



UNIVERSITY OF
PATRAS
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

School of Natural Sciences
Department of Mathematics

An Extended Web Service Composition Model with Teamwork Semantics

Paraskevi A. Tsoutsa

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

Patra, 2021



University of Patras

Department of Mathematics

Paraskevi A. Tsoutsa

Copyright © 2021 - All rights reserved

E-MAIL: ptsoutsa@uth.gr



PhD thesis

**An Extended Web Service Composition Model
with Teamwork Semantics**

Paraskevi A. Tsoutsa

Thesis Committee:

Assistant Professor Omiros Ragos, Advisor

Professor Panos Fitsilis

Professor Panagiotis Pintelas

Patra, 2021

Η παρούσα Διδακτορική Διατριβή με τίτλο “**Ένα διευρυμένο μοντέλο σύνθεσης υπηρεσιών ιστού με συνεργατική σημασιολογία**”, που εκπονήθηκε από την Παρασκευή Α. Τσούτσα, εξετάσθηκε και εγκρίθηκε από την ακόλουθη Επταμελή Εξεταστική Επιτροπή:

Όμηρος Ράγγος, Επίκουρος Καθηγητής
Τμήμα Μαθηματικών
Πανεπιστήμιο Πατρών

Παναγιώτης Φιτσιλής, Καθηγητής
Τμήμας Ψηφιακών Συστημάτων
Πανεπιστήμιο Θεσσαλίας

Παναγιώτης Πιντέλας, Καθηγητής
Τμήμα Μαθηματικών
Πανεπιστήμιο Πατρών

Αχιλλέας Καμέας, Καθηγητής
Σχολή Θετικών Επιστημών και Τεχνολογίας
Ελληνικό Ανοικτό Πανεπιστήμιο

Ιωάννης Ζαχαράκης, Καθηγητής
Τμήμα Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Πανεπιστήμιο Πελοποννήσου

Βασίλειος Γερογιάννης, Καθηγητής
Τμήμα Ψηφιακών Συστημάτων
Πανεπιστήμιο Θεσσαλίας

Χρήστος Πιερρακέας, Αναπληρωτής Καθηγητής
Τμήμα Διοικητικής Επιστήμης
και Τεχνολογίας
Πανεπιστήμιο Πατρών

Αφιερωμένο στους
αγαπημένους μου Παναγιώτη και Μαριλένα

Dedicated to my beloved husband and daughter.

Abstract ¹

Web services and service based architectures have created opportunities to establish more flexible collaboration among different domains, such as smart cities, smart factories, resource management, intelligent transportation, health and many other. The aim of research efforts around web services is mainly to facilitate their handling and composition. Service composition constitutes a challenging task of service provisioning as it can produce new composite services with features not present in the individual ones. Also, service composition can serve applications or users on-demand basis, so the ability to efficiently select and integrate heterogeneous services at runtime becomes an important requirement to the Web service provision. This is also due to the fact that, by automatically composing services, their capabilities can be extended at runtime, therefore theoretically an unlimited number of new services can be created from an existing set of components, thus making applications no longer restricted to the original set of operations specified and envisioned at the design time.

Initial web service efforts failed to hold the promise for providing generic mechanisms that support discovering, composing, evaluating, executing and monitoring web services in dynamic environments, especially with the issues raised by the development of new technologies such as Cloud Computing, Internet of Things and Cyber Physical Systems. Semantic technologies provide the tools to address various challenges related to the web services technology, particularly those to be performed in the dynamic mode. However, in order to realize this potential, significant hurdles still have to be overcome. To deal with these hurdles, an improvement of the process of service composition becomes necessary, especially a higher degree of autonomy. Following this direction, service composition has to proceed autonomously, which on the one hand side reduce human involvement to a minimum, but on the other side require certain capabilities on the part of the mechanisms that will support it.

This research considers the Web service composition problem from the viewpoint of teamwork and addresses the challenge of developing teamwork services, i.e. autonomously composite services capable of exhibiting teamwork behavior. A service capable of exhibiting teamwork behavior is one that can effectively cooperate with multiple potential teammates on a set of collaborative tasks to achieve the overall team goal. Firstly, we lay down

the foundations and describe the interaction of services by exploiting the role Model theory. The description of services with roles and behaviors act as mediator between the different Web service languages and a formal description that will enable the reasoning over new paths of collaboration for composition purposes. Then, we benefit from the teamwork theory, which has spanned diverse disciplines from business management to cognitive science, human discourse, and distributed artificial intelligence and contribute by determining the dominant teamwork roles that prevail during service group cooperation. Novel service teamwork roles are developed and the role modeling approach is extended to enable web services to automatically cooperate by enhancing service's interoperability. These roles aim to (i) enable services to manipulate the uncertainties by being proactive during their participation in specific events, (ii) capture potential emergent behavior that happens in the domain during service design and execution, (iii) arrest and handle any misbehaving of other participants that occurs in the domain, (iv) act as the glue needed to reconfigure and combine services appropriately in dynamic and unpredictable environments.

The teamwork role model becomes especially important in dynamic environments since it enables the development of more efficient composite services, which are stable teams of services such that their components-members promote better understanding and are able to cooperate effectively to achieve their objectives. Their teamwork behavior help them to manipulate the uncertainties by enabling proactiveness during their participation and by capturing emergent behavior during the process of composition, in which services may have neither complete nor correct view about their world or other services, that not only have changeable actions and goals but they are also subject to failure from external events.

¹ The reader can find an extended Greek Abstract at the end of this thesis

Acknowledgments

The intense process of obtaining a PhD degree turned out to be a hard and long one and I would never have been able to finish it without the guidance and unlimited support of some people, whom I would like to express my sincere thanks. First and foremost, I would like to thank the members of the advisory committee. I express my gratefulness and appreciation to my advisor, Assistant Professor Omiros Ragos, faculty member of the University of Patras, for his advices, the fruitful collaboration during all this years, his patient review of all my articles and dissertation numerous times and for providing me support during this journey. Additionally, I would like to wholeheartedly thank and express my deepest gratitude to Proffessor Panos Fitsilis, faculty member of the University of Thessaly, for his excellent guidance, caring, constant encouragement and for providing me support when needed to take critical decisions all these years. Without his support and patience, this work would have never come into existence. I also want to deeply thank Proffessor Panagiotis Pintelas for his valuable comments at key stages of my doctoral research and his scientific and technical support, especially during the last years of my doctoral research. In my defense I feel very honoured to have Professor Achilles Kameas, Professor Ioannis Zacharakis, Professor Vassilis Gerogiannis and Associate Professor Christos Pierakeas as members of my graduation committee. I want to thank you for accepting the invitation and for your constructive comments on my thesis. Undoubtedly, I owe a great deal of gratitude to all the colleagues and friends that encouraged and supported me morally throughout this journey.

Ultimately, I would like to express my respect and love to my parents, who, from the very childhood, taught me how to take responsibility and have always understood me for not having spent with them the time I wanted. The greatest gratitude, however, is rightfully owed to my family, Panagiotis and Marilena, for their endless, selfless and invaluable love and support, for their understanding and patience, for having always been near me throughout these years of study and hard work.

Vivi Tsoutsas
November 2020

Table of contents

1	Introduction	1
1.1	Why Composite Services?	1
1.2	Challenges and Aim	4
1.3	Thesis approach	10
1.4	Research Questions	13
1.5	Thesis Contribution	15
1.6	Thesis Structure	16
2	Background and Related Work	19
2.1	Web Services	20
2.1.1	Service Definition	20
2.1.2	Service Description and Modeling	21
2.1.3	The Many Approaches for Service Composition	24
2.2	Role Modeling	29
2.3	Teamwork	36
2.3.1	Different Approaches of Teamwork Modeling	36
2.3.2	Human Teamwork	37
2.3.3	Human Based Teamwork Models	38
2.3.4	Integration of the Roles of the Major Human Teamwork Models	43
2.3.5	Teamwork Theory to Agent Based Models	46
2.4	Discussion	51
3	Modeling Services with Role Models	61
3.1	Focusing on the Problem - Introduction	61
3.2	Role Modeling for Service Domains	63
3.3	Graphical Notation	65
3.4	Methodology for Establishing a Role Domain	68
3.5	Role Models Formally	69
3.5.1	The Upper Ontology of Role Models	71

3.5.2	State Changes of a Role Domain	72
3.5.3	Mathematical Definition of the RM and its Components	73
3.5.4	Web Service and the Composition Process	75
3.5.5	Summarize the Notation	76
3.5.6	Running Example	77
3.6	The Standard Algorithm for the Composition in $\varrho\Delta$	78
3.7	The Expressive Power of Role Models	81
4	Teamwork Behavior	85
4.1	Agent Based Teamwork Modeling	86
4.1.1	Identification of Agent Based Teamwork Factors	86
4.1.2	Clustering of the Teamwork Factors	90
4.2	Enhancing Teamwork Behavior of Services	95
4.2.1	Bridge Teamwork Factors to Service Teamwork Roles	96
4.2.2	Dynamics of Teamwork Roles during Service Composition	97
4.2.3	Using Design Patterns for Creating the Teamwork Roles	98
4.2.4	The Developer role.	99
4.2.5	The Organiser role.	102
4.2.6	The Monitor role	104
4.2.7	The Advisor role	107
4.2.8	The Innovator role.	110
4.2.9	The Promoter role.	112
4.2.10	The Producer role.	114
4.2.11	The Maintainer role.	116
4.3	Teamwork oriented Composition in Role Domains	117
4.3.1	Description of the Composition using Teamwork Roles	117
4.3.2	Integrated SD for the Composition with Teamwork Roles	119
4.4	Conclusions	121
5	An Ontology for Teamwork Enabled Services	127
5.1	Introduction	127
5.2	Foundations for the TrEW Service Ontology	130
5.2.1	The core Competencies for the Ontology	130
5.2.2	The Content Ontologies	132
5.3	TrEW Service Ontology Construction	139
5.3.1	An Ontological Specification of Teamwork Enabled Services	139
5.3.2	Rules within a Role Domain	140
5.3.3	Restrictions within a Role Domain	140

5.4	Ontology implementation	141
5.5	Discussion	142
6	Teamwork Based Composition in Smart City Domains	145
6.1	Smart cities' Services Challenges	146
6.2	The «Bus Transportation» Use Case	148
6.3	Smart City Nexus Services	149
6.3.1	Smart and Sustainable City Architectures	152
6.3.2	The Teamwork Layer	153
6.4	A Computational Framework	156
6.5	Role Modeling of a Smart City Domain	159
6.6	Bus Transportation Representation	161
6.7	The TrEWS Population	164
6.8	Reasoning over Behavior Adaptation	166
6.9	Discussion	167
7	Conclusions and Future Research	171
7.1	Fulfillment of the Thesis Goals	172
7.2	Limitations and Open Problems	177
7.3	Lessons Learned	177
7.4	Future Research	178

Keywords: Web Services, Microservices, Composition of Services, Service Teamwork, Role Modeling, Team, Agent Teamwork, Collaboration, Synergy, Groups and Teams, Group Performance, Team Decision Making, Business Information Systems.

List of Figures

1.1	Distributed services appearing in industry domains, cloud and www	5
1.2	Novel service implementations consisted of local and web services	12
2.1	Top level of the service ontology [Martin et al., 2007].	24
2.2	Classification of web services composition approaches [Nacer and Aissani, 2014].	26
2.3	Overview of a joint human-robot team decision making process [Bosse et al., 2015].	33
2.4	Mental Models [Jonker et al., 2011].	49
3.1	Graphical architectural elements of Web service composition by using Role Models. Behaviors that may contribute to accomplish the overall goal need to be discovered.	64
3.2	Elements that constitute the RM diagram of a domain.	67
3.3	The methodology steps for establishing and utilising the Role Domain . .	70
3.4	The UML class diagram for the RM ontology.	71
3.5	The behavior conceptualization.	74
3.6	The RM of the role domain.	77
3.7	The service repository $S = \{ser_1, ser_2, ser_3, ser_4, ser_5, ser_6, ser_7, ser_8\}$	78
3.8	The <i>precondition</i> and <i>effect</i> of each atomic <i>behavior</i> are stated.	79
3.9	The composition request for a composite behavior b_x	79
3.10	A solution for the composite behavior b_x	79
4.1	Wheel of factors that affect team working	95
4.2	The Sequence Diagram for the Developer role.	102
4.3	The Sequence Diagram for the Organiser role.	105
4.4	The Sequence Diagram for the Monitor role.	107
4.5	The Sequence Diagram for the Advisor role.	110
4.6	The Sequence Diagram for the Innovator role.	113
4.7	The Sequence Diagram for the Promoter role.	115

4.8	The Sequence Diagram for the Producer role.	116
4.9	The Sequence Diagram for the Maintainer role.	118
4.10	The Sequence Diagram for the composition with all involved roles.	120
5.1	TrEWSOnto Ontology based on its five building blocks [Tsoutsas and Ragos, 2020].	138
6.1	The bus transportation service recruited with services that exhibit teamwork behavior to enable the goal achievement. The presented model in four district states. (A) The initial situation, all students are at home. (B) A school bus on its route to the school. (C) After the bus brakes down, another bus will take students to school. (D) The goal situation achieved, all students are at school, the broken bus is going to get maintained.	150
6.2	Integration of operational and teamwork behaviors.	151
6.3	Smart and Sustainable City Architecture ITU-T [Focus Group Technical Specifications, 2015]	154
6.4	Smart and sustainable city service nexus – team working layer.	155
6.5	The framework for the service composition.	157
6.6	Part of the role model of the smart city domain without teamwork roles.	160
6.7	Part of the role model of the smart city domain.	162

List of Tables

2.1	Modeling with Roles	35
2.2	Table with mapped roles of selected models to Margerison model [Tsoutsas et al., 2019].	45
4.1	Brief description for each teamwork factor and the studies it appears (1/3).	87
4.2	Brief description for each teamwork factor and the studies it appears (2/3.)	88
4.2	Brief description for each teamwork factor and the studies it appears (3/3).	89
4.3	Agent teamwork factors for each teamwork service role	97
5.1	The five ontology modules of TrEW services.	132
5.2	Linked classes in TrEWS ontology.	139
6.1	Roles and behaviors of the smart bus service	163
6.2	Preconditions and effects for each behavior execution	163

Αρχή ήμισυ πωτός.

Πλάτων, 427-347 π.Χ.

The beginning is the most important
part of the work.

Plato, 427-347 b.C.

Ἡ ἰσχὺς ἐν τῇ ἐνώσει.
Αἰσωπος, 620-560 π.Χ.

Union gives strength.
Aesop, 620-560 b.C.

Chapter 1

Introduction

Contents

1.1 Why Composite Services?	1
1.2 Challenges and Aim	4
1.3 Thesis approach	10
1.4 Research Questions	13
1.5 Thesis Contribution	15
1.6 Thesis Structure	16

1.1 Why Composite Services?

What if an answer about the ability of a composite service to perform a complex task, having an added value we are looking for, is hidden within the myriad of available existing web services in the web? Web services are organized in the web by the same principles that guide distributed networks and in fact, services are considered to be structured in a mesh rather than a hierarchy. Throughout that mesh, we could have access to some of the greatest solutions, which they might uniquely give valuable information and solutions to existing problems. *Artificial Intelligence* and *Teamwork theory* are able to connect the correct paths between available services to identify and suggest novel solutions that otherwise would be impossible to be discovered and employed. Ideally, in service oriented architectures, new applications and services are created by novel compositions that derive from cooperation of existing services instead of writing new code to develop new ones. Though, the ability to cooperate with others is an ability that many human beings have it is also researched to a great deal in artificial intelligence in order for agents and robots

to cooperate.

This thesis investigates the very interesting research topic of service composition from the view point of *teamwork*. We focus on developing a teamwork composition model, that exploits the teamwork abilities that involved services have which are constituted by their teamwork behavior.

A service is essentially a well-defined abstraction of a set of computational or physical activities involving a number of resources, with the aim to fulfill user needs or business requirements [Sheng et al., 2014]. With the growing influence of computerized systems on nearly every aspect of today's life and the ubiquitousity of internet, users can make use of services and web based services in different places and situations. In many cases the requirements of a specific user cannot be fulfilled by an existing atomic service. In such situations the user could make use of several atomic services, which when composed fulfil his requirement. In other words, the service composition addresses the situation when a client request cannot be satisfied by any available service, but a composite service, obtained by combining "parts of" available component services, might be used. The composite service can be regarded as a kind of service with respect to its components, since it (indirectly) looks for and invokes them. Web services could be combined into composite services and deliver added value that fulfils different requirements of different users on the given time. Web service composition is one of the cornerstone technologies within service science. One of it's main purpose is in reusing the existing services as building blocks for new services with higher-level functionality. Examples for service composition include our everyday research on the internet for composing services regarding our transfer, our accommodation etc., or the economic arrangements of our financial obligations by using bank services, etc.

Overall, services are not only used by private users, they are also used in business environments. Business are under enormous pressure to comply with changes and the continual upgrades of their information systems due to their engagement with new technologies. Modern business information systems are often built for use over the internet as cloud systems and they are designed with the ability to integrate and collaborate with systems of other partners (e.g. suppliers, customers). Cloud business information systems are commonly provided via the Software as a Service (SaaS) model to benefit and release companies from complex software and hardware management. SaaS targets to the delivery of a software application over the internet to multiple users which makes them accessible on demand, when and where the users require them.

Services are used to implement an entire inter-organizational business process that could be specified by several parties and explicitly or implicitly describes the behavior of each participant. Conventional composite web services have been mainly established

and developed within companies or among companies that have common goals and could exploit the different service models. Nowadays, however, infrastructures for web services that are intended to be accessed by unspecified users have been developed. Web services which wrap programs or contents inside certain organizations have been realized on such infrastructures [Tanaka et al., 2010].

From a business perspective, service composition dramatically reduces the cost and risks of building new business applications in the sense that existing business logics are represented as services and could be reused [Sheng et al., 2014]. Given that business services can also be exposed to the Internet through their business information system as it is shown in figure 1.1, service composition that fulfill business goals can possibly be created on demand, driven by applications, or other services, whenever they require a given functionality that cannot be delivered by a single existing service. This enables new solutions that are provided in the web to be exploited and out of date services to be substituted by novel ones. It also allows each company to focus on a different problem, while reusing third party services for supporting auxiliary activities [Da Silva, 2011].

In practice, various organizations reuse third party service providers that are specialized in such activities. As an example, companies nowadays outsource their customer relationship management (CRM) system, or invoice system, to third party service providers. Another example is Logistic systems that exploit these capabilities and use transport services of other companies to enhance tracking of incoming goods and outgoing final products. This enables them to extend the control and analysis inside and outside the business domain while it improves the efficiency of the company, since they do not need to have a dedicated department or re-create yet another system to handle these business processes [Da Silva, 2011].

To implement the potential, the Service-Oriented Architecture (SOA) has appeared as a significant distributed computing paradigm to support rapid, low-cost development of distributed applications in distributed environments [Patel and Shah, 2016]. Service composition involves two different issues, namely composition synthesis and developing a service choreography. Given a set of available services and a client request, the problem of composition synthesis, or simply composition, is concerned with synthesizing a new composite service, thus producing a specification of how to coordinate the available services to realize the client request. Such a specification can be obtained either automatically, i.e., using a tool that implements a composition algorithm, or manually by a human. As a response and given the specification of the composite service synthesized in the previous phase, the problem of orchestration deals with coordinating the various component services and monitoring the data flow among them, in order to guarantee the correct execution of the composite service.

However, as already mentioned conventional composite web services have been mainly developed in advance, within companies that have common goals. Creating all possible service compositions could be predicted beforehand is difficult, or even impossible, as service developers would need to define all the possible combinations of the available atomic services that users may require. The service composition through a choreography design also involves issues of compatibility and substitution. To develop an understanding imagine the following situation. A service S_x that constitutes part of a composed service S_c becomes unavailable. In order to continue composing the S_c , another service S_y that offers at least the same functionalities of S_x should be found, so that the service S_c , developed from the substitution of S_x with S_y , behaves as intended.

According to web service composition reviews, it is extremely advantageous to give applicable, appropriate web service to users based on their preferences, constraints, context with high precision, having accuracy and composition correctness, in time limits and at the right time. [Patel and Shah, 2016], [Sheng et al., 2014]. Although the area of service composition is continuously being studied, there is still a significant need for research that connects with advancements in technology and how issues related to dependability, personalization, semantic ability, teamwork ability and intelligence among others, need to be addressed. The challenge is to develop composite services that could easily integrate cloud and web services and resources, would have autonomy and could adapt to domain changes, support intelligent capabilities and become robust when emergencies occur. Existing models should be revisited and examined whether they take into account these new concerns and in what degree they encompass the aforementioned characteristics. Only services that are built having these characteristics could exploit capabilities of the new technology paradigm such as IoT, Cyber physical systems [Chen et al., 2018], e.t.c..

1.2 Challenges and Aim

A challenge here is the ability of a domain to accommodate new augmented services by automatically composing and executing them as novel software services that are able to react to either predetermined or other abrupt and unexpected behaviors in order to adapt theirs.

As already mentioned, the problem of web service composition is a highly complex task and the composition process, in practice nowadays, face significant challenges due to the environment's openness, dynamism, the new technologies has emerged and the change of the way people interact with the environment in various organizations. Therefore, selecting the appropriate and optimal atomic services from a service pool, focus on the dynamic characteristics of the problem and address changes in the properties of the services or

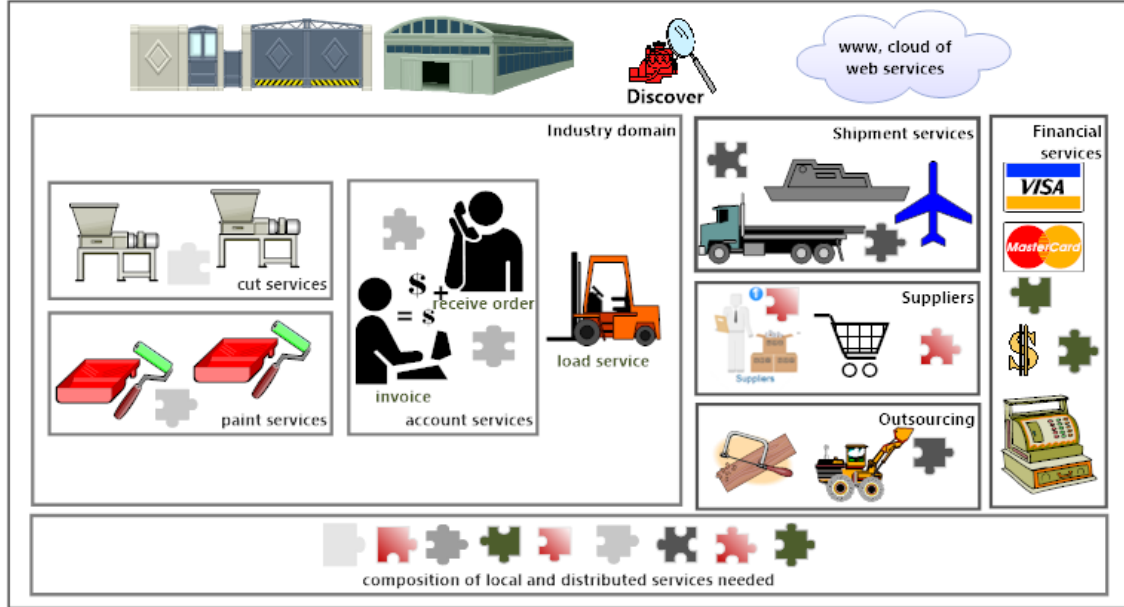


Figure 1.1: Distributed services appearing in industry domains, cloud and www

the environment during execution are only some among and network appear the most important issues that must be investigated and addressed. Reviews on the field such as [Patel and Shah, 2016], provide detailed insights of unresolved issues and challenges investigated about services, specifically in discovering and composing web services. We first identify a series of challenges and requirements that need to be met for the service composition, and in the following chapter a state-of-the-art for the related technologies of service composition is provided.

In the following, we underline some sources of service composition complexity and primary challenges identified in the composition problem.

1. Plethora of web services deployed on the web.

The web has become an inexhaustible provider of new services. Given the myriad of available web services, the service composition problem need to take into account their heterogeneity and their continual updating. In order to treat this chaotic and complex content a solution could be found in terms of clustering.

2. Manual vs Automatic composition.

Automatic service composition may dramatically improve development efficiency of Web Service applications. When service composition is performed manually, it is a very complex, time-consuming and error-prone task. The complexity comes from the fact that the developer in order to build a composition, have to make an efficient service discovery and to know details about composition requirements and other technical details of each service to be composed. With the plethora of web services evolved in a composition it is already beyond the human's capability

to analyze all of this manually. Such complexity becomes a real problem when is done in dynamic application domains where novel more competitive services appear while other outdated services are withdrawn. Consequently, it is critical to build composite web services with an automated way for the success of the web service applications.

3. Dynamic and on the fly composition.

Service composition approaches are categorized in those produce a static and in those produce a dynamic composition schema. A static composition approach involves selecting the component services to be used and aggregate them to form a composite process at the design time, which is then deployed. Conversely, the dynamic composition approaches first produce an abstract composition schema, without actual service bindings, and afterwards service components are determined and replaced during runtime where they are bound to actual service endpoints.

Static composition approaches enable tightly coupled interaction among services, and assume a quite static environment, since no or very little support is provided for example in establishing dynamic relationships among services, or to cope with changes of the involved services. In such cases a static composition schema becomes invalid, inconsistent and impossible to execute. A composition approach should absorb changes in the behaviors of Web services. Service providers sometimes update the program or the contents of their Web services. In order to use the service in the same way as previous, we have to absorb the changes in the behavior of the Web service.

Dynamic approaches provide a modular and extensible approach to composition. It is able to overcome issues such missing services during the execution and services being replaced by new ones by providers, since (if the global schema is well designed) whenever a component service is modified, only the mapping to the plan is affected, therefore it is suitable for highly dynamic environments. The Dynamic Web services composition requires the execution system to support automatic discovery, selection, and binding of service components. Dynamicity ensures that a produced composition schema is consistent and executable long after its initial design [Baryannis, 2014].

4. Effective Service interaction.

The interaction between web services is mainly done with two standardized ways: Service Orchestration and Service Choreography.

Service Orchestration, is concerned with how to achieve the coordination among services, by executing the specification produced by the composition synthesis and

by suitably supervising and monitoring both the control flow and the data flow among the involved services [Lohmann, 2010]. In service orchestration the coordinated and arranged atomic services are available as a single aggregate service. It takes the viewpoint of a single participant in the service composition and abstracts from the internal behavior of other participants. The orchestrator only considers the interfaces of the other participants rather than their concrete behavior or the detailed interaction between third parties. Research on orchestration draws from several research areas, including workflow. The successful orchestration of component services into a valid workflow specification depends on the domain knowledge the orchestrator has at his disposal. Developers utilize service orchestration to support the automation of business processes by loosely coupling services across different applications and enterprises. In other words, service orchestration is exploited as the combination of service interactions to create higher-level business services.

Services also interact to produce a service composition using the specification called chore-ography. This is also employed to create composite services and applications in service oriented architectures but with different approach. A choreography globally specifies the observable behavior of a composition. While a service orchestration model relies on a central controller to bind services a service choreography model works without a central messaging engine or orchestrator. By using the choreography, each participating service know the business logic and sequence of message exchanges. In orchestration, the participating services do not know that they are being orchestrated as part of a higher-level service; only the central controller knows the business logic and messaging sequence.

Although up to our days services are usually bound under predefined plans, for example through an orchestration or a choreography language to finish a specific goal, nowadays we need to create composite services that go beyond these limitations. Traditional composition methods that are based to centralized and hierarchical control approaches are good at following the best plan at the design time but weak at responding to change, mainly because the execution process is centralized. On the other hand, dynamic composition approaches (like choreography) respond well to change and unpredictable disturbances but, during the decision making process there is only partially knowledge of the system and decisions could not be as good as those take into account the global needs.

5. Lack of common definition.

The web services are heterogeneous and have different formats for exchanging data over the web. Services are commonly designed to be used with specific other services

in a composite service. They are usually developed by different developers that use different conceptual models for presenting services' characteristics that commonly gear towards specific domains. The gap between service oriented description and service specification according to specific characteristics of service has affected the service discovery and composition activities. This requires employment of relevant characteristics for matching and composition of web services. In addition, domain specific concepts lead to the non-uniformity in publishing data to the registries. However, to make all the services homogeneous during their construction is not feasible [Patel and Shah, 2016]. On the other hand, clustering of services as per requirement after their development could provide a solution [Bano and Ikram, 2010]. Web services could be grouped together by assigned to a desired category with the help of clustering process according to their respective domain and area. This process obviously will reduce the searching and discovery time and increases the domain knowledge of the stakeholder [Patel and Shah, 2016].

6. Monitoring the Composition Correctness.

Not unlikely, most of web services offer no guarantee of availability to the users. Unlike in a centralized monolithic system, parts of the composed systems may change and are not completely controlled by a single party. Consequently, an important aspect of web service composition is to maintain the correctness of behavior of the composite service as this was during the design time. In order to preserve compositional correctness various formal methods techniques could be integrated within the composition frameworks. The evaluation and verification of the compositional correctness is missing in most of the approaches of service composition [Khadka and Sapkota, 2010], [Patel and Shah, 2016]. In order to verify the correctness, compositional specifications have to be modelled into mathematical formalisms such that the composition process can include deadlock detection and consistency feature. Another solutions is to include compositional correctness features in the composition algorithm or monitor the composition process in order to enable service flexibility and correctness which is essential in a dynamic and unpredictable environment for ensuring that services tasks are consistent with the goal of the designed composite service.

7. Bringing to composite services Collective Intelligence.

Collective intelligence (CI) is the capacity of human collectives to engage intellectual cooperation in order to achieve a common goal [Iguider and Morita, 2011]. CI is a fundamentally different way of viewing how applications can support decision making and it could also be used in the field of Web services [Tanaka et al., 2010].

Service composition creates new value by combining services provided by various organizations via services computing technologies. In Service-oriented collective intelligence a composite service performs an intelligent comprehensive process of service selection, adaptation and coordination satisfying new constraints and policies in the domain is going to be executed.

8. QoS Awareness

The idea behind the QoS is to offer mechanisms that can facilitate a domain with services that operate at reasonable or acceptable level of performance. It also provides the description or measurement of the overall service performance. QoS aware approaches take into account not only functional characteristics of services but also non-functional ones, dealing with quality aspects such as response time, price, availability and so on. QoS has cost in order to a specified service has to operate at reasonable level of performance. Considering QoS aspects when deciding which services to include in a service composition schema is important when functional requirements are satisfied by more than one service. As a result, composite services produced by QoS-aware approaches not only offer the capabilities requested by the user but also guarantee the best possible quality. The composition problem of the running example requires QoS-awareness, since it contains non-functional goals both for parts of the process as well as the overall composition.

9. Adaptivity

Another challenge for a composite service that is examined involves the ability of a service to adapt itself. The adaptivity in general, refers to something that is associated with many factors of its environment, usually show the suitable, demand-satisfied and goal-achieved features. The trend to make software systems adaptive and more aware of the context have started to appear in application frameworks. We expect the ideas and mechanisms of context aware systems to become more available and adopted in the future, when most end-user applications will be developed using dedicated context-oriented technology so that software systems can seamlessly adapt to their users and context of use. In services also, adaptation is the process of modifying the product service in order to satisfy new requirements and to fit new situations dictated by the environment on the basis of predefined adaptation strategies. Adaptation goes one step further to dynamic composition approaches, in the sense that they not only seamlessly adapt to their users and context of use but also deal with changes in the requirements set by the requester.

Adaptation in services could be both proactive and reactive. In the former case the service aims to modify a behavior before a deviation occurs that could lead to

problems, while in the last case reactive adaptation handles faults and recovery from problems reported during execution.

1.3 Thesis approach

Service oriented computing allows to construct large distributed systems by composing several heterogeneous and decentralized services. At the same time, as this was discussed above, new challenges arise with the distributed execution of independent services in flexible compositions. In particular, the correctness of a service composition depends on the local correctness of each participating service and the correct interaction between them.

A main motivation for this research was the integration and adaptation of common teamwork approaches in relevant disciplines to enhance *cooperation of services*; according to author's point of view this will give momentum to the development and execution of robust, flexible and improved composite services.

Teamwork has become an important research field and its contribution to performance has tracked attention of various research groups from several disciplines e.g. from business management to cognitive science, human discourse, and distributed artificial intelligence. In recent years, many researchers studied why humans succeed or fail in joint activities and a variety of models have been developed that follow social - psychological approaches for team formation. Although services have been researched for many years and several scientific communities focus on service-related topics, to the best of our knowledge, there do not exist widely accepted teamwork oriented model for the composition which is specific to services. From a practical point of view, this brings us to the central research question which touches on several of the aforementioned challenges and it is investigated in this thesis:

How to design correct services and service compositions that can systematically be supported from inside?

In this thesis, we focus our research on web service composition to cover most of the challenges are described above. To respond to the central question, we propose a novel approach for the composition problem, which is a teamwork Driven Service Composition, by using the role modeling technique and support service's interoperability through novel service teamwork roles. According to this approach, services can be organized in domains that are modeled within role models which serve specific goals. The idea is that such a domain that has Web services as habitants could be designed by using a role model in a

sense that it will not treat the interactions between web services but between the roles they play. We use the role modeling approach and augment the process of service composition with eight teamwork roles that will support services' cooperation. Figure 1.2 depicts the proposed approach, and how it is able to collect, integrate and compose services across all functional units of the business environment and the web environment based on the role they play and the behavior they exhibit. Additionally, the teamwork roles that are embraced to construct teams result in implementing composite services of high quality since they enable and shorten the time for searching, improve communication, support in decision making, and achieve higher productivity.

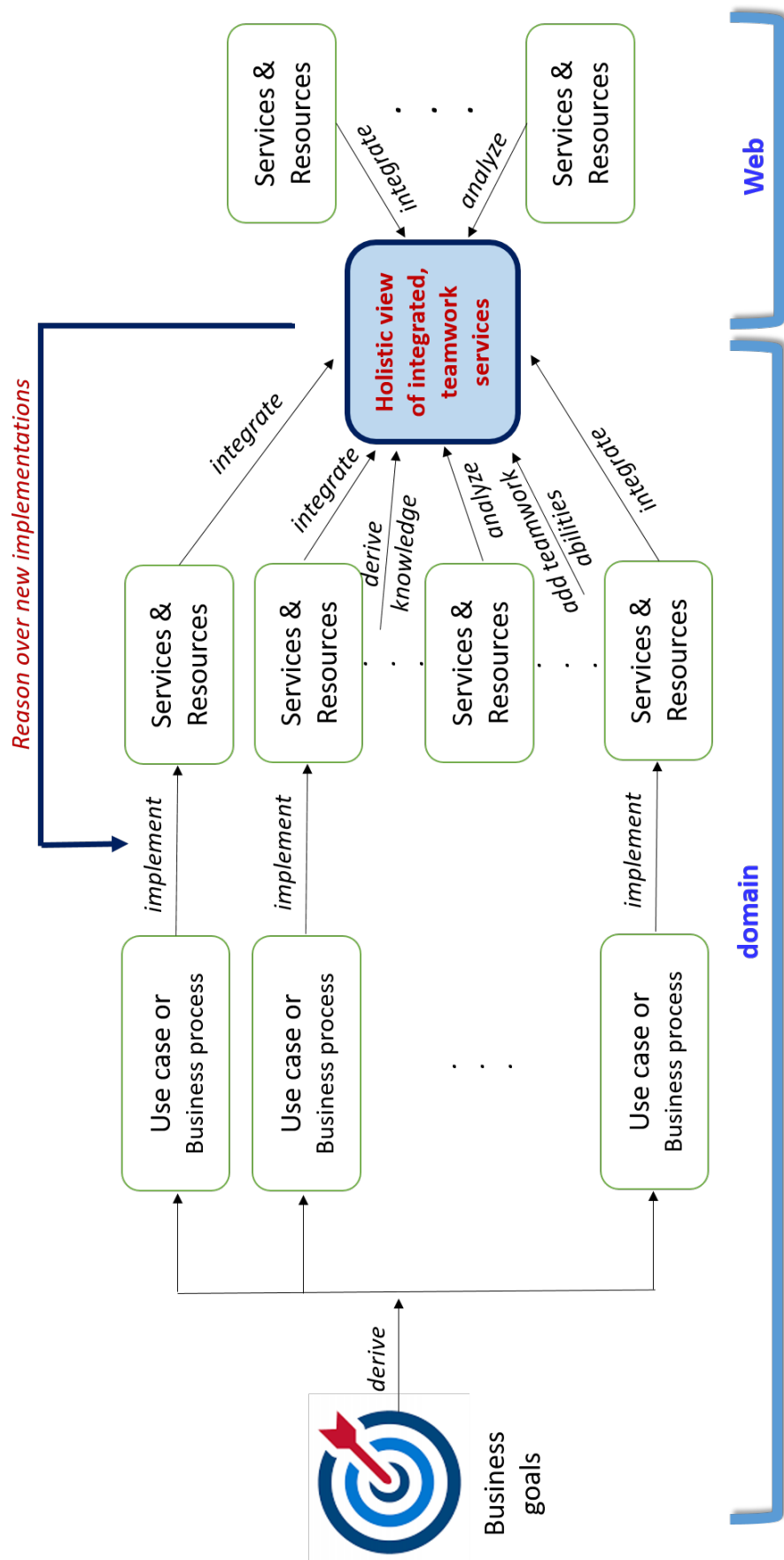


Figure 1.2: Novel service implementations consisted of local and web services

1.4 Research Questions

In order to be able to answer the central research question, we formulate and further investigate a set of basic research sub-questions that address the problem of service composition in detail and guided the study to support this thesis.

RQ1: How to model services in order to better organise them in a relevant domain and manage the existing complexity of services offered in the web?

- i How to achieve automatic synthesis of local and distributed services in a system that takes into account diverse composition concerns simultaneously including constraints that regard to the domain holistically?
- ii How participants of a domain will not be aware that the system consists of several distributed components which implement the composite system and that have been bound without revealing specific implementation details?
- iii How to cope and construct a plan for the composite service with the large number of services that are available on the web?
- iv How to model the heterogeneity of services that exist in a domain (business, IoT, IT, web services)?

RQ2: Is it possible for a number of services that have been aggregated in order to develop new services to construct new composite services that are going to act as teams to achieve their goal?

- i How to convert a group of services to a team in order for participants to cooperate more efficient?
- ii Can we exploit the teamwork theory and approaches applied to group of humans and agents?
- iii Is it possible for a composite service to be augmented with teamwork abilities?

RQ3: What teamwork roles should be included in a team of services? How the teamwork “enhanced” services interact and how these roles will systematically monitor and support the process of service composition from inside the team?

- i Which are the teamwork roles should be included for the successful service composition and execution?

- ii What is the scope of tasks each one will have? Another central issue in a composition approach is the error detection during the composition process and correction. In case an error is detected, how participating services are responsible for this error and how they solve it?

RQ4: How to guarantee the correctness of the system in use?

- i What formal methods could be used to guarantee the correctness of the proposed system?

RQ5: How this will be applied in a domain to verify it and test its correctness?

- i Which is an appropriate domain to be applied the proposed model?
- ii What is the architecture that it will be included?
- iii How the designed isolated systems will be integrated in a framework for the specific domain?

The aim of this thesis is to consider the web service composition problem from the viewpoint of teamwork and to propose a teamwork based framework for supporting the synthesis.

The concept of developing autonomously composite services capable of exhibiting teamwork behavior is introduced as a challenge in this thesis. A basic weakness detected in most of research efforts on service composition, is that although web services work together as a group to achieve a common objective they do not use any teamwork abilities as it happens with groups of human, agents or robots. A *composite service capable of teamwork behavior* is one that can effectively cooperate with multiple potential teammates on a set of collaborative tasks, promote the team and is able to catch errors and emergent behaviors during its operation. The corner stone of the proposed solution is the explicit teamwork model that abstracts composition requirements and constraints away from the details of service implementations. By linking services and by expressing composition requirements and domain constraints in terms of the teamwork role model, we create a formal setting in which abstract requirements and constraints, could be grounded to available service implementations.

The teamwork model proposed provides the scalability and flexibility to the composition system while ensuring the correctness and the completeness of the solution that will provide robust composite services. The approach is distinguished from related work since the focus is on the teamwork attitude of the participants. Incorrect service compositions that are automatic, could expose undefined and unpredictable behavior, and consequently affect not only technically the process of service composition but could also be considered

to be a threat to public in safety-critical systems or prejudice the commercial interests of individuals, companies, or governments.

1.5 Thesis Contribution

This thesis describes an in-depth study into the area of service composition in dynamic environments, and proposes a model that extends the existing composition models as contributions to the service science area. The contributions are all relating to the work performed towards services, their composition process and factors that affect their performance. A list of publications has resulted that have been published in earlier papers, as part of the research for this thesis. This research is depicted in the following publications:

Publications Related to this thesis

1. Tsoutsas, P., Fitsilis, P. and Ragos, O. (2017). Role modeling of IOT services in industry domains. In Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences, pages 290–295. New York, NY, USA: ACM. [Tsoutsas et al., 2017]
2. Tsoutsas, P., Fitsilis, P. and Ragos, O. (2018). Services simulation in Industry 4.0: A comparison of simulators, In 3rd International Scientific Conference on Industry 4.0.Varna, Bulgaria. [Tsoutsas et al., 2018b]
3. Fitsilis, P., Tsoutsas, P. and Ragos, O. (2018). Teamwork behavior in smart and sustainable cities ecosystems. In The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications, Venice, Italy. Springer [Fitsilis et al., 2018]
4. Tsoutsas, P., Fitsilis, P. and Ragos, O. (2019). Teamwork behavior: A review to interconnect Industry 4.0 entities. In Technological Developments in Industry 4.0 for Business Applications, pages 1–25. IGI Global. [Tsoutsas et al., 2019]
5. Tsoutsas, P., Fitsilis, P. and Ragos, O. (2019). Enhancing teamwork of services. In Business Information Systems, Berlin, pages 290–295. Springer. [Tsoutsas et al., 2018a]
6. Tsoutsas, P. and Ragos, O. (2020). Towards an Ontology for Teamwork Enabled Services. In International Conference on Algorithms, Computing and Systems, ICACS, pages 181–185. ACM (ISBN: 978-1-4503-7732-4) [Tsoutsas and Ragos, 2020]

7. Tsoutsas, P., Fitsilis, P., Anthopoulos, L., and Ragos, O. (2020). Nexus services in smart city ecosystems. *Journal of the Knowledge Economy*, pages 1–21 [Tsoutsas et al., 2020]
8. Tsoutsas, P., Iatrellis, O., Fitsilis, P. and Ragos, O. (2020). A Framework for developing Teamwork Enabled Services in Smart City Domains. Accepted for publication to 4th International Conference on Computers in Management and Business (ICCMB2021) [Tsoutsas et al., 2021]

This thesis summarizes and extends these results.

1.6 Thesis Structure

The thesis is structured along the way the study and research was carried out. In more details, the subsequent chapters are organized as follows.

Chapter 2 lays the groundwork in terms of terminology and core *background* concepts used throughout the thesis, necessary to understand the solution we introduce to the problem of service composition. It also presents the state of the art in the related areas used, that of role modeling and teamwork.

Chapter 3 describes the *Role Modeling* approach and how this is utilized for the description and composition of services. It also discusses service heterogeneity, which should be considered when designing role modeling domains for service composition.

Chapter 4 elaborates on the nature of teamwork and surveys contemporary teamwork models in relevant disciplines. It gives a detailed description of characteristics in agent teamworking and the latter ones are arranged in four basic categories that could well be used to map a wide range of services teamwork characteristics. It also presents the conceptual framework containing the *service teamwork roles* and discusses their analytic behavior which support the composition process.

Chapter 5 is dedicated to the conceptual formal modeling of service teamworking. We make the conceptual based modeling for *teamwork oriented composition of services* by developing the appropriate ontologies to define teams, roles, services and their modeling domains abstracting the most important concepts of team development whilst providing a decent level of formality and unambiguity.

Chapter 6 of the thesis details the case study developed to test the correctness, effectiveness and performance of the proposed model. The case study where the model is utilized is a *smart city*, which is an appropriate domain to show how the approach could be

implemented to improve the performance of service execution to improve the smart environment.

Finally, Chapter 7 concludes the thesis and summarizes the contributions and the remaining open problems. Furthermore, it sketches directions for future extensions of the presented results and state our forecast about the future work view.

References

- [Bano and Ikram, 2010] Bano, M. and Ikram, N. (2010). Issues and challenges of requirement engineering in service oriented software development. In *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*, pages 64–69. IEEE.
- [Baryannis, 2014] Baryannis, G. (2014). *A novel specification and composition language for services*. PhD thesis, University of Crete.
- [Chen et al., 2018] Chen, N., Cardozo, N., and Clarke, S. (2018). Goal-driven service composition in mobile and pervasive computing. *IEEE Transactions on Services Computing*, 11(1):49–62.
- [Da Silva, 2011] Da Silva, E. M. G. (2011). *User-centric service composition: towards personalised service composition and delivery*. University of Twente, Enschede, Netherlands.
- [Fitsilis et al., 2018] Fitsilis, P., Tsoutsas, P., Anthopoulos, L., and Ragos, O. (2018). Teamwork behavior in smart and sustainable cities ecosystems. In *The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications , Venice, Italy*. Springer??
- [Iguder and Morita, 2011] Iguder, Y. and Morita, H. (2011). A collective intelligence based approach to business-to-business e-marketplaces. *International Journal of Computer Science Issues (IJCSI)*, 8(5):12.
- [Khadka and Sapkota, 2010] Khadka, R. and Sapkota, B. (2010). An evaluation of dynamic web service composition approaches. In *ACT4SOC*, pages 67–79.
- [Lohmann, 2010] Lohmann, N. (2010). *Correctness of services and their composition*. PhD thesis, Eindhoven University of Technology.
- [Patel and Shah, 2016] Patel, S. and Shah, T. (2016). A survey on issues and challenges of web service development, composition, discovery. *Journal of Science and Technology (ISSN: 0975-5446), VNSGU*, 5(1).
- [Sheng et al., 2014] Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. (2014). Web services composition: A decade’s overview. *Information Sciences*, 280:218–238.

- [Tanaka et al., 2010] Tanaka, M., Murakami, Y., Lin, D., and Ishida, T. (2010). Service supervision for service-oriented collective intelligence. In *Services Computing (SCC), 2010 IEEE International Conference on*, pages 154–161. IEEE.
- [Tsoutsas et al., 2020] Tsoutsas, P., Fitsilis, P., Anthopoulos, L., and Ragos, O. (2020). Nexus services in smart city ecosystems. *Journal of the Knowledge Economy*, pages 1–21.
- [Tsoutsas et al., 2017] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2017). Role modeling of iot services in industry domains. In *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, pages 290–295. New York, NY, USA: ACM.
- [Tsoutsas et al., 2018a] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2018a). Enhancing teamwork of services. In *Business Information Systems, Berlin*, pages 290–295. Springer.
- [Tsoutsas et al., 2018b] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2018b). Services simulation in industry 4.0: A comparison of simulators. In *In 3rd International Scientific Conference on Industry 4.0*. In 3rd International Scientific Conference on Industry 4.0.
- [Tsoutsas et al., 2019] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2019). Teamwork behavior: A review to interconnect industry 4.0 entities. In *Technological Developments in Industry 4.0 for Business Applications*, pages 1–25. IGI Global.
- [Tsoutsas et al., 2021] Tsoutsas, P., Iatrellis, O., Ragos, O., and Fitsilis, P. (2021). A framework for developing teamwork enabled services in smart city domains. In *Accepted for publication in 4th International Conference on Computers in Management and Business (ICCMB2021)*. ACM.
- [Tsoutsas and Ragos, 2020] Tsoutsas, P. and Ragos, O. (2020). Towards an ontology for teamwork enabled services. In *Proceedings of the 2020 4th International Conference on Algorithms, Computing and Systems*, pages 69–75. ACM (ISBN: 978-1-4503-7732-4).

Όσο πιο πρωτότυπη είναι μια ανακάλυψη,
τόσο πιο προφανής φαίνεται μετά.

Άρθουρ Κέσλερ, 1905-1983,
Ούγγρος συγγραφέας.

The more original a discovery
the more obvious it seems afterwards.

Arthur Koestler, 1905-1983,
Hungarian novelist.

Chapter 2

Background and Related Work

Contents

2.1	Web Services	20
2.1.1	Service Definition	20
2.1.2	Service Description and Modeling	21
2.1.3	The Many Approaches for Service Composition	24
2.2	Role Modeling	29
2.3	Teamwork	36
2.3.1	Different Approaches of Teamwork Modeling	36
2.3.2	Human Teamwork	37
2.3.3	Human Based Teamwork Models	38
2.3.4	Integration of the Roles of the Major Human Teamwork Models	43
2.3.5	Teamwork Theory to Agent Based Models	46
2.4	Discussion	51

In this chapter we lay the groundwork in terms of employed terminology and core background notions used throughout the thesis. In particular, this chapter gives an insight to concepts related to services, service composition, role modeling and teamwork. Further, we present some of the main works published in the literature which constitute the state of the art on the related topics and are relevant to the direction this thesis follows. In the first sections, we concentrate on service definition and service composition. In the sequence, this chapter explores surrounding issues that are related to with role modeling

and teamwork. These issues are somehow connected to services, and they are included in solutions or approaches proposed in this field or could be applied to help solving open issues regarding services.

2.1 Web Services

Services are components that can be integrated in a loosely coupled manner into more complex distributed applications. The term web service describes a standardized way of integrating web based applications using standards such as the Extensible Markup Language (XML)¹, Simple Object Access Protocol (SOAP)², Web Services Description Language (WSDL)³, Java-Script Object Notation (JSON)⁴ and Universal Description, Discovery, and Integration (UDDI)⁵ open standards. A *web service* is a software artifact that is delivered over the Internet and interacts with its clients, that could be either human users or other services, in order to perform a specified task.

Semantic Web services (SWS) are the result of the evolution of the Web, since they add semantic annotation to the syntactic description of Web services and are the fruitful combination of Semantic Web technology and Web services. They are a mixture of XML Web services and semantic Web technologies (Ontology, semantic annotation). By using ontologies as the semantic data model for Web Service technologies Web Services we have machine-processable annotations instead of static data on the Web. Some important challenges of the application of SWS include the enhancement of them with semantic interoperability, composition and performance evaluation. Semantically enhanced information processing empowered by logical inference will eventually allow the development of high quality techniques for automated discovery, composition, and execution of Services on the Web, stepping towards seamless integration of applications and data on the Web [Feier et al., 2005]. In the following sections, we give detailed and formal definitions for services and their composition.

2.1.1 Service Definition

Web services are distributed and independent pieces of code, they solve specific tasks and communicate with each other through the exchange of messages. They are built on XML-based open standards, promise the interoperability of a variety of applications running on heterogeneous platforms, and enable dynamic connections and the automation

¹<https://www.w3.org/XML/>

²<https://www.w3.org/TR/soap/>

³<https://www.w3.org/TR/wsdl20/>

⁴https://www.w3schools.com/js/js_json_intro.asp

⁵<http://uddi.xml.org/>

of processes. A *Web service* essentially is a semantically well-defined abstraction of a set of computational or physical activities involving a number of resources, intended to fulfill a client need or a business requirement [Sheng et al., 2014]. A step further in making less abstract the definition of web service is the definition provided by the World Wide Web consortium (W3C) [W3C, 2002]. The W3C defines a Web service as:

“a software system identified by a Universal Resource Identifier (URI), whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols”.

The ability to integrate the Web services developed by different organizations together to fulfill the user’s requirement was intensely exploited. Such integration is based on the common standards of Web service interfaces, regardless of the languages that are used to implement them and the platforms where they are executed [Rao et al., 2004]. Some well-known problems related to Web services are how to specify them in a formal and expressive enough language, how to discover them through the web, how to compose them and how to ensure their correct execution especially in dynamic environments.

A variety of proposals that describe, compose and verify web services have emerged with some of them utilising formal methods. Formal methods are particularly well suited to address most of the aforementioned issues such as composition and correctness. A formal method is a mathematical based technique for the specification, development and verification of particular systems. This field of research combines many areas of computer science and comes with an impressive body of literature on specification languages and verification tools. The majority of these are based on state action models (e.g. labelled transition systems, automata, Petri nets, pi-calculus, e.t.c.).

2.1.2 Service Description and Modeling

There are several ways in which services are described in the related literature. This is especially clear when considering the enormous quantity of languages that are being proposed as standards by the industry [Berardi et al., 2005a]. For instance, there exist various specification and programming languages and only some of these are SOAP, WSDL, Semantic Annotations for WSDL (SAWSDL)⁶, Unified Service Description Language

⁶<https://www.w3.org/TR/sawSDL/>

(USDL)⁷, Web Ontology Language for Services (OWL-S)⁸, Web Service Modeling Ontology (WSMO)⁹. In order to give an insight into the current approaches to service description we present two major ones and highlight the open issues, with respect to their degree of completeness in relation to service composition and service execution.

WSDL [Chinnici et al., 2007] is a W3C recommendation that has been established as the de facto description language for web services according to the standard XML for its functionality at the syntactical level. A WSDL file describes the functionality and the localization of a Web service at the syntactical level. These are described in the four fundamental elements of WSDL which are (i) the *type*, that describes the kinds of messages that the service sends and receives, (ii) the *interface*, which offers an abstract description of the provided functionality, (iii) the *binding*, that describes how to access the service, (iv) the *service*, which describes where to access the service. But WSDL was mainly created as a language for the syntactic description of Web services and according to WSDL description, a service is simply a construct that produces a specific set of outputs, given a specific set of inputs. Without doubt, this does not allow service consumers to understand what a service does and under which conditions, making it difficult to decide whether a particular service is suitable for their purposes or not [Baryannis, 2014]. Extensions to WSDL are proposed in [Chen et al., 2006], which are WSDL-S, SAWSDL in order to support a way to manage the Quality of Services (QoS) or to support semantics, since a service specification only in terms of operations and bindings is not enough to specify its capabilities, its constraints and other situations like availability, reliability, e.t.c.

OWL-S [Martin et al., 2007], previously DARPA Agent Markup Language for Web Services (DAML-S), provides the ability for describing and reasoning over services semantically. It is a semantic markup for web services and establishes a framework within which web service descriptions are created and shared by employing an ontology.

OWL-S, as it is described in Figure 2.1, consists of three upper ontologies, namely:

- the *Service Profile* is used to describe what the service does; it allows service providers to advertise the services they offer for the purposes of discovery. Each service profile may be thought of as a summary of aspects of the process model plus additional information that is suitable for the purposes of advertising and selection. Although service providers are free to include whatever information they think that are necessary, the OWL-S standard offers a predefined subclass that should contain information on the service provider, the service functionality and a set of service characteristics. The functional description of a web service in OWL-S contains the

⁷<https://www.w3.org/2005/Incubator/usdl/>

⁸<https://www.w3.org/Submission/OWL-S/>

⁹<https://www.w3.org/Submission/WSMO/>

full set of functional properties (e.g. inputs, outputs, and preconditions) and non-functional properties (e.g. QoS parameters) in contrast to WSDL. Additionally, the service profile class can be specialized to create profile taxonomies that subsequently describe different classes of services.

- The *Service Model* sub-ontology in OWL-S is used to describe how the service is used; it specifies the manner in which a client may interact with the service in order to consume its functionality. OWL-S provides the basis for specifying a Web service as a business process or a workflow. A process specifies, among other things, the possible patterns of interaction with a web service. Two sorts of processes can be invoked: atomic and composite processes. An atomic process is one that has no internal structure. It corresponds to a single interchange of inputs and outputs between a consumer and a provider. A composite process is specified using control constructs such as Sequence, Split-Join or If-Then-Else, representing sequential, parallel and conditional composition schemas respectively.
- the *Service Grounding* sub-ontology describes how to access the service; it is used to specify how the abstract information exchanges that are detailed by atomic-process descriptions are realized, by concrete information exchanges between the consumer requesting a service and the service provider. It enables service designers to map an OWL-S description to existing service description languages, enabling service calls to be translated into and out of message transport protocol languages such as SOAP and WSDL. The purpose of this mapping is to allow existing WSDL descriptions to be reused and augmented by using OWL-S while at the same time it allows a service to have multiple alternative groundings.

OWL-S was the first service description framework to describe services with inputs, outputs, preconditions, effects which offer semantic capabilities and support composite service description. Inputs describe the objects that the service works on while outputs describe the objects that it produces. These descriptions are not known when the process is defined, just the specification of their types it is given (as OWL classes, at the most abstract level). A precondition is a proposition that must be true in order for the service to operate effectively. Results consist of effect and output specifications. An effect is a proposition that will become true when the service completes its operation. However, although precondition and effects for services are observed, there are not included mechanisms that locate not expected effects and react when these situations occur or other abnormal issues have appeared.

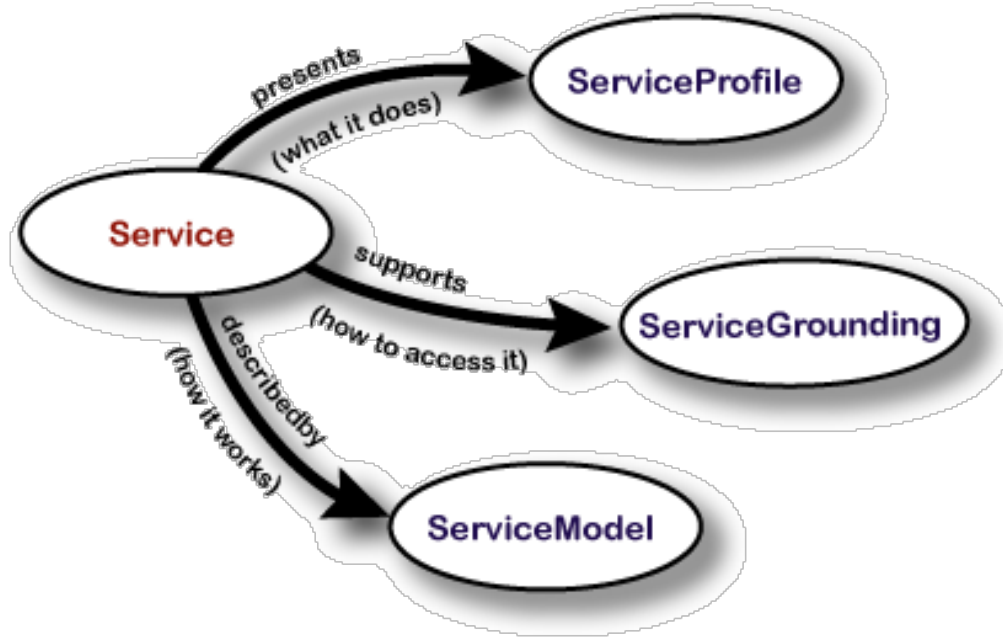


Figure 2.1: Top level of the service ontology [Martin et al., 2007].

2.1.3 The Many Approaches for Service Composition

Current service composition approaches range from practical languages aspiring to become industrial standards (e.g. BPEL and OWL-S) to more theoretical models and languages (e.g. automata, Petri nets, and process algebras).

We continue the literature review by presenting some of the most representative efforts in service composition. The problem of *service composition* is the problem of constructing a new composite service, by using available services that perform some desired functionality, when no single service satisfies it. According to [Fensel and Bussler, 2002], the composition is a process which functions, *in an intelligent way*, in order to discover services automatically, to negotiate between them and to compose them in more complex services, while according to [Nacer and Aissani, 2014] web service composition is a technique which assembles web services in order to achieve a particular goal, via primitives of control and exchange. During our research on service composition's major techniques, prototype and standards, our major goal was to identify open research challenges in this area. Apparently, this problem belongs in an evolving area, and very often appear new proposed methods for its solution. For this reason, the reviews we present could not be exhaustive, we only present some representative studies. Reviews on web service composition and their classification survey, e.g. manual vs. automated and static vs. dynamic have been published in [Bano and Ikram, 2010, Khadka and Sapkota, 2010, Patel and Shah, 2016, Sheng et al., 2014] among others. Two main approaches are investigated for static service composition. The first approach, referred to as web service orchestration,

combines available services by adding a central coordinator, i.e. the orchestrator, that is responsible for combining the single sub-activities. The second approach, referred to as web service choreography, defines complex tasks via the definition of the conversation that should be undertaken by each participant in the composition. Following this approach, the overall activity is achieved as the composition of peer-to-peer interactions among the collaborating services. To dynamically compose services is a more challenging problem. In particular, when a functionality that cannot be realized by the existing services is required, the existing services can be combined to fulfill the request. The dynamic composition of services requires the location of services based on their capabilities and the recognition of those services that can be matched to create a composition. The full automation of this process is still the subject of ongoing research. One of the problems for full automation of service composition is the gap between the concepts that people use and the data that computers interpret. This barrier can be overcome up to a point by using semantic web technologies.

The review of [Sheng et al., 2014] gives an analysis of existing major techniques, research prototypes, and standards on web services composition while the work of [Patel and Shah, 2016] indicates more open and unresolved research challenges in the area.

The need to compose services from disparate and heterogeneous environments initiated a number of research efforts both in industry and academia. On the whole, approaches to the automated service composition problem have been exceptionally diverse, as it is witnessed in classification from [Nacer and Aissani, 2014] presented in figure 2.2 and offer different interpretations of what should be addressed during service composition [Baryannis, 2014]. The classification is done by using categories that are commonly used in composition reviews and represent many known approaches of Web service composition.

