pages 301–308, Boston, MA.

[Tate, 1998] Tate, A. (1998). Roots of SPAR- shared planning and activity representation. *The Knowledge Engineering Review, Special Issue on Putting Ontologies to Use*, 13(1):121–128.

[Terpstra et al., 1993] Terpstra, P., van Heijst, G., Wielinga, B., and Shadtbolt, N. (1993). Knowledge acquisition support through generalised directive models. In David, J., Krivine, J., and Simmons, R., editors, *Second Generation Expert Systems*, pages 428–455. Springer-Verlag.

[The DAML-S Consortium, 2001] The DAML-S Consortium (2001). Daml-s: Semantic markup for web services. In *Proceedings of the International Semantic Web Workshop.*

[Tidhar et al., 1996] Tidhar, G., Rao, A., and Sonenberg, E. (1996). Guided team selection. In *In Proceedings of the 2nd International Conference on Multi-agent Systems (ICMAS-96).*

[Tidwell, 2000] Tidwell, D. (2000). Web services. the web's next revolution. http://www-106.ibm.com/developerworks/webservices.

[Torra, 1996] Torra, V. (1996). Weighted owa operators for synthesis of information. In *Proceedings 5th IEEE Inter. Conference on Fuzzy Systems*, pages 966–971.

[Tu et al., 1995] Tu, S. W., Eriksson, H., Gennari, J. H., Shahar, Y., and Musen, M. A. (1995). Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTEGE-II to protocol-based decision support. *Artificial Intelligence in Medicine*, 7(3):257–289.

[Valente and Lockenhoff, 1993] Valente, A. and Lockenhoff, C. (1993). Organization as guidance: A library of assessment models. In *Proceedings of the Seventh European Knowledge Acquisition Workshop*, volume 723 of *Lecture Notes in Artificial Intelligence*.

[Valente et al., 1994] Valente, A., Van de Velde, W., and Breuker, J. (1994). The commonkads expertise modelling library. In Breuker, J. and Van de Velde, W., editors, *CommonKADS Library for Expertise Modeling*, volume 21 of *Frontiers in Artificial Intelligence and Applications*, pages 31–56. IOS-Press.

[Van de Velde, 1993] Van de Velde, W. (1993). Issues in knowledge level modelling. In David, J., Krivine, J., and Simmons, R., editors, *Second generation expert systems*, pages 211–231. Springer Verlag.

[van der Aalst and ter Hofstede, 2002] van der Aalst, W. and ter Hofstede, A. (2002). Yawl: Yet another workflow language. QUT Technical report FIT-TR-2002-06, Queensland University of Technology.

[van der Aalst et al., 2001] van der Aalst, W., Verbeek, H., and Kumar, A. (2001). Rl/woflan: Verification of an xml/petri-net based language for inter-organizational workflows. In Altinkemer, K. and Chari, K., editors, *Proceedings of the 6th Informs Conference on Information Systems and Technology*, pages 30–45.

[van Heijst, 1995] van Heijst, G. (1995). *The Role of Ontologies in Knowledge Engineering*. PhD thesis, University of Amsterdam.

[Vasconcelos et al., 2001] Vasconcelos, W. W., Sabater, J., Sierra, C., and Querol, J. (2001). Skeleton-based agent development for electronic institutions. In *Proceedings UKMAS*.

[Vaughan-Nichols, 2002] Vaughan-Nichols, S. J. (2002). Web services: Beyond the hype. *Computer*, pages 18–21.

[Vidal and Durfee, 1995] Vidal, J. M. and Durfee, E. H. (1995). Task planning agents in the UMDL. In *Proceedings of the Fourth International Conference on Information and Knowledge Management (CIKM) Workshop on Intelligent Information Agents*.

[Vidal et al., 1998] Vidal, J. M., Mullen, T., Weinstein, P., and Durfee, E. H. (1998). The UMDL service market society. In *Proceedings of the Second International Conference on Autonomous Agents*.

[Walther et al., 1992] Walther, E., Eriksson, H., and Musen, M. (1992). Plug and play: Construction of task specific expert-system shells using sharable context ontologies. Technical Report KSl-92-40, Knowledge Systems Laboratory, Stanford University.

