Interoperability Infrastructure and Incremental learning for unreliable heterogeneous communicating Systems

Abdul Haseeb

Licentiate Thesis in
Electronic and Computer Systems
School of Information & Communication Tech. (ICT)
KTH - Stockholm, Sweden 2009

TRITA-ICT/ECS AVH 09:04
ISSN 1653-6363
ISRN KTH/ICT/ECS/AVH-09/04-SE
ISBN 978-91-7415-474-0

KTH School of Information and
Communication Technology
SE-16440 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiat examen 06 november 2009 klockan 14.00 i Sal E, Forum, Kista.

Tryck: Universitetsservice US AB

Acknowledgements

This thesis arose, in parts, out of couple of years of my research that started in 2007 when I came to KTH – Royal Institute of Technology, Stockholm, Sweden. During the course of my research, I worked with a great number of people whose contributions, in assorted ways, towards my research and the making of this thesis deserve a special mention. It's my pleasure to convey my deepest gratitude and thanks to them all in my humble acknowledgment.

First and foremost I offer my sincerest gratitude to my supervisor, Prof. Mihhail Matskin, who supported me throughout my research with his patience and expert knowledge whilst allowing me the room to work in my own way. His encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject and finish my thesis. One simply could not wish for a better or a friendlier supervisor. I am indebted to him more than he knows.

I gratefully acknowledge Dr. Vladimir Vlassov for being my second supervisor. I thank him for valuable advices during my research and his contributions and feedback towards writing my final thesis.

I am as ever, especially indebted to my parents for their love and inseparable support and prayers throughout my life. Words fail me to express my appreciation towards my parents whose dedication, love and persistent confidence in me, has taken the loads off my shoulder. My father, Saeed Ahmed Jan, in the first place is the person who put the fundament of my learning character, he showed me the joy of intellectual pursuit ever since I was a child and throughout my academic career he has provided me wonderful support and motivation. My mother, Tehseen Khalida, is the one who sincerely raised me with her caring and unquestionable love. I couldn't have achieved all this in the absence of her prayers. I simply cannot thank my parents enough. I also wish to thank my lovely brother, Hamood ul Hassan and my sister Humaira Hamood for their continuous support and love during my study. I would like to thank my wife Madiha Saeed for her patience, love, understanding and bearing with my late sitting hours during the last year.

I also feel blessed by having wonderful colleagues, Nima Dokoohaki and Shahab Mokarizadeh, for creating a very friendly environment and providing me with technical help and support wherever I needed.

Lastly, I offer my regards and blessings to all my friends and everyone who supported me in any respect during the completion of this thesis.

Abdul Haseeb

Dedication

I dedicate this thesis to my lovely brother Saud ul Hassan, who unfortunately passed away in 1999. He is always in my thoughts and prayers. I will never be able to forget him. I know he is looking at me from heavens with proud feelings. I wish he was here to cherish these moments with me.

Abstract

In a broader sense the main research objective of this thesis (and ongoing research work) is distributed knowledge management for mobile dynamic systems. But the primary focus and presented work focuses on communication/interoperability of heterogeneous entities in an infrastructure less paradigm, a distributed resource manipulation infrastructure and distributed learning in the absence of global knowledge. The research objectives achieved discover the design aspects of heterogeneous distributed knowledge systems towards establishing a seamless integration. This thesis doesn't cover all aspects in this work; rather focuses on interoperability and distributed learning.

Firstly a discussion on the issues in knowledge management for swarm of heterogeneous entities is presented. This is done in a broader and rather abstract fashion to provide an insight of motivation for interoperability and distributed learning towards knowledge management. Moreover this will also serve the reader to understand the ongoing work and research activities in much broader perspective.

Primary focus of this thesis is communication/interoperability of heterogeneous entities in an infrastructure less paradigm, a distributed resource manipulation infrastructure and distributed learning in the absence of global knowledge. In dynamic environments for mobile autonomous systems such as robot swarms or mobile software agents there is a need for autonomic publishing and discovery of resources and just-in-time integration for on-the-fly service consumption without any *a priori* knowledge. SOA (Service-Oriented Architecture) serves the purpose of resource reuse and sharing of services different entities. Web services (a SOA manifestation) achieves these objectives but its exploitation in dynamic environments, where the communication infrastructure is lacking, requires a considerable research. Generally Web services are exploited in stable client-server paradigms, which is a pressing assumption when dynamic distributed systems are considered. UDDI (Universal Description Discovery and Integration) is the main pediment in the exploitation of Web services in distributed control and dynamic natured systems. UDDI can be considered as a directory for publication and discovery of categorized Web services but assumes a centralized registry; even if distributed registries and associated mechanism are employed problems of collaborative communication in infrastructure less paradigms are ignored.

Towards interoperability main contribution this thesis is a mediator-based distributed Web services discovery and invocation middleware, which provides a collaborative and decentralized services discovery and management middleware for infrastructure-less mobile dynamic systems with heterogeneous communication capabilities. Heterogeneity of communication capabilities is abstracted in middleware by a conceptual classification of computing entities on the basis of their communication capabilities and communication issues are resolved via conceptual overlay formation for query propagation in system.

The proposed and developed middleware has not only been evaluated extensively using Player Stage simulator but also been applied in physical robot swarms. Experimental validations analyze the results in different communication modes i. active and ii. passive mode of communication with and without shared resource conflict resolution. I analyze discoverable Web services with respect to time, services available in complete view of cluster and the impact and resultant improvements in distributed Web services discovery by using caching and semantics.

Second part of this thesis focuses on distributed learning in the absence of global information. This thesis takes the argument of defeasibility (common-sense inference) as the basis of intelligence in human-beings, in which conclusions/inferences are drawn and refuted at the same time as more information becomes available. The ability of common-sense reasoning to adapt to dynamic environments and reasoning with uncertainty in the absence of global information seems to be best fit for distributed learning for dynamic systems.

This thesis, thus, overviews epistemic cognition in human beings, which motivates the need of a similar epistemic cognitive solution in fabricated systems and considers formal concept analysis as a case for incremental and distributed learning of formal concepts. Thesis also presents a representational schema for underlying logic formalism and formal concepts. An algorithm for incremental learning and its use-case for robotic navigation, in which robots incrementally learn formal concepts and perform common-sense reasoning for their intelligent navigation, is also presented. Moreover elaboration of the logic formalism employed and details of implementation of developed defeasible reasoning engine is given in the latter half of this thesis.

In summary, the research results and achievements described in this thesis focus on interoperability and distributed learning for heterogeneous distributed knowledge systems which contributes towards establishing a seamless integration in mobile dynamic systems.

Scientific Contributions

Following are the scientific contributions of this thesis towards interoperability of dynamic heterogeneous systems.

1. **Abdul Haseeb**, Mihhail Matskin and Peep Kungus, Distributed Discovery and Invocation of Web Services in Infrastructure-less dynamic environments. *IEEE International Journal of Web Services Practices (IJWSP), 2009.*

2. **Abdul Haseeb**, Peep Kungus and Mihhail Matskin, Mediator-based Distributed Web Services Discovery and Invocation for Infrastructure-less Mobile Dynamic Systems. *In the proceedings of 4th International IEEE Conference of Next generation Web services practices (NWeSP'08), October 2008, Seoul South Korea.*

3. Shahab Mokarizadeh, Alberto Grosso, Mihhail Matskin, Peep Kungas, **Abdul Haseeb**. Applying Semantic Web Service Composition for Action Planning in Multi-Robot Systems. *Proc. ICIW 2009, IEEE, May 24-28, Venice*.

4. **Abdul Haseeb**, Mihhail Matskin and Peep Kungus, Light-Weight Decentralized Autonomic Web Service Discovery for Systems with Heterogeneous Communication Capabilities, *In the proceedings of 12th IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA'08), August 2008, Hawaii, USA*

5. **Abdul Haseeb**, Peep Kungus and Mihhail Matskin, Semantic Middleware for Robotic Swarm Interaction. *In the proceedings of IASTED, Modeling, Identification and Control conference (MIC'08), Innsbruck, Austria.*

Following is the scientific contribution of this thesis towards distributed learning.

6. **Abdul Haseeb,** Mihhail Matskin and Peep Kungus, DeLP based Semantic Location Lattice for Intelligent Robotic Navigation. *Proceedings of International Conference on Artificial Intelligence (ICAI'08), July 2008, Las Vegas, USA.*

Table of Contents

Table of Figures

1. Introduction

Introduction chapter of this thesis gives a brief definition of knowledge, makes grounding for knowledge management and highlights motivation for the desired scientific objectives of this research – i.e. semantic interoperability and distributed learning towards knowledge management. This is done by defining interoperability in the context of dynamic environment and heterogeneous entities.

1.1. Knowledge and Knowledge Management

Knowledge, by definition, refers to expertise and skills acquired by an entity through its experience and/or education; thus it's a theoretical or practical understanding of an entity about a certain concept or a subject. Another definition of knowledge is factual data and information acquired by an entity during the course of its interactions with its environment and surrounding entities and experiences gained through such interactions (Oxford Dictionary).

Philosophical stand point [1] on knowledge considers a three facet definition, in which a certain statement is declared as knowledge if it satisfies justification/rationality, truth and belief. Another definition of knowledge from philosophical point of view is a justified personal belief that increases an individual's capacity to perform actions [2].

This thesis first presents knowledge and knowledge management in perspective of a number of cases to motivate the research direction.

1. Knowledge Acquisition – a case for cognitive reasoning and collaboration

Knowledge and knowledge acquisition is considered a complex cognitive process, which an entity employs to build-up its knowledge (experiences and intelligence). This cognitive process includes an entity's perception about its surroundings, its interactions with surrounding entities, its learning (stand alone learning or collaborative learning in case of collaborative systems), its association building across various data sources and its reasoning/inference. These processes (or various dimensions of knowledge acquisition) result in creation of an entity's understanding towards a certain subject or concept.

2. Knowledge Sharing – a case for collaborative interactions between entities

Another important aspect of knowledge is its implications for sharing of an entity's knowledge among other entities. Knowledge sharing requires effective communication – means to make shared knowledge usable/re-usable for other entities. Here usable refers to an entity's ability to make good use of shared knowledge. Notion of effective communication for knowledge sharing brings us to a discussion on semantics of communication, which is covered in the following sub-section of this chapter (and in chapter 2, in the context of dynamic system settings – in particular for Robot swarms

[27]). The highlight from sharing aspect is seamless communication and collaboration between entities to create a coherent and usable knowledge continuum.

3. Explicit and Implicit knowledge – a case for collaborative interactions based learning

Another dimension of knowledge is its explicit and implicit nature. The definition of knowledge and associated cognitive processes involved in knowledge acquisition is an on-going debate. This thesis doesn't delve into a confusing discussion of definition of knowledge, rather takes a general disclosure of interchangeable use of *Information* and *Knowledge*. Thus information of an entity is considered as its knowledge. Such a disclosure requires classification of knowledge and information into *focal* (explicit) and *tacit* (implicit) information. Thus there is a separation of concern between gathered factual information and implied/reasoned knowledge [3]. Formal languages are used for communication and sharing of explicit knowledge in terms of formulae, specifications, frameworks, articles etc. [4]. Tacit information on the other hand uses specified problem-specific solutions, for instance, formulation and sharing of tacit information in terms of rules, theorems, clauses etc. Important aspects of tacit information are its un-teachable, non-articulated and complex nature [5]. These aspects are elaborated in more detail in chapter 6 in the light of incremental distributed/collaborative learning.

4. Knowledge management – a case for acquisition and communication

Knowledge management (KM) is a process of acquisition, organization and communication of knowledge and information (both tacit and explicit). Knowledge management, defined from an activity perspective, is a process which deals with treating knowledge of activities at various levels and stages. It deals with creation of connections between intellectual assets (i.e. information, data from various activities and learnt experiences/decision-matrices) and desired outcomes. Thus, it's a process of identification and mapping of information, enabling generation of new knowledge, and providing access to information. Technologies for knowledge management include software agents, data warehousing, document management systems etc.

In computational networks or distributed systems, knowledge management refers to an infrastructure for manipulation of knowledge and resources. It deals with <u>communication capabilities</u> of entities (not the computational capabilities).

5. Semantic Interoperability and Learning for Knowledge management

The main purpose of giving a brief definition of knowledge here is to make some grounding for knowledge management (KM) and to give a motivation for the desired scientific objectives of this research – i.e. semantic Interoperability and distributed learning towards KM. After giving a brief introduction to knowledge and KM, notion of semantics towards KM and interoperability are elaborated.

To carry on the discussion on communication, usage of semantics towards information/ KM integration is given.

1.1.1. KM Integration via semantics

This thesis takes a general disclosure of information of an entity as its knowledge. In order to highlight the need for semantics, a consideration of complexities of semi-/un-structured data sources is mentioned in a general tone.

Rapid growth of information makes its search, organization, access and maintenance inherently complex [6]. Most of the complexities arise due to semi-/un-structured nature of data sources. Thus establishing a meaningful relationship between un-structured data sources is not possible. Towards such problems, *semantic augmentation*, an exploitation of machine-processable meta-information, is proposed to be used [7]. In general KM addresses the development of tools and techniques for management of information/knowledge [8]. <u>In other words it refers to finding knowledge and resources relevant to required knowledge/information/capabilities [9, 10], which leads us to resource discovery problem.</u>

Traditional knowledge management systems developed in commercial and research sectors generally suffer from the following shortcomings:

- *Key-word based searches*; Key-word based search for knowledge and resources often leads to irrelevant information/retrieved-resources. Such approaches are unable to create a meaningful inter-relationship between different fragments of knowledge and in terms of resources this refers to lack of relationships among resources. This lack of exploitation of inter-relationship of knowledge and resources results in isolated pieces of knowledge/un-linked resources [11],
- *manual knowledge extraction;* manual extraction from huge number of data sources is not a trivial task, thus automation is needed,
- Maintenance of *weakly-/un-structured information sources*; It is inherently complex to integrate knowledge and resources that are attributed by un-structured information. There is a need to structure knowledge and resources in a way to make integration more useful.

This thesis focuses on integration aspect of knowledge and resources. Integration activity is a creation of *coherent inter-related information*. As mentioned previously about problems of weakly-/un-structured information resources, lack of structure makes integration a difficult task. On the other hand, integration of structured data-structures such as data-bases is trivial using formal languages and integration schemas [12]. But databases represent highly structured data sources in contrast to semi-/un-structured natural language based data sources such as documents, web-pages, web-resources (for instance Web services) etc. From this endpoint the notion of semantics to annotation [7] (an explicit representation of meta-information, augmented by domain theories) is needed to address the issues of weakly structured data/resources (this aspect is covered in more detail in section 2.4 about semantics based communication solutions).

Semantics are machine-processable meta-data that refers to creation of a consensual data-structure for structuring un-structured data sources. Consensual means shared and agreed-upon data-structure across entities. Main problem with integration of heterogeneous un-structured data sources stems from *heterogeneous representation*. Domain models, such as expressed by ontologies, provide a way to cope with representational heterogeneity by giving a unifying structure for common information representation. Ontology driven knowledge management has already been proven to be useful for large scale knowledge environments [13, 14].

As an example of semi-/un-structured natural language based data sources such as documents and web-pages, there is an estimation of around 300 million static objects in WWW [8]. These repositories are usually searched by keyword-based search engines allowing a user to retrieve information by stating a combination of keywords. Documents downloaded from the Web are indexed according to their contents and only those matching the query are returned to the user. The results of this type of search usually suffer from problems derived from both the nature of the query and the lack of structure in the documents: there is some irrelevance in retrieved documents and some of the relevant documents are no retrieved – due to lack of semantics (thus a low precision and recall ratios).

Thus, semantics play a significant role in automated extraction of retrieved information [15]. In its conventional form an ontology is a representation of shared concepts in a certain domain and a hierarchy of defined concepts. This facilitates communication and collaboration. In general semantics have been found to be useful for:

- *Retrieving of appropriate information* by providing a structure for annotation of resources with semantic information [16, 17].
- *Integrating of information from various sources* by providing a structure for its organization and facilitating exchange of data, knowledge and models [18, 19].
- *Ensuring consistency and correctness* by formulating constraints on the content of information [19].
- Creating libraries of *reusable models* [18, 19].
- Supporting inference on gathered information [20, 21].

In conclusion, usage of semantics is significant from integration point of view of a resource management problem. Motive of giving KM integration via semantics, in a very general and abstract tone, was to motivate semantic interoperability and semantic learning. In more specific words, semantic play a huge role in interoperability of entities (even knowledge/information sources can be abstracted to an entity), and such is of more significance when loosely coupled/linked communication entities are considered (interoperability objective of this thesis). Another associated aspect is semantic learning when global knowledge constraints can't be met (distributed learning objective of this thesis).

1.1.2. Decentralized Resource Management

The current state of art resource management solutions are based on centralized server repositories with ontologies forming the conceptual backbone for resource management brokering [22]. Ontology, as an explicit specification of conceptualization [23], has provided significant contributions to develop resource management solutions [9, 10, 24, 25 and 26]. The centralized model brings advantages in terms of scope, control, and organization. But it poses problems for ad-hoc or loosely coupled systems [27]. To addresses such problems, distributed architectures based on distributed ontologies are proposed [28].

Recently there has been a focus on synergizing the potential of P2P computin, and formal approaches to resource organization. Ontologies, though, provide consensual terminologies (theory relating to concepts, their relationships and constraints) for data thus enabling machine-processability, however, building/maintaining ontologies is not a trivial task. Thus *a model of 'evolving' ontologies is needed*. There has been some work in the area of formal concept learning and automated ontology learning. In other words, instead of a central top-down approach, there is a need to have a distributed way of emerging and aligning ontologies. There is a need for a model for space and time gluing of network of incoherent ontologies (realistic assumption being a dynamic nature of such ontological networks if distributed resource management paradigms are to be considered).

Generally speaking, P2P computing paradigms remove the limitations of centralized solutions [29]. Applications of distributed computing for knowledge management can enable decentralized and distributed knowledge management thus achieving self organized management. Such distributed and decentralized knowledge management with a dynamic nature of content and resources are applicable in a number of domains, for instance mobile computing, ubiquitous computing, robotics [27], and collaborative intelligent systems [30]. Main objective is to integrate the advantages of ontology based KM solutions and distributed computing. But such a solution poses some research challenges. In a broader perspective these challenger are:

- *Selection of knowledgeable distributed entity*: with distributed computing paradigm selection of knowledgeable entities is needed to be done in a decentralized way. In most of the current distributed computing systems selection of an entity is based on network related parameters such as ping-time, network hops etc. This needs to be changed in a way that the query is routed to the next knowledgeable distributed entity instead. Thus a sort of *ontological similarity* is needed to be devised for efficient routing of queries. There is a need to define conceptual distance for semantic similarity in a generalized and scalable way.

- *Ontological drift*: ontological drift implies constant maintenance of dynamic/changing ontologies. In traditional systems such maintenances is possible because they employ a single ontology at a centralized server. In decentralized systems, there is a need to design a mechanism to cope with ontological drifts.

- *Inference on sloppy ontologies*: ontology based applications rely on clear definition of terms and relationship between them. With decentralized control each entity is autonomous and has its own ontology, which could be sloppy in the perspective of entire systems. Even a single miss-defined instance can invalidate the class relationship. Existing work on approximate inference approximation [31, 32] and local containment of inconsistencies [33] can provide a required basis to tackle the problem of sloppy ontologies.

- *Variation of ontologies*: distributed entities can use different yet overlapping ontologies. Emergent semantics [34] can be used for alignment and mapping of varying ontologies.

- *Mobility and dynamism of content and entities*. With constant mobility of distributed entities and evolution of knowledge resources, the system maintenance has to be done in a decentralized yet scalable way [27].

There have been some research efforts corresponding to P2P computing for Ontology based Resource management [35, 36]. Also there has been some work on Semantic Overlay Networks – SON [37] which tries to improve the query efficiency in P2P networks by clustering nodes with semantic similar contents. SON approach uses the rigid classification hierarchies to find semantic similarity. SON suffers from limitations of using a predefined hierarchy. Moreover, SON membership and clustering takes a conservative approach and uses generalized ontologies. Ontology merging and integration issues in literature are either abstracted or they are left to application layer.

The general introduction of decentralized resource management given here, enables to highlight the main focus of this dissertation, which is on *Mobility of* entities (problems caused thereof)*, dynamism of content* and *heterogeneous nature of entities.* After introducing decentralized resource management and semantics in communication, a more formal definition of interoperability in the context of dynamic systems is given.

1.2. A formal introduction to Interoperability

Interoperability is a property that refers to the ability of diverse systems to function together (i.e. an ability to interact with each-other). IEEE [38] defines interoperability as:

> *"The ability of two or more systems or components to exchange information and to use the information that has been exchanged".*

Interoperability can be classified into *Syntactic* and *Semantic* interoperability.

- *Syntactic Interoperability*: Communication and exchange of data between systems refers to syntactic interoperability. Data formats, communication protocols etc. are fundamental in achieving syntactic interoperability. For instance, usage of XML as data exchange format and HTTP as a protocol between any two entities can result in both systems communicating with each-other in an interoperable fashion.

- *Semantic Interoperability*: The ability of automatic interpretation of information exchanged between entities as defined by the end users of systems/entities refers to semantic interoperability. For achieving semantic interoperability, entities need to use a common information exchange reference model. For instance RDF (Resource Description Framework) and usage of ontologies to unambiguously define the content of the information exchange: i.e. what is sent by an entity is the same as what is understood by the recipient entity.

The term interoperability is used to describe the capability of different programs to exchange data via a common set of exchange formats, and to use the same protocols. According to ISO/IEC 2382-01, Information Technology Vocabulary, interoperability is defined as:

> *"The capability to communicate, and executes programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units".*

This definition focuses on the technical side of interoperability. As the user of a program can be another program/entity and if the latter is a portion of the set of program that is required to be interoperable, it requires having knowledge of the characteristics of other operational units. Notion of interoperability is usually used in the context of autonomous systems and the notion of integration for data. Data is typically considered in terms of its declarative representation i.e., it can be considered syntactically, structurally, and semantically. Thus integration is described in terms of the degree of syntactic and semantic correspondence between varying data sources with varying declarative representations of data [39]. Interoperability, on the other hand, implies interoperability between software constructs.

1.2.1. Interoperability and dynamic systems

Mobile autonomous systems like robot swarms or mobile software agents often operate in a dynamic environment which pertains self-organization, self-configuration and heterogeneity of computing entities [27]. The discussion in this dissertation, from this point onwards, refers to robots as entities or vice versa. It is important to highlight the difference between self-organization and self-configuration before delving into discussion of interoperability for dynamic systems.

- *Self-organization* refers to a process of attraction and repulsion in which the internal organization of a system, based on constituent autonomous entities, organizes to display emergent properties without a centralized source. In swarm robotics, self-organization is used to produce emergent behavior. A self-organizing system is a system that changes its basic structure as a function of its experience and environment [40].

- *Self-configuration*, on the other hand, refers to selection of best functional configuration of cooperating entities for allocation and connection among entities. In other words, a set of entities can perform different tasks by using different configuration. Selection of the best configurations, on the basis of available resources refers to self-configuration [41].

It is important to mention that in infrastructure-less heterogeneous environments interoperability issues are easy to arise due to heterogeneity of in-used application middleware and hardware, different APIs and programming languages. Thus an interoperable and just-in-time integration solution is needed. Earlier efforts on integration and connectivity were primarily focused on physical and syntactic layer, for instance by using ODBC (Open Database Connectivity), MOM (Message Oriented Middleware) etc. But such approaches require an *a priori* specification of semantics for communication. Furthermore, no assumption can be made about homogenous communication capabilities of certain environments [42]. This puts forward further requirements to integration and interoperability in the infrastructure-less autonomous systems for heterogeneous capable entities.

Interoperability in autonomous systems relies on effective and seamless knowledge sharing and communication. For instance, robot swarms are attributed by dynamic environment. In robot swarms the control to enable robots for accomplishing complex routines and intelligent behavior not only depends upon robot's perception of environment but it also depends upon its collaboration with other robots, exposing its own functionality and re-using the functionality provided by other robotic/computing entities in the surrounding and communication with outside world. Efficient operation of such mobile autonomous systems relies on seamless knowledge sharing, communication and collaboration with other entities both inside and outside the environment in a symmetric way. Let's consider context of environment and openness considerations for interoperability.

Context of external domain and capability sets

In autonomous systems or robot swarms, each entity plays a role in the context of an external domain or environment [43, 44 and 45] and provides certain functionality. Moreover, there are different capability sets in certain systems [27, 42].

Open environments and Just-in-time integration

Seamless communication and collaboration between entities both inside and outside the environment, puts forward requirements for open environments with autonomic and decentralized publishing and discovery of services and on-the-fly service consumption. In other words, knowledge sharing, communication and collaboration should be based on *on-the-fly resource discovery and consumption*. There is also an aspect of openness, which infers that services/resource-discovery/invocation can be provided both within and outside the environment in the external world [46]. In the latter case the external environment is needed to provide a programming logic for the systems [47, 48].

Moreover, towards openness of a system, this thesis conjectures that intelligent and adaptive behavior of entities not only emerges from relationships between different computing entities in an environment (between different entities and distributed knowledge sources) but also from re-using intelligent functionality implemented in external world [46]. Most of the current solutions assume independent actions of its constituent entities. In many cases it is assumed that a system is completely autonomous and just allows uploading of programs from the external environment [47, 48]. Thus there is a need to consider the interoperability aspect of autonomous systems from a, rather, broader perspective.

1.2.2. WS for Interoperability

Platform- and language-agnostic abstraction layer, as suggested by Web services (WS), can provide a mature solution for the above mentioned problems of integration and interoperability. Usage of external functionalities as Web services, to serve resource-constrained devices, has already been proposed [51] and symmetrically reverse approaches have been developed as well [52, 53]. In such approaches WS are applied in order to facilitate cooperation between distributed entities for joint tasks execution and distributed control. WS technology focuses on loose coupling, yet focus on semantics of communication is needed for seamless integration and interoperability [49, 50]. Towards these objectives ontologies provide a solution for semantic heterogeneity [49, 50].

An aspect which is overlooked in literature is the exploitation of WS solutions in distributed control and infrastructure-less systems. WS are generally exploited in stable client-server networks, where Web services registries [54] are used for discovering required services/resources. However, in case of infrastructure-less dynamic environments existence of centralized control/registries is a pressing assumption. This leads to issues of keeping an updated view of available services and providing a reliable Web services discovery. Furthermore, lack of stable connectivity [55, 27] and heterogeneous communication capabilities [27] puts forward the requirement to use a mediator-based communication [56].

Goal of this thesis is to extend and generalize these approaches to infrastructure-less mobile dynamic environments, in particular to robot swarms [27].

1.3. Design Objectives

There is a need to provide a simple communication infrastructure for infrastructure-less and heterogeneous communication-capable environments to address semantic heterogeneity and connectivity. Following are some design principles which this thesis follows for achieving the interoperability communication infrastructure:

- *Standardized communication infrastructure* that allows loose physical/syntactic coupling between heterogeneous entities.
- *Usage of semantics-based technologies* to overcome interoperability issues [57].
- *Dynamic adaptation of entities to newer domains* without a priori incorporation of semantics of domain knowledge in systems designs [58, 59].
- Flexibility *to reuse external context-aware and on-demand services* [46].

In order to support these design objectives, this thesis employs a standard solution based on semantic Web services technology.

SOAP (Simple Object Access Protocol) message exchange for interconnection between heterogeneous networked devices and Web services to open controllable interfaces has already been employed for a representative home information system [60]. According to this view entities/devices are seen as Web services, which are accessible from the environment. In other words, functionality of entities is exposed as Web services and different applications may invoke such services whenever needed. Similarly entities can also access external functionality (either from entities in environment or by discovering external services).

While infrastructures based on Ontologies and Web services provide a simple and unified access to a resource/entity; poor computational and service discovery capabilities of entities (in a resource constrained and/or heterogeneous communication-capable-entities, in an infrastructure less setting) require some sophisticated mechanism for Web services discovery and their provision. In order to provide a solution to this problem this thesis assumes a powerful Web services discovery mechanism outside the entities (i.e. in external environment – in form of a communication middleware).