Regarding the industry approaches, some standards that determine the composition as executable workflows are Business Process Execution Language for Web Services (BPEL4WS) and Business Process Modeling Language (BPML). BPEL provides a language for the formal specification of business processes and business interaction protocols. BPEL4WS like DAML-S ServiceModel is focused on representing service compositions where a process flow and bindings between services are known a priori. BPML is based on π calculus and like XLANG supports semantics such as nested processes and complex transactions that are not addressed by BPEL4WS.

Academic research activities have also been resolved by various models and a criterion for clustering that is used in general is the one that focus on issues related to Functional properties and non-Functional properties of the composition. The first category can be subdivided more to approaches considering only the syntactic description and to approaches considering semantic description. The later, usually deal also with a specific model of

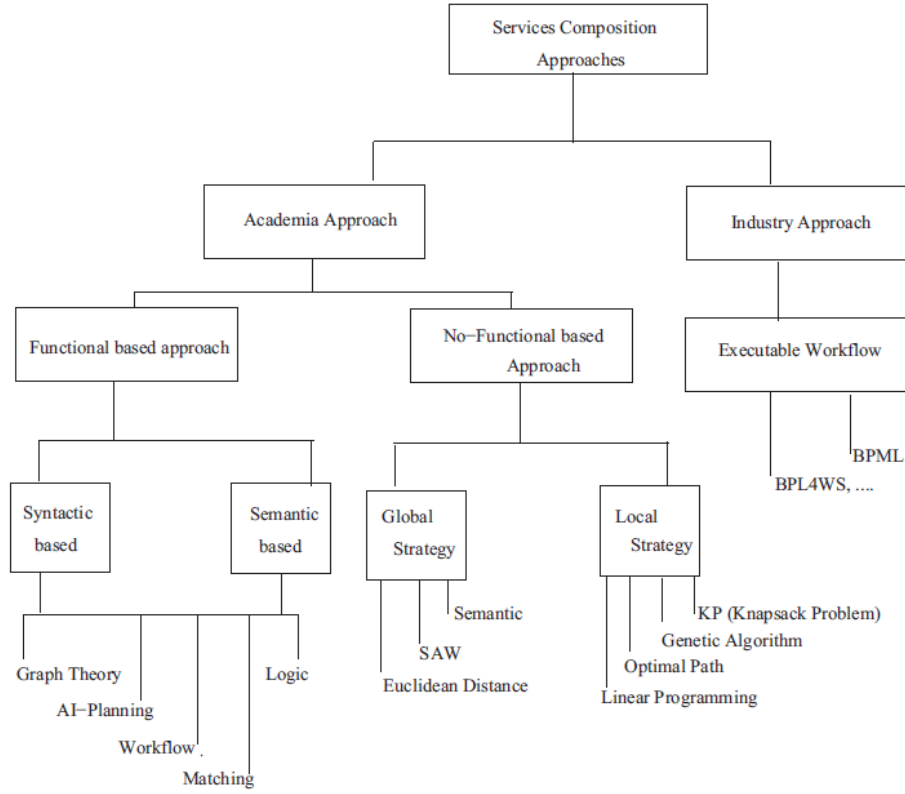


Figure 2.2: Classification of web services composition approaches [Nacer and Aissani, 2014].

interaction such as Graph Theory, Artificial Intelligence Planning Approaches, Workflow, Matching, and Logic-based Approaches. Some of AI approaches to planning that have been investigated for the purpose of Web service composition are Finite State machines (FSM), Situation calculus, Hierarchical Task Network (HTN), Petri Nets. Composition proposals that use semantic description use the principles of the Semantic Web to provide more concrete service composition by considering the semantic properties of services. Some of the semantic approaches are Semantic annotation, Rule based approaches and Knowledge based composition approaches. Semantic approaches examine the service compatibility, usually take into account both functional and non-functional requirements and allow reasoning on which service is appropriate to be used in a specific situation. [Rao et al., 2004, Berardi et al., 2005a, Baryannis, 2014, Hatzi et al., 2015] refer to composition methods among others in this category. The approaches based on non-functional properties concern the quality of the execution process affected by the Quality of Service (QoS) attributes which are non-functional properties of services.

In the following, we give a brief description of some of the most important academic efforts to solve the composition problem. The focus of our review on the composition methods was to examine if and in what extent they address the composition problem

holistically and *in an intelligent way*, as [Fensel and Bussler, 2002] states, and whenever they take into consideration situations that may arise due to unexpected errors or other abnormal issues, in order to prevent errors and ensure the proper composition and execution of the delivered service.

Rao et al. in [Rao et al., 2004] use the Semantic Web service language DAML-S, for the external presentation of Web services, while, internally, the services are presented by extralogical axioms and proofs in Linear Logic. Web services that are specified using the OWL-S description are translated to extralogical axioms in Linear Logic. A theorem prover is used to produce the composite service. The logic models state evolution by keeping track of resources. The process model for a composite service can be generated directly from the proof provided by the prover. The composition system is based on a multi-agent architecture which enables the different components of the web service composition system, such as the theorem prover, semantic reasoner and translator, to be integrated in a loosely coupled manner.

McIlraith et al. in [McIlraith and Son, 2002] use an augmented version of Golog, which is a logic programming language built on top of the situation calculus and give a formalism for service composition. They extend Golog by enabling description of programs that are customizable and usable in the context of the Web and augment the ConGolog interpreter to include these enhancements. Web services are considered as primitive and complex actions with precondition and effects. The Web service composition is described as a planning problem and a composite service is a set of component services that are connected with GOLOG operators.

Berardi et al. in [Berardi et al., 2005a] implement a model based technique for composition using Finite State Machines (FSM). Service behaviors are modeled in FSM and they are expressed by formulae in Deterministic Propositional Dynamic Logic. If the set of formulae is satisfiable, a FSM can be produced with the model of the composite service. In [Berardi et al., 2005b] the Colombo framework is presented, where web services are characterized in terms of (i) message exchanges, (ii) data flow and (iii) effects on the real world. The work was continued in [De Giacomo et al., 2013] where they are supporting partial descriptions of service behaviors.

Lécué et al. in [Lécué et al., 2008] exploit the semantic representation of parameters like input and output of a service and perform service composition by matchmaking between these parameters. To enable interaction between two selected services, there should be semantic compatibilities between the output of the first service with the input of another one. In order to select the next one to be composed, the output of the last one used will be matched with the input of all candidate services.

Baryannis, et al. in [Baryannis, 2014] approach the composition problem by defining a new language, the Web Service Specification Language (WSSL) that is based on fluent calculus. By the language extensions they realize service composition via planning, include specification of QoS profiles and support partially observable service states. A service composition and verification framework is also implemented, where they perform experimental evaluation and investigate scalability performance in terms of execution time and optimality with regard to the produced composite process.

In [Foster et al., 2003] the composition of web services workflows is modeled and verified using the Finite State Processes (FSP) notation. Service specification is assumed to be expressed using UML Message Sequence Charts and is translated to FSPs using a preexisting tool. Concrete service implementations are expressed using BPEL4WS and a mapping from BPEL4WS to FSPs is provided. To illustrate how these compositions are verified, workflow scenarios using message sequence charts, together with a model-checking tool to interactively verify the workflow properties are constructed. Verification is then done by creating joint labeled transition systems (LTS) based on FSP specifications and checking trace equivalence. [Foster et al., 2006] follow up on the previous work with a tool for a model-based approach to verifying compositions of web service implementations. The tool supports verification of properties created from design specifications and implementation models to confirm expected results from the viewpoints of both the designer and implementer. They use finite state machine representations of services and assign semantics to the distributed process interactions. The tool provides verification results from the orchestrated Web service interactions.

[Hatzi et al., 2015] utilize the description of services in OWL-S and translate them to the PDDL domain to offer enhanced semantic capabilities through the PORSCE framework. By adopting ontology analysis they semantically enhance the planning domain and the composition problem stated. The framework computes hierarchy relationship and semantic distances in order to determine semantic similarity among OWL-S concepts. Planning is realized through VLEPPO and PORSCE framework, an integrated system that visualizes planning problems and integrates several different planning algorithms, allowing the user to experimentally explore various solutions.

In general, while there exist several approaches that go beyond simple syntactic based discovery, we observe that most of them perceive services as a collection of inputs and outputs and lack the holistic view of service behavior during their engagement for the composition. Most of the approaches analyze services composition technologies only from specific perspectives and present a technique which assembles web services in order to achieve a particular output.

2.2 Role Modeling

Classical conceptual modeling, like other modeling techniques do, has its strengths and weakness. The Entity-Relationship Model (ER) [Chen, 1976] and the Unified Modeling Language (UML) [Rumbaugh et al., 2004] for example, are good at capturing a system's structure, but they can not model the system's behavior as it dynamically emerges through collaborating entities. Several researchers, e.g. [Steimann, 2000, Guizzardi and Halpin, 2008] [Reenskaug and Coplien, 2009], have pointed out the deficiencies when they are used to model more complex, context-dependent and dynamic domains. For example, a class diagram describes the static structure of classes in the system and relationships between those entities. It represents the structure of the system and the summarized essence for its operation but it does not reflect the system dynamics itself [Nikiforova et al., 2011].

On the contrary, role is a powerful modeling concept that naturally captures the behavior of participants in a collaboration. Roles are used in many areas like Multi Agent Systems (MAS), Object Oriented modeling and Programming Languages, Organization and Security. There are many definitions of roles, each one with an approach based on intuition, practical needs and, sometimes, on a formal account. In security, roles are seen as a way to distribute permissions, in organizational models roles give powers to their players in order to access an institution, in MAS roles could be seen as descriptions of the behavior which is expected by agents who play them [Genovese, 2007].

[Kühn et al., 2015] gathers twenty six features attributed to roles as they are described in different researches, these are:

1. Roles have properties and behaviors
2. Roles depend on relationships
3. Objects may play different roles simultaneously
4. Objects may play the same role (type) several times
5. Objects may acquire and abandon roles dynamically
6. The sequence of role acquisition and removal may be restricted
7. Unrelated objects can play the same role
8. Roles can play roles
9. Roles can be transferred between objects

10. The state of an object can be role-specific
11. Features of an object can be role-specific
12. Roles restrict access
13. Different roles may share structure and behavior
14. An object and its roles share identity
15. An object and its roles have different identities
16. Relationships between roles can be constrained
17. There may be constraints between relationships
18. Roles can be grouped and constrained together
19. Roles depend on compartments ¹⁰
20. Compartments have properties and behaviors
21. A role can be part of several compartments
22. Compartments may play roles like objects
23. Compartments may play roles which are part of themselves
24. Compartments can contain other compartments
25. Different compartments may share structure and behavior
26. Compartments have their own identity

Based each time on a different set of the aforementioned features roles have been often used to model the semantics of a system and they are especially applied to model complex and dynamic domains, since roles are able to capture both context dependent and collaborative behavior of objects. Over the years, different approaches for role based modeling methods were introduced in various fields of research ranging from data modeling to conceptual modeling, in object oriented software engineering [Zhu and Zhou, 2008], in business process modeling [Ertugrul and Demirors, 2015] and in agent engineering [Kazik, 2010].

Object Oriented Role Analysis and Modeling method (OOram), developed by [Reenskaug et al., 1996] and [Andersen and Reenskaug, 1992], present a general approach to the

¹⁰ the term compartment is used since its definition, as “objectified collaboration with a limited number of participating roles and a fixed scope” encompasses all of the other notions are used in other studies.

modeling of objects and collaboration. Their objective is to provide a tool that supports a seamless transition among the conceptual modeling (analysis/design), the implementation and configuration phases. A role is a “why” abstraction that has an identity, a behavior and a state that is encapsulated. It is an archetypical representation of the objects occupying the corresponding positions in the object system. All objects that serve the same purpose in a structure of collaborating objects in a certain context are said to play the same role.

Riehle and Gross study the problem of class and object collaboration complexity and their impact on framework design and integration [Riehle and Gross, 1998]. They introduce a role modeling approach for object-oriented design. A role type describes the view one object holds of another object. A role instance represents one specific role of an object to its clients. The role wraps the object which maintains its role instances for different clients. Cooperation is taken as a set of roles and their relationships. Hence, the design of a framework can be composed of role models, each of which may be a pattern instance. A role model actually describes a particular aspect of an object, so the framework supports the separation of concerns.

Tsiakmaki et al. in [Tsiakmaki et al., 2011] have presented an Upper Ontology for the modeling of Semantic Learning Domains based on role modeling. The major issue of their research is in delivering quality educational learning services by adapting them to suit the needs and requirements of the learner.

In [Kritsimalis, 2010] the author transform the preamble of a CDL package of a choreography into a role model in the sense that it does not treat the interactions of participants as a predetermined static plan but defines the interaction behaviors of the involved roles in a way that it is possible for them to decide during their acting.

The employment of roles is also present in business processes modeling [Ertugrul and Demirors, 2015]. Business processes have a high degree of collaboration since, usually, a number of participants are involved in different parts of the process [Riemer et al., 2011].

In agent oriented software engineering the role modeling approach is used as a method for specifying, analyzing, and designing systems based on roles that agents will play emphasizing on specific characteristics of agents such as the autonomy and ability to collaborate. A role model is stated as an abstraction that describes the patterns of interactions among a set of entities. In general, roles are usually an abstract representation of behavior common to different classes of agents. Moreover, it serves as an interface, through which the agents obtain their knowledge of the execution environment and how to affect it [Kazik, 2010]. Kendall [Kendall, 1999] also applies the role modeling in design of agent oriented software engineering.

Concerning the usage of roles in heterogeneous teams, researchers in [Bosse et al., 2015] use role modeling to support missions that are performed by collaborative team of

humans and technical systems in space flights. In the proposed model, they use intelligent electronic partners (ePartners) who can play a useful role in supporting human-robot teams in their problem solving process whenever a non-nominal situation is encountered.

The core functions of ePartners include:

- Utilization of procedural rules in order to respond fast to an anomaly, this can be performed fully automatically (e.g., using standard fault detection, isolation and recovery procedures)
- The procedural responses, which implement intelligence of the participants and they are complemented by the ePartner response that can apply its global intelligence (about the environment, mission goals, etc.) to establish a knowledge-based solution. Such processes are enabled by ePartners and would not have occurred without them. Without ePartners some local solutions would lead to an end solution.
- The cognitive task load of the human is minimised, but humans are taken into the loop if needed.
- The ePartner serves as a communication intermediary which raise the level of abstraction to a level which is shared among all team members.

To formally represent the functionality of human-robot teams in terms of different levels of abstraction the Object-Role Modeling (ORM) framework is used. ORM is called so because it pictures the world in terms of entities that play roles (parts in relationships). It is typically more expressive than ER or OO. Its role-based notation makes it easy to specify a wide variety of constraints, and its object types reveal the semantic domains that bind a schema together. One benefit of this is that conceptual queries may be formulated in terms of schema paths, where moving from one role through an object to another role is made with a conceptual join.

ORM use graphical symbols that are based on first order predicate logic and set theory, to enable the creation of an unambiguous definition of an arbitrary universe of discourse. It is one of the standards used by the European Space Agency, and many ORM models are already available in the space domain. For the mission scenario presented in Figure 2.3 an ORM ontology has been created to express all relevant information for all concepts and their relations. However, ORM models can be used not only to express static information about the mission, such as the fact that a task requires a particular functionality. Additional mechanisms let the system reason about the dynamics of the mission. Such reasoning is relevant, for instance, to conclude that the status of a task changes once that task has been completed, but also to relate a malfunction in a resource to the conclusion that a task, a process, or even the entire mission may be compromised.

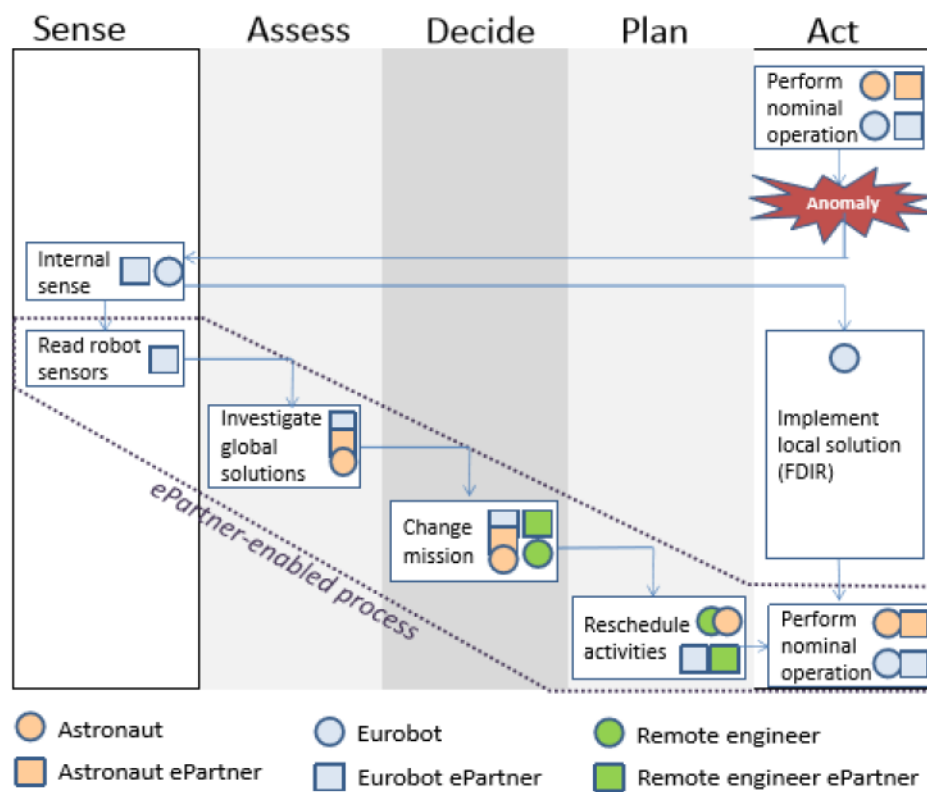


Figure 2.3: Overview of a joint human-robot team decision making process [Bosse et al., 2015].

Hence, dynamic reasoning rules could enable the system to make inferences and navigate between layers of the abstraction hierarchy in a flexible manner. A number of reasoning rules have been specified at a semi-formal level and these are (i) Observability rules, (ii) Predictability rules, (iii) Directability rules and (iv) Notification rules.

Overall, roles as a modeling concept, are used in different disciplines. In Table 2.1 we list the main modeling directions that roles have been used.

As it is concluded from the related work, the notion of role is a pivotal concept to model interaction among entities and can help to tame the increased complexity in dynamic domains. It enables to model participants as accurately as possible while be context dependant. Role modeling has been used to model interaction behaviors that are repeated and role models are considered as architectural patterns that depict the high level similarities between related systems. While roles are established in modeling languages their semantics differ and their full potential is not utilized. Overall, cooperation of web services during their composition could also be described as a sequence of interactions that are repeated and this is the rational about choosing role modeling as a technique to model services to describe and solve the composition problem.

Table 2.1: Modeling with Roles

Type of Modeling	Focus	Reference
Object Oriented conceptual Modeling	Modelling of objects and their role within the software system. Roles define relationships between objects.	[Zhu and Zhou, 2008] [Reenskaug et al., 1996], [Andersen and Reenskaug, 1992]
Object collaboration	Role models are used to define interaction among clients with roles and their relationships.	[Riehle and Gross, 1998]
Business Process Modeling	Use roles in Business Process Modeling	[Ertugrul and Demirors, 2015] [Riemer et al., 2011]
Agent Engineering (AE)	In AE is used for specifying, analyzing, and designing systems based on roles that agents emphasizing on specific characteristics play such as the autonomy and ability to collaborate.	[Kazik, 2010] [Kendall, 1999]
Heterogeneous teams of humans and systems	Use role modeling to support missions that are performed by collaborative team of humans and technical systems.	[Bosse et al., 2015]

2.3 Teamwork

Over the last decades, teamwork has become a thriving area of research and academic efforts, with a substantial body of research work produced. Several surveys with teamwork reviews have been published but many of these surveys were conducted at the earlier stage of teamworking and recent advancements were not covered and moreover they solely analyze teamwork behavior from specific perspectives and lack an integrated view of teamwork in relevant with this thesis areas. Consequently, we did a review with the aim to survey the existing major models and frameworks proposed on human and agent teamwork behavior and in this section we present the results. We contribute with this review in teamwork research by (i) providing an integrated table of major human roles that gives a holistic view of the current models on humans, (ii) presenting the major models-frameworks of team behavior proposed in agent based team working, (iii) identifying and analyzing the different teamwork factors among variant types of teams in agent based models and (iv) utilising these teamwork factors in service teamworking. The rationale is to lay the groundwork to teamwork for services that are called to form groups in dynamic environments, by employing some major roles that are used in human teams and habilitate their behavior with the key factors identified in agent based team working studies.

2.3.1 Different Approaches of Teamwork Modeling

A team creates additional value compared to the group [Mathieu et al., 2008]. From a broad perspective, only some of the advantages of teamwork in setting up work teams to organizations to increase productivity, innovation, and employee satisfaction [Rousseau, 2006]. Consequently, a variety of models following social - psychological approaches for human team formation, which involve different strategies and approaches has developed during the last decades.

At the same time, a plethora of research exists about group collaboration, where other type of entities like agents, robots, etc. construct a group and need to act in the same manner human members do. In the same way, during the last decades, agent collaboration have been thoroughly studied and motivated by psychological studies on human team formation. Software agent teamwork defined as “a set of agents having a shared objective and a shared mental state” [Sycara and Sukthankar, 2006]. In team-oriented agent architectures, agents have better understanding about team members since they are empowered to make critical decisions by proactively gather and share information with teammates and they can build coherent teams with other agents to effectively work on problems [Fan and Yen, 2004]. Additionally, the idea that, human, robots and agents could become an integral part of teams has been developed from a promising vision into

reality [Hinds et al., 2004, Jung et al., 2017]. Research in robot teamwork and human-robot teamwork has gained interest in academic and industrial research groups, motivated by real world applications for robot-human teams. Since the available technology has enabled use of robots in small teamwork settings in industry [Takayama and Go, 2012], robots are increasingly becoming a central part of teamwork [You and Robert, 2017], widely applicable to a number of domains, for instance, search-and-rescue teams employ remote-control robots that help respond to emergencies. Also, autonomous systems are frequently employed as part of groups and teams, for example by performing delivery tasks in hospitals [Ljungblad et al., 2012] or by working closely alongside people on manufacturing floors [Saupé and Mutlu, 2015].

2.3.2 Human Teamwork

Several authors aimed at identifying and measuring the factors that are likely to increase the effectiveness of teamwork in humans [Tannenbaum et al., 1992, Mathieu et al., 2008, Berlin et al., 2012, Parker, 2008]. According to the literature review we have done [Tsoutsas et al., 2019] the research on teamwork has been fragmented through the years in two main trends. These are (i) modeling teamwork according to team and team process characteristics and (ii) modeling teamwork according to role – behavioral attitude of team members. In the first category the studies witness the many different team characteristics, such as team structure, team size, team composition, physical distribution and team process characteristics, such as communication, coordination, shared knowledge, adaptability and planning which are critical to the success of a team. In the second category, which is central to this thesis, team models are based on role behaviors of participants. A role behavior is defined as the patterns of the behavioral characteristics of the ways in which one team member interacts with others in order to facilitate the progress of the team as a whole [Belbin, 2011]. In a team, the existence of team members that play predefined roles such as a coordinator, organizer, promoter etc. is a necessity for the performance of a team. In addition to role existence, studies defend the idea that high performing teams need to have a balanced representation of all team roles.

Overall, the research involved in this thesis draws upon studies where human teamwork models solely focus on roles and role behaviors, since we want to exploit it for a teamwork based model for services that is based on these elements, i.e. roles and behaviors. As such, we are interested in developing a review table with the series of key roles suggested in the various proposed models that can potentially be applicable to the role modeling approach which we have raised in the previous section. We proceed, by giving definitions about teamwork behaviors and teamwork roles, in order to distinct them from roles and behaviors of the role model context were introduced in the role modeling approach.

According to [Rousseau, 2006] teamwork behavior is a multifaceted concept that has been difficult to conceptualize. A team member's behavior involves (i) all the operations-related activities to be performed while working in the team and (ii) all activities that are based to the existence of the work team. [Aritzeta et al., 2007] defines the "role" concept from a psychosocial perspective as the behavior expected from an individual occupying a specific position. The cognition and expected behavior associated with the position are fundamentally important to success in the role. Similarly, Belbin defines "the tendency to behave, contribute and interrelate with others in a particular way" as a team role [Belbin, 2011]. Consequently, a role that a member of a team plays can be described by the behavior it expresses during the occupation of the specific position and this is the way that the terms role and behavior which will be used throughout this chapter. In general, teamwork roles follow the categorization of behaviors and they are divided into two main categories, (i) roles that are task-oriented and are concerned to solve tasks in the team and (ii) roles that are team-oriented, i.e. they are concerned with promoting group-centered behavior [Aritzeta et al., 2007]. Since teamwork behavior is required for effective team performance of team members, teams rely on different members to play a teamwork role and fulfill different needs through their behaviors [Aritzeta et al., 2007].

2.3.3 Human Based Teamwork Models

Human team working has been thoroughly studied by psychologists and human resource experts over the last decades. During these years, many human teamwork models have been proposed focusing on concepts that involve different strategies and approaches. In early studies, authors concentrated on specific characteristics that team members should have such as personality, functional expertise, competencies, goal orientations, teamwork orientations, e.t.c. [Delarue et al., 2008, Mathieu et al., 2008]. As the team research matured, research moved firmly from dealing with single characteristics that teams and members have to a variety of behaviors that team members expose [Rousseau, 2006, Salas et al., 2008]. Ultimately, in recent studies member's tasks and behaviors are clustered into distinct roles within the system that are aligned with the expertise of team members [Kozlowski et al., 2015]. Teamwork models are the evolution of isolated teamwork characteristics used. In teamwork models researchers propose that teammates, along with the operational tasks that they perform in a team, they have also to play some other role such that of coordinator, contributor, idea generator, e.t.c. Indeed, as it is witnessed by empirical studies, these approaches are effective for certain contexts, tasks and domains [Berlin et al., 2012].

The major human teamwork models were examined and are briefly described in the following:

1. The model of McCann and Margerison [McCann and Margerison, 1989], [Margerison and McCann, 1990] is based on team roles of eight different types that are: (a) explorer–promoter, (b) assessor–developer, (c) thruster–organizer, (d) concluder–producer, (e) checker–inspector, (f) upholder–maintainer, (g) reporter–advisor, and (h) creator–innova-tor. The authors recognize that individual work-style preferences that humans bring to teamwork are key factors influencing individual and team performance. When human teams are constructed and put together on an ad hoc basis with no choice of members, then the teams have potential weaknesses since some roles are not present. They recognize that team behavior is related to the tasks and the roles that members play. Additionally, high team performance is connected with teams that are well balanced with respect to the team roles of the various team members along with good linking skills of members.

2. The work of *Belbin* [Belbin, 1981] on team roles contained eight clusters of behavior that individuals adopt when participating in a team for constructing successful management teams. The model consists of eight roles, where each role is defined as the member’s tendency to behave, contribute and interrelate with others in a particular way. In sub-sequent research [Belbin, 1993] renames some team roles, and a ninth role, the Specialist, is added. These nine roles are (a) Plant, (b) Resource investigator, (c) Co-ordinator, (d) Shaper, (e) Monitor evaluator, (f) Implementer, (g) Team worker, (h) Completer–Finisher and (i) Specialist. Belbin defends the idea that high performing teams need to have a balanced representation of all team roles. The team role balance hypothesis assumes that if all team roles are present in a team then it will perform better than other teams without this balance. The team role that a member plays is distinguished from the functional role that refers to the technical skills and operational knowledge relevant to the member’s job. Consequently, several people may have the same functional role but vary greatly in their team roles [Aritzeta et al., 2007].

3. *Parker* in [Parker, 2008] developed a set of team player styles, which fall into four types: (a) Contributor, (b) Collaborator, (c) Communicator, and (d) Challenger. The styles are categories in which human with similar characteristics are found. A Contributor is a task-oriented team member who enjoys providing the team with good technical information and data. After doing his work, he pushes the team to set high performance standards and to use their resources wisely. A Collaborator is a member of the team who is flexible and open to new ideas, is willing to work outside the defined role and is able to share with other team members. A Communicator is a process-oriented member that is an effective listener and a positive person in general.

A Challenger is a member who questions the goals, methods, and even the ethics of the team, is willing to disagree with the team leader and other members and encourages the team to take well-conceived risks. While Margerison's and Belbin's team roles are based on Jungian tradition, Parker argued that his style-based team roles provide more latitude for members in taking different team roles and feedback for improvement [Mathieu et al., 2015].

4. According to *Salas et al.* in [Salas et al., 2005], team effectiveness takes a holistic perspective in considering not only whether the team performed and completed the team task but also how the team interacted to team processes to achieve the team outcome. They argue that, regardless of the team task that is examined, a set of teamwork components will be required to complete the task and present these components as the "Big Five" of teamwork. These five components are team leadership, mutual performance monitoring, back up behavior, adaptability and team orientation. Moreover, they suggest that these factors improve performance as long as three supporting mechanisms are also present within the team, which are shared mental models, closed loop communication and mutual trust.
5. *Mathieu et al.* in [Mathieu et al., 2015] reviewed and synthesized previous team research and argued that a team can be composed by members occupying six different team roles, which are referred to as Team Role Experience and Orientation (TREO) dimensions. The roles of each member are defined by taking into account the combination of its previous experiences and orientations. The authors notice that team composition serves as the foundation upon which other team factors are built, and represents a key enabling feature of teams. They synthesize the previous work of Belbin, Margerison's and other researchers work that focus on team roles and, finally, propose a six-dimensional framework consisting of the following roles: (i) Organizer, (ii) Doer, (iii) Challenger, (iv) Innovator, (v) Team Builder and (vi) Connector. Moreover, they validated the TREO model by measuring the outcomes and its predictive validity.

Although the existing literature has suggested numerous team roles, in this thesis we adopt Margerison et al. taxonomy [Margerison and McCann, 1990], since their study was among the first ones that proposed role behaviors in the extent that they are still used; the first version of the model was released during 1984 [Margerison and McCann, 1984]. In the following, there is a description of the eight roles necessary for a successful team composition, as they are recognized by Margerison's study. The roles are defined according to the key work elements need to be addressed in order for a team to be successful. The tasks and responsibilities of team members are related to work elements addressed and

they are all covered by the critical roles construct the team management wheel.

The goals and behavior of the teamwork roles as they were proposed by [Margerison and McCann, 1990] are:

- (i) Creator-innovator: People that play this role are future oriented and enjoy thinking up new ideas and new ways of doing things. They come up with new strategies and different approaches to tasks, creating new ideas and experimenting with them in order to handle various situations and challenges. They are looking to tomorrow rather than worrying about today. An organization needs creators-innovators to invent new products or services or ways to improve existing ones. They aim at creating fundamental ideas that will move forward the organization. Research and development departments need to be stuffed with employees that play this kind of role.
- (ii) Explorer-promoter: People that play this role enjoy being with other people and they are excellent at taking ideas and promoting them. The promoter helps bridge, connect members of the team with members outside of the team working in a twofold aspect by searching for new opportunities and exploring possibilities inside, and outside of the domain. They are good at bringing back contracts and resources that can help the organization move forward. They are excellent at seeing the “big picture” and they can capture opportunities. While innovators think up ideas, promoter’s task is to sell the ideas for the organization. Many good ideas may be lost simply because they are purely promoted from the organization since no member of the team is playing this role.
- (iii) Assessor-developer: People that play this role have characteristics of both explorers and organizers. They are excellent at making someone’s else idea work in practice. They can take an idea and match it to an opportunity without caring about organizations’ constraints. They respond well to new challenges and they can push forward an idea into a workable scheme. After an idea has been sold, the work activity follows often is based to the developers. Doing the project’s work often requires further development, which may involve prototype testing or checking alternative versions of the idea before implementation.
- (iv) Thruster-organizer: People that play this role enjoy making things happen. They are analytical decisions makers, always doing what is best for the task. Their aim is to implement ways and means of making things work on time and to budget. They monitor the global schedule and keep track of accomplishments and how teams are progressing relative to goals and timelines. They usually prefer to work according

to an established plan and in a structured manner so that objectives are clearly set out and everyone in the team knows what has to be achieved and by when. They organize others so that deadlines are met. They are task oriented, they set objectives, establish plans and then press for action. In an organization, once a project is ready to be implemented, plans have to be made, budgets approved and schedules established so that the project can be implemented according to deadlines. These activities are part of the organizers tasks.

- (v) Concluder–producer: People that play this role are practical and can be counted on to carry out things through to the end. Their most important goal is to work on a systematic way to produce work outputs. Their strength is in setting up plans so that output can be achieved on a regular basis. For this role, it is a challenge the motto “job well done”. They are usually more patient than other members and have the ability to work in a reliable and stable manner to deliver results. Since the production of the organization has to be done on a regular basis according to standards and effectiveness, producer roles are appropriate for the production work function.
- (vi) Checker–inspector: People that play this role enjoy the detailed side of work and like working with facts and figures. They are usually careful and can spend long periods of time on a particular task. They check and audit that systems and contracts are working focusing on the detailed and controlling aspects of work. They are comfortable working within rules and regulations that others have established to ensure that quality is maintained in an organization inspecting activities that are important. Often, in organization team projects are driven to fail due to weakness in inspecting. Members that play the inspector role are ideal to end such weaknesses.
- (vii) Upholder–maintainer: People that play this role have high concerns for other people and they strongly support others that share the same ideas with them. They prefer to work in a supportive environment where they make sure things are done according to their standards. Their goal is to ensure that standards and processes are upheld and to maintain team excellence. They prefer to take an advisor role rather than an executive role. All teams need to have a solid infrastructure so that all other activities can proceed in the most efficient manner. Personnel in administrative positions need to have maintaining activities through their supporting functions.
- (viii) Reporter–advisor: People that play this role are excellent in gathering information and putting it together in an understandable way. They are patient people who prefer to make sure they have all the information before they take action. They foresee

conflicts might happen and they usually move to defuse the conflict themselves. People working for information offices need to play this role. They spend most of their work time in advising activities, collecting data and passing them on for action to others in the team.

2.3.4 Integration of the Roles of the Major Human Teamwork Models

There were many attempts to classify roles and behaviors in the context of human team working. In fact, as it is witnessed in [Driskell et al., 2017], there is a seemingly unlimited number of roles that have been described as relevant to human team performance. For this reason, in this research, they are only selected and mapped five pivotal studies of the field. The complication, which in general is faced during the classification, is that different studies have often been used different names to refer to the same role or similar names have been used to refer to different roles and behaviors. The most common ways of organization used to previous reviews include classification according to dimensions of teamwork behaviors through a subset of teamwork characteristics that can potentially be grouped in to a specific teamwork behavior. Not all researchers state explicitly that they use roles or behaviors but it is subsumed so. More details about the diversity of ways that teamwork behaviors is conceptualized can be found at [Rousseau, 2006].

To have a holistic view of major human team roles, we draw upon five substantive studies and following the classification of roles is proposed in Margerison et al. as a base taxonomy, map the roles and behaviors from the above mentioned studies to the base taxonomy in Table 2.2.

The variance of the number of roles proposed by each model is noticeable in Table 2.2. The smallest number of roles which is four is proposed in Parker's study, while the study that proposes more detailed roles is that of Belbin where nine roles exist. Belbin's classification was based on observations of managerial personnel primary in business environments while during last years it is extended in nursery and education personnel. Mathiew et al. (2008) fuse the nine roles of Belbin's model to six while Salas et al. (2005) propose six behavioral attitudes which is near to the average of roles proposed in general. The number of different roles that is provided in a model provides an illustration of how roles structure the team interaction. Although these roles may vary within any particular team, it is accepted that the variation of the basic structure underlies the nature of the team performance. But the basic role structure of the team is less consensus when models with unspecialized team roles are used. In such a context, Aritzeta et al. (2007) raise the question about the existence of well-differentiated team roles in proposed models and whether these team roles potentially are better differentiated using some other grouping.

The next observation on table's content regards to the overlapping behaviors that exist

among roles in different models. A problem with the role taxonomies was that there was considerable divergence in terms of the team roles described and the context examined. The challenger role, for example, of Mathieu et al. (2008) model presents a behavior that matches both to explorer's and to reporter's behavior of the Margerison (1990) model. Finally, it is apparent that in models that the number of roles is small, the roles are more concentrated whereas in models that a large number of roles is proposed, there is a more detailed overall description for behaviors appearing in the team while the necessary expertise of members to play these roles is distinguished. In general, roles are not simply static sets of expectations and behaviors, they instead change and adapt according to role demands [Driskell et al., 2017]. However, we conducted the analysis of each model to identify what are the primary core role dimensions that we observe in these team models.

The main goal for the development of the mapping table is to create a basic structure of roles used in human teams. In Table 2.2 we integrate previous taxonomies within the team roles research literature and provide a deep perception of the structure of team roles. In other words, we want to determine whether there exists a primary set of role dimensions that are descriptive of performance in various team models proposed, in order to better understand the structure of team roles and how they are related to roles of other models. Once significant role dimensions have been identified, this may suggest a new way of approaching the teamwork roles during team formation.

Role name according to Margerison (2015)	Belbin (1981, 1993)	Parker (2008)	Salas et al. (2005)	Mathieu et al. (2008)
Explorer Promoter	Resource Investigator	Challenger	Team orientation	Challenger, Connector
Assessor Developer	Implementer, Team Worker	Collaborator		Team Builder
Thruster Organizer	Co ordinator, Shaper	Collaborator, Communicator	Team leadership, Backup behavior (Delegation), Adaptability, Team orientation	Organizer
Concluder-Producer	Plant, Implementer, Completer-Finisher	Contributor		Doer
Checker Inspector	Completer Finisher, Monitor Evaluator	Communicator	Mutual performance Monitoring, Backup behavior (Coaching)	Organizer
Upholder Maintainer			Backup behavior (Delegation), adaptability	Organizer
Reporter Advisor	Specialist	Contributor	Backup behavior (Assist)	Challenger
Creator Innovator	Idea generator	Contributor	Adaptability, team orientation	Innovator

Table 2.2: Table with mapped roles of selected models to Margerison model [Tsoutsas et al., 2019].

2.3.5 Teamwork Theory to Agent Based Models

Apart from human team working, another area where similar teamworking problems have been studied and such theories are applied is the area of software agents. As with humans, a group of software agents must accomplish given tasks by organizing themselves according to their individual characteristics and their role within the overall system. Furthermore, complex teams that comprise of humans and agents were studied, since in the real world scenarios, agents play a variety of roles in their interactions with humans; they support humans in tasks that they jointly perform or independently work to carry out tasks for which they are responsible [Van Wissen et al., 2012]. Examples of applications for human-agent teams include autonomous agent pilots that work together with people to complete a joint mission, mission execution assistants for astronauts [Milind Tambe, 1997], personal assistants interacting with users, e.g. Siri for iPhone and Cortana for Windows Phone [Kato et al., 2014] and hybrid teams consisting of humans, robots and virtual characters situated in industrial production environments [Schwartz et al., 2016]. Considering that humans and agents have different capacity and capabilities in order to combine their strengths, it is crucial to have good coordination [Bradshaw et al., 2012] and other behaviors that humans do. In such a context, the studies on agent teamwork have been motivated by concepts and theories that were developed on human teamwork. In doing so, common characteristics with human team working are used, e.g. planning, coordination, collaboration.

While past studies on team formation focused on interactions only between human or agents, in recent years there are studies with heterogeneous teams of humans, agents and agents in robotic settings, with agents or robots acting autonomously alongside humans as peers. In such heterogeneous teams, most studies are based on the assumption that the non-human team members need to act in the same manner that human ones do. Examples of such teams are teams of robots in rescue operations [Beck et al., 2016] and teams of humans, robots, and virtual agents cooperating in production settings [Schwartz et al., 2016].

Increased research towards this direction is encouraged by today's industry environment that is so complex and work is done in teams composed of members that are either humans, agents or robots where each one is specialized in specific tasks. For example, humans together with various e-commerce cooperative agents, created by different companies and having different standards, need to collaborate while the successful result for the endeavor depends on the team rather than the individual effort of the member. When referring to team with such content the term agent based teams is used in this thesis. In the context of Industry 4.0, embedded systems will be linked with organizational processes to transform

industries and this will enable the real time connection between humans, machines and smart objects. In other words, Industry 4.0 (i4.0) envisions a future where, in industries, humans and Cyber Physical Systems are effortlessly connected while exhibiting teaming behavioral attitudes. In order for these entities to effectively be connected and working as a team, models referring to team working in literature should be investigated.

The term agent is generally overloaded and is used in different disciplines to refer to different entities or artifacts. In artificial intelligence, agents are defined as autonomous entities that observe and act upon an environment and direct their activities toward achieving goals [Russell and Norvig, 2010]. Agents are either developed to work isolated or in collaboration with other entities to achieve a goal. This covers a variety of possible uses for agents, from agents that act auxiliary with limited autonomy to agents that respond to particular tasks in a manner that humans do [Chen and Barnes, 2013]. A lot of research effort was devoted to determine the basic dimensions that define agent team working in order to reach an optimal teamwork behavior [Jennings et al., 1998]. In dynamic environments, a team of agents is more than a group of intelligent entities working towards achieving a goal. An agent, in order to reach a certain level of intelligence, autonomy and team work behavior, has certain properties such as reactivity, proactivity, social ability, decision making ability, etc. Like other team members, they need to be coordinated using shared knowledge, collaborate by working together on common tasks and exhibit more teamwork characteristics.

The requirements for agents that interact with humans in a practical manner demand that agent capabilities need to be augmented with techniques that facilitate effective team abilities. In particular, agent based teamwork, where human and software agents interact as peers, is concerned with joint commitments and joint responsibility, requires establishing situation awareness collaboratively, developing shared mental models as situation evolves, adapting to mixed-initiative activities, etc. Furthermore, successful agent technologies take advantage of the diversity within teams and the differences between human and agent strengths, i.e. human reasoning has different characteristics than algorithmic reasoning [Bradshaw et al., 2012]. Although many theoretical and applied techniques have been applied and researchers have attempted to build on and integrate the work of others, due to the problems inherent complexity, most studies only focus to some of the identified challenges and do not treat the problem holistically. Moreover, it is imperative for models in agent based team working to be adapted and take into account concerns that arise from the development and application of new emerged technologies such as IoT, i4.0, CPS, etc. In the following, some major teamwork concepts and theories concerning the agent based teamwork research are presented that have contributed to the research of the agent team working problem.

One of the first theories, namely team cognition theory, was used for explaining team functioning [Cannon-Bowers et al., 1990]. In this theory, the concept of *Shared Mental Model (SMM)* is fundamental in both the context of human teams and heterogeneous teams of humans and agents. A mental model is a mechanism where humans are able to generate descriptions of system's purpose, form explanations of system functioning, observe system states and predict future system states. A mental model is shared among team members and may include: (i) task-specific knowledge, (ii) task-related knowledge, (iii) knowledge of teammates and (iv) attitudes/beliefs. The idea behind SMM is that team performance is improved if team members have a common understanding of the task and what needs to be done to achieve that task. This includes the task to be performed, knowledge and beliefs, goals and intentions and plans of the involved members. It actually refers to an overlapping understanding among members of the team, either humans or agents, regarding their objectives [Yen et al., 2006]. This is achieved by developing knowledge structures held by members of the team that enables the adaptation of their behaviors to the demands of the team.

[Jonker et al., 2011] present a more detailed definition of the concept of SMM by formalizing it with a mental model ontology. This is captured in the UML model in Figure 2.4. The notion of the system is central to the definition of mental model. In the context of shared mental models they are especially interested in a certain kind of system, namely a team. According to their definition of system, a team can be viewed as a system: it consists of a set of interacting team members, forming an integrated whole. A mental model resides in the mind of an agent. Following the diagram, they express that an agent has a mental model of a team since an agent has mind and a mind can have mental models. A mental model concerns a team, since a mental model is a model and a model concerns a system, and a team is a kind of system. Moreover, the team mental model has a part that describes team interaction, since the team mental model concerns a team, and a team has team interaction. The notion of goal expresses that a mental model is used by an agent for a certain purpose, expressed by the goal of the model. Several researches use SMM in a similar way to build teamwork models [Sycara and Sukthankar, 2006].

Another approach is *Joint Intention theory (JIT)* [Levesque et al., 1990] that focuses on a team's mental state joint by all members, which is called a 'joint intention'. Although the concept of joint intention is complex and constitutes subject of philosophical debate, the intuition here is relatively straightforward. In broadest terms, joint intention turns on the coordination between individual participants with respect to their individual beliefs, intentions, and the entailed subsidiary intentions, or sub-plans.

JIT stipulates what it means for agents to execute actions as a team. Accordingly, by using the JIT, [Cohen and Bailey, 1997] propose a formal approach to build artificial

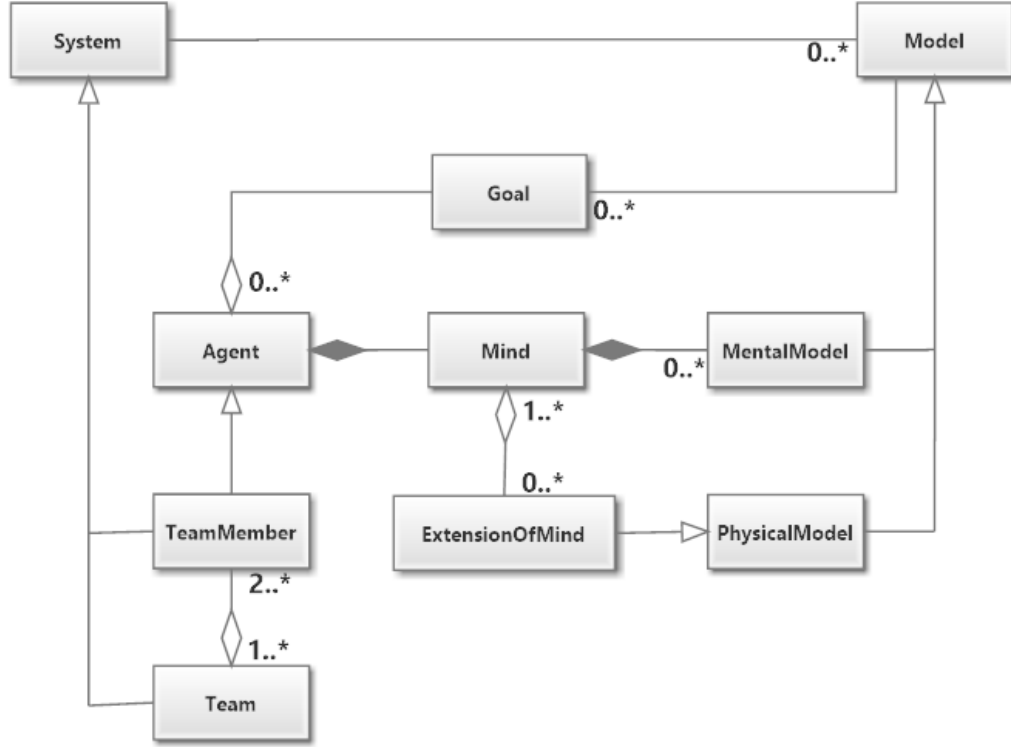


Figure 2.4: Mental Models [Jonker et al., 2011].

collaborative agents. Their view of JIT is the persistent commitment of the team to a shared goal and implies this commitment of all members to a mutual belief about the state of the goal. A way to make sense of, to accurately describe, the action is as a joint one where agents have common intentions towards the group goal; they demonstrate commitment and focus on handling unexpected failures. In order to enter a joint commitment, a team member that is in a specified mental state, communicates to establish appropriate mutual beliefs and to coordinate its actions towards the shared team plan. Members inform the team when they reach the conclusion about whether the goal is achievable or impossible. The concept of joint intention is meant to allow to create an account of the intentions behind such joint action products without undermining the idea of individual intentional agency and the norms of individual rationality associated with it.

This same theory is used to specify formal semantics of communicative acts as an extension of standard speech act theory [Kumar et al., 2002]. The JIT is expressed in a modal language with the usual connectives of a first order logic with equality, along with operators for propositional attitudes and event sequences. The primitive mental states include an agent's beliefs and goals. Temporal operators include $\Diamond p$ (eventually p) and $\Box p$ (always p). An action expression consists of action sequences, non-deterministic choices, concurrent actions, indefinite repetitions, and $p?$ test actions. Constructs for conditional and iterative actions can easily be built. Details of this modal language and its model

theory can be found in [Cohen and Bailey, 1997]. The notion of an agent’s commitment to achieving some state in the world is expressed as a persistent goal or PGOAL [Levesque et al., 1990]. An agent x has a persistent goal (PGOAL x p q) if x wants p to become true and cannot give up the goal until it believes that p is accomplished, or is impossible, or is irrelevant (i.e. the relativizing condition q is untrue). An intention to do an action is defined as a persistent goal in which the agent is committed to performing the action believing throughout that it is doing that action. A persistent weak achievement goal or PWAG represents the one-way commitment of one agent directed towards another and it is used to define the semantics of various communicative acts. The notion of teamwork is characterized by joint commitment. The definition of JPG states that the agents mutually believe they have the appropriate goal, and that they mutually believe a PWAG to achieve it persist until the agents mutually believe that the goal has either been achieved, is impossible, or is irrelevant. In logic, it can be shown that because of this joint commitment, agents have individual commitments that lead to individual intentions to do their parts relative to the overall joint commitment. Joint commitment further stipulates that agents bound together by JPG are committed to establishing these mutual beliefs in the event of private belief about any of those conditions.

Shared Plans theory originated from Pollack’s mental state of plans model [Pollack, 1990] and it was extended to provide formalization in order to accommodate group activity that handles complex actions while it allows plans to be partial and handles evolution of shared plans over time [Grosz and Kraus, 1996, Grosz and Kraus, 1999]. According to Pollack’s observation, there are different types of plans. More specifically, in agent planning, there is a difference between those plans that agents know that are applicable and potentially can be followed and those plans that agents adopt because they know the recipe to follow them and subsequently guide their action. Having a plan is not only knowing how to execute an action, but also having the intention to execute the actions required. The term “recipe” is used to describe how to execute an action while the term “plan” implies knowing and having a plan to execute the action. One of the focuses of shared plans is on specifying the capabilities and mental attitudes (beliefs and intentions) required for the participants of a group in an activity to have a collaborative plan. A shared plan for a group action implies that member’s individual and mutual beliefs about the action and its constituent sub actions are to be executed.

However, agents may differ in their beliefs about the ways to perform an action and in their assessments of the ability and intention to execute an action. To better infer the plan it adopts, it is necessary for an agent to analyze the nature of the mental state of the plan it has and to be aware about intents of other agent’s with whom it discourses. These activities require plan-based reasoning that arises from the participants’ attitudes

of intentions toward the actions of others and of the group as a whole. To model these accurately, the adoption of a mental state view of plans is essential. Using mental state is possible to accommodate differences among the beliefs of individual participants and distinguish between knowledge about action performance and intention to act. The theory provides a detailed account of an agent's motivations for initiation of communication that will lead to do an action. The extended model of shared plans deals with agents that initially have only partial knowledge of how to perform an action and distinguishes between partial and complete plans. Furthermore, it provides the rationale for the process of revising beliefs and intentions since both individual and team plans evolve over time. The formalization of shared plans has strong logical foundations and is defined in terms of modal operators, meta-predicates, i.e. abbreviations for complex formulas involving other predicates, modal operators and actions. It also can be used to define the propositional attitude of intending to do an action and introduces the attitude of intending that a proposition holds.

Ad Hoc team working is a relatively new research trend [Bowling and McCracken, 2005] and among other researchers is studied by [Agmon and Stone, 2011, Stone et al., 2010, Barrett et al., 2011]. In ad hoc teams the member's fundamental objective is to collaborate without pre-coordination. Ad hoc teams are heterogeneous teams of humans, agents or robots that are able to cooperate with multiple potential teammates without coordination in order to reach an optimal team behavior. [Stone et al., 2010] defined an ad hoc team setting as a problem in which team coordination strategies cannot be developed a priori. In their study they present an algorithm for determining the optimal strategy for an ad hoc agent to lead, with minimal cost, the best response agent to perform the action yielding optimal joint utility.

2.4 Discussion

The objective of this chapter is twofold: on the one hand, to give an overview of recent progress in web services composition techniques; on the other hand, to give an insight into role modeling and teamwork related models exist in human and agent teamworking. Regarding the first objective, we realize that several works exist on composition of web services, but despite this intense research activity there are still open issues. It is evident that we are still far from achieving fully automated and effective service composition. It is apparent how complicated the subject is, and that each part of the service composition process requires different approach, platforms and methods. By studying the background research, we derive a series of research topics that pose new or already identified challenges and drive the research that constitutes this thesis.

Firstly, we notice that the great majority of composition approaches that are based on compatibility of the service processes or service behaviors focus on properties that are related to inputs and outputs of services as well as message exchanges and conversations. Given that, modeling service composition solely takes into account to blindly follow the plan, i.e. the sequence of processes to be executed determining the input and output messages as well the precondition and effects involved that hold. This is made regardless any knowledge that could derive from the data of the environment that a member service might hold and could possibly contribute with, for example about whether an exchange message could cause an abnormal situation or whether an actual message exchange should be avoided due to an unexpected error that might occur.

Secondly, despite all the diverse efforts in service composition, we are seeing that the approaches ignore the business side of providing and exploring composite services. In practice, users have to develop and manage a variety of business services, sometime quite complex, that will enable them to access and exploit component services offered either by known providers like suppliers and customers or possible unknown third parties. Under this perspective, effectiveness of service composition is a much more complicated issue than the ability to provide a composition algorithm with a satisfying performance.

Finally, a basic weakness detected in most of research efforts on service composition, is that although web services work together as a group to achieve a common objective they do not use any teamwork abilities, as it happens with groups of human, agents or robots. The concept of developing *teamwork services*, i.e. autonomously composite services capable of exhibiting teamwork behavior, is introduced as a challenge in this thesis. A service capable of exhibiting teamwork behavior is one that can effectively cooperate with multiple potential teammates on a set of collaborative tasks and that is able to prevent or catch errors and emergent behaviors that will prevent the achievement of the overall team goal.

On completing the review on service description and composition, the common conclusion is that none of the studied approaches is capable of developing composite services as they envisaged by the author. According to our opinion, the proposed approaches that do not take into account a teamwork aspect of services during their composition do not achieve the required completeness. Our objective in researching composition of services is to benefit from teamwork, and instead of having a set of services that form a *group* to develop a *team* of services, i.e. a composite service having a *teamwork attitude*. Towards this direction, firstly, we make an attempt to couple human behavior and agent behavior to address the challenges associated during interconnection and enhance services by using them. Then, we propose web services to automatically cooperate using the role modeling approach by supporting service's interoperability through novel service teamwork roles.

We exploit the teamwork theory, by binding it with the role modeling approach that use roles and behaviors in order for the theory of teamwork behavior to be implemented.

This concludes the Background and Related work chapter. In next chapter, we continue by presenting how we exploit the role modeling to define services and their composition.

References

- [Agmon and Stone, 2011] Agmon, N. and Stone, P. (2011). Leading multiple ad hoc teammates in joint action settings. In *Interactive Decision Theory and Game Theory*.
- [Andersen and Reenskaug, 1992] Andersen, E. P. and Reenskaug, T. (1992). System design by composing structures of interacting objects. In *European Conference on Object-Oriented Programming*, pages 133–152. Springer.
- [Aritzeta et al., 2007] Aritzeta, A., Swailes, S., and Senior, B. (2007). Belbin’s team role model: Development, validity and applications for team building. *Journal of Management Studies*, 44(1):96–118.
- [Bano and Ikram, 2010] Bano, M. and Ikram, N. (2010). Issues and challenges of requirement engineering in service oriented software development. In *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*, pages 64–69. IEEE.
- [Barrett et al., 2011] Barrett, S., Stone, P., and Kraus, S. (2011). Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 567–574. International Foundation for Autonomous Agents and Multiagent Systems.
- [Baryannis, 2014] Baryannis, G. (2014). *A novel specification and composition language for services*. PhD thesis, University of Crete.
- [Beck et al., 2016] Beck, Z., Teacy, L., Rogers, A., and Jennings, N. R. (2016). Online planning for collaborative search and rescue by heterogeneous robot teams. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1024–1033. International Foundation for Autonomous Agents and Multiagent Systems.
- [Belbin, 1981] Belbin, R. M. (1981). Management teams: Why they succeed or fail. *Human Resource Management International Digest*, 19(3).
- [Belbin, 1993] Belbin, R. M. (1993). Team roles at work: A strategy for human resource management. *zitiert in: Teamarbeit und Teamentwicklung*, page 321.
- [Belbin, 2011] Belbin, R. M. (2011). Management teams: Why they succeed or fail. *Human Resource Management International Digest*, 19(3).

- [Berardi et al., 2005a] Berardi, D., Calvanese, D., De Giacomo, G., Hull, R., and Mecella, M. (2005a). Automatic composition of transition-based semantic web services with messaging. In *Proceedings of the 31st international conference on Very large data bases*, pages 613–624. VLDB Endowment.
- [Berardi et al., 2005b] Berardi, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Mecella, M. (2005b). Automatic service composition based on behavioral descriptions. *International Journal of Cooperative Information Systems*, 14(04):333–376.
- [Berlin et al., 2012] Berlin, J. M., Carlström, E. D., and Sandberg, H. S. (2012). Models of teamwork: ideal or not? A critical study of theoretical team models. *Team Performance Management: An International Journal*, 18(5/6):328–340.
- [Bosse et al., 2015] Bosse, T., van Diggelen, J., Neerincx, M. A., and Smets, N. J. (2015). Supporting human-robot teams in space missions using epartners and formal abstraction hierarchies. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 383–399. Springer.
- [Bowling and McCracken, 2005] Bowling, M. and McCracken, P. (2005). Coordination and adaptation in impromptu teams. In *AAAI*, volume 5, pages 53–58.
- [Bradshaw et al., 2012] Bradshaw, J. M., Feltovich, P. J., and Johnson, M. (2012). 13 human-agent interaction. *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*.
- [Cannon-Bowers et al., 1990] Cannon-Bowers, J. A., Salas, E., and Converse, S. (1990). Cognitive psychology and team training: Training shared mental models and complex systems. *Human factors society bulletin*, 33(12):1–4.
- [Chen and Barnes, 2013] Chen, J. Y. and Barnes, M. J. (2013). Human-agent teaming for multi-robot control: A literature review. Technical report, ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD.
- [Chen, 1976] Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1):9–36.
- [Chen et al., 2006] Chen, Y.-p., Li, Z.-z., Jin, Q.-x., and Wang, C. (2006). Study on qos driven web services composition. In *Asia-Pacific Web Conference*, pages 702–707. Springer.
- [Chinnici et al., 2007] Chinnici, R., Moreau, J.-J., Ryman, A., and Weerawarana, S. (2007). Web services description language (wsdl) version 2.0 part 1: Core language. *W3C recommendation*, 26(1):19.
- [Cohen and Bailey, 1997] Cohen, S. G. and Bailey, D. E. (1997). What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of management*, 23(3):239–290.
- [De Giacomo et al., 2013] De Giacomo, G., Patrizi, F., and Sardina, S. (2013). Automatic behavior composition synthesis. *Artificial Intelligence*, 196:106–142.
- [Delarue et al., 2008] Delarue, A., Van Hoetegem, G., Procter, S., and Burrridge, M. (2008). Teamworking and organizational performance: a review of survey-based research. *International Journal of Management Reviews*, 10(2):127–148.

- [Driskell et al., 2017] Driskell, T., Driskell, J. E., Burke, C. S., and Salas, E. (2017). Team roles: A review and integration. *Small Group Research*, page 1046496417711529.
- [Ertugrul and Demirors, 2015] Ertugrul, A. M. and Demirors, ..., O. (2015). An exploratory study on role-based collaborative business process modeling approaches. In *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management*, page 14. ACM.
- [Fan and Yen, 2004] Fan, X. and Yen, J. (2004). Modeling and simulating human teamwork behaviors using intelligent agents. *Physics of life reviews*, 1(3):173–201.
- [Feier et al., 2005] Feier, C., Polleres, A., Dumitru, R., Domingue, J., Stollberg, M., and Fensel, D. (2005). Towards intelligent web services: The web service modeling ontology (wsmo).
- [Fensel and Bussler, 2002] Fensel, D. and Bussler, C. (2002). Semantic web enabled web services. In *Advances in artificial intelligence: 25th Annual German Conference on AI*, volume 2479, page 316. Springer.
- [Foster et al., 2003] Foster, H., Uchitel, S., Magee, J., and Kramer, J. (2003). Model-based verification of web service compositions. In *Automated Software Engineering, 2003. Proceedings. 18th IEEE International Conference on*, pages 152–161. IEEE.
- [Foster et al., 2006] Foster, H., Uchitel, S., Magee, J., and Kramer, J. (2006). Ltsa-ws: a tool for model-based verification of web service compositions and choreography. In *Proceedings of the 28th international conference on Software engineering*, pages 771–774. ACM.
- [Genovese, 2007] Genovese, V. (2007). A meta-model for roles: Introducing sessions. In *Proceedings of the 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multiagent Systems, and Ontologies*, pages 27–38.
- [Grosz and Kraus, 1996] Grosz, B. J. and Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357.
- [Grosz and Kraus, 1999] Grosz, B. J. and Kraus, S. (1999). The evolution of sharedplans. In *Foundations of rational agency*, pages 227–262. Springer.
- [Guizzardi and Halpin, 2008] Guizzardi, G. and Halpin, T. (2008). Ontological foundations for conceptual modelling. *Applied Ontology*, 3(1-2):1–12.
- [Hatzi et al., 2015] Hatzi, O., Nikolaidou, M., Vrakas, D., Bassiliades, N., Anagnostopoulos, D., and Vlahavas, I. (2015). Semantically aware web service composition through ai planning. *International Journal on Artificial Intelligence Tools*, 24(01):1450015.
- [Hinds et al., 2004] Hinds, P. J., Roberts, T. L., and Jones, H. (2004). Whose job is it anyway? a study of human-robot interaction in a collaborative task. *Human-Computer Interaction*, 19(1):151–181.
- [Jennings et al., 1998] Jennings, N. R., Sycara, K., and Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous agents and multi-agent systems*, 1(1):7–38.
- [Jonker et al., 2011] Jonker, C. M., van Riemsdijk, M. B., and Vermeulen, B. (2011). *Shared Mental Models*, pages 132–151. Springer Berlin Heidelberg, Berlin, Heidelberg.