[Walton and Krabbe, 1995] Walton, D. N. and Krabbe, E. C. (1995). *Commitment in Dialogue*. SUNNY Press.

[Wegner, 1984] Wegner, P. (1984). Capital-intensive software technology. *IEEE Software*, 1(3):7–45.

[White and Sleeman, 1999] White, S. and Sleeman, D. H. (1999). A constraint-based approach to the description of competence. In *Proceedings of Knowledge Acquisition, Modeling and Management*, volume 1621 of *Lecture Notes in Artificial Intelligence*, pages 291–308. Springer Verlag.

[Wickler and Tate, 1999] Wickler, G. and Tate, A. (1999). Capability representations for brokering: A survey. submitted to the knowledge engineering review.

[Wiederhold, 1992] Wiederhold, G. (1992). Mediators in the architecture of future information systems. *Computer Magazine of the Computer Group News of the IEEE Computer Group Society*.

[Wiederhold, 1993] Wiederhold, G. (1993). Intelligent integration of information. In *Proceedings of ACM SIGMOD Conference on Management of Data*, pages 434–437.

[Wiederhold and Genesereth, 1997] Wiederhold, G. and Genesereth, M. R. (1997). The conceptual basis for mediation services. *IEEE Expert*, 12(5):38–47.

[Wielinga et al., 1993] Wielinga, B. J., Schreiber, A. T., and Breuker, J. A. (1993). Kads: A modelling approach to knowledge engineering. In Buchanan, B. G. and Wilkins, D. C., editors, *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems*, pages 92–116. Kaufmann, San Mateo, CA.

[Wil M. P. van der Aalst, 2002] Wil M. P. van der Aalst, K. M. v. H. (2002). *Workflow Management: Models, Methods, and Systems*. MIT Press.

[Wilsker, 1996] Wilsker, B. (1996). Study of multi-agent collaboration theories. Technical report, University of Southern California. Information Science Institute. RS-96-449.

[Wittig et al., 1994] Wittig, T., Jennings, N., and Mamdani, E. (1994). ARCHON - a framework for intelligent cooperation. *IEE-BCS Journal of Intelligent Systems Engineering*, 3(3):168–179.

[Wong and Sycara, 2000] Wong, H. C. and Sycara, K. (2000). A taxonomy of middle-agents for the internet. In *Proceedings of the International Conference on Multi-Agent Systems*.

[Wooldridge, 1998] Wooldridge, M. (1998). Agents and software engineering. *AI*IA Notizie*, XI(3):31–37.

[Wooldridge and Jennings, 1994] Wooldridge, M. and Jennings, N. R. (1994). Towards a theory of cooperative problem solving. In *Proceedings Modelling Autonomous Agents in a Multi-Agent World*, pages 15–26.

[Wooldridge and Jennings, 1999] Wooldridge, M. and Jennings, N. R. (1999). The cooperative problem-solving process. *Journal of Logic and Computation*, 9(4):563–592.

[Wooldridge et al., 2000] Wooldridge, M., Jennings, N. R., and Kinny, D. (2000). The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312.

[Xu and Shatz, 2001] Xu, H. and Shatz, S. M. (2001). A framework for modeling Agent-Oriented software. In *Proceedings of the 21st International Conference on Distributed Computing Systems*, pages 57–64.

[Yellin and Strom, 1997] Yellin, D. M. and Strom, R. E. (1997). Protocol specifications and component adaptors. *ACM Transactions on Programming Languages and Systems*, 19(2):292–333.

[Yoav Shoham, 1993] Yoav Shoham (1993). Agent-oriented programming. *Artificial Intelligence*, (60):51–92.

[Zaremski and Wing, 1995] Zaremski, A. M. and Wing, J. M. (1995). Signature matching: a tool for using software libraries. *ACM Transactions on Software Engineering and Methodology*, 4(2):146–170.

[Zaremski and Wing, 1997] Zaremski, A. M. and Wing, J. M. (1997). Specification matching of software components. *ACM Transactions on Software Engineering and Methodology*, 6(4):333–369.

[Zhang, 2000] Zhang, Z. (2000). Enhancing component reuse using search techniques. In *Proceedings of 23rd conference on Information System Research in Scandinavia*.