1.4. Mediator based distributed WS infrastructure

This thesis proposes a *mediator-based distributed Web services discovery and management middleware* which is primarily targeted (evaluated and deployed) for robot swarms [27]. However, the proposed principles are general and can be applied to any infrastructure-less dynamic environment. In proposed solution, dynamic environment and computing entities are abstracted to a P2P system and a conceptual classification of entities according to their communication capabilities is used. It is assumed that the physical environment has “mailboxes” (RFID [Radio Frequency IDentification] tags or some other type of mediator entities), which can be used for communication when point-to-point communication between entities is not possible.

Developed solution provides a loose coupling in terms of space and time and overcomes the shortcomings of availability of both Web services requesters and Web services providers at the same time [69].

1.5. Focus of Research

This dissertation is primarily focused on interoperability objectives in mobile dynamic systems such as robot swarms and partly on distributed learning in the absence of complete information. The research work is aligned towards ROBOSWARM project [27]. To introduce briefly, the general objectives are to develop an open knowledge environment for self-configurable, low-cost robust robot swarms which could be usable in everyday applications such as cleaning etc.

As mentioned in the introduction of knowledge management, integration of ontology based KM solutions and distributed computing principles pose some challenges. Specifics of this thesis towards those aspects are *mobility and dynamism entities*.

Systems such as robot swarms, due to constant mobility and evolution of robots, impose challenges such as decentralized system maintenance, interoperability of heterogeneous robots, coping with communicational heterogeneity and distributed collaborative learning etc. An operational scenario, taken in this thesis is a robotic swarm in which robots are attributed by heterogeneous communication capabilities (i.e. few can communicate wirelessly in a point-to-point fashion to other robots, while the others need mediators/RFID for their communication). Communication environment of robot swarms is attributed to dynamism and unreliable-nature. Dynamism and mobility infers that during a system execution no guarantee can be given about the successful communication between any two robots. That can stem from highly mobile and dynamic nature of robots, their constrained communication capabilities and robots being out of communication coverage at a certain point in time etc.

Two main aspects covered in detail in this thesis are:

- ***Interoperability of entities/robot-swarm*** both with environment entities/robots and with outside world in a symmetric way.
- ***Distributed learning*** in the absence of global knowledge.

Objectives of this thesis are to provide a guideline for creation and maintenance of distributed communication infrastructure based on intelligent inter-entity (inter-robot) communication. Scalability of the system arises from decentralization of control and local manipulation of entities/robots. For seamless integration/interoperability objectives Web services standards are considered for exposing robotic/entities' functionality to outside world (thus each entity/robot is abstracted to a Web services interface).

Towards distributed learning a rule based learning approach is used. This thesis elaborates an implementation of a defeasible reasoner which is employed for building a collaborative location lattice for intelligent robotic navigation.

1.6. Thesis disposition

This chapter introduced knowledge management, need for interoperability, role of semantics in communication and distributed learning.

Chapter 2 elaborates the case for thesis – i.e. dynamic system and environments. Chapter 3 establishes the grounding for distributed Web services infrastructure by highlighting the existing research efforts and missing gaps in existing solutions in literature. Based on chapter 3, chapter 4 presents "mediator based decentralized Web services infrastructure". Chapter 5 presents detailed evaluations and results of developed communication infrastructure, while chapter 6 focuses on incremental learning in the absence of global information.

2. Interoperability and learning aspects in distributed systems

Previous chapter presented motivation for interoperability and distributed learning towards distributed knowledge and resource management. This chapter enlists the design aspects of heterogeneous distributed systems. This chapter takes a representative example of swarm of mobile autonomous robots and motivates a Web services based solution by giving an overview of existing solutions for interoperability in robot swarms that use semantics for interoperability.

2.1. Heterogeneous distributed systems

Distributed systems refer to a collection of independent yet interacting components/entities that cooperate together to reach a defined system goal. In a distributed system there can be many different kinds of hardware and software components working together in cooperation with each other for distributed problem solving. Not only are the entities/components versatile, there may be many different data representations in the distributed setting as well. The heterogeneity among entities can also stem from different capabilities among various distributed components. These different capabilities might include clock cycles, memory capacities, different services/functionalities, communication capabilities etc.

If a generic distributed system is considered, the following aspects/features (a non-exhaustive list) from a distributed computing standpoint can be highlighted that should be considered in a distributed system design. Few of the aspects, that are paramount in understanding the case of robot swarms [27], are listed as following.

1. Distributed control – lack of underlying infrastructure

In a true distributed system there is no central point of control, rather, a decentralized management paradigm is employed. Each distributed entity operates independently of other entities in the system. Entities can dynamically join and leave the system at any point of system execution and can perform reasoning/decision-making autonomously. A simple case for autonomous decision making could be decision about which entities to communicate to.

Thus in a distributed system actions of individual entities are governed by policy guidelines which reflect the behavior changes of the distributed system. Distributed systems rely on the availability of core underlying infrastructure to provide a distributed control mechanism.

In some cases the core infrastructure and its associated services can be missing as well (as in representative setting of swarm of robots [27]).

2. Asynchronous distributed communication

In a distributed system, communication between entities is inherently asynchronous. There is neither a global clock nor a global synchronization mechanism. The need for asynchrony can be viewed from two aspects, first from the point of view of sequence of operations and secondly from the point of view of communication between any two entities. Underlying infrastructure for distributed systems should provide services for asynchronous operations/interactions between entities.

3. Partial Failures – Dynamic Join/Leave – Dynamic Partitioning

At any instant of system execution, any number of distributed entities of the distributed system can fail (or disappear). Such is referred to as a churning behavior. This comes from the fact that distributed entities can dynamically join/leave the system, or in some settings it can be caused by entities moving out of communication coverage area. In the design of distributed systems, *high availability* is needed be realized, which enables the system to operate properly even in case of partial failures.

Another aspect of partial failures is a graceful degradation, which means that the distributed system gradually loses capabilities as more and more of the elements in the system fail. Just because some of the services are not available does not mean that useful work cannot be accomplished (this aligns with the settings in which disappearing of entities is the cause of churning behavior). Thus there is a need for distributed mechanism for tracking changes in the systems and reacting when a target entity/resource becomes available. Another facet of failures is the dissolution of distributed system into partitions. Such can also be seen from the point of view of certain entities not being able to communicate to certain other entities resulting in the formation of disconnected communication clusters.

4. Absence of a global State – distributed coordination/decision making

In a distributed system there is an absence of a global state. The state is divided among distributed entities. In a non-distributed system, global state is typically defined and maintained in terms of finite state machines (FSM). In distributed settings, local decisions should propagate higher to achieve global decision or system-wide goals.

5. Concurrency – resource locking (conflict resolution mechanisms)

Components in a distributed system work independently. There could be cases in which a shared resource is utilized by distributed entities in a concurrent fashion. Concurrent access to shared resources has to be dealt with to avoid deadlocks and race conditions among entities.

6. Mobility - change is the only constant

Distributed systems are also attributed by mobility. In mobile settings entities/services can join and leave, without interfering with other service/entities. If there is dependence among services/entities then the dependent service/entity cannot complete their task in the absence of required services/entities.

7. Openness of distributed system

In heterogeneous distributed systems, individual system components can cooperate via open protocols (here openness refers to a mutual understanding of various components about the used communication protocols, terminologies, conventions and standards). Openness is the ability to plug and play. In other words, if two equivalent services follow the same interface contract then they can interchange information with each other.

Presented (and on-going) research work does not cover all the above aspects. It primarily focuses on distributed learning and interoperability. This thesis addresses the following aspects of distributed systems of mobile heterogeneous entities.

- *Asynchronous distributed communication*
- Coping with mobility of entities and ability of *system to function under partial failures*
- *Resource locking* (conflict resolution mechanism) for concurrent access to shared resources
- *Distributed learning/decision-making* with local information.

Now a representative distributed setting (swarm of robots) is presented.

2.2. Swarms of mobile robots

A Representative distributed heterogeneous system

This thesis takes swarm of mobile heterogeneous robots as a representative distributed system. Current research work does not cover all aspects of distributed systems as mentioned above. Main contributions of this thesis are “mediator based decentralized Web services infrastructure” (that provides a solution for interoperability in a heterogeneous mobile settings) and distributed incremental learning of autonomous entities in the absence of global information towards establishing a bottom-up decision-making in robot swarms.

The research is carried out in the context of ROBOSWARM [27] - an “EU framework 6 project”. The main objectives of the project are to develop an open knowledge environment for self-configurable, low-cost and robust robot swarms operating in dynamic environments. The system is attributed by heterogeneity of communication and computation – i.e. no assumptions are made on the availability of communication capabilities of robots and robots provide versatile services. In a typical setting, as assumed in this thesis, few robots are equipped with a wireless capability while the other robots, in the absence of wireless capability, use RFID tags for their communication.

Main objectives of ROBOSWARM are self-organization of swarm, collaborative task-sharing, and interoperability of swarm both with environment entities and with outside world in a symmetric way. In ROBOSWARM project a distributed data environment, based on intelligent inter-robot communication and a global robot database solution for knowledge reuse is considered. Scalability of the system is attributed by decentralization of control and local manipulation of robots (i.e. local decision making).

The general objectives of ROBOSWARM project are:

- Effective and intelligent communication on a large scale for tasks division for increased functionality of the swarm (scalability).
- Robotic learning from their experiences via the global and/or local knowledge base(s), to assure the optimal behavior in difficult situations (self-leaning).
- Autonomous operation with minimal sensing capabilities (cost efficiency).
- Reuse of swarm functionality from outside world, and reuse of external functionality in a symmetric way (Interoperability).

Each robot in robot swarm is, considered to be, capable of performing certain actions and the coordination among such autonomous robots contributes to a higher intelligence of swarm than each member of the swarm independently. Web services are employed in this thesis for exposing robotic functionality. Thus each robot is abstracted to a Web services provider. Robotic actions are formulated by a local robotic reasoning engine which performs local decision making. Such local decision making is based on factual information, which a robot gathers during its execution, and a set of ground rules. These facts and rules as referred to as **LRKB (Local Robot Knowledge Base)**. Detailed discussion of solution for local inference in the absence of global knowledge is presented in chapter 6.

Apart from LRKB of each robot, a **GRKB (Global Robot Knowledge Base)** is considered which refers to collective swarm knowledge. GRKB is assumed to be a server, equipped with wireless capability, which performs higher level reasoning, and provides resource intensive services to swarm entities. Also GRKB is used in this thesis as a gateway entity to swarm for communication with outside world. In principle any swarm entity/robot with wireless connectivity can be considered as a swarm gateway.

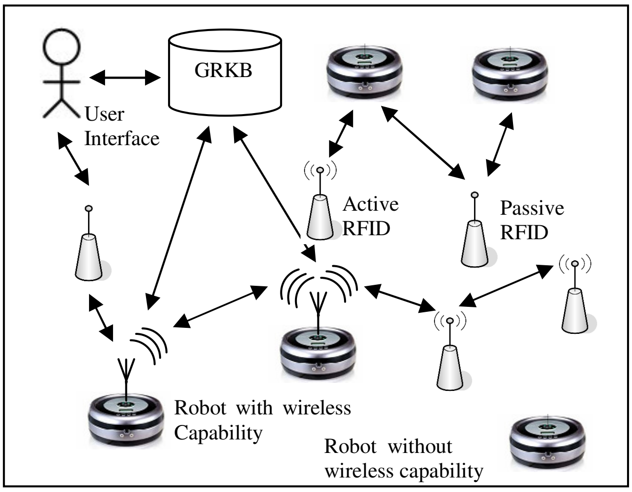


Figure 1: Knowledge environment for robot swarms

A general usage scenario of robot swarm is given in Figure 1. One general classification criteria (which will be elaborated in detail in chapter 4) is the wireless capability of robots. In developed middleware the classification criteria is used to define roles for robots for Web services discovery and request/response dissemination. Similar to an earlier solution, in which a Web services based robot control platform was developed using RFID tags as information mediators [56], RFID tags are used in ROBOSWARM as message postboxes i.e. message relays for non-wireless capable robots in the swarm. As mentioned previously, access to GRKB depends upon wireless capability of robots in swarm. User/operator instructions could come via GRKB or, in principle, by direct manipulation of any computing entity of swarm.

Summary of goals of this thesis in robot swarms settings are:

- ***Interoperability of entities/swarm*** both with environment entities/robots and with outside world. For seamless integration/interoperability objectives Web services standards for exposing robotic functionality to outside world are considered. Main contribution for this objective is establishing of an underlying communication platform/infrastructure which supports distributed Web services discovery and invocation in a highly mobile and dynamic setting. (Developed infrastructure is elaborated in detail in chapter 3-5).

- ***Distributed learning*** in the absence of global knowledge. Towards distributed learning this thesis follows a rule based learning approach. Defeasible logic is employed for incremental learning of robots, which is used for incremental location lattice building for robotic navigation (as a representative use-case of incremental learning) and control using local information (chapter 6).

Here both aspects and existing approaches in literature are introduced.

2.3. WS based interoperability solutions in robot swarms

As highlighted in introduction to knowledge management chapter, the key to efficient knowledge management and sharing paradigm relies on seamless knowledge sharing, effective communication and collaboration with other entities both inside and *outside the environment*. The notion of “outside the environment” is taken as a higher level abstraction for openness of the system. Here “outside” can refer to out-of-domain communication with external world entities i.e. entities dispersed in internet etc.

Seamless knowledge sharing and communication puts forward the requirements for open environments which allows autonomic and decentralized publishing and discovery of resources. Here Web services (WS) are considered as resources and the need for just-in-time integration for on-the-fly service consumption without any *a priori* knowledge about services and entities in the system is highlighted. The services can be provided both within and outside the environment in the external world [46]. In the latter case the external environment provides a programming logic for the systems [47, 48].

It is important to note that systems such as robot swarms and mobile ad-hoc networks are attributed by lack of communication infrastructure. Interoperability issues are easy to arise in such systems due to heterogeneity of in-used application middleware and hardware, different access APIs and programming languages employed by control program of involved entities. Furthermore, one of the assumptions of this thesis is heterogeneity and versatile communication capabilities of the entities [70].

Platform- and language-agnostic abstraction layer, as suggested by Web services, can provide a mature solution for above mentioned problems. But semantics of communication play a significant role for seamless integration as well [71]. Towards semantics of communication, ontologies provide a solution to overcome the problems of semantic heterogeneity [49, 50]. To follow-up on this comment, some existing solutions of communication in robotic world using semantics are given in subsection 2.4.

As motivating examples, let's consider few Web services based interoperability solutions for robot swarms.

- Usage and abstraction of external functionalities as Web services was proposed to serve resource-constrained devices with higher level and resource intensive computational capabilities such as AI functions [46].
- Reverse approaches to that of [46] were also proposed in literature. For instance, Web services have been applied on robots to facilitate cooperation between robots for joint tasks execution and robot control [52, 53].

Both of the above mentioned solutions [46, 52 and 53] assume an established communication infrastructure, which, is a pressing and unrealistic assumption. Such can't be assumed in robot swarms. Thus, there is a need to extend and generalize these approaches for communication infrastructure-less settings.

Another keynote of this thesis is the negation of inherent assumption of Web services exploitation in stable client-server networks, where Web services registries [72] provide means for discovering the required services. However, in case of infrastructure-less dynamic environments, existence of centralized control/registries cannot be assumed. This leads to issues of keeping an updated view of available services and providing reliable Web services discovery. Furthermore, lack of stable connectivity [27, 73] and heterogeneous communication capabilities [27] puts forward the requirement to use mediator-based communication [56].

In short there is a need to establish a solution for the following higher-level goals (which have been overlooked in literature and in earlier solutions):

- Exploitation of Web services in an **Infrastructure less** setting.
- Dealing with **communication heterogeneity** using mediator based communication.

In the solution proposed in this thesis, dynamic environment and computing entities are abstracted to a P2P system and a conceptual classification of entities according to their communication capabilities is used. It is assumed that the physical environment has “mailboxes” (RFID tags or some other type of mediator entities), which can be used for communication when point-to-point communication between entities is not possible. Proposed solution in this thesis provides a loose coupling in terms of space and time and overcomes the shortcomings of availability of both Web services requesters and Web services providers at the same time [69].

2.4. Semantics based communication solutions in robot swarms

Ontological representation enables knowledge sharing and communication and as a result leads to improved interoperability in and between autonomous systems and entities. Based on this comment, few significant research activities in the area of ontological knowledge representation in robotic paradigms are discussed below.

One aspect of robotic ontology can be the normalization of incorporated components and thus the specification of subsystems of robot, simulators, fault detection system and metrics. Another aspect counters the behavioral aspect of robots by referring to ontological concept classification of environment, which a robot might encounter during its task execution. This poses another aspect of task representation in a certain context.

Incorporation of semantics for robotic concepts, domain descriptions, tasks, plans and dynamic action management finds its roots in earlier applications of description logics [89, 90] and in machine intelligence [91, 92 and 93]. In earlier work such as [89], taxonomic reasoning tools were used for robotic action management and control. They proposed using a taxonomy representation of robotic actions, plans and domain concepts expressed as temporal logic. Such proposed solutions corresponded to manual specification of input, and domain concept description for robotic systems, which imposed limitations on robot adaptation to newer domains, environments and understanding the semantics of newer inputs. Such existing notion of conventional functionalism is argued strongly in literature [94].

Few of the noteworthy efforts of semantics based communication solution in robot swarms are as following:

- **Development, test and validation of mobile robot system using ontological models.** For a functional validation there is a need for a representative model of a robot and its environment. One of the earliest efforts in using ontologies for mobile robots validation is mentioned in [74]. Their approach employed environment representation with a set of descriptors specified from the ontology. Robot behavior is, then, expressed as patterns using chronicles formalism [75] and is matched by the values taken by environment descriptors.

- **Modeling situational context for robot environments.** A robot environment represents a situational context for robot and features physical notions like geometry, space and time. Ontology can present different sets of views on the domain as subsystem relations of aggregation/decompositions, physical components, notion of tasks and flow/transmission of data between tasks and components. The approach given in [74] translated the ontologies and modeled them into prolog knowledge-base. They assigned agents to subsystem components[1] thus forming a multi-agent simulator.

  Another approach for situation context modeling of robots is given in [84] that proposed a number of steps for situational context modeling using rules and ontologies. Those steps included the following:

  1. Categorization of real world (based on similar features, functionalities)
  2. Definition of context vocabulary and terms called context concepts
  3. Development of criteria for context concepts
  4. Development of rules using context concepts and criteria. Rules model dynamic information, for instance current location
  5. Development of ontology of real world categorization and other static information

- **Evolutionary robotic learning**. Another motivating work in metaphorical extensions of physical concepts and evolutionary robot learning is presented in [76]. Their work is based on understanding of input and robotic evolution to produce the right outputs. Such view of functionalism is mostly explored in a conventional approach of manual specification of semantics of input and desired response. To counter manual specification, [76] used the approach of natural semantics[2] for robots. In which robot could learn concepts and semantics by interacting with its environment. More specifically in their proposed approach each robot is equipped with a set of basic activities and a notion of curiosity to explore things. Robots collect experiences[3] and statistical procedures can find common sequences from experiences. A cluster of such common sequences correspond to common activities of robots, for instance, bumping into wall or grasping a cup. This way, robots can identify the roles (object, subject) in common activities. If a robot can perceive the roles played by objects and actions, it can develop a conceptual system (an organization of objects and actions into conceptual classes) which defines the observed experiences.

- **Addressing semantic consistency between robot and domain**. In multi-robot systems, each robot plays a role in the context of an external domain, so a mechanism to address semantic consistency between robot and the domain is

[1] Assigned agents have an associated simulation function representing the expected behavior using the specified environment descriptors for respective subsystem/component

[2] A system which acquires and maintains meanings and experiences itself. For instance humans learn meanings of states and representations, and keep on refining them themselves throughout their lives.

[3] For instance, series of sensor vector.

needed. Traditional approaches incorporate domain knowledge directly into robot software and system design. Such limitations were addressed by [77], who proposed an approach for semantic integration based on a persona structure [77]. A persona in essence refers to a robot ontology which represents roles, goals, capabilities, limitations of robot, thus it represents the public information about a robot. The proposed methodology in [77] proposed a usage of domain adapters for dynamic mapping of persona structure (public information about robot) and ontology of a given domain.

- **Developing contextual model for real world**. Intelligent adaptive behavior based on a certain context depends mainly upon the understanding of real world. There can be lots of ways for developing contextual models for real world but the ontological approach is useful in terms of interoperability between context models. Intelligent behavior of autonomous entities, robots, depends upon the management of meaningful relationship between the environment entities and context models. In literature ontology based approaches to address the grounding problem have focused on feature based and context based object categorizations. A methodology proposed in [81] is among the ones that tried to improve the robot context understanding by using a machine-understandable representation of objects and their features in the form of object ontology. They [81] considered feature-based perspective of environment, objects and their categorization and classification of features according to intended functions and usage information. Such classification followed by a form-function reasoning[4] results in association among groups of objects commonly used together. Such association between objects and robot activities provides a better context understanding of robot. In a similar research agenda, an approach based on ontologies and a related reasoning mechanism to address the grounding problems was proposed in [82]. It was based on the proposition given in [83] which showed the categorization as an essential element for approaching the grounding problem. Approach in [82] proposed OBOC (Ontology based categorization system) implemented on AIBO[5] to enhance its object recognition and communication capabilities. OBOC system provided the following features:

    - Object categorization based on properties/features recognized by system
    - Object categorization based on context
    - Concept creation and learning of concepts from other ontologies and robots on the fly

- **Rational robotic reasoning on the basis of ontological reasoning.** One of the uses of ontologies for robotic agents is rational agent reasoning based on ontological reasoning. In this aspect approach given in [85] tried to use the ontology based reasoning to augment the vehicle path planner to make inference about the costs and consequences of colliding with other objects. The proposed

[4] To deduce functional elements of a composite environment entity.
[5] AIBO is an autonomous robot in the form of a dog. See www.aibo.com for details.

solution advocated the use of ontologies in enhancing the capabilities and performance of autonomous vehicles. Based on the inference aspect of [85], [86] proposed ontological representation of the robot capabilities and tasks. They performed experimental validation of (inferred) task execution with real robots. They considered the use of ontology for the description of locomotion subsystem of mobile robots and tasks a robot needs to address.

In [86] ontological inference was employed to find the suitable mobile robot based on task and robot descriptions. The major contribution of the work is in knowledge representation technologies to robotic agents inhabiting in physical environments. Their experimental scenario consisted of two mobile robots and a coordinator computer. The task execution in their proposed approach proceeds in two phases.

1. *Initialization*; each robot shares the capability model expressed as ontology
2. *Task execution*; given a task description (expressed as task ontology), a suitable mobile robot is selected and roles are assigned to robot.

In summary, given a task, the coordinator performs the inference on the capability model of mobile robots to find the suitable robot for task execution. After selection of suitable robot the coordinator dispatches a sequence of instructions to each mobile robot to assign their task roles.

- **Comparative evaluation and normalization of robotic platforms.** In order to focus on comparative evaluation and selection of optimal platforms in rescue support activities, [87] emphasized the need for a standard framework for representing different aspects of robotic platform. Their approach proposed three basic aspects which are used to define a system, these aspects are:

1. *Physical entity*; that describes structural description of a system. For instance, locomotion mechanism, sensory and computational component etc.
2. *Functionality*; that defines what a system can manifest using its structural components. This includes description of activities, capabilities and limitations of system etc.
3. *Interaction Element*; describes how a system relates to the world outside its boundary for performing meaningful interactions with other functional entities. This includes communication protocol, media and content etc.

In a very similar effort, [88] presented a knowledge presentation that captured the relevant information about robots and their capabilities to assist in development, testing and certification of technologies as sensing, mobility, planning, operator interaction etc. Their developed robot ontology for the urban search and rescue followed a similar categorization as [87], which included:

- *Structural characteristics.* These included size, weight, power source etc.

- *Functional capabilities.* These included sensory, locomotion and communication capabilities etc.)
- *Operational considerations.* These included intra and inter-group interaction and human factors etc.

- **Contextual constraint management in robots**. Embedding of real-time constraints into the robotic system as hard coded into reactive behaviors of robot or into deliberative layers makes it difficult to understand and maintain the constraints. Such gets exacerbated in heterogeneous multi-robot environments. Similarly there is also a need to cope with context augmentation to constraint adjustment. Towards these aspects, [78] tried to address these problems of contextual constraint management in a flexible and transparent manner by using semantic policy[6] services. They considered mainly the authorization (i.e. which actions an actor is allowed/disallowed in a certain context) and obligation policies (i.e. which actions are required to be performed by actors in a given context). For instance an obligation policy in the context of robot movement could be; "Actors who are, robots, are obligated to beep before moving".

  The proposed policy services based solution was demonstrated in the context of NASA human-robot teamwork (HRT). Such policy based approach provides flexibility in dynamic regulation of a system behavior. But it is not appropriate for low level control with real time constraints. Such policy mechanisms can serve as a supplement for robot control for effective human-machine interaction and context based robotic inference based on ontologies.

- **Robotic task modeling ontology.** In the context of robotic task modeling ontology, [80] proposed five types of ontology;

  1. *Device ontology*; classification of devices for target domains
  2. *Function ontology*; identification of functions of each device
  3. *Operation ontology*; identification of operation steps in function ontology
  4. *Manual ontology*; conceptual explanation of each device
  5. *Conditional ontology*; inter device dependencies and device environment dependencies

  Task modeling and allocation in [80] referred to these ontologies and used those in task modeling of home appliance network. The essence was the conceptualization of available resources in terms of functionalities and inter-dependence to incorporate complex task activities.

In summary ontological representation has been used in flair of research agendas for robotics. This comprises context modeling (object and environment ontologies), ontological reasoning based task execution/planning, robotic ontologies for development,

[6] An enforceable, well-specified constraints on the performance of machine-executable action by an actor in a given situation

test and validation of robotic platforms. Overall development of a conceptual organization of environment and robot capabilities contributes significantly for development, test and validation of intelligent autonomous systems.

2.5. Distributed learning in the absence of global knowledge

As noted in design aspects of distributed system there is an absence of a global state in distributed systems. In a non-distributed system, global state is typically defined and maintained in terms of finite state machines (FSM). On the other hand in distributed settings, local decision makings should propagate higher to achieve global decision making or system-wide goals. In decentralized systems such as robot swarms, having a universal global knowledge is too restrictive and newer domains are the norms of computational environments. Porting distributed and collaborative intelligence and inference capabilities in robot swarms or, in general, in distributed systems is beneficial. Towards distributed learning this thesis follows a rule based learning approach and uses defeasible logic for incremental learning of robots. Incremental learning is based on epistemic cognition in human beings and formal concept analysis is used as a case for incremental learning of formal concepts in robots.

Developed formalism and algorithm for intelligent robotic navigation is based on an algorithm of partial lattice generation i.e. incomplete information without *a priori* knowledge of a complete global location model. Apart from that, unlike existing models, presented methodology can easily incorporate semantic information, and expressing such in terms of rule updates, provides a very flexible way of updating robotic control platform. The use of defeasible rules and use of meta-data/contextual information of facts and rules for intelligent robotic navigation with situational context as robotic permissions, room status and scheduled navigation etc. provide intelligence and adaptive nature to robotic location models.

2.6. Concluding remarks

This chapter motivated interoperability and distributed learning towards distributed knowledge management. Motivation was complemented with existing approaches and solutions in literature. This chapter also elaborated a representative example of swarm of mobile autonomous robots and overviewed of interoperability solutions based on Web services technology.

In the following chapters, a detailed design, discussion and evaluation of Web services based interoperability solution and distributed learning for dynamic heterogeneous robot swarms is explained.

Part I

Interoperability in Mobile Heterogeneous Systems

Decentralized Web services Middleware

3. Decentralized Web Services and Limitations of UDDI

One of the most compelling arguments of SOA (Service-Oriented Architecture) is the reuse and sharing of services and resources between different entities. Web services (a SOA manifestation) achieve these objectives using clearly defined standards of WSDL (Web services Description Language), SOAP (Simple Object Access Protocol) and UDDI (Universal Description Discovery and Integration). Among these standards UDDI [54] provides mechanisms for Web services publishing and discovery using a centralized registry; a directory for publication and discovery of categorized Web services.

This chapter highlights the limitations of UDDI towards decentralized Web services and overviews existing approaches in distributed UDDI registries and P2P (peer to peer) solutions of UDDI repositories and in the end summarizes with gathered requirements/guidelines for establishing “mediator based communication middleware architecture” for decentralized Web services.