- [Jung et al., 2017] Jung, M. F., Šabanović, S., Eyssel, F., and Fraune, M. (2017). Robots in groups and teams. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 401–407. ACM.
- [Kato et al., 2014] Kato, J., Sakamoto, D., Igarashi, T., and Goto, M. (2014). Sharedo: to-do list interface for human-agent task sharing. In *Proceedings of the second international conference on Human-agent interaction*, pages 345–351. ACM.
- [Kazik, 2010] Kazik, O. (2010). Role-based approaches to development of multi-agent systems: A survey.
- [Kendall, 1999] Kendall, E. A. (1999). Role model designs and implementations with aspect-oriented programming. In *ACM SIGPLAN Notices*, volume 34, pages 353–369. ACM.
- [Khadka and Sapkota, 2010] Khadka, R. and Sapkota, B. (2010). An evaluation of dynamic web service composition approaches. In *ACT4SOC*, pages 67–79.
- [Kozłowski et al., 2015] Kozłowski, S., Grand, J. A., Baard, S. K., and Pearce, M. (2015). Teams, teamwork, and team effectiveness: Implications for human systems integration. *The handbook of human systems integration*, pages 535–552.
- [Kritsimalis, 2010] Kritsimalis, A. (2010). Role modeling for cdl specifications of web services. *staffordshire university*.
- [Kühn et al., 2015] Kühn, T., Böhme, S., Götz, S., and Aßmann, U. (2015). A combined formal model for relational context-dependent roles. In *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, pages 113–124. ACM.
- [Kumar et al., 2002] Kumar, S., Huber, M. J., Cohen, P. R., and McGee, D. R. (2002). Toward a formalism for conversation protocols using joint intention theory. *Computational Intelligence*, 18(2):174–228.
- [Lécué et al., 2008] Lécué, F., Silva, E., and Pires, L. F. (2008). A framework for dynamic web services composition. *Emerging Web Services Technology, Volume II*, pages 59–75.
- [Levesque et al., 1990] Levesque, H. J., Cohen, P. R., and Nunes, J. H. (1990). On acting together. In *AAAI*, volume 90, pages 94–99.
- [Ljungblad et al., 2012] Ljungblad, S., Kotrbova, J., Jacobsson, M., Cramer, H., and Niechwiadowicz, K. (2012). Hospital robot at work: something alien or an intelligent colleague? In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 177–186. ACM.
- [Margerison and McCann, 1984] Margerison, C. and McCann, D. (1984). Team mapping: a new approach to managerial leadership. *Journal of European Industrial Training*, 8(1):12–16.
- [Margerison and McCann, 1990] Margerison, C. J. and McCann, D. (1990). *Team management: Practical new approaches*. Mercury Books.
- [Martin et al., 2007] Martin, D., Burstein, M., Mcdermott, D., Mcilraith, S., Paolucci, M., Sycara, K., McGuinness, D. L., Sirin, E., and Srinivasan, N. (2007). Bringing semantics to web services with owl-s. *World Wide Web*, 10(3):243–277.

- [Mathieu et al., 2008] Mathieu, J., Maynard, M. T., Rapp, T., and Gilson, L. (2008). Team Effectiveness 1997-2007: A Review of Recent Advancements and a Glimpse Into the Future. *Journal of Management*, 34(3):410–476.
- [Mathieu et al., 2015] Mathieu, J. E., Tannenbaum, S. I., Kukenberger, M. R., Donsbach, J. S., and Alliger, G. M. (2015). Team Role Experience and Orientation : A Measure and Tests of Construct Validity. *Group & Or*(January 2014):6–34.
- [McCann and Margerison, 1989] McCann, D. and Margerison, C. (1989). Managing high-performance teams. *Training and development journal*, 43(11):46–53.
- [McIlraith and Son, 2002] McIlraith, S. and Son, T. C. (2002). Adapting golog for composition of semantic web services. *KR*, 2:482–493.
- [Milind Tambe, 1997] Milind Tambe (1997). Towards Flexible Teamwork.
- [Nacer and Aissani, 2014] Nacer, H. and Aissani, D. (2014). Semantic web services: Standards, applications, challenges and solutions. *Journal of Network and Computer Applications*, 44:134–151.
- [Nikiforova et al., 2011] Nikiforova, O., Sejans, J., and Cernickins, A. (2011). Role of uml class diagram in object-oriented software development. *Scientific Journal of Riga Technical University. Computer Sciences*, 44(1):65–74.
- [Parker, 2008] Parker, G. M. (2008). *Team Players and Teamwork*.
- [Patel and Shah, 2016] Patel, S. and Shah, T. (2016). A survey on issues and challenges of web service development, composition, discovery. *Journal of Science and Technology (ISSN: 0975-5446)*, VNSGU, 5(1).
- [Pollack, 1990] Pollack, M. E. (1990). Plans as complex mental attitudes. *Intentions in communication*, pages 77–103.
- [Rao et al., 2004] Rao, J., Kungas, P., and Matskin, M. (2004). Logic-based web services composition: From service description to process model. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 446–453. IEEE.
- [Reenskaug et al., 1996] Reenskaug, T., Wold, P., Lehne, O. A., et al. (1996). *Working with objects: the OOram software engineering method*. Manning Greenwich.
- [Riehle and Gross, 1998] Riehle, D. and Gross, T. (1998). Role model based framework design and integration. In *ACM SIGPLAN Notices*, pages 117–133. ACM.
- [Riemer et al., 2011] Riemer, K., Holler, J., and Indulska, M. (2011). Collaborative process modelling-tool analysis and design implications. In *ECIS*, page 39.
- [Rousseau, 2006] Rousseau, V. (2006). Teamwork Behaviors: A Review and an Integration of Frameworks. *Small Group Research*, 37(5):540–570.
- [Rumbaugh et al., 2004] Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *Unified modeling language reference manual, the*. Pearson Higher Education.
- [Russell and Norvig, 2010] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River NJ, 2010. Third edition.
- [Salas et al., 2008] Salas, E., Cooke, N. J., and Rosen, M. A. (2008). On teams, teamwork, and team performance: Discoveries and developments. *Human factors*, 50(3):540–547.

- [Salas et al., 2005] Salas, E., Sims, D. E., and Burke, C. S. (2005). Is there a “big five” in teamwork? *Small group research*, 36(5):555–599.
- [Sauppé and Mutlu, 2015] Sauppé, A. and Mutlu, B. (2015). The social impact of a robot co-worker in industrial settings. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 3613–3622. ACM.
- [Schwartz et al., 2016] Schwartz, T., Feld, M., Burckert, C., Dimitrov, S., Folz, J., Hutter, D., Hevesi, P., Kiefer, B., Krieger, H. U., Luth, C., Mronga, D., Pirkel, G., Rofer, T., Spieldenner, T., Wirkus, M., Zinnikus, I., and Straube, S. (2016). Hybrid teams of humans, robots, and virtual agents in a production setting. *Proceedings - 12th International Conference on Intelligent Environments, IE 2016*, pages 234–237.
- [Sheng et al., 2014] Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. (2014). Web services composition: A decade’s overview. *Information Sciences*, 280:218–238.
- [Steimann, 2000] Steimann, F. (2000). A radical revision of uml’s role concept. In *International Conference on the Unified Modeling Language*, pages 194–209. Springer.
- [Stone et al., 2010] Stone, P., Kaminka, G. A., Kraus, S., Rosenschein, J. S., et al. (2010). Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI*.
- [Sycara and Sukthankar, 2006] Sycara, K. and Sukthankar, G. (2006). Literature review of teamwork models. *Robotics Institute, Carnegie Mellon University*, 31.
- [Takayama and Go, 2012] Takayama, L. and Go, J. (2012). Mixing metaphors in mobile remote presence. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 495–504. ACM.
- [Tannenbaum et al., 1992] Tannenbaum, S. I., Beard, R. L., and Salas, E. (1992). Team building and its influence on team effectiveness: An examination of conceptual and empirical developments. *Advances in psychology*, 82:117–153.
- [Tsiakmaki et al., 2011] Tsiakmaki, M., Hartonas, C., and Gana, E. (2011). Adaptivity as an emerged property of the composition of learning services in the semantic web. In *Proceedings of the 2nd Panhellenic conference on the use of Information systems on Educational process*.
- [Tsoutsas et al., 2019] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2019). Teamwork behavior: A review to interconnect industry 4.0 entities. In *Technological Developments in Industry 4.0 for Business Applications*, pages 1–25. IGI Global.
- [Van Wissen et al., 2012] Van Wissen, A., Gal, Y., Kamphorst, B., and Dignum, M. (2012). Human-agent teamwork in dynamic environments. *Computers in Human Behavior*, 28(1):23–33.
- [W3C, 2002] W3C (2002). Web services architecture requirements. Technical report, <https://www.w3.org/TR/ws-arch/>.
- [Yen et al., 2006] Yen, J., Fan, X., Sun, S., Hanratty, T., and Dumer, J. (2006). Agents with shared mental models for enhancing team decision makings. *Decision Support Systems*, 41(3):634–653.
- [You and Robert, 2017] You, S. and Robert, L. (2017). A working framework for human-robot teamwork.

[Zhu and Zhou, 2008] Zhu, H. and Zhou, M. (2008). Roles in information systems: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):377–396.

Είμαστε αυτό που επαναλαμβανόμενα πράτ-
τουμε.

Αριστοτέλης [384-322 π.Χ.]
Έλληνας φιλόσοφος

We are what we repeatedly do.

Aristotle [384-322 b.C.]
Greek philosopher

Chapter 3

Modeling Services with Role Models

Contents

3.1 Focusing on the Problem - Introduction	61
3.2 Role Modeling for Service Domains	63
3.3 Graphical Notation	65
3.4 Methodology for Establishing a Role Domain	68
3.5 Role Models Formally	69
3.5.1 The Upper Ontology of Role Models	71
3.5.2 State Changes of a Role Domain	72
3.5.3 Mathematical Definition of the RM and its Components	73
3.5.4 Web Service and the Composition Process	75
3.5.5 Summarize the Notation	76
3.5.6 Running Example	77
3.6 The Standard Algorithm for the Composition in $\varrho\Delta$	78
3.7 The Expressive Power of Role Models	81

3.1 Focusing on the Problem - Introduction

Our main focus in this research is modeling semantic Web service composition and in this chapter we lay down the foundations to model the problem in a domain such as it is ensured automatic composition that integrates cloud and Web services which their

collaboration is defined by their behaviors.

A Web service can be described as a reusable software component which can be invoked over the network. Web services can be published using WSDL which is a standard containing all details on how to technically invoke the service in a machine processable format. By default, as it is already described in the previous chapter, standard description languages like WSDL does not include semantics for the description of Web services. Different services could have similar WSDL descriptions while the result of their execution could have been different. Such ambiguities can be resolved by giving semantic meaning to the invocation of a service, which is essential for the automation of service composition. To address the challenges of heterogeneity, autonomy and concealment of service interaction details we propose an architecture for the domain supporting roles. The notion of the role is a concept closely bound up with interaction among entities. In our approach, services are organized in domains which serve specific goals and are modeled with role modeling. Role modeling is used in software engineering, and its main objective is to identify reoccurring interaction behaviors within a structure. We will provide a specification for modeling composition of services with role modeling inspired from the way role modeling is used in other sections but with different usage. Generally speaking, we notice that well-developed domains of human activity also feature standard practices in actions and interactions to deliver predictable and reliable results. For example, accountants complete audits by following a step-by-step review of balance accounts, audited journals and supporting documentation. Similarly, astronauts land aircrafts safely by following carefully documented check-lists. The idea is that a domain with habitants web services could be designed by using Role Models in a sense that it will not treat the interactions between web services but between the roles they play. Given this initial motivation, we proceed in this chapter to the presentation of a model that will meet these requirements. We first clarify why, on the contrary to the work is done, we choose to model service composition by using a different technique, that of role modeling. The reason for distinguishing the service specification from already defined languages in our model is that this model is implemented for different purposes in order to describe web services. The motivation is that we want to make this approach applicable to a domain and enhance accessibility to groups of services in the web instead of a specific one, which are similar in the sense that they have the same functionality of what they can implement in terms of the domain. In reality, although there are thousands of services in the web available, it would not be applicable to consider the functionality of each one separately in order to solve a composition problem. A main difficulty is that, when all distinct services are counted on for finding a plan for the composition, the problem becomes complicated due to the scale it is applied. The solution to this problem, if it is found, becomes impractical and

error-prone when a web service repository involves a large number of services. Therefore, we model services as groups since they have the same functionality in order to cope with the myriad of available web services living in the web, exploiting the abilities that the behavioral description of the method provides.

Another reason is that with this methodology we aim among other objectives to create a model that would also act as a mediator for filling in the interoperability gap between different service providers which serve the same users need encapsulated in different services. Finally, by using this approach we aim to exploit the behavioral description of role models to add a teamwork behavior to services. This behavior of services will help them to manipulate the uncertainties by capturing emergent acting during the process of composition.

In the rest of this chapter, we will focus on details of describing services by roles and behaviors presenting their functionality features and their ability to compose them, while in Chapter 5 we exploit this description of services using role and behaviors for assigning to them teamwork behavior.

3.2 Role Modeling for Service Domains

In our approach for service composition, services are organized in domains such as information domains, market domains, industry domains, e.t.c.. A similar approach we see in [Seremeti et al., 2009] where researchers introduce the notion of “*activity sphere*” in order to better organize the way everyday activities are carried out within an Ambient Intelligence environment. In the notion of domains we propose, services should be defined in terms of their roles and their behaviors. These domains, in general, provide goods and can achieve through services a range of abstract goals. Implicitly, the goals express the functionality of the domain, i.e. what a domain can achieve. For example, a domain can provide info or produce and sell cars. Apparently, this could form the profile of the domain by (i) using concepts related to user requirements with regard to the functionality requested by the domain and (ii) specifying the key objectives for users of a domain. The profiles, that contain abstract goals, can be described-organized by using an ontology and they are published to common registries being available for users or other domains, to retrieve them in order to find the appropriate services to use. Users of the domain could be employees, clients, providers, consumers or services and other cyber physical systems in general. With the proposed architecture, the service discovery and composition problem is simplified for the users, since a user does not look for specific services; instead, the user is interested in specifying a goal that would be achieved by using atomic and composite services that are enrolled in the domain. Focusing on what is to be achieved rather than

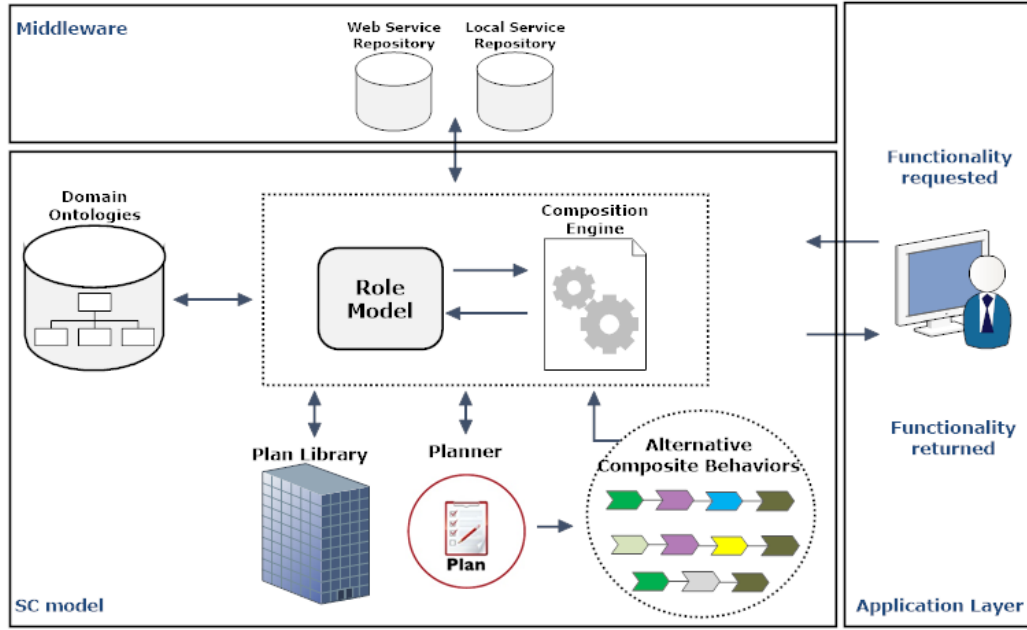


Figure 3.1: Graphical architectural elements of Web service composition by using Role Models. Behaviors that may contribute to accomplish the overall goal need to be discovered.

how activities are to be carried out in the domain, there are pre-built scenarios and use cases that lead to domain goals. The system stores the newly composed services and this is the experience gained for future execution of them. A graphical architecture presenting domains that are constructed with role models is shown in Figure 3.1.

In this architecture we propose (i) a middleware layer that is hosting the repository of local and web services, (ii) the service composition model as a backbone of the whole architecture and (iii) a generic application layer where it is described how applications specify their needed functionalities by sending requests to the service composition model usually by means of semantics and ontologies.

The middleware layer is hosting the repository of local and web services and is responsible for their execution and interaction. It could range from a simple network to a more sophisticated system that handles many important functionalities for distributed applications, providing support for distribution and mobility. In turn, the service composition model is standing between the middleware and the application layer. It is an abstract layer that does not consider a particular service technology, language or platform used for the composition process. The components of the service composition model are (i) the domain ontologies, (ii) the role model (RM) of the domain, (iii) a plan library, (iv) a planner and (v) the composition engine. As depicted in Figure 3.1, the SC model interacts with the application layer by receiving functionality requests from users or applications. SC model needs to respond to the functionality requests by providing services that fulfill the

demand. These services can be atomic or composite. The Service Repository represents all the local and distributed service repository where services are registered. The composition engine is aware about services in the Service Repository and, according to the behaviors they play through their roles, it chooses the appropriate services to compose. It either selects a predefined plan from the library or asks from the planner to develop a new one, in case no predefined plan fulfills the user demand. The planner returns possible plans in terms of composite behaviors which can be executed by services and this might provide more potential solutions. The SC model interacts with the Service Repository to choose services to compose.

In order to construct the domain and starting from scratch, a designer identifies it by developing goal statements which give the reason the domain exists. Once the goals have been captured and structured, he/she then analyses existing needs and solutions and builds scenarios (use cases) that they should be partitioned to roles of the domain. A main reason for developers to design domains with the role modeling approach is that this model gives better understanding of communication among participants by identifying and explaining their key characteristics while it gives the opportunity to document the constraints as well as to make them clear to users. For more effective analysis, he/she needs to emphasize on what the domain is trying to achieve. By modeling, should answer the following questions:

- What is the scope of the domain, what does it do?
- What kind of processes, users and resources appear in this domain?
- What goals, responsibilities and tasks are required?
- How do the entities interact?

3.3 Graphical Notation

This section facilitates the graphical notation for role models by illustrating such a model for an example application. The notation for the Role models originates from the Unified Modeling Language (UML) class diagram notation. UML is an industrial standard for object-oriented software specification which offers a notation for class modeling during object oriented software development. The UML class diagram is considered as a “bridge” between software specification at the user side and software realization at the developer side. It requires to follow strong guidelines for identification of class objects from the problem domain and notational conventions for modeling of the class diagram for its further usage in system coding. During the analysis, while collecting information on the

problem domain, the class diagram is viewed from the conceptual perspective, thus it is called a “conceptual class diagram”.

In the design process class diagrams are supplemented with technical details, thus more precision and more notation is used. Classes are populated with attributes and methods, as well as different kinds of relations, which represent overall the system structure. Depending on the software development process, a UML class diagram may contain a sufficiently detailed design, while in an iterative process it can still be a general system structure.

Finally, in the implementation process, class diagrams can be used to generate system basic structure code. Given that the class diagram represents the structure of the system, the notation does not provide behavior of the method. A class diagram describes the static structure of classes in the system and relationships between those entities. This way, a class diagram represents the structure of the system, as the summarized essence, the base for system operation, but not the system dynamics itself. The key element in the class diagram is “class”. The other elements are different types of relationships between classes, such as aggregation, composition or dependency [Nikiforova et al., 2011].

The key difference between class diagrams and RM diagrams is that class diagrams describe the static relationships between classes, whereas the role models describe explicitly the dynamic interactions between roles, i.e. the collaboration could be achieved among all the existing roles in the domain. A detailed comparison of the two modeling methods is provided in [Halpin and Bloesch, 1999].

In a RM, each interaction depicts two roles that interact with each other to describe collaboration, which refers to an explanation of the interaction found to predefined *use cases* are defined. The purpose of these use cases is to describe a sequence of role collaborations in terms of its domain obligations and case-specific functionality that lead to predefined *domain goals*. So, each of the interactions shown in the collaboration diagram also appears in a corresponding use case which act as a plan library for role collaborations. In the library, the interactions of a set of roles are actually described which represent the means through these actors perform specific activities, such as accessing databases, or reading information from a user interface, e.t.c. Except from the specification and analysis of a domain, the Role models are obviously utilised as patterns of interaction. The patterns have been chosen after analysing existing use cases and grouping together those that address similar problems to exhibit similar behavior in order to act as best practices for implementing new ones.

Figure 3.2 depicts an example role model. Before providing any formal definition, it is crucial to classify the different kinds of concepts employed by the modeling approach. The main elements of a RM that are used are roles, behaviors and communicative actions as it is shown in Figure 3.2. The extended notation used in RM diagrams include a rounded

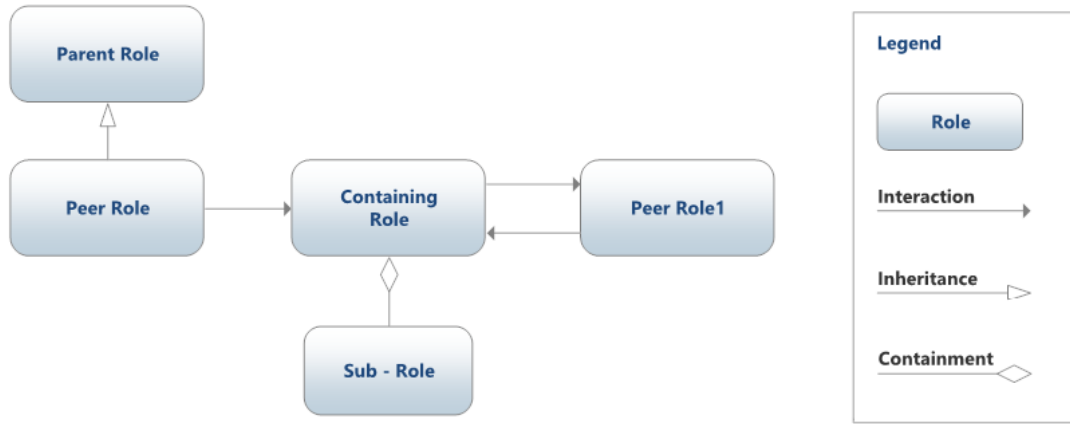


Figure 3.2: Elements that constitute the RM diagram of a domain.

box that represents a role and a filled arrow that depicts a collaboration path. The two roles that collaborate, i.e. a role and a peer one are connected by a filled arrow which additionally, indicates who the initiator is. Role specialization is indicated by an arrow with an unfilled triangle from the specialized role to the parent role. A role is said to be a container role when it has subroles and this is displayed as a solid line with a unfilled diamond at the end of the container role which is connected to one of each subrole it has.

The RM diagram of a domain could be identified as the intuitive model that depicts the way that services interact with other services. A service is regarded as a set of roles and its behavior is precisely that of the roles "it plays". A RM consists of roles and their collaboration and encloses all possible aspects of a given role collaboration with other roles exist in the domain. A role is a set of behaviors and describes the observable behavioral aspect of a participant of the domain. Actually, each behavior is a mean of communication from a role to another role. Potential local and distributed services that are interested in a specific domain can become participants of the domain and may act according to different roles. At any time, the roles of the domain are assigned to services; i.e. a service is a participant that plays the roles that are assigned to it while different services may play the same role. Consequently, the service cooperation in domains can be modeled as a role model, since this has the capability to depict the collaborations of services.

Definition 1 We call role domain ($\rho\Delta$) a domain that is designed by using the role modeling approach and accommodates services that are defined through the roles they play in the domain.

The business objectives and business logic of the domain are expressed with concepts of the domain ontology through constraints, goals and available plans. Since we focus on the

semantic web, a $\varrho\Delta$ is defined by using ontologies, considering information derive from the definition and execution of web services should be machine accessible and understandable using semantic web technologies. Consequently, the service composition process in $\varrho\Delta$ assembles a service always in accordance to the business logic of the domain, since it aims to achieve business objectives through the implementation of the combined services.

Given the description of services using roles, we make the following adaptation to the composition approach, where the web service composition problem is regarded as a web service composition planning problem. Instead of finding a plan that is a synthesis of actions as in traditional planning, we search for a plan that is constituted from a sequence of behaviors. We exploit the abstraction that has been achieved with role modeling and transform the problem of finding a composition of processes to the problem of finding the appropriate behaviors to be composed.

In the next section we present the methodology that is followed to construct a $\varrho\Delta$ and describe how it operates.

3.4 Methodology for Establishing a Role Domain

In this section we aim to present how to create a role domain that focuses on developing a methodology which could be implemented for mapping the different service processes to common behaviors been executed in a domain. The $\varrho\Delta$ also serves as a mediator between the different types of services hosted in the same domain and operates each time some business is needed. It uses the standard service description provided by each provider and transforms it into roles and behaviors in a form that it can serve the role modeling approach.

The proposed methodology includes the following nine steps:

1. Define the objectives of the role domain.
All domain goals identified in the domain are organized in an ontology.
2. From domain goals, derive the communicative actions and abstract domain behaviors that they are partitioned to roles of the domain.
3. Define preconditions and effects of the behaviors.
Once the role communication is defined, the preconditions and effects of the behaviors to be executed are stated.
4. Create the RM of the domain that depicts the roles and behaviors identified in previous steps.

5. Collect and analyse local and web services available to enroll in the domain.

For each service we need to determine the roles it plays. For this reason, we need to know other extra details or attributes and their meanings (semantics). Each attribute must be identified by its name, data type, constraints, and default values.

6. Develop the mapping.

The core function of the mapping process is to create a relation between each service and the roles to be mapped to. This is achieved by deciding how the functionality of services looks like, which attributes are included, which values are allowed and what are the constraints for each process to be executed. This is usually one of the most important and the most difficult step during the process of service enrolment.

7. Design use cases and alternative solutions via finding plans that correspond to domain goals.

Since the process of service enrolment to the semantic domain is completed, different solution designs that correspond to use cases could be created as a sequence of paths of behaviors that could be executed to accomplish given domain goals. All the solutions are stored as behavior paths in the plan library.

8. Implement a solution.

Make the service assignment for each behavior participates in the plan. Make the composite service formation in order to execute one of the previous solution designs that perform the service integration.

9. Test the solution.

Simulate the execution, evaluate the performance. This step is the most challenging. It requires an iterative effort and inspection to ensure the completeness and accuracy of the product service. If it is executed successfully, then store it to the inventory of composite services having the specific goal. Otherwise, try another set of service allocation if exists. In case it doesn't exist, try to find another plan with a similar goal.

A simple flowchart presenting this methodology is shown in Figure 3.3. In the next section we formally define the role domain.

3.5 Role Models Formally

We present a general framework where a domain is defined with the role modeling technique and services are characterized in terms of their behavioral description with respect to the domain usage. In other words, we clearly define for the role modeling

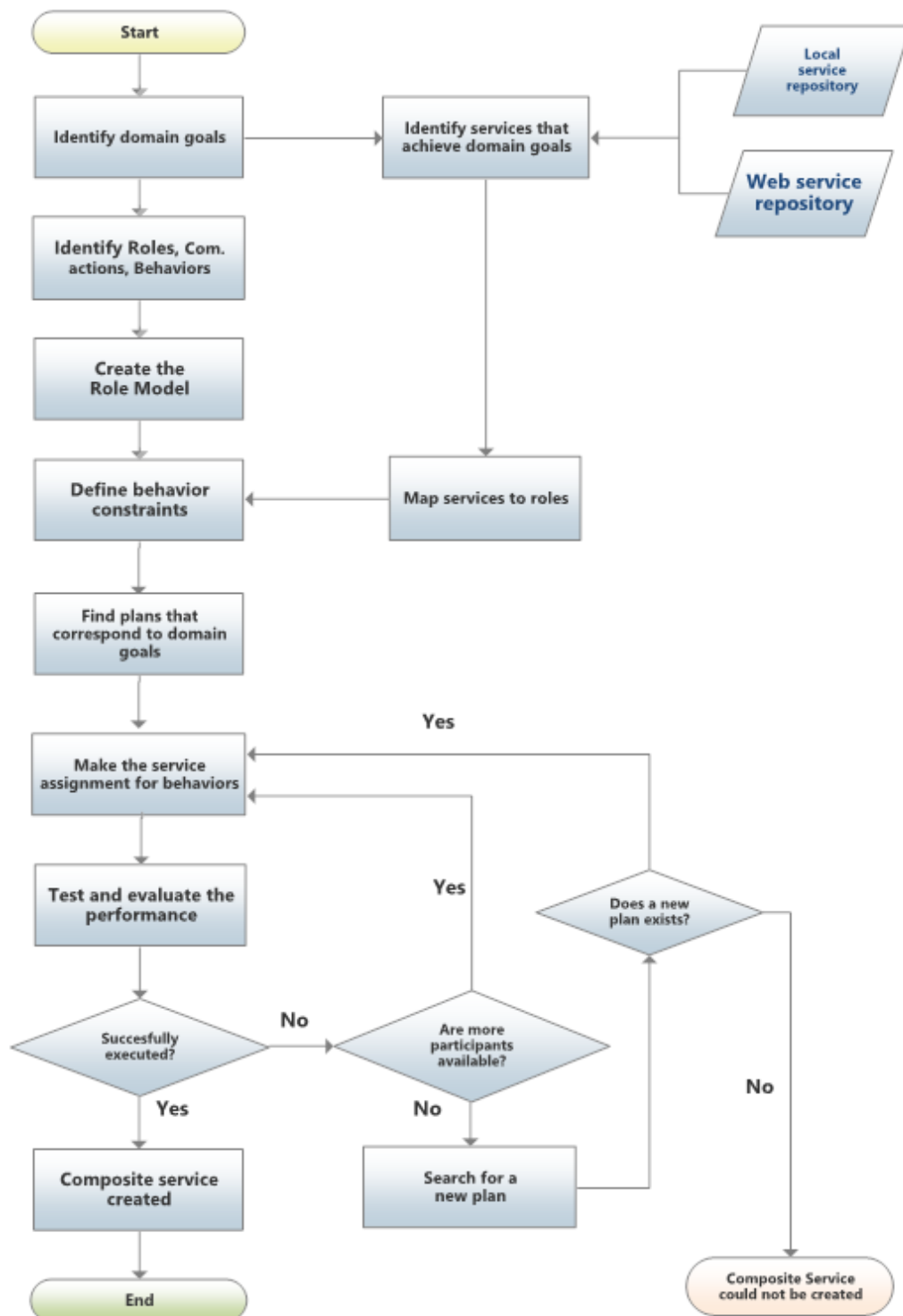


Figure 3.3: The methodology steps for establishing and utilising the Role Domain

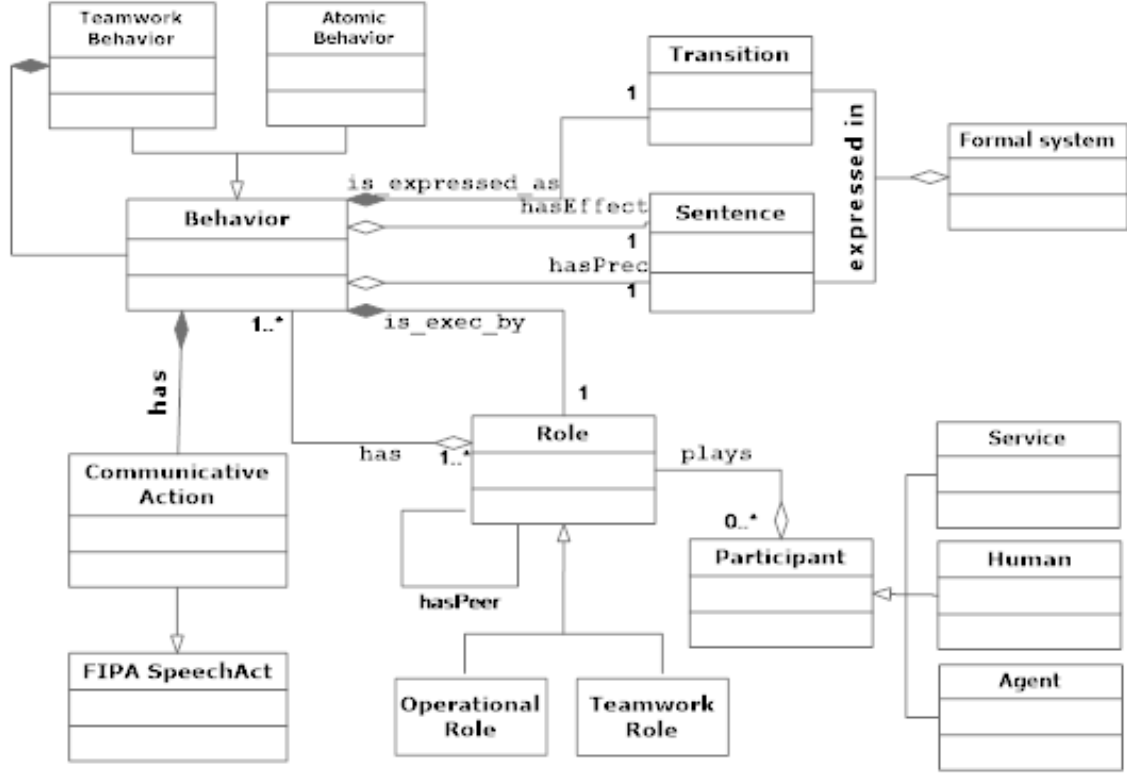


Figure 3.4: The UML class diagram for the RM ontology.

approach the semantics of services in terms of their behavioral description. Then, on top of this, we formally present the problem of service composition.

3.5.1 The Upper Ontology of Role Models

In order to present the modeling in a formal way, we have constructed the UML class diagram of the role modeling ontology for web services. In Figure 3.4 we present the RM specification by using class diagrams to depict how objects constituting the system are interrelated.

The main concepts used in the role modeling approach, as it is shown in UML class diagram in Figure 3.4, are *roles*, *communicative actions* and *behaviors*. In our approach, in order to model from an abstract notion each dimension that services exhibit within domains, we adopt the main modeling ideas, with dissimilar perspective to other research in role modeling, and give them the meaning presented in the definitions that follow.

A *communicative action* (*ca*) is a type of interaction between two roles and expresses, encoded, the reason for interaction. Example names could be: *acceptOrder*, *fail*, *inform*, *propose*, *queryIf*, *rejectInvoice*, *requestBalance*.

Communicative actions are inherited from FIPA-ACL¹ specification which is based on

¹Foundation for Intelligent Physical Agents, <http://www.fipa.org>

speech act theory [FIPA, 2008]. They consist of a set of message types and the description of their pragmatics, meaning the effects on the mental attitudes of the sender and receiver agents. They describe every communicative act with formal semantics based on modal logic (FIPA-SL, preconditions – rational effects). Communicative actions are viewed as a special class of actions that correspond to building blocks of dialogues between roles.

A *behavior* (b) expresses a potential collaboration between roles that could communicate using a communicative action ca . A role defines and describes a behavior while a participant addresses and implements this behavior.

The way that a role interacts with another peer role is considered as a behavior of the first role. A role expresses different aspects of collaboration to other roles through different behaviors. So, a role is considered as a set of the expected behaviors it exhibits, which states the interactions in which it is potentially involved. Although the RM diagram presents the possible interaction among roles through the execution of the appropriate behavior, we need, additionally, to state explicitly under which circumstances the role has the ability to execute a behavior. These are expressed as statements and constitute the constraints of the transition that happens in the domain when a behavior is executed.

3.5.2 State Changes of a Role Domain

There is a substantial body of research work [Rao et al., 2004, Berardi et al., 2005, Sheng et al., 2014, Zeshan and Mohamad, 2011] that contributes to the solution of the service composition problem, exploiting research efforts in Artificial Intelligence (AI) planning [Russell and Norvig, 2010]. The main idea behind these approaches is that the problem of automatic or semi-automatic composition of tasks can be faced as an AI planning problem where the service composition is seen as an issue of workflow design, where workflow tasks invoke services or service processes. Following this direction, and instead of modeling web services by functions or relations as other approaches do, we propose the composition method to be based on the possible interactions between services and their environment during the course of their invocation which are modeled as behaviors of the role that services play. The main reason is that according to role modeling, services that play a role and extract a behavior will change the state of their environment after each execution of a behavior.

AI planning involves finding a plan containing the series of transitions required to reach a goal state set by the requester. An AI planning problem, in the general case, can be described as a quintuple $(W, \varphi_0, \varphi_g, A, \gamma)$ where W is a set with all possible states of the domain, φ_0 is a formula describing the initial state of the domain, φ_g is a formula describing the goal state of the domain that the planning system attempts to reach, A is

a set of actions that contains formulas about the actions that the planner can perform in attempting to change one state to another one and γ is a subset of $W \times W \times A \times W$ and defines for each action to which state is applicable and which is the result state [Ghallab et al., 2004].

There is a number of ways to formalize the planning problem such as state-transition system models, State-space search algorithms, planning by SAT, etc. and solve it automatically. In our approach, each behavior, when executed, changes the world state. In our case, the state change produced by the execution of the service behavior is specified through the precondition and effect properties which could be expressed in an appropriate logic language, preferable one that supports state change. In this case, a behavior execution further denotes the state change of each role acting. Preconditions refer to the conditions that should be satisfied prior to the behavior being requested to be executed while effects are the result of the successful execution of a behavior. At any time, a behavior is executed, it means that preconditions hold and two roles communicate while after the execution the statement describing the effects holds. The behavior concept can be interpreted as a type of action in planning, so that the system continues to evolve from one state of the world towards another.

3.5.3 Mathematical Definition of the RM and its Components

In this section we mathematically define the RM components in order to proceed with the composition of behaviors.

Definition 2 *A role model (ρM) of a $\rho\Delta$ is defined as a triple*

$$\rho M = \langle R, CA, B \rangle$$

where

- R denotes the set of all roles appear in the RM of the domain.
- CA denotes a set of communicative actions and
- B denotes a set of behaviors in $\rho\Delta$.

Definition 3 *A behavior (b) is an ordered tuple*

$$b = \langle ca, r_{peerRole} \rangle$$

where

- $ca \in CA$ and

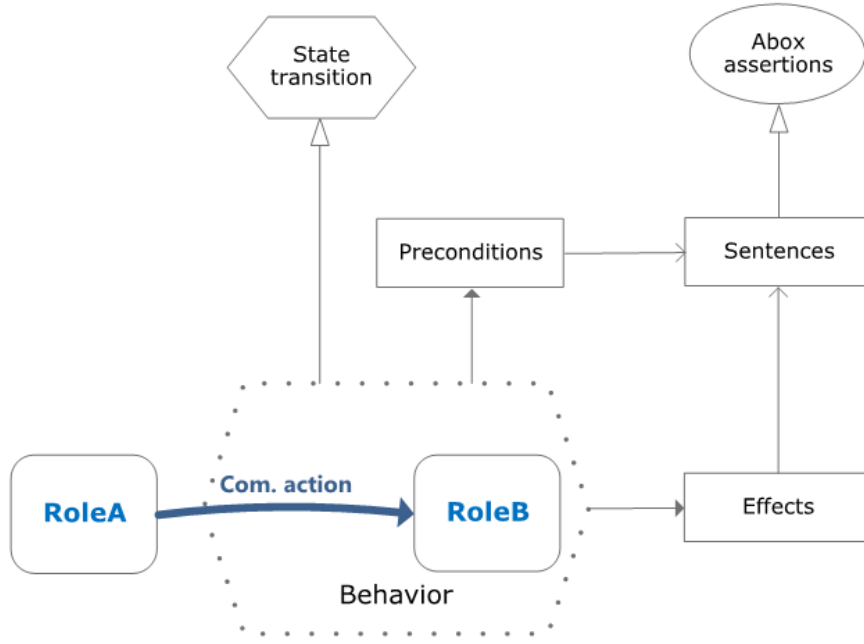


Figure 3.5: The behavior conceptualization.

- $r_{peerRole} \in R$.

A behavior indicates that a role containing the specific behavior has the ability to communicate with the role $r_{peerRole}$ with a *ca* intention for communication.

A behavior encapsulates the process of communicating a role to a peer role. The behavior belongs to the role that initiates the communication. For example, the behavior $b_1 = \langle invoice, r_{client} \rangle$ represents the communication between the role expresses the behavior b_1 with the peer role $RClient$ having the "invoice" intention.

So far, the discussion revolved around the structure and nature of roles and behaviors in role models and did not cover the constraints of their execution. Domain regulations require that some conditions are indispensable in order for a behavior to be executed. Furthermore, after the execution of a behavior in the result state the effects of the execution of the specific behavior are satisfied. A diagram presenting the behavior conceptualization is shown in Figure 3.5. *Preconditions* and *effects* are sentences in a logic language which coincide with the Abox assertions [Xiao et al., 2018] that derive from the domain ontology. These sentences could partially describe the evolution of the situations of the system. Precondition and effects of behaviors have the following semantics: In order for a behavior to be executable, the precondition must hold and, upon successful execution, the behavior will produce the effects.

Definition 4 *A role r of a ϱM is a set of behaviors*

$$r = \{ b_i, \dots, b_j \}$$

where $b_i, \dots, b_j \in B$.

The capital letter R represents the set of all roles appear in the RM of the domain.

3.5.4 Web Service and the Composition Process

The definition of a service is based on the semantics of roles it plays, rather than its structure. Through semantic annotation, each role within the $\varrho\Delta$ refers to a semantic individual in the domain content ontology. A semantic individual in turn is a member of an ontology class, linking the role element of the ϱM to a semantic concept. As a result, any role R_i within the context of the domain is mapped to a set of concepts of the domain ontology and any service s_i to a set of role concepts. The semantic information enables reasoning about service execution as a base for semantic matching between the resulted state after the execution of one service and the preconditions required for the execution of another.

Definition 5 *A Service $s \in S$ is the set of roles it plays in a $\varrho\Delta$:*

$$s = \{ r_i, \dots, r_j \}$$

where $r_i, \dots, r_j \in R$.

For example, service $s_1 = \{r_5, r_6\}$ plays the role r_5 and the role r_6 .

The focus of the service composition lies on the appropriate ordered set of behaviors to be executed rather than services, since services that play the roles containing the concerned behaviors could be invoked. Thus, according to service definition, the web service composition is concluded from the behavior composition.

Definition 6 *A goal statement is expressed through the domain ontology and describes achievements that the domain could accomplish.*

The requested goals are usually combinations of original purposed goals of the domain ontology, which are expected to express a result that a user of the domain want to achieve. For example, a goal is to achieve effect x and effect y.

An *objective plan* is a plan could be followed to achieve a goal specified within the domain. It is stored in the plan library of the domain and it is a behavioral pattern which

express reusable solutions to a commonly occurring problem within a given context in a domain.

A user request can be expressed (with different semantics), by determining the desired goal to be achieved, having an initial and a goal situation. The initial situation which partially is provided by the user expresses the parameters needed for the composition. The goal situation is a sentence that describes what a successful composition must provide. The solution, in order to become an objective plan, must be able to provide a plan where the goal situation is satisfied. We define formally a composition request q as follows:

Definition 7 *A composition request is a tuple which contains the preconditions and effects of the composite behavior we are looking for to be executed. The request*

$$q_x = \langle qx_{initial} , qx_{goal} \rangle$$

is a query for a composite behavior to be executed in a situation that $qx_{initial}$ is satisfied, and according to the solution we are looking for qx_{goal} is satisfied in the resulted situation.

For better manipulating composition requests descriptions we will use the helper functions $init(q_x) = qx_{initial}$ and $goal(q_x) = qx_{goal}$

We assume that the request description q_x is well formed, that is, neither has an empty set of initial nor an empty set as goal situation: $init(q_x), goal(q_x) \neq \emptyset$.

Definition 8 *A behavior composition (bc) is a n -tuple*

$$bc = \langle b_1, ..., b_n \rangle$$

where $b_1, ..., b_n \in B$.

The elements of the n -tuple are called the *component behaviors*. The turn that behaviors appear in the tuple defines the execution order of the behaviors.

In the next section we give the extra notation that is required in order to specify the whole composition process in terms of the role modeling approach.

3.5.5 Summarize the Notation

The following conventions refers to concepts and other components of the system used in the problem definition that have not previously defined.

Domain: $\varrho\Delta$ is used to express the Role domain.

Service: s_i or ser_i is used to express the identity of a service, e.g., s_1 would represent a service named service 1, while S represents the set of all services. When referring to

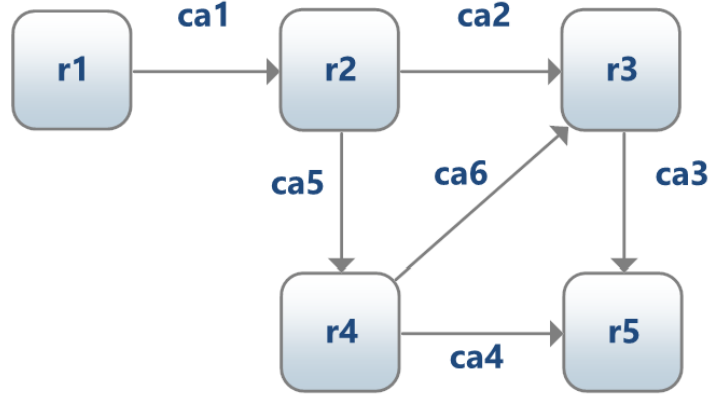


Figure 3.6: The RM of the role domain.

arbitrary services we use names such as i and j , and when we are referring to the number of services we use n . For example, when we use the term $\forall s_i \in S$ we are referring to all arbitrary services in the set of all services, and when we are using something that requires two arbitrary services, such as communication, we will say that arbitrary service s_i communicates to arbitrary service s_j . Similar convention is used with roles, behaviors and communicative actions.

Role: r_{Name} , additionally to r is used to express the identity of a role, e.g., r_{Client} would represent the role is played by a client of the domain, while R represents the set of all roles.

Communicative action: ca_i is used to express an intention from a role to communicate with another role, while CA represents the set of all communicative actions have been recorded in the domain.

3.5.6 Running Example

The running example is based on the formal definition of the challenge presented in the previous section and concerns a $\varrho\Delta$ consisting of a service repository S and the RM depicted in Figure 3.6. A composition request q will also be provided and its results.

The repository S contains eight services: $ser_1, ser_2, ser_3, ser_4, ser_5, ser_6, ser_7, ser_8$. Figure 3.7 displays the role each service plays. The annotation of service messages with role individuals from the domain ontology is represented by linking each service directly to the corresponding roles by linking each service to the roles it plays. For example, service ser_1 plays the role r_1 and r_3 .

In Figure 3.8 we depict for each atomic behavior follows from the RM the precondition and effect that derive from the domain constraints. There are six different behaviors

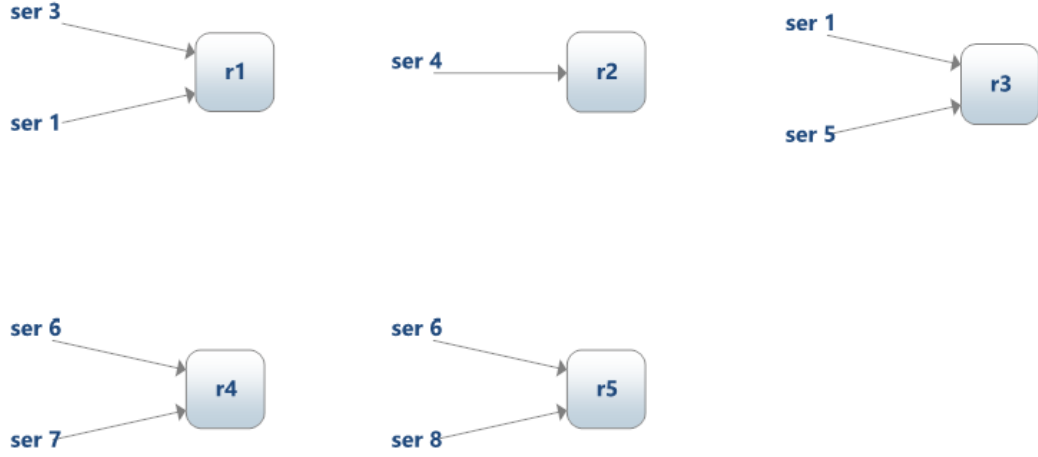


Figure 3.7: The service repository $S = \{ser_1, ser_2, ser_3, ser_4, ser_5, ser_6, ser_7, ser_8\}$.

executed by the five roles and each one has either one or more sentences to be satisfied in order to be executed, and similarly, after the execution it has one or more sentences that are fulfilled in the resulted situation.

In Figure 3.9, the composition request $q_x = \langle (pr_a, pr_b), (ef_x, ef_y) \rangle$ is displayed which is an x composite behavior having as precondition (two input sentences to hold) and as effect (two output sentences to hold). The request calls for a composition of behaviors that should return all paths that as an initial situation have one that all the preconditions stated are satisfied and as a goal situation, they have one that the all effect conditions stated are satisfied as well. The composite behavior b_x we are looking for is a tuple of the form $\langle b_k, b_m, \dots, b_n \rangle$ where all the elements of the tuple are the component behaviors to be executed.

The solution for the composite behavior, which is an objective plan for the domain, is depicted in Figure 3.10 where

$$s_1 \wedge s_2 \models pr_a \wedge pr_b \quad \text{and} \quad s_{10} \wedge s_{11} \models ef_x \wedge ef_y$$

3.6 The Standard Algorithm for the Composition in $\varrho\Delta$

We model the composition problem as a planning problem where, given an initial and a goal situation we are looking for a sequence of behaviors, that when executed starting from the initial situation will reach the goal situation. By using a modal logic, for example

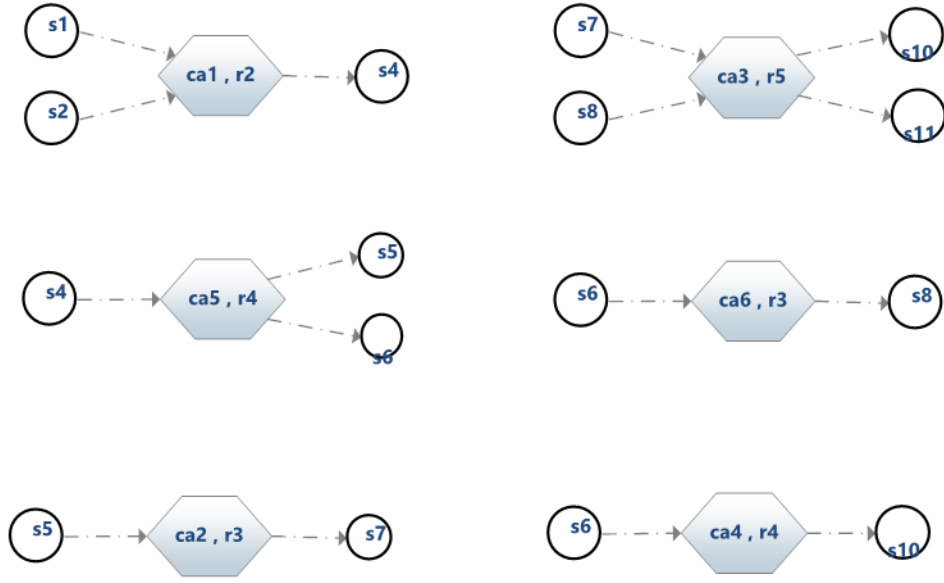


Figure 3.8: The *precondition* and *effect* of each atomic *behavior* are stated.

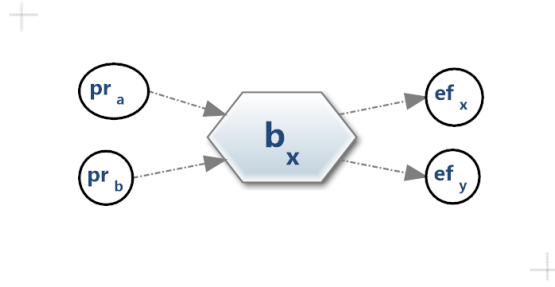


Figure 3.9: The composition request for a composite behavior b_x .

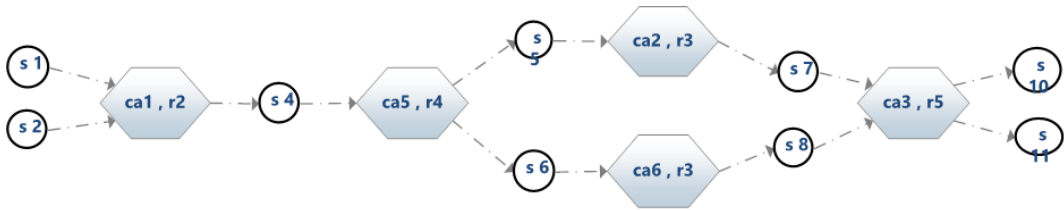


Figure 3.10: A solution for the composite behavior b_x .

the PDL logic language [Harel et al., 2001] for encoding the problem, a solution to this problem following the planning as satisfiability approach, according to [Ghallab et al., 2004], is a plan π which is a synthesis of actions such that:

$$A_x \rightarrow (\varphi_0 \rightarrow \langle \pi \rangle \varphi_g)$$

where A_x is the conjunction of a set of formulas, called extra logical axioms, and each action is described with its preconditions and effects. The intuitive meaning of the above formula is that, given the axioms that are described in A_x , under an initial situation partially described φ_0 , after the execution of the composite behavior π we reach a state by φ_g , i.e. that the goal of the composite service is achieved. The solution to the planning problem we are looking for, as this is described in the above logical sentence, could be generated from a theorem prover and it is actually the sequence of atomic behaviors that should be executed.

The *basic composition algorithm* that composes services from a repository, which are described as roles, is given in Algorithm 3.1. It takes as input the domain $\varrho\Delta$ and the request q_x having some precondition and effects which correspond to the description of initial and goal situation. It returns the set of services that can form the composite service along with the role each service play in the RM and the behavior they execute to achieve the goal according to the role modeling definition of services presented in this chapter.

The Algorithm 3.1 shows the practice that can be used to extract a plan for a given request. First, in line 2, the plan library of the role domain is assigned to a data structure p . p is then examined, with the *acquirePlan* method in line 3, to get from the plan library those plans that are consistent with the goal of the request q_x . These plans, which are the objective plans for the request q_x , are assigned to the data structure *plans*. In case there are no consistent plans already stored in the library (line 4), then we involve a planner by calling the method *findPlan*($\varrho\Delta, q_x$) to search for a new plan of behaviors (line 5). As services can evolve role collaboration patterns may change. Suppose consistent plans are found, either existing in the library or newly extracted by reasoning, thus the structure *plans* is not empty (line 7). Then a valuation is made to select the most appropriate plan, according to the domain rules (line 8) to derive the optimum plan, and the result is assigned to *optp* variable. The *optp* is the objective plan that is selected to be followed in order to achieve the goal of the given request. In lines 9 to 11 the behaviors are first extracted from the objective plan, then the roles which form the behaviors and finally the services that will participate in the composition. The set of involved services to the composition is returned, if there exists a plan (line 12) or null, otherwise (line 13).

The aforementioned methods are described with more details in Algorithms 3.2 and

Algorithm 1. Returns a service that corresponds to a query

Input : the domain $\varrho\Delta$, a request q_x for a service

Output : A set of services for the composition or null when no solution exists

```
1 begin
2    $p \leftarrow \varrho\Delta.planLibrary$ 
3    $plans \leftarrow acquirePlan(q_x, p)$ 
4   if  $plans = \emptyset$  then
5      $plans \leftarrow findPlan(\varrho\Delta, q_x)$ 
6   end if
7   if  $plans \neq \emptyset$  then
8      $optp \leftarrow evaluate(plans)$ 
9      $behav\_inv \leftarrow extractBehaviors(optp)$ 
10     $roles\_inv \leftarrow extractRoles(behav\_inv)$ 
11     $serv\_inv \leftarrow extractServ(roles\_inv)$ 
12    return  $serv\_inv$ 
13  else return null
14  end if
15 end
```

Algorithm 3.1.

3.3 in the following:

acquirePlan($q_x, planLibrary$): A data structure *plans* is initialized (line 2) in order to store all consistent and objectives plans that are consistent to the request. The domain library is searched (line 3) and all previously executed plans whose preconditions have as logical consequence those of the request and whose effect entails the effect of the request (line 5) are stored to the *plans* structure (line 5) as potential solution plans.

findPlan($\varrho\Delta, q_x$): Reason in order to find new plans for the composition of behaviors. Build a sentence using the conjunction of extralogical axioms derived from the domain constraints (line 3). Describe the initial situation by using the preconditions of the request (line 4); this is defined by this statements that should be valid before the behavior is executed. Furthermore, construct the effect situation by using the request effects (line 5) and utilise the behaviors (line 6) that act as state transition factors to formulate the sentence to be satisfied. Build the sentence that describe the planning problem (line 7), and call a theorem prover to check the satisfiability of the to find solutions and store the results to the *plans* structure (line 8). From the models that satisfy this sentence, we can extract the plans indicating the sequence of behaviors to be executed.

3.7 The Expressive Power of Role Models

There exist many modern approaches for composing web services. Considering the specific requirements of role modeling, the available services are similarly, to other approaches,

Algorithm 2. *acquirePlan(q_x, p)* Acquire existing plans from the domain library

Input : query q_x , the *planLibrary* of the domain

Output : A set of consistent library plans or $\emptyset$ when no plan exists

```

1  begin
2    plans  $\leftarrow \emptyset$ 
3    for each  $p \in \text{planLibrary}$ 
4      if  $p.\text{preconditions} \models q_x.\text{preconditions}$  and  $p.\text{effects} \models q_x.\text{effects}$ 
5        add  $p$  to plans
6      end if
7    return plans
8  end

```

Algorithm 3.2.

Algorithm 3. *findPlan($Q\Delta, q_x$)* Reason over new plans

Input : query q_x , the domain $Q\Delta_{\text{new}}$

Output : A set of consistent plans or null when no solution exists

```

1  begin
2    plans  $\leftarrow \emptyset$ 
3     $A_x \leftarrow \text{conjunctionOf}(\text{domain extralogical axioms})$ 
4     $\varphi_{\text{init}} \leftarrow q_x.\text{preconditions}$ 
5     $\varphi_{\text{goal}} \leftarrow q_x.\text{effects}$ 
6     $b \leftarrow \text{union\_of\_domain\_behaviors}$ 
7     $\psi \leftarrow A_x \mid - \varphi_{\text{init}} \rightarrow < b > \varphi_{\text{goal}}$ 
8    plans  $\leftarrow \text{checkSatisfiability}(\psi)$ 
9  return plans
10 end

```

Algorithm 3.3.

modeled as black boxes, by which the services are viewed in terms of their observable behavior while the inside of the system is hidden from being viewed. In our case, we have introduced the method for composition of web services by role modeling. Due to the vast diversity in the available web services, from the client's point of view, it has become difficult to find and decide what services should be used for the composition and decide the basis for their selection. We use role models as mediators between the Web service languages and a formal description that will enable the reasoning for composition purposes. By exploiting the role models we create reusable valid sequences of role behaviors while this make possible to infer new paths of collaboration of existing behaviors. The pivotal problems of Web services composition are how to model data dependency of candidate and how to satisfy efficiently a service request by composition. Since a Web service is a set of roles, which in turns is a set of behaviors, the transition is made by the execution of a behavior that changes the state of the environment. In a sense, the modeling of service composition concerns the modeling of the changes made by their behavior execution, from the point of view of domain changes.

The most distinguishing characteristic of the proposed model, up to this stage, lies in its grouping ability to deal with services compared to other AI planning based WSC methods. From the perspective of applicability, the grouping ability is crucial to a WSC problem. It determines whether a WSC method can respond to a composition request within a reasonable time frame.

This chapter depicts how services and their composition could be expressed through role modeling. Our interests involve extending the standard way of composing services by using a plan that is extracted from the RM in order to develop *teamwork enabled composed services* that are constituted from local and distributed services. In next chapters, we will present the teamwork roles that will support the composition synthesis. We exploit them to extend the composition approach and specify a new teamwork oriented role modeling one.

References

- [Berardi et al., 2005] Berardi, D., Calvanese, D., De Giacomo, G., Hull, R., and Mecella, M. (2005). Automatic composition of transition-based semantic web services with messaging. In *Proceedings of the 31st international conference on Very large data bases*, pages 613–624. VLDB Endowment.