3.1 Web services and its component technologies

Web services is a technology that promotes creation of open distributed systems. Using standard protocols like XML and SOAP, Web services is quickly gaining popularity in making distributed applications. Web services is a SOA (Service Oriented Architecture) manifestation and achieves the objectives of reuse and sharing of services and resources between different entities using clearly defined standards of WSDL (Web services Description Language), SOAP (Simple Object Access Protocol) and UDDI (Universal Description Discovery and Integration). In Web services perspective, an application can be built by integrating multiple services together resulting in the creation of a more complex service. In order to integrate the services, there is a need to locate and acquire specified services. For such purposes UDDI compliments Web services by providing the functionality to search for Web services located on different sites.

Existing UDDI technology uses central server/repository to store pointers to registered Web services which may be located elsewhere. Using a centralized approach has many drawbacks, which are discussed in this chapter. Though, this research doesn’t cover all limitations, but limitations of current research work and a guideline for creating an ideal UDDI are given in a broader sense.

3.1.1 SOAP (Web services communication standard)

From a communication point of view, SOAP is a *de facto* standard in Web services communication. SOAP has been designed for a client server type of interaction which is typically implemented as RPC (Remote Procedure Call) or some other variations thereof. The advantages of SOAP arise from its ability to provide a universal vehicle for conveying information across heterogeneous middleware platforms, applications and entities. Adherence of SOAP to client server model causes its limitations, such as:

- Coupling of data exchange to parameters in a method invocations and
- rigid interaction patterns (referred to as Message Exchange Patterns - *MEP*) those are highly synchronous.

With the recent trend in utilizing P2P technology and applicability of Web services for heterogeneous communicating entities, synchronous interaction patterns and coupling of SOAP with method invocation are bottlenecks, which are addressed in this thesis.

A simple SOAP - RPC based conversation can be given as:

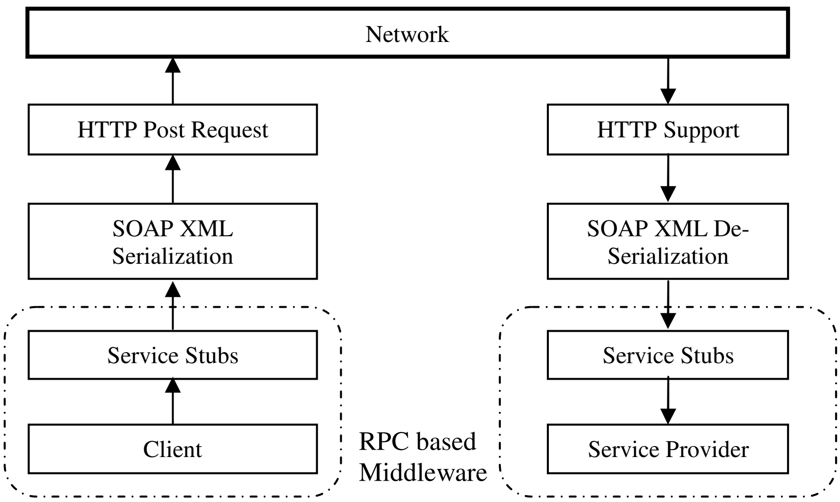


Figure 2: SOAP – RPC based conversation

Asynchrony and message queuing requirements

Generally message exchange patterns in Web services are used to implement a synchronous client/server message exchange as shown in Figure 2, which is just a particular case of more complex message exchange patterns. SOAP messages, however, can also be used as part of asynchronous interactions between a set of peers or entities as shown in Figure 3.

Furthermore, by using techniques developed as part of traditional Message Oriented Middleware - *MOM*, asynchronous messaging can be built on top of synchronous interactions, by introducing a queuing system for storing and forwarding of messages. This thesis refers to such queuing system as message post boxes as shown in Figure 4.

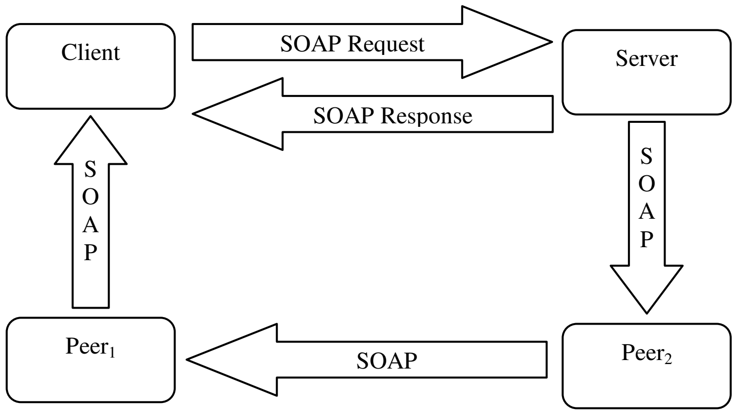


Figure 3: SOAP in non RPC Messaging P2P Environment

3.1.2 WSDL (Web services description standard)

Web Services Description Language (WSDL) provides a specification and an XML format for describing Web services. WSDL enables a separation of description of the abstract functionality offered by Web services from the concrete details of a Web services such as *how* and *where* that Web services functionality is offered. A WSDL document describes Web services using the following major elements:

WSDL Ports: The <*portType*> element is the most important WSDL element, which describes Web services operations that can be performed and the messages that are involved in those operations. The <*portType*> element can be compared to a function library in a traditional programming language.

WSDL Messages: The <*message*> element defines the data elements of an operation. A message can comprise of one or more parts. The parts of a message can be compared to the parameters required by a function call in a traditional programming language.

WSDL Types: The <*types*> element defines the data types that are used/consumed by a Web services. WSDL uses XML Schema syntax to define its data types.

WSDL Bindings: The <binding> element defines the message format and transport protocol details for defined ports of Web services.

The main structure of a WSDL document looks like the following:

```
<definitions>
        <types>  Definition of types........ </types>
                <message>  Definition of a message.... </message>
                        <portType>  Definition of a port....... </portType>
                                <binding> Definition of a binding.... </binding>
</definitions>
```

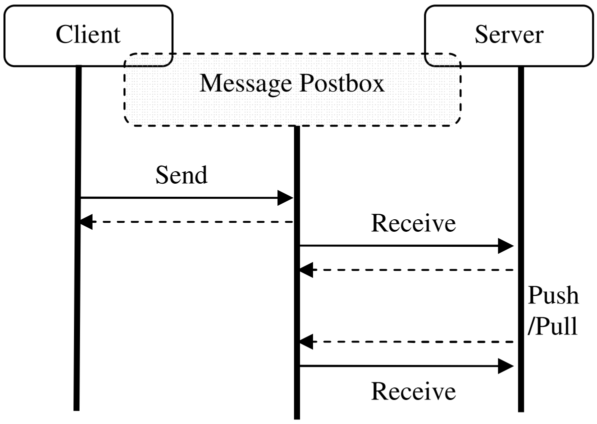


Figure 4: Utilization of SOAP in Message Oriented Middleware

3.1.3 UDDI (Web services repository standard)

UDDI provides mechanisms for Web services publishing and discovery using a centralized registry; a directory for publication and discovery of categorized Web services. UDDI was initially proposed as a standard to enable universal discovery of Web services. Web services registered at a UDDI repository can be searched and then remotely invoked. UDDI is the main pediment in the exploitation of Web services in distributed control and dynamic natured systems. The first prominent problem with UDDI is its basis on client/server model which suffers from a single point of failure. This can occur when a traditional UDDI system exclusively assumes the server role, in client-server architecture.

UDDI was a specification that was ahead of its time. It was designed to enable management of a large numbers of Web services, but at the time of its release (first specification); there were relatively few Web services to manage. But recently the number of Web services and its usage has progressed quite rapidly. UDDI was originally intended to serve as a USR (Universal Service Registry) to cater for problems of discovery and integration with Web services and Web services-based applications. The inherent design of UDDI was based on centralized resource management. The traditional centralized resource management frameworks have limitations both in their fault tolerance and scalability [97]. The biggest criticism of Web services from distributed computing standpoint is its reliance on centralized Web services repositories. UDDI, though, is the *de facto* standard for Web services discovery and invocation, but tight-replication requirements among Web services registries, and lack of autonomous control has severely hindered the widespread usage of UDDI [98]. On the contrary P2P (Peer-to-Peer) networks promise high availability of published content, optimized use of network resources, content distribution cost and scalability using decentralized architectures.

In general the limitations of UDDI can be listed as follows:

1. Centralized UDDI solution suffers from a single point of failure.
2. UDDI was intended to be used only for Web services discovery.

3. UDDI does not provide any guarantees to the validity and quality of information it contains. UDDI does not maintain or provide Web services lifecycle information. UDDI doesn't provide Quality of Service (QoS) measurements for registered Web services.
4. There is a disconnection between UDDI and the current Web.
5. UDDI Specification doesn't support caching of Web services descriptions.

3.2 Distributed UDDI registry approaches

Communication models based upon the client/server architecture comprise of a dedicated server that provides certain services to clients via specific communication protocols such as File Transfer Protocol (FTP), SimpleMail Transport Protocol (SMTP) etc. Most of the computations take place on the server side which leverages clients from heavy processing load. However servers can suffer from the limit of number of clients where the server is not capable of handling more requests.

Unlike the centralized approaches, the peer-to-peer (P2P) model highly supports the idea that any entity can assume either a client or a server role. In other words, at one time the entities can provide services to others entities (i.e. being in a server role) and at another time they can request services from others entities (i.e. being in a client role). Due to decentralized nature, there is no restriction on entities in a P2P system to be permanently available at all time. Entities can perform normal operations by requesting online entities to act on behalf of offline entities. P2P can, thus, maximize the use of resources from entities (peers) connected to a P2P network. It gives highly available services at the lower cost than the centralized approaches.

Based on P2P and distributed technology, there are a number of efforts which have tried to establish distributed UDDI. Two important dimensions in distributed UDDI registry are:

1. Federation of UDDI registries
2. Web services registries for Infrastructure-less Systems.

3.2.1 Federated UDDI Registries

Federation of registries is defined as a collection of autonomous but cooperating Web services registries. The goal of a federation is to form a registry community that serves a specific domain. A federation of registries also aims to form associations between various registries. The members of the registry federation can be heterogeneous and can have different data models, schemas and access APIs etc. Thus it provides a loose definition of constituent/member registries.

Federation of registries can provide several advantages over individual/centralized stand-alone registries. The benefits of a typical federated system include, but are not limited to the following:

- Managing distributed information infrastructure (i.e. sharing data, establishing inter-system data references and dependencies) without creating monolithic information aggregate.

- Accessing and integrating information from member registries without manual integration of information from member registries. Thus, achieving data model transparency.
- Providing a scalable approach for accommodating new registries to the existing applications.
- Providing QoS methods.

3.2.1.1 Research efforts in Federated UDDI Registries

Towards establishing a federation of UDDI registries into different (business) domains or communities, [101] proposed an approach for abstract meta-data based classification of community/domain registries with a single root/centralized registry for all communities/domains. Similarly [102] proposed a federated architecture for P2P Web services, in which a federation for UDDI-enabled peer registries is employed in a decentralized fashion. Publishing of Web services is done, in their approach, on a centralized UDDI peer registries and then peers join service syndication.

Similarly, an approach based on the notion of an active and distributed Web services registry (ad-UDDI) was proposed by [103]. ad-UDDI refers to an active monitoring mechanism enabled UDDI to maintain periodic Web services information. They considered a layered approach for distributed UDDI registry into a management root layer, a business layer comprising of domain specific ad-UDDI registries and a service layer.

Such federated approaches again suffer from a single point of failure, same as centralized approaches.

3.2.2 WS registries for Infrastructure-less systems

Towards infrastructure-less environments, [55] proposed a mechanism for dynamic management of distributed UDDI in the absence of wired communication infrastructure. They introduced the notion of Messenger (a mobile user with a user-agent) which, upon reaching the vicinity of some UDDI, updates the Web services descriptions it has cached already for user needs. Exchange of information depends upon ad-hoc nature of network and occurs when a user enters a new cell (a cell is defined as a network coverage area). Proposed environment of [55] suits best the requirements of poorly reliable and unpredictable coverage feature of communication infrastructure, but their approach did not consider the use of mediator entities [56] and moreover message routing aspects were not taken into consideration.

Similarly, towards synergizing P2P networks and Web services, [105] proposed a declarative dynamic composition and execution framework for Web services in P2P networks. But their work did not consider the Web services publishing and discovery in P2P networks, which is the primary aspect to overcome the scalability issues with centralized UDDI. To address the scalability issues with UDDI, [107] proposed DUDE (Distributed UDDI Deployment Engine). DUDE proposed the leveraging of structured DHT (Distributed Hash Table - a P2P system that forms a structured overlay, allowing more efficient routing than the underlying network) as a rendezvous mechanism between different registries. In their approach Web services description message dispersion to several distributed UDDI registries promoted scalability and replication. But their approach cannot cope with dynamism, mobility and scenarios where inter-communication between entities is not possible in a point-to-point fashion.

Similarly [106] proposed Web services architecture based on a P2P network (built on top of JXTA – Juxtapose [110]). They proposed a classification of peers on the basis of computation power and memory into Service- and Super-peers, where Super-peers are responsible for publishing of Web services, query routing and formation of peer-groups for Service-peers. Such architecture assumes homogenous communication capabilities of peers, which cannot be considered in heterogeneous environments. Moreover, as we discussed previously in our case environment of robot swarms, a full coverage of a robot/entity of rest of the network is not (should not be) assumed. Moreover in mobile environments, mobility of an entity/robot can restrict the communication coverage of an entity/robot.

3.3 Concluding remarks – missing bits in research

To summarize the missing bits in research, there is a missing gap in existing work for Infrastructure-less and mobile dynamic systems. Most of the existing work considers homogeneous communication capabilities of entities comprising a system and mobility aspects are generally overlooked.

In particular, following aspects have been overlooked in existing decentralized Web services solutions (decentralized UDDI solutions):

- Web services description dispersion in a distributed way.
- Dynamism of environments and mobility of entities.
- Coping with scenarios when entities are unable to communicate in a point-to-point fashion to other entities (need for usage of mediators).
- Partial coverage of entities of rest of the network/entities due to their constant mobility and constrained communication capabilities.
- Adaptive protocols or various levels of communication to cope with communication heterogeneity of entities.

This thesis focuses on these aspects and tries to overcome the limitations of existing work in the proposed "mediator based communication middleware architecture" for decentralized Web services. General/broader objectives of this branch of research are to create a flexible and open distributed knowledge environment for self-configurable, low-cost entities that operate in dynamic environments with no assumption about their communication capabilities.

As quoted earlier, one example of such environments is a community of robot swarms, which are attributed by heterogeneous communication capabilities (i.e. few robots can communicate wirelessly in a point-to-point fashion, while the others might require mediators for communication of their messages). Apart from communication heterogeneity the operational environment is attributed by dynamism (unreliable nature) i.e. during system execution no guarantee can be given about the successful communication between any two robots. Moreover, mobility contributes to partial coverage of robots of rest of network, or robots being out of communication coverage.

Key notes of proposed mediator based communication middleware are:

- Decentralized control
- Heterogeneity in knowledge representation
- Un-guaranteed communication, and
- Ability to operate heterogeneous communication capabilities of entities in a system

This thesis proposes a distributed data environment, based on intelligent inter-entity communication and a global database solution for knowledge reuse. Decentralization of control and local manipulation of entities contributes to scalability of the proposed architecture/middleware. Proposed solution exploits Web services standards for exposing entities' functionalities to outside world (thus abstracting each entity to a Web services interface). Proposed and developed middleware uses mediator-based communication for entities, which enables communication for those entities that are unable to communicate in point-to-point fashion to other entities. Conceptual classification of entities according to their communication capabilities enables the use of various levels and views of communication.

The following chapter elaborates the proposed middleware, which is evaluated in subsequent chapter.

4. Mediator based Decentralized Web Services Infrastructure

In dynamic environments for mobile autonomous systems such as robot swarms or mobile software agents there is a need for autonomic publishing and discovery of resources and just-in-time integration for on-the-fly service consumption without any a priori knowledge. Based on the limitations of UDDI as mentioned in previous chapter, this thesis proposes a mediator-based distributed Web services discovery and invocation middleware - a collaborative and decentralized Web services discovery and management middleware for infrastructure-less mobile dynamic systems with heterogeneous communication capabilities. To cope with heterogeneity of communication capabilities, a conceptual classification of computing entities on the basis of their communication capabilities and a conceptual overlay formation for query propagation in system is proposed.

This chapter discusses semantic architecture and communication support for infrastructure-less and mobile dynamic systems such as robot swarms [27].

4.1. Architecture of decentralized WS middleware

To support effective, flexible and loosely coupled system integration, there is a need for an architecture which provides loose coupling and semantics to deal with semantic heterogeneity among entities and systems (sub-system components). Interoperability between autonomous mobile systems relies on effective and seamless knowledge sharing and communication. In such systems each entity plays a specific role according to external environmental context [43, 44, 45, 46, 62 and 85], which results in different view sets of information and domain. Moreover, heterogeneity of entities with varying capability sets emphasizes further on semantic consistency [57] between entities and environment and among entities themselves. Furthermore, only by re-using external functionality [51] and symmetric invocation from outside world, openness of system can be achieved.

In order to support this, proposed and developed middleware employs standard solutions based on Web services technology. According to this view, entities are seen as Web services accessible from the environment. In other words, functionalities of entities are exposed as Web services and different applications may invoke such services whenever needed. Symmetrically, entities can also access external functionality or information.

The architecture, based on ontologies (semantics) and Web services, provides a simple and unified access to entities/system's functionality. However, poor computational and service discovery capabilities of resource-constrained entities (such as robots) require some sophisticated mechanism for Web services discovery and their provision from the entities.

General architecture of middleware is presented in Figure 5. Its main components are:

- **Global** (Robot) **Knowledge Base (GRKB)** stores ontologies, which bind together entities' internal data with external knowledge. The ontologies describe the main internally and externally used concepts together with relations between them. External knowledge is applied to describe semantically available external Web services. The latter simplifies location and orchestration of required Web services while automating aggregation of required data.
- Information in **Local** (Robot) **Knowledge Base (LRKB)** is used by entities' behaviors, which constantly write new internal knowledge and read externally acquired knowledge from there. LRKB is thus not only used as a communication channel between entities' behaviors, but also between entities' behaviors and external knowledge sources and Web services.
- **Communication Gateway (CG)** routes knowledge messages to entities or external Web services. Global servers can also access specific entities via CG-s. Thus CG works in both directions:

  - It locates entities where to send messages from servers;
  - It locates servers where to deliver messages by using configured network connections.

- The **Web services ecosystem** consists of Web services middleware together with an extensible set of Web services (together with their descriptions). It also includes some knowledge-based reasoning components such as Web services gateway.
- **Web Services Gateway (WSG)** is a part of Web services ecosystem. Its task is to discover, bind together and execute public Web services. The gateway also allows addressing specific entities as Web services. WSG manages its internal directory where semantically enriched Web services descriptions can be published, discovered, orchestrated and deployed. Descriptions of the objects are stored at GRKB.

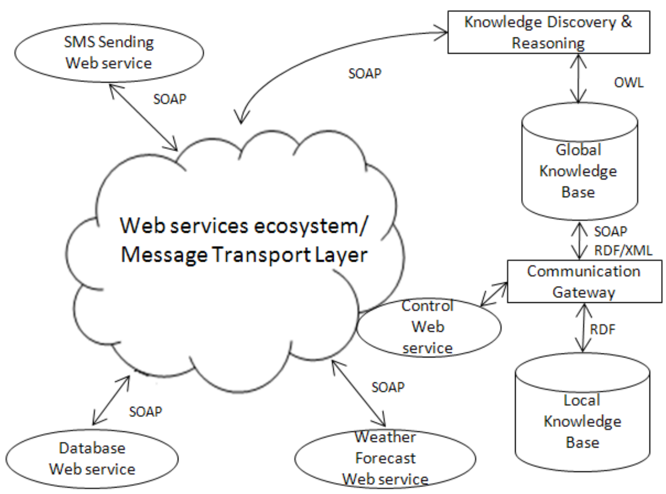


Figure 5: Semantic Architecture for Entity Interaction through WS

4.2. Mediator-based distributed WS middleware

Web services abstraction layer allows applications and entities to communicate in a programming language- and platform-agonistic manner and has developed into a mature solution for providing interoperability between heterogeneous computing entities. Peer services [108] and Web services are the most referred technologies for decentralized applications, which apparently address different problem domains yet have a significant conceptual overlap.

Web services provide a just-in-time integration solution for on-the-fly service consumption without any *a priori* knowledge of the service and procedures a consumer needs to reuse. But the biggest criticism of Web services from distributed computing standpoint is its reliance on centralized Web services repositories [109]. Peer services [108] on the other hand negate the notion of centralized repositories in SOA paradigm, though also leverage SOA but synergize producer and consumer as peers and solve the centralized nature of Web services by using cooperation of peers [110, 111]. Web services have clearly defined standards in form of WSDL and SOAP. While JXTA [110] considers a more abstract approach yet depending upon exchanging XML based messages.

4.2.1. Basic Philosophy

Proposed communication middleware considers the basic philosophy of JXTA (project juxtapose) [110], which is a P2P stack and XML-based set of peer platform protocols and fundamental services for discovery, presence management and communication between peers. XML-based protocols allow any device connected to network to exchange messages and collaborate independently of underlying network topology thus making XML-based JXTA transport- and language-agnostic. Apart from XML-based communication the introduction of unidirectional asynchronous communication channels (referred to as Pipes) is one of the basic technology shifts in JXTA. Asynchronous and non-deterministic nature of network contributes significantly towards high availability of published content, services, optimized use of network resources, content distribution cost and scalability.

Basic approach to communication middleware applies the communication principles from JXTA and uses the defined standards of WSDL, RDF (representation of ontology) and SOAP to achieve interoperability. This thesis proposes an abstract approach of communication capabilities based classification of computing entities. In principle there are different types of protocols depending upon the communication capabilities of computing entities.

As compared to JXTA, the proposed middleware not only relies on communication capabilities on entities, but also due to dynamic nature (caused by constant mobility of entities) it supports dynamic joining of entities in different groups during system execution.

4.2.2. Conceptual Classification of Entities

Rendezvous computing entities (RCE) are the computing entities with wireless connectivity or entities which can communicate outside a domain or with external world such as Internet. Rendezvous entities provide efficient dissemination of requests from one domain/group to another. Thus they can be considered as routing peers [112, 113 and 114]. In a way, rendezvous entities create a virtual overlay network with other rendezvous entities and serve as distributed points of centralization of network, serving entity-groups [106] at a lower level.

Edge computing entities (ECE) are considered those entities that don't have a capability to communicate in a point-to-point fashion with other computing entities and require some sort of information mediator/relay to communicate their messages.

Message Relays (MR) or information mediators are effectively entities that serve ECE to communicate their messages.

RFID tags could be one example of message relays, that could serve ECE (if ECE are only equipped with RFID read-write capability and no other point-to-point communication capability with other entities).

Another example is distributed knowledge bases; each knowledge base can be abstracted as a message relay for those software agents that can only communicate within domain of that knowledge base, while some software agents can have an additional capability to serve as a gateway and can communicate to other distributed knowledge bases by communicating with respective gateway agent.

RFID tags are seen as a special example of MR, which are being employed in numerous robotic environments. In general, RFID tags have limited memory, thus storing an XML based message is not a viable solution. For such memory-constrained MR, this thesis considers the communication via *extended RDF* block rows data model. Apart from typical (Subject, Predicate and Object) some additional fields such as time of data insertion/update, source of data, context field etc are considered.

4.2.3. Communication Middleware Architecture

Modular architecture of communication middleware (CMA) is depicted in Figure 8. Basic architectural elements of CMA are:

Message Transport Layer (MTL) is the core element of CMA which provides an implementation of asynchronous communication channels identified by endpoints, similarly to normal Web services invocation. Asynchronous nature of MTL serves well for dynamic environments and towards creating a self-organized federation of entities, where entities and associated services can join and leave at any point of system execution, thus achieving loose coupling of entities. An entity can logically bind itself to any MTL of a destination entity, which can bind itself further to subsequent destination MTL(s), thus creating a chained communication overlay network.

Local Service Registry (LSR) serves as a local cache for Web services discovery. For LSR, this thesis, implements a light-weight UDDI.

An incoming Web services request is first searched from LSR before being cached to **Query Response Cache (QRC)**. QRC caches all incoming Web services requests and propagates back response messages. Because of back-propagation of query response in reverse order, QRC helps to avoid sending multiple Web services requests/responses to a particular entity and upon discovery back propagation it stores the query response in LSR.

Entity Discovery Registry (EDR) serves as a record for discovered entities. EDR is used to create a semantic topology based on set of Web services an entity advertises, its *expertise* (semantics of Web services). EDR serves for semantic query propagation.

Ontology repository stores entities' Ontologies in form of RDF triples which are used for mapping Web services discovery queries to published Web services.

Local (Robot) **Knowledge Base (LRKB)** serves as a knowledge base of factual information, which an entity might gather during its execution cycle.

4.2.4. Semantic Topology

In such an abstraction to P2P architecture, each entity operates with identical functionality. Such an abstraction features redundancy, dynamic selection of entities, and fault-tolerance. Despite these benefits, the inherent difficulties related to P2P technology, which need to be solved, are propagation of query to an appropriate entity and efficient routing of messages.

Several researchers [112, 113 and 114] have tried to solve the efficient dissemination of messages from one entity to another while Berners-Lee [7] advocated on the enrichment of information with well-defined meanings to enable computing entities to collaborate by expressing the knowledge in a well-defined formal way. Notion of RCE, RPV and dissemination of messages from one entity-group to another provides quick dissemination of messages across entity-groups, while MR performs dissemination of messages within an entity-group.

To incorporate semantics and to perform semantic-based query propagation to appropriate entities, a model of expertise is used, in which entities publish their *expertise* in the network along with their exposed functionalities as Web services.

The knowledge entities have about the expertise of other entities forms a semantic topology.

Basic philosophy of creating a semantic topology is to re-route the query to an entity which is likely to answer to query instead of broadcasting or sending the query to random entities. Proposed solution uses a shared common ontology *O* of computing environment, which provides a shared conceptualization of domain. Entities publish their expertise *e* in the network. Knowledge of an entity about the expertise of other entities forms a semantic topology regardless of the underlying network topology. In other words an entity knows another entity in this semantic topology if it can compute the semantic distance of query to its published expertise. Semantic topology or entity's knowledge of other entities can be expressed as:

$Entity_1$ knows $Entity_2$, if $Entity_1$ knows the expertise of $Entity_2$

Shared ontology of environment/domain is expressed as RDF triples in *Ontology repository* for describing the functional expertise of entities. As elaborated earlier, entities advertise their Web services as advertisements, when coupled with their set of expertise; it provides a decentralized solution to advertise Web services.

Ontology repository is transformed to a set of facts and rules in LRKB and defeasible reasoner is applied afterwards. Currently a light-weight implementation of defeasible reasoner is used for formal concept/ontology learning/evolution. Similarity function is expressed as a set of rules in defeasible logic as well. The purpose of developing a rich set of rules for communication, concept learning and associated semantic similarity is to provide a rule-based expert system, using which both communication and reasoning capabilities of entities can be made intelligent. A detailed discussion is given in next section and defeasible reasoning and reasoner are detailed in Chapter 6.

4.2.5. Entity & WS Registration

Upon a system bootstrap, entities (RCE and ECE) are not aware of any other information apart from their LSR. LSR contains their own published Web services. Similarly their EDR is empty as well. Entities can navigate in the environment as programmed by their control application. Even though RCE can communicate in a point-to-point fashion with other RCE, a discovery mechanism is needed. For this purpose entities perform their registration (i.e. indication to environment about their presence, their published Web services and their expertise in the domain)

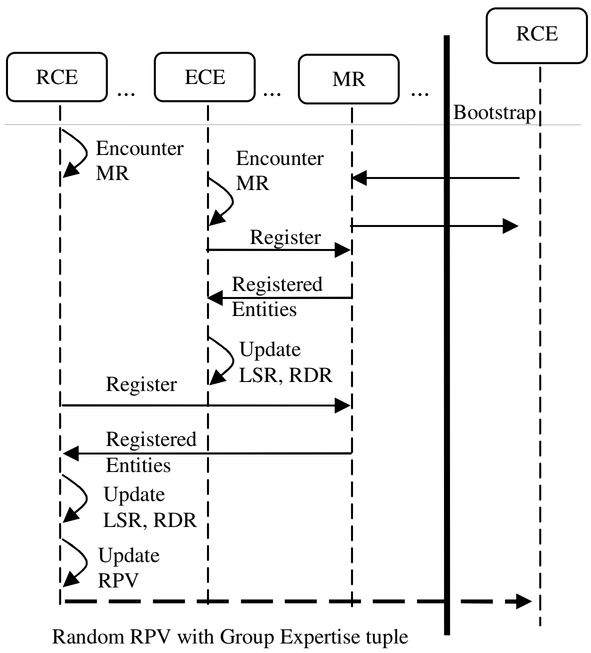


Figure 6: Entity Registration and Overlay formation

Entities (ECE and RCE) upon encountering a MR, register themselves by expressing their presence information using MR's data model (in case of RFID using an extended RDF data model, but such is independent of underlying data model). Registration also includes advertising Web services descriptions and a set of entity's expertise (in principle each published Web services can be augmented by an expertise). For instance expertise can be *Weather Information Services*, *Cleaning Services* etc. depending upon the kind of entity and services entity might provide (representative ontology example is given in Chapter 5).

Thus registration marks MR as an information container (similar to *Shared Spaces* [115] but a non-persistent information container) for services and entities surrounding it. Latter serves for discovery of entities, which occurs by entities receiving a list of other registered entities from the MR. This is a naïve epidemic way of discovering other entities (populating EDR) in the vicinity via MR scattered in the environment.

This thesis refer to entities around MR as *entity-group* i.e. all entities in a group can have pair-wise communication among each other. This pair-wise link can be either in a point-to-point fashion or via MR. Thus; MR can be used for dissemination of message within an entity-group. A number of entity-groups can exist in the system depending upon the distributed of MR. RCE being able to communicate wirelessly perform inter-entity-group message dissemination. Such is done by a *View Management protocol* (referred to as *Rendezvous Peer View* (RPV) protocol.

When RCE discovers by another RCE, they initiate RPV to establish a *Rendezvous View (RV)*. RV is a list of entities which are currently acting as RCE. RPV serves for creating finger table pointers among RCE for inter entity-group discovery and communication.

RPV protocol enables RCE to organize them into a virtual overlay; a higher level communication links between RCE for inter-entity-group communication). For RPV each RCE maintains a local list of active RV and periodically exchanges random RV to other RCE in its RV. In a similar fashion each rendezvous entity sends a periodic RV and purges non-RCE from its RV.

Thus effectively each RCE maintains an entity-group around a MR and serves for dissemination of messages from one entity-group to another using the Internet communication by using its RV. MR at the same time is responsible for dissemination of messages inside a particular entity-group. The registration process is illustrated in Figure 6.

4.2.6. Active & Passive communication

In term of communication, entities follow two modes of communication.

1. **Active Communication Mode**
   - Web services descriptions are pushed to other entities/MR.
   - Active mode corresponds to a normal Web services publishing to UDDI.
   - Web services descriptions are first pushed inside a cluster/entity-group via MR and later RCE disseminates them to rest of network.

2. **Passive Communication Mode**
   - In passive communication mode a Web services discovery occurs when an entity's

(RCE or ECE) request is answered by some entity/MR. In other words entities don't publish their Web services descriptions unless requested.
  - Passive communication mode is less bandwidth intensive than active communication mode.

MR resembles to Shared Spaces [115, 116] but instead of using a persistent information container MR is considered as non-persistent communication white-boards, which are prone to conflicts when multiple entities try to access it simultaneously. For conflict avoidance entities also employ a resource locking mechanism for conflict avoidance.

- **Conflicting mode** refers to a greedy mode in which entities don't wait for another entity to release MR.

- **Conflict-resolution mode** refers to a mode in which an active push or a passive Web services request locks the MR which is released upon either
  - Passing of Web services description or message request to another entity (i.e. at-least one entity has read the initiator's message). Or
  - Time-out

Due to constant mobility (and entities being out of communication range/coverage), locks to MR are not constrained by all entities reading a description or request.

4.2.7. Entity selection via Semantic Topology

When an entity receives a query, it can either refer the requesting entity to an entity that has the expertise similar to the query, or it can discard the query if the target entity doesn't have knowledge about the suitable entity for such a query.

The recommendation/referral is based on matching the query's subject *S* to known expertise *E* in the network. Subject *S* is an abstraction of query expressed in terms of shared ontology of environment in other words it's the expertise queried by the requesting entity. Recommendation mechanism relies on the semantic similarity function used [117]. In principle any sophisticated semantic similarity function can be used. As entities store shared domain Ontology in *Ontology Repository* and EDR is used for storing discovered entities and their published expertise. Semantic topology is created by using parent-child relationship.

$$SemDistance\,(s, e) = exists\_parent\_child\_rel\,(s, e) \rightarrow [0, 1]$$

Value 0 implies that *s* and *e* do not have a parent child relationship and 1 implies otherwise. Using such a semantic similarity function comes from the observation of [118]. They proved that number of edges/links separating two concepts is a metric for measuring the conceptual distance between any two concepts. But using just a "Parent Child Relationship" is a very restrictive similarity function but any similarity function *SemDistance* → *[0, 1]* can be used, where an increasing value indicates increasing similarity between any two concepts.

Based on the *SemDistance* value, an entity can recommend a set of entities whose expertise similarity value with subject is 1. For a more general similarity function, a more sophisticated entity selection algorithm can be used for ranking/recommending a set of entities. Ongoing work is trying to incorporate a more general similarity function [117] for mapping subject of query to Web services expertise description.

As elaborated earlier about *entity-group* (a group of entities around a MR) and role of RCE in dissemination of messages from one entity-group to another. For dissemination of expertise of entity-group RCE maintains the expertise of the entity-group (referred to as *expertise tuple* which comprises of expertise of entities in entity-group). *Expertise tuple* is built-up by the registered entities' expertise set at a particular MR. Expertise tuple are shared among RCE using RPV protocol (as elaborated earlier) thus achieving a view of expertise of various entity-groups in the system.

4.2.8. WS caching

Middleware considers the following policy for Web services caching

- RCE maintains all Web services descriptions and expertise tuple associated with an *entity-group*.
- ECE maintains only the Web services descriptions from those entities whose expertise are at a semantic distance of a certain threshold.

RCE registry hierarchy can be viewed as a federation of Web services registries for each *entity-group*, where RCE Web services registry serves as a composite domain registry. Though RCE registry can mimic the Web services an entity-group might offer however, high dynamism, lack of entity coverage (limited communication capabilities) and mobility renders RCE unable to give a consistent view of available Web services. Web services descriptions of an entity-group are not advertised by RCE across other entity-groups, instead RCE uses RPV protocol to share the expertise tuple of its respective entity-group with another RCE.

ECE selective Web services caching allows only the Web services descriptions from those entities whose expertise are at a semantic distance of at most 1 from its own expertise set.

4.2.9. Distributed WS discovery

The process of discovering Web services that satisfy user needs is referred to as Web services discovery. There are a number of Web services discovery models, broadly they can be classified as:

- **Static Model:** local storage of Static Web services descriptions
- **Centralized Model:** storage of Web services descriptions in a centralized Web services directory, such is also referred to as join lookup (mechanism employed in Sun – Jini [119])
- **Decentralized Model:** Web services descriptions are stored in dynamic and distributed Web services directories which employ distributed technologies such as federated approaches [101, 102], Peer-to-Peer [120] and agent based techniques [121].

In previous sub-sections entity classification on the basis of communication capabilities of entities, their registration and expertise publication using message relays was proposed. Rendezvous entities are used for efficient dissemination of messages across dynamic entity-groups leveraging dynamic join and leave of entities due to mobility and dynamic nature of environment and entities.

In proposed and developed architecture each entity servers as a small scale decentralized UDDI, using its *Local Service Registry (LSR)*; a local cache for Web services discovery. Decentralized design of Web services publishing and discovery is more scalable and reduces the overhead of centralized updates. Moreover, it provides higher degree of fault-tolerance as compared to existing centralized architecture of UDDI [97]. In fact, main popular use of centralized UDDI has been in enterprise systems, which suffer from scalability and consistency issues and the combination of UDDI, WSDL and coarse-grained business entity descriptions are generally used for automated Web services discovery and invocation.

In proposed middleware each entity implements a light-weight Web services registry (see Figure 7 – here the Web services registry uses the terminology as used for our case of robot swarms, however the Web services registry structure is general and can be used for any other domain as well). Current implementation of Web services registry doesn't consider the semantic information of inputs/outputs, and operations etc. Current discovery mechanism is not powerful enough to perform dynamic Web services discovery and invocation as semantics are ignored during the discovery process and in the current implementation of MTL the discovery is done on the basis of keywords and expertise of Web services using a shared ontology of environment. Thus there is a need for formal specification of semantics for Web services description, which are being addressed in ongoing work.

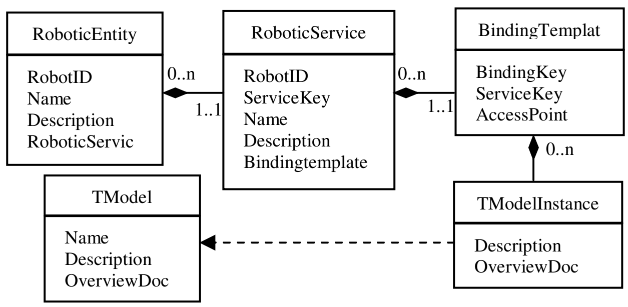


Figure 7: Web services registry

As elaborated earlier (see Figure 6), ECE and RCE publish their Web services descriptions and sets of expertise via MR. For this purpose *Web Services Description Language* (W3C - WSDL 2.0 2006) document is mapped to a suitable data-model for MR. If MR can handle extensive XML based WSDL documents then such transformation is not necessary. Similar argument can be given for SOAP messages encoding as well.

Using a hash function to store Web services descriptions and a rendezvous mechanism between multiple distributed registries as proposed by [120] is not possible due dynamic and partial view of network, thus, such mechanisms cannot be leveraged for ECE or RCE.

Approach for Web services querying/discovery (for passive communication mode) is as following:

1. Entities (ECE and RCE) publish their Web services request or expertise required at respective MR. These requests can be read by any other entity in the locality of that specific MR.
2. When an entity receives a query, it first checks its LSC. If it can find a mapping of query's subject with expertise of some entity from its LSC, then it will provide a recommendation as per its LSC about the appropriate target entity, i.e. entity will advertise about the entity which has an expertise which is semantically similar to query's subject (see Figure 8).
3. The requesting entity can take following action:
    1. Entity can re-query after discovering of availability of a semantically similar expertise in the network (if an exact match is not available in network)
    2. Entity can trigger discovery protocol to query the service descriptions of the entity which has been recommended.
4. When after a certain time threshold same the request remains unanswered or is published again at same MR, RCE can forwards the query to another RCE upon matching the subject of query with expertise tuples of RCEs in its RV.

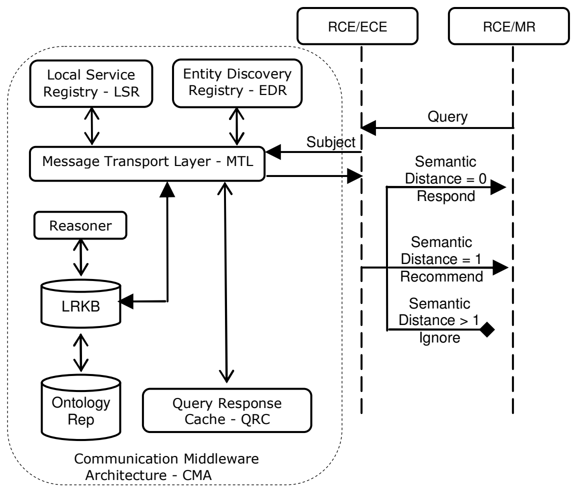


Figure 8: CMA and Semantic Distance based Query Recommendation

Approach for Web services querying for active communication mode remains the same, however entities publish their Web services descriptions at the time of registration. Afterwards the Web services discovery process proceeds as elaborated above for passive mode of communication. Detailed experimental analysis of active vs. passive mode of communication is given in Chapter 5.

Few aspects of middleware are needed to be highlighted with a resembling work on Shared Spaces [115] and a proposed architecture by [116] based on Shared Spaces. (Figure 9 and 10).

- **Persistence of Information:**
  - Shared Spaces [115] provide an application to read and write information in both synchronous and asynchronous communication in a *persistent* way.
  - Notion of MR as considered in middleware allows communication in both synchronous and asynchronous communication without any persistence. Middleware doesn't assume any notion of infinite storage capability, which leads us further to degree of guaranteed communication.

- **Guaranteed Communication:**
  - Inherent dynamism, mobility and infrastructure-less system paradigm coupled with limited storage capabilities of MR infer that no guarantee can be given about a successful communication or a Web services discovery. Such scenario can be compared, though not fully understood by considering a limited shared space with more than sufficient subscribers.

- **Loose Coupling**
  - Loose coupling can be achieved by any application using a virtual space, though the middleware provides loose coupling both in terms of space and time.

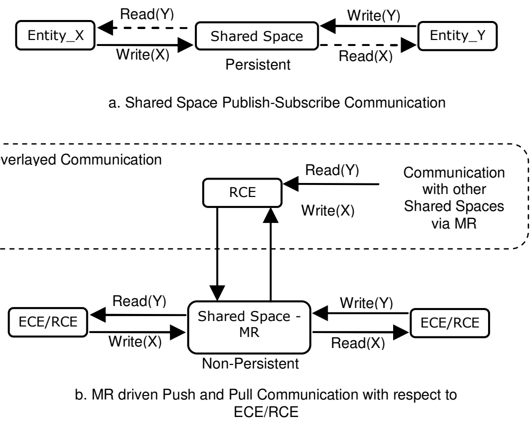


Figure 9: Shared Space vs. Mediator/MR based communication

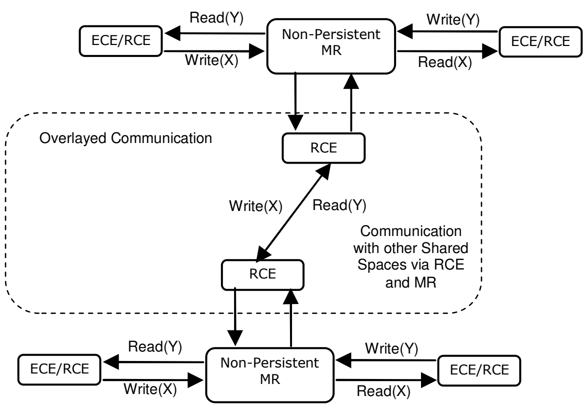


Figure 10: Mediator/MR and Overlayed communication in CMA

4.3. Concluding remarks

In this chapter a semantic middleware is proposed, which provides loose coupling between heterogeneous autonomous entities, and provides mechanisms for distributed Web services discovery and invocation. Proposed middleware overcomes the shortcomings of availability of both Web services requesters and Web services repositories at the same time. Developed mediator-based distributed Web services discovery and management middleware, though tested and deployed for robot swarms, can be applied to any infrastructure-less dynamic environment. Middleware abstracts dynamic environment and computing entities to a P2P system and a conceptual classification of entities according to their communication capabilities is used. It is assumed in the middleware that the physical environment has some type of mediator entities, which can be used for communication when point-to-point communication between entities is not possible. In implementation of middleware the discovery of Web services is done on the basis of keywords and expertise of an entity is used for efficient entity selection for query propagation. But there is a need for formal specification of semantics for Web services description as well. This issue is being addressed in ongoing work.

The following chapter will discuss in detail the evaluation of proposed middleware.

5. Case study and Evaluation

This chapter discusses the experimental results of CMA in a robot swarm setting. It first elaborates the evaluation system setting followed by evaluation results. Results are analyzed in active and passive mode of communication with and without conflict resolution. This thesis analyzes discoverable services with respect to time, services in complete view of cluster, impact and improvement of caching and semantics in service discovery results. In the end this chapter concludes and mentions noteworthy scientific contributions of research.

5.1. System evaluation setting

This chapter elaborates the results and evaluation of concepts and principles described in chapter 3 and chapter 4 as applied in the ROBOSWARM project [27]. The motivation of ROBOSWARM project is to develop an open knowledge environment for self-configurable, low-cost and robust robot swarms operating in dynamic environments. The swarms of robots are attributed by heterogeneous communication capabilities - i.e. few robots may be equipped with a wireless capability, while the other robots, in the absence of wireless capability, have to use RFID tags for reading/writing data.

This project intends to create and maintain a distributed data environment, based on intelligent inter-robot communication, and a global robot database solution for knowledge reuse. Thus communication interoperability of robot swarms both with environment entities and outside world is needed. Web services are used for exposing robotic functionality and abstracting each robot to a Web services interface. Web services based robot control platform using RFID tags as information mediators has been proposed previously as well [56].

Apart from LRKB of each robot, middleware also consider a GRKB which refers to collective swarm knowledge [122]. GRKB runs on a server, equipped with wireless connectivity, which can perform higher level reasoning and can provide resource-intensive services to swarm entities. Also it can serve as a gateway entity to swarm while communicating with outside world. In principle any swarm entity with wireless connectivity can serve as swarm gateway.

In ROBOSWARM project RFID tags are considered as message mailboxes/relays that serve to integrate distributed information and knowledge in computing environment. In ROBOSWARM, RFID tags with a small internal memory are used. RFID tags enable both human operators and robots to read/write information to the chips. In general, data on RFID can contain control information from humans, robotic presence information, facts and knowledge which a robot intends to share with other computing entities, local or relative location information of tags, service request and response messages.

Data is encoded as extended RDF block rows in RFID. Thus all data items written to the RFID are essentially data rows with predefined fields. It resembles to a small database table. RFID data I/O API designed for ROBOSWARM allows RFID data manipulation which provides further flexibility of treating any device conforming to data API to serve as mailbox/relay. Data model for RFID corresponds to extended RDF [123]. Apart from typical (Subject, Predicate and Object) some additional fields as time of data insertion/update, source of data, context field etc are considered. Though data encoding is specific to ROBOSWARM requirements but any data encoding scheme at RFID can be used.

Using the entity classification as proposed earlier, robots are classified on the basis of their communication capabilities. Apart from robots, there are a number of RFID tags in the operational environment which serve as information mediators. The swarm entities are classified as following:

- **Rendezvous Robots (RCE)**: The robots with wireless connectivity,
- **Edge Robots (ECE)**: The robots that do not have a wireless connection and require some relay entity to relay their Web services request(s) or Web services response(s) or user-level messages,
- **Message Relays (MR)**: Effectively RFID tags. Extended RDF data model is used for encoding data for RFID tags.

Operational scenario of ROBOSWARM with entity classification is shown in Figure 11. Control messages from User Interface can be disseminated into swarm either by using wireless interface provided by rendezvous robots or by writing the RDF encoded message to RFID tags. Similarly GRKB can interface to either RFID tags or to rendezvous robots. Rendezvous robots create an overlay network and run RPV protocol to maintain a RPV and robot-group (entity-group) expertise tuple. Lower level communication between edge robots and RFID tags comprises of Web services advertisements, expertise advertisements, request and response message and miscellaneous control messages. This lower level communication uses extended RDF data model.

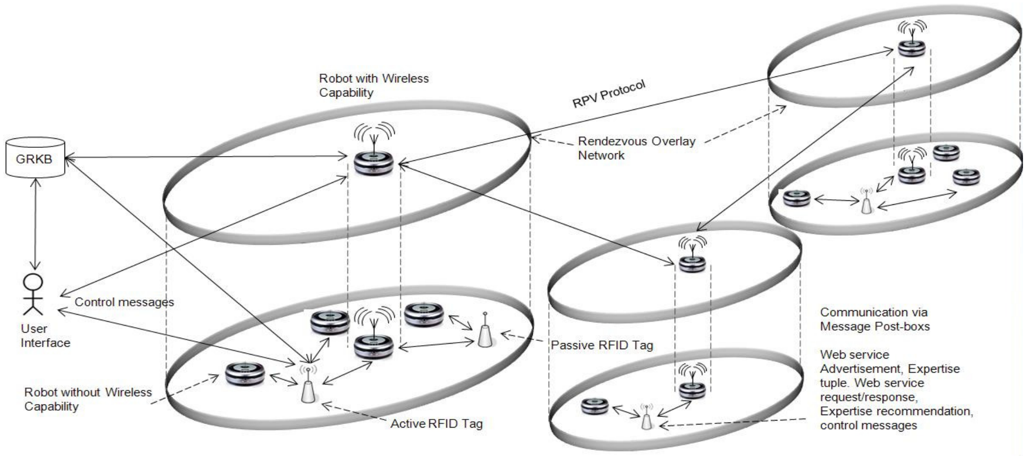


Figure 11: RCE overlay formation and communication between ECE & MR

First consider a case with only edge robots. During registration each edge robot presents its name and Web services provided. Because of limited memory in RFID tags, only a portion of

Web services descriptions is presented at a RFID tag. Latter on other robots read the information posted on RFID tags and submit own ID and Web services. This can possibly overwrite the data on RFID when RFID memory is full (if there is no new information then memory is not overwritten). Communication progresses in such epidemic way and eventually all robots around relay/RFID may have information about available Web services. However, this is no guaranteed in most of the cases due to constant mobility of robots and limited memory of RFID tags. Similarly, ECE can also publish their Web services queries and replies in RFID as well.

Situation becomes more interesting when robots move and join different robot-groups (entity-groups). In such situations, timestamps are used to check when a particular robot has left the area/RFID locality. In the new area the robot may register and publish both own and other discovered Web services descriptions (thus, serving as a mobile relay). Robot can also cache Web services queries that it cannot satisfy and may post them at a new RFID when it joins another robot-group.

In cases where rendezvous robots, in addition to edge robots, can spread information about available Web services descriptions among other rendezvous robots using RV, they post the discovered Web services descriptions to other rendezvous robots in their RV. Recipient rendezvous robots post the received Web services descriptions from other rendezvous robots to their corresponding RFID tags. Thus Web services descriptions of a robot-group becomes available to another robot-group.

5.1. Experimental setup

For experimentation of proposed middleware, Player/Stage Simulator [124] is used. Player/Stage Simulator provides a client control program that can talk to player interface of robot actuators and sensors over TCP Socket. Current implementation of middleware (in C++) is small and efficient (15KB memory footprint).

A brief snapshot is shown in Figure 13. For the sake of simplicity of presentation, only two mobile robots with Wi-Fi communication capability (i.e. rendezvous robots, represented by blue robots, with the enclosing circle representing their Wi-Fi range) are shown. Red robots represent edge mobile robots which are only capable of communicating via RFID (message relays, which are expressed by brown entities in simulation). The communication infrastructure modeled for experiments is attributed by instability and partial communication coverage due to robotic mobility. Thus robots can get out of coverage area of Wi-Fi or read/write range of static RFID tags in the environment.

Web services descriptions are expressed in form of extended RDF triplets. The reason for selection of RDF is to leverage the simplicity and semantic richness of RDF. RDF-based Web services descriptions have already been used by [116]. In current implementation MTL provides a set of API for Web services registration which is invoked by a robot client. Robot client automatically transforms the robot proxies (e.g. PlayerCc::Position2dProxy, PlayerCc::WiFiProxy) expressed by Player drivers into RDF-based Web services description as well to achieve the goal of exposing a robot and its all interfaces as Web services.

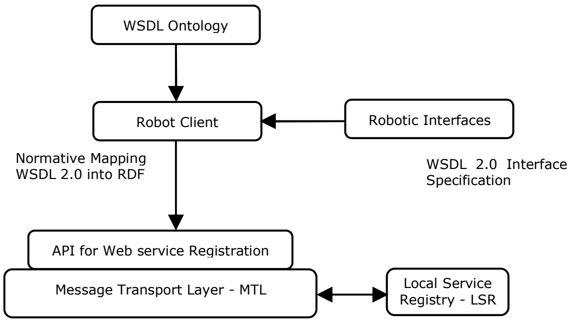


Figure 12: Robotic Web services registration

Robot functional interfaces as specified by Player Proxies are first expressed in Web Service Description Language (WSDL 2.0) [125]. W3C also provides a working group note for normative mapping of WSDL 2.0 into RDF [126]. Mapping specification [126] comprises of WSDL ontology and formal mapping from WSDL 2.0 to RDF data using proposed WSDL ontology [127]. Implementation uses the same proposed mapping of WSDL component model with corresponding classes in WSDL Ontology and component properties to RDF properties.

As a representative example let's consider a stripped down PlayerCc::Position2dProxy [128] of robots as the following generated WSDL 2.0 document:

```
<description
   xmlns="http://www.w3.org/ns/wsdl"
   targetNamespace= "http://roboswarm.eu/wsdl"
   xmlns:tns= "http://roboswarm.eu/wsdl"
   xmlns:ghns = "http://roboswarm.eu/schemas"
   xmlns:wsoap= "http://www.w3.org/ns/wsdl/soap"
   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
   xmlns:wsdlx= "http://www.w3.org/ns/wsdl-extensions">
 <types>
   <xs:schema
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
     targetNamespace="http://roboswarm.eu/schemas"
     xmlns="http://roboswarm.eu/schemas">
      <xs:element name="GetXSpeedRes" type="xs:double"/>
   </xs:schema>
  </types>
<interface  name = "Position2DProxy" >
   <operation name="GetXSpeed"
        pattern="http://www.w3.org/ns/wsdl/out"
        style="http://www.w3.org/ns/wsdl/style/iri"
        wsdlx:safe = "true">
     <output messageLabel="GetXSpeedRes"
         element="ghns:GetXSpeedRes" />
   </operation>
```

```
    </interface>
  <binding name="Position2DProxySOAPBinding"
          interface="tns:Position2DProxy"
type="http://www.w3.org/ns/wsdl/soap"wsoap:protocol="http://www.w3.org/2003/05/soa
p/bindings/HTTP/">
      <operation ref="tns:GetXSpeed"
        wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>
    </binding>
    <service name="Position2DService" interface="tns:Position2DProxy">
      <endpoint name="Position2DServiceEndpoint"
              binding="tns:Position2DProxySOAPBinding"
              address ="id:1_addr:6060"/>
    </service>
  </description>
```

A fragment of mapping of WSDL 2.0 to RDF is as follows:

```
<http://roboswarm.eu/wsdl#wsdl.description()> <http://www.w3.org/1999/02/22-rdf-
syntax-ns#type> <http://www.w3.org/ns/wsdl-rdf#Description>

<http://roboswarm.eu/wsdl#wsdl.description()> <http://www.w3.org/ns/wsdl-
rdf#interface> <http://roboswarm.eu/wsdl#wsdl.interface(Position2DProxy)>

<http://roboswarm.eu/wsdl#wsdl.interface(Position2DProxy)>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/ns/wsdl-
rdf#Interface>

<http://roboswarm.eu/wsdl#wsdl.interface(Position2DProxy)>
<http://www.w3.org/2000/01/rdf-schema#label> "Position2DProxy"

<http://roboswarm.eu/wsdl#wsdl.interface(Position2DProxy)>
<http://www.w3.org/ns/wsdl-rdf#interfaceOperation>
<http://roboswarm.eu/wsdl#wsdl.interfaceOperation(Position2DProxy/GetXSpeed)>
        …
        …
```

As mentioned earlier, the *Ontology repository* in CMA stores ontologies in form of RDF triples which are used for mapping Web services discovery queries to published Web services using semantic distance and recommendation mechanism (see Figure 6). For this purpose OWL ontology is translated into RDF triples. For achieving this first the OWL is transformed into *"OWL Abstract Syntax"* (OWL-AS) a W3C recommendation [130]. OWL Abstract Syntax is a high-level abstract syntax for both OWL DL and OWL Lite variants of OWL. OWL-AS based ontology is further parsed into RDF triples using W3C Group note for parsing OWL-AS to RDF [129].

Resultant RDF triples are stored in *Ontology Repository* of robots. One of the purposes is to be able to compute semantic distance of Web services query and expertise of robots for semantic recommendations of Web services queries. For the sake of elaboration let's consider a fragment of RoombaCleaning ontology. Other conceptual hierarchies can also co-exist in *Ontology Repository*, such as Ontology of platform/domain/other service types as WeatherInformationServices, SensingServices etc.