- [FIPA, 2008] FIPA, T. (2008). Fipa communicative act library specification. *Foundation for Intelligent Physical Agents*, <http://www.fipa.org/specs/fipa00037/SC00037J.html> (30.6. 2004).
- [Ghallab et al., 2004] Ghallab, M., Nau, D., and Traverso, P. (2004). *Automated Planning: theory and practice*. Elsevier.
- [Halpin and Bloesch, 1999] Halpin, T. and Bloesch, A. (1999). Data modeling in uml and orm: a comparison. *Journal of Database Management (JDM)*, 10(4):4–13.
- [Harel et al., 2001] Harel, D., Kozen, D., and Tiuryn, J. (2001). Dynamic logic. In *Handbook of philosophical logic*, pages 99–217. Springer.
- [Nikiforova et al., 2011] Nikiforova, O., Sejans, J., and Cernickins, A. (2011). Role of uml class diagram in object-oriented software development. *Scientific Journal of Riga Technical University. Computer Sciences*, 44(1):65–74.
- [Rao et al., 2004] Rao, J., Kungas, P., and Matskin, M. (2004). Logic-based web services composition: From service description to process model. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 446–453. IEEE.
- [Russell and Norvig, 2010] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River NJ, 2010. Third edition.
- [Seremeti et al., 2009] Seremeti, L., Goumopoulos, C., and Kameas, A. (2009). Ontology-based modeling of dynamic ubiquitous computing applications as evolving activity spheres. *Pervasive and Mobile Computing*, 5(5):574–591.
- [Sheng et al., 2014] Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. (2014). Web services composition: A decade’s overview. *Information Sciences*, 280:218–238.
- [Xiao et al., 2018] Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., and Zakharyashev, M. (2018). Ontology-based data access: A survey. *IJCAI*.
- [Zeshan and Mohamad, 2011] Zeshan, F. and Mohamad, R. (2011). Semantic web service composition approaches: overview and limitations. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 1(3):640–651.

Αν θες να πας γρήγορα, πήγανε μόνος.
 Αν θες να πας μακριά, πήγανε με άλλους.
 Αφρικανική Παροιμία.
 If you want to go fast, go alone.
 If you want to go far, go together.
 African Proverb.

Chapter 4

Teamwork Behavior

Contents

4.1 Agent Based Teamwork Modeling	86
4.1.1 Identification of Agent Based Teamwork Factors	86
4.1.2 Clustering of the Teamwork Factors	90
4.2 Enhancing Teamwork Behavior of Services	95
4.2.1 Bridge Teamwork Factors to Service Teamwork Roles	96
4.2.2 Dynamics of Teamwork Roles during Service Composition	97
4.2.3 Using Design Patterns for Creating the Teamwork Roles	98
4.2.4 The Developer role.	99
4.2.5 The Organiser role.	102
4.2.6 The Monitor role	104
4.2.7 The Advisor role	107
4.2.8 The Innovator role.	110
4.2.9 The Promoter role.	112
4.2.10 The Producer role.	114
4.2.11 The Maintainer role.	116
4.3 Teamwork oriented Composition in Role Domains	117
4.3.1 Description of the Composition using Teamwork Roles	117
4.3.2 Integrated SD for the Composition with Teamwork Roles	119
4.4 Conclusions	121

The core issue of the investigation in this chapter is to benefit from teamwork theory in order to effectively compose a set of services. In the previous chapter, we described how services could be modeled using the role modeling approach, and following this

direction, we define the teamwork roles to be used within the RM that will improve services cooperation.

4.1 Agent Based Teamwork Modeling

We review the key factors that affect teamwork behavior in agent based teamwork models and indicate how they could be related to human teamwork and benefit the service composition. A review of related studies was conducted, and, as a result, a range of factors that affect teamwork behavior were identified and analyzed. From the analysis, factors that traditionally gain attention to the majority of studies stand out, while new-appeared factors are determined from recent studies about models gearing towards dynamic and realistic environments that point to new aspects of teamwork behavior. To highlight for our model particular characteristics of teamwork, we adopted those could pair to service behavior. More explicitly, after identifying a range of factors that affect team working, we contribute by determining *eight dominant teamwork roles* that prevail during service group interaction and provide their primary teamwork behavior. For these roles, except from pairing them with teamwork characteristics are used to agents and could be applicable to services, we also look for new aspects of behavior concerning teamwork to indicate their primary teamwork behavior and complete their definition. We are doing so by (i) providing a classification of teamwork factors recognized in the review, (ii) integrate and adapt the reviewed roles in the literature to eight service teamwork roles (iii) habilitate the novel service teamwork roles with indicative behavior that encapsulate the factors which largely affect the effectiveness of service composition.

4.1.1 Identification of Agent Based Teamwork Factors

In [Tsoutsas et al., 2019] we presented a systematic search of the literature on teamwork, focusing on identifying principles that influence teamwork. Rather than attempting to provide a comprehensive review of work that has been done over the past years, in this research our orientation was to highlight particular trends and factors in the team research arena that could be used in service teamwork.

A selective list of key factors was derived from the afformentioned review, i.e. the team attributes each model focuses on to support teamwork. These factors give an indication of how members are inter-related in the various proposed models and how the expected team working behavior of systems is depicted, where agent members are collaborating similar to human collaboration. The fifteen team factors revealed are discussed in relative detail and their implication during their activity is examined in Chapter 6 with a case study in order to gain further insight into their team role structure and performance.

Table 4.1: Brief description for each teamwork factor and the studies it appears (1/3).

AA	Teamwork factor	Description	
1	Ad hoc team setting (ADH)	A team member's behavior that is able to cooperate with multiple potential unknown and heterogeneous members without precoordination in order to obtain a common goal and reach an optimal team behavior.	[Agmon and Stone, 2011], [Genter et al., 2011], [Stone et al., 2010], [Barrett et al., 2011]
2	Autonomy (AUT)	The ability of the member to assist the team by having a gree of reasoning and taking a decision.	[Bradshaw et al., 2004]; [Bradshaw et al., 2008]; [Harbers et al., 2012]; [Schwartz et al., 2016]; [Wu et al., 2011], [Kaminka and Frenkel, 2005]
3	Collaboration (COL)	The establishment of the ability to act collaboratively with mutual awareness among team members that work together performing a task or carrying out the team activities needed to satisfy a shared goal.	[Barange et al., 2014], [Leng et al., 2006], [Dunin-Keplicz and Verbrugge, 2011], [Bradshaw et al., 2004] [Skubch et al., 2010]
4	Communication (COM)	Through communication, members provide to other members the essential information about working together to achieve shared objectives.	[Barange et al., 2014], [Grant et al., 2005], [Schwartz et al., 2016], [Hanna and Richards, 2013] [Dunin-Keplicz and Verbrugge, 2011]
5	Coordination (COO)	Through coordination the group face the interdependence problem among team members when they jointly perform a task.	[Barange et al., 2014]; [Bradshaw et al., 2008]; [Harbers et al., 2012], [Leng et al., 2006] [Kaminka and Frenkel, 2005] [Skubch et al., 2010]

Table 4.2: Brief description for each teamwork factor and the studies it appears (2/3.)

AA	Teamwork factor	Description	
6	Delegation (DLG)	The ability of a member to choose the tasks to delegate to other team members in order to increase team performance when unavailability to perform a task occurs.	[Schwartz et al., 2016]
7	Evaluation (EVL)	Team members use evaluation methodologies and metrics to determine the performance and the team effectiveness and in what degree they achieve the intended effects.	[Stone et al., 2010], [Yen et al., 2006]
8	Joint-intention (INT)	The common intentions, the commitment and focus members demonstrate in order to handle unexpected failures and reach the group goal.	[Barange et al., 2014]; [Grant et al., 2005] [Leng et al., 2006] [Skubch et al., 2010] [Jonker et al., 2011]
9	Knowledge of team capabilities(KCT)	In order to improve collaboration, members learn about capabilities of their teammates, model them and use that knowledge to improve team performance.	[Kennedy et al., 2008]
10	Knowledge sharing (KNL)	The shared understanding of the task to be performed, common knowledge about shared goals, intentions and plans, that the involved team members have which help them to reason and predict future system states.	[Barange et al., 2014] [Skubch et al., 2010]
11	Learning (LRN)	The ability of members to "learn" as they identify new patterns in data that will enable participants to effectively discover opportunities and create strategies to improve teamwork experience.	[Barrett et al., 2011], [Kyrarini et al., 2016] [Liemhetcharat and Veloso, 2013]

Table 4.2: Brief description for each teamwork factor and the studies it appears (3/3).

AA	Teamwork factor	Description	
12	Planning (PLN)	The ability of some members to plan, recognize and resolve conflicts to scheduled plans.	[Bradshaw et al., 2008], [Grant et al., 2005] [Schwartz et al., 2016], [Korsah et al., 2012] [Fan et al., 2006], [Skubch et al., 2010]
13	Decision making (DM)	Decision making process and problem solving is essential for effective team performance, thus reasoning agents exist in teams are responsible for organizing the system's decision making process.	[Fan et al., 2006], [Chen and Barnes, 2013] [Korsah et al., 2012]
14	Teamwork policies (POL)	Policies assuring the appropriate functionality of members according to domain and other safety rules.	[Bradshaw et al., 2004]
15	Trust (TRS)	All team members take into account the issue of trust and the factors that help to build trusted relationships that encompass both the social and ethical aspects in order to adjust their behavior.	[Bradshaw et al., 2008], [Van Wissen et al., 2012], [Chen and Barnes, 2013] [Harbers et al., 2012]

In Table 4.1, a brief description is presented for each teamwork factor recognized in the review based to the corresponding studies the factor appears.

4.1.2 Clustering of the Teamwork Factors

The team factors that merit particular attention in the literature are presented in three categories; these specific factors regard teamwork behavioral attitude with reference to models concerning agent based systems. The aim is to draw from the results a roadmap on using teamwork theory in making synergies among human-agent-robots-services and other intelligent system members, forming heterogeneous teams capable of effective teamwork in dynamic and realistic environments. Generally, factors address different collaboration attributes, such as (i) how the team is organizing itself, e.g. by creating rules for communication and priority execution, (ii) how the team is forming its strategy for future direction, e.g. by planning and decision making and (iii) how the team works together to achieve synergy, e.g. rules of trust and engagement.

A. Organizing factors

The factors belonging in this category promote the organization of the group.

1. Collaboration (COL): The establishment of mutual awareness and common grounding among team members that work together performing a task or carrying out the team activities needed to satisfy a shared goal. Multi agent systems require that agents have the ability to plan and act collaboratively and avoid conflicting actions. In contrast with human collaboration, collaboration in agent teamwork poses many important challenges since in many cases there exist no global resource that agent team members can rely on to share their knowledge. In recent years, a recognized challenge regarding collaboration about multi agent systems is the collaboration among agents without prior coordination [Stone et al., 2010], [Balachandar et al., 2019].
2. Communication (COM): The general purpose of communication is to transfer information from one place to another. Communication is absolutely central to all aspects of working with others in groups and teams, e.g. making decisions, solving problems, sharing information, resolving conflict, planning and coordination and developing new ideas. In general, almost everything is done involves interacting with others. It involves the exchange of messages between senders and receivers, through a particular channel, possible subject to noise and interference. Effective communication entails the successful transmission of information, where the message received is the same message the sender intended. But there exist communication

problems, for example breakdowns in the transmission process. These could happen when a wrong channel is used to transmit a message, or noise interferes with the proper reception of the message. Communication plays an important role in multi agent action. To establish communication between collaborating members is an activity that usually requires shared knowledge, situation awareness and planning. Within the team context, through communication, agents send messages that represent their goals and planning mechanisms for fulfilling these goals. Therefore, an agent is aware at least of the objectives it can plan for and the tasks it can perform. In multi agent systems, agents use different means of communication across different channels (e.g., verbal, text, nonverbal). There are different ways of theorizing communication. Different explanations for what is going on when people or agents interact with others. But all matters for understanding of group interaction and working with other participants. As an example, dialogues and speech acts have been used to model communication [Poslad et al., 2000], [Sukhbaatar et al., 2016].

3. Coordination (COO): From early studies on agent teamwork, the problem of coordination is tackled as an essential requirement, the same that happens for human team working, too. Teamwork implies autonomy but, at the same time, interdependence among the members is often the case; the problem arises when members jointly perform a task and one member is dependent on other team members for achieving a subtask. When the task involves interdependence among members, coordination is required to achieve good team performance [Harbers et al., 2012]. According to [Barange et al., 2014] the coordination between human-agent team members is the problem where agents need to adopt the flexibility of human behavior, as humans may not strictly follow the rules of coordination. In order to achieve coordination in team context, members should share team knowledge and establish common grounding and mutual awareness. Team leaders that have the authority to guide their teams in order to affect the dynamics of coordination. One of the most well-known heuristics in team coordination was originally formulated by [Cohen and Bailey, 1997], where any team member who discovers privately that a goal is impossible, it is already has been achieved or it is irrelevant should notify this the team as a whole. For example, if a robot was asked to get a wrench from the garage so that it can help fix a sink, and the robot finds the garage door locked, it would be expected to inform the other partners about this [Bradshaw et al., 2008].

B. Strategy related factors

This category concentrates on characteristics that are fundamental for designing the strategy behavior of the team. Not all team members need to have such abilities, but it is

imperative some members to exist that could lead the team following the defined strategy.

1. Planning (PLN): A key issue in agent system teamwork is the planning task including recognizing and resolving conflicts among agents and within plans. In classical planning, the planning problem involves modeling explicitly the states of the world and searches for a sequence of actions that change the world from an initial state to a goal one. After a plan with a sequence of actions is found, the actions are assigned to the members of the team. They execute the actions in order to reach the final state with the requested goal.
2. Learning (LRN): The ability of members to "learn" as they identify new patterns in data that will enable participants to effectively discover opportunities and create strategies to improve teamwork experience.
3. Decision making (DM): Decision making process and problem solving is essential for effective team performance. Team members usually collaborate according to decisions made for a plan to follow. In recent years, researchers have designed agents that make decisions to support team behaviors. [Korsah et al., 2012] presented a multi robot control system, called xBots, that could support optimal task-allocation planning and flexible execution of the plans. The system optimizes the plans based on temporal constraints in the tasking environments [Chen and Barnes, 2013]. [Fan and Yen, 2004] developed the R-CAST agent system where agents adapt their decisions by detecting and responding to situations that are anomalies in comparison to the original expectancy (i.e., what is observed conflicts with what was expected); such anomalous situations can frequently happen in dynamic environments.
4. Evaluation (EVL): Evaluation methodologies and metrics that are used to determine the performance and effectiveness of human-agent teams and in what degree they achieve their intended effects. The team effectiveness is the result of the evaluation of the outcomes relative to some set of criteria that determines in what degree the team is workable under that context. This evaluation usually measures performance from the perspective of team formation alternatives, interaction efficiency [Scholtz et al., 2004], cost efficiency, trust issues, ad hoc capabilities [Barrett et al., 2011, Stone et al., 2010], time efficiency and plans evaluation mechanisms. For example, when using an evaluation strategy to select potential members during team formation, a team setting is distinguished from another not only according to member's capability actions but also according to the history of member's collaboration or other relevant factors that are used in order to approximate the optimal team behavior.
5. Teamwork policies (POL): Having the vision of heterogeneous teams for humans

and agents loosely or more tightly bound, e.g. hybrid or ad hoc teams, teamwork policies and mechanisms were developed. As an example, in [Bradshaw et al., 2004] policies are developed assuring that unsupervised autonomous agent behavior is kept within safe and secure limits of prescribed bounds, even if buggy, poorly designed, unsophisticated, or malicious agent code is developed. In addition, researchers attempt to use policies to assure effective and natural human-agent team interaction, without individual agents having to be specifically programmed with the knowledge to do so.

C. Synergy related factors

In this category the factors that are needed by members of a group in order to work as a team to achieve a shared goal are grouped. These factors are considered as the fundamental characteristics that a member should have in order to be able to participate within a team.

1. Ad hoc (ADH): The ad hoc factor determines the behavior that a team member exposes by which it is capable of cooperating with potentially unknown members without pre-coordination in order to achieve a common goal and reach an optimal team behavior. Potential members could be heterogeneous, with different capabilities and have no prior agreement regarding how to work together. In realistic environments, many teams are ad hoc, such as teams met in rescue, military and industrial domains. Applications include rescue robots brought to an earthquake site from different parts of the world, or e-commerce cooperative agents created by different companies with different standards [Stone et al., 2010].
2. Autonomy (AUT): Generally, autonomous systems are those that have decision-making ability and do not depend directly from another coordinating system. There exist different levels of autonomous systems where, at the highest autonomy level, the system decides everything in order to act autonomously, and at the lowest autonomy level, the system offers no assistance while human members must take all decisions and actions. Fully autonomous systems are capable of more functionality. In such systems, the tasks usually are not assigned centrally by the system, but they are chosen by the team members according to their individual skills and capabilities to take it over. In this thesis, we took into consideration only studies where an agent participating in a team has a minimum level of autonomy, being able to take simple or more complex decisions based on its current state. Some of the studies investigate team members autonomy are [Harbers et al., 2012, Wu et al., 2011, Schwartz et al., 2016].

3. Delegation (DLG): Delegation is defined as a transfer mechanism of all or a subset of member's activities to other members and its aim is to increase flexibility. For the better performance of the entire team, it is often needed for members to have the ability to choose to transfer their tasks. The delegation behavior is essential for those members that can assign responsibility of tasks and are able to choose what to delegate from a member to other members and what to leave. While delegation can allow substitution of a member and solves the unavailability problem of a member successfully, it might also potentially lead to undesirable situations if it is applied incorrectly [Saidani and Nurcan, 2006].
4. Joint-intention (INT): The common intention that systems have toward the group goal; the commitment and focus they demonstrate in order to handle unexpected failures. Joint intention is a central concept during the whole process of teamwork and changes dynamically during planning. Managing the maintenance of joint intention to all individual involved through the whole process in changing environments is necessary according to [Dunin-Keplicz and Verbrugge, 2011].
5. Knowledge of teammate's capabilities (KCT): The teamwork literature has identified the need for agent members, except from knowing their own capabilities, to also be aware about capabilities of their teammates. There are systems that enable agents to learn about capabilities of their teammates, model them and then use that knowledge to improve collaboration. [Kaminka and Frenkel, 2005] present a model that allows an agent to predict the future behavior of other agents in order to coordinate their activities by observing those agents and building a model of their behavior. Also [Kennedy et al., 2008] exploit cognitive science approaches to produce an autonomous robotic system that "mentally" simulates the decision-making of each teammate.
6. Knowledge sharing (KNL): For teamwork to succeed, participants need to establish and share a common view for specific issues related to the environment. Performance is improved when members have a shared understanding of the tasks, goals, intentions and plans of the involved. According to [Dunin-Keplicz and Verbrugge, 2011], the awareness of the group is of vital importance. This could be established through knowledge sharing and helps the group to predict future system states.
7. Trust (TRS): For agents to be successful as team members, it is imperative to act similarly to human's behavior, meeting people's expectations of how they will behave. Not only human but also agent's behavior in group settings has been shown to depend on sociological and psychological aspects, such as trust and fairness

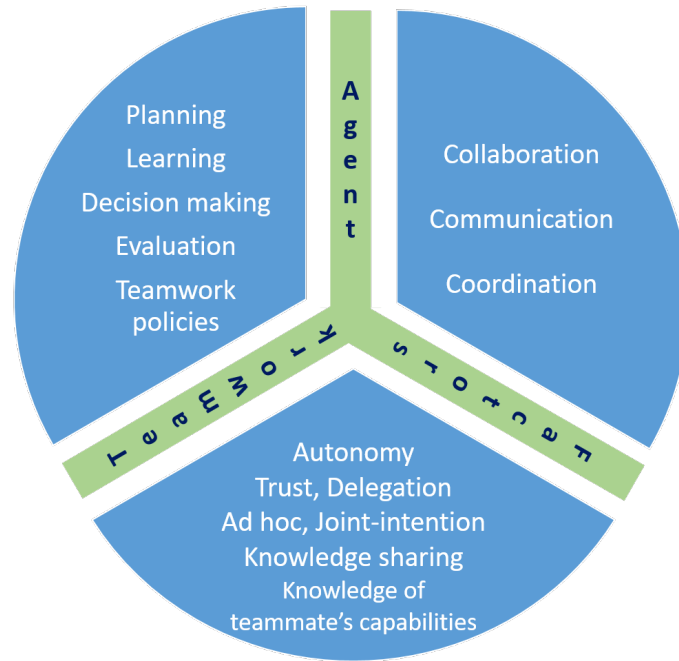


Figure 4.1: Wheel of factors that affect team working

[Van Wissen et al., 2012]. In autonomous systems, the issue of trust is taken into account and the factors that help to build trust relationships encompass both the social and ethical aspects. As intelligent systems become increasing sophisticated and are capable of accessing information from other networks, it is necessary to examine the implications of these capabilities on trust of the systems [Chen and Barnes, 2013]. Researchers suggest that agent designers should include the issue of trust generation in agents' decision-making models in order to promote the development of trust relationships between participants when they act as team members [Van Wissen et al., 2012].

Factors that affect team working are distributed into three clusters as they are highlighted in the team work agent wheel given in Figure 4.1. The wheel has three sectors labeled as: (a) Organizing, (b) Strategy related, and (c) Synergy related factors, which are the aforementioned clusters.

4.2 Enhancing Teamwork Behavior of Services

Motivated by the organizational behavior and the teamwork theory [Margerison and McCann, 1990, Belbin, 1993], in order to bring teamwork abilities to service composition, we argue that services similarly to agents and robots, except from functional behavior they also have to exhibit teamwork behavior when they cooperate. In case that there do not

exist member services that exhibit such behavior, then other services should be included into the composition in order to provide this behavior. This behavior will be beyond the service's individual capabilities. Following it, they exploit their ability to better cooperate since for reactive systems in the real world things do not always go as planned.

To achieve this and based on the role-model paradigm for services presented in Chapter 3, we introduce teamwork role models enhanced by service teamwork roles. We propose eight teamwork service roles to be involved during the composition process which are developer, organiser, promoter, monitor, producer, advisor, innovator and maintainer. These roles aim to (i) enable services manipulate the uncertainties by being proactive during their participation in specific events, (ii) capture potential emergent behavior happens in a domain during service design and execution, (iii) arrest and handle any misbehaving of other participants and emergent behavior that occurs in the domain, (iv) act as the glue needed for reconfigure and combine services appropriately in dynamics and unpredictable environments. The eight roles were derived by combining the teamwork theory focusing on the behavior of high performing human teams and the fifteen teamwork factors emerged from agent team working, which are actually the behavioral descriptors for these roles. The enriched model can be proved beneficial to create more efficient and stable teams of services, i.e. composite services such that their components-members promote better understanding and much more likely they are able to cooperate effectively and achieve their objectives.

4.2.1 Bridge Teamwork Factors to Service Teamwork Roles

We research the potential of applying the theory focusing on the qualities and the skills of high performing human teams towards providing more efficient and intelligent services. By surveying to existing major teamwork models in human behavior [Belbin, 1993, Margerison and McCann, 1990, Mathieu et al., 2008], we derived eight integrated roles that model human behavior. After adapting the role description from the human model to the service model we direct and combine the agent factors that affect teamwork to agent models and as a result, we contribute with eight *service teamwork roles* in order to improve the process of service composition.

We present these novel roles in Table 4.3. The proposed roles are defined with the aim to meet the performance challenges that dynamic domains are facing, in terms of various types of services and their non-functional objectives, as they are also indicated in [Zeshan and Mohamad, 2011, Nacer and Aissani, 2014]. They are employed in order to act as the glue needed for configure, reconfigure and connect services appropriately in dynamic and unpredictable environments. We outline the behavior for each service teamwork role by coupling the desired behavior of service team members with the discoveries from

human and agent based teamwork. By this, services are capable of capturing the foreseen and emergent behavior during their execution in dynamic domains and they are able to successfully bring to success the cooperation and execution process.

By characterizing the concept of each teamwork role in terms of teamwork factors identified in agents a synergy between the agent teamwork and services systems perspectives and results is created. Thereafter, Table 4.3 presents for each teamwork role the teamwork factors that describe their behavior it should extract during the composition process while in the next section we present the main goal for each one with more details.

Table 4.3: Agent teamwork factors for each teamwork service role

AA	Role	Factor
1	Developer	AUT, INT, KCT, KNL, COO, COL, COM, DM, POL, EVL, PLN
2	Organiser	ADH, DLG, INT, KCT, KNL, TRS, COO, COL, COM, DM, POL, EVL, PLN
3	Promoter	ADH, INT, KCT, TRS, COO, COL, COM, DM, POL
4	Monitor	ADH, AUT, INT, KCT, KNL, TRS, COO, COL, COM, DM, POL, EVL, PLN
5	Producer	ADH, AUT, DLG, KCT, KNL, COO, COL, COM, DM, POL
6	Advisor	KCT, KNL, COO, COL, COM, DM, POL, LRN
7	Innovator	KCT, KNL, TRS, COO, COL, COM, DM, POL
8	Maintainer	ADH, KCT, KNL, COO, COL, COM, DM, POL, PLN

4.2.2 Dynamics of Teamwork Roles during Service Composition

Teamwork roles technically assist during the creation, modification and monitoring of execution of composite services. They enable the design of a long-lived system in a domain that could constantly be evolving to meet the needs of the users. The reference service teamwork roles are describing the building blocks needed to meet the main functional and non-functional requirements that will guide the development of next-generation composite

services. The description of each role is basic in the sense of dealing solely with the most substantial aspects of teamwork during the composition of a service. An additional definition is necessary in specific domains and cases may be addressed by updating their behaviors.

Once we have identified the teamwork roles participate in the composition, it is then important to write a description on what these roles do. From this description, we can easily figure out the interactions that would occur between the roles once the composition is executed, and these are depicted at sequence diagrams in the following sections. The aim of the following sections is to provide a basic description of roles that will be used by highly automated teamwork services. With this effort, we intend to facilitate service designers and developers with the most important top-level behaviour patterns by formalizing them into a useful reference. The focus is to address services teaming characteristics that give them a context-aware implementation, enable to better manage other resources, while at the same time deliver lasting solutions. Among others, role teaming behaviour patterns will help domains so that, instead of building every team role from scratch, to have a guide that will help them solve a common problem that may have been encountered before.

4.2.3 Using Design Patterns for Creating the Teamwork Roles

Design Patterns (DP) [Gamma, 1995] systematically names, explains and evaluates an important and recurring design in systems. Design patterns are defined by their purpose or intent, and not the exact code. We use the DP like a template method to define the skeleton for the operation of each role, and have the possibility of deferring the implementation of some steps until writing subclasses. The template classes contain both abstract methods and non-abstract methods. A template method lets subclasses redefine certain steps of the algorithm (abstract methods) without changing the algorithm's structure. The objective is, to follow where possible, both the service reusability and service composability principles for services playing the teamwork roles. While the first dictates that services should be designed to maximize reuse, the later advocates designing services so that they can be composed in various ways. By following the model DP paradigm we give the opportunity to reuse non-abstract methods and customize the abstract ones. For example, developer roles use the abstract Developer template method when developing a new service. A typical developing includes the invocation of a number of services with different variations available for each. The variant or customizable steps of the composition algorithm represent teamwork abilities, that can be supplied by the domain to the concrete derived classes. We mandate the required steps for the composition algorithm but allow the component role to extend or replace some number of these steps.

We proceed by giving the description of each role, information about its responsibilities

and by naming all the necessary teamwork factors it includes. Additionally, we present for each role a design pattern providing a reusable solution to occurring problems within a given context in a role domain. The objectives of the following role descriptions is to describe the intent, applicability, and consequences of the eight teamwork roles during the service composition. As already mentioned, it is not a finished design that can be transformed directly into source code. It is a template for how to solve a problem that can be used in many different situations, since design patterns are formalized best practices that can be used to solve common problems when designing an application or a system.

Moreover, we give examples of the behavior of each role by using UML Sequence Diagrams. In UML, Sequence Diagrams (SD) describe how objects, or groups of objects, interact within a system [Rumbaugh et al., 2004] and they are typically used to describe system dynamics. They depict it by showing the participating objects in the interaction and the sequence of messages exchanged. A SD is also used in visualizing the behaviour of software systems and we use it to show how actors and objects interact to realise the composition by focusing on the message interchange. A SD has two dimensions: the vertical dimension represents time and the horizontal dimension represents the objects participating in the interaction. Time flows from top to bottom and the focus is on the order in which messages are sent. The emphasis in these diagrams is what happens first, second, and so on. Objects or roles are shown as vertical lines, which are called lifelines, and messages as horizontal arrows extending from a sender object to a receiver object. Activations bars are the boxes are placed on the lifelines. They indicate that an object is active during an interaction and its length indicates the duration of the interaction. Spacing is irrelevant, that is, only the order of messages matters, not the distance between them. Invalid behaviours in our diagrams are presented using the Negative (neg) operator. The interaction operator *neg* describes combined fragment of traces that are defined to be invalid, that is, behaviours that should not happen and be captured by teamwork roles. Negative traces are the traces which occur when the system fails. to ensure that exceptions are dealt with properly. When an abnormal situation occurs, one of the teamwork role is expected to handle it. All interaction fragments that are different from the negative are considered positive, meaning that they describe traces that are valid and should be possible. The role pattern and sequence diagrams of all roles that follows aims at providing solutions for common teamwork problems may have been encountered in the domain concerning the scope of the composition process.

4.2.4 The Developer role.

The goal of the *Developer role* is to compose new services. It is the “team builder” of the domain; it binds the appropriate services which play the required roles having the

specific behaviors to be executed for the service composition.

Since many candidate web services that are published in the web are enrolled in the domain to play a role and implement a behavior, the number of possible behavior allocations is very large and the developer needs to narrow the results of the candidate services to be included as potential participants to the composition. Moreover, if we take into consideration the different quality attributes of each service, as the result of the composition plans, the instants of composite services may be produced are quite different from each other. The team development involves everything groups do to form and operate, i.e. how participants come together, how they evolve and dissolve, and the different phases participants can go through. The developer role initially focuses its attention in finding a plan according to the goal of the desired query. It receives a request as a (*precondition, effect*) pair and search for a feasible plan for the composition. It looks for a plan of behaviors that if executed will satisfy the user requirement. It makes a constructive reasoning to generate a plan for the composition in the sense that it has to decide before the service team formation for the service-members of the team. After a plan is found, either an existing one from the plan library of the domain or by reasoning, it does an evaluation for candidate services enrolled to domain in order to select the most appropriate to play the role and then makes the service allocation for the execution of behaviors. During the allocation, if a service is unavailable the developer role proceeds with another candidate service allocation.

After the service allocation is completed, the team is formatted, it calls the monitor role to inspect how the team of the selected services is performing while executing the plan. The service that will play this role is a vital ingredient of the new composed service since it is needed to have awareness of the environment that the team lives and possible reconfigure and replans to adapt according to new needs.

To conclude, the main goal of the developer is to create composite web services that have teamwork abilities. The functionality of the composition would be greatly improved by the introduction of teamwork roles that support the whole process. Developers design, develop and upload teamwork enabled services that expand their capabilities.

Role Pattern Template for the Developer Teamwork Service Role

Name :	Developer role that form the new composite services of the domain.
Type :	Teamwork service role
Problem:	Develop teamwork enabled composed services
Context:	Composition process
Primary behaviors	AUT, INT, KCT, KNL, COO, COL,

of the role: COM, DM, POL, EVL, PLN

Forces: Advisor, Monitor, services participating in the composition.

Proposed Solution: During the composition of services the developer role finds a plan and in cooperation with other teamwork roles of the domain qualifies the development of teamwork enabled composed services.

Resulting Context: A composite service is generated, by establishing teamwork attitude during its execution. The team development involves everything groups do to form and operate, i.e. how they come together, how they evolve and dissolve, and the different phases participants can go through.

Design Rationale: The notifications of updates and actions on the semantic domain is through the behaviors are executed when the events happen. The state of the system is updated by the executed behaviors. Examples of behaviors include operational behaviors, e.g. creation and dispersion of a new plan across the peer involved roles and the inclusion of services to participate. Allocation of behaviors also include the teamwork ones, like monitoring and notify for situations that are previously partially described to be tackled.

Here are the steps that occur from the composition process where the developer role is involved.

- The advisor search for an appropriate plan in the plan library.
- If no plan exists, the advisor then requests the developer to reason for new plans.
- The developer returns the plans or empty if it is not possible to find one.
- If plans exist, either from the library or new ones as the result of reasoning the advisor evaluates them.
- The sequence of the behaviors to be executed from the selected plan is sent to the developer.
- The developer then continues with the behavior allocation until all the behaviors are allocated to a service.
- In case a service is unavailable the behavior is allocated to another candidate service having the same behavior.
- Once the behavior allocation have finished the team is formatted including all the necessary atomic services for the composition.

- The developer commission then the monitor role to inspect the team operation.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the developer that follows in Figure 4.2.

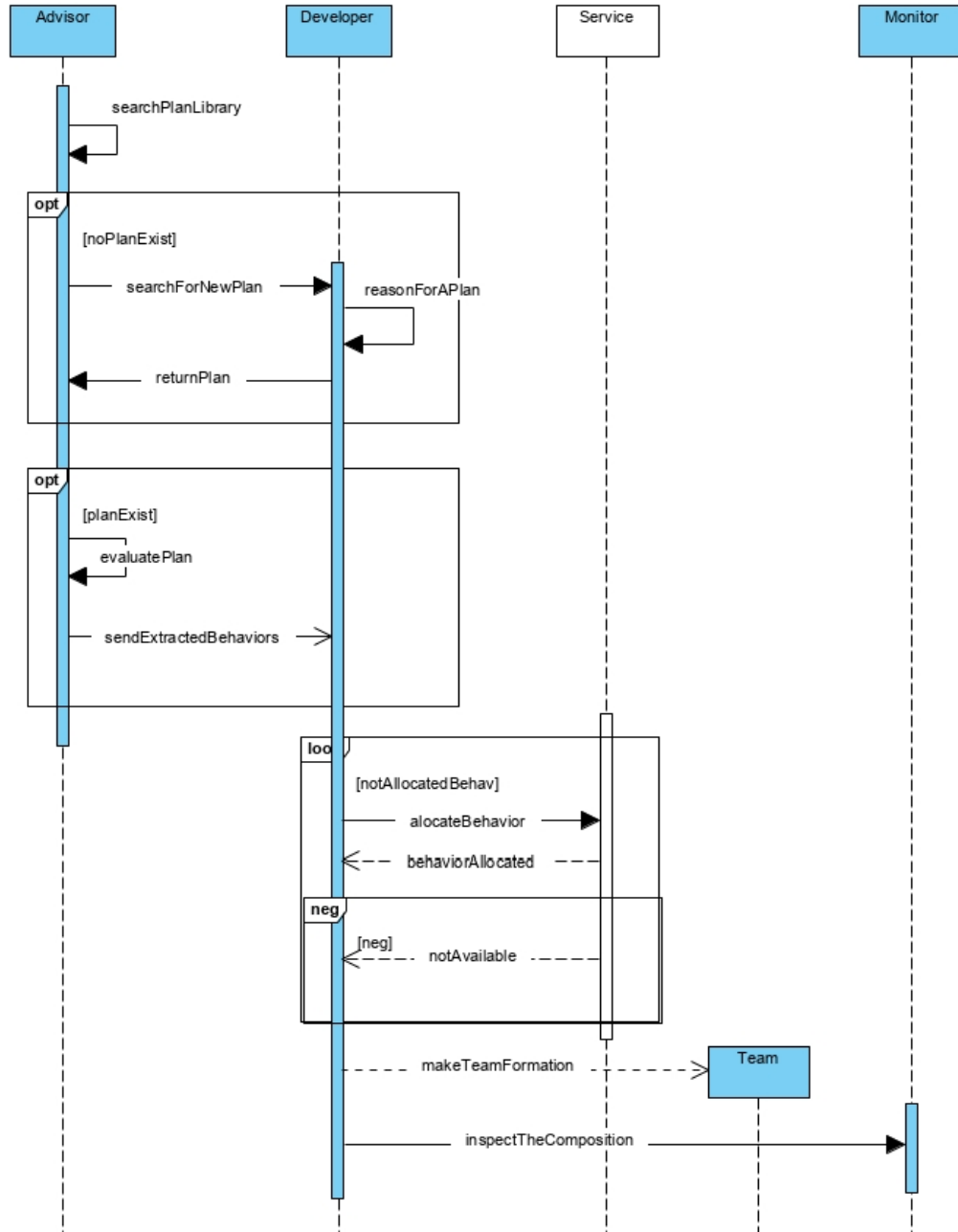


Figure 4.2: The Sequence Diagram for the Developer role.

4.2.5 The Organiser role.

When developing a composite service, it is important to keep track of how the different services work together towards the common goal. Each service is used is possible a piece

of a larger solution. The plan of behaviors that derive from the role model depicts the pattern in which the independent services cooperate towards the common goal. In order for each one to do its work effectively, it needs to have a clear purpose that is known to other participating members of the team especially to the organiser.

The organiser monitors the global schedule and implements ways of making things work for the team purpose to be accomplished. It keeps track of accomplishments and how teams are progressing relative to goals and timelines ensuring that standards and processes are upheld. It keeps track of queues, average time responses and urgent. It has the ability to present the availability of the overall team to other roles. It gives feedback to promoter in case the system is overloaded, to look for other outlets. It is responsible to distribute work among services. In case a service is unavailable, it delegates the execution of the behavior to another available and appropriate service. It organizes how each service does a single cohesive function and also how independent services work towards a common goal. Also, when a gap between services that run for an existing plan occurs, it re-assigns the service execution to another service and substitutes the missing services in order to fill the gap. It also receives the maintenance queries for services of the domain and schedules them.

Role Pattern Template for the Organiser Teamwork Service Role

Name :	The organiser role organises the different services to work together towards a common goal.
Type :	Teamwork service role
Problem:	Organization of how each service does a single cohesive function and also on how independent services to work effectively towards a common goal. The team needs a member that is aware of the availability of the overall involved services that can distribute the work among available services.
Context:	Composition process and during prudential supervision of operations.
Primary behaviors of the role:	ADH, DLG, INT, KCT, KNL, TRS, COO, COL, COM, DM, POL, EVL, PLN
Forces:	Promoter, Monitor, Maintainer, services participating in the composition
Proposed Solution:	During the composition of services, the organiser role gets involved as a teamwork role to maintain a balanced structure among the involved services and organises the work of individual members inside the team and possible work is made between different teams.
Resulting Context:	All the participating services in the team to work effectively to

achieve the common team goal.

Design Rationale: This role is designed to organise the team and also act as the link to enable the connection of different teams.

Here are the steps that occur from the composition process where the organiser role is involved.

- An organiser monitors the global schedule.
- It keeps track of accomplishments and how teams are progressing relative to goals and timelines.
- It gives feedback to promoter in case the system is overloaded, to look for other outlets.
- It makes delegation to address unavailability of services or to distribute work among services.
- It receives and plan the service maintenance queries.
- It commission new services to substitute missing services.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the organiser that follows in Figure 4.3.

4.2.6 The Monitor role

In dynamic environments where anything is changeable, some of the potential that the monitor role has is that, it can establish stability in the current scope by maintaining a high level of surveillance while staying consistent to the domain objectives. The regulations under which a domain operates and should be monitored are depicted as rules in a semantic domain content ontology. More obligations also derive from the execution of a specific plan for the composition. It monitors the consistency of these to maintain team functionality during the whole process of service composition. It ensures that participants do not infringement these regulations and are complied with the obligations, due to their role acting in the specific case within the domain. The monitor focus on the detailed and controlling aspects of the work and they check and audit that system and contracts are working by inspecting all activities happen that are caused by team members. It is also aware about the performance of the components, for example the memory, storage

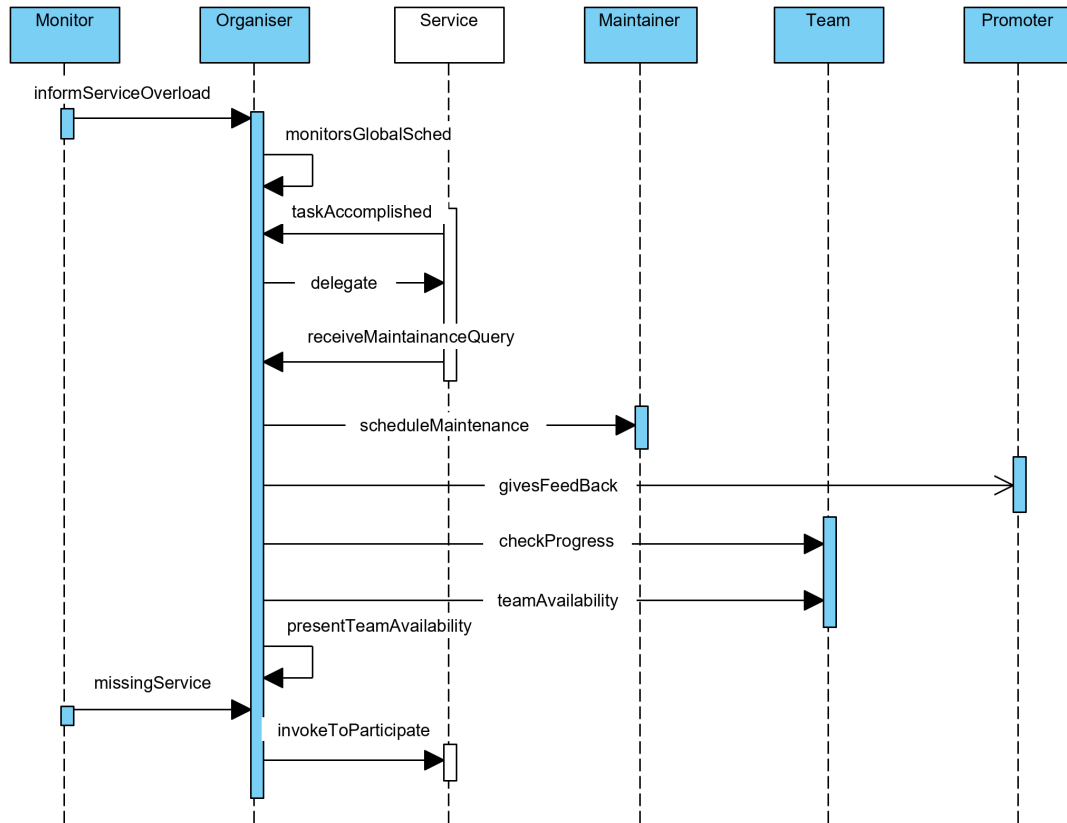


Figure 4.3: The Sequence Diagram for the Organiser role.

performance, network latency, available bandwidth, etc. Its purpose is to help roles to execute a behavior with success by providing the necessary information could be missing.

It receives the plan for the execution from the developer role. This role is also enabled to achieve cooperation between peer roles by catching errors that might occur during the communication. For example, there are web services which might be inherently unreliable e.g., due to the failure of the hosting. Role implementations that rely on such resources need to be able to adapt to such situations. If emergent behavior occurs, that could be adapted it will contact the advisor role. Although the developer will do its best to prevent conflicts and it can try to avoid it when it comes up, but eventually, if they happen conflicts need to be addressed and confronted by the monitor role. When a service is caught to be out of order, the monitor role informs the maintainer role in order to take care of it while, in case a service suffers from overload, informs the organiser role to re-plan the production process by delegating work to available services.

Role Pattern Template for the Monitor Teamwork Service Role

Name : Monitor roles establish protection and surveillance in order for the composition process to stay consistent with the objectives

	of the domain.
Type :	Teamwork service role
Problem:	To inspect, recognize and predict failed actions that block the main goal of the team or interferences that occur on business objectives during the composition process.
Context:	Composition process
Primary behaviors of the role:	ADH, AUT, INT, KCT, KNL, TRS, COO, COL, COM, DM, POL, EVL, PLN
Forces:	Developer, Advisor, Organiser, Producer, Maintainer, services participating in the composition.
Proposed Solution:	The monitor role is on duty while the whole composition process takes place. It inspects all team participants and works in consultation to the organiser and the adviser role in order to take remedial actions for team survival. This could include substitution of a team member, maintenance of a service, even change to the plan if the advisor instructs the team to do so.
Resulting Context:	By monitoring the composition process, this role intervenes when an unexpected event being caused or predicted and either acts according to predefined actions or informs the advisor role to act. This is a reason why it is possible in the same domain to be developed composite services with the same effects but each time following different purposes and in different ways.
Design Rationale:	The behavior of the monitor role is mainly a detection and reconfiguration behavior which deals with failures of others behaviors execution. It is divided in a number of cases that predict or arrest misbehaving of participants or other sources of the environment.

Here are the steps that occur from the composition process where the monitor role is involved.

- Monitor the consistency of participant services to maintain team functionality during the whole process of service composition.
- Ensure that participants do not infringement domain regulations and are complied with the obligations.
- Inspect all activities happen that are caused by team members.
- Provide among team members necessary information could be missing.

- Catch errors that might occur during communication of services.
- Inform the maintainer and the organiser role to take care of out of order or overload services.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the monitor that follows in Figure 4.4.

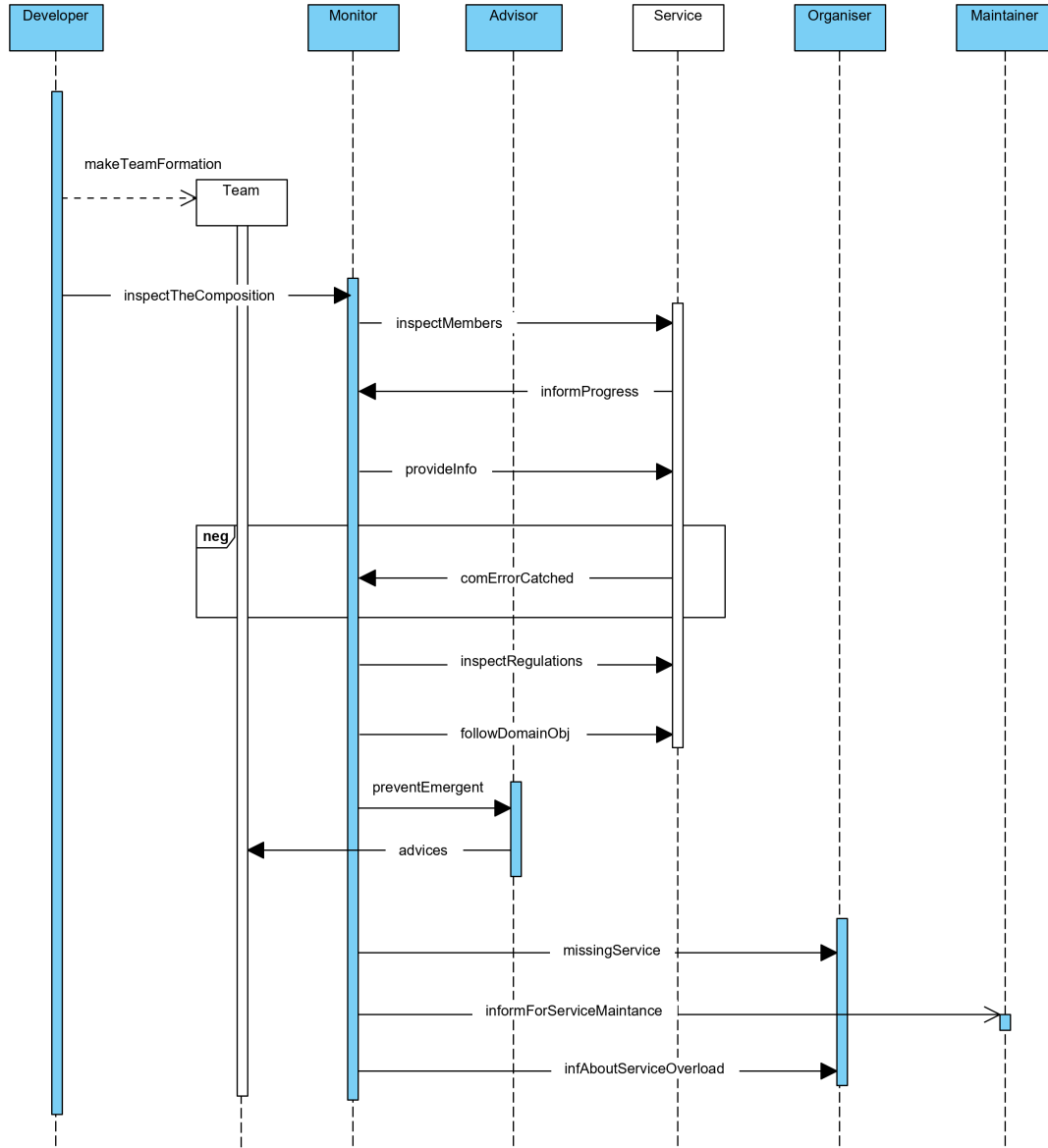


Figure 4.4: The Sequence Diagram for the Monitor role.

4.2.7 The Advisor role

Upon to a user request for a service, the advisor of the domain takes the request and searches if it matches according to its semantic similarity with existing goals of the

domain repository. If it is so, it retrieves a plan and provides it to the developer role in order to proceed with the service allocation. Otherwise, it informs the developer role to reason over a new plan. Moreover, this role gathers information and advises other roles. It concerns about data accuracy to assure consistency and checks the data integrity if decisions based on those data are to be taken. For some services, e.g. the IoT services that are enrolled to the domain and bring data from the physical world (e.g., via sensors), a degree of unreliability and uncertainty is introduced. It also tests the applicability of new approaches proposed by the innovator role. It assists the team by analyzing previous plan executions taking into consideration previous errors and recommends solutions, e.g. bad communication. Communication problems not always are resolved with a straightforward adjustment. A solution for example, could be to try a different channel, or to eliminate some of the noise. But if we go deeper, we realize that there is so much more going on in most communication situations than just message transmission, although there is certainly is at least that.

The effectiveness of all interactions during the composition, planning schedules, requesting information, giving directions depends on when the messages are transmitted without distortion, and when messages are received, and interpreted in the same way that they were intended. Otherwise the transmission could break down and have a failure to communicate. The fact is that many of the communication problems are marked by instances where participants actually do understand each others messages fairly well, yet still don't align, connect, or agree. Part of the problem is that many times the transmission model is developed to understand and explain information processing in static systems that involve transmission of information but not in cases especially for group interaction. If the conditions necessary for a team model to work effectively are not satisfied then directions are not followed, regulations are not obeyed, and general chaos often ensues. To account for these issues participants possibly should be advised to communicate the right information, to the right participants, at the right time, in the right ways, otherwise the composition is going to fall apart.

With this role we also want to make use of a knowledge base could be build from existing data of the domain and information available in the web. Data are collected from previous web services compositions and executions. The collected data will constitute a database of knowledge, this knowledge will help the domain and eventually the total procedure of choosing and using an appropriate WS. This knowledge could derive over reasoning on domain ontologies which are the main vehicle for semantically and conceptually oriented techniques and applications. Using the new knowledge, it conducts various analysis and correlation operations on the query comparing the candidate web services, and finally this role advises the developer to compose the service.

Role Pattern Template for the Advisor Teamwork Service Role

Name :	The advisor role is concerned to advice the user asking the composition by proposing relevant plans. It also advices the team by testing new approaches, evaluating previous compositions and by exploiting the knowledge that derive from the semantic description of the role domain, its components and their functionality.
Type :	Teamwork service role
Problem:	To find appropriate plans for the composition, if exist, that their goals have matching based measures that are better compatible to the request. To benefit from domain knowledge and advice team members in order to ensure more successful group communication.
Context:	Composition process
Primary behaviors of the role:	KCT, KNL, COO, COL, COM, DM, POL, LRN
Forces:	Developer, Monitor, Innovator, services participating in the composition
Proposed Solution:	Before the composition, the advisor role search for an available plan that semantically matches with the user request. To ensure that the team works effectively it checks the conditions necessary to communicate the right information, to the right participants, at the right time.
Resulting Context:	A team of services that will be advised by a member and can reuse tested solutions or explore new opportunities and surely avoid predetermined fault situations.
Design Rationale:	This role is designed to explore the knowledge derive from the semantic description of the role domain and its components in order to transfer it to other participants of the team.

Here are the steps that occur from the composition process where the advisor role is involved.

- Searches if the client request matches with existing goals of the domain repository.
- provides a selected plan to the developer role in order to proceed with the service allocation.
- It gathers information, maintain the knowledge base and advices other roles.

- It checks the data integrity.
- It tests the applicability of new approaches proposed by the innovator role.
- It analyzes previous compositions and recommends solutions.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the advisor that follows in Figure 4.5.

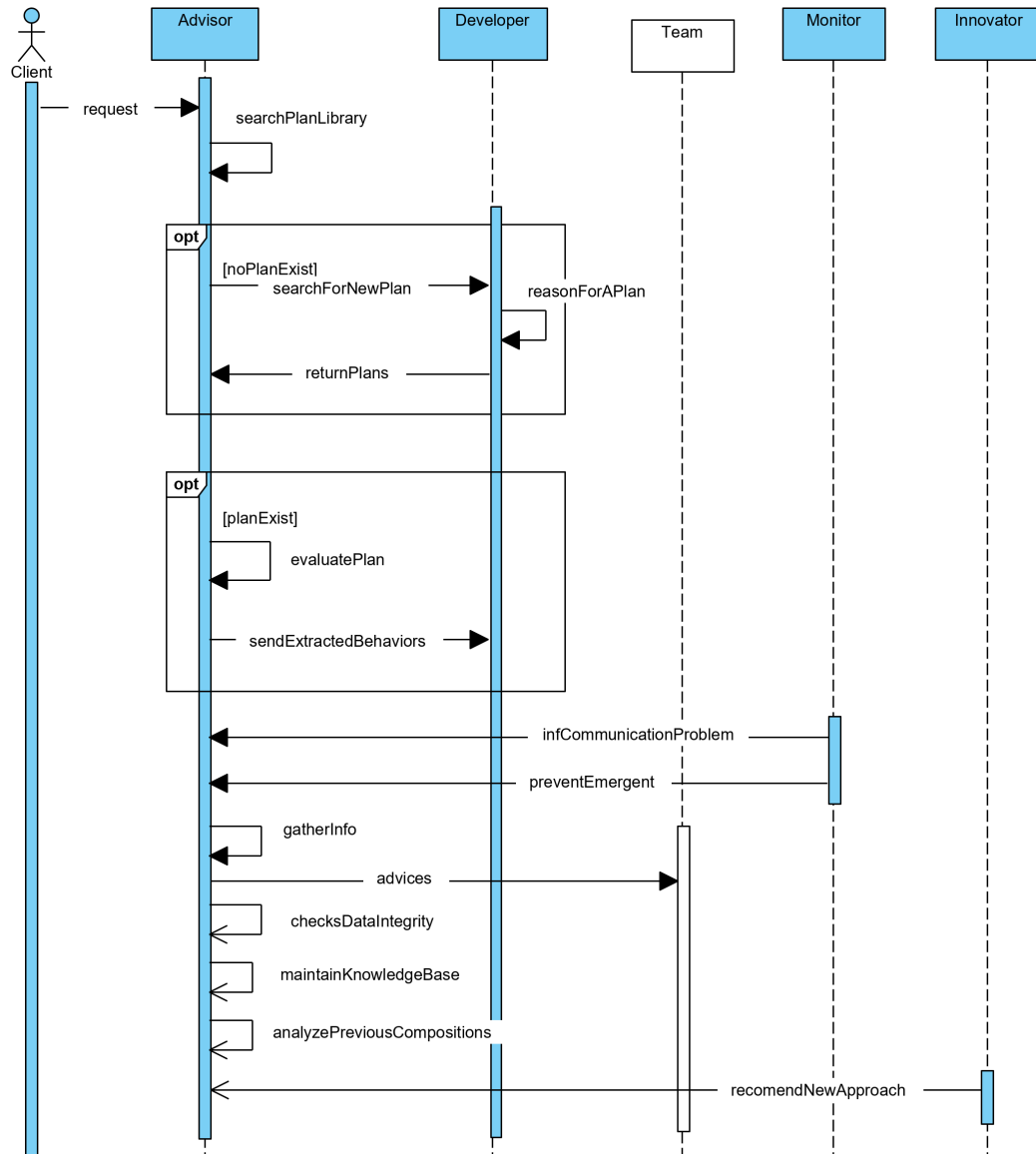


Figure 4.5: The Sequence Diagram for the Advisor role.

4.2.8 The Innovator role.

The innovator role explores new ways in the process of composing specific services in order to foster innovation. It explores new plan workflows, checks new updates for

services to substitute outdated ones. It searches for new opportunities in services that have similar goals and consider them as new resources, explores possibilities and looks up for new solutions from new deployed services in the web after testing the applicability of these new approaches to form novel services.

Although in a domain many of the compositions involve routine decisions that could be complicated, they are relatively straightforward since many of them are predesigned. But in order for the domain to be dynamic, participants could generally involve applying existing standards and methods to decide among various alternatives. In some cases, decisions have a different dynamic, since they require a degree of novelty and originality to come up with new ideas. Creativity and innovation could emerge when certain services interact with each other in certain ways which not always follow the original plan. It is the outcome of a process of analysis that results in new ideas and innovative ways to solve problems and are subject of new strategies, creating and experimenting with different approaches in order to handle various queries. The best ideas usually come from a back and forth process or team work and individual work, refining ideas both individually and in connection with others which enhances the overall creativity and innovation of the domain.

Another common situation during the composition problem is to have a service that expects one sort of input, but there are available services that provide a different input. Because of the mismatch, we can not make that bind among the services. What we need here is an 'adapter' service that as an effect, turn the different output into one that some other service expects. These kind of services are frequently used because different vendors will have different and evolving connectors for their devices. It isn't always practical to replan for a composition of the system in order to properly bind with new third-party libraries or external systems. Just because an input does not conform to what a service is expecting, does not mean that the whole composition process has to change. The innovator role by using the adapter's technique can help bridge the gap between two incompatible services. This will let the composition continue to partially use existing plan and resources by integrating some external sources into them.

Role Pattern Template for the Innovator Teamwork Service Role

Name :	The innovator role enhances the innovation and overall creativity of the domain by developing new strategies.
Type :	Teamwork service role
Problem:	To have a dynamic domain that not only use existing standards methods but also explore various alternatives and make decisions

	that have a degree of novelty and originality.
Context:	Composition process and as standard prudential operation.
Primary behaviors of the role:	KCT, KNL, TRS, COO, COL, COM, DM, POL
Forces:	Advisor, services participating in the composition
Proposed Solution:	Test new plan workflows, check new updates for services and exploit the outcome of conversations and connections to make an analysis process with the aim to result in new ideas and innovative ways to solve problems.
Resulting Context:	Innovative composite services that could be developed differently each time and seize opportunities of newly deployed services in the web.
Design Rationale:	This role is designed to mostly work asynchronously for finding alternatives solutions. When adaptations are needed to transform a non suitable service to a suitable one it intervenes and search for an adaptor service in the course of specific service invocation.

Here are the steps that occur from the composition process where the innovator role is involved.

- Explores new ways in the process of composing specific services in order to foster innovation.
- It checks new updates for services to substitute outdated ones.
- It searches for new resources.
- It recommends new solutions from new deployed services to advisor.
- It can help to bridge the gap between two incompatible services.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the innovator that follows in Figure 4.6.

4.2.9 The Promoter role.

The promoter works in a twofold aspect, by searching for new opportunities and exploring possibilities inside and outside of the domain. It enables clients of the system to uncover new relevant service opportunities that could be used. It takes feedback from organiser and producer roles and, if the system is overloaded, it looks for solutions, e.g. other resources inside or outside of the domain that could help. If the system has

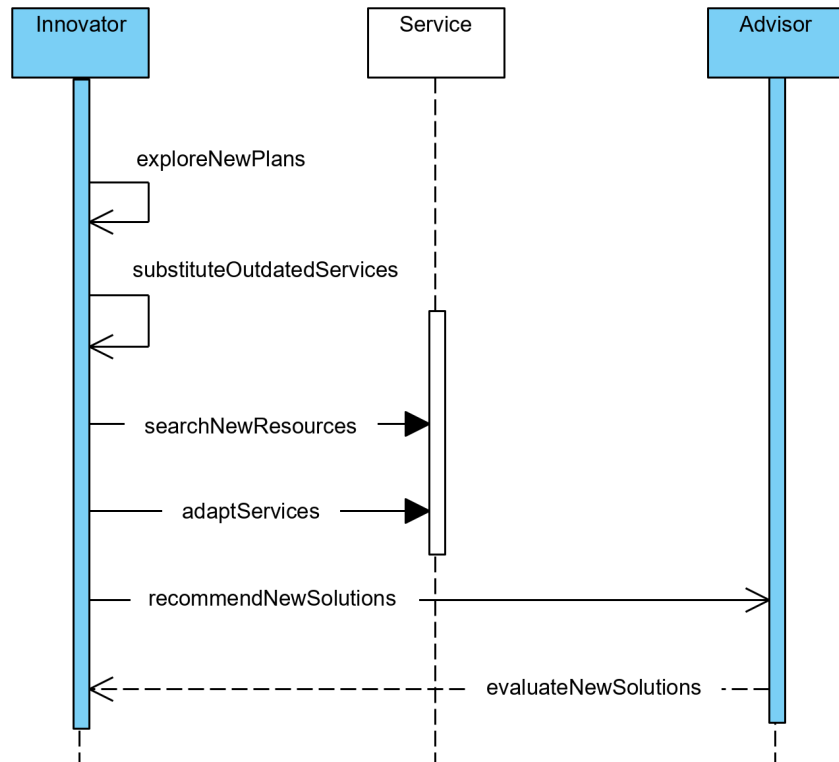


Figure 4.6: The Sequence Diagram for the Innovator role.

unused available services, searches for new opportunities to promote their usage and it is responsible for the integration of composed services with external systems. After analysis, it recommends solutions of the domain to new markets. When resources inside the domain cannot overcome a problem, the maintainer informs it to search for providers outside of the domain. It maintains a service catalogue where it stores service's features that are advertised by various web providers. Since it is common to integrate local with external services, commonly through new composite services, the promoter helps bridge and connect services of the domain with services outside of it something that improves the service's integration capability.