A fragment of automated translation of Roomba robot "cleaning" ontology is given as:

class('http://www.roboswarm.eu/RoombaCleaning.owl#Cleaning', false, complete, [], []).

class('http://www.roboswarm.eu/RoombaCleaning.owl#OutdoorCleaning', false, complete, [], []).

class('http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning', false, complete, [], []).

class('http://www.roboswarm.eu/RoombaCleaning.owl#Vacuuming', false, complete, [], []).

class('http://www.roboswarm.eu/RoombaCleaning.owl#FloorWashing', false, complete, [], []).

subclassOf('http://www.roboswarm.eu/RoombaCleaning.owl#OutdoorCleaning', 'http://www.roboswarm.eu/RoombaCleaning.owl#Cleaning').

subclassOf('http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning', 'http://www.roboswarm.eu/RoombaCleaning.owl#Cleaning').

subclassOf('http://www.roboswarm.eu/RoombaCleaning.owl#Vacuuming', 'http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning').

subclassOf('http://www.roboswarm.eu/RoombaCleaning.owl#FloorWashing', 'http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning').

A fragment of mapping of OWL-AS representation of RoombaCleaning to RDF (comma separated Subject, Predicate and Value) is as follows:

'http://www.roboswarm.eu/RoombaCleaning.owl#Cleaning', rdf:type, owl:Class

'http://www.roboswarm.eu/RoombaCleaning.owl#OutdoorCleaning', rdf:type, owl:Class

'http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning', rdf:type, owl:Class

'http://www.roboswarm.eu/RoombaCleaning.owl#Vacuuming', rdf:type, owl:Class

'http://www.roboswarm.eu/RoombaCleaning.owl#FloorWashing', rdf:type, owl:Class

'http://www.roboswarm.eu/RoombaCleaning.owl#OutdoorCleaning', rdfs:subClassOf, 'http://www.roboswarm.eu/RoombaCleaning.owl#Cleaning'

'http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning', rdfs:subClassOf, 'http://www.roboswarm.eu/RoombaCleaning.owl#Cleaning'

'http://www.roboswarm.eu/RoombaCleaning.owl#Vacuuming', rdfs:subClassOf, 'http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning'

'http://www.roboswarm.eu/RoombaCleaning.owl#FloorWashing', rdfs:subClassOf, 'http://www.roboswarm.eu/RoombaCleaning.owl#IndoorCleaning'

As mentioned earlier, an extended RDF data model is used for RFID tags. The simulation snapshot (see Figure 13) represents RDF encoded knowledge base, expressing the presence and registration of an edge robot with *ID 7010*. Service registration by robot is *vacuumPos2d*. The *context field* of RDF sextet refers to interpretation of RDF sextet for instance: *reg* (register the presence), *pub* (publish the expertise or Web service), *query* (query a Web service), *invoke* (invoke an interface) etc. In the snapshot the robot with ID 7011 queries for a Web service by name *Vacuum* and then requires against expertise *Cleaning*. Robot 7010 responds the query by referring to query identifier and the robot with expertise of *Cleaning*.

The WSDL to RDF mapping in further simplified in Figure 13. This simplification is solely for the purpose of debugging and pretty printing.

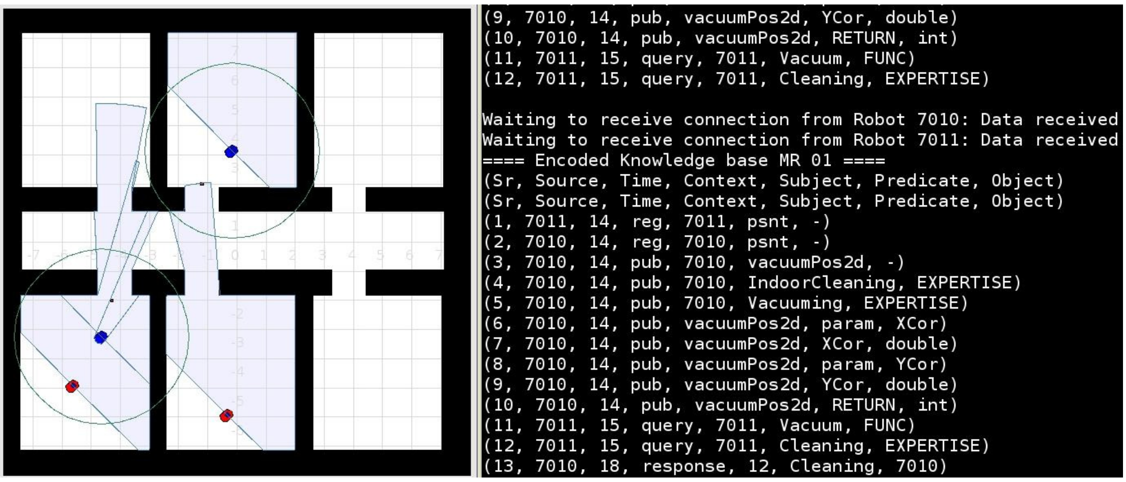


Figure 13: Sample Simulation & extended RDF knowledge snapshot

5.2. Experimental Results

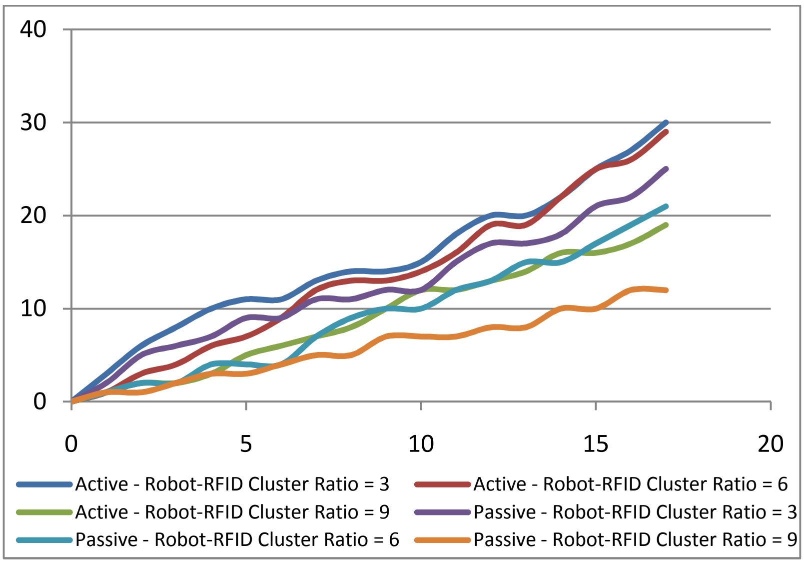
The following experimental results analyze the middleware in different communication modes i). Active and ii). Passive mode of communication with and without shared resource **conflict resolution (CR)**. This thesis analyzes discoverable Web services with respect to time, Web services available in complete view of cluster and the impact and resultant improvements in distributed Web services discovery by using caching and semantics.

5.2.1. Discoverable services without CR

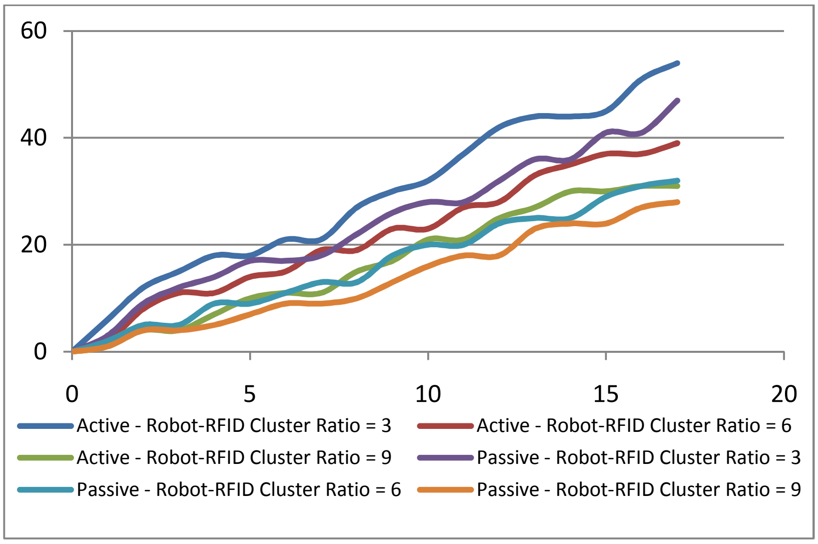
Discoverable services refer to those services that have more than one point of discovery in the system. In other words, if a Web service is cached at another robot other than the service host/provider robot then that service is considered a discoverable service. The definition comes from the fact that robot mobility can create situations where the Web services host robot is out of communication coverage. Service discovery and invocation are considered different independent steps in service management. Thus by having an additional host robot for Web services gives a higher probability of Web services being discovered even when host robot is inaccessible (due to partial failures).

Measurement of discoverable services with respect to time, measures the number of Web services that get cached at any other robot during the system execution. In simpler words, a higher number of discoverable Web services mean higher fault tolerance of system towards transient failures. Experiments (see Figure 14) show discoverable services metric in the absence of conflict resolution (CR). Experimental data is taken for varying RFID memory sizes and Robot-RFID cluster ratios. *Robot-RFID cluster ratio is the number of robots sharing a MR/RFID.* Experiments reveal that the active mode of communication achieves better performance in terms of discoverable Web services and with an increased Robot-RFID cluster ratio the performance deteriorates. This decrease in number of discoverable Web services, with higher Robot-RFID cluster ratio, is caused by increased number of conflict/messages-re-requests in a setting. This aspect will be shown separately in the comparison results of active vs. passive communication (subsection 5.2.6 and 5.2.7).

Result also reveals that in different RFID memory settings, a similar behavior repeats itself. Results reveal the effectiveness of active mode of communication over passive mode. This result gives a partial analysis, as it will be observed latter that such improvement in discoverable Web services comes with an additional cost of increased bandwidth consumption in active mode of communication and rather no real added value is achieved by using active mode of communication (due to message re-requests) in terms of performance and Web services discovery and invocation.



RFID Memory = 3

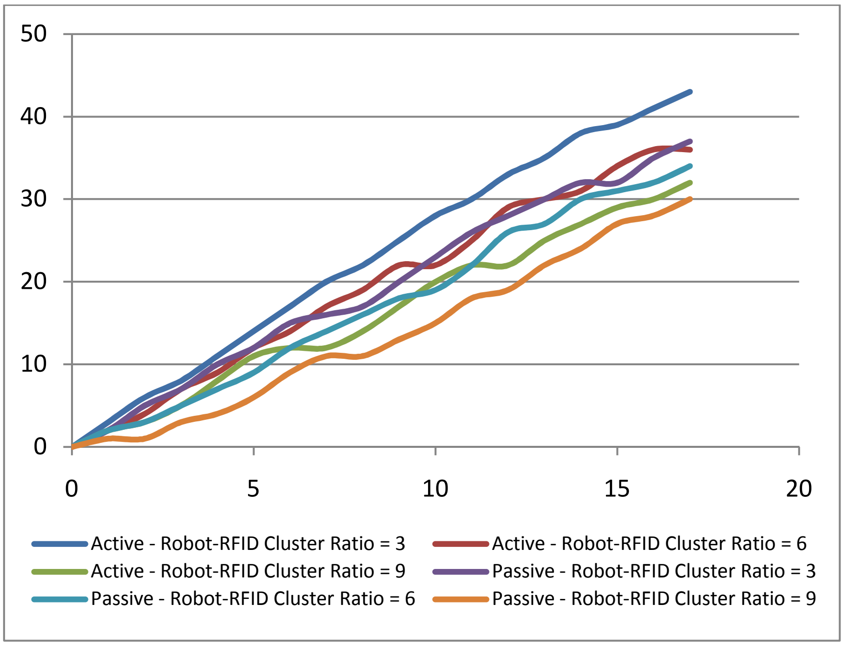


RFID Memory = 6

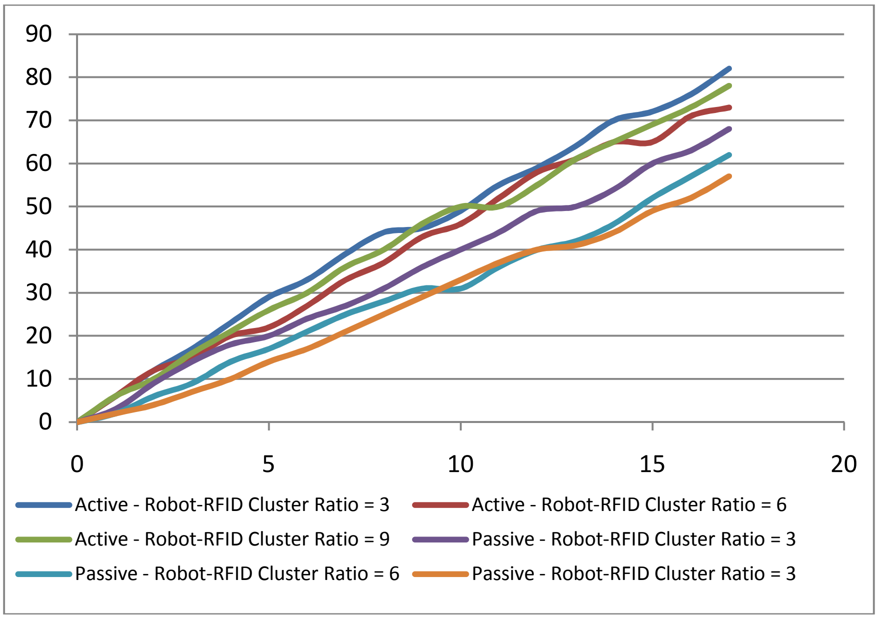
Figure 14: Discoverable Services without Conflict Resolution

5.2.2. Discoverable services with CR

Same experiment as above is performed with conflict resolution enabled. Here in this case, the number of discoverable Web services with respect to time in conflict resolution mode (see Figure 15) reveals a similar result as that of non-conflict resolution mode (see Figure 14). The number of discoverable Web services decreases as compared to non-conflict resolution mode of communication. But such a decrease also impacts the decrease in number of total messages communicated and Web services in complete view of cluster impact (see Figure 16 and Figure 17).



RFID Memory = 3



RFID Memory = 6

Figure 15: Discoverable Services with Conflict Resolution

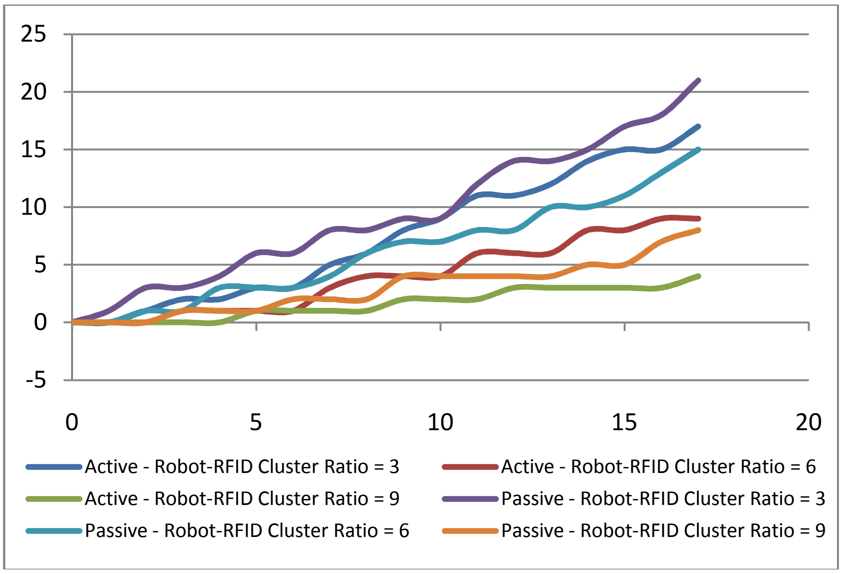
In this case, results (see Figure 15) are mostly similar for various settings. Active mode of communication still achieves the better results as compared to passive mode of communication. In different RFID Memory settings similar behavior repeats itself with active conflict resolution mode making a majority of Web services discoverable.

5.2.3. Services in complete view of cluster without CR

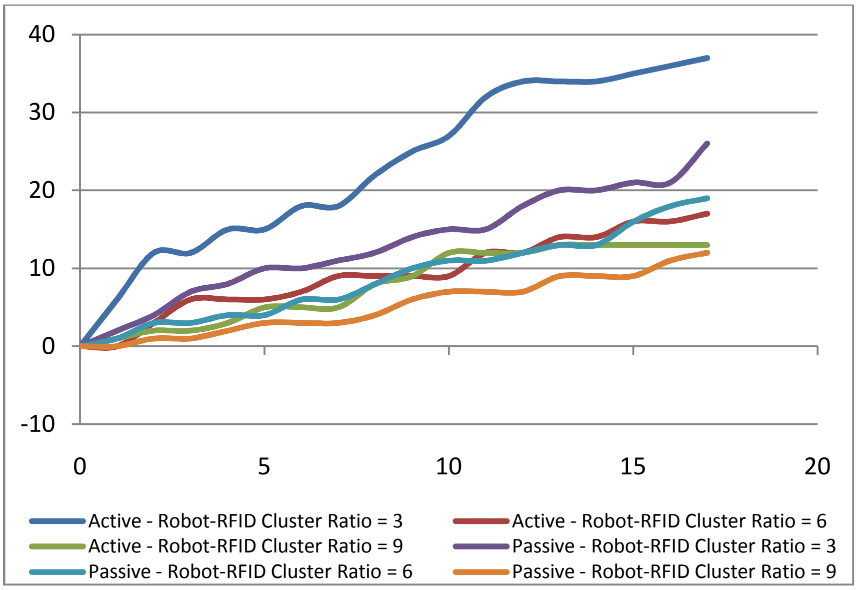
Services in complete view of cluster represents discovery of a service at majority of entities in Robot-RFID cluster. In other words this metric imposes much tighter constraints on Web services dissemination. In the previous result it was observed that the number of discoverable Web services decreases in conflict resolution mode as compared to non-conflict resolution mode of communication. Here another side effect of such conflict resolution can be observed.

Results (see Figure 16) reveal that passive mode of communication achieves better result as compared to active mode of communication. This can be viewed in a perspective that passive mode of communication is a delayed mode of communication, and while there is an increase in the number of Web services disseminations in case of active mode of communication, passive mode provides a request triggered response which is benefitted by majority of robots in a Robot-RFID cluster.

With different RFID memory settings as well, the same behavior repeats itself with passive mode of communication performing better than active mode. Here it can be observed that, with a MR that can accommodate more messages, active mode can perform better with smaller Robot-RFID cluster ratio. The reason for such a behavior is that the Web services descriptions pushed in active mode of communication can occupy MR for a longer duration of time, thus a delayed flush of Web services discovery occurs with lower Robot-RFID cluster ratio with a higher MR capacity, in a way this achieves a delayed mode of communication similar to passive mode of communication.



RFID Memory = 3



RFID Memory = 6

Figure 16: Services in complete view of cluster without CR

5.2.4. Services in complete view of cluster with CR

Similar result (see Figure 17) is achieved in conflict resolution mode as well. Passive mode of communication achieves better result as compared to active mode of communication. Results are slightly clearer in this case (though they give a similar insight as previous result without conflict resolution). This result does not reveal much improvement of conflict resolution mode over non-conflict resolution mode; however improvement are more clear in message dissemination and comparison of active vs. passive mode of communication results (see Figure 18, 19 and 20).

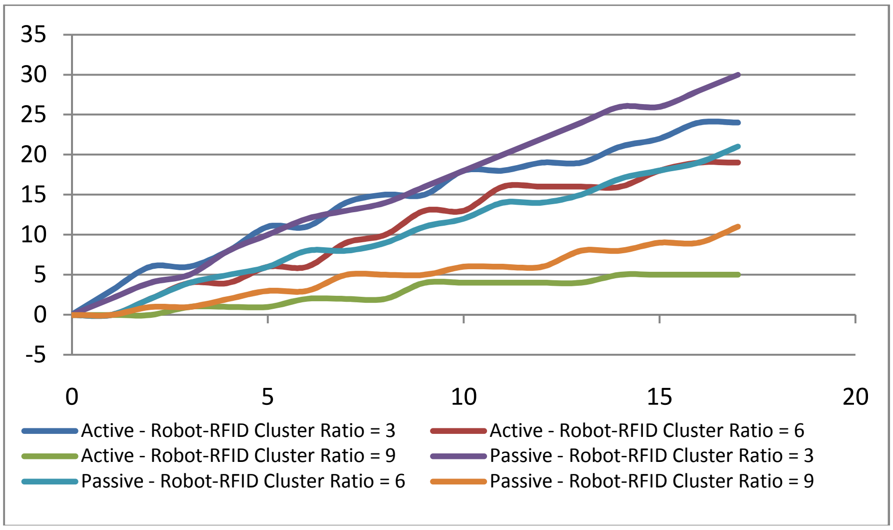
In conflict resolution mode different RFID memory settings show the same behavior with passive mode of communication performing better than the active mode. Same as non-conflict resolution mode it can be observed that with a MR that can accommodate more messages, the active mode can perform better with smaller Robot-RFID cluster ratio.

5.2.5. Impact of Cluster Ratio on Average Messages at a MR

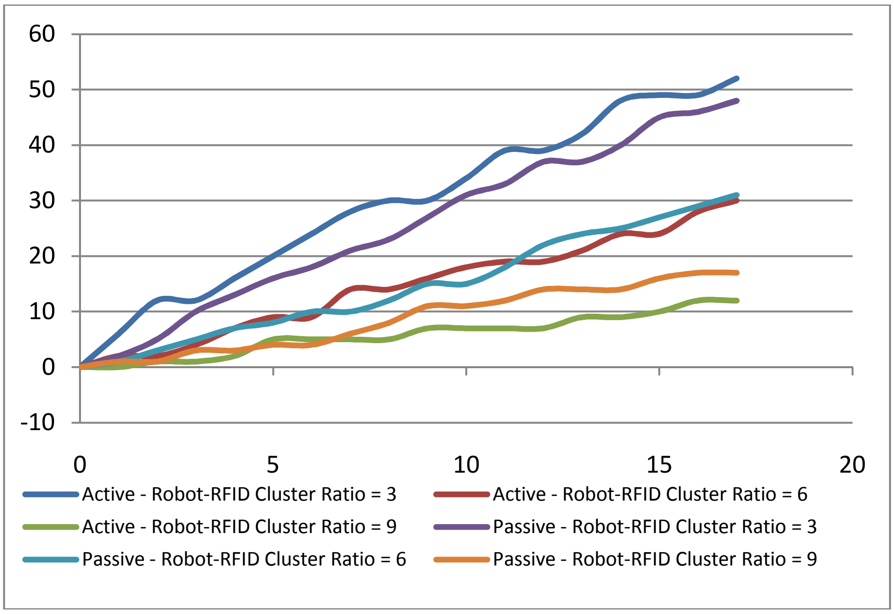
The comparisons and impact of robot-RFID cluster ratio on the average number of messages disseminated in message postboxes is shown in Figure 18. This result highlights a significant rise in disseminated messages in active mode of communication, while passive mode of communication gives improved results. With higher robot-RFID cluster ratio as well, the rise in number of disseminated messages in passive mode of communication is better to that of active mode of communication.

As observed in previous comparison of active and passive mode of communication with higher RFID memory, similar behavior is shown here as well. With increased RFID memory size, passive mode of communication gives improved results with higher Robot-RFID cluster ratio as compared to that of lower Robot-RFID cluster ratio of active mode of communication. This is shown in the overlap of Active Mode – average msgs with RFID memory = 3 over Passive Mode – Average msgs with RFID memory = 6.

In other words passive mode of communication exhibits controlled losses with increased robot-RFID cluster ratio and with higher RFID memory results of passive mode improves further. Active mode of communication on the other hand incurs more losses with higher robot-RFID cluster ratios.



RFID Memory = 3



RFID Memory = 3

Figure 17: Services in complete view of cluster with CR

5.2.6. Active vs. passive communication without CR

Difference in conflict/total-communicated messages (see Figure 19) highlights the fact that there are fewer conflicts with smaller robot-RFID cluster Ratio. Passive communication mode serves better as it has less overhead and conflicts are the number of request losses not as compared to the service description losses as in case of active mode of communication. With higher RFID memory size and lower robot-RFID cluster ratio both modes of communication show similar results.

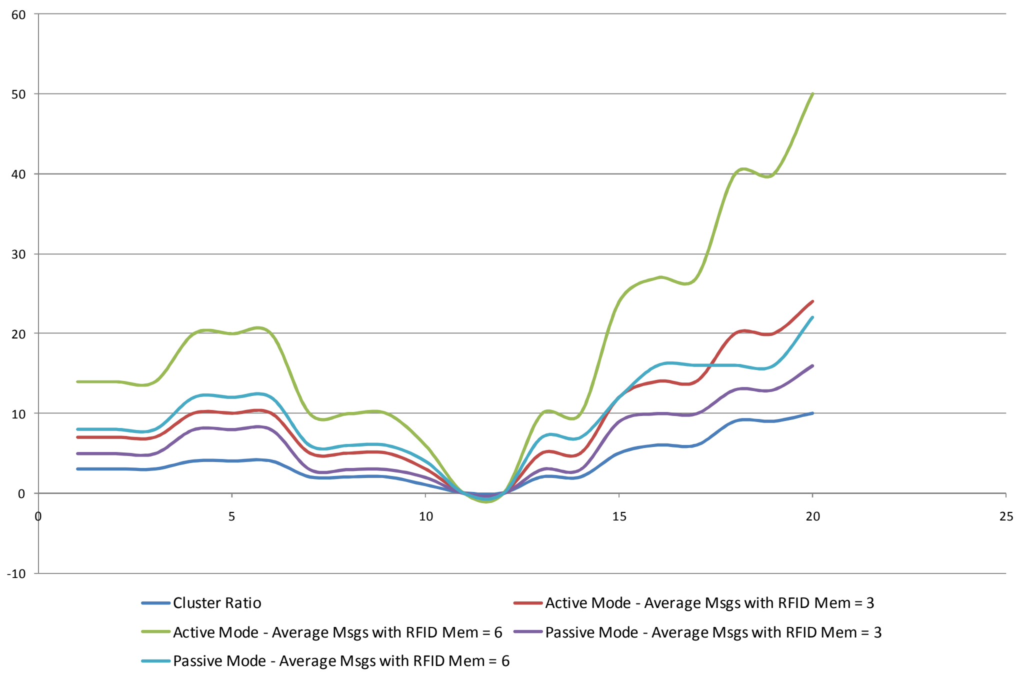


Figure 18: Impact of Cluster Ratio on Average Messages

5.2.7. Active vs. passive communication with CR

In case of conflict resolution mode as compared to non-conflict resolution mode experimental setting, there is a significant difference in active and passive mode of communication (see Figure 20). Such a difference is obvious from previous experimental results of passive mode conflict resolution communication as well. Such a difference is insignificant with higher robot-RFID cluster ratios.

With increased RFID memory size, passive mode of communication with conflict resolution gives further improved results with higher robot-RFID cluster ratio as compared to lower robot-RFID cluster ratio of active mode of communication with conflict resolution (as the above graph with RFID memory = 6 shows better results for Passive Mode – robot-RFID cluster ratio = 6 over Active Mode – robot-RFID cluster ratio = 3). The reason of such a behavior is the impact of service description memory requirement as compared to that of service requests in case of passive mode of communication. Such a behavior continues further with higher memory sizes of RFID.

The comparisons show that passive mode of communication serves much better in terms of number of conflicts and total messages as compared to active mode of communication. Passive mode shows better results with higher robot-RFID cluster ratio with higher memory size and with improved RFID memory it can support higher robot-RFID cluster ratios.

5.2.8. Impact of caching and semantics in service discovery

Lastly validation of the impact of caching and semantics in service discovery is done. Passive mode of communication is used with conflict resolution with different settings of robot-RFID cluster ratio. Results (see Figure 21) reveal that Web services discovery with caching augmented with semantics provides the best results (i.e. least number of hops/messages required for Web services discovery). There can be few abnormalities in result, for instance execution 14 shows better performance of syntactic discovery as compared to semantic service discovery with caching. This case represents a drawback of caching in mobile setting. The particular case exhibits a scenario in which a query is routed to a cluster (based on caching results) but due to mobility host robot has joined a different cluster – thus stale information in service cache effects additional number of hops across clusters for service discovery.

5.3. Conclusions and on-going work

This thesis proposes a semantic middleware which provides loose coupling between heterogeneous autonomous entities, and provides mechanisms for distributed service discovery. Developed mediator-based distributed Web services discovery and management middleware, though tested primarily for robot swarms only, can be applied to any infrastructure-less dynamic environment. Thesis proposes and develops a solution where dynamic environment and computing entities are abstracted to a P2P system and a conceptual classification of entities according to their communication capabilities is used. It is assumed that the physical environment has some type of mediator entities, which can be used for communication when point-to-point communication between entities is not possible.

In implementation of middleware the discovery of Web services is done on the basis of keywords and expertise of an entity is used for efficient robot selection for query propagation. In ongoing work the approach of [131] is being used to model the real-world concepts related to Web services and map the ontological concepts with WSDL. The semantic model and associated mapping will then be stored in LSR as RDF. Thus any entity/robot communicating with another entity/robot or a request for Web services can be inspected against the ontological concepts. In a broader perspective, it is intended to create a rule-based service discovery in which rules can provide the foundation for semantic reasoning and formal concept learning [132] to overcome the limitations of using a pre-defined ontology of domain.

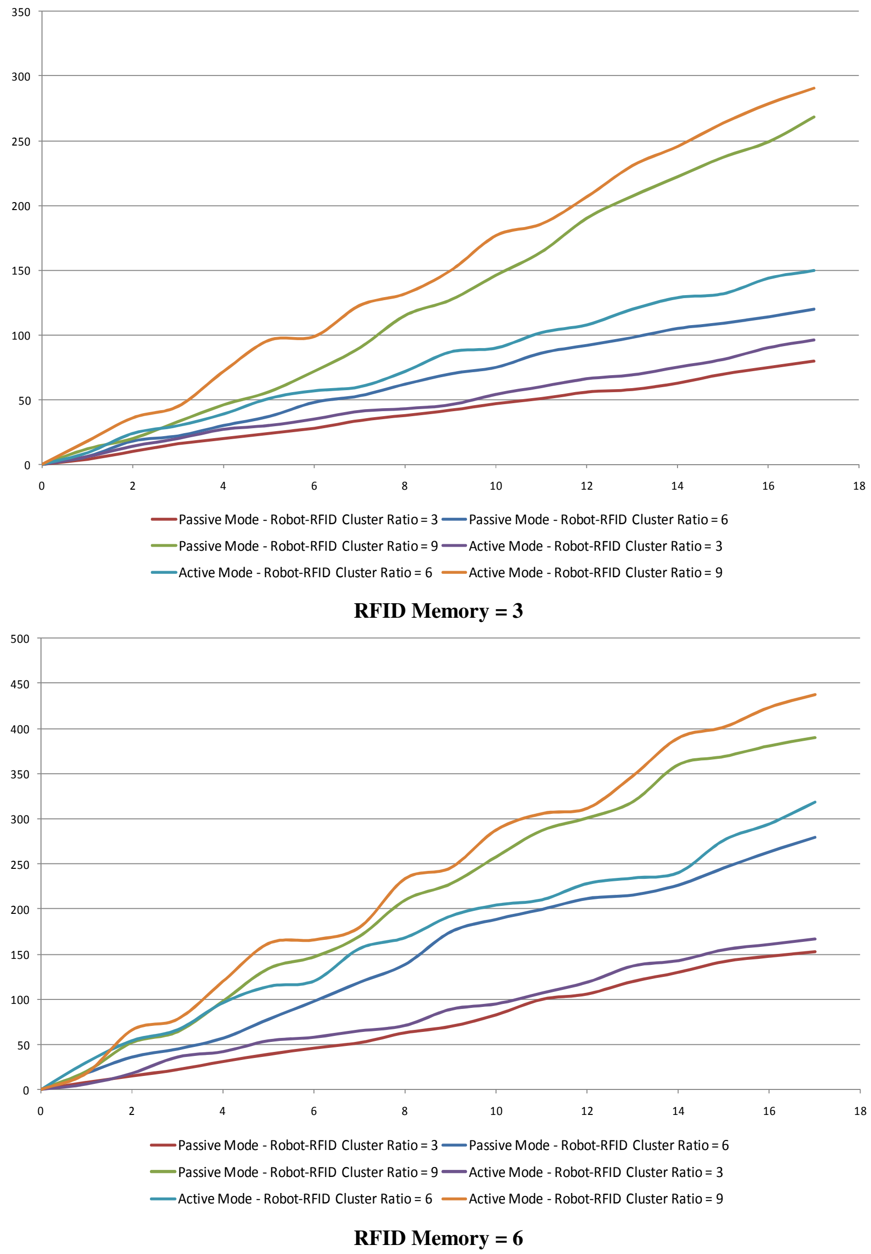
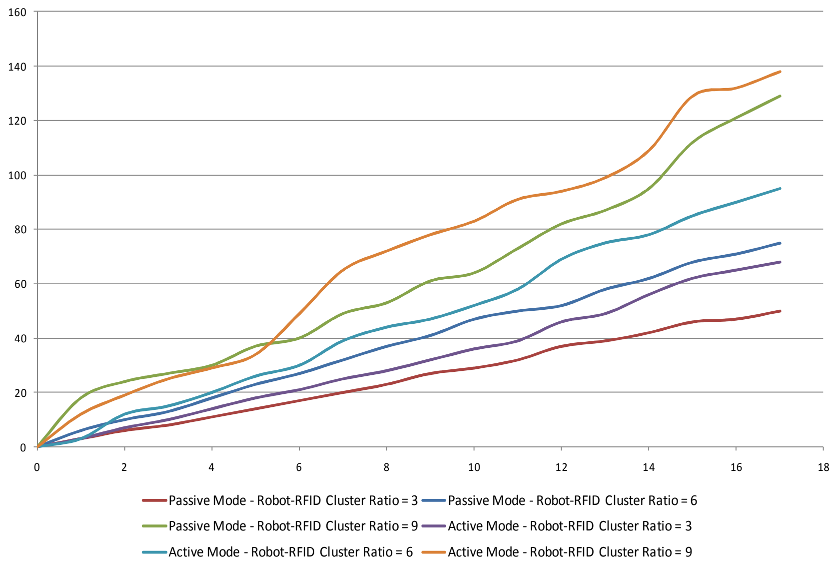
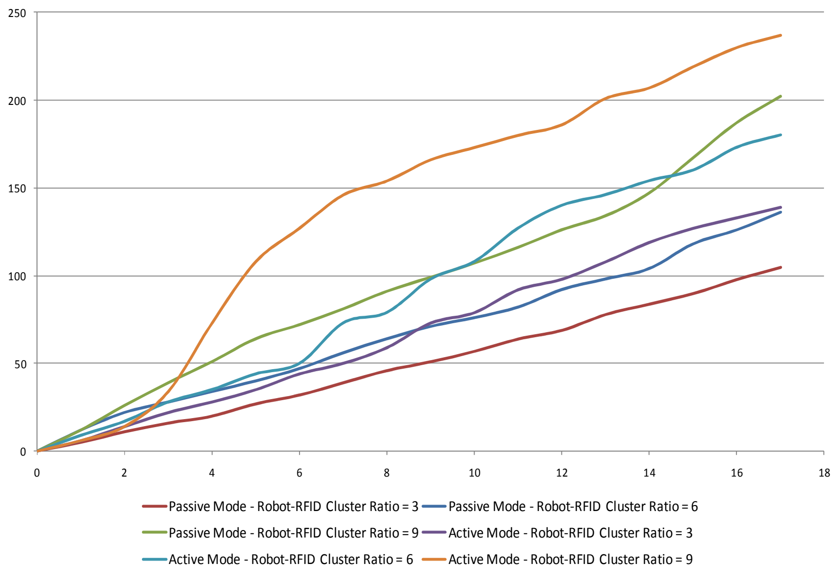


Figure 19: Active vs. Passive communication without CR



RFID Memory = 3



RFID Memory = 6

Figure 20: Active vs. Passive communication with CR

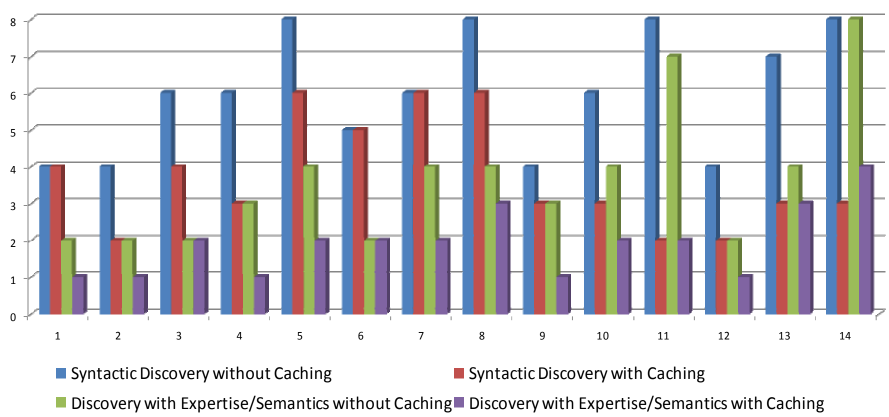


Figure 21: Impact of caching and semantics in service discovery

Ongoing work is also extending similarity function [133] for query's subject mapping to expertise description. Also an algorithm for adaptive degree of separation based on network topology to cache services for the creation of a flexible semantic topology is currently being experimented.

5.4. Scientific contributions

1. **Abdul Haseeb**, Mihhail Matskin and Peep Kungus, Distributed Discovery and Invocation of Web Services in Infrastructure-less dynamic environments. *IEEE International Journal of Web Services Practices (IJWSP), 2009.*

2. **Abdul Haseeb**, Peep Kungus and Mihhail Matskin, Mediator-based Distributed Web Services Discovery and Invocation for Infrastructure-less Mobile Dynamic Systems. *In the proceedings of 4th International IEEE Conference of Next generation Web services practices (NWeSP'08), October 2008, Seoul South Korea.*

3. Shahab Mokarizadeh, Alberto Grosso, Mihhail Matskin, Peep Kungas, **Abdul Haseeb**. Applying Semantic Web Service Composition for Action Planning in Multi-Robot Systems. *Proc. ICIW 2009, IEEE, May 24-28, Venice*.

4. **Abdul Haseeb**, Mihhail Matskin and Peep Kungus, Light-Weight Decentralized Autonomic Web Service Discovery for Systems with Heterogeneous Communication Capabilities, *In the proceedings of 12th IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA'08), August 2008, Hawaii, USA*

5. **Abdul Haseeb**, Peep Kungus and Mihhail Matskin, Semantic Middleware for Robotic Swarm Interaction. *In the proceedings of IASTED, Modeling, Identification and Control conference (MIC'08), Innsbruck, Austria.*

Part 2

Distributed Learning in the absence of Global Information

Defeasible argumentation based incremental learning solution

6. Defeasible Argumentation and Intelligence in the absence of Global Knowledge

It has been advocated that defeasibility forms the basis of intelligence in human-beings. In common-sense inference (human reasoning/inference), conclusions are drawn and refuted at the same time as more information becomes available from surrounding environment and interaction with other entities. Noteworthy aspects of common-sense reasoning are its ability to adapt to changing and dynamic environments and reasoning with uncertainty in the absence of global information.

In autonomous and decentralized systems such as robot swarms, having a universal global knowledge is too restrictive and newer domains are the norms of computational environments. Porting such intelligence and inference capabilities is beneficial. Towards these objectives, this thesis develops a solution for incremental learning in robot swarms.

This chapter first takes a brief overview of epistemic cognition in human beings, which will motivate the need of similar epistemic cognitive solution in fabricated systems such as robot swarms [27]. Then a brief elaboration of formal concept analysis is given, which forms a case for incremental learning of formal concepts. A representational scheme for underlying logic formalism and formal concepts in robotic knowledgebase is also presented. An algorithm for robotic navigation in which robots can incrementally learn formal concepts and perform common-sense reasoning for their intelligent navigation is proposed. This chapter also elaborates the logic formalism employed and details of implementation of developed defeasible reasoning engine.

6.1. Perceptual Knowledge and Epistemic Cognition

Philosophical argumentation advocates the continuity of entities and their interactions with the surrounding world. Towards such continuum Pollock [134] presented a notion of perceptual knowledge (which comes from the entity's interferences with the surrounding – i.e. lower-level inference results) and epistemic cognition (higher level inter-belief relationships). Primary need is a direct access of entities to the world, as a part of a continuous whole via touch, vision and other sensory information. Such a stand point highlights the perceptual knowledge of entities as a result of their inference from indirect evidences with core-knowledge obtained from the surroundings.

A representative cognitive system comprises of a belief system that comprises of perceptual and higher-level beliefs. Perceptual beliefs refer to beliefs that an entity infers in direct correspondence with its sensory information. Higher-level beliefs (belief revisions or higher level epistemic cognition) are those beliefs which undergo more sophisticated reasoning. Epistemic cognition system can be expressed as following (see Figure 22).

In other words, perceptual beliefs are direct correspondences with surrounding world using sensory information and epistemic cognition is the intelligence of an entity. This intelligence or epistemic cognition explains an entity's rationality i.e. the rational influences of beliefs over each-other.

Few main and relevant characteristics of a human epistemic cognition are their *adaptability to reason in dynamic and varying environments* and *reasoning with uncertainty*. This means that reasoning capabilities of humans are agnostic to environment. Moreover lack of global knowledge doesn't render their inference useless, though suffers from uncertainty.

In general, humans can not only form new beliefs but also retract the existing beliefs when new information becomes available. Such flexibility is of paramount significance towards scenarios where having a universal global knowledge is restrictive and changing domains are a norm. In autonomous and decentralized systems such as robot swarms (or in general any multi-agent system), having a universal global knowledge is too restrictive and newer domains are the norms of computational environments. Porting such intelligence and inference capabilities is beneficial. Towards these objectives, this thesis develops a solution for incremental learning in robot swarms which mimics common-sense reasoning as exhibited by human beings.

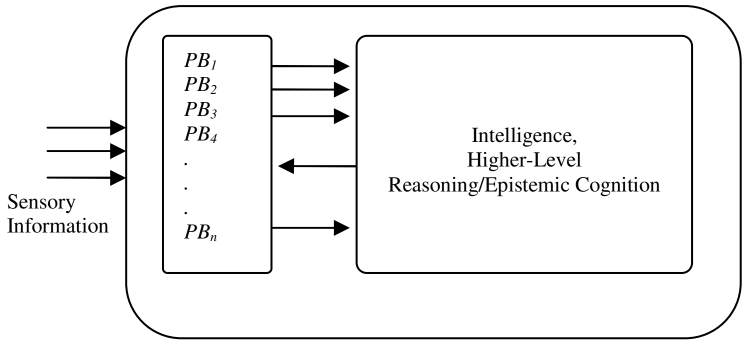


Figure 22: Epistemic Cognition in Human Beings

Epistemic cognition in human beings is argued to be based on some sort of defeasible reasoning [135]. Proposed solution towards establishing common-sense reasoning in robot swarms is, thus, based on defeasible logic. First an overview of basic concepts of defeasible reasoning is given, and then a solution for defeasible intelligence in robot swarms is elaborated.

6.2. Defeasible Reasoning and Logic formalisms

Defeasible reasoning [135] (also referred to as *a fortiori* reasoning) is an argument based reasoning, in which reasons provide the links in arguments for forming new beliefs (epistemic cognition). Defeasibility arises from those reasons that are not conclusive (i.e. reasons that can potentially cause uncertainty), thus requiring other augmenting reasons for inference. A simple example of defeasible reasoning can be illustrated by the following example:

> "We often judge the colors of objects on the basis of what color they look to us… Thus, for example, –x looks *red* to me; provides a reason for a judgment about the actual color… But if Jones, whom I regard reliable, insists that it is not *red* and provides a rebutting defeater, then one has to revise one's belief … a candidate defeating reason might be that the object is illuminated by red lights, which might make things look *red* when really they are not … "

J. L. Pollock, *"How to reason defeasibly"*, Artificial Intelligence 57(1) 1992

It is obvious from the above example that defeasibility and refutation of beliefs forms the basis of intelligence in human beings. This thesis conjectures the same for robotic intelligence. In other words, in a common-sense reasoning, conclusions (or beliefs formulated – perceptual beliefs) drawn at a certain point in time and space with respect to available knowledge can be latter refuted when more information becomes available. In some circumstances refutation of beliefs comes from existing beliefs in belief system, for instance in case of conflicting and contradictory beliefs (contradictory beliefs are illustrated latter in this chapter). There is an inherent non-monotonic growth of conclusions in such common-sense/defeasible reasoning – thus it's referred to as a non-monotonic reasoning formalism.

In order to formally define non-monotonic reasoning let's consider a set of facts *F* entailing a set of conclusions *c* at a certain time *t* (here facts refer to information of an entity and conclusions are the beliefs drawn from that information). At time *t', with t'>t,* a new set of facts *F'* are gathered (i.e. information of an entity gets updated). Set of conclusions drawn from *F'* are given as *c'*. Classical monotonic reasoning will entail a guaranteed subset relationship between *c'* and *c*. Non-monotonic reasoning, on the other hand, doesn't guarantee such a relationship between these sets of conclusions. In other words a logical conclusion of a super-/sub-set of *F'* might not be a logical conclusion of any super-/sub-set of *F'*. Thus addition or removal of facts can either reduce or increase the set of logical conclusions. Contrary to classical monotonic reasoning, in which a logical conclusion of any superset of facts also remains the logical conclusion of any subset.

The main characteristics of such non-monotonic reasoning are *adaptability to changing environments* and *reasoning with uncertainty*. This means that the system can not only form new beliefs but also retract the existing beliefs when new information becomes available. Such sort of adaptive and incremental reasoning is advocated to be used by humans, who update their belief system during a passage of time as new facts and conclusions are drawn and old are refuted on the basis of new information obtained.

There are a number of approaches for non-monotonic reasoning. Noteworthy and relevant to formalism and application are *Argumentation system* proposed by Pollock [138. 139. 140 and 141] and *Defeasible Logic* proposed by Donald [136, 137]. In both approaches the notion of defeat or refutation is central. Nute's formalism proposed the refutation of beliefs on the basis of rules, i.e. rules form the basic criterion for inferring new facts and refuting the existing ones.

The language proposed by Nute resembles the approach of Pollock but instead of defining argument defeaters, he considered rule defeaters on the basis of rule priorities. The underlying defeasible theory or knowledge base consists of five key elements of knowledge.

1. *Facts:* These are non-refutable statements, i.e. knowledge that is true under all circumstances and no knowledge base addition or deletion counters the belief held by those facts.
2. *Strict Rules:* Classical rules are referred to as strict rules, i.e. whenever premises are true, conclusions of the rule become true.
3. *Defeasible Rules:* Rules that can be defeated by some contradictory information (rules). The basic philosophy of defeasible rules is that one can conclude the conclusion of defeasible rule unless there is another superior evidence of not believing so.
4. *Defeater Rules:* These are the rules that can't draw conclusions on their own but can provide contradictory evidences.
5. *Priority Relations among rules*, these priorities define strength of rules among each other. Priority relations are used for conflict resolution when more than one contradictory rule can be triggered.

Defeasible theory can be illustrated by an example.

Facts:	*emu (tweety)*
Strict Rules:	*emu(X) → bird(X)*
Defeasible Rules:	*r: bird (X) ➔flies (X)*
	s: brokenWing (X) ➔ ~flies (X)
Defeaters:	*t: heavy (X)➔ ~flies (X)*
Priority Relation:	*s > r*

The above defeasible theory provides an indisputable statement that "*Tweety is an Emu*", i.e. under no circumstances the opposite can be inferred. Strict rule states that "*If X is an Emu then X is a Bird*", such an inference cannot be refuted (it corresponds to classical logic rules). Defeasible rule *r* in the above defeasible theory states that that "*If X is a bird then **there is an evidence** that X can fly, **unless and otherwise a there is no superior rule inferring the opposite***", similarly defeasible rule *s* states that "*If X has broken wings then **there is an evidence** that X cannot fly, **unless and otherwise a there is no superior rule inferring the opposite***".

Let's consider a situation in which some animal *tweety*$_2$ is given as an *emu,* and that *tweety*$_2$ has broken wings, the rule chain above will infer that *tweety*$_2$ is a bird using strict rule. Defeasible rules *r* and *s* both get triggered and give a contradictory conclusion. Rule *r* concluding that $tweety_2$ can fly, while rule *s* inferring the opposite. To avoid such conflicts priority relations between rules are used as given by $s > r$ priority relation rule. Please note that to avoid rule cycles, *non-commutative* property should hold for all priority relations.

Using the above rules and facts, a system can incrementally learn new facts and rules and can update its conclusions. For instance if latter a new fact is gathered that $tweety_3$ is an *emu* and it's a *heavy animal* the above rule chain will infer that $tweety_3$ can fly. If system at a certain future point in time learns another rule *t* that *t: heavy (X)* ➔ *~flies (X)* inferring that if an animal is heavy that it might not fly. Again such a defeasible theory is subject to contradictory conclusions. Thus a priority relation $t > r$ is needed, so that the system can refute its belief about flying capabilities of $tweety_3$.

To summarize, a common-sense human epistemic cognition requires a capability to perform inference in partial knowledge and should be able to provide incremental learning capabilities. Moreover, there is a separation of concern (see Figure 22) in terms of inferred knowledge and beliefs. Systems such as robot swarms are attributed by *partial knowledge*, and there is a need to *learn concepts in an incremental and refutable manner*. This forms the main motivation of this research effort.

This thesis develops a solution for incremental learning in robot swarms using defeasible reasoning in the light of formalism elaborated above. A representative application of robotic navigation is considered in which robots are equipped with few basic rules (rules that trigger robotic actions) and robots learn new concepts and perform inference as they gather new information from their surrounding and interaction with other robots. For concept learning, formal concept analysis (FCA) is used. Such systems of autonomous entities are attributed by incomplete information and classical logic based approaches are not suitable as it is latter motivated by abstracting a system with preferences. Proposed methodology easily incorporates semantic information, and expresses such in terms of rule updates, and provides a very flexible way of updating robotic control platform. This thesis develops robotic navigation on the basis of partial lattice i.e. incomplete information without a priori knowledge of a location model, unlike existing models, which rely on global knowledge of location models for navigations. Basic concepts of formal concept analysis are elaborated first before moving to contribution and results of research.

6.3. Formal concept analysis

First overview of basic mathematical foundations of formal concept analysis (FCA) is given which forms the basis of defeasible reasoning based semantic location model for robotic agents. Formal concept analysis [142] proposed by R. Wille has been used for creation of concept lattice and hierarchical structure formation among formal concepts [143, 144]. This thesis uses a similar approach for robotic concept learning.

Notion of formal concepts is based on a binary relationship between objects and their attributes. This binary relationship between object and attributes is referred to as *formal context*. If we consider *Obj* as set of objects having a binary relation *Rel* with a set of attributes *Attr*, then a formal context can be defined as a tuple $(obj_i, Rel, attr_j)$, which refers to a binary relation between object obj_i and attribute $attr_j$ where:

$$\left(obj_i \in Obj \; and \; attr_j \in Attr\right)$$

Based on the definition of formal context, a *formal concept* can be defined as *(X, Y)* where following conditions are satisfied.

$$X \subseteq Obj \; and \; Y \subseteq Attr$$

$$X = Y' \wedge Y = X' \; where$$

$$X' = \left\{ attr_i \in Attr \mid obj_j \; Rel \; attr_i, \forall obj_j \in X \right\}$$

$$Y' = \left\{ obj_i \in Obj_i \mid obj_i \; Rel \; attr_j, \forall attr_j \in Y \right\}$$

Thus *X'* is the set of common attributes among all objects in set *X* and *Y'* is the set of common objects among all attributes in set *Y*.

One of the potential applications of formal concept analysis is finding a minimal path among formal concepts using partial ordered relation ≤ [145]. To elaborate the idea let's consider a simple example of a building in which rooms and their corresponding exit locations form objects and attributes respectively, with existence and non-existence of an exit forming a binary relationship between rooms and exits. The example is visually depicted in Figure 23.

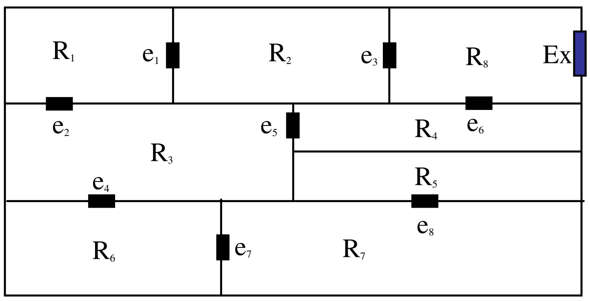


Figure 23: Navigation building with location and exit markings.

In this example R_1, R_2 … R_8 are sets of objects, and e_1, e_2 … e_8, Ex are the set of attributes. $(\{R_1\}, \{e_1, e_2\})$ is a formal concept according to definition of formal concept at level 1. Similarly $(\{R_2\}, \{e_1, e_3\})$ and $(\{R_3\}, \{e_2, e_4, e_5\})$ are also formal concepts.

Similarly $(\{R_1, R_2\}, \{e_1\})$ and $(\{R_1, R_3\}, \{e_2\})$ are formal concepts which can be formed from formal concepts of level 1 determined by sub-concept relation ≤ (partial order relation) which states that

$$\left(\left(X_1, Y_1\right) \leq \left(X_2, Y_2\right) \leftrightarrow \left(X_1 \subseteq X_2 \vee Y_2 \subseteq Y_1\right)\right).$$

Such formalizations can give us a naïve way of computing an optimal path between two formal concepts which can be formally expressed as:

$$\Big(\big(X_1, Y_1\big) \sim \big(X_2, Y_2\big) \leftrightarrow \neg\exists \big(X_i, Y_j\big) \mid \big(X_i, Y_j\big) \leq \big(X_2, Y_2\big)\, AND \big(X_1, Y_1\big) \leq \big(X_i, Y_j\big)$$

The set of all formal concepts for formal context represented by $\mathbb{C}\left(obj_i\ Rel\ attr_i\right)$ with partial order relationship ≤ can be represented as a concept lattice. For instance, lattice for shortest path between R_1 and R_8 which contains the exit '*Ex*' can be given as:

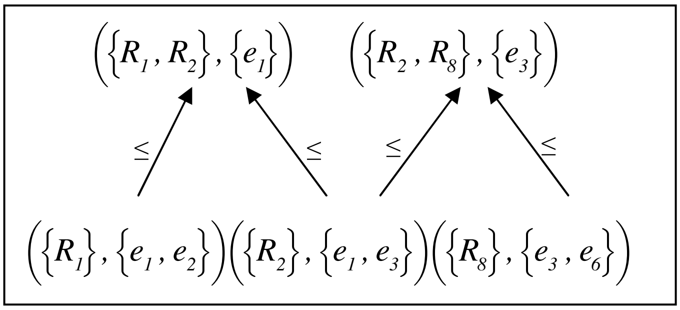


Figure 24: Generated concept lattice from example in Figure 23.

Though such a notion of lattice-based representation of formal concept relationships can provide a significant benefit yet the following shortcomings render it insignificant for mobile autonomous robots with incomplete information of surrounding and concepts. Namely, one of the basic assumptions of such a location model is on a priori knowledge of location model. For instance a mobile robot in R_1 can only assert about the formal concept $(\{R_1\}, \{e_1, e_2\})$ but no assertion can be made about R_2, R_3 thus rendering it unable to create any sort of relationship between other formal concepts. More specifically a robot in R_5 cannot even assert $(\{R_5\}, \{e_8\})$ as a formal concept. Thus an incremental way of discovering formal concepts and their relationships is needed.

Moreover, some sort of semantic-based strength mechanism should be provided to create partial order relationship on the basis of robotic profiles, permissions, task and/or states of rooms. This, however, leads us to another problem of adaptive robotic behavior in which a robot should be able to refute its current belief system and update itself to dynamically changing environment conditions.

6.4. Defeasible Learning Model in Robots

This section presents a model for robotic navigation in which robots can incrementally learn formal concepts and perform common-sense reasoning for their intelligent navigation. As elaborated earlier, systems of autonomous entities such as Robot Swarms are attributed by incomplete information and in such paradigms classical logic based approaches are not suitable. This thesis develops a robotic navigation model on the basis of partial lattice i.e. incomplete information without a priori knowledge of a global location model, unlike existing models which rely on global knowledge of location models for navigations. Proposed methodology easily incorporates semantic information, and expresses such in terms of rule updates, and provides a very flexible way of updating robotic control platform.

Autonomous mobile robotic systems rely on identification of locations via sensory information and perform the delegated task in mainly non-contextual manner. Considering a swarm of autonomous mobile robots with each robotic entity playing a certain role in a bigger context, puts forward a requirement on a well-formed spatial knowledge. To provide context based or location aware services, a location model to capture spatial connectivity and associated relationship between various objects/concepts is of significant importance. Many researchers have tried to address the issue of developing location models to describe physical space and relationship of concepts.