Role Pattern Template for the Promoter Teamwork Service Role

Name :	The promoter roles search for opportunities to facilitate and promote usage of services inside and outside of the domain.
Type :	Teamwork service role
Problem:	Available services must find room for exploiting their unused potential while when system is overloaded, solutions should be found e.g. other resources inside or outside of the domain.
Context:	Composition process and during standard domain operations

	on appropriate sections
Primary behaviors of the role:	ADH, INT, KCT, TRS, COO, COL, COM, DM, POL
Forces:	Organiser, Maintainer, services participating in the team.
Proposed Solution:	It receives messages from the organiser and the producer role when there exist unused services or there are missing resources.
Resulting Context:	Services that their execution is blocked due to lack of resources can be used again. Potential unused services can be promoted to domains that new markets are present.
Design Rationale:	This role is designed to work both synchronously and asynchronously. In case a composition is on-going it facilitates possible missing services. During the standard operation of the domain it can check the overall availability and promote service usage to new markets.

Here are the steps that occur from the composition process where the promoter role is involved.

- It searches for new resources and opportunities.
- It promotes unused services and new solutions to clients.
- It takes feedback from the organiser and the producer.
- It maintains a service catalogue.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the promoter that follows in Figure 4.7.

4.2.10 The Producer role.

The producer role works on a systematic way to produce work outputs. Just like a factory in the real world, the purpose of services play this role is to produce. It is the doer of the system; this role characterizes, for example, all manufacturing services. It is invoked by the developer role to participate in a case. In general, they give feedback to the promoter who searches for opportunities to promote their usage. If smart machines play this role, they have a list of tasks to do in order to know their schedule and they could individually keep track of queues, balances, etc.. According to these lists, they inform the organiser about under using or if they exceed workload capacity.

Role Pattern Template for the Producer Teamwork Service Role

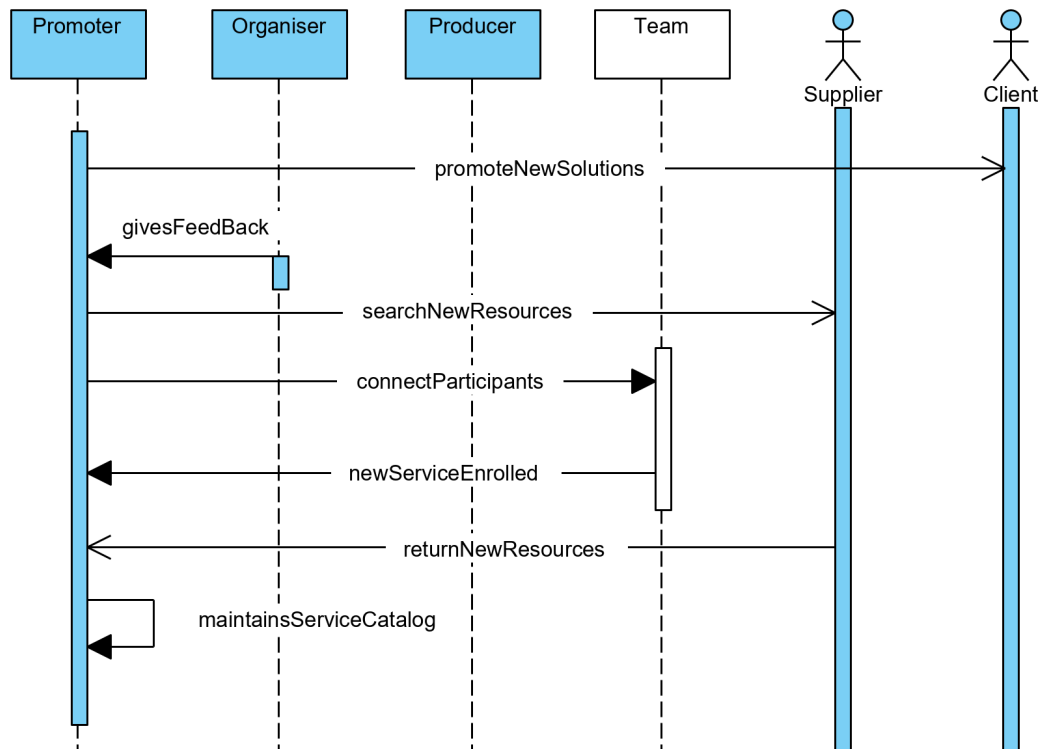


Figure 4.7: The Sequence Diagram for the Promoter role.

Name :	The producer role works on a systematic way to produce work outputs.
Type :	Operational and teamwork role
Problem:	Better management of doers of the system.
Context:	Composition process
Primary behaviors of the role:	ADH, AUT, DLG, KCT, KNL, COO, COL, COM, DM, POL
Forces:	Monitor, Organiser, Promoter
Proposed Solution:	Producer services of the system should be empowered to give feedback about their usage in order to adopt the measures necessary to ensure the proper functioning.
Resulting Context:	Potential unused services inform about the case and they are promoted to ensure their proper functioning while overloaded services also inform to be treated by delegating work to other producers that are found inside or outside of the system.
Design Rationale:	The doers of the system needs to be included in the team as a role for having better perception of behaviors of other doers, to perform delegation for example, and the team in whole.

Here are the steps that occur from the composition process where the producer role is involved.

- It informs the organiser about under using or if they exceed workload capacity.
- It gives feedback to the promoter to promote their usage.
- It is invoked by the developer role to participate in a case.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the producer that follows in Figure 4.8.

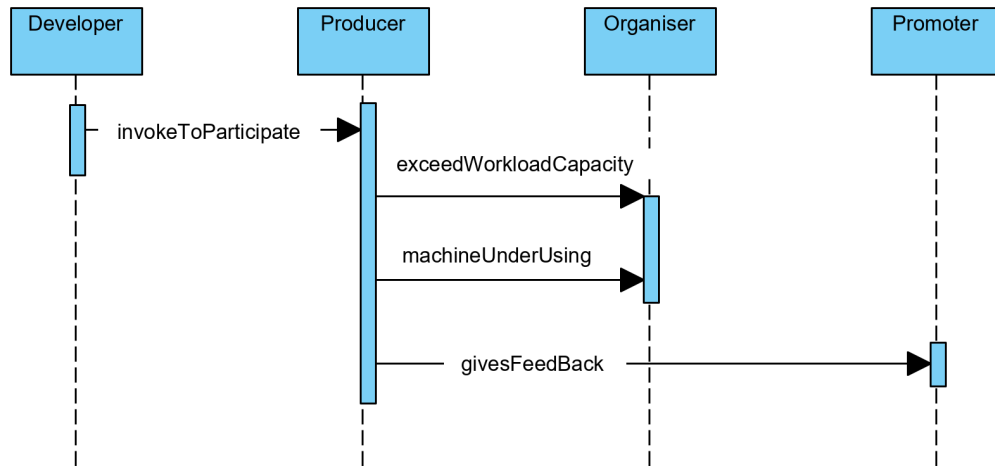


Figure 4.8: The Sequence Diagram for the Producer role.

4.2.11 The Maintainer role.

A service is designed, invoked, updated and maintained by peer services. Each service in the domain is based on services play the maintainer role for their maintenance. The maintainer role except maintaining services also maintains the team functionality. The role domain through this role can provide composite teamwork services with built in integrity checking. It receives queries for maintenance of services, e.g. manufacturing services that have maintenance errors. If resources inside the domain cannot overcome the maintenance problem, it informs the promoter role to search for maintainers outside of the domain. It also keeps the configuration of the system. Each member of the domain must remain in sync with the current state of the domain. It is the job of the maintainer to come to a consensus on the contents of each update to the plan. This ensures that each individual peer of the composite service is informed for each change took place before the execution.

Role Pattern Template for the Maintainer Teamwork Service Role

Name: The maintainer role maintains the functionality of the team.

Type:	Teamwork service role
Problem:	To develop services with built in integrity checking and maintaining.
Context:	Composition process
Primary behaviors of the role:	ADH, KCT, KNL, COO, COL, COM, DM, POL, PLN
Forces:	Organiser, Promoter, Monitor, services participating in the team
Proposed Solution:	The maintainer role perform either regular maintenances or urgent maintenances upon to other roles notifications.
Resulting Context:	All planned and urgent maintenance actions take place from inside of the system. They are either in schedule since the maintainer role keeps track or the maintenance role is informed by the organiser, the monitor or the service itself for an urgent maintenance.
Design Rationale:	In order to have a built in maintaining process the solution is to include in the team a role that will be responsible for this. This way the team can react and allow the system to get away from maintenance problems instead of forcing the system be out of order due to a maintenance demand.

Here are the steps that occur from the composition process where the maintainer role is involved.

- Receive queries for maintenance of services.
- Schedule the maintenance of services.
- Maintains peer services.
- Maintains the team functionality.
- Informs the promoter role to search for maintainers outside of the domain.
- It keeps the configuration of the system.

From each of these steps, we can specify what messages should be exchanged to perform the composition and draw the sequence diagram of the maintainer that follows in Figure 4.9.

4.3 Teamwork oriented Composition in Role Domains

4.3.1 Description of the Composition using Teamwork Roles

Upon to a user request for a service, which is expressed as a goal statement, the involved advisor role of the domain takes the request and searches if it matches according

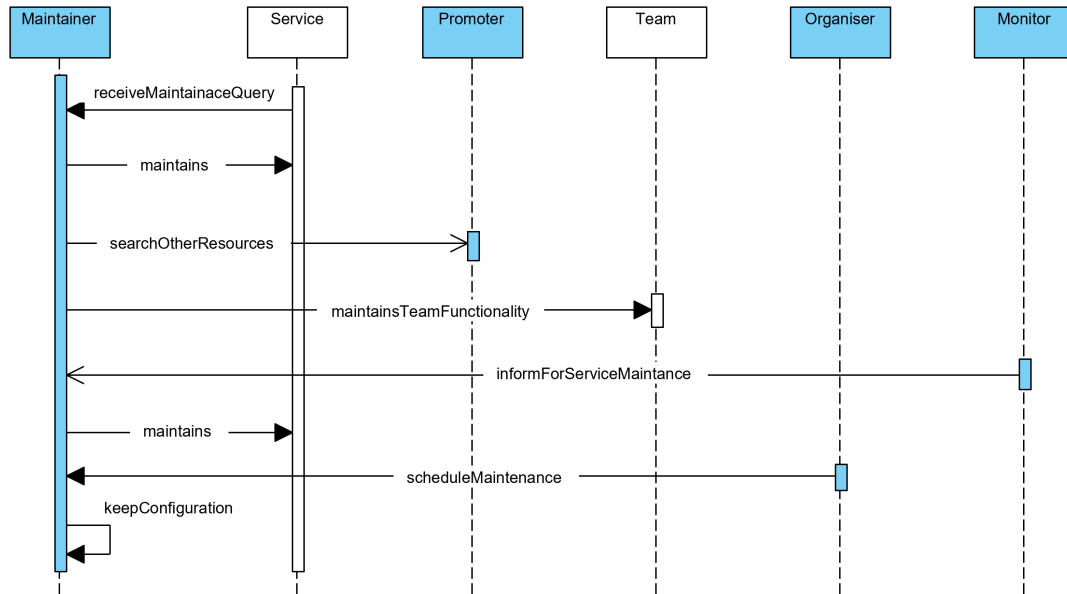


Figure 4.9: The Sequence Diagram for the Maintainer role.

to its semantic similarity with existing business goals of the plan library of the domain repository. Goals in the repository are linked with predefined plans which are predesigned business processes that can be achieved by an atomic or composite service execution. If it is so, it retrieves all the suitable solutions, which are expressed as sets of sequence of behaviors to be executed. Then the adviser role makes an evaluation to find the most appropriate plan according to the current situation of the domain and informs the developer role, about the plan to follow in order to continue with the service allocation and the team formation. If it does not exist a predefined plan, it informs the developer to search for a new plan of behaviors which when they are executed sequentially they will lead to the goal situation. The developer role search for new plans and sends the results for evaluation. Then it receives the sequence of behaviors of the selected plan to be executed and according to the RM of the domain looks up for roles that can execute these behaviors. It finds the appropriate services that can play these roles and makes the service allocation for the execution of each behavior which actually would be the team to operate.

During the allocation, if a service is unavailable the developer role proceeds with another candidate service allocation. Since many candidate web services are published in the web and enrolled in the domain to play a role and implement a behavior, the number of possible behavior allocations is very large and needs to narrow the results of the candidate services to be included as potential participants to the composition. Moreover, if we take into consideration the different quality attributes of each service, (as the result of the composition plans) the instants of composite services may be produced are quite different from each other. The developer role actually makes the team formation since it makes

the service allocation to services that play operational roles of the domain, i.e. accounting role, loader role, e.t.c. After the service allocation is completed, the developer role calls the inspector role to monitor how the team of selected services is proceeding with the composition.

4.3.2 Integrated SD for the Composition with Teamwork Roles

In Figure 4.10, we present the integrated SD where all team roles participate for the service composition. As we notice, it is a decentralized SD that uses primary means of teamwork for structuring communications among multiple participants. More specifically, Jacobson et al. [Rumbaugh et al., 2004] describes a decentralized SD as one where “each object only knows a few of the other objects and knows which objects can help with a specific behavior. Here we have no ‘central’ object.” Apparently, this is an evidence that the same idea lies behind teamwork role modeling composition of services, no central service exists to control the composition.

4.4 Conclusions

Robots, services and other intelligent systems are increasingly used in modern environments and this evolution was widely possible after the dissemination of cloud-based technologies and the everything-as-a-service approaches. Researchers are looking for ways to help humans work better with their new teammates (Kyrarini et al., 2016). In such environments, teams composed of humans, agents, services and other entities requires teamwork abilities to solve complex problems. Team research that is not directly applied to service oriented domains is potentially useful for developing intelligent teams, whose members could behave as human teammates do. Research involved in this chapter is conducted within human and agent based teamwork and is targeted at coupling human based behavior and agent based behavior to enrich services with supportive behavior which is related to collaborative capability. Besides, the collective intelligence paradigm [Rosenberg, 2019] has changed the way groups operate, similarly could affect the way service composition is done.

Composite services that, except their operational behaviour, also have a teamwork behavior are not just a concrete set of source code that will be put into software that develop the service, like Java libraries or frameworks. Teamwork roles are more conceptual. It is gained knowledge that domain experts can apply within service composition to guide its structure, and make it more efficient, flexible and reuseable.

Trying to compose multiple services, in a way, is similar of playing a game of chess. There are many ways to win by putting your opponent in checkmate. Accordingly, a beginner member of a domain may only know the basic patterns of services provided to achieve the composition whereas an expert could look for more appropriate patterns for the composition leading always to the same goal. There are established commonly asked goals that can be served by specific patterns, that for example follow common conventions. And these are patterns that teamwork services will look to exploit while executing the service in a particular situation. Through experience, a team of services instead of a group of services is better able to judge which patterns to use or create in a particular situation to compete a successful composition.

During service consumption, the system often comes across the same problems many times. There are so many ways to deal with these recurring problems, but over time some solutions are preferred over others, because they're more flexible or reusable. Though a survey of established patterns to such problems the domain can gain a foundation for more complex composition solutions. Teamwork roles provided through role patterns are practical solutions to these problems.

In sum, the main goal was to survey existing major models and frameworks to bring

together theories and models within the research literature, on both human based teamwork and agent based teamwork. From the survey, an integrated table of major human roles that gives an holistic view of the current models on humans and a core set of agent team work factors are derived. The obtain results revealed the indicative behaviors which can be used in order to obtain the teamwork abilities needed. Our goal is to define the analytical primitive behaviors that should be extracted for each team role containing the necessary key factors according to the dimensions emerged for effective teamwork. This constitutes for services a definition of their motivational attitudes and a specification of their evolution in dynamic environments.

In this chapter, we discussed the design and some implementation details of a teamwork model to address the challenges for teamwork composition of services in role domains. The results are a step towards the definition of a model that will enable augmenting services to more intelligent ones, so that to interact with each other autonomously, to crawl within their domain to discover other trusted services and devices, to delegate their assignments to other similar trustworthy services, and state explicitly their capabilities within their domain.

References

- [Agmon and Stone, 2011] Agmon, N. and Stone, P. (2011). Leading multiple ad hoc teammates in joint action settings. In *Interactive Decision Theory and Game Theory*.
- [Balachandar et al., 2019] Balachandar, N., Dieter, J., and Ramachandran, G. S. (2019). Collaboration of ai agents via cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1907.00327*.
- [Barange et al., 2014] Barange, M., Kabil, A., De Keukelaere, C., and Chevaillier, P. (2014). Task-oriented conversational behavior of agents for collaboration in human-agent teamwork. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 25–37. Springer.
- [Barrett et al., 2011] Barrett, S., Stone, P., and Kraus, S. (2011). Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 567–574. International Foundation for Autonomous Agents and Multiagent Systems.

- [Belbin, 1993] Belbin, R. M. (1993). Team roles at work: A strategy for human resource management. *zitiert in: Teamarbeit und Teamentwicklung*, page 321.
- [Bradshaw et al., 2004] Bradshaw, J. M., Acquisti, A., Allen, J., Breedy, M. R., Bunch, L., Chambers, N., Feltovich, P., Galescu, L., Goodrich, M. A., Jeffers, R., et al. (2004). Teamwork-centered autonomy for extended human-agent interaction in space applications. In *AAAI 2004 Spring Symposium*, pages 22–24.
- [Bradshaw et al., 2008] Bradshaw, J. M., Feltovich, P. J., Johnson, M. J., Bunch, L., Breedy, M. R., Eskridge, T., Jung, H., Lott, J., and Uszok, A. (2008). Coordination in human-agent-robot teamwork. In *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*, pages 467–476. IEEE.
- [Chen and Barnes, 2013] Chen, J. Y. and Barnes, M. J. (2013). Human-agent teaming for multi-robot control: A literature review. Technical report, ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD.
- [Cohen and Bailey, 1997] Cohen, S. G. and Bailey, D. E. (1997). What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of management*, 23(3):239–290.
- [Dunin-Keplicz and Verbrugge, 2011] Dunin-Keplicz, B. and Verbrugge, R. (2011). *Teamwork in multi-agent systems: A formal approach*, volume 21. John Wiley & Sons.
- [Fan et al., 2006] Fan, X., Sun, B., Sun, S., McNeese, M., and Yen, J. (2006). Rpd-enabled agents teaming with humans for multi-context decision making. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 34–41. ACM.
- [Fan and Yen, 2004] Fan, X. and Yen, J. (2004). Modeling and simulating human teamwork behaviors using intelligent agents. *Physics of life reviews*, 1(3):173–201.
- [Gamma, 1995] Gamma, E. (1995). Helm. r., johnson, r., vlissides, j.: Design patterns: elements of reusable object-oriented software. *Addison Wesley Longman, Inc, January*, 1(5):1.
- [Genter et al., 2011] Genter, K. L., Agmon, N., and Stone, P. (2011). Role-based ad hoc teamwork. In *Plan, Activity, and Intent Recognition*.
- [Grant et al., 2005] Grant, J., Kraus, S., and Perlis, D. (2005). A logic-based model of intention formation and action for multi-agent subcontracting. *Artificial Intelligence*, 163(2):163–201.

- [Hanna and Richards, 2013] Hanna, N. and Richards, D. (2013). A collaborative activity for evaluating hat-com: human-agent teamwork communication model. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1369–1370. International Foundation for Autonomous Agents and Multiagent Systems.
- [Harbers et al., 2012] Harbers, M., Bradshaw, J. M., Johnson, M., Feltovich, P., Van Den Bosch, K., and Meyer, J.-J. (2012). Explanation in human-agent teamwork. In *Coordination, Organizations, Institutions, and Norms in Agent System VII*, pages 21–37. Springer.
- [Jonker et al., 2011] Jonker, C. M., van Riemsdijk, M. B., and Vermeulen, B. (2011). *Shared Mental Models*, pages 132–151. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Kaminka and Frenkel, 2005] Kaminka, G. A. and Frenkel, I. (2005). Flexible teamwork in behavior-based robots. In *Proceedings Of The National Conference On Artificial Intelligence*, volume 20, page 108. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- [Kennedy et al., 2008] Kennedy, W. G., Bugajska, M. D., Adams, W., Schultz, A. C., and Trafton, J. G. (2008). Incorporating mental simulation for a more effective robotic teammate. In *AAAI*, volume 8, pages 1300–1305.
- [Korsah et al., 2012] Korsah, G. A., Kannan, B., Browning, B., Stentz, A., and Dias, M. B. (2012). xbots: An approach to generating and executing optimal multi-robot plans with cross-schedule dependencies. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 115–122. IEEE.
- [Kyrarini et al., 2016] Kyrarini, M., Leu, A., Ristić-Durrant, D., Gräser, A., Jackowski, A., Gebhard, M., Nelles, J., Bröhl, C., Brandl, C., Mertens, A., et al. (2016). Human-robot synergy for cooperative robots. *Facta Universitatis, Series: Automatic Control and Robotics*, 15(3):187–204.
- [Leng et al., 2006] Leng, J., Fyfe, C., and Jain, L. (2006). Teamwork and simulation in hybrid cognitive architecture. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 472–478. Springer.
- [Liemhetcharat and Veloso, 2013] Liemhetcharat, S. and Veloso, M. (2013). Synergy graphs for configuring robot team members. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 111–118. International Foundation for Autonomous Agents and Multiagent Systems.

- [Margerison and McCann, 1990] Margerison, C. J. and McCann, D. (1990). *Team management: Practical new approaches*. Mercury Books.
- [Mathieu et al., 2008] Mathieu, J., Maynard, M. T., Rapp, T., and Gilson, L. (2008). Team Effectiveness 1997-2007: A Review of Recent Advancements and a Glimpse Into the Future. *Journal of Management*, 34(3):410–476.
- [Nacer and Aissani, 2014] Nacer, H. and Aissani, D. (2014). Semantic web services: Standards, applications, challenges and solutions. *Journal of Network and Computer Applications*, 44:134–151.
- [Poslad et al., 2000] Poslad, S., Buckle, P., and Hadingham, R. (2000). The fipa-os agent platform: Open source for open standards. In *proceedings of the 5th international conference and exhibition on the practical application of intelligent agents and multi-agents*, volume 355, page 368.
- [Rosenberg, 2019] Rosenberg, L. B. (2019). Methods for analyzing decisions made by real-time collective intelligence systems. US Patent 10,222,961.
- [Rumbaugh et al., 2004] Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *Unified modeling language reference manual, the*. Pearson Higher Education.
- [Saidani and Nurcan, 2006] Saidani, O. and Nurcan, S. (2006). A role-based approach for modeling flexible business processes. *BPMDs*, 6:111–120.
- [Scholtz et al., 2004] Scholtz, J., Young, J., Drury, J. L., and Yanco, H. A. (2004). Evaluation of human-robot interaction awareness in search and rescue. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2327–2332. IEEE.
- [Schwartz et al., 2016] Schwartz, T., Feld, M., Burckert, C., Dimitrov, S., Folz, J., Hutter, D., Hevesi, P., Kiefer, B., Krieger, H. U., Luth, C., Mrona, D., Pirkl, G., Rofer, T., Spieldenner, T., Wirkus, M., Zinnikus, I., and Straube, S. (2016). Hybrid teams of humans, robots, and virtual agents in a production setting. *Proceedings - 12th International Conference on Intelligent Environments, IE 2016*, pages 234–237.
- [Skubch et al., 2010] Skubch, H., Wagner, M., Reichle, R., Triller, S., and Geihs, K. (2010). Towards a comprehensive teamwork model for highly dynamic domains. In *ICAART (2)*, pages 121–127.
- [Stone et al., 2010] Stone, P., Kaminka, G. A., Kraus, S., Rosenschein, J. S., et al. (2010). Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI*.

- [Sukhbaatar et al., 2016] Sukhbaatar, S., Fergus, R., et al. (2016). Learning multiagent communication with backpropagation. In *Advances in neural information processing systems*, pages 2244–2252.
- [Tsoutsas et al., 2019] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2019). Teamwork behavior: A review to interconnect industry 4.0 entities. In *Technological Developments in Industry 4.0 for Business Applications*, pages 1–25. IGI Global.
- [Van Wissen et al., 2012] Van Wissen, A., Gal, Y., Kamphorst, B., and Dignum, M. (2012). Human-agent teamwork in dynamic environments. *Computers in Human Behavior*, 28(1):23–33.
- [Wu et al., 2011] Wu, F., Zilberstein, S., and Chen, X. (2011). Online planning for ad hoc autonomous agent teams. In *IJCAI*, pages 439–445.
- [Yen et al., 2006] Yen, J., Fan, X., Sun, S., Hanratty, T., and Dumer, J. (2006). Agents with shared mental models for enhancing team decision makings. *Decision Support Systems*, 41(3):634–653.
- [Zeshan and Mohamad, 2011] Zeshan, F. and Mohamad, R. (2011). Semantic web service composition approaches: overview and limitations. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 1(3):640–651.

Μία μέλισσα μέλι ου ποιεί.
Αρχαιοελληνική παροιμία.
One bee cannot make honey.
Ancient greek adage.

Chapter 5

An Ontology for Teamwork Enabled Services

Contents

5.1 Introduction	127
5.2 Foundations for the TrEW Service Ontology	130
5.2.1 The core Competencies for the Ontology	130
5.2.2 The Content Ontologies	132
5.3 TrEW Service Ontology Construction	139
5.3.1 An Ontological Specification of Teamwork Enabled Services . . .	139
5.3.2 Rules within a Role Domain	140
5.3.3 Restrictions within a Role Domain	140
5.4 Ontology implementation	141
5.5 Discussion	142

5.1 Introduction

This chapter is dedicated to the conceptual formal modeling of service teamworking by considering the concepts defined within the vast range of models and theories of human and agent team development. The aim of the formalization is to be used for the teamwork based role modeling composition of services.

A necessary condition for services, especially those activate in a dynamic environment, is to essentially have the same abstract description, so that information about their usage

can be uniformly extracted and new composite services from existing components can be built. As already mentioned in the second chapter, by default, standard description languages like WSDL do not include semantics in the description of Web services. Different services could have similar WSDL descriptions while the result of their invocation could have been different. Such ambiguities can be resolved by giving semantic meaning to the invocation of a service, which is essential for the automation of service composition. Moreover, existing conceptualizations of team concepts are informal and thus cannot avoid ambiguities even in the definition of roles themselves. We share the motivation of OWL-S [Martin et al., 2007] and WSMO, that descriptions of Web services should be formulated according to an ontology in order to support the automation of service related task. But, while the aforementioned define service related concepts in relation to each other by using an ontology, they lack the formal semantics to relate these concepts to the basic categories of teamwork theory.

Regarding ontologies similar to our work, we evidenced the existence of ontologies for coalitions of humans and agents that is already developed in [Bradshaw et al., 2012] covering specific issues though having different foci. The authors present how they implement the human-machine coalitions in Sol, a framework for cyber operations. The research focuses in addressing the challenges of flexible and uniform policy management in complex military network operations. Based on the KAoS Policy Services framework [Uszok et al., 2011], it offers an ontology-based approach to policy representation and reasoning. The basic elements of the KAoS architecture are concepts from three layers of functionality that correspond to three different policy groups i.e. (i) Human Interface Layer, (ii) Policy Management Layer and (iii) Policy Monitoring and Enforcement Layer.

We also verified that few ontologies have conceptualized the relationships between humans and agents, that is, ontologies oriented to describe knowledge of human-agent teamwork [van Diggelen et al., 2018], [Guzzoni et al., 2007], [Lemaignan et al., 2017]. But these ontologies only partially cover the aspects related to the role modeling teamwork approach for service composition for services. The aspects that characterize the service roles as they are described in Chapter 4, are not comprehensively included in any of the ontological models and vocabularies previously described. The majority of the proposed ontologies cover aspects associated with interactions between agents and between agents and humans. This has provided the basis for the development of the proposed ontology having a particular conceptualization for the description of the entities, elements, relationships and mechanisms necessary for the proposed model.

Since, there is no previous attempt to formalize teamwork in the area of services, thus concepts are ambiguous, terminology is confounding, and important concepts are not defined. Consequently, a conceptual formalization is needed and the initiative was taken

to create the conceptual based modeling by using ontologies to define teams, roles, services and their modeling domains abstracting the most important concepts of team development whilst providing a decent level of formality and unambiguity. The attempt provides a concept schema for team development of teamwork enabled services by updating and extending the role modeling foundational ontology, which has been partially developed [Kritsimalis, 2010].

The ontology, named TrEWSOnto, is proposed as a model to assist the development process of teamwork enabled services in tandem with the representation of their composition process activity. This ontology abstracts specific roles, behaviors and situations from proposed theories. It describes the general concepts that can be used across domains and operate in tandem as unambiguous computer interpretable specification. More specifically it includes (i) the definition of concepts that are used to improve the teamwork ability of peer services, (ii) the description of the main concepts are used for the role modeling composition, and (iii) the description of situations that teamwork roles should monitor to catch potential “unhealthy” behavior happen during the service activity in order to run proactively and avoid obstacles or collisions. The result is a structure of five ontology sections together forming a representation for TeamwoRk Enabled Web (TrEW) Services and the environment they live.

This chapter’s contribution is a formal model for service teamworking, which shows its usefulness in conceptual modelling by exploiting its main characteristics regarding teamwork in groups of services: expressiveness ability, semantic power and unambiguity. The ontology’s evaluation is based on scenarios and competency questions which verify the coverage and consistency. It is available to be used for testing and evaluation by a framework for a smart city case we present in next chapter. The resulting ontology is expected to support the design of more specific ontologies that will detail the specific domain needs. Our proposal is meant to be useful for service developing that wishes to formalize knowledge in a domain, their systems, methods and techniques and enable the development of TrEW services.

In the following sections we describe the initial version of the ontology for teamwork enabled services, the TrEWSOnto, as a computational model that forms the basis to meet the teamwork requirements identified. The TrEWSOnto is OWL-based and is devised to be used as an extensible model for service composition by using role modeling whith its foundation to become from teamwork theory, service and composition role modeling.

5.2 Foundations for the TrEW Service Ontology

The ontology is created for developing teamwork enabled services is a structured class of concepts that constitute the subject of teamwork, i.e. the content teamwork roles of the product service and how these are related to service concepts and the composition procedure. The creation process of an ontology can be done through the following activities: planning, requirements specification, knowledge capture, delivery, integration, implementation, development, documentation, and maintenance. With this in mind and taking into consideration the methodology proposed in [Suárez-Figueroa et al., 2012] we constructed a network of ontologies, in order for different resources to be managed by different people or systems, possibly in different organizations. The specific network of ontologies can be understood as an upper ontology that formalizes the most fundamental concepts which are required to be modeled of our teamwork oriented composition model, that is used to structure and understand all concepts used concerning teamwork and services. This includes concepts such as service, behavior, plan, goal or participant. Such concepts are formalized by reusing the modelling basis of role modeling. For this reason, we separate the fundamental concepts in five content ontologies to facilitate reuse.

TrEWSOnto was developed using the Ontology Web Language and the OWL DL language that is based on Description Logics, which is a decidable fragment of First Order Logic. During the requirements analysis process, we considered (i) concepts, methodology and the representation of role modeling composition of Web services and (ii) concepts defined within plurality of the models and theories of team development in order to enrich the ontology with teamwork roles that derived from the literature research on human and agent teamworking [Tsoutsas et al., 2019]. Typically, we created the definitions of the concepts in the hierarchy and then continued by describing properties of these concepts.

5.2.1 The core Competencies for the Ontology

The capital question to answer for the creation of this new ontology is why and what we are talking about. Its creation will extend the role model ontology and in order to model services as roles and use teamwork roles to follow either predesigned or new designed patterns for service collaboration. We want to talk about this because we want to formalize the service teamworking in order to make sure that the development of a composite service will succeed and no errors are inadvertently being introduced during the process of service composition. The application domain and the universe of Web services will act as the base for the reusable knowledge that the ontology will find, which is the objective for the ontology creation. The problem space of the ontology to be created are the behaviors of teamwork roles to be specified that services could play during service composition. It will

serve team purposes for a composition, e.g. finding participants (other services, agents or persons) that could play the teamwork role needed for the specific topic, matching project topics to participant competencies, resources required, available time, etc.

The five ontologies we developed to formally represent knowledge about teamwork and services define concepts with logic; the expressiveness and precision of these structures can support computational reasoning in order to gain new knowledge. One of the ways to evaluate the scope of the ontology is to sketch a list of questions that the ontology should be able to answer, the competency questions [Wisniewski and Ławrynowicz, 2019]. These questions will serve as the test to examine if the ontology contains enough information to answer these types of questions. In the teamwork ontology, we aim for the following to be some possible example competency questions.

- Is the composed service a teamwork enabled one?
- Which are the teamwork roles participating in a specific composite service?
- Does the composite service contain a specific teamwork behavior, e.g. collaboration, planning?
- Find all the goals of the domain that are subsumed by a given request for a service.
- Can a client's request be served by enlisted services?
- Present complete description for a composite service, each component and potential provider.
- What are the plans that satisfy a specific goal?
- Which services can participate in a specific team according to their behavior?
- Are unengaged services available to serve that plan?
- Which services that are able to play an organiser role remain unengaged in the domain?
- What service that has a specific goal also plays the monitor role?
- Which are the failed plan executions that a specific service has been involved?

This list of questions is only indicative. Judging from these, our aim is to reflect the information that the ontology will include and reason on which derive from roles and behaviors of services that are used for the composition process.

5.2.2 The Content Ontologies

Five ontologies have been created and each one intends to capture the knowledge for a specific domain which is required for the modeling of TrEW services. The developed ontology modules are: RoleModel, Team, Sensing, Service and RoleDomain Ontology. The prefixes **rm**, **tm**, **sen**, **ser** and **rdm** refer to the URIs of the aforementioned ontologies, respectively. In Table 5.1 we give a brief description for each ontology while in the following sections we give details about classes, properties and its axiomatization for each individual ontology.

Table 5.1: The five ontology modules of TrEW services.

Ontology module name	Description
RoleModel	The RoleModel ontology is the one that represents the role model of a domain.
Team	The Team Ontology defines the concepts related with the teamwork aspect of the model. For each teamwork role they are identified the potential risks and vulnerabilities in its area of competence.
Sensing	This ontology describes and factors the sensors, actuators and observations that take place in the domain.
RoleDomain	The domain of interest, having the hierarchy of concepts, properties and rules related with the context of a domain is modeled in this ontology.
Service	The Service Ontology to define teamwork enabled services

A fundamental entity of the ontology is the role and is extended to describe the teamwork roles needed for the service composition. But the teamwork ontology is not just a controlled terminology. We have to capture the conceptual schema underlying the use of certain team roles, in order to make them reusable for design, interoperability, meaning negotiation, etc. It will become the foundation for simple service interaction, the reference conceptual schemas that will go beyond the predefined patterns of interaction concerning the operational part of the communication.

The RoleModel Ontology

We further introduce the minimal notions necessary for representing composition, based on the role modeling approach of the composition process, as they were described in

Chapter 3. The RoleModel ontology is the one that represents the role model of a domain, i.e. the content available roles that can be used to represent services and their behaviors. With this extended ontology we provide precise representation for the following concepts and properties:

- **rm:Role**: represents a class whose individuals are the roles that a participant of the team could play, e.g. A Client, a Supplier. All roles that appear in the role model of the domain are stored in this entity.

$$\text{rm:Client} \sqsubseteq \text{rm:Role}$$

- **rm:ComAct**: This class stores the information about the communicative actions a role can use to communicate with another role. The notion of the communicative action is borrowed from the Speech Acts which are used in Agent Communication Languages (FIPA ACL, KQML). Instances of this concept are a subset of those already defined extended by new ones thought as appropriate for the description of the domain service interactions.
- **rm:Behavior**: This is the fundamental class by which it could be defined for each role with which other roles it communicates. This class is exploited by the role class since a role is defined as a set of behaviors indicating its ability to communicate with others.
- **rm:Precondition**: It is a class that describes the preconditions that must hold in order for a behavior to be executed. A logical language is used in order to write sentences that describe preconditions.

$$\text{rm:Precondition} \sqsubseteq \text{lg:LogicalSentence}$$

- **rm:Effect**: Respectively, this class describes the effects of the behavior execution. Both precondition and effect are subclasses of the LogicalSentence class which corresponds to sentences from a logic language.

$$\text{rm:Effect} \sqsubseteq \text{lg:LogicalSentence}$$

- **rm:ActingRole**: models a role which executes a behavior to a peer role.

$$\text{ActingRole} \sqsubseteq \text{rm:Role} \sqcap \exists \text{rm:executesBehaviorTo}.\text{rm:ConfereeRole}.$$

- **rm:ConfereeRole**: as its name indicates, it models a role which receives a behavior specifically coming from an initiator the communication role.

$\text{ConfereeRole} \sqsubseteq \text{rm:Role} \sqcap \exists \text{rm:receivesBehaviorFrom.rm:ActingRole}.$

- **rm:hasBehavior:** The behavior that a role has. Furthermore, this property subsumes a more specialized property called **tm:hasTeamBehavior**, and is defined to subsume both the operational and teamwork behaviors.
- **rm:executesBehaviorTo:** provides the relation between two peer roles, the sender and the receiver of a behavior.

It is also the inverse of the property **rm:receivesBehaviorFrom**.

$\text{rm:receivesBehaviorFrom} \equiv \text{rm:executesBehaviorTo}^{-}$

- **rm:hasPrecondition:** This property indicates that the a behavior has a precondition which must hold in order to be executed.
- **rm:hasEffect:** Similarly this property indicates the effect of a behavior.

The Team Ontology

The Team Ontology aims to define the concepts related with the teamwork aspect of the model. The team is formed by role participants and its intent is to follow a plan that will achieve a specific goal. Some of the main classes of this ontology are following.

- **TMember:** Any member of the team that play a specific role. These members are mapped to subclasses of the role class and they describe specific behavior for each functional or teamwork role that is needed in the team like **Maintainer**, **Monitor**, **Promoter**, **Developer**, **Organiser**, **Innovator**, **Advisor**, **Producer**.
- **TGoal:** as its name indicates it models the goal of the team that is constituted for.

$\text{TeamGoal} \sqsubseteq \text{Situation} \sqcap \exists \text{hasEffect.Effect}$

- **TPlan:** The plan to follow in order to achieve the team goal. The plan is a sequence of role behaviors to be executed. After their execution, the team reaches the final state of the plan which is supposed to coincides with the team goal.
- **TRole:** This is the specialization of the class **rm:Role**. This class is designed to represent the teamwork roles participating in the team of services that are used for the composition process. For each teamwork role participate in the team, they are identified the potential risks and vulnerabilities in its area of competence.
- **TBehavior:** A teamwork behavior could only be executed by teamwork roles.

$$TBehavior \sqsubseteq rm:Behavior \sqcap \forall rm:hasBehavior^-.rm:TRole.$$

We make the following assumption about teamwork behaviors that are executed within a team: if a service is assigned to a teamwork role containing a teamwork behavior, then this behavior enables the service, when a given state of affair u is presented to evaluate for a given sentence $s \in S$ whether s is true or false in u according to the rules expressed for a specific teamwork behavior. This assumption allows us to reason when a teamwork role must intervene and try to overcome the obstacle of a specific situation.

- **TeamFactor:** A teamwork behavior in a role domain can be specialized and expressed by specific properties of factors of interest (i.e., collaboration, learning, etc.)

The Sensing Ontology

This ontology mainly relies on the SSN ontology [Compton et al., 2012] which describes sensors, actuators and observations. It is extended in order to be included in the role domain ontology, allowing us to describe the sensing process which provides information to the teamwork roles about the observation of a role domain in the context of composition of services.

A situation is a snapshot of the environment at a certain moment. To partially describe a state propositions are used. The initial situation, where no actions have taken place, is denoted by s_0 . A predefined function *State* maps a situation to the state of the environment in that situation.

- **Sensor:** This class is designed to represent any sensing device used for the observation process.

$$Satellite \sqsubseteq sen:Sensor$$

$$SmartPhone \sqsubseteq sen:Sensor$$

- **Observation:** This class represents a monitoring process that has been setup to observe the behavior of a feature of interest as this will be determined in the applied domain.
- **ConfigurationProcedure:** This class allows us to define specific configuration required for each sensor to be used in an observation process that can be defined according to the application domain. It will detail what property is observed and to whom is associated with.
- **EntailingSituation:** A situation that, according to the observations constitutes an indication that there is a need for a teamwork behavior to intervene. These

situations can either be produced by indications directly captured from sensor data or by reasoning over data that derived through other participant behaviors.¹

The Service Ontology

The Service Ontology defines teamwork enabled services composed by services that are enrolled in a domain that use the role modeling terminology. In this ontology we have extracted the following main concepts:

- **SName**: the identity of the service.
- **SRole**: is the specialization of the class `rm:Role`,

$$\text{sr:SRole} \sqsubseteq \text{rm:Role}.$$

The role is further specialized with teamwork roles and form teamwork enabled services that are able to be proactive during their participation in specific events.

- **SBehavior**: is the specialization of the class `rm:Behavior`

$$\text{sr:SBehavior} \sqsubseteq \text{rm:Behavior}$$

The RoleDomain Ontology

The RoleDomain Ontology is the global ontology that represents the domain of interest Δ , having the hierarchy of concepts, properties and rules related with the context of a domain that is modeled as a role domain. Δ hosts services that are defined using the role modeling approach. It combines all previously defined classes and it imports the DomainTopic Ontology, which defines a specific common set of terms, a common vocabulary consisting of various concepts, properties and rules related to the domain theme in order for the participants to be able to understand the exchanged information for the entire domain of discourse. By a domain of discourse, we understand a topic that one or more people may be interested in. Examples include market domains, smart city domains and industry domains. For a smart city domain for example, concepts like Transport, Energy, Museums and Resources are defined in DomainTopic or imported from a separate group of ontologies (e.g. CityConcepts.owl).

- **rdm:RoleModel**: This class is used to define the role model of a role domain which is constituted by roles and behaviors. The role model is represented as a set of collaborative roles. Each role is the partial observable behaviors that a team participant exhibits, i.e. the interactions in which the service is involved. The sender and the receiver of a behavior are two collaborative roles which interact according to the RoleModel ontology.

- **rdm:Participant**: The provided services

$\text{srv:SName} \sqsubseteq \text{dm:Participant}$

- **rdm:DomainGoal**: A range of sentences describing the goal a service could achieve, which are expressed according to the business objectives of the domain.
- **Plan**: Predefined plans that could be followed according to the role model of the domain to achieve the business objectives of the domain.
- **rdm:Query**: A client query for a specific service expressed as the preconditions and effects of the composite behavior to be executed.
- **rdm:Situation**: Situations are a way to characterize a realistic domain, they can describe the different states of affair that participants may come about in this domain and they are partially described using propositions about the domain. They are situations that the system could reach evolving from state to state. Some of them are partially predefined describing how the system evolves following the sequence of behavior execution or others are situations that a role should identify and intervene since they are partially described that indicate an “unhealthy” behavior of a service.

$\text{Precondition} \sqsubseteq \text{dm:Situation}$

$\text{Effect} \sqsubseteq \text{dm:Situation}$

$\text{DomainGoal} \sqsubseteq \text{dm:Situation} \sqcap \exists \text{rm:hasEffect.rm:TeamRole.}$

- **rdm:isUndertakenBy**: A situation in a role domain is more specialized and expressed by specific states the system enter and related to a teamwork factor or role of interest. The property **dmn:isUndertakenBy** associates a relevant situation with a teamwork behavior (i.e., the state that a machine has high temperature and the behavior of the maintainer role that should undertake the responsibility of the situation). ^{II}
- **rdm:DomainTopic**: A classification of concepts related with the topic of the domain. It is constituted by concepts describing the purpose, its processes and resources. ^{III}

Figure 5.1 displays an overview of the overall ontology in terms of its main classes and their relations. The figure is somewhat simplified in order to be more readable, only a restricted set of classes and relations was chosen to be displayed of each subontology.



5.3 TrEW Service Ontology Construction

In this section, we show how a semantic role domain is constructed by integrating the five separate ontology modules have been created representing different aspects of a semantic environment, i.e. the domain aspect, the teamwork aspect and the service aspect.

5.3.1 An Ontological Specification of Teamwork Enabled Services

A teamwork enabled service which is defined in a role domain is formed as the result of combining all the five aforementioned ontology that belong to a network of ontologies. The connection between each pair of ontologies is accomplished by specializing the definition of a concept in an ontology and then linking them together. For instance, the class *rm:Role* used in the role model ontology is further specialized via the link to the class *tm:TRole* in the team ontology. Some of the specialized relations between classes that are illustrated in Figure 5.1, are also listed in Table 5.2.

Table 5.2: Linked classes in TrEWS ontology.

Ontology A	Ontology B	Axiom
RoleDomain	RoleModel	$rdm : Participant \sqsubseteq \exists play.rm : Role$
Team	Sensing	$tm : TRole \sqsubseteq \exists interpret.sen : EntailingSituation$
Sensing	RoleDomain	$sen : EntailingSituation \sqsubseteq rdm : Situation$
Service	RoleModel	$ser : SRole \sqsubseteq rm : Role$
Team	RoleModel	$tm : TRole \sqsubseteq rm : Role$
		$tm : TBehavior \sqsubseteq rm : Behavior \sqcap \forall rm : hasBehavior \rightarrow .tm : TRole$
RoleDomain	Team	$rdm : Participant \sqsubseteq \exists as.tm : TBehavior$
		$tm : TMember \sqsubseteq rdm : Participant$

In the following, we list some axioms for the team ontology that illustrate the connections among classes for defining a service through the role domain terminology. The following axioms, for example, enable the system to classify a service, after it is submitted to a role domain. This uses automatic classification to locate the roles the service plays or the behavior it could execute within the team service ontology to achieve a goal. The intuitive meaning of the following is that a service plays a role, and a role has a behavior which has an effect that is described as a situation. Then, it is stated by the last three axioms that a teamwork plan leads to a goal which should be consistent with the domain goals

that is also described as a situation.

$$Service \sqsubseteq \exists play.Role$$

$$Role \sqsubseteq \exists hasBehavior.Behavior$$

$$Behavior \sqsubseteq \exists hasEffect.Effect$$

$$Effect \sqsubseteq Situation$$

$$Plan \sqsubseteq \exists leads.Goal$$

$$Goal \sqsubseteq \exists isAccordant.DomainGoal$$

$$DomainGoal \sqsubseteq \exists describe.Situation$$

5.3.2 Rules within a Role Domain

Below we specify some sample rules abiding in the suggested Role Domain. These rules ensure that deductions concerning specific properties of classes are made correctly.

IV

- A service is capable of achieving a goal if it plays a role having a behavior that its effect is that goal:

$$\begin{aligned} &Service(?x) \wedge Role(?z) \wedge play(?x; ?z) \wedge hasBehavior(?z; ?b) \wedge hasEffect(?b; ?y) \\ &\rightarrow hasGoal(?x; ?y) \end{aligned}$$

- A service is a teamwork enabled service if it plays a role that is a teamwork role:

$$Service(?x) \wedge TRole(?z) \wedge play(?x; ?z) \rightarrow TrEWSservice(?x)$$

5.3.3 Restrictions within a Role Domain

In order to establish communication among services some conventions need to be globally stated and agreed. We assume the following restrictions exist in a role domain.

- One role domain has one domain reference ontology.
- One or more service providers live in one role domain.
- Service providers collaborate (bind) with other service providers.
- Service providers exchange services.
- Service providers own/ have access to one or more services.

- Service providers have explicitly stated the domain goals, e.g. to extend their repository of services achieving specific goal, to help clients achieve their goals by collaborating with other service providers.
- Services are rated.

5.4 Ontology implementation

To implement the ontology, an ontology editor tool was employed. Protégé was the main tool used in this work as an ontology editing tool; it is a graphical tool for ontology construction developed at Stanford University (Stanford Center for Biomedical Research 2015). The reasoner was used to validate that models are consistent and to infer new information based on the analysis of an ontology. The TrEWSOnto is a proposal for conceptualization of an important field, whose concepts were always considered separately and ambiguously.

The sequence of steps used is as follows:

1. Create classes
2. Create properties
3. Add individuals and assign properties
4. Run Reasoner and validate subsumption testing

The ontology's concepts were defined as classes. Note that all classes are of subclass 'Thing'. Next, the object properties connecting the individuals between classes were defined. To denote ownership between the domain and range between classes, a convention of starting each object property with 'has' was employed, e.g., the object property connecting class 'Role' with class 'Behavior' was named, 'hasBehavior'. Next, the data properties for the individuals were defined. Here, given the range for the data property is a value, and not a class, the 'has' convention for the object properties was not employed. Next, each individual was assigned to their respective class type and its data property (as applicable). Finally, once all the individuals were defined, Protege's reasoner tool was employed. The reasoner is used to test whether or not one class is a subclass of another and if all concepts are correctly specified. Protege version 5.5.0 has many reasoner options. HermiT 1.4.3 [Shearer et al., 2008] is a Description Logics (DLs) reasoner that employs 'hyper-tableau' calculus to address problems due to nondeterminism and model size. The ontology was run through the reasoner and no errors merged.

5.5 Discussion

In this chapter, we continue shaping our own view for evolved services that we named TrEW services, those that could be developed through composition having teamwork abilities. In order to support their development, we laid an emphasis on designing and implementing an ontology. A relatively basic ontology on teamwork roles was created and linked it with the RM methodology that allows a more flexible approach to the service composition problem. The TrEWSOnto is intended to formalize the composition process, incorporating relevant concepts from teamwork. By using ontologies to define TrEW services, we do not only introduce sharable and reusable knowledge representation concerning the whole composition process, but also be able to add new knowledge about the domain. The TrEWSOnto is a proposal for conceptualization of an important field, whose concepts were always considered separately and ambiguously. Having an underlying ontology of teamwork for services is necessary for this purpose, thus we have put forth in the present chapter a basis for developing a detailed ontology of teamwork roles for services. Such roles, in a fully developed ontology for a specific domain may span across different responsibilities, but they are related within the same ontology and their relations can be uncovered and then captured in the ontology presenting the different business strategic that specific models follow.

The function of this ontology is to “encode” a description of the world with the teamwork roles that Semantic Web services could play through the situations of the domain they pass (actual, possible, impossible, desired, etc.) for some purpose of the composite Web service. TrEWSOnto is a proposal for conceptualization of an important field, whose concepts were always considered separately and ambiguously. We have created a relatively basic ontology focused on teamwork roles instead of listing team factors in general. The TrEWSOnto ontology could be exploited for the team roles and contribute to the service composition. It is extensible, human and machine readable, and the OWL source file will be available for research use. We have presented the development of the

core concepts of the ontology, as well as an indicative scenario that could be exploited.

References

- [Bradshaw et al., 2012] Bradshaw, J. M., Bunch, L., Eskridge, T., Feltovich, P. J., Hoffman, R. R., Johnson, M., Lott, J., and Carvalho, M. M. (2012). Human-agent teamwork for cyber sensemaking in network operations.
- [Compton et al., 2012] Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. (2012). The ssn ontology of the w3c semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25–32.
- [Guzzoni et al., 2007] Guzzoni, D., Baur, C., and Cheyer, A. (2007). Modeling human-agent interaction with active ontologies. In *AAAI Spring Symposium: Interaction Challenges for Intelligent Assistants*, pages 52–59.
- [Kritsimalis, 2010] Kritsimalis, A. (2010). Role modeling for cdl specifications of web services. *staffordshire university*.
- [Lemaignan et al., 2017] Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A., and Alami, R. (2017). Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence*, 247:45–69.
- [Martin et al., 2007] Martin, D., Burstein, M., Mcdermott, D., Mcilraith, S., Paolucci, M., Sycara, K., McGuinness, D. L., Sirin, E., and Srinivasan, N. (2007). Bringing semantics to web services with owl-s. *World Wide Web*, 10(3):243–277.
- [Shearer et al., 2008] Shearer, R., Motik, B., and Horrocks, I. (2008). Hermit: A highly-efficient owl reasoner. In *Owled*, volume 432, page 91.
- [Suárez-Figueroa et al., 2012] Suárez-Figueroa, M. C., Gómez-Pérez, A., and Fernández-López, M. (2012). The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer.
- [Tsoutsas et al., 2019] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2019). Teamwork behavior: A review to interconnect industry 4.0 entities. In *Technological Developments in Industry 4.0 for Business Applications*, pages 1–25. IGI Global.

- [Tsoutsas and Ragos, 2020] Tsoutsas, P. and Ragos, O. (2020). Towards an ontology for teamwork enabled services. In *Proceedings of the 2020 4th International Conference on Algorithms, Computing and Systems*, pages 69–75. ACM (ISBN: 978-1-4503-7732-4).
- [Uszok et al., 2011] Uszok, A., Bradshaw, J. M., Lott, J., Johnson, M., Breedy, M., Vignati, M., Whittaker, K., Jakubowski, K., Bowcock, J., and Apgard, D. (2011). Toward a flexible ontology-based policy approach for network operations using the kaos framework. In *2011-MILCOM 2011 Military Communications Conference*, pages 1108–1114. IEEE.
- [van Diggelen et al., 2018] van Diggelen, J., Neerincx, M., Peeters, M., and Schraagen, J. M. (2018). Developing effective and resilient human-agent teamwork using team design patterns. *IEEE intelligent systems*, 34(2):15–24.
- [Wisniewski and Ławrynowicz, 2019] Wisniewski, D. and Ławrynowicz, A. (2019). A tagger for glossary of terms extraction from ontology competency questions. In *European Semantic Web Conference*, pages 181–185. Springer.

Δεν είναι τα πιο δυνατά είδη που επιβιώνουν
ή τα πιο έξυπνα, αλλά αυτά που
ανταποκρίνονται καλύτερα στις αλλαγές.

Κάρολος Δαρβίνος,
1809-1882, Γεωλόγος, Βιολόγος.

It is not the strongest of the species
that survives, nor the most intelligent.
It is the one that is the
most adaptable to change.

Charles Darwin,
1809-1882, Geologist and Biologist.

Chapter 6

Teamwork Based Composition in Smart City Domains

Contents

6.1 Smart cities' Services Challenges	146
6.2 The «Bus Transportation» Use Case	148
6.3 Smart City Nexus Services	149
6.3.1 Smart and Sustainable City Architectures	152
6.3.2 The Teamwork Layer	153
6.4 A Computational Framework	156
6.5 Role Modeling of a Smart City Domain	159
6.6 Bus Transportation Representation	161
6.7 The TrEWS Population	164
6.8 Reasoning over Behavior Adaptation	166
6.9 Discussion	167

In this chapter of the thesis, we present and detail how to implement a use case by utilising the proposed composition model. This was mainly done to test its correctness, effectiveness and performance. The case was chosen to be from an evolving area, that of smart cities, since according to the market research, the number of smart cities is

increasing rapidly [Maddox, 2018]. This growth is expected to continue, smart cities have grown from approximately two dozen to more than 100, with more than 600 cities to work on getting smart in the next years [Bibri, 2018]. Certainly, it is not feasible to test the model in real smart cities since the application of such approach is a major project that cannot fit in the context of a PhD thesis. Alternatively, the model validation could be done by simulating specific service behaviour that has been done to a certain extend as it has also been presented to [Fitsilis et al., 2018], [Tsoutsas et al., 2020, Tsoutsas and Ragos, 2020].

6.1 Smart cities' Services Challenges

Some of the main challenges to model services, living in a smart city domain [Soto et al., 2015], [Novotný et al., 2014], [Balakrishna, 2012] are:

1. A smart city provides an intelligent way to manage components such as energy, transport, schools and buildings on their own, but what if these systems could interoperate in a domain designed to serve communication for specific purposes? Various devices and associated services for location, intelligent recognition, tracking, monitoring and management are available for the citizens to consume. Numerous IoT-based services are likely to be available and a key challenge is to allow mobile users perform their daily tasks dynamically, by integrating the services available in their city domain. But all the above are composed of heterogeneous services interacting in different ways.
2. A smart city is constituted of various infrastructure components that form a complex system of systems, which are expected to collaborate effectively. Information and communication technologies that provide the smart infrastructure are the foundation for all of the key themes related to a smart city, such as smart economy, smart governance, smart mobility, smart health, smart buildings, smart water, etc.
3. A smart city constitutes a dynamic system that continuously incorporates new services, which usually have a large variation in the way they perform the same task. Thus, services that collaborate alongside other services or systems for performing tasks, could not always follow the same predetermined plans for the composition and be adopted according to changes happened to the dynamic environment.
4. Architectures, which used to be operational in the near past, may have deficiencies and be inadequate for the future. For example, they are tuned for static service provisioning but not for dynamic service composition, they are controlled in most

cases centrally and they do not allow ad-hoc collaboration between services, they are constituted of smart city subsystems that are interconnected and integrated without exhibiting any signs of intelligent behavior (situation awareness, adaptive behavior), etc.

5. Since services live in a "smart" environment need to encompass the ability for the development of new *evolved* composite services having a level of intelligence throughout their composition in order to act in such an environment.

Through the proposed model for developing teamwork enabled services we aim to give solutions to the above challenges as following:

1. Services play a central role in this architecture, as they can be used as building blocks for effective collaboration, so they have to achieve interoperability between different parties and independence from the underlying infrastructure. Thus, such an architecture needs to be modeled as service oriented.
2. IoT objects and other resources of a smart city could be modelled as services since a service is an autonomous and loosely coupled unit of functionality. Role modeling of these services will help to overcome the complexity and heterogeneity problems.
3. Services need to be aware of either predetermined or other abrupt and unexpected behaviors in order to adapt them. We use roles, having teamwork behavior, to represent the composition procedure in such domains, and we model the team of services through the individual behavior of each participant as well as their group goal. Our proposal is to benefit from teamwork and, instead of having a set of services that form a group, to develop a team of services, i.e. a composite service having a teamwork attitude.
4. In order to overcome these deficiencies, in the sequel, we present the introduction of a new layer, which will provide teamwork functionality and could be utilized to solve some of the above-mentioned problems.
5. This imposes that smart cities should encompass the development of new composite services having a level of intelligence throughout their composition in order to establish reliable services. The teamwork roles we propose to participate in a team, like promoter, innovator etc. will enable services to meet the requirements a smart city may compel.

In order to keep control of composite services in a smart city complex environment, we propose to develop services that are able to act and interact independently and on

a demand basis. More analytically, the research in this chapter contributes to the smart cities area of services by (i) presenting an architecture that includes a teamwork software layer, (ii) developing the role model of part of the city domain, (iii) enhancing them by the basic teamwork behaviors needed, (iv) extending the ontology to include concepts and constraints regarding the city role domain, and (v) proposing a framework for the above to be implemented.

6.2 The «Bus Transportation» Use Case

A typical context from a smart city domain for showing the development of teamwork enabled services is that of city transportation. Some of the transport services that many cities provide are the following:

- bus route and timetable information,
- book and buy tickets,
- location aware services that get real-time location information to the commuter,
- get on and off transport alarms,
- get information about the details of the active means of transport a commuter enters,
- receive free alerts about expected bus arrival times for the interested buses and related routes via SMS and e-mails
- provide tracking system services that passengers can scan through QR (Quick Response) codes placed at bus stops to view estimated bus arrival times, buses' current locations, and bus routes

Let us consider the bus transportation of a smart city, which is provided to people who live within the boundaries of the city area. In terms of the context, citizens are consumers of services, while the municipality is the main provider who is imperative to use additional external services and reuse knowledge from past in order to ensure successful service launch and minimize unavailability of city services. The route that a bus follows for the commuters' transfer is calculated according to the distribution of commuter's residence. The information to design the paths is derived from the e-governance system and users can plan their trip accordingly. In fact, this is a solved problem and this kind of capability for traffic systems is included in city's information systems. Nevertheless, this only covers the extraction of a straightforward plan, which may give the shorter route for picking the commuters and transferring them to their destination.

However, suppose that, from last year's evaluation questionnaires, users state that, in specific areas, buses usually delay to pick up commuters whenever the weather is rainy. This is a remarkable fact, but it could not help officers improve the service, unless the system could get the extra information related to weather forecast in time for changing the bus path, or at least inform commuters in time about the delay. The municipal information system is very unlikely to store weather results, but there are plenty of weather forecast services in the web that they do. On its own, the traffic system used by the municipal simply calculates the daily commutes to a specific destination that each bus should follow. Nevertheless, in a smart city role domain, enclosing services of the above system with services that provide observations of traffic according to the data produced from existing cameras (IoT services) and, after combining them with weather forecast data, could predict the future in a way that expert participants (teamwork enabled) can act quickly to prevent possible delays or other dangers for bus routes.

We depict schematically in Figure 6.1 a potential daily route where a student's bus breaks down and another service needs to be employed in order for the students to get to school in time.

Following the description of the aforementioned scenario, we depict in Figure 6.2 the necessary actors/services, dividing them to those reflecting operational and those reflecting teamwork behavior that enable the adopted behavior to give solution to the specific problem has arisen and achieve the goal of tranfering students to their school. It is intended to provide the basis and avoided to define technical approaches and policies more than necessary. The vision is to establish an IT service choreography infrastructure that provides pertinent and timely transactions information between entities across the smart city domain. Claims and exception management can be automated, resulting in significant administrative efficiencies as well as lower cost and risk.