Contrary to most of existing work, which make an assumption of a priori build-up of location models, this thesis considers a defeasible reasoning based approach, which enables a robot or robotic agent to develop and maintain the location model as it navigates and performs actions. Thus a robot is totally oblivious of its surrounding conditions, yet equipped with some basic rules which enable it to learn new concepts and relationships in an incremental and contextual manner.

Earlier location models can be classified into symbolic and geometric location models [146]. In symbolic location models [147], all objects are identified by unique names ("symbols") and a partial ordering of symbols forms a location hierarchy in form of tree or a graph [148, 149 and 150]. Symbolic models specify containment and connectedness relationship in a spatial space without specific geometric information. Though symbolic location models with semantic relationship provide an intuitive and understandable representation but suffer from the manual costs of building and maintaining such symbolic models. On the contrary geometric location models abstract physical space as Euclidean space and describe each object as a set of coordinates. GPS is one of the noteworthy applications of geometric location models, which is based on longitude-latitude-altitude coordinate systems. Geometric location models can represent relative structure among the objects in spatial space but cannot express the semantic relationship in physical world.

Proposed methodology relies on Symbolic Model and on the notion of 'location and exit' proposed by [151], thus this thesis focuses mainly on robotic navigation and establishing a robotic view of semantic relationship and semantic distance to express location model. Notion of location refers to a bounded geometric area equipped with a number of exit points. It is assumed that the robotic sensory module can extract semantic information embedded on RFID tags placed at exit locations. This approach builds up on the exit-location matrix proposed by [152] in which any element of exit-location matrix a_{ij} refers to presence or absence of i^{th} exit at j^{th} location. Contrary to approach of [152] an incremental learning of surrounding and establishing of semantic based strength of connections is considered, thus resulting in contextual/semantic based concept lattice.

First Defeasible Logic Programming (DeLP) formalism in robotic knowledge-bases is presented followed by formal concept representation and an incremental formal concept learning method. Defeasible reasoning is used in conjunction with formal concept learning for action management of robots.

6.4.1. Robotic Knowledge-base representation as DeLP

Defeasible argumentation inference proposed by Defeasible Logic Programming (DeLP) formalism, is based on weak rules (defeasible rules) and is a non-monotonic reasoning formalism, which entities can employ to trigger their actions or to build-up a belief system. Notion of weak defeasible rules [152] is the fundamental aspect which forms a relationship between knowledge and represents tentative information [155, 156, and 157]. One representative notions is given in [154, 158], which will be used in following sections.

Knowledge base of a robot can be represented as $\kappa = (\Psi, \Delta)$

- Knowledge base comprises of a consistent set of facts Ψ, a set of rules Δ and priorities among defeasible rules. Priorities are considered among rules as a part of rule-set Δ as well, as it is conjectured that priorities are also a useful knowledge that can be inferred and updated collaboratively.
- A rule, in general, comprises of antecedent/body providing either strong or defeasible inference on consequent/head. Rule-set Δ can comprise of two types of rules:
- Strict rules, expressed by ←, to represent sound knowledge such as facts or some inference that cannot be refuted.
- Defeasible rules, expressed by כ, to represent tentative information, which is defined as "reasons to believe in antecedent/body provides reasons to believe in consequent/head" [159].

To elaborate a potential reuse and implementation methodology let us consider the following example.

$$\Psi = \left\{ \begin{array}{l} At\ (s, R_1),\ Status\ (R_1, Safe),\ Role\ (s, ?) \\ Exit\ (R_1, e_1),\ Exit\ (R_2, e_1) \\ Exit\ (R_1, e_2),\ Exit\ (R_3, e_2) \end{array} \right\}$$

Rule₁: wants_move(Y) כ at(s, X), Exit (X, Y)

Rule₂: ~wants_move(Y) כ at(s, X), Exit (X, Y,), Exit_Status(Y, Halt)

Rule₃: Rule₂ > Rule₁

In the example knowledge base comprises of a set of facts and a rule-set comprising of two defeasible rules and a conflict resolution rule (priority relation). Knowledge base asserts about a robot *s* being at some location R_1 via *At (S,* R_1*)* factual data. This thesis assumes that the robotic sensory module can extract semantic and some minimal location information embedded on RFID tags that are placed at exit locations (see Figure 23). In this scenario each RFID tag is pre-programmed to contain the room number of its respective room.

Knowledge base further asserts that the status of room is *Safe*, and e_1 and e_2 are its exit for locations R_1 and R_2 respectively. Basic perceptual rules $Rule_1$ for robot infer that robot would want to move via some exit *Y* provided it is currently at a location *X* and *X* possesses an exit *Y*. In this case robot s is at location R_1 and R_1 leads to exit locations e_1 and e_2. Thus a natural tendency of robot is to use either of the exit to move to room R_2 or R_3. Here it only refers to robot's willingness to move a certain location. Triggering of a certain rule just updates the belief of a robot to want to move to some location, actual management of movement is controlled via action planning and its associated defeasible reasoning, which will be discussed in the following sections.

Consequently $Rule_2$ infers that robot should not want to move to a room *Y* if the *Exit_Status* of exit *Y* is *Halt*. In other words if a location is halted then robot will not show willingness to move to that particular location. Finally $Rule_3$ provides a conflict resolution rule which is used in a situation in which both $Rule_1$ and $Rule_2$ are triggered. Thus entity would prefer to trigger rule $Rule_2$ over $Rule_1$ on the basis of rule $Rule_3$.

For instance let's consider a simple scenario:

$$\Psi = \left\{ \begin{array}{l} At\ (s, R_1),\ Status\ (R_1, Safe),\ Status\ (R_2, Safe),\ Status\ (R_3, Halt),\ Role \\ (s, ?),\ Exit\ (R_1, e_1),\ Exit\ (R_2, e_1), \\ Exit\ (R_1, e_2),\ Exit\ (R_3, e_2) \end{array} \right\}$$

$Rule_1$*: wants_move(Y) ⊃ at(s, X), Exit (X, Y)*

$Rule_2$*: ~wants_move(Y) ⊃ at(s, X), Exit (X, Y,), Exit_Status(Y, Halt)*

$Rule_3$*:* $Rule_2 > Rule_1$

In this situation, $Rule_1$ triggers the willingness of robot to move to location R_2 but $Rule_2$ infers that robot should not move to R_2. For situations when multiple rules get triggered $Rule_3$ provides conflict resolution. Thus robot would prefer to trigger rule $Rule_2$ over *Rule1* on the basis of rule $Rule_3$. It is important to note that such conflict resolution rules are needed to be non-commutative for all conflict resolution rules to avoid rule cycles.

There can other semantic information in robotic knowledge base as well, such as role of a robot (expressed by *Role (s, ?)* in this example), task assigned and so on. For instance, there can be a rule

$Rule_4$*: definitely_move(Y) ⊃ at(s, X), Exit (X, Y), Role (s, Maintenance).*

This way contextual information can be added by a simple update in rules of rules and factual data. The main idea is to incorporate semantics at a lower level and propagate the intelligence in a bottom-up way.

Using formalism for representing knowledge base of a robot on the basis of rules provides a very flexible way for updating a robot behavior. Updates to rules and factual data provide refutation here. The underlying mechanism for refutation inference is explained in the following sections.

6.4.2. Robotic Action plan Representation in DeLP

As mentioned in previous section that robotic knowledge base representation in DeLP does not entail actual actions performed by robots. DeLP based knowledge base representation only corresponds to belief system of a robot. For instance it only refers to robot's willingness to move a certain location. Triggering of a certain rule just updates the belief of a robot to show willingness to move to a certain location, actual management of movement is controlled via action planning and its associated defeasible reasoning, which will be discussed in the following sections.

This thesis employs argumentation-based action plan formalism [153, 154] for representing robotic actions, which can define entity's actions as a set of preconditions on literals, and warrant condition on constraints. Formally an action can be expressed as:

$$\{Preconditions\},\ not\{Constraints\} \xrightarrow{Action} \{Consequence\}$$

$$\Psi = \Psi - (Consequence)'\ U\ Consequence$$

For instance *move_out* action for robot which is at location *X* is given as:

$$\{\ at(s, X),\ Exit\ (X, Y),\ wants\_move(Y)\} \xrightarrow{move\_out} \{\ at(s, Y),\ \hat{}\ at(s, X)\}$$

LRKB (see Section 2.2) of a robot is augmented with meta-data for facts and rules. Let us assume that robot R_1 is navigating in a room and gets stuck into a corner. Robot's LRKB not only comprises of robotic *TroubleStatus* but also some extra attributes to the associated fact, namely value type, date/time at which the fact is held, identifier of source of fact, and context which identifies a data group or addressee.

6.4.3. Formal concept representation in DeLP

Representation of concepts in robot internal knowledge base is expressed as a set of factual information and rules (as elaborated in previous section). Triggered rules form the preconditions for action formalism of robotic agent. For instance robotic knowledge base of a robot *s* at room R_1 can be expressed by following set of facts. Here $Rule_1$ forms the basic motion trigger for the robotic agent.

$$\Psi = \left\{ \begin{array}{l} At\ (s, R_1),\ Status\ (R_1, SAFE),\ Role\ (s, ?) \\ Exit\ (R_1, e_1),\ Exit\ (R_2, e_1) \\ Exit\ (R_1, e_2),\ Exit\ (R_3, e_2) \end{array} \right\}$$

$Rule_1$: *wants_move(Y) ⊃ at(s, X), Exit (X, Y)*

$$\{ at(s, X),\ Exit\ (X, Y),\ wants\_move(Y)\} \xrightarrow{move\_out} \{ at(s, Y),\ \wedge\ at(s, X)\}$$

Now let's consider a number of case-studies. In first situation, let's consider robot *s* which entered the building in room R_8 using exit *Ex*. Robot navigated to room R_1 after passing R_2.

$$(\{R_2, R_8\}, \{e_3\}) \qquad (\{R_1, R_2\}, \{e_1\})$$

$$\leq \nearrow \quad \nwarrow \leq \qquad \leq \nearrow \quad \nwarrow \leq$$

$$(\{R_8\}, \{e_3, e_6\})(\{R_2\}, \{e_1, e_3\})(\{R_1\}, \{e_1, e_2\})$$

$$(\{R_8\}, \{e_3, e_6\}) \sim (\{R_2, \{e_1, e_3\}\}) \sim (\{R_1, \{e_1, e_2\}\})$$

Thus the path trail for robot from R_1 will go through R_2 which is the nearest neighbor to R_8, as per partial order relation ≤. In this case it is assumed that robot has built up a partial lattice for explored concepts as shown above. Now let's consider a situation in which there is some sort of emergency which triggers an immediate exit of robot, since robot has partial knowledge of the building, the robot will try to trail back the path which it followed to reach current location. Nearest neighbors is represented by a simple predicate 'nearest' in robotic knowledge base.

$$\Psi = \left\{ \begin{array}{l} \ldots \\ Exit\ (R_1, e_1),\ Exit\ (R_2, e_1),\ nearest\ (R_1, R_2) \\ Exit\ (R_2, e_3),\ Exit\ (R_8, e_3),\ nearest\ (R_2, R_8) \\ \ldots \end{array} \right\}$$

If contextual information does not permit robotic movement through the path trail then robot explores the path which converges to the initial path trail of robot. For instance, let us assume that R_2 is on fire, which renders the exits of R_2 to halt. This fact can be represented in knowledge base by the following facts.

$$\Psi = \left\{ \begin{array}{l} At\ (s, R_1),\ Status\ (R_1, SAFE),\ Role\ (s, ?) \\ Exit\ (R_1, e_1),\ Exit\ (R_2, e_1),\ Exit\ (R_1, e_2) \\ Exit\ (R_3, e_2),\ Status\ (R_2, Emergency), \\ Exit\_Status(e_1, Halt)\ .... \end{array} \right\}$$

Rule₁: wants_move(Y) ⊃ *at(s, X), Exit (X, Y)*

Rule₂: ~wants_move(Y) ⊃ *at(s, X), Exit (X, Y,), Exit_Status(Y, Halt)*

Rule₃: Rule₂ > Rule₁

Such contextual information can be on the basis of room status being emergency or exit being halt. Meta-data about facts and rules can also assert priority and provide contextual information for robotic navigation. For instance, a robot might not be allowed to enter a particular room on the basis of location permissions, or a certain rule be given a priority depending upon the source of the rule.

In the above scenario $Rule_2$ has priority over $Rule_1$ thus the robot would choose e_2 and move to room R_3. From R_3 the robot creates a new formal concept $(\{R_3\}, \{e_2, e_4, e_5\})$. Since the robot is already aware of $(\{R_1\}, \{e_1, e_2\})$ it can create a composite concept $(\{R_1, R_3\}, \{e_2\})$, (pseudo code for incremental concept learning is shown in Figure 25). Robot's knowledge base is thus updated with the following nearest neighbor fact:

$$\left(\{R_1\}, \{e_1, e_2\}\right) \leq \left(\{R_1, R_3\}, \{e_2\}\right)$$

Since the robot trail tries to find and optimal path for R_8, the robot navigates to R_4 creating formal concept $(\{R_4\}, \{e_5, e_6\})$ and subsequent formal composite concept $(\{R_3, R_4\}, \{e_5\})$ and associated nearest neighbor relationship. In subsequent navigation the robot uses e_6 to enter R_8 and leaves the building using *Ex*.

6.4.4. Incremental concept learning

Now let's consider a case where a robot has no a priori knowledge. In this case, the robotic agent has to build up formal concepts and their relationships incrementally as it navigates around the building. Pseudo code of incremental concept analysis is given in Fig 25.

In the current case the robot boots at location *R5* and has no a priori information about location hierarchy. Upon receiving identifier of the current room and its exits from a RFID tag, the robot will learn a non formal concept $(\{R_5\}, \{e_8\})$ and its local robotic knowledgebase will be updated by following facts:

At (s, R₅), Exit (R₅, e₈), Exit (R₇, e₈).

Upon warranting of $Rule_1$ for literal *wants_move(e₈)*, the robot triggers *move_out* action from its action repository and reaches location R_7. At R_7 the robot establishes a formal concept $(\{R_7\}, \{e_7, e_8\})$ and forms a composite formal concept $(\{R_5, R_7\}, \{e_8\})$. Then it establishes a nearest relationship between R_5 and R_7 and updates robot's LRKB with learned concepts and relationships. The robot updates its LRKB as it establishes new concepts and their relationships.

Upon any change in LRKB the defeasible reasoner interacts with LRKB and checks for available literals from its rule set. This may trigger the action repository and would potentially lead to actual robotic action by robotic control platform.

Algorithm: Algorithm for creating incremental formal concepts and partial lattice.

Input: Current room 'r', and set S of (R_j, e_i), $e_i \in$ Exits, $R_j \in$ Rooms.
Output: Formal concepts (X, Y); $X \subseteq$ Rooms, $Y \subseteq$ Exits, updated robotic knowledge base facts and partial concept lattice.

Formal_concepts = $\varnothing$, Non_formal_concepts = $\varnothing$ (at Robotic agent boot)
If size(S) = 1 **then**
  Non_formal_concepts = Non_formal_concepts $\cup$ (r, e_i) | $e_i \in (R_j, e_i)$ and $(R_j, e_i) \in S$.
elseif size(S) > 1 **then**
  Formal_concepts = Formal_concepts $\cup$ (r, Y) | Y = $\{e \in (r, e)\ \forall\ (r, e) \in S\}$
end if
Robotic_knowledgebase = Robotic_knowledgebase $\cup$ Exit (X) | $X \in S$
for $((X_1, Y_1), (X_2, Y_2)) \in$ Formal_concepts x Non_formal_concepts and $(Y_2 \subseteq Y_1)$ **do**
  Formal_concepts = Formal_concepts $\cup$ (X', Y') | $X' = (X_1 \cup X_2)$ and $Y' = Y_1 \cap Y_2$
end for
for $((X_1, Y_1), (X_2, Y_2)) \in$ Formal_concepts x Formal_concepts and $(Y_1 \cap Y_2 \neq \varnothing)$ **do**
  Formal_concepts = Formal_concepts $\cup$ (X', Y') | $X' = (X_1 \cup X_2)$ and $Y' = Y_1 \cap Y_2$
end for
for $((X_1, Y_1), (X_2, Y_2)) \in$ Formal_concepts x Formal_concepts and $(X_1 \subseteq X_2$ or $Y_2 \subseteq Y_1)$ **do**
  create partial order relationship
  $(X_1, Y_1) \prec (X_2, Y_2)$
  Robotic_knowledgebase = Robotic_knowledgebase $\cup$ nearest(R_1, R_2) | $R_1 \in X_1$ and $R_2 \in X_2$
end for

Figure 25: Algorithm for incremental formal concepts formation and partial lattice

6.4.5. Dialectical Analysis Algorithm

Dialectical analysis algorithm can be read in greater details in [160]. Here only those portions of the dialectical process are elaborated which are employed in developed defeasible reasoner. In DeLP dialectical analysis is performed to find non-warranted or warranted statements/literals. By definition a literal *h* is considered to be *warranted* if there exists a non-defeated argument structure *(A, h)* for literal *h*. Dialectical analysis is employed to establish whether *(A, h)* is non-defeated or not. *A* is a set of defeaters for literal *h*.

Dialectical process is recursive in finding non-defeated argument structure, as each defeater *D* for any argument structure *A* is itself an argument structure. Thus to find non-defeated argument structure, defeaters for *D* are also considered, and so on. Therefore, to establish a warrant condition for a literal, more than one argumentation line could arise. Such leads to a tree structure that is referred to as *dialectical tree* [160].

Let's consider (A_0, h_0) to be an argument structure from a program *P*. As mentioned before a program comprises of set of facts, defeasible and strict rules. The algorithm for creation of a dialectical tree for (A_0, h_0), denoted $T(A_0, h_0)$, is given in Figure 26.

Algorithm: Algorithm for creating a Dialectical Tree

Input: (A_0, h_0) root of a dialectial tree
Output: Dialectical tree for (A_0, h_0) is formed.

1. Mark the root of the tree with label (A_0, h_0)
2. **For** (A_n, h_n) – non-root nodes **do**
3. **Compute** $\Lambda = [(A_0, h_0), (A_1, h_1), (A_2, h_2) \ldots (A_n, h_n)]$ - the sequence of labels of the path from (A_0, h_0) to (A_n, h_n).
4. **If** Λ != NULL – a path exists from root to (A_n, h_n)
5. **Compute** (B_m, q_m) - all the defeaters for (A_n, h_n)
6. **For** (B_m, q_m) goto line 2 – computer argumentation line $\Lambda' = [(A_0, h_0), (A_1, h_1), (A_2, h_2) \ldots (A_n, h_n), (B_m, q_m)]$
7. **If** Λ' is acceptable, then the node N has a child N_i labeled (B_m, q_m) **end**
8. **If** no (B_m, q_m), then mark N as leaf **end.**
9. **End**

Figure 26: Algorithm for creation of dialectical tree

In a dialectical tree every node, apart from the root node labeled by (A_0, h_0), represents a defeater (proper or blocking) of its parent. Leaves for (A_0, h_0) correspond to non-defeatable arguments for (A_0, h_0).

Each path from the root to a leaf corresponds to one acceptable argumentation line.

Dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is warranted or defeated.

Let's consider a following program, which comprises of facts (*e, i, d, k, d, j*) and defeasible rules ([a ɔ b], [~b ɔ c,d] etc.). This program is taken from [160] for the sake of elaboration purposes.

a ɔ b	~b ɔ e	~b ɔ c, f	~f ɔ i
b ɔ c	e	f ɔ g	i
c	~f ɔ g, h	g	~h ɔ k
~b ɔ c,d	h ɔ j	k	
d	j		

Here, the literal *a* is supported by an argument structure *{(a ɔ b), (b ɔ c)}*. Corresponding defeaters (for literal *b*) and their argumentation lines (B_1, B_2, B_3) is given as following:

$$(B_1,\ \sim b) = (\{(\sim b \supset c,d)\}, \sim b),$$
$$(B_2, \sim b) = (\{(\sim b \supset c, f), (f \supset g)\},\ \sim b),\ and$$
$$(B_3, \sim b) = (\{(\sim b \supset e)\}, \sim b).$$

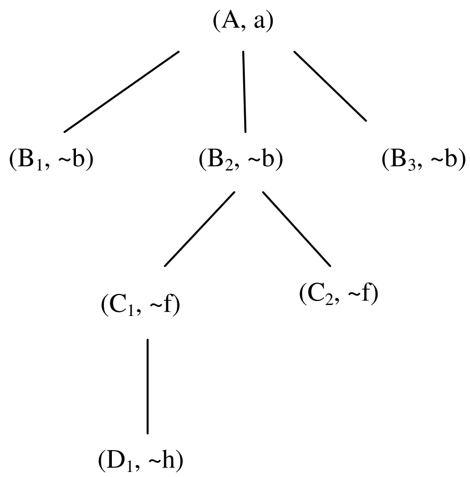


Figure 27: An example Dialectical Tree

In order to decide whether the root of a dialectical tree is defeated, a marking process is used. In marking process nodes of tree are recursively marked as *"D"* (defeated) or *"U"* (undefeated) using algorithm a bottom up algorithm of marking nodes. A node is marked as undefeated if none of its child nodes are undefeated. A single defeated node results in defeat of all its parent nodes. This procedure suggests a bottom-up marking process, through which it is possible to determine the marking of the root of a dialectical tree.

Result of algorithm given in Figure 28 when applied on dialectical tree in Figure 27 results in Figure 29 (a marked dialectical Tree).

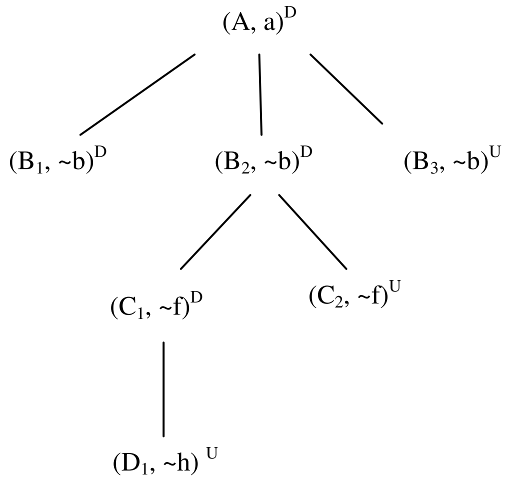


Figure 29: Marked Dialectical Tree from example in Figure 28

Detailed discussion of dialectical analysis algorithm can be read in [160].

6.4.6. Defeasible reasoning engine

Defeasible reasoner uses the dialectical analysis algorithm as elaborated in previous section. For instance, let's consider a literal L_i = *wants_move*(e_1), argument structure can be given as:

> Arg_1 = ({ {*wants_move*(e_1) $\supset$ *at*(*s*, R_1), *Exit* (R_1, e_1)}, *At* (*s*, R_1), *Exit* (R_1, e_1) } , *wants_move*(e_1)).

Thus using factual information from Ψ, *At* (*s*, R_1) and *Exit* (R_1, e_1), the literal *wants_move*(e_1) is warranted (asserted to be true). Please note that facts are always considered as true proposition. Now for L_i = *wants_move*(e_1) the dialectical analysis tries to find an argument structure Arg_2 that contradicts the Arg_1. In other words literal *wants_move*(e_1) can be warranted only if there doesn't exist another argument to warrant ~*wants_move*(e_1).

Arg2 = ({ {~*wants_move*(e_1) $\supset$ *at*(*s*, R_1), *Exit* (R_1, e_1), *Exit_Status*(e_1, *Halt*)}, *At* (*s*, R_1), *Exit* (R_1, e_1), *Exit_Status*(e_1, *Halt*) } , ~*wants_move*(e_1)).

Thus *wants_move*(e_1) cannot be warranted. In this situation literal L_i = *wants_move*(e_2) will be warranted as there is no argument structure which would warrant ~*wants_move*(e_2).

Figure 26 presents a log fragment of developed defeasible reasoner in operation. Lines 21-24 show the warrant condition of literals *wants_move*(e_1) and *wants_move*(e_2) based on facts 10-16. Then the robot learns about the Halt situation of exit e1 (line 31), thus robot infers a conflict for literal *wants_move*(e_1) meaning it can't decide whether to move to e_1 or not (lines 35, 37).

```
10: fact:F1:at(s,R1):status = true
11: fact:F2:Status(R1,Safe):status = true
12: fact:F3:Role(R1,?):status = true
13: fact:F4:Exit(R1,e1):status = true
14: fact:F5:Exit(R1,e2):status = true
15: fact:F6:Exit(R2,e1):status = true
16: fact:F7:Exit(R3,e2):status = true
… …
21: rule:Rule1:wants_move(e1):status = true
22: rule:Rule1:wants_move(e2):status = true
23: rule:Rule2:~wants_move(e1):status = false
24: rule:Rule2: ~wants_move(e2):status = false
… …

… …
31: fact:F8:Exit_Status(e1,Halt):status = true
… …
35: rule:Rule1:wants_move(e1):status = CONFLICT
36: rule:Rule1:wants_move(e2):status = true
37: rule:Rule2:~wants_move(e1):status = CONFLICT
38: rule:Rule2: ~wants_move(e2):status = false

… …
42: rule:Rule1:wants_move(e1):status = false
43: rule:Rule1:wants_move(e2):status = true
44: rule:Rule2:~wants_move(e1):status = true
45: rule:Rule2: ~wants_move(e2):status = false
46: rule:Rule3: Rule2>Rule1:status = true
```

Figure 28: Defeasible reasoner output.

When the robot learns about conflict resolver rule *Rule*$_3$ (line 46), the conflict is resolved by warranting ~*wants_move*(e_1). Thus the robot decides not to move to exit e_1.

Similarly, rules on the basis of meta-data/contextual information can be derived to provide prioritized robotic navigation. For instance, a robot can use nearest predicate and give the stronger connection strength to a certain link. Partial lattice does not incorporate such semantic information; rather those are expressed in terms of rule updates since it is easier to update rules as per situational context.

6.5. Concluding Remarks

This chapter discussed the use of defeasible rules and use of meta-data/contextual information of facts and rules for intelligent robotic navigation. The situational context as robotic permissions, room status and scheduled navigation etc. provides intelligence and adaptive nature to robotic location models.

Here the proposed robotic navigation is on the basis of partial lattice i.e. incomplete information without a priori knowledge of a location model. Apart from that, unlike existing models, developed methodology can easily incorporate semantic information, and expressing such in terms of rule updates, provides a very flexible way of updating robotic control platform.

This, still on-going work has been conducted in the context of a bigger project of semantic discovery and invocation of robotic swarm entities as web services and incorporating the robotic intelligence in communication and collaboration of semantically rich facts and rules. Dialectical analysis based defeasible reasoner is implemented in C++ with design objectives of accommodating meta-data/contextual attributes of communicated facts and rules in robotic reasoning. (i.e. forming context based rules, rule priorities based on timestamps and sources of the rules). Current implementation is small and efficient (11KB memory footprint) to cope with restrictions set on robot execution platforms and it can be easily embedded to robotic execution environment.

Current implementation only considers a single robot building up a semantic location model on the basis of room and exit locations, and defeasible rules which govern the implicit connection strengths. In future work it is intended to incorporate multiple robot interaction, thus robots can learn from each other. One possible extension of work is the incorporation of semantic information by a robot at RFID tags at exit locations in terms of neighborhood i.e. a degree of nearness (nearest neighbor, close_to etc). Such incomplete information can be used by another robot to build up its incremental concept lattice.

6.6. Scientific contributions

1. **Abdul Haseeb**, Mihhail Matskin and Peep Kungus, DeLP based Semantic Location Lattice for Intelligent Robotic Navigation. *Proceedings of International Conference on Artificial Intelligence (ICAI'08), July 2008, Las Vegas, USA.*

7. Conclusions

This thesis focused on interoperability of heterogeneous entities in an infrastructure less paradigm, a distributed WS discovery and management middleware and distributed incremental learning in the absence of global knowledge.

Thesis motivated the need for autonomic publishing and discovery of resources and just-in-time integration for on-the-fly service consumption without any a priori knowledge in mobile dynamic systems. First this thesis highlighted the limitations of SOA and UDDI in the exploitation of Web services in such distributed control and dynamic natured systems. To overcome the limitations, a mediator-based distributed Web services discovery and invocation middleware was designed and developed which provided a collaborative and decentralized services discovery and management middleware for infrastructure-less mobile dynamic systems with heterogeneous communication capabilities. Heterogeneity of communication capabilities was abstracted in middleware by a conceptual classification of computing entities on the basis of their communication capabilities and communication issues were resolved via conceptual overlay formation for query propagation in system.