6.3 Smart City Nexus Services

Collaboration of smart infrastructure will provide the cognitive framework that smart cities subsystems need in order to have the ability to cooperate transparently and autonomously for offering composite, complex and more efficient services. We name this smart collaboration city infrastructure as the city nexus. However, developing smart city nexus is not straightforward since, according to [Vlacheas et al., 2013], while a wide set of predefined services and applications is available for employment in smart cities, there are technological barriers among objects when they are used across application domains.

We use the term *Smart City Nexus Services (SCNS)* to differentiate the typical composite services that could isolated be produced in the web, from compositions that

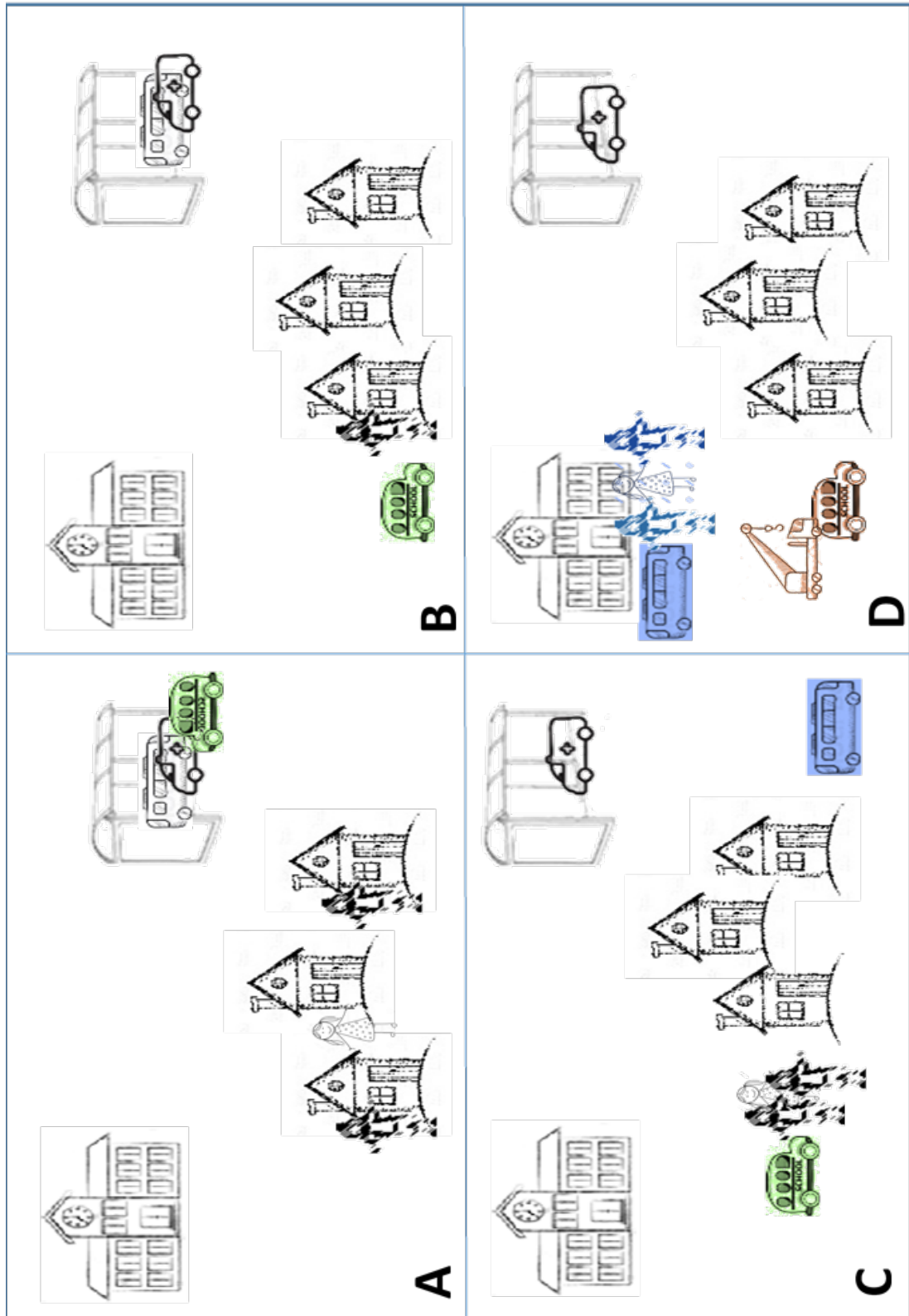


Figure 6.1: The bus transportation service recruited with services that exhibit teamwork behavior to enable the goal achievement. The presented model in four district states. (A) The initial situation, all students are at home. (B) A school bus on its route to the school. (C) After the bus brakes down, another bus will take students to school. (D) The goal situation achieved, all students are at school, the broken bus is going to get maintained.

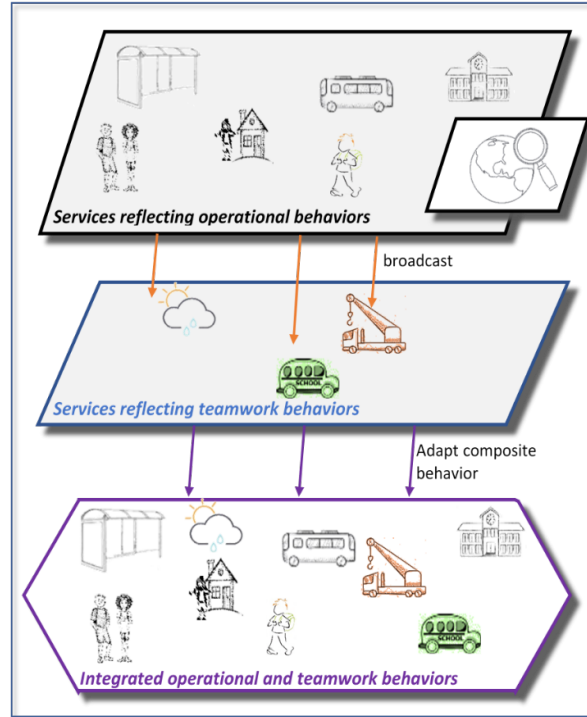


Figure 6.2: Integration of operational and teamwork behaviors.

take place in a role domain having a teamwork attitude that is specific oriented to the goals of the smart city domain. According to the Oxford dictionary nexus is “a complicated series of connections between different things”. The concept of nexus for modelling location aware software applications was used by other researchers but in different contexts [Ananthakrishnan et al., 2013]. In SCNS, the final product is a combination of operational and teamwork services that act as a glue to produce nexus services that aim to serve Smart City’s goals. They integrate resources of the domain software information systems, e.g. e-governance, with data of IoT services and other sophisticated tools in a sense that services forming these systems could also participate to play a critical role during the composition process. A SCNS has the ability to come up with the requested goal or another similar to the requested one. If a goal is not achievable using city’s own resources, it has to employ other services in achieving this goal. To successfully do so, the designed composite service needs to be enabled with one or more of teamwork behaviors.

To employ the composition model proposed in this thesis for the specific purpose, we exploit an existing smart city architecture [Focus Group Technical Specifications, 2015] and augment it with a teamwork layer that can introduce the notion of teamwork collaboration within a set of services, agents and other systems that “live” in a smart city forming the smart city nexus. This is important to be done, since smart cities are composed of a number of interacting subsystems and become more and more complex. Moreover, the number of offered services is increasing rapidly making the management of the smart city

cumbersome and costly. Therefore, there is a clear need to add autonomy, automation and self-organization features to smart cities services in a fashion that resembles human teamworking. To this end, we have developed a framework tool that has been designed to fulfill smart city requirements and supports engineers on semantic technologies in the creation and use of TrEW services.

6.3.1 Smart and Sustainable City Architectures

Smart cities are distributed-computing environments composed of a large number of distributed information systems based on architectures that enable the continuous evolution of smart cities ecosystems. The service composition process in a smart city is critical, since in many cases there are interrelationships between city's core systems, given that these systems cannot work in isolation. Interconnecting these systems obviously improves their efficiency and intelligence, since, they most probably operate in close collaboration, e.g., smart transport network relies on traffic management. Numerous examples of such synergies exist e.g., smart water - smart energy, smart energy – smart buildings, etc.

In most cases, composition among services provided of such systems is based on service orchestration, which is a basic concept of Service Oriented Architectures (SOA). Service orchestration is a centralized approach for composing services out of existing atomic software services. However, as it was analyzed in Chapter 2, service orchestration is better suited on static environments, when there exists a coordinator, plans are known in advance and minimal changes happen during the execution. Nevertheless, this is not the case in smart cities ecosystems, where changes are introduced frequently and the number and the type of offered services are not upfront defined. For such cases, an adaptive service choreography is a preferred solution, since composition of services is done on a peer-to-peer fashion, leading to autonomously operating services.

In the smart city case, services need to be transformed to smart services that need to exhibit even richer behavior and functionality for sustainable cities. According to ITU Telecommunication Standardization Sector (ITU-T) [[Focus Group Technical Specifications, 2015](#)], the overall smart city architecture should provide support for Transportation services, E-government services, E-business services, Safety and emergency services, Smart health services, Tourism services, Education services, Smart buildings, Waste management services, Smart energy services, and Smart water services.

The development of smart cities' systems and applications demands software development environments that are able to support a number of functional and non-functional requirements. Functional requirements, in general, represent what a service does. Some of the functional requirements that need to be satisfied by a development environment, as

described in [Santana et al., 2018], are:

- Data management, which includes collection, storage, analysis, and visualization of city data.
- Applications run-time support for facilitating deployment and integration of smart cities' applications.
- Sensor network data management and control.
- Data analytics functionality for analyzing massive volumes of data produced by a smart city.
- Service management according to SOA or other service management standards.

Furthermore, a number of non-functional requirements are introduced to enrich the existing list, related mainly but not exclusively to the quality characteristics of the provided services. The characteristics of services, such as those related to quality, privacy and security, performance, transactions, interoperability, scalability, security, privacy, configurability, etc. constitute the non-functional requirements.

According to ITU-T [Focus Group Technical Specifications, 2015], as shown in Figure 6.3, the Smart and Sustainable City (SSC) architecture should be layered and it consists of (i) the sensing layer, (ii) the network layer, (iii) the application layer and (iv) the Operation, Administration, Maintenance and Provisioning, and Security. Many such frameworks have been presented in the literature, focusing on different architectural aspects. For example, in the work of [Gaur et al., 2015] a software framework is presented that includes a semantic layer, which enables exploitation of domain specific data based on the concepts and relationships between these concepts.

6.3.2 The Teamwork Layer

We exploit the existing architecture of smart cities and augment it with a teamwork layer that can introduce the notion of teamwork collaboration within a set of services, agents and other systems that “live” in a smart city, forming the smart city nexus. The components of the architecture that are presented by now are only the basic, since all functional requirements about the specific domain do not exist; they are generic and not analytical for a complete smart city implementation. What we discuss here, is best described as a conceptual application architecture layer, capturing the most prominent requirements for exhibiting an intelligent teamwork-collaborative behavior to smart cities' applications. However, this conceptual teamwork layer could be modified and transformed as necessary, to address all functional and non-functional requirements for a smart city.

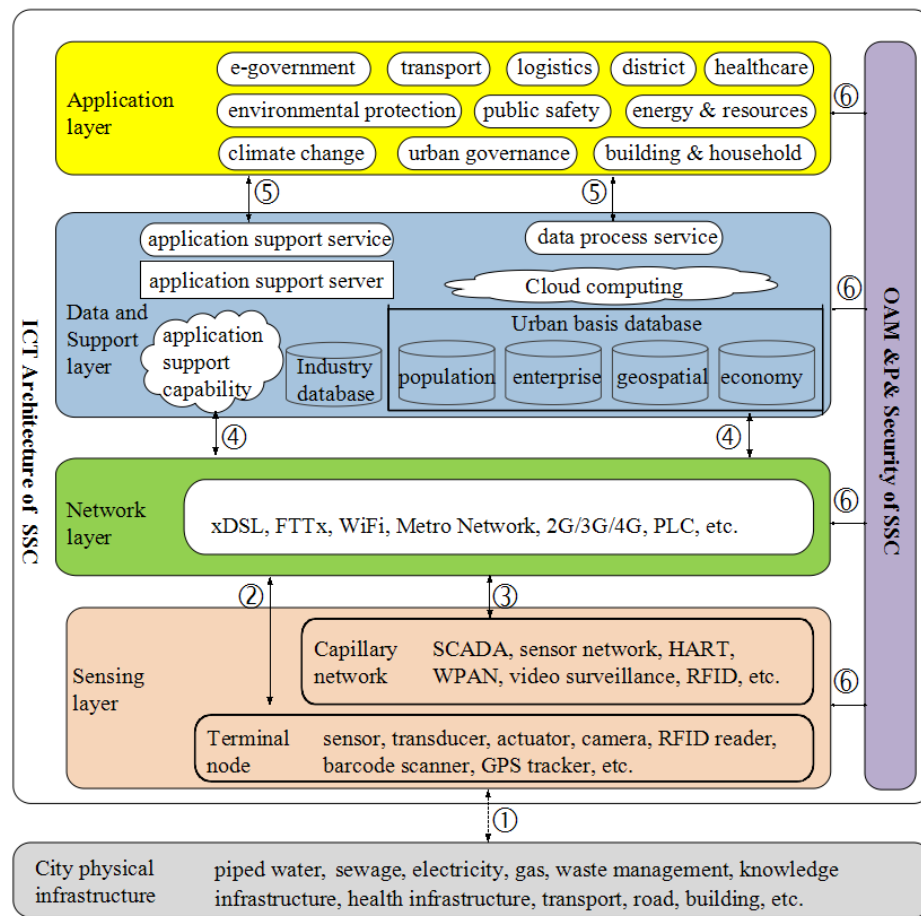


Figure 6.3: Smart and Sustainable City Architecture ITU-T [Focus Group Technical Specifications, 2015]

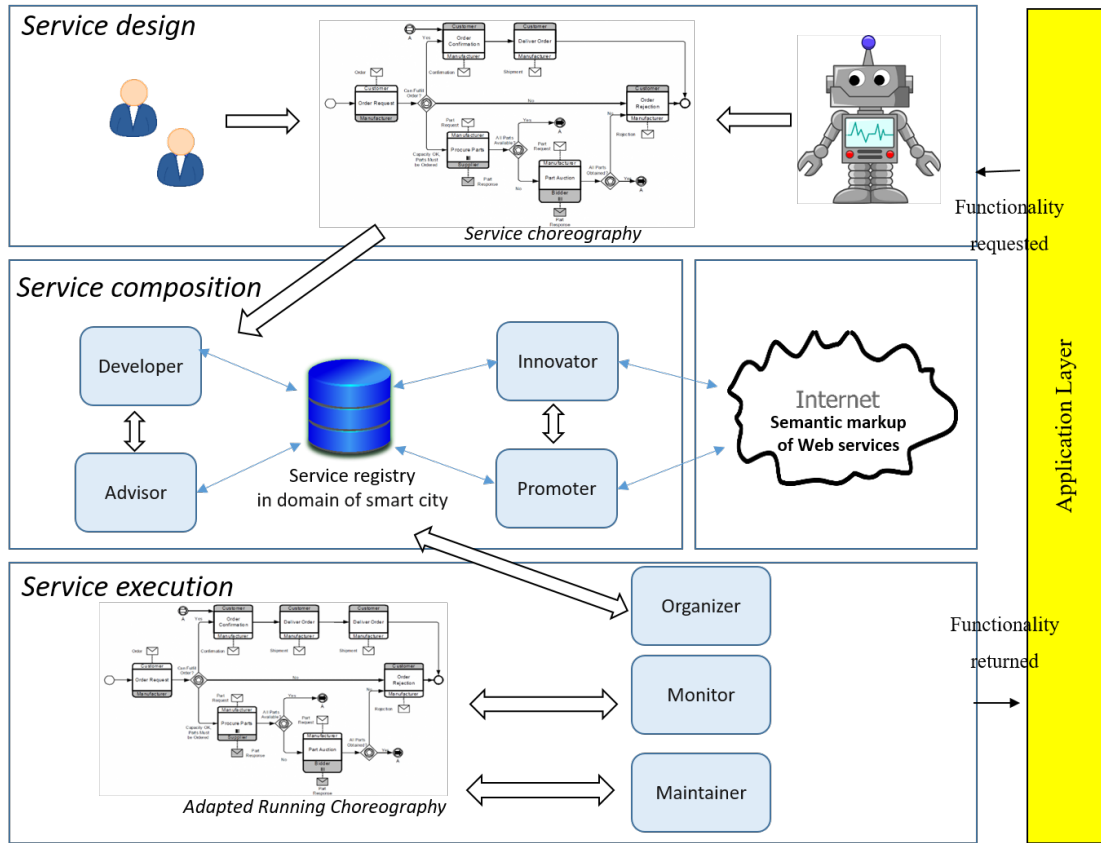


Figure 6.4: Smart and sustainable city service nexus – team working layer.

The teamwork layer is used with the role based modelling approach, for presenting the services-components that cooperate within the smart city ecosystem. According to this approach, and in order to develop a smart city role domain, the role model of the domain is defined. This model contains all roles and behaviors of the domain as they are described in Chapter 3. The set of roles that each service acquires implies the paths for the collaboration of the service with other participants of the city domain. By defining services as set of roles, we apply an abstraction that helps to capture other entities that might exist and cooperate in the city domain in order to form heterogeneous teams, e.g., a smart bus that is constituted of services, agents and robots. This is important since, although there are available city services that have the same functional requirements, not all of them exhibit the same behavior concerning nonfunctional requirements.

Our intention is to enhance the smart city service teamwork roles with appropriate team work behavior for the specific use case of the smart city domain, which will give them the ability to intervene when particular situations occur, for infusing team working behavior within smart and sustainable city services. Additional aspects that will occur in specific cases may be addressed by modifying and extending the teamwork roles and

adapting their detailed behavior that the domain imposes.

The implementation of the SCNS, as described in Figure 6.4, essentially follows a three-step process, which is: (i) Service design phase, (ii) Service composition phase and (iii) Service execution phase. During the service design phase, initially, a choreography could be planned using a business process modelling language, such as BPMN, based on predefined plans of behaviors. The design is done after capturing the service requirements in collaboration with domain experts and users. At this phase, the developer cooperates with the advisor role to provide a plan for the composition. This is done in order to evaluate and extract the set of services which are enrolled and are available within the domain. For describing the service plan, the sequence of behaviors has to be specified and this is accomplished in the context of the developer role. When the result of participant services to be executed is estimated, the monitor role needs to be activated. Meanwhile, service innovator is in collaboration with service promoter and, according to the availability of services in service registry, they either promote existing services to be used or search for new available services that could be found from other providers through the internet. The monitor role in collaboration with the organizer role are the components running and monitoring the produced composite service.

6.4 A Computational Framework

The ontology we produced so far in the previous chapter, inherently, is a static structure without activity. The representation system will determine how the ontology is used. Consequently, we propose an approach, which implements a workflow management module that is combined with a rule engine in order to handle the composition of services in a smart city environment. The rule base component is responsible for handling the required streams of knowledge enclosed in the teamwork roles in order to support the necessary choreography. The proposed approach is conceptualized beyond current protocol-based ones included in the web services protocol stack.

As depicted in Figure 6.5, the rule-based expert system will be able to reason upon the next steps by taking facts, rules and processes into account jointly. If a behavior needs to be executed, the rule engine will request the workflow engine to execute that step. This can be achieved easily because the behavior execution engine is equipped with an Event Listener component that "listens" for any events sent from the rule engine. Once the current step is completed, the behavior engine returns control to the expert system to derive the next step. The expert system implementation gives the control to the rule engine to execute the semantic rules and to decide and notify the workflow engine of which step to execute next. Thus, the rule engine acts as a complementary tool to support the

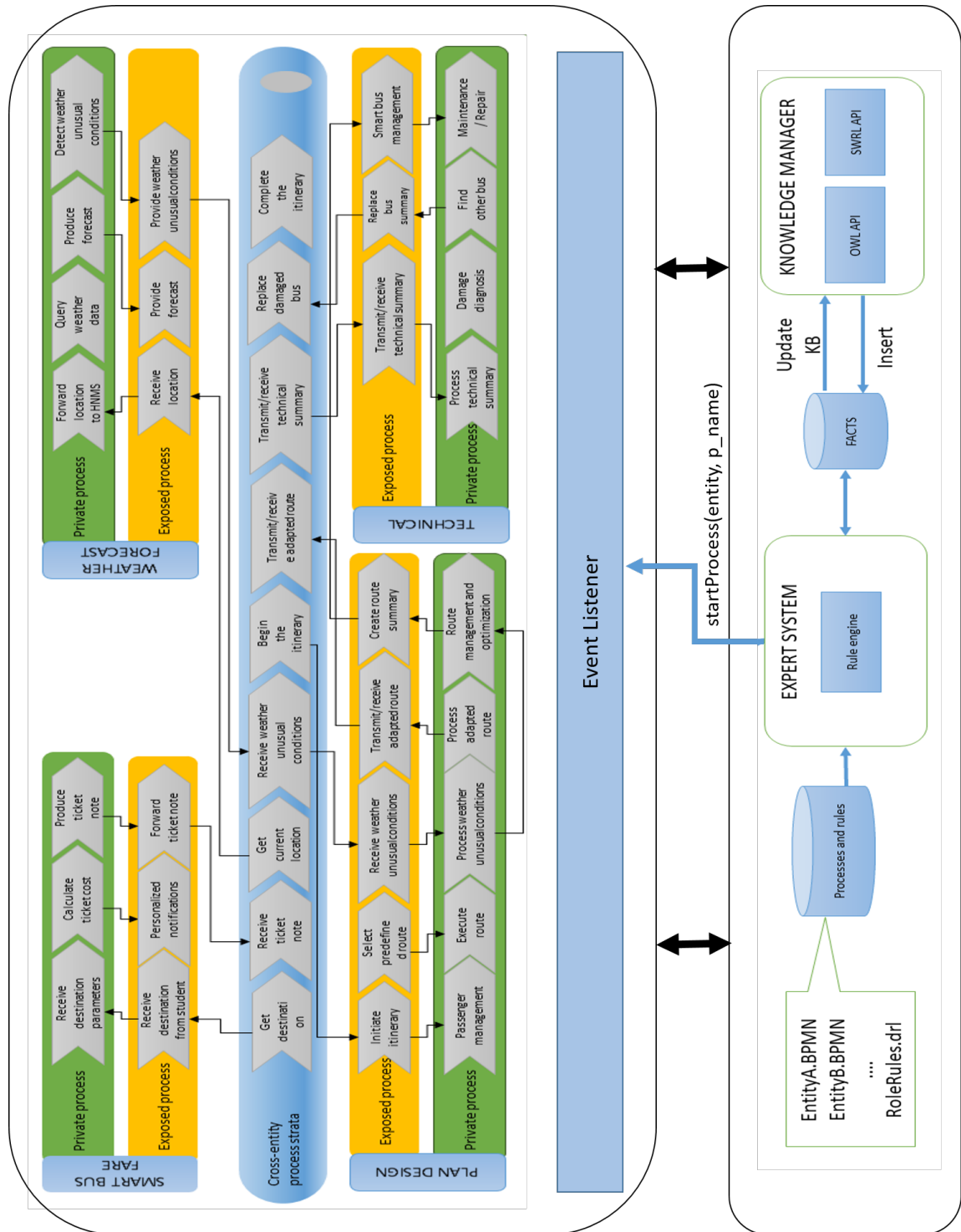


Figure 6.5: The framework for the service composition.

choreography based on teamwork roles and allows dynamic behavior composition and adaptation.

At the upper level, the proposed modeling methodology consists of three aspects:

- Performing complex cross-entity behaviors by applying swimlanes.
- Introducing a layer of abstraction above private behaviors, called “Exposed behaviors”, so as to allow a scalable exposition of internal behaviors.
- Introducing BPMN 2.0 specification as a modeling methodology.

For a successful integration, entity owners have to model their interaction from a high-level point of view, taking into account each teamwork role. Moreover, they have to link their existing internal behaviors and resources to the agreed interaction model and offer process-oriented interfaces to the outside world. An important requirement in this context is to enable entities to conceal their private behaviors for preserving autonomy. The above-mentioned example scenario is comprised by specific behaviors on different levels. As depicted in Figure 6.5, the specific modeling approach introduces the following concepts:

- Cross-entity Behavior Strata - This element constitutes the main behavior that is executed during the specified scenario. The process strata contains the steps of the example scenario. These steps are building blocks or triggers of behaviors inside a smart city entity.
- Exposed Behavior - The Exposed Behavior of each smart city entity constitutes the interface of its internal behaviors to the outer world so as to interact with the external applications in a seamless and transparent to the end user’s manner.
- Private Behavior - The Private Behavior contains all the appropriate steps required in order to implement internal workflows. The Private Behavior is triggered by the Exposed Behavior and is executed inside the entity.

Figure 6.5 presents the interacting smart cities entities during the example scenario. These entities are the following:

- Smart bus fare entity: The “Smart bus fare entity” is the first entity that the passenger interacts with. The passenger uses the specific service to calculate the bus fare between two bus stops and to purchase the ticket.
- Information entity (Weather forecast): The “Information” service is responsible for the provision of weather forecasts for various locations. Depending on the weather

forecast, unusual conditions and warnings may be generated that could lead to route deviations and changes.

- Plan design: The “Plan design” entity encapsulates services for initializing, monitoring and executing the selected itinerary. The Plan design interoperates with the “Information” and “Technical” entity in a seamless and transparent way so as to encounter the exceptions and deviations that may arise.
- Technical: The “Technical” entity is responsible for the technical inspections, bus replacements and maintenance of the fleet.

According to the above analysis, there are 3 layers of behaviors. Firstly, the cross-entity behavior strata includes the steps that need to be followed so as to execute the required bus route. Moreover, each entity comprises an “Exposed Behavior” layer that provides public behavior interfaces to other entities. Finally, each entity contains a “Private Behavior” layer that includes the “internal” parts of its behaviors.

Given the populated ontology, that was described in the previous chapter, we can start observing the environment where each sensor is assigned to a role of interest for detecting changes and consequently recognizing the related behaviors caused the situation. In order to map the sensor data into our representation model, we need the TrEWS ontology. Depending on the features of the domain, the level of uncertainty and complexity, we may apply different reasoning methods on the evolved teamwork behaviors. The aforementioned population process can be extended in order to cover roles and behaviors that are of interest in the specific environment.

6.5 Role Modeling of a Smart City Domain

Modeling services in a smart city role domain could add value to the composite services that have been derived to implement specific domain processes. Role modeling has the advantage to model both participants and their interactions through their behavior. We exploit the applicable concepts and technologies offered for automated composition and further enhance services by enabling teamwork behaviors to these entities.

We remind here that the main concepts using the role modeling approach are roles, communicative actions and behaviors. The way that a role interacts with another role is considered as a behavior of the first role. A role expresses different aspects of collaboration to other roles through different behaviors. Therefore, a role is modeled as a set of the expected behaviors it exhibits, which states the interactions in which it is potentially involved.

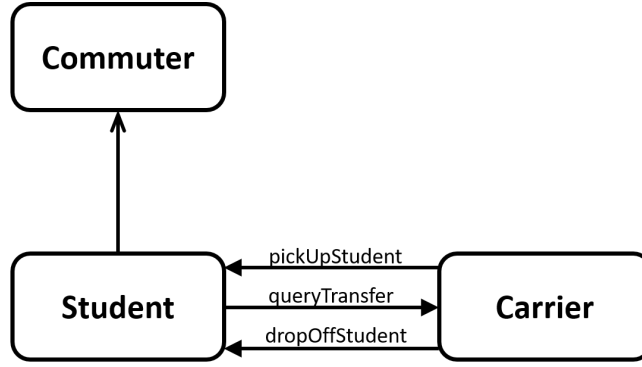


Figure 6.6: Part of the role model of the smart city domain without teamwork roles.

Although the RM diagram presents the possible interaction among roles through the execution of the appropriate behavior, we need, additionally, to state explicitly under which circumstances the role has the ability to execute a behavior, and which are the changes that happen when the behavior is executed. These are the preconditions and effects, respectively, and they are described as concepts in the ontology, in order to be express in a formal way the transition that happens in the domain and enable reasoning over state transitions.

To model a smart city domain as a role domain, we first need to define its objectives. According to the role modeling approach, all domain goals identified are organized in the ontology using the goal concept. Subsequently, the communicative intends and abstract behaviors of the domain, that they are partitioned to roles of the domain, derive from the domain goals. Once roles and their communications are defined, the preconditions and effects of each communication to be executed are stated. After having all the details to design the role model of the domain, we facilitate the role model with the participant services. Available services are enrolled to the domain and for each service the roles that it plays are determined. For this reason, extra details or attributes and their meanings (semantics) must be identified by name, data type, constraints, and default values. Consequently, each service is mapped to the roles to be played. In the next step for the utilization of the role model domain, solutions are designed via finding plans that correspond to domain goals. All the solutions are stored as behavior paths in the plan library. Using the method's representation primitives in Figure 6.6 we firstly show a part of the role model which contains only the operational roles of the domain that are relevant to the case study described in section 6.2. These roles are Student and Carrier. The Student queries a transfer from the Carrier role and the Carrier role replies to pick up the Student and drop him/her to the school.

The domain, apart from the operational roles, also contains the teamwork roles that contribute to its functionality and are obviously employed during the composition process.

Upon to a request for a service, the Advisor role intervenes in order to interact with the clients and understand their needs. It takes the request, collects all the requirements and investigates, whether it matches with existing goals of the domain repository according to its semantic similarity. If that is the case, it retrieves a plan and provides it to the Developer role in order to proceed with the service allocation. Otherwise, it informs the Developer role to search for a new plan, i.e. a sequence of behaviors with the given preconditions and effects. The Developer role extracts the sequence of behaviors from the plan and maps the appropriate services for the service composition. It is the “team builder” of the domain; it makes an evaluation to find the appropriate set of services, as a final product, that could become a team to execute the composite service. In the sequence, it calls the Monitor role to supervise the execution of the plan.

For the effective exploitation of services and assets in a smart city nexus, a continuous flow of information is needed in order for services, when executed, to be adapted to the city changes that happen in realistic environments. Through the perpetual reception of data indicating the behavior of participants in the city domain, local decision makers could be informed about events of their concern. For example, if a municipal service is hardly used, the time constraints for using it should be changed and a number of its resources could be moved to some other area where they may be better exploited. Similarly, by knowing the working hours and days of the employees of the companies housed in a building, could aim in optimizing the energy needed in this building and predicting with more accuracy the parking seats they need or the bus routes they could use in order to improve their daily commute to work. Such everyday processes that take place require to be adapted to everyday needs.

In the next section, we use an example to prove that the services which are chosen according to the possible business rules governing the composition in the smart city, they have an impact on the final product. There are many examples where the rules themselves have an impact on the costs and profits of an organisation. The problem that is presented as an example has a realistic base but has been deliberately simplified for presentation purposes.

6.6 Bus Transportation Representation

Based on the above requirements and using the method’s representation primitives, we show part of the role model that is relevant to our example in Figure 6.7. The involved participants are commuters, which play the Commuter role, but actually they are students that use the local transport. City busses play the Carrier role, information services play the Forecast role, while car technicians play the Maintener role. The labeled arrow in

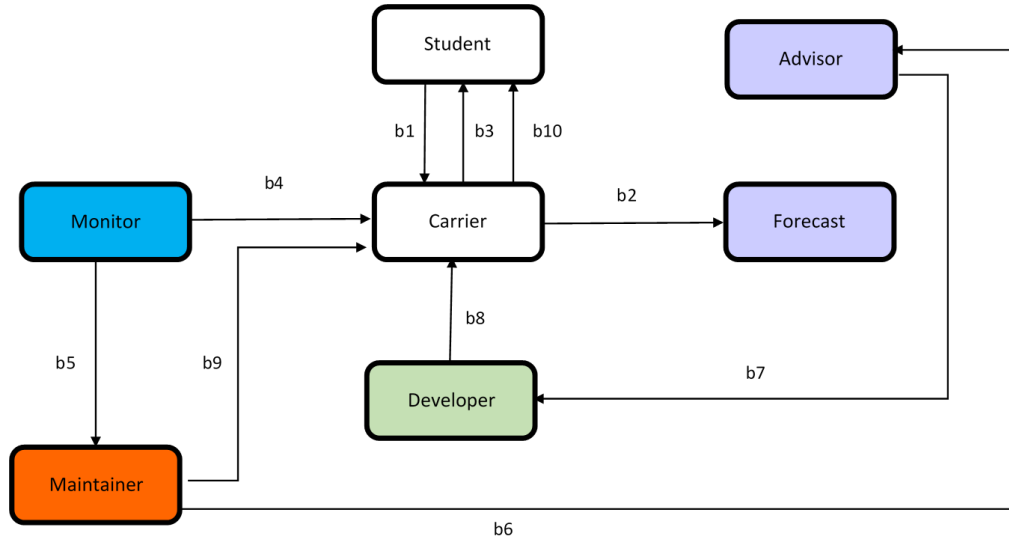


Figure 6.7: Part of the role model of the smart city domain.

the role model is role's behavior and depicts a collaboration path between two roles and indicates who the initiator is. The colored roles are teamwork roles that contribute to the domain functionality.

The example describes a simple but not trivial transfer case. Through this use case, we present how a TrEWS, that is derived from teamwork role modeling, can adapt its behavior and add value in a smart city domain. A goal of the composite service we are looking for is to get all students at school. Before leaving the garage, the bus should check the weather service in case it finds something unusual. This is depicted in the role model with the arrow labeled b2, i.e. the behavior to be executed. The result is for the Carrier to get knowledge about the weather condition. Supposing that everything is controllable with the weather, the bus collects the students according to the designed path as this is defined before it starts the itinerary.

Meanwhile, on its way to the school, the bus displays an urgent maintenance alarm. Then, how shall the composite service that performs the student transfer will achieve its goal? In this situation, the teamwork roles Developer, Monitor, Advisor, Maintainer should intervene and contribute with solutions in order to complete successfully the student transfer. The Monitor will catch the error and ask the advisor for alternative solutions. The Maintainer is informed about the error, and depending on its type, it gives a prediction about how long the bus will be out of order. Suppose that the Maintaner by a brief diagnosis says that the bus need to go to the garage to get maintained, the Advisor would look for other school buses in the surrounding area, check their capacity and inform if there is anyone which can change its route, provided that it can get there in a reasonable amount of time, come to pick up students, so that they could change bus and get to their

destination. The Developer role would help in changing the initial plan. The decision that the composite service should make is critical for the success of its execution, i.e. the students transfer and should take into consideration the proposals of the teamwork roles involved.

In Table 6.1, we see the different roles and behaviors of the smart bus composite service which could be played by different services that are enrolled in the domain.

Table 6.1: Roles and behaviors of the smart bus service

Role	Behavior	CA	Role
Student	b1	queryTransfer	Carrier
Carrier	b2	getWeatherPrediction	Forecast
Carrier	b3	pickUpStudent	Student
Monitor	b4	monitorRoute	Carrier
Monitor	b5	informFailure	Maintainer
Maintainer	b6	estimateFailure	Advisor
Advisor	b7	findAlternative	Developer
Developer	b8	delegateItinerary	Carrier
Maintainer	b9	fixDamage	Carrier
Carrier	b10	dropOffStudent	Student

In Table 6.2 we see the preconditions and effects for each behavior execution.

Table 6.2: Preconditions and effects for each behavior execution

Behavior	Precondition	Effect
b1	schoolDay	studentInRoute
b2	planCalculated	weatherKnown
b3	studentInRoute	studentInBus
b4	busInRoute	failure(aBus)
b5	busFailure	failureKnown
b6	failureKnown	failureEstimated
b7	failure(aBus)	newPlanReuired
b8	available(bBus), onSite(bBus)	newPlan(bBus)
b9	failure(aBus)	fixed(aBus)
b10	onBus(bBus,student)	arriveAtSite

6.7 The TrEWS Population

The following axioms are examples of statements used for the TrEWSOnto population. For the sake of clarity, these examples are either in the form of a unary or a binary predicate. It is worth mentioning that, apart from the A-box, the population also includes a process of extension of the T-box in order to define more specialized subclasses of TrEWSOnto. From the next axioms we can see the roles that have to be adopted by different services participating in the smart bus composite service.

Role Description:

Carrier $\subseteq$ *Role*

Student $\subseteq$ *Role*

Carrier(*aBus*)

Carrier(*bBus*)

Student(*aStudent*)

Advisor(*aForecast*)

Goal(*aGoal*)

TeamworkPlan(*aTPlan*)

Advisor(*anAdvisor*)

Monitor(*anInspector*)

Developer(*aDeveloper*)

Maintainer(*aMaintainer*)

Behavior Description:

hasBehavior(*aBus*, *b1*), *hasBehavior*(*anAdvisor*, *b2*)

hasBehavior(*aBus*, *b3*), *hasBehavior*(*anInspector*, *b4*)

hasBehavior(*aDeveloper*, *b5*), *hasBehavior*(*aBus*, *b6*)

hasBehavior(*aBus*, *b7*), *hasBehavior*(*aMaintainer*, *b8*)

hasBehavior(*aMaintainer*, *b9*), *hasBehavior*(*aStudent*, *b10*)

hasCA(b1, getWeatherPrediction), hasCA(b3, pickUpStudent)
hasCA(b3, performTheRoute), hasCA(b4, dropOffStudent)
hasCA(b5, acceptProposalExchange)
hasCA(b6, informAlternativePlans)
hasCA(b7, findAlternative)
hasCA(b8, delegateToOtherCarrier)
hasCA(b9, maintenance), hasCA(b10, queryTransfer)
execBeh(b1, aForecast), execBeh(b2, aStudent)
execBeh(b3, aBus), execBeh(b4, aStudent)
execBeh(b5, anAdvisor), execBeh(b6, aBus)
execBeh(b7, aDeveloper), execBeh(b8, aCarrier)
execBeh(b9, aCarrier), execBeh(b10, aBus)
hasPrec(b1, planCalculated), hasEffect(b1, weatherKnown)
hasPrec(b2, heavyWeather), hasEffect(b2, adaptedRoute)
hasPrec(b3, newPlanCalcul), hasEffect(b3, onBus(aStudent))
hasPrec(b4, break(aBus), hasEffect(b4, availableBus)
hasPrec(b5, available(bBus)), hasEffect(b5, onSite(bBus))
hasPrec(b6, onBus(bBus, student))
hasEffect(b6, arriveSite)
hasPrec(b7, arriveSite), hasEffect(b7, offBusPassengers)
hasPrec(b8, break(aBus), hasEffect(b8, fixDamage)
leads(tplan, aGoal), isConstOf(tplan, b1)
isConstOf(tplan, b2)

Given the structure of the role model, we continue the population process by adding the behaviors of each role, their preconditions and effects. In other words, the precondition of the behavior needs to be fulfilled when the behavior activity occurs. Once the specific

behavior is triggered, the system can immediately infer whether the next behavior can be executed. We focus here only on basic behaviors that are used in the scenario and axioms referring to the plan will be followed.

6.8 Reasoning over Behavior Adaptation

Given the ontology schema and instances of the smart city domain, the reasoning could be applied to the specific ontology is concerning

- instance checking: given the knowledge base of a city domain K , an individual s and a concept expression C , check whether $K \models C(s)$ and
- query answering: given the knowledge base of the domain K and a concept expression C , find the set of individuals s such that $K \models C(s)$.

The application of the logical rules with regard to teamwork behavior, through the derived facts enable the adaptation of the adopted behavior. A selected reasoner performs the semantic enrichment of the knowledge model by inferring new knowledge while each composition is designed and executed. As rules fire, new facts are inserted into the fact base and the engine keeps inferring relevant facts until the team goal is reached. These facts are then used in further inference and ontology update. An indicative rule for the transfer situation is shortly presented. This rule is conducted to find out which bus will be prioritized to substitute the broken one. **Rule A** describes the decision concerning a bus selection to take about the substitution of the broken one. The hypothesis of the specific rule is that the candidate bus b has the same direction with the broken $aBus$, the candidate bus is in the vicinity area and it is not late while it has available sits for more students.

Rule A

$$\begin{aligned} &hasDestination(?aBus, ?d) \wedge hasDestination(?b, ?d) \wedge Vicinity(?b, ?aBus) \wedge \\ &notLate(?b) \wedge availableSits(?b) \rightarrow onDuty(?b, ?i) \end{aligned}$$

where d is the destination of the $aBus$ and i its new itinerary.

There are many other examples of rules that affect the performance of services in a smart city environment. For example, in public services we have the problem of prioritizing when cases come in and the problem is to determine the urgent ones. This is particularly important in health and social service, such as child protection. But a good definition of urgent requires a rule and some proper parameters. Classifying too many cases as urgent does not benefit, if it is impracticable with the available resources; a queue is simply

created with later on todos. In these cases, it is important not to leave important for the urgent.

6.9 Discussion

Service composition in a smart city will not be effective without a shared policy and a process framework that will support appropriate collaboration models and provide a sustainable environment in which interoperable solutions can be created, deployed, and managed. Coordinated interactions between various entities require a common understanding of their core functions even though alternative delivery mechanisms and channels may be employed. In this chapter, we proposed an integrated framework, aiming at describing and composing a large number of heterogeneous services, that is based on the role model abstract service description which represents them in terms of their roles and behaviors (i.e., context-aware preconditions and effects).

The ontology of teamwork enabled services is extended and populated, targeting to cover the representational aspects of smart city environments. The cornerstone of the proposed framework is the cross-entity behavior strata, which is responsible for adjusting the contributions of those entities in terms of effective service provision, appropriate timing, and best quality, since effectiveness of service composition is a much more complicated issue than simply the ability to provide a plan to follow.

To evaluate our composition framework, we developed an OWL based use case. To conclude, we state that our TrEWS composition framework enables the composition process to be closer to the smart city vision, thanks to the Internet of Things, mobile technologies and next generation networks that are expected to play a key role to the city service provision.

References

- [Ananthakrishnan et al., 2013] Ananthakrishnan, R., Bryan, J., Chard, K., Foster, I., Howe, T., Lidman, M., and Tuecke, S. (2013). Globus nexus: An identity, profile, and group management platform for science gateways and other collaborative science applications. In *2013 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 1–3. IEEE.
- [Balakrishna, 2012] Balakrishna, C. (2012). Enabling technologies for smart city services and applications. In *2012 sixth international conference on next generation mobile applications, services and technologies*, pages 223–227. IEEE.
- [Bibri, 2018] Bibri, S. E. (2018). The iot for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. *Sustainable Cities and Society*, 38:230–253.
- [Fitsilis et al., 2018] Fitsilis, P., Tsoutsas, P., Anthopoulos, L., and Ragos, O. (2018). Teamwork behavior in smart and sustainable cities ecosystems. In *The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications , Venice, Italy*. Springer??
- [Focus Group Technical Specifications, 2015] Focus Group Technical Specifications, F. (2015). Setting the framework for an ict architecture of a smart sustainable city. *ITU-T*, 1:1.
- [Gaur et al., 2015] Gaur, A., Scotney, B., Parr, G., and McClean, S. (2015). Smart city architecture and its applications based on iot. *Procedia computer science*, 52:1089–1094.
- [Maddox, 2018] Maddox, T. (2018). Smart cities: A cheat sheet. In *Retrieved from <https://www.techrepublic.com/article/smart-cities-the-smart-persons-guide>*. TechRepublic.
- [Novotný et al., 2014] Novotný, R., Kuchta, R., and Kadlec, J. (2014). Smart city concept, applications and services. *J. Telecommun. Syst. Manag*, 3(1).

- [Santana et al., 2018] Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., and Milojevic, D. S. (2018). Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *ACM Computing Surveys (CSUR)*, 50(6):78.
- [Soto et al., 2015] Soto, J. Á. C., Werner-Kytölä, O., Jahn, M., Pullmann, J., Bonino, D., Pastrone, C., and Spirito, M. (2015). Towards a federation of smart city services. In *International Conference on Recent Advances in Computer Systems*. Atlantis Press.
- [Tsoutsas et al., 2020] Tsoutsas, P., Fitsilis, P., Anthopoulos, L., and Ragos, O. (2020). Nexus services in smart city ecosystems. *Journal of the Knowledge Economy*, pages 1–21.
- [Tsoutsas and Ragos, 2020] Tsoutsas, P. and Ragos, O. (2020). Towards an ontology for teamwork enabled services. In *Proceedings of the 2020 4th International Conference on Algorithms, Computing and Systems*, pages 69–75. ACM (ISBN: 978-1-4503-7732-4).
- [Vlacheas et al., 2013] Vlacheas, P., Giaffreda, R., Stavroulaki, V., Kelaidonis, D., Foteinos, V., Poulios, G., Demestichas, P., Somov, A., Biswas, A. R., and Moessner, K. (2013). Enabling smart cities through a cognitive management framework for the internet of things. *IEEE communications magazine*, 51(6):102–111.

Αν θες να μάθεις για τη διαδρομή,
 ρώτησε αυτούς που επιστρέφουν.
 Κινέζικο γλωσσικό.
 "To know the road ahead,
 ask those coming back."
 Chinese proverb.

Chapter 7

Conclusions and Future Research

Contents

7.1 Fulfillment of the Thesis Goals	172
7.2 Limitations and Open Problems	177
7.3 Lessons Learned	177
7.4 Future Research	178

In this thesis we have defined a formal and comprehensive framework for the problem of service composition, when services collaborate in the context of a large complex system. We investigated several scenarios of service-oriented computing, defined formalizations, approaches for service composition and correctness notions. We shaped our own view for evolved services, those that have teamwork abilities, the TrEW services. A teamwork role model for service collaboration based on the role modeling approach and enhanced by the introduction of service teamwork roles was proposed. After an extensive literature review, dominant teamwork roles that prevail during service group cooperation were determined. In this model, the main goal of each role is to intervene and “act as a connector” during collaboration in order to keep the team of component services consistent with the goal of the group–team. This vision is to establish role domains that have the IT service infrastructure, which provide pertinent and timely transactions information between entities across the domain. With the usage of team roles, claims and exception management can be automated, resulting in significant administrative efficiencies as well as lowered costs and risk.

In the following sections, we provide a summary of the contributions with respect to the research questions imposed in the first Chapter and discuss the theoretical and

practical limitations of the approach presented. We conclude by sketching directions for future extensions of the contributions of this thesis.

7.1 Fulfillment of the Thesis Goals

In order to summarize the achievements of the research and its contributions to knowledge, the following list recaps the research challenges alongside with the corresponding solutions obtained by the proposed approach. The results for each research question are presented in a collection of selected papers as they are indicating in each issue and they depict the contributions of the thesis. The list testifies that the implemented approach addresses solutions to all research questions defined.

RQ1: How to model services in order to better organise them in a relevant domain and manage the existing complexity of services offered in the web?

- i How to model the heterogeneity of services that exist in a domain (business, IoT, IT, web services)?
- ii How to achieve automatic synthesis of local and distributed services in a system that takes into account diverse composition concerns simultaneously including constraints that regard to the domain holistically?
- iii How to cope and construct a plan for the composite service using the large number of services that are available on the web?
- iv How participants of a domain will not be aware that the system consists of several distributed components which implement the composite system and that have been bound without revealing specific implementation details?

ARQ1: We model web services by using the *Role Modeling* approach which help us to tackle simultaneously several issues. More specifically, we introduced in [Tsoutsas et al., 2017] and described in more details in Chapter 3 the service role modeling, where services are modeled as roles appearing in a role domain and their collaboration is defined by their behaviors. The distinguishing characteristic of the proposed model lies in its grouping ability. Due to the vast diversity in the available web services, from the client’s point of view, it is difficult otherwise to find and decide which services should be used for the composition and establish the basis for their selection. Moreover, from the perspective of applicability, the grouping ability is crucial to a WSC problem since it determines whether a WSC method can respond to a

composition request within a reasonable time frame. Defining them as set of roles allows an abstraction and helps to capture heterogeneous services that exist and need to cooperate in the domain to form composite ones. With this methodology we aim to create a model that would act as a mediator for filling in the interoperability gap between different services which may serve the same users needs.

Exploiting the abilities that the behavioral description of the method provides, we are able to achieve the automatic synthesis by considering the composition problem as a 'planning as satisfiability problem' to provide a solution. The solution lies in creating service composition plans of the behavioral description of services. By exploring the role collaborations in a role model, we create reusable valid sequences of role behaviors while this makes possible to infer new paths of collaboration of existing roles. Since a Web service is a set of roles, which in turns is a set of behaviors, the transitions that happen to the states of the environments are made by the behavior executions. In a sense, the modeling of service composition concerns the modeling of the changes made by their behavior execution, which from this point of view are the domain changes. In Chapter 3, we presents an upper ontology of service role models, their mathematical definition and the composition process.

RQ2: Is it possible for a number of services that have been aggregated in order to develop new services to construct new composite services that are going to act as teams to achieve their goal?

- i How to convert a group of services to a team in order for participants to cooperate more efficient?
- ii Can we exploit the teamwork theory and approaches applied to group of humans and agents?
- iii Is it possible for a composite service to be augmented with teamwork abilities?

ARQ2: The idea to exploit *team working theories* for infusing teamworking behaviour to services was motivated by human teamworking and the recent trends to agents and robots towards a team formed to achieve its goal. Consequently, to provide a solution to the second research question, we carried out an investigation about existing methods and tools that intended to identify potential correlation between teamwork theory in human, agent and service teams [Tsoutsas et al., 2019]. A main goal of the study was to exploit teamwork roles and factors that they were identified within the research literature on both human and agent teams and use them in our role modeling approach of service composition by employing teamwork roles within

the composition process. In Chapter 3, the state of art in this field was given and the basic factors were described, especially those that were needed for defining the teamwork roles. These include factors that (i) promote the organization of the group, (ii) strategy related factors which are fundamental for defining the strategy of the team and (iii) synergy related factors which are used by members of a group in order to work as a team in order to achieve a shared goal.

RQ3: Which teamwork roles should be included in a team of services? How the teamwork “enhanced” services interact and how these roles will systematically monitor and support the process of service composition from inside the team?

- i Which are the teamwork roles that should be included for the successful service composition and execution?
- ii What is the scope of tasks each one will have? Another central issue in a composition approach is the error detection during the composition process and correction. In case an error is detected, how participating services are responsible for this error and how they solve it?

ARQ3: In [Tsoutsas et al., 2018a] we discussed the design and implementation details of a teamwork model to address the challenges for teamwork composition of services in role domains. The results are presented in more details in Chapter 5, which are a step towards the definition of a model that will enable services to transform to more intelligent services and to interact with each other autonomously, to crawl within their domain to discover other trusted services and devices, to delegate their assignments to other similar trustworthy services, and state explicitly their capabilities.

With the proposed model, the state of the art is advanced by providing a definition for eight service teamwork roles to be used during the composition process which are developer, organiser, promoter, monitor, producer, advisor, innovator and maintainer. The teamwork roles constitute the conceptual model for gaining knowledge which can be applied within service composition to guide its structure, and make it more efficient, flexible and reusable. We proceed by giving the detailed description for each role, information about its responsibilities and by naming all the necessary teamwork factors it includes focusing on how our approach complements these aspects. Additionally, we present for each role a design pattern providing a reusable solution to occurring problems within a given context in a role domain. The objectives

of the following role descriptions are to describe the intent, applicability, and consequences of the eight teamwork roles during the service composition. It is not a finished design that can be transformed directly into source code but it is a template for how to solve a problem that can be used in many different situations, since design patterns are formalized best practices that can be used to solve common problems when designing an application or a system. Moreover, we explain the behavior of each role by using UML Sequence Diagrams.

RQ4: How to guarantee the correctness of the system in use?

- i What formal methods could be used to guarantee the correctness of the proposed system?

ARQ4: The challenge of developing web services that use Semantic Web technologies is still quite formidable when implemented in practice. In order to support the development of such services, an emphasis needs to be placed on designing and implementing an ontology which extends the role model ontology with appropriate service team roles, based on continuous peer-guidance of team participants [Tsoutsas and Ragos, 2020]. We present in Chapter 5 the TrEWSOnto, aimed at formalizing concepts from teamwork theory relevant to services and their composition process. TrEWSOnto is extensible, human and machine readable, and the OWL source file will be available for research use.

By using ontologies to define TrEWS we do not only introduce sharable and reusable knowledge representation concerning the teamwork abilities, but also add new knowledge about the domain. Having an underlying ontology of teamwork for services is necessary for this purpose, thus we have put forth in Chapter 5 a basis for developing a detailed ontology of teamwork roles for services. Such roles, in a fully developed ontology for a specific domain, may span across different responsibilities, but simultaneously they will be related within the same ontology and their relations can be uncovered and then captured in the ontology presenting the different business strategic that specific models follow.

RQ5: How this will be applied in a domain to verify it and test its correctness?

- i How the designed isolated systems will be integrated in a framework for a specific domain?
- ii What is the architecture that it will be included?

- iii What are the challenges for a specific domain that would be considered for the specification of the teamwork roles?

ARQ5: In Chapter 6, we partially implement and detail a use case by utilising the proposed composition model. This was done to test its correctness, effectiveness and performance. The case was chosen to be from a smart city environment, which is an evolving area. With the specific use case we want to prove that the proposed method can be applied to realistic environments such is the smart city. Consequently, the final contribution of this research in [Fitsilis et al., 2018], [Tsoutsas et al., 2020], [Tsoutsas et al., 2021] is an architecture and a framework, for supporting the service composition problem in smart city domains by developing teamwork enabled services. To the best of our knowledge, such a framework is not available neither as a commercial system nor as a research prototype. Based on this approach, the framework consists of those IT systems that could serve a choreography between a large number of heterogeneous services so as to achieve a seamless and cooperative environment suitable for a smart city. Using the framework enables composite city services to adapt their behavior during execution and themselves intervene from inside the team if a possible unexpected behavior happens during the service activity, in order to run proactively and avoid obstacles or collisions. The scenario illustrates that services having different teamwork abilities are composed to a new one which inherits teamwork features and combines them to something novel.

One of the main concerns in this model is for participants of the smart city to improve the collaboration of the composed systems. Services enrolled within the same domain will be aware of this fact, and will be able to share knowledge artifacts of the domain existing in various forms and formats. Further, through the teamwork roles, e.g. the role of the advisor, services are informed about other services that have similar behavior, their capacity, which of them are trusted, their functioning within the system in relation to the smart city's semantic model (e.g., smart city ontology) and the geospatial information of each service (e.g., which smart building is in the proximity of a smart vehicle). The satisfaction of these requirements would lead to the implementation of important operations within a team such as delegation, common intention and knowledge sharing behavior.

The framework consists of:

- a semantic model (ontology), which aims at the representation of the set of entities participating in the service choreography and their relations by utilizing semantic web technologies,

- the representation of the teamworking roles through the creation of the complete set of Semantic Web Rule Language (SWRL) rulesets that model the corresponding behaviors, and
- the composition algorithms for services based on workflow management technology which helps to organize, automate and improve collaborated processes.

Through the proposed framework, the produced composite services are more independent resulting to more efficiently operating smart cities services, although they are based on volatile conditions which are made by web services, cloud services and IoT services.

7.2 Limitations and Open Problems

There are certain limitations to the proposed model. For example, it requires an amount of work to apply the model in a real scale domain starting from scratch, since it implies that all services of the domain need to be mapped manually for the first time they are developed to a role domain. In order, to overcome this problem the semantic technology could be exploited and the abilities of a service could be defined in terms of their behaviors.

Further, while our ontology is a sound initial offering, we recognize that it can be further extended to include other relevant forms of specific process knowledge related to the relevant domain. Clearly, the teamwork behaviors defined are not enough and the analytical primitive behaviors should be designed according to the proposed role descriptions and the specific application domain.

Whereas a clear framework of ontology development and utilization is offered, the composition process can still be improved from an automation standpoint. For example, although queries can be developed to draw accurate and rich information, this level of knowledge can not be used for mass consumption. Consequently, additional efforts can focus on expanding the existing algorithm and include heuristic approaches allowing improvement of the composition process.

7.3 Lessons Learned

One of the difficulties for this thesis was the multi-disciplinary nature that covers, at least the areas of Web Services, artificial intelligence planning, multi-agent systems and teamwork. Therefore, this required different research areas to be studied and exploit synergy among them. It was often difficult to get the right focus for researching the appropriate research papers because, depending on the background, the contents and the

relativeness of the paper studied was often interpreted differently. However, this helped enrich the feedback that was received for this work.

Another lesson learned from our experimentation, especially during the first stage of implementation, is basically that when we have in parallel so many open problems to solve, major assumptions and adoption of concrete solutions should be preferred, in order for the work to progress, at least in the beginning.

The core part of the work presented in previous chapters has been published in international conferences and journals. We have received valuable feedback through the anonymous referees during the reviewing process. We have also got helped a lot from the presentation of the papers in the conferences, including both questions and observations from the conferee and the discussion with researchers who have the same research interests.

7.4 Future Research

This thesis is preparing the ground for several possible directions for future research, some of which are outlined in this section. In the following list, we also discuss additional extensions that we are currently developing and highlight some possible future directions.

1. We intend to continue **expanding the ontology** and improving its expressivity, by incorporating more concepts, relations and properties in order to conceptualize more teamwork behavioral aspects of TrEW services. The enhancement of the TrEWSOnto will focus on quality related knowledge representation and organizational issues modeling so as to expand the scope of the ontology. We are currently working on using TrEWSOnto as the core element ontology in the development of a framework of services regarding industry and e-government domains. There, we expect to validate some novel concepts with the collaboration of domain experts.
2. With regard to usability, it would be helpful to develop a teamwork role model **graphical editor**. The graphical teamwork role editor will be a software component that could enhance dramatically the user friendliness of the IT infrastructure. Such a graphical designer could allow users of a domain to model the corresponding behaviors without deep IT knowledge. This editor will also enable users to register service specifications either from scratch or by translating existing descriptions in WSDL, OWL-S or WSMO and filling up information that is exclusive to the proposed model.
3. A further research direction is to include in the existing framework a **Simulation** component that allows to preview the designed and resulting service compositions, possible modifications according to expert knowledge of the domain and execute

them (e.g., on a BPMN engine). Violations that should trigger adaptation may occur due to various events, from infrastructure failures to errors related to functionality and many of these could be detected by using a simulator [Tsoutsas et al., 2018b]. In cases where adaptation is not effective, the composition can be extended to support re-planning in order to partially recreate the plan based on a modified goal specification. These different sources of violation require attention during the design process and bring the demand for a simulation in order to combine them for a treatment.

4. Finally, another significant research direction is to investigate whether the proposed service specification and composition approach will benefit from a **weight (priority) model** as metadata in each behavior. This weight model will establish the required infrastructure to optimize the service composition. This could be combined to give feedback to the composition algorithm in order to handle the optimisation problem [Wang et al., 2020] during the team formation and service assignment.

References

- [Fitsilis et al., 2018] Fitsilis, P., Tsoutsas, P., Anthopoulos, L., and Ragos, O. (2018). Teamwork behavior in smart and sustainable cities ecosystems. In *The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications , Venice, Italy*. Springer??
- [Tsoutsas et al., 2020] Tsoutsas, P., Fitsilis, P., Anthopoulos, L., and Ragos, O. (2020). Nexus services in smart city ecosystems. *Journal of the Knowledge Economy*, pages 1–21.
- [Tsoutsas et al., 2017] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2017). Role modeling of iot services in industry domains. In *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, pages 290–295. New York, NY, USA: ACM.
- [Tsoutsas et al., 2018a] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2018a). Enhancing teamwork of services. In *Business Information Systems, Berlin*, pages 290–295. Springer.

- [Tsoutsas et al., 2018b] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2018b). Services simulation in industry 4.0: A comparison of simulators. In *In 3rd International Scientific Conference on Industry 4.0*. In 3rd International Scientific Conference on Industry 4.0.
- [Tsoutsas et al., 2019] Tsoutsas, P., Fitsilis, P., and Ragos, O. (2019). Teamwork behavior: A review to interconnect industry 4.0 entities. In *Technological Developments in Industry 4.0 for Business Applications*, pages 1–25. IGI Global.
- [Tsoutsas et al., 2021] Tsoutsas, P., Iatrellis, O., Ragos, O., and Fitsilis, P. (2021). A framework for developing teamwork enabled services in smart city domains. In *Accepted for publication in 4th International Conference on Computers in Management and Business (ICCMB2021)*. ACM.
- [Tsoutsas and Ragos, 2020] Tsoutsas, P. and Ragos, O. (2020). Towards an ontology for teamwork enabled services. In *Proceedings of the 2020 4th International Conference on Algorithms, Computing and Systems*, pages 69–75. ACM (ISBN: 978-1-4503-7732-4).
- [Wang et al., 2020] Wang, C., Ma, H., Chen, G., Hartmann, S., and Branke, J. (2020). Robustness estimation and optimisation for semantic web service composition with stochastic service failures. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ



Ένα διευρυμένο μοντέλο σύνθεσης
υπηρεσιών ιστού
με συνεργατική σημασιολογία

Παρασκευή Α. Τσούτσα

Εκτεταμένη Ελληνική Περίληψη
Διδακτορικής Διατριβής

Πάτρα, 2021



Πανεπιστήμιο Πατρών

Τμήμα Μαθηματικών

Παρασκευή Α. Τσούτσα

Copyright © 2021 - Με την επιφύλαξη παντός δικαιώματος

1.1 Εισαγωγή

Οι διαθέσιμες υπηρεσίες ιστού βρίσκονται στο διαδίκτυο με τη λογική που λειτουργούν τα κατανεμημένα δίκτυα. Ενώ λοιπόν δεν έχουν συγκεκριμένη οργάνωση και δεν ακολουθούν κάποια ιεράρχηση, δύνανται μέσα από αυτό το χάος να παρέχουν πολύτιμες πληροφορίες όπως και να κρύβουν λύσεις που με μοναδικό τρόπο θα μπορούσαν να επιλύσουν συγκεκριμένα προβλήματα. Πώς όμως θα μπορούσαμε να εκμαιεύσουμε μια λύση ώστε να δημιουργηθεί μια σύνθετη υπηρεσία με προστιθέμενη αξία, όταν η αναζήτηση γίνεται ανάμεσα σε μυριάδες και ετερογενείς διαθέσιμες υπηρεσίες ιστού; Η τεχνητή νοημοσύνη, σε συνεργασία με την θεωρία συνεργασίας, είναι εφικτό να μας δείξει τον τρόπο που θα συνδεθούν οι κατάλληλες υπηρεσίες ώστε να αναγνωρισθούν και να προταθούν νέες λύσεις που σε διαφορετική περίπτωση θα ήταν αδύνατο να διαπιστωθεί η ύπαρξη τους και να χρησιμοποιηθούν.