The developed middleware was evaluated extensively using Player Stage simulator and was practically applied in physical robot swarms as well. Experimental validations included analysis of results in different communication modes i. active and ii. passive mode of communication with and without shared resource conflict resolution. Developed approach overcame the shortcomings of availability of both service requesters and service providers at the same time. Mediator-based distributed Web services discovery and management middleware, though was tested primarily for robot swarms only, can be applied to any infrastructure-less dynamic environment. In current state, semantic expertise of an entity is used for semantic entity selection for query propagation. Ongoing work, though, is extending the middleware with semantic Web services. Moreover, creation of rule-based service discovery is in progress, in which rules will provide the foundation for semantic reasoning to overcome the limitations of using a pre-defined ontology of domain.

Second part of thesis extended common-sense inference for robot swarms. This enabled dynamic adaptation of a system to changing environments with local inference of entities. Formal concept analysis was considered as a case for incremental and distributed learning of formal concepts. Thesis also presented an algorithm for incremental common sense reasoning and applied it for intelligent robotic navigation by building partial lattice i.e. incomplete information without a priori knowledge of location model.

Even though, this thesis mainly focused on interoperability and distributed learning aspects of distributed knowledge management, ongoing research work, builds-upon the results achieved towards establishing a seamless integration in mobile dynamic systems.

References

[1]. Aristotle and Barnes J., (1994), Aristotle Posterior Analytics *(Clarendon Aristotle Series), Oxford University Press, USA; 2nd edition.*

[2]. Alavi, M., Leidner, D. E. (2001). Review: Knowledge Management and Knowledge Management Systems: *Conceptual Foundations and Research Issues. MIS Quarterly, 25(1), pp.107-136.*

[3]. Rolf, B., Two Theories of Tacit and Implicit Knowledge.

[4]. Nonaka, I., Toyama, R., Byosière, P. (2001): A Theory of Organizational Knowledge Creation: Understanding the Dynamic Process of Creating Knowledge. *In Handbook of Organizational Learning and Knowledge. Oxford University Press, New York, pp 491-517.*

[5]. Polanyi, M. (1967). The tacit dimension. *London: Routledge and Kegan Paul*

[6]. Bharat, K., and Broder, A. (1998), A technique for measuring the relative size and overlap of public web search engine. *In 7th WWW Conference, Brisbane, Australina.*

[7]. Berners-Lee, T., Hendler, J., and Lassila, O.(2001), The semantic web. *Scientific American.*

[8]. Davies, N. J. (2000), Knowledge management, *BT Technology Journal, 18, 1, Kluwer, Netherlands.*

[9]. Fischser, G., Ostwarld, J. (2001) Knowledge Management: Problems, Promises, Realities and Challenger. *IEEE Intelligent Systems.*

[10]. Stojanovic, L., Stojanovic, N., Handschuh, S. (2002), Evolution of metadata in the Ontology based Knowledge Management Systems. *FZI, University of Karlsruhe, Experience Management, Berlin.*

[11]. Davies N. J. and Chen, C.(1999), Integrating spatial, semantic and social structures for knowledge management. *In Proceedings HICSS-99, Hawaii.*

[12]. Hearst, M. A. (1998), Information integration. *IEEE Intelligent Systems.*

[13]. Lau, T., Sure, Y. (2002), Introducing ontology-based skills management at a large insurance company. *In Proceedings of the Modellierung, Germany.*

[14]. Abecker, A., Elst, L. van. (2003), Ontologies for Knowledge Management, *in Handbook on Ontologies, S. Staab and R. Studer (editors), Ch. 22, pp. 435-454, Springer.*

[15]. Fensel, D., (2001), Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. *Heidelberg, Germany.*

[16]. Alani, H., Kim, S., Millard, D., Weal, M., Hall, W., Lewis, P., and Shadbot, N., (2003), Automatic ontology based knowledge extraction from web documents. *IEEE Intelligent Systems, 18(1):14–21.*

[17]. Gibbins, N., Harris, S., and Shadbolt, N., (2003), Agent-based semantic web services. *In The Twelfth International World Wide Web Conference (WWW), Budapest, Hungary, ACM Press*

[18]. AgentCities.NET. IST project IST-2000-28384 Agentcities, 2000. http://www.agenticities.net/.

[19]. OntoWeb. IST project IST-2000-29243 OntoWeb, 2002. http://www.ontoweb.org.

[20]. On-To-Knowledge. IST project IST-1999-10132 On-To-Knowledge, 1999. http://www.ontoknowledge.org/.

[21]. Gomez, M., Abasolo, C., and Plaza, E., (2001). Domain-independent ontologies for cooperative information agents. *Lecture Notes in Artificial Intelligence, 2128:118–129.*

[22]. Neches, R., Fikes, R., Finn, T., Gruber, T., Patil, R., Senator, T., Swartout, W. (1991), Enabling Technology for Knowledge Sharing. *AI magazine, vol. 12, no. 3, pp. 36-56, 1991.*

[23]. Grubber, T. (1993), A translation approach to portable Ontology specifications, *Knowledge Acquisiion, 199-220.*

[24]. Staab, S., Studer, R., Schnurr, H. P., Sure, Y. (2001), Knowledge processes and Ontologies. *IEEE Intelligent Systems.*

[25]. Smirnow, A. V., Chandra, C. (2000), Ontology-based Knowledge Management for Cooperative Supply Chain Configuration, *AAAI.*

[26]. Thacker, S., Sheth, A., Patel, S. (2001), Complex relationship for semantic web. *MIT Press.*

[27]. RoboSwarm. (2007). Knowledge Engineering for Interacting Robot Swarms, www.roboswarm.eu

[28]. Bonifacio, M., Bouquet, P., Manzardo, A. (2000), A distributed Intelligence paradigm for Knowledge Management, Bringing Knowledge to the business

process, *In Proceedings of AAAI Spring Symposium, Technical Report SS-00-03, AAAI Press*.

[29]. Oram, A. (2001), Peer-to-Peer Harnessing the Benefits of a Disruptive Technology. *O'Reilly and Associations, Inc*.

[30]. NASA: Collective Intelligence Networks. Http://is.arc.nasa.gov/AR/tasks/ColInt.html.

[31]. Cadoli, M. and Schaerf, M. (1995), Tractable reasoning via approximation, *Artificial Intelligence, 16(4): 18-26.*

[32]. Dalal, M. (1996), Semantics of anytime family of reasoners, *in Wahlster, W. (ed.), Proceedings of ECAI'96, pp. 360-364.*

[33]. Wasserman, R., Resource-bounded belief revision. *Ph.D. Thesis, ILLC, University of Amsterdam.*

[34]. Aberer, K., Cudre-Mauroux, P. (2004), Emergent Semantics Principles and Issues, *IPSI-2004, Fraunhofer.*

[35]. Axton, C., Gear, R., Macehiter, N., Woods, E. (2002), Peer-to-Peer Computing: Applications and Infrastructure.

[36]. Woods, E. (2001), Knowledge Management and Peer-to-Peer Computing: Making connections. *Knowledge Management World. KMWorld, Vol. 10.*

[37]. Crespo, A., Garcia-Molina, H., (2001), Semantic Overlay Networks for P2P Systems. *Google Technologies Inc.*

[38]. IEEE: Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: *A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990*

[39]. Smith, B.; C. Welty. (2001). Ontology: Towards a New Synthesis. Introduction to Formal Ontology in Information Systems: *Collected Papers from the Second International Conference, October 17-19, Ogunquit, Maine, C. Welty and B. Smith, eds.*

[40]. Yovits, M.C., G.T. Jacobi, and Goldstein, G.D. (1962), "Self-Organizing Systems," *Self-Organizing Systems, Washington D.C., McGregor & Werner.*

[41]. Lundh, R., Karlsson, L., Saffiotti, A (2007), Dynamic self-configuration of an ecology of robots, *IEEE International Conference on Intelligent Robots and Systems, IROS 2007, Pp: 3403 – 3409.*

[42]. Amigoni, F., et. al. (2005), “An application of ontology technologies to robotic agents”. *Int. Conf. on Intelligent Agent Technology, IEEE/ACM, pp. 751 - 754.*

[43]. Chatterjee, R., Matsuno, F. (2004), On the need for robot description ontology and scene description ontology for effective rescue operations involving robot platforms. *Proc. SSR ’04 IEEE Intl. Workshop on Safety, Security and Rescue Robotics.*

[44]. Messina, E., et. al. (2005),Statement of requirements for urban search and rescue robot performance standards. *Technical report, Department of Homeland Security Science and Technology Directorate and NIST.*

[45]. Chatterjee, R., Matsuno, F. (2005), Robot description ontology and disaster scene description ontology: Analysis of necessity and scope in rescue infrastructure context. *IEEE Intl. workshop on Safety, Security and Rescue Robotics. 2005, pp. 839-859.*

[46]. Hartanto, R., Hertzberg, J. (2005), “Offering Existing AI Planners as Web Services”. *GI workshop Planen und Konfigurieren, Germany.*

[47]. Marmelstein, R. E. (2002), “Force templates: A blueprint for coalition interaction within an Infosphere”. *Int. Conf. on Intelligent Systems, IEEE, pp. 36-41.*

[48]. DiLeo, J., et. al. (2002), “Integrating Ontologies into multi-agent systems engineering”. *AOIS, 2002, pp.15-16.*

[49]. Uschold, M., Gruninger, M. (2004), Ontologies and semantics for seamless connectivity. *ACM SIGMOD, 33(4). Pp 58-64.*

[50]. Wache, H., et al. (2001), Ontology-Based Integration of Information – A Survey of Existing Approaches. *In Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing, Seattle, WA.*

[51]. Hartanto, R., Hertzberg, J. (2005), “Offering Existing AI Planners as Web Services”. *GI workshop Planen und Konfigurieren, Germany.*

[52]. Vasiliu, L., et. al. (2006), “A Semantic Web Services driven application on Humanoid Robots”. *4th Int. workshop on Software Technologies for Future Embedded & Ubiquitous Systems, IEEE, 2006, pp. 236 – 244.*

[53]. Levine, J., Vickers, L., (2001), “Robots Controlled through Web Services”. *Technogenesis Research Project.*

[54]. Clement, L., et. al (2004), Universal Description Discovery and Integration (UDDI) 3.0.2. http://www.uddi.org/.

[55]. Maamar, Z., et al. (2007), "Dynamic management of UDDI registries in a wireless environment of web services: Concepts, architecture, operation, and deployment". *Journal of Intelligent Information Systems, Springer Netherlands, pp. 105-131.*

[56]. Kim, B. K., et al (2005), "Web Services based robot control platform for ubiquitous functions", *Proc. of Int. Conf. on Robotics and Automation, IEEE, pp. 691- 696.*

[57]. Pollock, J. (2002), Integration's dirty little secret: "It's a matter of Semantics". *Whitepaper; Modulant, The Interoperability Company.*

[58]. Marmelstein, R. E. (2002), "Force templates: A blueprint for coalition interaction within an Infosphere". *Int. Conf. on Intelligent Systems, IEEE, pp. 36-41.*

[59]. DiLeo, J., et. al. (2002), "Integrating Ontologies into multi-agent systems engineering". *AOIS, 2002, pp.15-16.*

[60]. Araki, M., Miyajima, A. (2006), Application of Ubiquitous Web Technologies to Home Information Appliances. *W3C.*

[61]. Johnson, M., et. al., (2006), Jeffery Bradshaw, Paul Feltovich, Renia Jeffers, Hyuckchul Jung, Andrzej Uszok. A Semantically rich policy based approach to robot control. *Proceedings of the International Conference on Informatics in Control, Automation and Robotics.*

[62]. Schlenoff, C., (2005), Robots and capabilities classifications. *Public Forum Presentation.*

[63]. Schlenoff, C., Messina, E., (2005). A robot ontology for urban search and rescue. *ACM workshop on Research in knowledge representation for autonomous systems.*

[64]. Leonhardt, U. (1998). Supporting location-awareness in open distributed systems. PhD Thesis, *Departement of Computing, Imperial College London.*

[65]. Hu, H., Lee, D. (2004). Semantic location modeling for location navigation in mobile environment. *In the proceedings of IEEE 5th International Conference on mobile data management, pages 52 - 61.*

[66]. Karlsen, B.S. (2006). Enabling a Ubiquitous Location Based Service on Campus. *Ms. Thesis, Norwegian University of Science and Technology.*

[67]. Lee, D., Zhu, M., Hu, H. (2005) When location-based services meet databases. *In the proceedings of mobile information Systems, Pages: 81 - 90.*

[68]. Brumitt, B., Shafer, S. (2001). Topological world modeling using semantic spaces. *In the proceedings of Location Modeling Workshop at Ubicomp.*

[69]. Bussler, C., (2005) “A minimal Triple Space Architecture” *2nd WSMO Implementation workshop, Innsbruck.*

[70]. Amigoni F., et. al. (2005), “An application of ontology technologies to robotic agents”. *Int. Conf. on Intelligent Agent Technology, IEEE/ACM, pp. 751 - 754.*

[71]. McCarthy, J., Hayes, P.J., (1969), Some philosophical problems from the standpoint of Artificial Intelligence. *In Machine Intelligence. Vol. 4. pp 463-502. Edingburgh University Press, Edinburgh, UK.*

[72]. Clement, L., et. al. (2004), Universal Description Discovery and Integration (UDDI) 3.0.2. http://www.uddi.org/.

[73]. Maamar, Z., et al (2007), Dynamic management of UDDI registries in a wireless environment of web services: Concepts, architecture, operation, and deployment”. *Journal of Intelligent Information Systems, Springer Netherlands, pp. 105-131.*

[74]. Yriarte, L., Deplanques, P., Sallantin, J., Zapata, R. (1996). An ontology for robot modelling and testing.

[75]. Ghallab, M., (1994). Past and future chronicles for supervision and planning. *In J.P. Haton, editor, Proceedings of the 14th Int. Avignon Conference, pages 23-34. EC2 and AFIA.*

[76]. Cohen, P.R., Carole R. Beal. (2000) Natural Semantics for a Mobile Robot. *University of Massachusetts Computer Science Department Technical Report 00-59*.

[77]. Long M., and Murphy, R., A proposed approach to Semantic Integration between Robot and Agent Systems.

[78]. Johnson, M., Bradshaw, J., Feltovich, P., Jeffers, R., Jung, H., and Uszok, A., (2006), A Semantically rich policy based approach to robot control. *Proceedings of the International Conference on Informatics in Control, Automation and Robotics.*

[79]. Bradshaw, J. M., et al, (2004). Making agents acceptable to people.

[80]. Araki, M., and Miyajima, A., (2006), Application of Ubiquitous web technologies to home information appliances. *W3C*.

[81]. Wang, E., Kim, Y.S., Kim, S.A., (2005) An object Ontology using Form-Function reasoning to support robot context handling. *Computer-Aided Design and Applications, Vol. 2, No.6, pp. 815-824*.

[82]. Mendoza,R., and Williams, M., (2005), Ontology based object categorization for robots. *Proceedings of the australasian Ontology Workshop - Volume 58*

[83]. Vogt, P., (2003), Anchoring of semiotics symbols. *Robotics and autonomous systems, vol. 43, pp. 109-120.*

[84]. Cheol Go, Y., Sohn, J., (2005) Context modeling for intelligent robot services using rule and ontology. *The 7th International Conference on Advanced Communication Technology, ICACT.*

[85]. Provine, R., Schlenoff, C., Balakisrsky, S., Smith, S., Uschold, M., (2004), Ontology-based methods for enhancing autonomous vehicle path planning. *Robotics and Autonomous Systems, Volume 49.*

[86]. Amigoni, F., and Neri, M.A., (2005), An application of ontology technologies to robotic agents. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology.*

[87]. Chatterjee R., Takao, I., Matsuno, F., Tadokoro, S., (2005), Robot Description Ontology and bases for surface locomotion evaluation. *In IEEE International workshop on Safety, Security and Rescue Robotics, Kobe, Japan.*

[88]. Schlenoff, C., Messina, E., (2005), A Robot Ontology for urban search and rescue. *In the proceedings of the ACM workshop on Research in knowledge representation for autonomous systems.*

[89]. Artale, A., Franconi, E., (1999), Representing a robotic domain using temporal description logics. *AI for Engineering design, Analysis and manufacturing.*

[90]. Artale, A., Francone, E. (1998), A temporal logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research 9, pp 463 – 506*

[91]. Lifschitz, V. (1986), On the semantics of STRIPS. *The Workshop on Reasoning about actions and plans. pp. 1-10. Morgan Kaufman, Los Altos, California.*

[92]. McCarthy, J., Hayes, P.J. (1969), Some philosophical problems from the standpoint of Artificial Intelligence. *In Machine Intelligence. Vol. 4. pp 463-502. Edingburgh University Press, Edinburgh.*

[93]. Sandewall, E., Shoham, Y. (1994), Non-monotonic temporal reasoning. In Handbook of AI and Logic Programming. *Oxford University Press, Oxford, UK.*

[94]. Cohen, P. R., Beal, C.R., (2000), Natural Semantics for a Mobile Robot. *University of Massachusetts Computer Science Department Technical Report 00-59*

[95]. Long, M., and Murphy, R. (2005), A proposed approach to Semantic Integration between Robot and Agent Systems. *American Association of Artificial Intelligence.*

[96]. Long, M. (2004), Creating a distributed field robot architecture for multiple robots. *Master's thesis, University of South Florida.*

[97]. Cai, M., Frank, M., (2004), "RDFPeers: A Scalable Distributed RDF Repository Based on a Structured Peer-to-Peer Network". *In Intl. World Wide Web Conference, ACM, pp. 650-657.*

[98]. Hartman, F., et. al. (2004), "Was the Universal Service Registry a Dream?" *Web Services Journal.*

[99]. Thaden, U., Siberski, W., and Nejdl, W., (2003) A Semantic Web based Peer-to-Peer Service Registry Network, *Technical Report, Learning Lab Lower Saxony,*

[100]. Schmidt, C., and Parashar, (2003) M., A Peer-to-Peer Approach to Web Service Discovery,*World Wide Web, Internet and Web Information Systems, Kluwer Academic Publishers.*

[101]. Stanescu, E., Stanescu, I., and Popa, V., (2005) "Distributed Infrastructure for Semantic Web Services", *Technical Report, National Institute for Research and Development in Informatics, Bucharest.*

[102]. Papazoglou, M.P., et. al., (2003) "Leveraging Web-Services and Peer-to-Peer Networks", *In Proc. of 15th Intl. Conf. on Advanced Information Systems Engineering (CAiSE), Springer, pp. 485-501.*

[103]. Du, Z., et. al., (2006) "Ad-UDDI: An Active and Distributed Service Registry", *VLDB Intl. workshop on Technologies for EServices, LNCS, 2006, pp. 58-71.*

[104]. Sheth, A., and Larson, J., (1990), "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM Computing Surveys, 22 (3), pp. 183-236. Japanese translation appears in special issue of BIT "Computer Science,'90", Japan.*

[105]. Benatallah, B., et. al., "Declarative Composition & Peer-to-Peer Provisioning of Dynamic Web Services". *Proc. of 18th Int. Conf ICDE, IEEE, San Jose, pp. 297-308.*

[106]. Wang, Z., and Hu, Y., (2007), "A P2P Network Based Architecture for Web Service", *Proc. of Int. Conf. on Wireless Communications, Networking and Mobile Computing, IEEE, pp. 3446-3449.*

[107]. Banerjee, S., et. al., (2005), "Scalable Grid Service Discovery based on UDDI", *3rd Int. workshop on Middleware for grid computing, ACM, Grenoble, pp. 1-6.*

[108]. Pantazoglou, M., et. al, (2006), "A unified approach for the discovery of Web and Peer-to-Peer systems". *Proc. of ICWS'06, IEEE, pp. 901-902.*

[109]. DiLeo, J., et. al. (2002), "Integrating Ontologies into multi-agent systems engineering". *AOIS, 2002, pp.15-16.*

[110]. Sun Microsystems Inc., "JXTA Technology-Overview, http://www.sun.com/software/jxta/", June 2008.

[111]. Sun Microsystems Inc., "JXTA-SOAP bindings, https://soap.dev.java.net/", June 2008.

[112]. Stoica, I., et. al., (2001) "Chord: Scalable P2P lookup service for internet applications", *SIGCOMM, ACM, 2001, pp 149-160.*

[113]. Ratnasamy, S., et. al., (2001) "A scalable content-addressable network", *SIGCOMM, ACM, pp. 161-172.*

[114]. Rowstron, A., et. al. (2001), "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", *IFIP/ACM, 2001, pp. 329-350.*

[115]. Carriero, N., and Gelernter, D., (2001), A Computational model of Everything. *CACM, pp 77-81, 2001.*

[116]. Sapkota, B., et al., (2006), Distributed Web Service Discovery Architecture, *In Proc. of the Advanced Int'l Conf on Telecommunications & Int'l Conf. on Internet and Web Applications and Services, pp 136, 2006.*

[117]. Li, Y., et. al., (2003) "An approach for measuring semantic similarity between words using multiple information sources", *Transaction on Knowledge and Data Engineering, July 2003, 15(4): pp. 871-882.*

[118]. Rada, R., Mili, H., Bicknell, E., and Blettner, M., (1989), Development and application of a metric on semantic nets. *In IEEE Transaction on Systems, Man and Cybernetics, pp 17-30.*

[119]. Sun Microsystems Inc. Jini Network Technology, Available at http://www.sun.com/software/jini/specs/

[120]. Lin, Q., Rao, R., and Li, M., (), DWSDM: A Web Services Discovery Mechanism Based on a Distributed Hash Table. *In the Proceedings of 5th Intl. IEEE Conference. Pp 176-180.*

[121]. Montebello, M., et al. (2003),DAML enabled Web services and Agents in the Semantic Web, *Springer Berlin LNCS, Web, Web-Services, and Database Systems, pp 46-58*.

[122]. Haseeb, A., Küngas, P., & Matskin, M., (2008), "Semantic Middleware for Robotic Swarm Interaction". *Proc. of IASTED MIC'08, IASTED, Innsbruck.*

[123]. Haseeb, A., Matskin, M., & Küngas, P., (2008), "Light-Weight Decentralized Autonomic Web Service Discovery for Systems with Heterogeneous Communication Capabilities", *Proc. of 12th IASTED IMSA'08, IASTED, Hawaii.*

[124]. Gerkey, B.P., et. al., (2003), "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems". *11th Int. Conf. on Advanced Robotics (ICAR), Coimbra, pp. 317-323.*

[125]. Web Services Description Language (WSDL) Version 2.0. (2006) Candidate Recommendation, W3C, January 2006. Available at http://www.w3.org/TR/wsdl20/

[126]. Web Services Description Language (WSDL) Version 2.0 (2007): RDF Mapping, W3C Working Group Note, June 2007, Available at http://www.w3.org/TR/2007/NOTE-wsdl20-rdf-20070626/

[127]. Kopecký, J., (2006), WSDL RDF Mapping: Developing Ontologies from Standardized XML Languages, In Proc. of 1st International Workshop on Ontologizing Industrial Standards, pp 312-322, 2006.

[128]. http://playerstage.sourceforge.net/doc/Player-2.1.0/player/classPlayerCc_1_1Position2dProxy.html

[129]. W3C, Parsing OWL in RDF/XML, W3C working group note (2004), Available at http://www.w3.org/TR/owl-parsing/#sec-rdf

[130]. W3C, OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation (2004), Available at http://www.w3.org/TR/owl-semantics/

[131]. Paolucci, M., et. al., (2002), "Importing the Semantic Web in UDDI" Web Services, E-Business and Semantic Web Workshop (CAiSE), Springer, 2002, pp. 815-821

[132]. Haseeb, A., Matskin, M., and Küngas, P., (2008), "DeLP based Semantic Location Lattice for Intelligent Robotic Navigation", *Proc. of ICAI', Las Vegas, USA.*

[133]. Li, Y., et. al., (2003), "An approach for measuring semantic similarity between words using multiple information sources", *Transaction on Knowledge and Data Engineering, July 2003, 15(4): pp. 871-882.*

[134]. Pollock, J.L. (1999). Cognitive carpentry: A blueprint for how to build a person. *The Knowledge Engineering Review, vol. 14, Issue 1, Pages: 91 – 95.*

[135]. Garcia, A.J., Simari, G.R., (2002), "Defeasible Logic Programming An Argumentation Approach". *Journal of Theory and Practice of Logic Programming.*

[136]. Nute, D., (2001), Defeasible Logic, *INAP, 2001, pages: 87-114.*

[137]. Nute, D., and Erk, K., (1996), Defeasible Logic Graphs for Decision Support. *HICSS (2) 1996: 11-19*

[138]. Pollock, J. L., (1987), Defeasible Reasoning. *Cognitive Science 11(4): 481-518 (1987)*

[139]. Pollock, J.L., (1996), A General-Purpose Defeasible Reasoner. *Journal of Applied Non-Classical Logics 6(1).*

[140]. Pollock, J.L., (2000), Defeasible Reasoning in OSCAR CoRR *cs.AI/0003012.*

[141]. Pollock, J.L., (2001), Defeasible reasoning with variable degrees of justification. *Artif. Intell. 133(1-2): 233-282.*

[142]. Ganter, B., Wille, R., (1999), Formal concept analysis: Mathematical Foundations. *Springer-Verlag*.

[143]. Davey, B., Priestley, H., (1990), Introduction to Lattices and Order. *Cambridge University Press*.

[144]. Gratzer, G., (1998), General Lattice Theory. *Birkhauser Verlag*.

[145]. Li, D., Lee, D.L., (2008), A lattice-based semantic location model for indoor navigation. *In the Proc. Of 9th Intl. Conf. of Mobile data management*.

[146]. Leonhardt, U., (1998), Supporting location-awareness in open distributed systems. *PhD Thesis, Departement of Computing, Imperial College London*.

[147]. Brumitt, B., Shafer, S., (2001), Topological world modeling using semantic spaces. *In Proc. Of Location Modeling Workshop at Ubicomp*.

[148]. Hu, H., Lee, D.L., (2004), Semantic location modeling for location navigation in mobile environment. *In Proc. Of 5th Int. Conf on mobile data management.*

[149]. Karlsen, B.S., (2006), Enabling a Ubiquitous Location Based Service on Campus. *Ms. Thesis. Norwegian University of Science and Technology.*

[150]. Lee, D.L., Zhu, M., Hu, H., (2005), When location-based services meet databases. *Mobile information Systems*.

[151]. Hu, H., Lee, D.L., (2004), Semantic Location modeling for location Navigation in Mobile Environments. *In Proc. of IEEE intl. Conf on Mobile Data Management.*

[152]. Li, D., Lee, D.L., (2008), A lattice-based semantic location model for indoor navigation. *In the Proc. Of 9th Intl. Conf. of Mobile data management.*

[153]. Pollock, J, (1999), "Cognitive carpentry: A blueprint for how to build a person". *The Knowledge Engineering Review Vol 14 , Issue 1.*

[154]. Garcia, A.J., Simari, G.R., (2002), Defeasible Logic Programming An Argumentation Approach. *Journal of Theory and Practice of Logic Programming.*

[155]. Prakken, Henry and Vreeswijk, (2002), "Logical systems for defeasible argumentation". *Handbook of Philosophical Logic 2002.*

[156]. Chesñevar, C., Maguitman, A., and Chesnevar, R. L. (2000), Logical models of Argument. *ACM Computing Surveys Vol. 32, pages "337-383".*

[157]. Garcia, A., Gollapally, D., Tarau, P., and Simari, G., (2000), Deliberative stock market agents using jinni and defeasible logic programming, *In Proc. of the ECAI Workshop on Engineering Societies in the Agents' World, Berlin, Germany, Springer Verlag.*

[158]. Antoniou, G., and Bikakis, A., (2004), A Defeasible Logic Programming System for the Web. *In the Proc of 16th IEEE International Conference on Tools with Artificial Intelligence.*

[159]. Simari, G.R., and Loui, R.P., (1992), A Mathematical treatment of defeasible reasoning and its implementation, *Artif. Intell. 53(2-3): 125-157.*

[160]. Garcia, A., Simari, G.R., (2004), Defeasible logic programming: an argumentative approach, *Theory and Practice of Logic Programming (2004), 4:95-138 Cambridge University Press.*