Για τους ιδιώτες, παραδείγματα σύνθεσης υπηρεσιών περιλαμβάνουν την καθημερινή έρευνα στο διαδίκτυο για τη χρησιμοποίηση υπηρεσιών σχετικά με την έκδοση εισιτηρίων για τη μεταφορά τους ή την ρύθμιση των οικονομικών τους υποχρεώσεων με τη χρήση τραπεζικών υπηρεσιών και σύγχρονων υπηρεσιών ηλεκτρονικής διακυβέρνησης που συνεχώς αυξάνονται. Σε επιχειρησιακά περιβάλλοντα με υπηρεσιοστραφείς αρχιτεκτονικές, αρκετά συχνά, αντί να γράφεται κώδικας για τη δημιουργία νέων εφαρμογών και υπηρεσιών, ιδανικά αυτές δημιουργούνται από συνθέσεις των υπαρχόντων [1,2]. Ωστόσο, παρά την επιτακτική ανάγκη συνεργασίας των υπηρεσιών για τη σύνθεση, η δυνατότητα αυτών να λειτουργούν ως ομάδα παραμένει αδιερεύνητη. Στον αντίποδα, η δυνατότητα της συνεργασίας και της ομαδικής συμπεριφοράς έχει διερευνηθεί στους ανθρώπους αλλά και μέσω της τεχνητής νοημοσύνης στους πράκτορες και στα ρομπότ [3,4]. Η διατριβή αυτή διερευνά τη σύνθεση των υπηρεσιών από τη σκοπιά της *συνεργασίας (teamwork)*. Δίνεται έμφαση στην ανάπτυξη ενός μοντέλου σύνθεσης που στηρίζεται στην ομαδικότητα, και εκμεταλλεύεται τις δεξιότητες ομάδας που οι εμπλεκόμενες υπηρεσίες φέρουν μέσω της συμπεριφοράς που εκφράζουν.

Παρόλο που η σύνθεση υπηρεσιών ερευνάται εδώ και χρόνια, εξ όσων γνωρίζουμε, δεν υπάρχει ευρέως αποδεκτό μοντέλο προσανατολισμένο στη συνεργασία κατά την σύνθεση των υπηρεσιών. Πρακτικά, αυτό μας φέρνει στο κεντρικό ερευνητικό ερώτημα που αγγίζει αρκετές ερευνητικές προκλήσεις που αναγνωρίζονται στη βιβλιογραφία και διερευνώνται σε αυτή τη διατριβή:

Πως θα δημιουργήσουμε σύνθετες υπηρεσίες που μπορεί συστηματικά να υποστηριχθεί η λειτουργία τους εκ' των έσω;

Για να απαντήσουμε στο κεντρικό ερευνητικό ερώτημα αλλά και σε μεγάλο μέρος των προκλήσεων που τίθενται για τη σύνθεση, προτείνουμε μια καινοτόμο προσέγγιση για το πρόβλημα, τη σύνθεση υπηρεσιών βασισμένη στην συνεργασία (Teamwork Oriented Service Composition), χρησιμοποιώντας την τεχνική μοντελοποίησης ρόλων και υποστηρίζοντας τη διαλειτουργικότητα των υπηρεσιών μέσω των ρόλων ομαδικής εργασίας (Teamwork Roles - TR). Σύμφωνα με αυτή την προσέγγιση, οι υπηρεσίες μπορούν να οργανωθούν σε τομείς που μοντελοποιούνται με ρόλους και εξυπηρετούν συγκεκριμένους στόχους. Η ιδέα είναι, πως ένα τέτοιο πεδίο προβλήματος που έχει ως «κατοίκους» υπηρεσίες ιστού, θα μπορούσε να σχεδιαστεί χρησιμοποιώντας ρόλους, με την έννοια ότι δεν θα αντιμετωπίζει τις αλληλεπιδράσεις μεταξύ των υπηρεσιών ιστού, αλλά μεταξύ των ρόλων που υποδύονται. Χρησιμοποιούμε την προσέγγιση μοντελοποίησης ρόλων και ενισχύουμε τη διαδικασία της σύνθεσης των υπηρεσιών με οκτώ ρόλους ομαδικής εργασίας που θα υποστηρίξουν τη συνεργασία των υπηρεσιών. Οι TR έχουν ως αποτέλεσμα την δημιουργία σύνθετης υπηρεσίας υψηλής ποιότητας, ικανής να επιδείξει συμπεριφορά ομαδικής εργασίας ώστε να μπορεί να συνεργαστεί αποτελεσματικά με πολλούς πιθανούς συνεργάτες σε ένα σύνολο εργασιών και να μπορεί να εντοπίσει ή να αποτρέψει λάθη και αναδυόμενες συμπεριφορές προκειμένου να καταπολεμηθεί ο κίνδυνος επισφάλειας που αντιμετωπίζουν για την επίτευξη του συνολικού στόχου της ομάδας.

Από επιχειρηματικής άποψης, η σύνθεση συνεργατικών υπηρεσιών θα μειώσει δραματικά το κόστος και τους κινδύνους δημιουργίας νέων πληροφοριακών συστημάτων, υπό την έννοια, ότι υπάρχουσες επιχειρησιακές λογικές μπορούν να απεικονίζονται ως συνεργατική υπηρεσία που συμπεριλαμβάνει όλα τα στάδια οργάνωσης, σχεδιασμού και ελέγχου κατά την εκτέλεση τους. Δεδομένου ότι οι επιχειρηματικές υπηρεσίες μπορούν επίσης να προσφερθούν στο διαδίκτυο μέσω των επιχειρηματικών πληροφοριακών συστημάτων η σύνθεση της συνεργατικής υπηρεσίας είναι εφικτό να γίνει κατόπιν συγκεκριμένου αιτήματος. Αυτό επιτρέπει την εκμετάλλευση νέων λύσεων που βρίσκονται στο διαδίκτυο και την αντικατάσταση παρωχημένων υπηρεσιών από νέες. Επίσης, δίνει τη δυνατότητα σε κάθε επιχείρηση να εστιάσει στη δημιουργία συνεργατικών υπηρεσιών για την επίλυση συγκεκριμένων προβλημάτων, επαναχρησιμοποιώντας τις υπηρεσίες συνεργατών για την υποστήριξη επικουρικών της δραστηριοτήτων από τρίτους παρόχους που ειδικεύονται σε τέτοιες δραστηριότητες. Για παράδειγμα, οι επιχειρήσεις αναθέτουν σήμερα τη διαχείριση του συστήματος διαχείρισης πελατών (CRM) ή συστήματος τιμολόγησης σε τρίτους. Αυτό τους δίνει τη δυνατότητα να επεκτείνουν τον έλεγχο και την ανάλυση μέσα και έξω από το πεδίο - τομέα της επιχείρησης, ενώ βελτιώνει την αποδοτικότητα της, καθώς δεν χρειάζεται να έχουν εξειδικευμένο τμήμα ή να κατασκευάσουν ένα νέο σύστημα για τον χειρισμό αυτών των υπηρεσιών.

Η προτεινόμενη προσέγγιση διακρίνεται από τις υπάρχουσες ερευνητικές εργασίες, δεδομένου ότι εστιάζει στην προοπτική της συνεργασίας των συμμετεχόντων υπηρεσιών. Το μοντέλο

συνεργασίας παρέχει την δυνατότητα ευελιξίας της σύνθεσης στο σύστημα διασφαλίζοντας παράλληλα την ορθότητα της λύσης που θα προσφέρει εύρωστες σύνθετες υπηρεσίες. Εσφαλμένες αυτόματες συνθέσεις υπηρεσιών θα μπορούσαν να επιδείξουν απροσδιόριστη και απρόβλεπτη συμπεριφορά και συνεπώς να επηρεάσουν όχι μόνο τεχνικά τη διαδικασία της σύνθεσης των υπηρεσιών αλλά επίσης να θεωρηθεί ότι αποτελούν απειλή για το κοινό σε συστήματα κρίσιμα για την ασφάλεια ή να βλάψουν τα εμπορικά συμφέροντα ατόμων ή οργανισμών.

1.2 Βιβλιογραφική έρευνα

Στη συνέχεια παρουσιάζουμε βασικές έννοιες της σύνθεσης, της μοντελοποίησης με ρόλους και της ομαδικής εργασίας καθώς και επιλεγμένες σύγχρονες προσεγγίσεις σύνθεσης από τη βιβλιογραφία που μελετήθηκε.

1.2.1 Οι διαφορετικές προσεγγίσεις για τη σύνθεση

Το πρόβλημα της σύνθεσης υπηρεσιών είναι το πρόβλημα της κατασκευής μιας νέας σύνθετης υπηρεσίας, χρησιμοποιώντας τις ατομικές διαθέσιμες υπηρεσίες με συγκεκριμένη λειτουργικότητα, όταν καμία από τις ατομικές υπηρεσίες δεν ικανοποιεί την αιτούμενη ανάγκη. Το πρόβλημα αυτό ανήκει σε μια εξελισσόμενη ερευνητική περιοχή και συχνά εμφανίζονται νέες προτεινόμενες μέθοδοι για τη λύση του. Οι τρέχουσες προσεγγίσεις περιλαμβάνουν γλώσσες περιγραφής αυτών και βιομηχανικά πρότυπα (π.χ. BPEL και OWL-S) αλλά και περισσότερα θεωρητικά μοντέλα και γλώσσες (π.χ. automata, δίκτυα Petri net, κ.α.). Είναι γενικά παραδεκτό πως η διαδικασία της σύνθεσης θα πρέπει να περιλαμβάνει έξυπνους τρόπους λειτουργίας ανακάλυψης και διαπραγμάτευσης των απαιτούμενων ατομικών υπηρεσιών.

Μελετήθηκαν εργασίες και βιβλιογραφικές επισκοπήσεις που αναλύουν τις υπάρχουσες κύριες τεχνικές, ερευνητικά πρότυπα και στάνταρτς, καθώς και εργασίες που αναδεικνύουν περισσότερες ανοιχτές και ανεπίλυτες ερευνητικές προκλήσεις στην περιοχή [5]. Συνολικά, οι προσεγγίσεις στο πρόβλημα της σύνθεσης των υπηρεσιών είναι εξαιρετικά διαφορετικές και προσφέρουν ποικίλες ερμηνείες για τα θέματα που πρέπει να αντιμετωπιστούν κατά τη σύνθεση. Παρόλα αυτά, ενώ υπάρχουν πολλαπλές προσεγγίσεις που ξεφεύγουν από την απλή συντακτική περιγραφή της υπηρεσίας για το πρόβλημα της σύνθεσης, παρατηρούμε ότι οι περισσότερες στερούνται της αντίληψης της ολιστικής άποψης για την συμπεριφορά της υπηρεσίας καθ' όλη τη διάρκεια της δέσμευσής της για τη σύνθεση. Επιπλέον, αναλύουν την σύνθεση υπηρεσιών ως μια τεχνική συνάθροισης υπηρεσιών ιστού για να επιτύχουν κάποιο συγκεκριμένο προϊόν που είναι η συντιθέμενη υπηρεσία, χωρίς να μοντελοποιούν τη συμπεριφορά του συστήματος που αναδύεται δυναμικά μέσω των συνεργαζόμενων οντοτήτων.

Στον αντίποδα, η μοντελοποίηση με ρόλους είναι ένα ισχυρό εργαλείο που μπορεί να καταγράψει τη συμπεριφορά των συμμετεχόντων σε μια συνεργασία. Οι ρόλοι χρησιμοποιούνται σε πολλούς τομείς, όπως συστήματα πολλαπλών πρακτόρων (Multi Agent Systems MAS), Object Oriented

modeling and Programming Languages, Organization and Security. Υπάρχουν διαφορετικοί ορισμοί των ρόλων, άλλοι με μια προσέγγιση βασισμένη στη διαίσθηση, άλλοι στις πρακτικές ανάγκες και, ενίοτε, σε άλλους επιβάλλεται να ακολουθούν τυπικά πρότυπα. Στην ασφάλεια για παράδειγμα, οι ρόλοι θεωρούνται ως ένας τρόπος διανομής αδειών, σε οργανωτικά μοντέλα, δίνουν δικαιώματα στους συμμετέχοντες για να έχουν πρόσβαση στα πολυπρακτορικά συστήματα και μπορούν να θεωρηθούν ως περιγραφή της αναμενόμενης συμπεριφοράς για τους πράκτορες που τους υποδύονται.

1.2.2 Μοντελοποίηση με ρόλους (Role Modeling)

Η Object Oriented Role Analysis and Modeling μέθοδος (OOram), που αναπτύχθηκε από τους Reenskaug και Andersen παρουσιάζει μια γενική προσέγγιση στη μοντελοποίηση αντικειμένων και τη συνεργασία τους μέσω ρόλων [6]. Ένας ρόλος είναι μια αφαιρετική έννοια που έχει μια ταυτότητα, μια συμπεριφορά και μια κατάσταση· αποτελεί μια αρχέτυπη αναπαράσταση των αντικειμένων που καταλαμβάνουν τις αντίστοιχες θέσεις στο συνολικό σύστημα που απαρτίζεται από αντικείμενα. Όλα τα αντικείμενα που εξυπηρετούν τον ίδιο σκοπό, σε μια τέτοια δομή συνεργαζόμενων σε ένα συγκεκριμένο πλαίσιο, θεωρείται ότι έχουν τον ίδιο ρόλο. Η χρήση των ρόλων είναι παρούσα επίσης στη μοντελοποίηση των επιχειρηματικών διαδικασιών [7]. Εκεί υπάρχει υψηλός βαθμός συνεργασίας δεδομένου ότι συνήθως υπάρχουν αρκετοί συμμετέχοντες σε μέρη της διαδικασίας.

Επιπρόσθετα, στις αρχιτεκτονικές των πρακτόρων, η προσέγγιση μοντελοποίησης με ρόλους χρησιμοποιείται ως μέθοδος προσδιορισμού, ανάλυσης και σχεδίασης συστημάτων με βάση τους ρόλους τους οποίους οι πράκτορες θα παίξουν με έμφαση σε συγκεκριμένα χαρακτηριστικά όπως η αυτονομία και η ικανότητα συνεργασίας. Γενικά, οι ρόλοι είναι αναπαράσταση της συμπεριφοράς που είναι κοινή σε διαφορετικές κατηγορίες πρακτόρων. Επιπλέον, χρησιμεύει ως διεπαφή, μέσω της οποίας οι πράκτορες αποκτούν τις γνώσεις τους σχετικά με το περιβάλλον εκτέλεσης και το πως θα το επηρεάσουν.

Όπως συνάγεται από την σχετική βιβλιογραφία, η έννοια του ρόλου είναι μια κεντρική ιδέα για τη μοντελοποίηση της αλληλεπίδρασης των οντοτήτων και μπορεί να βοηθήσει στη μείωση της αυξημένης πολυπλοκότητας σε δυναμικά περιβάλλοντα. Επίσης, έχει χρησιμοποιηθεί για να περιγράψει συμπεριφορές αλληλεπίδρασης που επαναλαμβάνονται και θεωρούνται αρχιτεκτονικά πρότυπα που μπορούν να απεικονίσουν τις ομοιότητες υψηλού επιπέδου μεταξύ συναφών συστημάτων. Ενώ οι ρόλοι έχουν χρησιμοποιηθεί σε διαφορετικά μοντέλα, η σημασιολογία τους μπορεί να διαφέρει και η δυναμική τους να εξελιχθεί. Η συνεργασία των υπηρεσιών διαδικτύου κατά τη σύνθεσή τους μπορεί να περιγραφεί ως αντίστοιχη επαναλαμβανόμενη αλληλουχία αλληλεπιδράσεων και αυτή είναι η λογική που επιλέχθηκε η τεχνική των ρόλων για τη μοντελοποίηση των υπηρεσιών και την περιγραφή - επίλυση του προβλήματος της σύνθεσης.

1.2.3 Συνεργασία σε ανθρώπους και πράκτορες

Αρκετά μοντέλα έχουν αναπτυχθεί τις τελευταίες δεκαετίες ακολουθώντας κοινωνικο-ψυχολογικές προσεγγίσεις για τη δημιουργία ανθρώπινων ομάδων που περιλαμβάνουν διαφορετικές στρατηγικές και προσεγγίσεις. Μόνο μερικά από τα πλεονεκτήματα της εργασίας μέσω ομάδων εργασίας σε οργανισμούς είναι η αύξηση της παραγωγικότητας, της καινοτομίας και της ικανοποίησης των εργαζομένων.

Η έννοια του «ρόλου» από ψυχοκοινωνική άποψη είναι η συμπεριφορά που αναμένεται από ένα άτομο που καταλαμβάνει μια συγκεκριμένη θέση. Ως επακόλουθο, ένας ρόλος που υποδύεται ένα μέλος μιας ομάδας μπορεί να εκφραστεί από τη συμπεριφορά που εκφράζει κατά την κατάληψη της συγκεκριμένης θέσης. Αρκετές μελέτες ασχολήθηκαν με τον εντοπισμό και τη μέτρηση των παραγόντων που είναι πιθανό να αυξήσουν την αποτελεσματικότητα της ομαδικής εργασίας στους ανθρώπους. Τα μοντέλα ομαδικής εργασίας είναι η εξέλιξη των μεμονωμένων χαρακτηριστικών ομαδικής εργασίας που χρησιμοποιούνταν πριν την εφαρμογή των ρόλων [8,9]. Κάποια από τα βασικότερα μοντέλα ανθρώπινης ομαδικής εργασίας που εξετάστηκαν είναι: (1) Το μοντέλο των McCann and Margerison, (2) Το μοντέλο του *Belbin*, (3) Το μοντέλο του *Parker*, (4) Το μοντέλο του *Salas*, (5) Το μοντέλο του *Mathieu*. Στα μοντέλα αυτά και γενικότερα στη βιβλιογραφία προτείνονται διαφορετικοί συνδυασμοί ομαδικών ρόλων. Για παράδειγμα, οι οκτώ ρόλοι που απαιτούνται για τη δημιουργία μιας επιτυχημένης ομάδας, σύμφωνα με τον Margerison είναι οι (i) Creator-innovator, (ii) Explorer-promoter, (iii), Assessor-developer, (iv) Thruster-organizer, (v) Concluder-producer, Checker-inspector, Upholder-maintainer, Reporter-advisor. Για να έχουμε μια ολιστική άποψη των κύριων ρόλων στις ανθρώπινες ομάδες, βασιστήκαμε στις παραπάνω μελέτες και δημιουργήσαμε μια ταξινόμηση των ρόλων και των συμπεριφορών τους. Ο κύριος στόχος αυτής της εργασίας ήταν η δημιουργία μιας χαρτογράφησης της βασικής δομής των ρόλων που χρησιμοποιούνται σε ανθρώπινες ομάδες, προκειμένου να κατανοήσουμε καλύτερα τη δομή και τη συσχέτιση τους.

Επιπροσθέτα διαπιστώνουμε πως διερευνάται η συνεργασία, όπου τεχνολογικά νέοι τύποι οντοτήτων όπως πράκτορες, ρομπότ κ.λπ. λειτουργούν σε ομάδες και μιμούνται τον τρόπο που ενεργούν οι άνθρωποι ως μέλη της ομάδας. Όπως και με τους ανθρώπους, μια ομάδα πρακτόρων λογισμικού πρέπει να επιτελέσει συγκεκριμένες εργασίες οργανώνοντας τα μέλη της σύμφωνα με τα ατομικά χαρακτηριστικά και τον ρόλο τους στο σύστημα. Η αυξημένη έρευνα προς αυτή την κατεύθυνση ενθαρρύνεται από το βιομηχανικό περιβάλλον όπου η εργασία γίνεται σε ομάδες που αποτελούνται από μέλη ανθρώπους, πράκτορες και ρομπότ, και ο καθένας εξειδικεύεται σε συγκεκριμένες εργασίες. Για παράδειγμα, άνθρωποι και διάφοροι πράκτορες ηλεκτρονικού εμπορίου, που δημιουργήθηκαν από διαφορετικές επιχειρήσεις και έχουν διαφορετικά πρότυπα, πρέπει να συνεργαστούν, ενώ το επιτυχές αποτέλεσμα για την προσπάθεια εξαρτάται από την ομάδα και όχι από την ατομική προσπάθεια του μέλους. Στο πλαίσιο της Βιομηχανίας 4.0,

οραματιζόμαστε ένα μέλλον όπου, στις βιομηχανίες, οι άνθρωποι και τα CPSs συνεργάζονται αβίαστα ενώ παρουσιάζουν ομαδική συμπεριφορά.

Η ομαδική εργασία στους πράκτορες λογισμικού ορίζεται ως «ένα σύνολο πρακτόρων που έχουν έναν κοινό στόχο και μια κοινή «διανοητική κατάσταση». Στις αρχιτεκτονικές των πρακτόρων που είναι προσανατολισμένες στην ομάδα, οι πράκτορες έχουν καλύτερη κατανόηση για τα μέλη της ομάδας, δεδομένου ότι έχουν τη δυνατότητα να λαμβάνουν κρίσιμες αποφάσεις συλλέγοντας και μοιράζοντας προληπτικά πληροφορίες με τους συνεργάτες τους ενώ μπορούν να δημιουργήσουν συνεκτικές ομάδες με άλλους πράκτορες για την αποτελεσματική επίλυση προβλημάτων. Διερευνήθηκαν οι βασικές διαστάσεις που καθορίζουν την ομαδική εργασία των πρακτόρων προκειμένου να επιτευχθεί μια βέλτιστη συμπεριφορά ομαδικής εργασίας όπως η θεωρία της ομαδικής γνώσης (Shared Mental Models (SMM), η θεωρία της κοινής ψυχικής κατάστασης (Joint Intention (JIT)) και η Ad Hoc ομαδική εργασία. Σε δυναμικά περιβάλλοντα, μια ομάδα πρακτόρων είναι κάτι περισσότερο από μια ομάδα ευφών οντοτήτων που εργάζονται για την επίτευξη ενός στόχου. Ένας πράκτορας, προκειμένου να φτάσει σε ένα ορισμένο επίπεδο νοημοσύνης, αυτονομίας και συμπεριφοράς ομαδικής εργασίας, έχει ορισμένες ιδιότητες όπως η αντίδραση, η προληπτικότητα, η κοινωνική ικανότητα, η ικανότητα λήψης αποφάσεων κ.λπ. Όπως και άλλα μέλη της ομάδας, πρέπει να συντονιστούν χρησιμοποιώντας τη γνώση από κοινού, να συνεργαστούν δουλεύοντας μαζί στις ίδιες εργασίες και να επιδείξουν χαρακτηριστικά ομαδικής εργασίας.

Από τη μελέτη της βιβλιογραφία, αντλήθηκαν μια σειρά ερευνητικών θεμάτων για τις προκλήσεις που οδήγησαν την έρευνα σε αυτή τη διατριβή. Παρατηρήθηκε πως, στη μεγάλη πλειονότητα των προσεγγίσεων σύνθεσης υπηρεσιών, δίνετε έμφαση αποκλειστικά στο πρόβλημα της εύρεσης ενός πλάνου που περιέχει μια ακολουθία ενεργειών προς εκτέλεση. Αυτό όμως γίνεται ανεξάρτητα από τη γνώση που προέρχεται από τα δεδομένα του περιβάλλοντος που μπορεί να κατέχει μια υπηρεσία και θα μπορούσε ενδεχομένως να συνεισφέρει, για παράδειγμα σχετικά με το εάν ένα μήνυμα θα μπορούσε να πυροδοτήσει μια μη φυσιολογική κατάσταση ή εάν η αποστολή ενός μηνύματος πρέπει να αποφευχθεί λόγω ενός απροσδόκητου σφάλματος που έχει προκύψει. Επιπλέον, μια βασική αδυναμία που εντοπίστηκε, είναι ότι παρόλο που οι υπηρεσίες συνεργάζονται ως ομάδα για την επίτευξη ενός κοινού στόχου, δεν χρησιμοποιούν ικανότητες συνεργασίας, όπως συμβαίνει με ομάδες ανθρώπων, πρακτόρων ή ρομπότ.

1.3 Μοντελοποίηση των υπηρεσιών με μοντέλα ρόλων

Σε αυτή την ενότητα θα παρουσιάσουμε τη μοντελοποίηση του προβλήματος της σύνθεσης των σημασιολογικών υπηρεσιών ιστού με τη χρήση ρόλων για να αντιμετωπιστεί η πρόκληση της ετερογένειας και του μεγάλου αριθμού διαθέσιμων υπηρεσιών, εκμεταλλευόμενοι τις δυνατότητες που παρέχει η περιγραφή της συγκεκριμένης μεθόδου [10]. Με αυτό τον τρόπο, στοχεύουμε μεταξύ

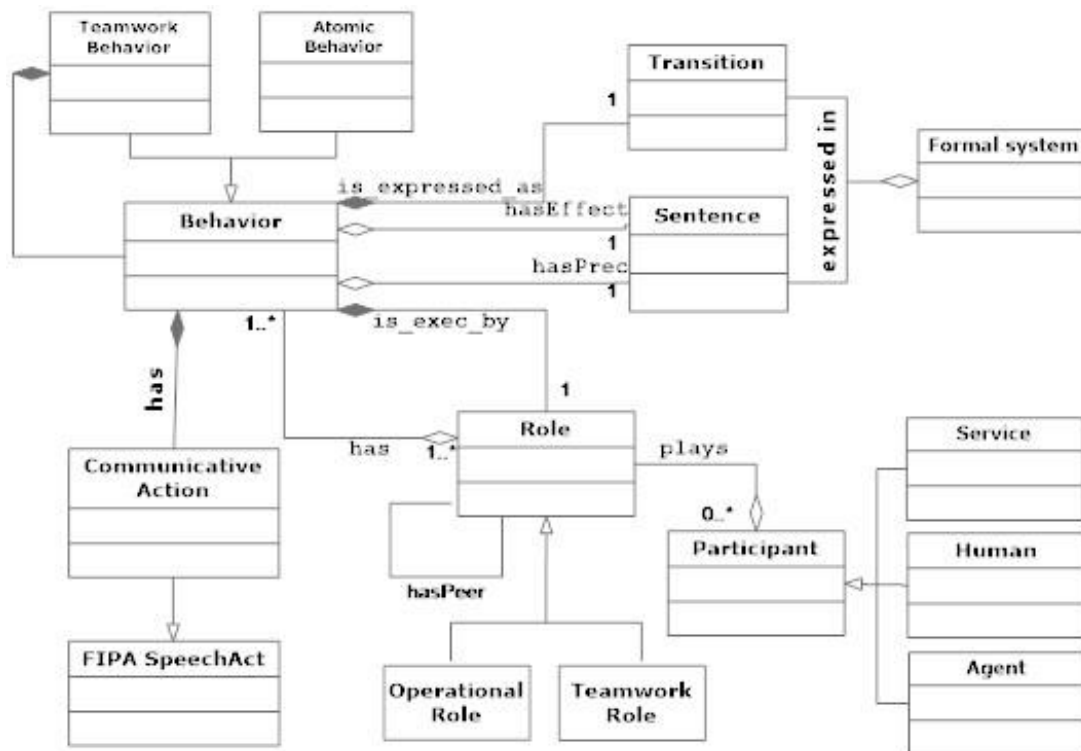
άλλων, να δημιουργήσουμε ένα μοντέλο που θα λειτουργεί ως μεσολαβητής για την κάλυψη του χάσματος διαλειτουργικότητας μεταξύ διαφορετικών παρόχων υπηρεσιών και επιπλέον, εκμεταλλευόμενοι τη συμπεριφορική περιγραφή, θα εμπλουτίσουμε τις υπηρεσίες με ομαδική συμπεριφορά.

1.3.1 Μοντελοποίηση με ρόλους σε Service Domains

Στην προσέγγισή μας για τη σύνθεση των υπηρεσιών με ρόλους, οι υπηρεσίες οργανώνονται σε τομείς όπως τομέας πληροφόρησης, τομέας αγορών, τομέας βιομηχανίας, κ.λπ.. Αυτοί οι τομείς, γενικά, παρέχουν αγαθά και υπηρεσίες και μπορούν να επιτύχουν μέσω των υπηρεσιών ιστού μια σειρά αφηρημένων στόχων. Προκειμένου να δημιουργηθεί ο τομέας και ξεκινώντας από το μηδέν, ο σχεδιαστής προσδιορίζει τους στόχους που αποτελούν και λόγο ύπαρξης του τομέα. Εφόσον καταγραφούν οι στόχοι, τότε αναλύονται οι υφιστάμενες ανάγκες και λύσεις και αναπτύσσονται σενάρια (περιπτώσεις χρήσης) που μπορούν να ανατεθούν προς υλοποίηση στους ρόλους του τομέα. Αυτό σημαίνει ότι για τη μοντελοποίηση θα πρέπει να απαντηθούν οι ακόλουθες ερωτήσεις: (i) Ποιο είναι το πεδίο εφαρμογής του τομέα, τι κάνει; (ii) Τι είδους διαδικασίες, χρήστες και πόροι εμφανίζονται σε αυτόν τον τομέα; (iii) Ποιοι στόχοι, ευθύνες και τι εργασίες απαιτούνται; (iv) Πώς αλληλεπιδρούν οι οντότητες;

Προκειμένου εννοιολογικά να παρουσιάσουμε τα παραπάνω, στο Σχήμα 1 δείχνουμε τα Role Model (RM) χρησιμοποιώντας διαγράμματα κλάσης για να απεικονίσουμε πώς αλληλοσυνδέονται τα στοιχεία του συστήματος. Οι κύριες έννοιες που χρησιμοποιούνται στην προσέγγιση ρόλων, είναι οι ρόλοι, οι επικοινωνιακές δράσεις και οι συμπεριφορές. Ουσιαστικά, υιοθετούμε τις κύριες ιδέες μοντελοποίησης με διαφορετική ερμηνεία σε σχέση με άλλες έρευνες στη μοντελοποίηση ρόλων, και τους δίνουμε το νόημα που παρουσιάζεται στους ορισμούς που ακολουθούν.

Μια επικοινωνιακή δράση (*communicative action - ca*) είναι ένας τύπος αλληλεπίδρασης μεταξύ δύο ρόλων και εκφράζει, κωδικοποιημένα, το λόγο της αλληλεπίδρασης. Παραδείγματα θα μπορούσαν να είναι: *acceptOrder*, *fail*, *inform*, *propose*, *queryIf*, *rejectInvoice*, *requestBalance*. Τα *ca* κληρονομούνται από τη FIPA-ACL προδιαγραφή που βασίζεται στη θεωρία της ομιλίας (*speech act theory*) [11]. Μια συμπεριφορά (*behavior - b*) εκφράζει μια πιθανή συνεργασία μεταξύ των ρόλων που επικοινωνούν χρησιμοποιώντας μια επικοινωνιακή δράση *ca*. Ένας ρόλος καθορίζει και περιγράφει μια συμπεριφορά, ενώ ένας συμμετέχων εντοπίζει και εφαρμόζει αυτήν τη συμπεριφορά



Σχήμα 1: Το UML διάγραμμα κλάσεων για το μοντέλο ρόλων (Role model).

Το διάγραμμα RM ενός πεδίου θα μπορούσε να χαρακτηριστεί ως το διαισθητικό μοντέλο που απεικονίζει τον τρόπο με τον οποίο οι υπηρεσίες αλληλεπιδρούν με άλλες υπηρεσίες. Μια υπηρεσία θεωρείται ως ένα σύνολο ρόλων και η συμπεριφορά της είναι ακριβώς αυτή των ρόλων που "υποδύεται". Ένα RM αποτελείται από ρόλους και τις συνεργασίες τους και περιλαμβάνει όλες τις πιθανές πτυχές επικοινωνίας του δεδομένου ρόλου και των συνεργασιών του με άλλους ρόλους. Ο τρόπος με τον οποίο ένας ρόλος αλληλεπιδρά με έναν άλλο ρόλο θεωρείται ως συμπεριφορά του πρώτου ρόλου και στην πραγματικότητα, απεικονίζει τον τρόπο επικοινωνίας μεταξύ τους. Ο ρόλος είναι ένα σύνολο συμπεριφορών και περιγράφει την παρατηρήσιμη συμπεριφορική πλευρά ενός συμμετέχοντα στο πεδίο και τις διαφορετικές πτυχές της συνεργασίας με άλλους ρόλους μέσω διαφορετικών συμπεριφορών. Αν και το διάγραμμα RM παρουσιάζει την πιθανή αλληλεπίδραση μεταξύ ρόλων μέσω της εκτέλεσης της κατάλληλης συμπεριφοράς, πρέπει, επιπλέον, να δηλώσουμε ρητά υπό ποιες συνθήκες ο ρόλος έχει την ικανότητα να εκτελεί μια συμπεριφορά. Αυτές εκφράζονται ως δηλώσεις προτάσεων και αποτελούν τους περιορισμούς των μεταβάσεων που συμβαίνουν στο πεδίο κατά την εκτέλεση μιας συμπεριφοράς. Ανά πάσα στιγμή, οι ρόλοι του πεδίου ανατίθενται σε υπηρεσίες, δηλ. μια υπηρεσία είναι ένας συμμετέχων που έχει τους ρόλους που της έχουν ανατεθεί, ενώ διαφορετικές υπηρεσίες μπορεί να έχουν τον ίδιο ρόλο.

Ορισμός 1 Ονομάζουμε πεδίο ρόλων ($\rho\Delta$) ένα πεδίο που έχει μοντελοποιηθεί με ρόλους και φιλοξενεί υπηρεσίες που ορίζονται μέσω των ρόλων που υποδύονται στον τομέα.

Δεδομένου ότι εστιάζουμε στον σημασιολογικό ιστό, ένα $\rho\Delta$ ορίζεται με τη χρήση οντολογιών, εφόσον οι πληροφορίες που προέρχονται από τον ορισμό και την εκτέλεση των υπηρεσιών ιστού πρέπει να είναι προσβάσιμες και κατανοητές από τις μηχανές χρησιμοποιώντας τεχνολογίες σημασιολογικού ιστού. Οι επιχειρηματικοί στόχοι και η επιχειρηματική λογική του πεδίου εκφράζονται με έννοιες της οντολογίας του πεδίου μέσω των στόχων, πλάνων και περιορισμών.

Λαμβάνοντας υπόψη την περιγραφή των υπηρεσιών με ρόλους, κάνουμε την ακόλουθη προσαρμογή στην προσέγγιση σύνθεσης, όπου το πρόβλημα σύνθεσης υπηρεσιών Ιστού θεωρείται ως πρόβλημα σχεδιασμού ενεργειών. Αντί να βρούμε ένα πλάνο ενεργειών όπως στον παραδοσιακό σχεδιασμό, αναζητούμε ένα πλάνο που αποτελείται από μια ακολουθία συμπεριφορών. Εκμεταλλευόμαστε την αφαίρεση που έχει επιτευχθεί με τη μοντελοποίηση ρόλων και μεταμορφώνουμε το πρόβλημα της εύρεσης μιας σύνθεσης ενεργειών στο πρόβλημα της εύρεσης των κατάλληλων συμπεριφορών που θα συντεθούν.

1.3.2 Ορισμοί των RM και των συστατικών τους

Ακολούθως, ορίζουμε μαθηματικά τα στοιχεία του RM ώστε να προχωρήσουμε στη σύνθεση των συμπεριφορών.

Ορισμός 2 Ένα μοντέλο ρόλων (ρM) του $\rho\Delta$ ορίζεται ως μία τριπλέτα

$$\rho M = \langle R, CA, B \rangle$$

όπου R αναπαριστά το σύνολο των ρόλων που εμφανίζονται στο RM του τομέα, CA αναπαριστά το σύνολο των επικοινωνιακών δράσεων και B αναπαριστά το σύνολο των συμπεριφορών στο $\rho\Delta$.

Ορισμός 3 Μια συμπεριφορά (b) είναι το διατεταγμένο ζεύγος

$$b = \langle ca, r_{peerRole} \rangle, \text{ όπου } ca \in CA \text{ και } r_{peerRole} \in R.$$

Μια συμπεριφορά υποδεικνύει ότι ένας ρόλος που περιέχει τη συγκεκριμένη συμπεριφορά έχει τη δυνατότητα να επικοινωνεί με έναν άλλο ρόλο $r_{peerRole}$ με πρόθεση για επικοινωνία ca .

Μια συμπεριφορά εκφράζει τη διαδικασία επικοινωνίας ενός ρόλου με έναν άλλο ρόλο. Η συμπεριφορά ανήκει στον ρόλο που ξεκινά την επικοινωνία. Για παράδειγμα, η συμπεριφορά $b1 = \langle invoice, rclient \rangle$ αντιπροσωπεύει την επικοινωνία μεταξύ του ρόλου που εκφράζει τη συμπεριφορά $b1$ με τον ομότιμο ρόλο $rclient$ με πρόθεση για επικοινωνία 'invoice', δηλαδή την τιμολόγηση του πελάτη.

Ορισμός 4 Ένας ρόλος r του ρM είναι ένα σετ συμπεριφορών

$$r = \{ b_i, \dots, b_j \}, \text{ όπου } b_i, \dots, b_j \in B.$$

Μέχρι στιγμής, η συζήτηση περιστράφηκε γύρω από τη δομή και τη φύση των ρόλων και των συμπεριφορών στο RM και δεν αναφέρθηκαν οι συνθήκες της εκτέλεσής τους. Οι κανονισμοί του πεδίου απαιτούν ορισμένες προϋποθέσεις να είναι απαραίτητες για την εκτέλεση μιας συμπεριφοράς και επιπλέον, μετά την εκτέλεση μιας συμπεριφοράς αποτυπώνονται τα αποτελέσματα της εκτέλεσής της. Οι προϋποθέσεις και τα αποτελέσματα είναι προτάσεις σε λογική γλώσσα που συμπίπτουν με τις προτάσεις που προέρχονται από την οντολογία του τομέα. Αυτές οι προτάσεις θα μπορούσαν εν μέρει να περιγράψουν την εξέλιξη των καταστάσεων του συστήματος. Οι προϋποθέσεις και τα αποτελέσματα των συμπεριφορών έχουν την ακόλουθη σημασιολογία: Προκειμένου μια συμπεριφορά να μπορεί να εκτελεστεί, η προϋπόθεση πρέπει να ισχύει και, μετά την επιτυχή εκτέλεση της θα παραχθούν τα αποτελέσματα που αυτή ορίζει.

1.3.3 Υπηρεσίες Ιστού και η διαδικασία της σύνθεσης

Ο ορισμός μιας υπηρεσίας βασίζεται στη σημασιολογία των ρόλων που υποδύεται, παρά στη δομή της. Μέσω του σημασιολογικού σχολιασμού, κάθε ρόλος στο $\mathcal{R}\Delta$ αναφέρεται σε ένα σημασιολογικό άτομο στην οντολογία περιεχομένου τομέα. Ένα σημασιολογικό άτομο με τη σειρά του είναι μέλος μιας κλάσης οντολογίας, συνδέοντας το στοιχείο ρόλου του $\mathcal{R}\Delta$ με μια σημασιολογική έννοια. Ως αποτέλεσμα, οποιοσδήποτε ρόλος r_i στο πλαίσιο του πεδίου αντιστοιχίζεται σε ένα σύνολο εννοιών της οντολογίας πεδίου και οποιαδήποτε υπηρεσία s_i σε ένα σύνολο εννοιών ρόλου. Οι σημασιολογικές πληροφορίες καθιστούν δυνατή την παραγωγή συμπεράσματος σχετικά με την εκτέλεση κάθε ατομικής υπηρεσίας ως βάση για τη σημασιολογική αντιστοίχιση μεταξύ της κατάστασης που προέκυψε μετά την εκτέλεση μιας συμπεριφοράς και των προϋποθέσεων που απαιτούνται για την εκτέλεση μιας άλλης.

Ορισμός 5 Μια υπηρεσία $s \in S$ είναι το σύνολο των ρόλων που υποδύεται στο $\mathcal{R}\Delta$:

$$s = \{ r_i, \dots, r_j \}, \text{ όπου } r_i, \dots, r_j \in R.$$

Το επίκεντρο της σύνθεσης υπηρεσιών έγκειται στο κατάλληλο διαταγμένο σύνολο συμπεριφορών που πρέπει να εκτελεστούν και όχι στις υπηρεσίες, καθώς θα μπορούσαν να χρησιμοποιηθούν διαφορετικές υπηρεσίες που έχουν ρόλους που περιέχουν τις αναζητούμενες συμπεριφορές. Έτσι, σύμφωνα με τον ορισμό, η σύνθεση υπηρεσίας διαδικτύου συνάγεται από τη σύνθεση συμπεριφοράς.

Ορισμός 6 Οι δηλώσεις στόχων εκφράζονται μέσω της οντολογίας του πεδίου και περιγράφουν τα επιτεύγματα που μπορεί να επιτύχει ο τομέας.

Οι ζητούμενοι στόχοι αφορούν συνήθως συνδυασμούς αρχικών στόχων της οντολογίας τομέα, οι οποίοι αναμένεται να εκφράσουν ένα αποτέλεσμα που ένας χρήστης του πεδίου επιθυμεί να επιτύχει.

Ένα αντικειμενικό πλάνο είναι ένα πλάνο που θα μπορούσε να ακολουθηθεί για την επίτευξη ενός στόχου που καθορίζεται στον τομέα. Είναι αποθηκευμένο στη βιβλιοθήκη πλάνων και είναι ένα μοτίβο συμπεριφοράς που εκφράζει επαναχρησιμοποιήσιμες λύσεις σε ένα κοινό πρόβλημα του τομέα. Ένα αίτημα χρήστη μπορεί να εκφραστεί καθορίζοντας τον επιθυμητό στόχο που πρέπει να επιτευχθεί, έχοντας μια αρχική και μια κατάσταση στόχου. Η αρχική κατάσταση που παρέχεται εν μέρει από τον χρήστη εκφράζει παραμέτρους που απαιτούνται για τη σύνθεση. Η κατάσταση στόχου είναι μια πρόταση που περιγράφει τι πρέπει να παρέχεται από μια επιτυχημένη σύνθεση. Η λύση, προκειμένου να βρεθεί ένα αντικειμενικό πλάνο, πρέπει να είναι σε θέση να παρέχει ένα πλάνο όπου ικανοποιείται η κατάσταση του στόχου. Ορίζουμε τυπικά ένα αίτημα σύνθεσης q ως εξής:

Ορισμός 7 Ένα αίτημα σύνθεσης είναι ένα ζεύγος που περιέχει τις προϋποθέσεις και τα αποτελέσματα της σύνθετης συμπεριφοράς που αναζητούμε να εκτελεστεί. Το αίτημα

$$q_x = \langle q_{x_{initial}}, q_{x_{goal}} \rangle$$

είναι ένα ερώτημα για μια σύνθετη συμπεριφορά που θα εκτελεστεί σε μια κατάσταση που το $q_{x_{initial}}$ ισχύει και σύμφωνα με τη λύση που ψάχνουμε το $q_{x_{goal}}$ θα ικανοποιείται στην προκύπτουσα κατάσταση.

Ορισμός 8 Η συντιθέμενη συμπεριφορά (bc) είναι μια πλειάδα $bc = \langle b_1, \dots, b_n \rangle$, όπου $b_1, \dots, b_n \in B$.

Τα στοιχεία της πλειάδας αποτελούν τις συντιθέμενες συμπεριφορές. Η σειρά που εμφανίζονται οι συμπεριφορές στην πλειάδα καθορίζει και τη σειρά εκτέλεσής τους.

1.3.4 Ο αλγόριθμος για τη σύνθεση

Μετατρέπουμε το πρόβλημα σύνθεσης ως πρόβλημα σχεδιασμού ενεργειών όπου, δεδομένης μιας αρχικής και μιας κατάστασης στόχου, αναζητούμε μια ακολουθία συμπεριφορών, που όταν εκτελούνται ξεκινώντας από την αρχική κατάσταση το πλάνο θα οδηγήσει στην κατάσταση του στόχου. Χρησιμοποιώντας μια τροπική λογική, για παράδειγμα το σύστημα λογικής PDL[12] για την κωδικοποίηση του προβλήματος με την προσέγγιση του σχεδιασμού της ικανοποιησιμότητας ενός συνόλου προτάσεων, μια λύση σε αυτό το πρόβλημα είναι ένα πλάνο π που αποτελεί μια σύνθεση ενεργειών $A_x \rightarrow (\varphi_0 \rightarrow \langle \pi \rangle \varphi_g)$ όπου A_x αποτελείται από ένα σύνολο προτάσεων, που ονομάζονται έξτρα λογικά αξιώματα, και κάθε ένα περιγράφει τις προϋποθέσεις και τα αποτελέσματα κάθε εφικτής ενέργειας [13]. Η διαισθητική έννοια του παραπάνω τύπου είναι ότι, δεδομένων των προϋποθέσεων που περιγράφονται από τα αξιώματα A_x σε μια αρχική κατάσταση που περιγράφεται μερικώς από τη φ_0 , μετά την εκτέλεση της σύνθετης συμπεριφοράς π φτάνουμε σε μια κατάσταση φ_g , όπου επιτυγχάνεται ο στόχος της σύνθετης υπηρεσίας. Η λύση στο πρόβλημα σχεδιασμού που αναζητάμε, όπως περιγράφεται στην παραπάνω λογική πρόταση, θα

μπορούσε να παραχθεί από έναν αποδεικτική θεωρημάτων και είναι στην πραγματικότητα η ακολουθία ατομικών συμπεριφορών που πρέπει να εκτελεστούν.

Ο βασικός αλγόριθμος σύνθεσης που ακολουθεί, δείχνει την τακτική που μπορεί να χρησιμοποιηθεί για την εξαγωγή ενός πλάνου για ένα συγκεκριμένο αίτημα, ώστε να συντίθενται υπηρεσίες που περιγράφονται με τη μέθοδο των ρόλων. Δέχεται ως εισαγωγή το πεδίο $q\Delta$ και το αίτημα q_x που έχει τις προϋποθέσεις και τα αποτελέσματα που αντιστοιχούν στην περιγραφή της αρχικής κατάστασης και του στόχου που θα επιτευχθεί αντίστοιχα. Επιστρέφει το σύνολο των υπηρεσιών που μπορούν να σχηματίσουν τη σύνθετη υπηρεσία μαζί με το ρόλο που κάθε υπηρεσία έχει στο RM και τη συμπεριφορά που εκτελούν για την επίτευξη του στόχου.

Algorithm 1. Returns a service that corresponds to a query

Input : the domain $q\Delta$, a request q_x for a service

Output : A set of services for the composition or null when no solution exists

```

1  Begin
2     $p \leftarrow q\Delta.planLibrary$ 
3     $plans \leftarrow acquirePlan(q_x, p)$ 
4    if  $plans = \emptyset$  then
5       $plans \leftarrow findPlan(q\Delta, q_x)$ 
6    end if
7    if  $plans \neq \emptyset$  then
8       $optp \leftarrow evaluate(plans)$ 
9       $behav\_inv \leftarrow extractBehaviors(optp)$ 
10      $roles\_inv \leftarrow extractRoles(behav\_inv)$ 
11      $serv\_inv \leftarrow extractServ(roles\_inv)$ 
12     return  $serv\_inv$ 
13   else return null
14   end if
15  End

```

Αλγόριθμος 1: Ο αλγόριθμος για τη σύνθεση.

Έχουμε εισαγάγει τη μέθοδο σύνθεσης των υπηρεσιών διαδικτύου μέσω της μοντελοποίησης τους ως ρόλους. Το πιο διακριτικό χαρακτηριστικό του προτεινόμενου μοντέλου, μέχρι αυτό το στάδιο, έγκειται στην ικανότητα του να χειρίζεται υπηρεσίες ομαδοποιημένα σε σύγκριση με άλλες μεθόδους που επίσης βασίζονται στον σχεδιασμό AI. Από την άποψη της δυνατότητας εφαρμογής, η ικανότητα ομαδοποίησης είναι ζωτικής σημασίας για ένα τέτοιο πρόβλημα, καθώς καθορίζει εάν μπορεί να ανταποκριθεί ένα αίτημα σύνθεσης εντός ενός εύλογου χρονικού διαστήματος. Επόμενος στόχος μας είναι η επέκταση του τυπικού τρόπου σύνθεσης υπηρεσιών χρησιμοποιώντας ένα πλάνο που εξάγεται από το RM, προκειμένου να αναπτύξουμε συνεργαζόμενες υπηρεσίες που θα έχουν και ρόλους ομαδικής εργασίας για να υποστηρίξουν τη σύνθεση.

1.4 Συνεργατική συμπεριφορά

Μελετήσαμε τους βασικούς παράγοντες που επηρεάζουν τη συμπεριφορά σε μοντέλα ομαδικής εργασίας σε πράκτορες και δείχνουμε πως αυτά συσχετίζονται με την ανθρώπινη ομαδική εργασία και πως μπορούν να ωφεληθούν στη σύνθεση υπηρεσιών.

1.4.1 Η ομαδική εργασία στους πράκτορες

Πραγματοποιήθηκε βιβλιογραφική ανασκόπηση των σχετικών μελετών και ως αποτέλεσμα, προέκυψε μια σειρά παραγόντων που επηρεάζουν τη συμπεριφορά της ομαδικής εργασίας. Από την ανάλυση, ξεχωρίζουν οι παράγοντες που στην πλειονότητα των μελετών παραδοσιακά κερδίζουν την προσοχή, ενώ υπάρχουν και νεοεμφανιζόμενοι παράγοντες που βρίσκουμε σε πρόσφατες μελέτες και αφορούν μοντέλα που προσανατολίζονται σε δυναμικά και ρεαλιστικά περιβάλλοντα και αναδεικνύουν νέες πτυχές στη συμπεριφορά της ομαδικής εργασίας. Οι δεκαπέντε παράγοντες που προέκυψαν από τη βιβλιογραφική μελέτη παρουσιάζονται σε τρεις κατηγορίες. Γενικά, αυτοί χαρακτηρίζουν (i) την οργάνωση της ομάδας, π.χ. δημιουργώντας κανόνες επικοινωνίας, προτεραιότητας (ii) τη διαμόρφωση στρατηγικής της της ομάδας για μελλοντική κατεύθυνση, π.χ. με το σχεδιασμό και τη λήψη αποφάσεων και (iii) τη λειτουργία της ομάδας για να επιτύχει συνέργεια, π.χ. κανόνες εμπιστοσύνης και εμπλοκής [14].

- A. Οργανωτικοί παράγοντες: (1) Collaboration (COL), (2) Communication (COM) και (3) Coordination (COO)
- B. Παράγοντες που σχετίζονται με τη στρατηγική: (1) Planning (PLN), (2) Learning (LRN), (3) Decision making (DM), (4) Evaluation (EVL), (5) Teamwork policies (POL).
- C. Παράγοντες που σχετίζονται με τη συνέργεια: (1) Ad hoc (ADH), (2) Autonomy (AUT), (3) Delegation (DLG), (4) Joint-intention (INT), (5) Knowledge of teammate's capabilities (KCT), (6) Knowledge sharing (KNL), (7) Trust (TRS).

1.4.2 Ρόλοι συνεργασίας στις υπηρεσίες

Με κίνητρο την οργανωτική συμπεριφορά και τη θεωρία της ομαδικής εργασίας και προκειμένου να επωφεληθούμε από τις ικανότητες αυτές στη σύνθεση των υπηρεσιών, υποστηρίζουμε ότι οι υπηρεσίες εκτός από τη λειτουργική συμπεριφορά τους θα πρέπει επίσης να παρουσιάζουν συμπεριφορά ομαδικής εργασίας για να συνεργάζονται. Σε περίπτωση που δεν υπάρχουν υπηρεσίες που ήδη έχουν τέτοια συμπεριφορά, τότε άλλες υπηρεσίες πρέπει να συμπεριληφθούν στη σύνθεση προκειμένου να παρέχεται αυτή η συμπεριφορά στη συντιθέμενη υπηρεσία. Μέσα από αυτή, αποκτούν ικανότητα καλύτερης συνεργασίας, διότι κατά την εκτέλεση τους σε ρεαλιστικά συστήματα στον πραγματικό κόσμο οι πραγματικές μεταβάσεις δεν ακολουθούν πάντα τις προγραμματισμένες.

Για να το επιτύχουμε αυτό και με βάση το μοντέλο ρόλων που παρουσιάστηκε για τις υπηρεσίες, ενισχύουμε το μοντέλο με ρόλους ομαδικής εργασίας. Προτείνουμε οκτώ ρόλους ομαδικής εργασίας υπηρεσιών που θα συμμετέχουν στη διαδικασία σύνθεσης [15]. Αυτοί οι ρόλοι αποσκοπούν στο (i) να επιτρέψουν στις υπηρεσίες να χειραγωγούν τις αβεβαιότητες με την προληπτική δράση κατά τη συμμετοχή τους σε συγκεκριμένες δραστηριότητες, (ii) να συλλάβουν πιθανή αναδυόμενη συμπεριφορά που συμβαίνει σε ένα πεδίο κατά τη διάρκεια του σχεδιασμού και της εκτέλεσης της υπηρεσίας, (iii) να προλάβουν τυχόν ανεπιθύμητη συμπεριφορά άλλων συμμετεχόντων που εμφανίζονται στον τομέα, (iv) να ενεργούν ως ο καταλύτης που απαιτείται για τον συνδυασμό και την αναδιάρθρωση κατάλληλων υπηρεσιών σε δυναμικά και απρόβλεπτα περιβάλλοντα.

Πιο συγκεκριμένα, συνάγουμε τη συμπεριφορά για κάθε ρόλο ομαδικής εργασίας, συνδέοντας την επιθυμητή συμπεριφορά των μελών της ομάδας με τους παράγοντες και τα συμπεράσματα που προέκυψαν από την μελέτη της ομαδικής εργασίας σε ανθρώπους και πράκτορες.

Ο ρόλος του *Προγραμματιστή*: Ο σκοπός του ρόλου *Προγραμματιστή* είναι να συνθέτει νέες υπηρεσίες. Είναι ο "δημιουργός των ομάδων" του τομέα. Δεσμεύει τις κατάλληλες υπηρεσίες που υποδύονται τους απαιτούμενους λειτουργικούς και οργανωτικούς ρόλους και έχουν τις συμπεριφορές που πρέπει να εκτελεστούν για τη σύνθεση της υπηρεσίας.

Ο ρόλος του *Διοργανωτή*: Κατά την ανάπτυξη μιας σύνθετης υπηρεσίας, είναι σημαντικό να παρακολουθείται ο τρόπος με τον οποίο οι διάφορες υπηρεσίες συνεργάζονται προς τον κοινό στόχο. Κάθε υπηρεσία που χρησιμοποιείται είναι πιθανόν ένα κομμάτι μιας μεγαλύτερης λύσης. Η ακολουθία συμπεριφορών που προέρχεται από το πλάνο αποτελεί το πρότυπο στο οποίο οι υπηρεσίες συνεργάζονται προς τον κοινό στόχο.

Ο ρόλος του *Επιβλέποντα*: Σε δυναμικά περιβάλλοντα όπου πολλά μπορούν να αλλάξουν, μερικές από τις δυνατότητες που έχει ο ρόλος του *Επιβλέποντα* είναι ότι, μπορεί να δημιουργήσει σταθερότητα στο συγκεκριμένο περιβάλλον, διατηρώντας ένα υψηλό επίπεδο επιτήρησης. Οι κανονισμοί υπό τους οποίους λειτουργεί ένας τομέας και πρέπει να παρακολουθούνται απεικονίζονται ως κανόνες, την εποπτεία των οποίων αναλαμβάνει αυτός ο ρόλος.

Ο ρόλος του *Συμβούλου*: Μετά από ένα αίτημα χρήστη για μια υπηρεσία, ο *Σύμβουλος* του πεδίου που το λαμβάνει αναζητά εάν αυτό ταιριάζει με τους υπάρχοντες αποθηκευμένους στόχους του πεδίου σύμφωνα με τη σημασιολογική του ερμηνεία. Εάν είναι έτσι, ανακτά ένα πλάνο και ο ρόλος του προγραμματιστή προχωρά στην κατανομή των ρόλων σε υπηρεσίες. Διαφορετικά, ενημερώνει τον ρόλο του προγραμματιστή να αναζητήσει ένα νέο πλάνο. Επεμβαίνει για να προτείνει λύσεις όταν υπάρχει απόκλιση από το πλάνο.

Ο ρόλος του *Καινοτόμου*: Ο ρόλος του *Καινοτόμου* διερευνά νέους τρόπους στη διαδικασία σύνθεσης υπηρεσιών. Αναζητά νέα πλάνα εργασιών, ελέγχει σύγχρονες αναβαθμισμένες υπηρεσίες

ώστε να αντικατασταθούν οι ξεπερασμένες, διερευνά δυνατότητες και αναζητά καινοτόμες λύσεις από νέες αναπτυσσόμενες υπηρεσίες στο Διαδίκτυο.

Ο ρόλος του *Υποστηρικτή*: Ο *Υποστηρικτής* λειτουργεί σε διπλή βάση, αναζητά νέες ευκαιρίες και διερευνά δυνατότητες εντός και εκτός του τομέα. Επιτρέπει στους πελάτες του συστήματος να ανακαλύψουν νέες προσφερόμενες υπηρεσίες που θα μπορούσαν να χρησιμοποιηθούν. Ζητά ανατροφοδότηση από τους ρόλους του Οργανωτή και του Παραγωγού ώστε εάν το σύστημα είναι υπερφορτωμένο, να βρει νέες λύσεις, π.χ. άλλους πόρους εντός ή εκτός του πεδίου που θα μπορούσαν να βοηθήσουν.

Ο ρόλος του *Παραγωγού*: Ο ρόλος του *Παραγωγού* λειτουργεί με συστηματικό τρόπο ώστε να παραγάγει αποτελέσματα συγκεκριμένης εργασίας. Ακριβώς όπως ένα εργοστάσιο στον πραγματικό κόσμο διαθέτει παραγωγικές υπηρεσίες, έτσι και ο σκοπός των υπηρεσιών που έχουν αυτόν τον ρόλο είναι η παραγωγή. Είναι ο «παραγωγός» του συστήματος, αυτός ο ρόλος χαρακτηρίζει για παράδειγμα, όλες τις κατασκευαστικές υπηρεσίες. Ενημερώνει τον *Υποστηρικτή* που αναζητά ευκαιρίες προώθησης της χρήσης των.

Ο ρόλος του *Συντηρητή*: Μια υπηρεσία σχεδιάζεται, καλείται, ενημερώνεται και συντηρείται από άλλες ομότιμες υπηρεσίες. Κάθε υπηρεσία στο πεδίο βασίζεται στις υπηρεσίες που έχουν τον ρόλο του *Συντηρητή* για τη συντήρησή τους. Ο ρόλος αυτός εκτός από τη συντήρηση των υπηρεσιών συντηρεί επίσης τη λειτουργικότητα της ομάδας. Ο τομέας, μέσω αυτού του ρόλου, μπορεί να παρέχει σύνθετες υπηρεσίες ομαδικής εργασίας με ενσωματωμένο έλεγχο ακεραιότητας.

1.4.3 Σύνθεση υπηρεσιών με τη βοήθεια των ρόλων συνεργασίας

Μετά από ένα αίτημα χρήστη για μια υπηρεσία, η οποία εκφράζεται με μια πρόταση που περιγράφει το στόχο της υπηρεσίας, ο εμπλεκόμενος ρόλος του συμβούλου του πεδίου παίρνει το αίτημα και αναζητά αν ταιριάζει σύμφωνα με τη σημασιολογική ομοιότητά του με τους υπάρχοντες επιχειρηματικούς στόχους της βιβλιοθήκης πλάνων του τομέα. Οι στόχοι στο αποθετήριο συνδέονται με προκαθορισμένα πλάνα που αποτελούν τυποποιημένα πλάνα και μπορούν να επιτευχθούν με την εκτέλεση κάποιας ατομικής ή σύνθετης υπηρεσίας. Εφόσον υπάρχει έτοιμο τέτοιο πλάνο, ανακτά όλες τις δυνατές λύσεις. Στη συνέχεια, ο ρόλος του συμβούλου κάνει μια αξιολόγηση για να βρει το καταλληλότερο πλάνο σύμφωνα με την τρέχουσα κατάσταση του πεδίου και ενημερώνει τον ρόλο του προγραμματιστή, σχετικά με το σχέδιο που θα ακολουθηθεί ώστε να συνεχίσει με την ανάθεση των υπηρεσιών και τη δημιουργία της ομάδας. Εάν δεν υπάρχει έτοιμο προκαθορισμένο πλάνο, ενημερώνει τον προγραμματιστή να αναζητήσει ένα νέο πλάνο συμπεριφορών που όταν αυτές θα εκτελεστούν διαδοχικά θα οδηγήσουν στην κατάσταση του στόχου. Τα παραγόμενα πλάνα αξιολογούνται και στη συνέχεια λαμβάνεται η ακολουθία συμπεριφορών του επιλεγμένου πλάνου που θα εκτελεστούν και σύμφωνα με το RM του πεδίου

αναζητούνται ρόλοι που μπορούν να εκτελέσουν αυτές τις συμπεριφορές. Βρίσκει τις κατάλληλες υπηρεσίες που μπορούν να παίξουν αυτούς τους ρόλους και κάνει την ανάθεση κάθε συμπεριφοράς, και έτσι προκύπτει τελικά η ομάδα που θα εκτελέσει τη σύνθετη υπηρεσία. Κατά τη διάρκεια της ανάθεσης, εάν μια υπηρεσία δεν είναι διαθέσιμη, ο ρόλος του προγραμματιστή προχωρά την ανάθεση σε νέα υποψήφια υπηρεσία. Ο ρόλος του προγραμματιστή στην πραγματικότητα κάνει τη δημιουργία της ομάδας, καθώς κάνει την κατανομή των ρόλων σε υπηρεσίες που έχουν και τους λειτουργικούς ρόλους του τομέα, δηλαδή ρόλο λογιστή, ρόλο διανομέα, κ.λπ. Μετά την ολοκλήρωση της κατανομής, ο ρόλος του προγραμματιστή καλεί τον ρόλο του επιβλέποντα που θα επιβλέπει τον τρόπο με τον οποίο η ομάδα των επιλεγμένων υπηρεσιών προχωρά με τη σύνθεση.

Οι σύνθετες υπηρεσίες που, εκτός από τη λειτουργική τους συμπεριφορά, έχουν επίσης συμπεριφορά ομαδικής εργασίας δεν είναι απλώς ένα συγκεκριμένο σύνολο πηγαίου κώδικα που θα τοποθετηθεί στο λογισμικό που κατασκευάζεται η υπηρεσία, για παράδειγμα μια βιβλιοθήκη Java. Οι ρόλοι της ομαδικής εργασίας ορίζονται αρχικά εννοιολογικά. Αποτελούν αποκτηθείσα γνώση που οι ειδικοί του πεδίου θα μπορούν να εφαρμόσουν κατά τη σύνθεση υπηρεσιών ώστε να την καθοδηγήσουν και να την καταστήσουν πιο αποτελεσματική και επαναχρησιμοποιήσιμη. Κατά την εκτέλεση υπηρεσιών, το σύστημα συναντά συχνά τα ίδια προβλήματα πολλές φορές. Υπάρχουν διάφορες λύσεις και τρόποι αντιμετώπισης αυτών των επαναλαμβανόμενων προβλημάτων, αλλά με την πάροδο του χρόνου ορισμένες λύσεις προτιμώνται από άλλες, επειδή είναι πιο ευέλικτες ή ευκολότερα υλοποιήσιμες. Οι ρόλοι της ομαδικής εργασίας που παρέχονται μέσω προτύπων ρόλων αποτελούν πρακτική λύση σε τέτοιου είδους προβλήματα.

Οι προτεινόμενοι ρόλοι συνεργασίας αποτελούν ένα βήμα προς τον ορισμό ενός μοντέλου που θα επιτρέψει την αναβάθμιση των συντιθέμενων υπηρεσιών σε πιο έξυπνες, έτσι ώστε να αλληλεπιδρούν μεταξύ τους αυτόνομα, να ανακαλύπτουν άλλες αξιόπιστες υπηρεσίες εντός του τομέα, να μεταβιβάζουν τυχόν αναθέσεις τους σε άλλες παρόμοιες αξιόπιστες υπηρεσίες και να δηλώνουν με τυπικό τρόπο τις δυνατότητές τους στον πεδίο που ανήκουν.

1.5 Η Οντολογία για τις TrEW υπηρεσίες

Η ενότητα αυτή περιγράφει την εννοιολογική μοντελοποίηση της ομαδικής εργασίας των υπηρεσιών με τυπικό τρόπο, κάνοντας χρήση της οντολογίας TrEWSOnto (Teamwork Enabled Web Service Ontology) που δημιουργήθηκε για το σκοπό αυτό, ώστε να διευκολύνει τη διαδικασία ανάπτυξης υπηρεσιών με δυνατότητες συνεργασίας. Η συγκεκριμένη οντολογία, αποτελεί μια δομημένη κατηγορία εννοιών που αποτελούν το αντικείμενο της ομαδικής εργασίας στις υπηρεσίες, δηλ. τους ρόλους ομαδικής εργασίας που ορίστηκαν και πώς αυτοί σχετίζονται με τις έννοιες των υπηρεσιών και της διαδικασίας της σύνθεσης. Πιο συγκεκριμένα, κατασκευάσαμε ένα δίκτυο οντολογιών, προκειμένου να τις διαχειρίζονται διαφορετικοί χρήστες από διαφορετικά συστήματα, πιθανόν σε διαφορετικούς οργανισμούς. Αυτό περιλαμβάνει έννοιες όπως υπηρεσία, συμπεριφορά,

πλάνο, στόχος, συμμετέχων, κ.λπ.. Συνεπώς, χωρίζουμε τις θεμελιώδεις έννοιες σε πέντε οντολογίες περιεχομένου για να διευκολύνουμε την επαναχρησιμοποίηση αυτών [17].

1.5.1 Οι συνιστώσες οντολογίες

Οι πέντε οντολογικά ανεπτυγμένες ενότητες είναι: RoleModel, Team, Sensing, Service and RoleDomain Ontology. Τα προθέματα rm, tm, sen, ser και rdm αναφέρονται στα URI των προαναφερθέντων οντολογιών, αντίστοιχα. Στον Πίνακα 1 δίνουμε μια σύντομη περιγραφή για κάθε οντολογία, ενώ στις επόμενες ενότητες παραθέτουμε λεπτομέρειες σχετικά με τις κλάσεις, τις ιδιότητες και τα αξιώματα της κάθε οντολογίας.

Πίνακας 1: Οι πέντε ενότητες της οντολογίας των TrEW υπηρεσιών.

Ενότητα	Περιγραφή
RoleModel	Είναι η οντολογία που περιγράφει το RoleModel του τομέα.
Team	Η Team οντολογία καθορίζει τις έννοιες που σχετίζονται με την ομαδική εργασία του μοντέλου. Για κάθε ρόλο ομαδικής εργασίας εντοπίζονται οι πιθανές προκλήσεις που θα πρέπει να ανταπεξέλθει στον πεδίο αρμοδιότητάς του.
Sensing	Σε αυτή την οντολογία ρυθμίζονται οι αισθητήρες, που θα ακροάζονται τις παρατηρήσεις που λαμβάνουν χώρα στον τομέα.
RoleDomain	Εδώ περιγράφεται το πεδίο του προβλήματος, με την ιεραρχία των εννοιών, των ιδιοτήτων και των κανόνων που σχετίζονται με το πλαίσιο του.
Service	Σε αυτή την οντολογία ορίζονται οι υπηρεσίες με ομαδική συμπεριφορά που συντίθενται από άλλες υπηρεσίες που είναι εγγεγραμμένες στο πεδίο του προβλήματος.

Η RoleModel Οντολογία

Η οντολογία RoleModel είναι αυτή που αντιπροσωπεύει το RM του τομέα, δηλαδή τους διαθέσιμους ρόλους που μπορούν να χρησιμοποιηθούν για την αναπαράσταση των υπηρεσιών και των συμπεριφορών τους. Ενδεικτικές έννοιες και ιδιότητες αυτής είναι:

- rm:Role: Αντιπροσωπεύει μια κλάση της οποίας τα άτομα είναι οι ρόλοι που μπορούν να παίξουν οι συμμετέχοντες στην ομάδα, π.χ. Πελάτης, Προμηθευτής.

$$\text{rm:Client} \sqsubseteq \text{rm:Role}$$

- rm:ComAct: Αυτή η κλάση αποθηκεύει τις πληροφορίες σχετικά με τους λόγους που ένας ρόλος επιδιώκει την επικοινωνία με έναν άλλο ρόλο. Επεκτείνει την έννοια της επικοινωνιακής δράσης που ακολουθείται στη FIPA speech act theory.
- rm:Behavior: Αυτή είναι η θεμελιώδης κλάση με την οποία μπορεί να οριστεί για κάθε ρόλο με ποιους άλλους ρόλους επικοινωνεί. Αυτή η κλάση αξιοποιείται από την κλάση ρόλων, δεδομένου ότι ένας ρόλος ορίζεται ως ένα σύνολο συμπεριφορών που υποδηλώνουν την ικανότητά του να επικοινωνεί με άλλους.

- **rm:Precondition:** Είναι η κλάση που περιγράφει τις προϋποθέσεις που πρέπει να ισχύουν για να εκτελεστεί μια συμπεριφορά. $rm:Precondition \sqsubseteq lg:LogicalSentence$
- **rm:Effect:** Αντίστοιχα, αυτή η κλάση περιγράφει τα αποτελέσματα της εκτέλεσης της συμπεριφοράς. $rm:Effect \sqsubseteq lg:LogicalSentence$
- **rm:ActingRole:** Δείχνει τον ενεργό ρόλο που εκτελεί μια συμπεριφορά σε έναν άλλο συμμετέχοντα ρόλο.

$$ActingRole \sqsubseteq rm:Role \sqcap \exists rm:executesBehaviorTo.rm:ConfereeRole.$$

- **rm:ConfereeRole:** Δείχνει τον ρόλο που συμμετέχει σε μια συνεργασία.

$$ConfereeRole \sqsubseteq rm:Role \sqcap \exists rm:receivesBehaviorFrom.rm:ActingRole.$$

- **rm:hasBehavior:** Δείχνει τη συμπεριφορά που έχει ένας ρόλος. Επιπλέον, αυτή η ιδιότητα επεκτείνεται στην πιο εξειδικευμένη ιδιότητα που ονομάζεται **tm: hasTeamBehavior**, που ορίζεται ώστε να συμπεριλάβει και τις ομαδικές συμπεριφορές.
- **rm:executesBehaviorTo:** δείχνει τη σχέση μεταξύ δύο συνεργαζόμενων ρόλων, του αποστολέα και του παραλήπτη μιας συμπεριφοράς.

Είναι επίσης η αντίστροφη της σχέσης **rm:receivesBehaviorFrom**.

$$rm:receivesBehaviorFrom \equiv rm:executesBehaviorTo^{-}$$

- **rm:hasPrecondition:** Αυτή η ιδιότητα υποδεικνύει ότι η συμπεριφορά έχει μια προϋπόθεση που πρέπει να ικανοποιηθεί ώστε να εκτελεστεί.
- **rm:hasEffect:** Αυτή η ιδιότητα αντίστοιχα, υποδηλώνει το αποτέλεσμα μιας συμπεριφοράς.

Η Team οντολογία

Η Team οντολογία στοχεύει στον καθορισμό των εννοιών που σχετίζονται με την ομαδική εργασία του μοντέλου. Η ομάδα σχηματίζεται από συμμετέχοντες ρόλους και η πρόθεσή της είναι να ακολουθήσει ένα πλάνο που θα επιτύχει έναν συγκεκριμένο στόχο. Ακολουθούν ορισμένες από τις κύριες κλάσεις αυτής της οντολογίας.

- **TMember:** Κάθε μέλος της ομάδας που κατέχει συγκεκριμένο ρόλο. Τα μέλη χαρτογραφούνται σε υποκατηγορίες της τάξης ρόλων και περιγράφουν συγκεκριμένη συμπεριφορά για κάθε λειτουργικό ή ομαδικό ρόλο που απαιτείται στην ομάδα.
- **TGoal:** Ο στόχος της ομάδας για τον οποίο αυτή έχει συσταθεί.

$$TeamGoal \sqsubseteq Situation \sqcap \exists hasEffect.E \text{fect}$$

- **TPlan:** Το πλάνο που πρέπει να ακολουθήσει η ομάδα για την επίτευξη του στόχου. Το πλάνο είναι μια ακολουθία συμπεριφορών ρόλων που πρέπει να εκτελεστούν. Μετά την εκτέλεσή τους, η ομάδα φτάνει στην τελική κατάσταση του πλάνου που θα πρέπει να συμπίπτει με τον στόχο της ομάδας.
- **TRole:** Είναι η επέκταση της κλάσης **rm:Role**. Έχει σχεδιαστεί για να περιγράφει τους ρόλους της ομαδικής εργασίας που συμμετέχουν στην ομάδα των υπηρεσιών για τη σύνθεση,

όπως Maintainer, Monitor, Promoter, Developer, Organiser, Innovator, Advisor, Producer που εντοπίζονται οι αρμοδιότητές του καθώς και οι πιθανοί κίνδυνοι και τα τρωτά σημεία στο πεδίο αρμοδιότητάς του.

- TBehavior: Η συμπεριφορά ομαδικής εργασίας που μπορεί να εκτελεστεί από ρόλους ομαδικής εργασίας. $TBehavior \sqsubseteq rm:Behavior \sqcap \forall rm:hasBehavior - rm:TRole$.
- TeamFactor: Η συμπεριφορά ομαδικής εργασίας στο πεδίο όπως αυτή εξειδικεύεται και εκφράζεται από συγκεκριμένες ιδιότητες.

Η Sensing οντολογία

Βασίζεται κυρίως στην ανεπτυγμένη οντολογία SSN [18] που περιγράφει αισθητήρες, ενεργοποιητές και παρατηρήσεις. Επεκτείνεται για να συμπεριληφθεί στην οντολογία RM, επιτρέποντάς μας να περιγράψουμε τη διαδικασία ανίχνευσης που παρέχει πληροφορίες στους ρόλους της ομαδικής εργασίας σχετικά με την παρατήρηση ενός RM στο πλαίσιο της σύνθεσης των υπηρεσιών. Μια κατάσταση είναι ένα στιγμιότυπο του περιβάλλοντος σε μια συγκεκριμένη στιγμή και για να την περιγράψουμε εν μέρει χρησιμοποιούμε προτάσεις που προκύπτουν από την περιγραφή μέσω των οντολογιών.

- Sensor: Αυτή η κλάση έχει σχεδιαστεί για να περιγράφει τις συσκευές ανίχνευσης που χρησιμοποιούνται για τη διαδικασία της παρατήρησης.

$Satellite \sqsubseteq sen:Sensor, Smartphone \sqsubseteq sen:Sensor$

- Observation: Αυτή η κλάση περιγράφει τη διαδικασία παρακολούθησης που έχει ρυθμιστεί για την παρατήρηση της συμπεριφορά ενός χαρακτηριστικού ενδιαφέροντος, όπως αυτό θα οριστεί στο πεδίο του προβλήματος.
- ConfigurationProcedure: Αυτή η κλάση μας επιτρέπει να ορίσουμε συγκεκριμένη διαμόρφωση που απαιτείται για κάθε αισθητήρα που θα χρησιμοποιηθεί σε μια διαδικασία παρατήρησης. Θα αναφέρει λεπτομερώς ποια ιδιότητα παρατηρείται και με ποιον σχετίζεται.
- EntailingSituation: Μια κατάσταση που αποτελεί ένδειξη ότι υπάρχει ανάγκη παρέμβασης μιας ομαδικής συμπεριφοράς. Αυτές οι καταστάσεις μπορούν είτε να παραχθούν από ενδείξεις που καταγράφονται απευθείας από δεδομένα αισθητήρων είτε από συλλογισμό δεδομένων που προέρχονται από συμπεριφορές άλλων συμμετεχόντων.

Η Service οντολογία

Η οντολογία Service ορίζει υπηρεσίες με ομαδική συμπεριφορά που συντίθενται από άλλες υπηρεσίες που είναι εγγεγραμμένες στο πεδίο ενδιαφέροντος. Εδώ έχουμε τις ακόλουθες κύριες έννοιες:

- SName: η ταυτότητα της υπηρεσίας.
- SRole: είναι η εξειδίκευση της κλάσης $rm:Role$, $sr:SRole \sqsubseteq rm:Role$.

Οι ρόλοι εξειδικεύονται περαιτέρω με ρόλους ομαδικής εργασίας ώστε να σχηματίσει υπηρεσίες ομαδικής εργασίας που είναι σε θέση να είναι ενεργές για τη συμμετοχή τους σε συγκεκριμένες περιστάσεις.

- SBehavior: είναι η εξειδίκευση της κλάσης `rm:Behavior`
 $\text{sr:SBehavior} \sqsubseteq \text{rm:Behavior}$

Η RoleDomain οντολογία

Η RM οντολογία είναι αυτή που καθολικά αντιπροσωπεύει το πεδίο του προβλήματος Δ. Συνδυάζει όλες τις προηγούμενες και εισαγάγει τη DomainTopic Ontology, η οποία καθορίζει ένα συγκεκριμένο κοινό λεξιλόγιο που αποτελείται από διάφορες έννοιες, ιδιότητες και κανόνες που σχετίζονται με τον εν' λόγω τομέα, ώστε οι συμμετέχοντες να είναι σε θέση να κατανοήσουν την επικοινωνία και ανταλλαγή πληροφοριών σε όρους της RM μοντελοποίησης.

Παραδείγματα τέτοιων τομέων είναι ένας τομέας έξυπνης πόλης ή τομέας βιομηχανίας. Για ένα πεδίο έξυπνο πόλης, για παράδειγμα, έννοιες όπως Μεταφορές, Ενέργεια, Μουσεία και Πόροι ορίζονται στην DomainTopic ή εισάγονται από μια ξεχωριστή ομάδα οντολογιών (π.χ. CityConcepts.owl).

- `rdm:RoleModel`: Αυτή η κλάση χρησιμοποιείται για τον καθορισμό του RM του τομέα.
- `rdm:Participant`: Οι παρεχόμενες υπηρεσίες, $\text{srv:SName} \sqsubseteq \text{dm:Participant}$
- `rdm:DomainGoal`: Ένα σετ προτάσεων που περιγράφουν το στόχο που μπορεί να επιτύχει μια υπηρεσία και που εκφράζονται σύμφωνα με τους επιχειρηματικούς στόχους του τομέα.
- `Plan`: Προκαθορισμένα πλάνα που θα μπορούσαν να ακολουθηθούν σύμφωνα με το RM του πεδίου για την επίτευξη των επιχειρηματικών στόχων του.
- `rdm:Query`: Μια αιτούμενη υπηρεσία που αναζητείται με τις προϋποθέσεις και τα αποτελέσματα της.
- `rdm:Situation`: Οι καταστάσεις είναι ένας τρόπος περιγραφής ενός τομέα, μπορούν να περιγράψουν τις διάφορες καταστάσεις που ρεαλιστικά ενδέχεται να προκύψουν από τις συμπεριφορές των συμμετεχόντων σε αυτόν το πεδίο και μερικώς προσδιορίζονται χρησιμοποιώντας τα Abox assertions της οντολογίας. Είναι καταστάσεις που το σύστημα μεταβαίνει κατά την εξέλιξη του και κάποιες από αυτά μερικώς προκαθορίζονται από την εξέλιξη καθώς ακολουθείται το πλάνο των εκτελέσιμων συμπεριφορών ή άλλες αποτελούν καταστάσεις που πρέπει να προσδιοριστούν ώστε ένας ρόλος να παρέμβει, καθώς υποδηλώνουν μια «ύποπτη» συμπεριφορά κάποιας υπηρεσίας.

$\text{Precondition} \sqsubseteq \text{dm:Situation}$,

$\text{Effect} \sqsubseteq \text{dm:Situation}$

$\text{DomainGoal} \sqsubseteq \text{dm:Situation} \sqcap \exists \text{rm:hasEffect.rm:TeamRole}$.

- `rdm:isUndertakenBy`: Μια κατάσταση μπορεί να εκφράζεται από συγκεκριμένες προτάσεις και σχετίζεται με έναν ρόλο ομαδικής εργασίας. Η ιδιότητα `dmn:isUndertakenBy` συσχετίζει

μια αντίστοιχη κατάσταση με μια συμπεριφορά ομαδικής εργασίας (π.χ., την κατάσταση ότι ένα μηχάνημα έχει υψηλή θερμοκρασία και τη συμπεριφορά του ρόλου του συντηρητή που πρέπει να αναλάβει την ευθύνη της κατάστασης).

- *rdm:DomainTopic*: Μια ταξινόμηση των εννοιών που σχετίζονται με το θέμα του τομέα. Αποτελείται από έννοιες που περιγράφουν τον σκοπό, τις διαδικασίες και τους πόρους του.

1.5.2 Δημιουργία της TrEW Service οντολογίας

Ένα RD ενσωματώνει τις πέντε ξεχωριστές οντολογικές ενότητες που έχουν δημιουργηθεί και στον πίνακα που ακολουθεί δείχνουμε μερικές από τις εξειδικευμένες σχέσεις μεταξύ των κλάσεων στις διαφορετικές οντολογίες.

Πίνακας 2: Συνδεδεμένες κλάσεις στην TrEWS οντολογία.

Οντολογία A	Οντολογία B	Αξίωμα
RoleDomain	RoleModel	$rdm : Participant \sqsubseteq \exists play.rm : Role$
Team	Sensing	$tm : TRole \sqsubseteq \exists interpret.sen : EntailingSituation$
Sensing	RoleDomain	$sen : EntailingSituation \sqsubseteq rdm : Situation$
Service	RoleModel	$ser : SRole \sqsubseteq rm : Role$
Team	RoleModel	$tm : TRole \sqsubseteq rm : Role$ $tm : TBehavior \sqsubseteq rm : Behavior$ $\Pi \forall rm : hasBehavior - .tm : TRole$
RoleDomain	Team	$rdm : Participant \sqsubseteq \exists as.tm : TBehavior$ $tm : TMember \sqsubseteq rdm : Participant$

Η TrEWSOnto είναι μια πρόταση για την εννοιολογική απεικόνιση των υπηρεσιών και της ομαδικής εργασίας, των οποίων οι έννοιες εξετάζονταν πάντα χωριστά. Προκειμένου να υποστηρίξουμε την ανάπτυξη των ομαδικών υπηρεσιών, δώσαμε έμφαση στο σχεδιασμό και την εφαρμογή της οντολογίας για την τυποποίηση τους. Δημιουργήθηκε μια σχετικά βασική οντολογία για τους ρόλους της ομαδικής εργασίας και τη συνδέσαμε με τη μεθοδολογία RM για να τυποποιηθεί η διαδικασία της σύνθεσης με την ομαδική εργασία. Χρησιμοποιώντας οντολογίες για τον ορισμό των υπηρεσιών TrEW, δεν εισάγουμε μόνο ευπροσάρμοστη και επαναχρησιμοποιήσιμη αναπαράσταση γνώσεων σχετικά με ολόκληρη τη διαδικασία σύνθεσης, αλλά μπορούμε επίσης να ενσωματώσουμε νέες γνώσεις σχετικά με τον τομέα. Η ύπαρξη της υποκείμενης οντολογίας θέτει τη βάση για την ανάπτυξη μιας λεπτομερούς οντολογίας των ρόλων ομαδικής εργασίας για τις υπηρεσίες στον εκάστοτε τομέα. Τέτοιοι ρόλοι, σε μια πλήρως ανεπτυγμένη οντολογία για έναν συγκεκριμένο πεδίο μπορεί να εκτείνονται σε διαφορετικές ευθύνες που ο καθένας αναλαμβάνει, αλλά πάντα στηριζόμενοι στην ίδια οντολογία ώστε συλλογικά να παρουσιάζουν τις επιχειρηματικές στρατηγικές που ακολουθούν οι συγκεκριμένοι τομείς.

1.6 Σύνθεση ομαδικών υπηρεσιών σε Έξυπνες Πόλεις

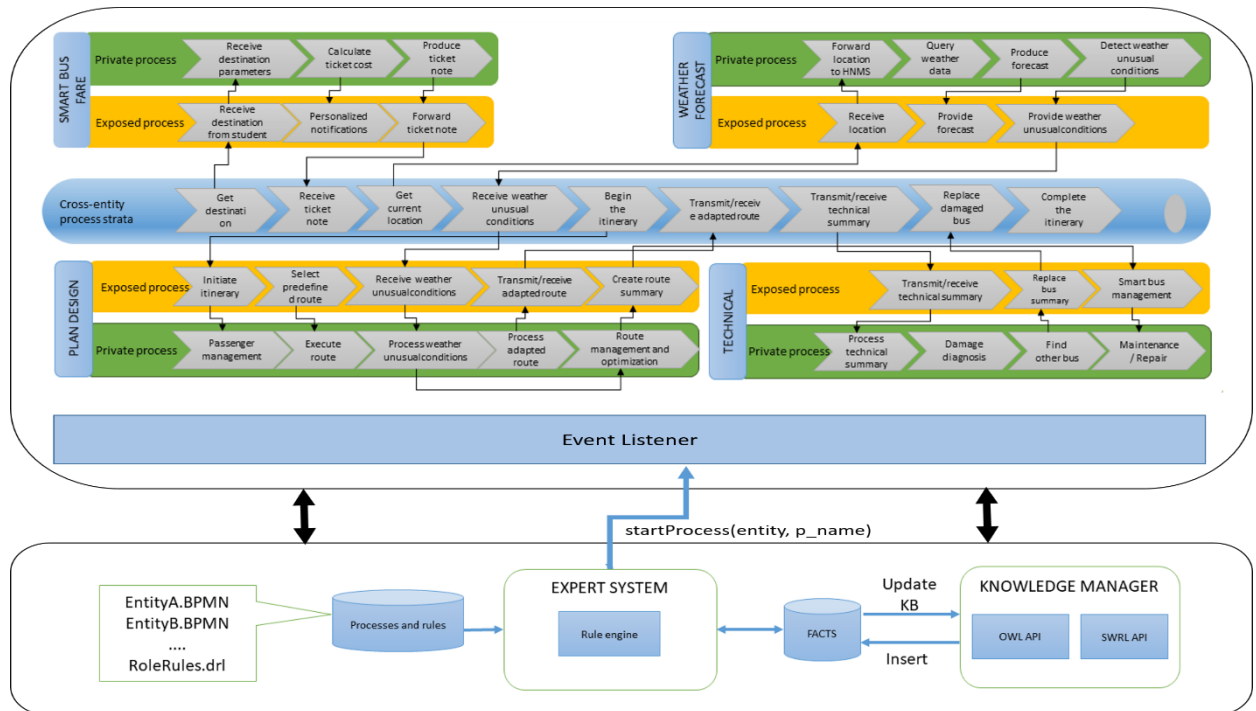
Για την παρουσίαση του τρόπου υλοποίησης του προτεινόμενου μοντέλου σύνθεσης, επιλέχθηκε ένα σενάριο χρήσης από μια εξελισσόμενη περιοχή, εκείνη των έξυπνων πόλεων (ΕΠ) [19]. Για την εφαρμογή του μοντέλου, εκμεταλλευόμαστε μια υπάρχουσα αρχιτεκτονική ΕΠ και την ενισχύουμε με ένα επίπεδο ομαδικής εργασίας που μπορεί να εισαγάγει την έννοια της συνεργασίας στις υπηρεσίες που «ζουν» σε μια ΕΠ για την αξιοποίηση των υποδομών της. Επιπλέον, αναπτύσσουμε ένα πλαίσιο που έχει σχεδιαστεί για να ικανοποιήσει τις απαιτήσεις μιας ΕΠ και να υποστηριχθεί το μοντέλο συνεργασίας [20,21].

1.6.1 Το «Bus Transportation» σενάριο

Ένας υποψήφιος τομέας εφαρμογής, ώστε να φανεί το όφελος της ανάπτυξης υπηρεσιών με δυνατότητα συνεργασίας είναι εκείνος των μεταφορών. Ας εξετάσουμε τη μετακίνηση με λεωφορείο σε μια ΕΠ και πιο συγκεκριμένα τη μεταφορά των μαθητών στο σχολείο τους. Σε αυτού του είδους τις μεταφορές παρόλο που ο δήμος είναι ο κύριος πάροχος, πολλές φορές υπάρχει η ανάγκη να χρησιμοποιηθούν περισσότεροι πάροχοι και εξωτερικές υπηρεσίες αλλά και να μελετηθεί η προηγούμενη χρήση και απόδοση τους από το παρελθόν, προκειμένου να εξασφαλισθεί η επιτυχής εκμετάλλευσή τους και να ελαχιστοποιηθούν οι περιπτώσεις αδυναμίας παροχής των υπηρεσιών. Οι πληροφορίες για το σχεδιασμό της διαδρομής που ακολουθεί ένα σχολικό λεωφορείο προέρχονται συνήθως από το ΠΣ του δήμου και στην πραγματικότητα, αυτό είναι ένα επιλυμένο πρόβλημα και τα συστήματα κυκλοφορίας περιλαμβάνουν τη δυνατότητα αυτή. Ωστόσο, αυτό καλύπτει μόνο την εξαγωγή ενός πλάνου, το οποίο μπορεί να δώσει τη συντομότερη διαδρομή για την παραλαβή των μαθητών και τη μεταφορά τους στον προορισμό τους. Δεν λαμβάνονται υπόψη για παράδειγμα καθυστερήσεις που μπορεί να συμβαίνουν στην παραλαβή των μαθητών όταν ο καιρός είναι βροχερός ώστε να βελτιωθεί η υπηρεσία ή όταν κάποια βλάβη συμβεί σε κάποιο λεωφορείο και πρέπει να δοθούν λύσεις έγκαιρα που θα αποτρέψουν πιθανές καθυστερήσεις ή άλλους κινδύνους στις διαδρομές των λεωφορείων. Στις επόμενες ενότητες παρουσιάζουμε τη λύση σε ένα πρόβλημα σχεδιασμού μιας καθημερινής διαδρομής όπου ένα λεωφορείο μεταφοράς μαθητών σταματά τη μεταφορά τους λόγω βλάβης και πρέπει να υποβοηθηθεί από άλλη υπηρεσία προκειμένου οι μαθητές να φτάσουν εγκαίρως στο σχολείο.

1.6.2 Ένα υπολογιστικό πλαίσιο

Η οντολογία που παρουσιάσαμε στην προηγούμενη ενότητα για την περιγραφή της συνεργασίας των υπηρεσιών, εγγενώς, είναι μια στατική δομή χωρίς δραστηριότητα και θα πρέπει να υιοθετηθεί ένα πλαίσιο που θα καθορίσει πώς θα χρησιμοποιηθεί. Ως επακόλουθο, είναι ανάγκη να προτείνουμε ένα πλαίσιο, το οποίο εφαρμόζει μια ενότητα διαχείρισης ροής εργασίας που συνδυάζεται με έναν rule engine για να διαχειριστεί τη σύνθεση των υπηρεσιών σε ένα έξυπνο



Σχήμα 2: Το πλαίσιο για τις υπηρεσίες της ΕΠ

περιβάλλον πόλης. Το rule base συστατικό είναι υπεύθυνο για τη διαχείριση της απαιτούμενης ροής γνώσης (required streams of knowledge) που περιλαμβάνεται στους ρόλους της ομαδικής εργασίας, προκειμένου να υποστηρίξει την απαραίτητη επικαιροποιημένη χρονογραμμή για την εκτέλεση. Η προτεινόμενη προσέγγιση λειτουργεί πέραν των τρέχουσων προσεγγίσεων που βασίζονται στη στοίβα πρωτοκόλλων υπηρεσιών Ιστού.

Όπως απεικονίζεται στο Σχήμα 2, το rule-based expert σύστημα που βασίζεται σε κανόνες θα μπορεί να παραγάγει συμπεράσματα για τα επόμενα εκτελέσιμα βήματα λαμβάνοντας υπόψη από κοινού γεγονότα, κανόνες και διαδικασίες. Όταν μια συμπεριφορά πρέπει να εκτελεστεί, η rule engine θα ζητήσει από τη workflow engine να εκτελέσει αυτό το βήμα. Αυτό μπορεί να επιτευχθεί εύκολα επειδή η μηχανή εκτέλεσης συμπεριφοράς είναι εξοπλισμένη με ένα στοιχείο Event Listener που "ακροάζεται" τα συμβάντα που αποστέλλονται από τη rule engine. Μόλις ολοκληρωθεί το τρέχον βήμα, η behavior engine επιστρέφει τον έλεγχο στο expert system για να λάβει το επόμενο βήμα. Η εξειδικευμένη εφαρμογή του συστήματος δίνει τον έλεγχο στη rule engine για να εκτελέσει τους σημασιολογικούς κανόνες και να παραγάγει συμπεράσματα ώστε να ενημερώσει τη μηχανή ροής εργασίας για το εκείνο το βήμα που θα εκτελεστεί στη συνέχεια. Έτσι, ο rule engine λειτουργεί ως συμπληρωματικό εργαλείο για τη στήριξη της χρονογραμμής που βασίζεται σε ρόλους ομαδικής εργασίας και επιτρέπει τη δυναμική σύνθεση και την προσαρμογή της συμπεριφοράς.

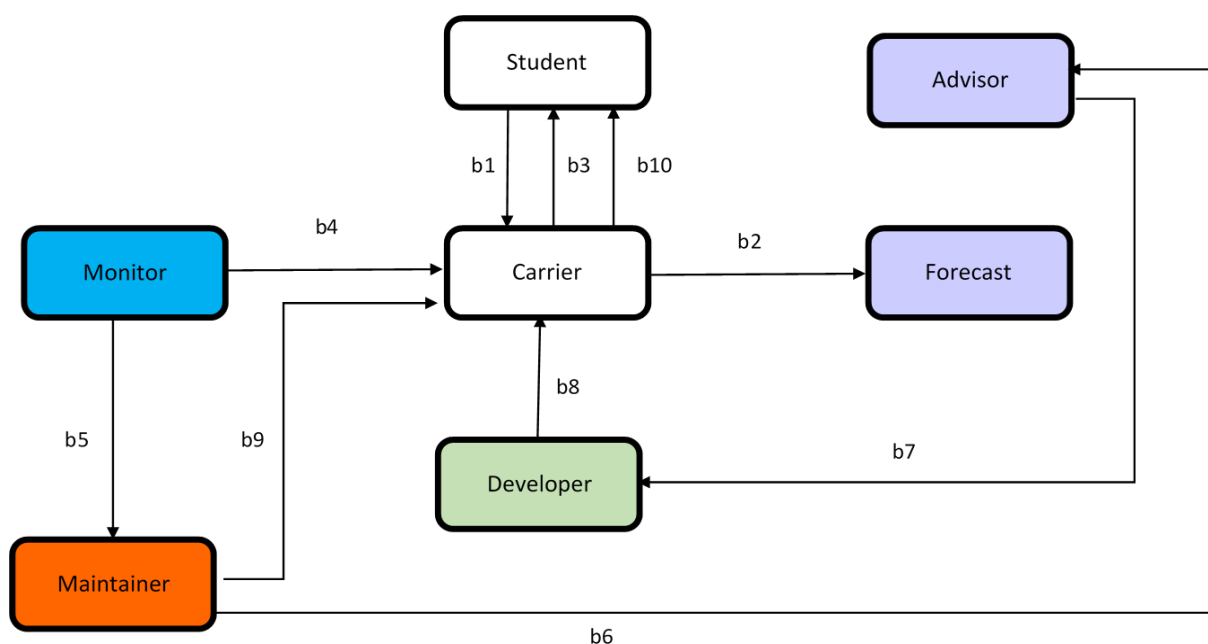
Για την επιτυχημένη ενσωμάτωση, οι κάτοχοι υπηρεσιών θα πρέπει να τις περιγράψουν λαμβάνοντας υπόψη κάθε ρόλο του πεδίου του προβλήματος. Επιπλέον, πρέπει να συνδέσουν τις

υπάρχουσες ιδιωτικές συμπεριφορές και τους πόρους με το συμφωνημένο μοντέλο αλληλεπίδρασης και να προσφέρουν τις αντίστοιχες διεπαφές στον έξω κόσμο. Στο Σχήμα 2 απεικονίζονται οι αλληλεπιδρούμενες οντότητες ΕΠ.

Κάθε οντότητα περιλαμβάνει ένα επίπεδο "Exposed Behavior" που παρέχει διεπαφές δημόσιας συμπεριφοράς σε άλλες οντότητες, ενώ περιέχει επίσης ένα επίπεδο "Private Behavior" που περιλαμβάνει τα "εσωτερικά" μέρη των συμπεριφορών της. Δεδομένης της οντολογίας, μπορούμε να παρατηρούμε το περιβάλλον όπου κάθε αισθητήρας έχει ανατεθεί σε έναν ενεργό ρόλο ενδιαφέροντος για την ανίχνευση αλλαγών και, επιλογικά, την αναγνώριση των σχετικών συμπεριφορών που προκάλεσαν την κατάσταση. Για να αντιστοιχίσουμε τα δεδομένα αισθητήρων στο μοντέλο αναπαράστασής μας, χρειαζόμαστε την οντολογία TrEWS. Ανάλογα με τα χαρακτηριστικά του τομέα, το επίπεδο αβεβαιότητας και της πολυπλοκότητας, ενδέχεται να εφαρμόσουμε διαφορετική μέθοδο για παραγωγή συμπεράσματος στις υπό διαμόρφωση συμπεριφορές ομαδικής εργασίας.

1.6.3 Μοντελοποίηση του πεδίου έξυπνης πόλης με ρόλους

Για να περιγράψουμε μια ΕΠ με ρόλους, πρέπει πρώτα να καθορίσουμε τους στόχους του πεδίου του προβλήματος. Σύμφωνα με την προσέγγιση ρόλων, όλοι οι στόχοι του πεδίου που προσδιορίζονται οργανώνονται στην οντολογία ως στιγμιότυπα της έννοιας του στόχου. Βάση των στόχων, των επικοινωνιακών προθέσεων και των συμπεριφορών του πεδίου κατασκευάζεται το RM. Επιπλέον, ορίζονται για τις συμπεριφορές οι προϋποθέσεις και τα αποτελέσματα ώστε να εκτελεστεί η κάθε μία. Έχοντας όλες τις λεπτομέρειες για το σχεδιασμό του RM του τομέα, απεικονίζουμε τις διαθέσιμες υπηρεσίες και αυτές αποτελούν συστατικά του τομέα. Στο επόμενο



Σχήμα 3: Μέρος του role model ενός πεδίου έξυπνης πόλης.

βήμα, σχεδιάζονται τα πλάνα για τα αιτήματα που αντιστοιχούν στους στόχους του τομέα. Διαφορετικά πλάνα για λύσεις σε αιτήματα που αντιστοιχούν στις διάφορες περιπτώσεις χρήσης θα μπορούσαν να δημιουργηθούν ως μια ακολουθία συμπεριφορών που θα εκτελεστούν για την επίτευξη συγκεκριμένων στόχων.

Για την αποτελεσματική εκμετάλλευση των υπηρεσιών και των πόρων απαιτείται συνεχής ροή πληροφοριών, προκειμένου οι υπηρεσίες, όταν εκτελούνται, να προσαρμοστούν στις αλλαγές ενός ρεαλιστικού περιβάλλοντος. Μέσω της διαρκούς λήψης δεδομένων που έμμεσα υποδεικνύουν τη συμπεριφορά των συμμετεχόντων, η υπηρεσία θα μπορεί να ενημερωθεί για συναφή γεγονότα που θα επηρεάσουν τη λήψη κάποιας απόφασης, π.χ. όταν ένα περιουσιακό στοιχείο ή μια υπηρεσία σε μια συγκεκριμένη περιοχή σπάνια χρησιμοποιείται, θα μπορούσε να αξιοποιηθεί καλύτερα, αλλάζοντας ενδεχομένως κάποιους περιορισμούς για τη χρήση του ή μετακινώντας το σε άλλη περιοχή. Παρομοίως, γνωρίζοντας τις ώρες που ο υπάλληλος ενός συγκεκριμένου κτηρίου εργάζεται θα μπορούσε να στοχεύει στη βελτιστοποίηση της ενέργειας που απαιτείται σε αυτό το κτίριο, ενώ προβλέπει με μεγαλύτερη ακρίβεια τις θέσεις στάθμευσης που χρειάζονται ή τις διαδρομές των λεωφορείων που θα χρησιμοποιηθούν για να βελτιώσουν την καθημερινότητα.

Με βάση την περιγραφή του παραδείγματος, παρουσιάζουμε μέρος του RM μιας ΕΠ που σχετίζεται με αυτό στο Σχήμα 3. Συμμετέχοντες είναι οι μαθητές που έχουν τον ρόλο των μετακινούμενων, τα αστικά λεωφορεία που έχουν ρόλο μεταβιβαστή, οι υπηρεσίες πληροφοριών έχουν το ρόλο πρόβλεψης, ενώ οι τεχνικοί οχημάτων έχουν το ρόλο της συντήρησης. Το βέλος μεταξύ των ρόλων απεικονίζει τη συνεργασία μεταξύ δύο ρόλων και δείχνει ποιος αρχίζει την συνεργασία. Οι έγχρωμοι ρόλοι αποτελούν ρόλους ομαδικής εργασίας για τον τομέα. Το παράδειγμα περιγράφει μια καθημερινή αλλά όχι απλή περίπτωση μεταφορών και μέσω αυτής παρουσιάζουμε πώς μια TrEWS που προέρχεται από το μοντέλο συνεργασίας μπορεί να προσαρμόσει τη συμπεριφορά της σε μια ΕΠ. Ο στόχος της σύνθετης υπηρεσίας που σχεδιάζουμε είναι να μεταφερθούν όλοι οι μαθητές στο σχολείο. Το λεωφορείο πριν φύγει από το γκαράζ, θα πρέπει να ελέγξει την υπηρεσία καιρού σε περίπτωση που βρει κάτι ασυνήθιστο. Αυτό απεικονίζεται στο RM με το βέλος που έχει την ένδειξη b2, δηλαδή τη συμπεριφορά που θα εκτελεστεί. Το αποτέλεσμα είναι να λάβει ο μεταφορέας γνώσεις σχετικά με την καιρική κατάσταση. Υποθέτοντας ότι όλα είναι ελεγχόμενα με τον καιρό, το λεωφορείο συλλέγει τους μαθητές σύμφωνα με τη σχεδιασμένη διαδρομή, όπως αυτή ορίζεται πριν ξεκινήσει το δρομολόγιο. Εν τω μεταξύ, στη διαδρομή για το σχολείο, το λεωφορείο εμφανίζει μια βλάβη. Σε αυτήν την περίπτωση, ο ρόλος του προγραμματιστή, του επιβλέποντα, του συμβούλου και του συντηρητή θα πρέπει να παρέμβουν σταδιακά και να συνεισφέρουν με λύσεις προκειμένου να ολοκληρωθεί επιτυχώς η μεταφορά των μαθητών. Ο επιβλέπων θα εντοπίσει το σφάλμα και θα ζητήσει από τον σύμβουλο εναλλακτικές λύσεις. Ο συντηρητής ενημερώνεται για τη βλάβη και ανάλογα με τον τύπο της, δίνει μια πρόβλεψη για το χρονικό διάστημα που το λεωφορείο θα είναι εκτός λειτουργίας

ώστε να αποφασισθεί εάν αυτή είναι άμεσα επισκευάσιμη ή το λεωφορείο θα πρέπει να αντικατασταθεί. Ο σύμβουλος ελέγχει άλλα διαθέσιμα σχολικά λεωφορεία στη γύρω περιοχή, τυχόν χωρητικότητα αυτών και ενημερώνει εάν είναι εφικτό κάποιος να τροποποιήσει τη διαδρομή, υπό την προϋπόθεση ότι μπορεί να φτάσει εκεί σε εύλογο χρονικό διάστημα, έτσι ώστε να μπορέσουν οι μαθητές να αλλάξουν λεωφορείο και να φτάσουν έγκαιρα στον προορισμό τους. Ο ρόλος του προγραμματιστή θα βοηθήσει τελικά στην αλλαγή του αρχικού πλάνου.

Στον Πίνακα 3, βλέπουμε τους διαφορετικούς εμπλεκόμενους ρόλους και συμπεριφορές της συνεργατικής σύνθετης υπηρεσίας SBus.

Πίνακας 3: Ρόλοι και συμπεριφορές της έξυπνης υπηρεσίας SBus

Role	Behavior	CA	Role	Precondition	Effect
Student	b1	queryTransfer	Carrier	schoolDay	studentInRoute
Carrier	b2	getWeatherPrediction	Forecast	planCalculated	weatherKnown
Carrier	b3	pickUpStudent	Student	studentInRoute	studentInBus
Monitor	b4	monitorRoute	Carrier	busInRoute	failure(aBus)
Monitor	b5	informFailure	Maintainer	busFailure	failureKnown
Maintainer	b6	estimateFailure	Advisor	failureKnown	failureEstimated
Advisor	b7	findAlternative	Developer	failure(aBus)	newPlanReuired
Developer	b8	delegateltinerary	Carrier	available(bBus)^ onSite(bBus)	newPlan(bBus)
Maintainer	b9	fixDamage	Carrier	failure(aBus)	fixed(aBus)
Carrier	b10	dropOffStudent	Student	onBus(bBus,student)	arriveAtSite

1.6.4 Παραγωγή συμπεράσματος για τη σύνθετη συμπεριφορά

Μετά την πληθυσμιακή ενημέρωση της οντολογίας, ακολουθεί η εφαρμογή των λογικών κανόνων για τη συμπεριφορά της ομαδικής εργασίας, μέσω των παραγόμενων γεγονότων που επιτρέπουν την προσαρμογή της συμπεριφοράς. Ο συλλογιστής εκτελεί τον σημασιολογικό εμπλουτισμό του μοντέλου γνώσης, συνάγοντας νέα γνώση κατά τον σχεδιασμό και την εκτέλεση της σύνθεσης. Ενώ οι κανόνες ενεργοποιούνται, εισάγονται νέα γεγονότα στη βάση και συνεχίζουν να συνάγονται σχετικά γεγονότα έως ότου επιτευχθεί ο στόχος της ομάδας. Αυτά τα γεγονότα στη συνέχεια χρησιμοποιούνται ως υποθέσεις για την παραγωγή νέων συμπερασμάτων και ενημέρωση της οντολογίας. Ενδεικτικά, ο Κανόνας A, περιγράφει μέρος της απόφασης που θα ληφθεί σχετικά με την επιλογή του λεωφορείου που θα αντικαταστήσει εκείνο με τη βλάβη. Η υπόθεση του συγκεκριμένου κανόνα είναι ότι το υποψήφιο λεωφορείο (b) θα πρέπει να έχει τον ίδιο προορισμό (d) με το χαλασμένο aBus, επιπλέον το υποψήφιο λεωφορείο βρίσκεται στην ίδια ή σε γειτονική

περιοχή και έχει διαθέσιμες θέσεις για τη μεταφορά περισσότερων μαθητών για να αναλάβει το δρομολόγιο (i).

Rule A : $hasDestination(?aBus, ?d) \wedge hasDestination(?b, ?d) \wedge Vicinity(?b, ?aBus) \wedge notLate(?b) \wedge availableSits(?b) \rightarrow onDuty(?b, ?i)$

1.7 Συμπεράσματα και Μελλοντική Έρευνα

Σε αυτή τη διατριβή έχουμε ορίσει ένα τυπικό και ολοκληρωμένο μοντέλο-πλαίσιο για την επίλυση του προβλήματος της σύνθεσης των υπηρεσιών διαδικτύου μέσω της ομαδικής εργασίας. Διερευνήσαμε προηγούμενες προσεγγίσεις για τη σύνθεση και διαμορφώσαμε τη δική μας άποψη για υπηρεσίες με ικανότητες συνεργασίας, τις υπηρεσίες TrEW όπως τις ονομάσαμε. Αρχικά, προτείναμε ένα μοντέλο συνεργασίας των ΥΙ με ρόλους. Μετά από μια εκτεταμένη βιβλιογραφική ανασκόπηση συγκροτήσαμε τους κυρίαρχους συνεργατικούς ρόλους που επικρατούν κατά τη διάρκεια της λειτουργίας μιας ομάδας υπηρεσιών και εμπλουτίσαμε το μοντέλο με αυτούς. Ο κύριος στόχος κάθε ρόλου είναι να ενισχύσει την ομάδα, να παρέμβει και να «ενεργήσει ως σύνδεσμος» κατά τη διάρκεια της συνεργασίας, προκειμένου να διατηρήσει την ομάδα των συντιθέμενων υπηρεσιών σε συνέπεια με τον στόχο της. Έχοντας ορίσει την τυποποίηση του μοντέλου με τη χρήση οντολογιών και τη βοήθεια του πλαισίου που παρουσιάσθηκε, είναι εφικτός ο ορισμός ενός πεδίου που διαθέτει υποδομή υπηρεσιών πληροφορικής με ρόλους που μπορούν να παρέχουν σχετικές και έγκαιρες πληροφορίες συναλλαγών μεταξύ των οντοτήτων σε ολόκληρο το πεδίο. Με αυτό τον τρόπο, τα αιτήματα προς τις υπηρεσίες καθώς και η διαχείριση εξαιρέσεων μπορούν να αυτοματοποιηθούν, να υπάρχει βελτίωση στη διοικητική απόδοση καθώς και μείωση του κόστους και του κινδύνου.

Τα αποτελέσματα αυτής της έρευνας που αποτελούν συνεισφορά αυτής της διατριβής έχουν παρουσιαστεί σε διεθνή περιοδικά και συνέδρια και συνοψίζονται στη λίστα που ακολουθεί:

- i. Μοντελοποίηση του προβλήματος της σύνθεσης των υπηρεσιών με τη χρήση ρόλων για να αντιμετωπιστεί η πρόκληση της ετερογένειας και του μεγάλου αριθμού διαθέσιμων υπηρεσιών [9]
- ii. Αναγνώριση των παραγόντων (teamwork factors) που επηρεάζουν τη συμπεριφορά των υπηρεσιών στην ομαδική εργασία [14]
- iii. Προσδιορισμός οκτώ ρόλων συνεργασίας υπηρεσιών και των χαρακτηριστικών τους προκειμένου να βελτιωθεί η διαδικασία της σύνθεσης υπηρεσιών [15]
- iv. Εννοιολογική περιγραφή της σύνθεσης ομαδικών υπηρεσιών με τυπικό τρόπο μέσω της δημιουργίας οντολογιών [17]

- ν. Σχεδίαση της αρχιτεκτονικής και του πλαισίου που βασίζεται στο μοντέλο σύνθεσης υπηρεσιών μέσω ρόλων και ομαδικής συμπεριφοράς για την εφαρμογή του σε ένα πεδίο έξυπνης πόλης [16,20,21]

Υπάρχουν ορισμένοι περιορισμοί στο προτεινόμενο μοντέλο που ιδανικά αποτελούν και αντικείμενα μελλοντικής έρευνας. Για παράδειγμα, απαιτείται εργασία υποδομής για την εφαρμογή του μοντέλου σε ένα πεδίο προβλήματος πραγματικής κλίμακας ξεκινώντας από την αρχή, δεδομένου ότι αυτό συνεπάγεται ότι όλες οι υπηρεσίες του πεδίου πρέπει να εκφραστούν σε ρόλους και μερικώς να αναδιαμορφωθούν. Προκειμένου να ξεπεραστεί αυτό το πρόβλημα, η σημασιολογική τεχνολογία θα μπορούσε να αξιοποιηθεί και να συνάγονται οι ρόλοι που μπορεί να παίξει μια υπηρεσία από τις δεξιότητες της ώστε να καθοριστούν οι συμπεριφορές της. Επιπλέον, ενώ η οντολογία μας για την ομαδική εργασία αποτελεί μια αρχή, αναγνωρίζουμε ότι μπορεί να επεκταθεί περαιτέρω ώστε να συμπεριλάβει και άλλες σχετικές διεργασίες που σχετίζονται με ειδικότερους τομείς. Είναι σαφές ότι οι αφηρημένες ομαδικές συμπεριφορές που καθορίζονται δεν επαρκούν και οι αναλυτικές τελικές συμπεριφορές θα πρέπει να σχεδιαστούν ώστε να εξάγονται από τον κάθε ρόλο σύμφωνα με το πεδίο εφαρμογής. Τέλος, ενώ περιγράφεται ένα σαφές πλαίσιο ανάπτυξης και χρήσης υπηρεσιών, η διαδικασία της σύνθεσης μπορεί περαιτέρω να βελτιωθεί από την άποψη της αυτοματοποίησης. Για παράδειγμα, αν και μπορούν να αναπτυχθούν ερωτήματα για την εξαγωγή ακριβούς και πολύπλοκης πληροφορίας, αυτό το επίπεδο γνώσης, λόγω πολυπλοκότητας δεν μπορεί να χρησιμοποιηθεί για μαζική κατανάλωση. Επομένως, επιπρόσθετες προσπάθειες μπορούν να επικεντρωθούν στην βελτίωση του αλγορίθμου που θα βασίζεται σε ευρετικές προσεγγίσεις ώστε να επιτρέπουν τη βελτίωση της διαδικασίας χωρίς να χρειάζεται να προγραμματίζετε το είδος του συλλογισμού για την παραγωγή συμπεράσματος κάθε φορά.

Η παρούσα εργασία μπορεί να αποτελέσει θεμέλιο και για επιπλέον κατευθύνσεις έρευνας, όπως: (i) επέκταση της οντολογίας για να βελτιωθεί η εκφραστικότητα της ενσωματώνοντας περισσότερες έννοιες, σχέσεις και ιδιότητες προκειμένου να αποτυπωθούν περισσότερες συμπεριφορικές πτυχές των υπηρεσιών κατά την ομαδική εργασία, (ii) ανάπτυξη ενός γραφικού επεξεργαστή (graphical editor) για τη διαχείριση του μοντέλου ομαδικότητας ρόλων που α επιτρέπει σε χρήστες ενός πεδίου χωρίς προχωρημένες γνώσεις πληροφορικής να μοντελοποιήσουν τις αντίστοιχες συμπεριφορές, (iii) προσομοίωση σχεδιασμού και εκτέλεσης των υπηρεσιών [22] που θα επιτρέπει την προεπισκόπηση των σχεδιασμένων συνθέσεων, αλλά και ενδεχόμενες τροποποιήσεις και παρεμβάσεις στο τελικό αποτέλεσμα.

Βιβλιογραφία

1. Martín Garriga, Andres Flores, Alejandra Cechich & Alejandro Zunino (2015) Web Services Composition Mechanisms: A Review, *IETE Technical Review*, 32:5, 376-383.
2. Huf, A., & Siqueira, F. (2019). Composition of heterogeneous web services: A systematic review. *Journal of Network and Computer Applications*, 143, 89-110.
3. Yourdshahi, E. S., Pinder, T., Dhawan, G., Marcolino, L. S., & Angelov, P. (2018, July). Towards Large Scale Ad-hoc Teamwork. In *2018 IEEE International Conference on Agents (ICA)* (pp. 44-49). IEEE.
4. Schwartz, T., Zinnikus, I., Krieger, H. U., Bürckert, C., Folz, J., Kiefer, B., ... & Schmitz, N. (2016, September). Hybrid teams: flexible collaboration between humans, robots and virtual agents. In *German Conference on Multiagent System Technologies* (pp. 131-146). Springer, Cham.
5. Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: A decade's overview. *Information Sciences*, 280, 218-238.
6. Vranić, V., & Laslop, M. (2016). Aspects and roles in software modeling: A composition based comparison. *Computer Science and Information Systems*, 13(1), 199-216.
7. Ertugrul, A. M. and Demirors, O. (2015). An exploratory study on role-based collaborative business process modeling approaches. In *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management*, page 14. ACM.
8. Margerison, C. J. and McCann, D. (1990). *Team management: Practical new approaches*. Mercury Books.
9. Belbin, R. M. (2011). Management teams: Why they succeed or fail. *Human Resource Management International Digest*, 19(3).
10. Tsoutsas, P., Fitsilis, P., and Ragos, O. (2017). Role modeling of IOT services in industry domains. In *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, pages 290-295. New York, NY, USA: ACM.
11. Fipa communicative act library specification. Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00037/SC00037J.html> (30.6. 2004).
12. Harel, D., Kozen, D., and Tiuryn, J. (2001). Dynamic logic. In *Handbook of philosophical logic*, pages 99-217. Springer.
13. Ghallab, M., Nau, D., and Traverso, P. (2004). *Automated Planning: theory and practice*. Elsevier.
14. Tsoutsas, P., Fitsilis, P., and Ragos, O. (2019). Teamwork behavior: A review to interconnect industry 4.0 entities. In *Technological Developments in Industry 4.0 for Business Applications*, pages 1-25. IGI Global.
15. Tsoutsas, P., Fitsilis, P., and Ragos, O. (2018a). Enhancing teamwork of services. In *Business Information Systems*, Berlin, pages 290-295. Springer.
16. Tsoutsas, P., Fitsilis, P., Anthopoulos, L., and Ragos, O. (2020a). Nexus services in smart city ecosystems. *Journal of the Knowledge Economy*, pages 1-21.
17. Tsoutsas, P. and Ragos, O. (2020). Towards an ontology for teamwork enabled services. In *International Conference on Algorithms, Computing and Systems*, pages 1-1. ACM (ISBN: 978-1-4503-7732-4).
18. Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. (2012). The ssn ontology of the w3c semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25-32.

19. Sivrikaya, F., Ben-Sassi, N., Dang, X. T., Görür, O. C., & Kuster, C. (2019). Internet of smart city objects: A distributed framework for service discovery and composition. *IEEE Access*, 7, 14434-14454.
20. Fitsilis, P., Tsoutsas, P., Anthopoulos, L., and Ragos, O. (2018). Teamwork behavior in smart and sustainable cities ecosystems. In *The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications*, Venice, Italy. Springer.
21. Tsoutsas, P., Iatrellis, O., Ragos, O., and Fitsilis, P. (2021). A framework for developing teamwork enabled services in smart city domains. Accepted for publication in 4th International Conference on Computers in Management and Business (ICCMB2021).
22. Tsoutsas, P., Fitsilis, P., and Ragos, O. (2018b). Services simulation in industry 4.0: A comparison of simulators. In *3rd International Scientific Conference on Industry 4.0*. In 3rd International Scientific Conference on Industry 4.0.

Απόδοση ξενόγλωσσων όρων

Ξενόγλωσσος όρος	Απόδοση
Abstract goal	Αφηρημένος στόχος
Agent	Πράκτορας
Artificial Intelligence	Τεχνητή νοημοσύνη
Behavior	Συμπεριφορά
Behavior effects	Αποτελέσματα εκτέλεσης συμπεριφοράς
Behavior preconditions	Προϋποθέσεις συμπεριφοράς
Business services	Επιχειρηματικές υπηρεσίες
Bussiness proccess	Επιχειρησιακή διαδικασία
Client request	Αίτημα χρήστη
Cloud	Υπολογιστικό νέφος
Communicative action	Επικοινωνιακή δράση
Composite service	Σύνθετη υπηρεσία
Composition algorithm	Αλγόριθμος σύνθεσης
Composition model	Μοντέλο σύνθεσης
CRM	Σύστημα διαχείρισης πελατών
Cyber Physical Systems	Κυβερνο-Φυσικά Συστήματα
Design patterns	Σχεδιαστικά πρότυπα
Domain	Τομέας, πεδίο
Domain ontology	Οντολογία περιεχομένου πεδίου
E-commerce agents	Πράκτορες ηλεκτρονικού εμπορίου
E-government	Ηλεκτρονική διακυβέρνηση
Event Listener	Ακροατής συμβάντων
Exposed behaviors	Εκτιθέμενες συμπεριφορές
Extra logical axioms	Έξτρα λογικά αξιώματα
Formal description	Τυπική περιγραφή
Framework	Πλαίσιο
Goal situation	Κατάσταση στόχου
Graphical editor	Γραφικός επεξεργαστής
Industry 4.0	Βιομηχανία 4.0
Initial state	Αρχική κατάσταση
IoT	Διαδίκτυο των πραγμάτων
Joint Intention (JIT)	Θεωρία της κοινής ψυχικής κατάστασης
Knowledge workflow	Ροή γνώσης
Modal Logic	Τροπική λογική
Multi Agent Systems (MAS)	Συστήματα πολλαπλών πρακτόρων
Object oriented	Αντικειμενοστραφή

Ontology	Οντολογία
PDL	Προτασιακή Δυναμική Λογική
Plan library	Βιβλιοθήκη πλάνων
Planning as satisfiability	Ικανοποιησιμότητα συνόλου προτάσεων
Planning problem	Πρόβλημα σχεδιασμού ενεργειών
Population	Πληθυσμιακή ενημέρωση
Private Behavior	Ιδιωτική συμπεριφορά
Reasoner	Συλλογιστής
Request for a composition	Αίτημα σύνθεσης
Role	Ρόλος
Role Domains	Πεδία ρόλων
Role model	Μοντέλο ρόλων
Role modeling	Μοντελοποίηση με ρόλους
Role semantics	Σημασιολογία των ρόλων
Rule engine	Μηχανή κανόνων
Semantic web	Σημασιολογικό ιστό
Semantic web service	Υπηρεσία Σημασιολογικού Ιστού (SWS)
Service	Υπηρεσία
Service composition	Σύνθεση υπηρεσιών
Service Domains	Πεδία με υπηρεσίες
Service-Oriented Architecture (SOA)	Υπηρεσιοστραφής αρχιτεκτονική
Shared Mental Models (SMM).	Κοινή διανοητική κατάσταση
Simulation	Προσομοίωση
Smart City Nexus Services (SCNS)	Πλέγμα υπηρεσιών έξυπνης πόλης
Software as a Service - SaaS	Λογισμικό ως υπηρεσία
Speech act theory	Θεωρία της ομιλίας
System behaviour	Συμπεριφορά του συστήματος
Team competences	Δεξιότητες ομάδας
Teamwork	Συνεργασία, Ομαδική εργασία
Teamwork behavior	Συνεργατική συμπεριφορά
Teamwork Driven Service Composition	Σύνθεση υπηρεσίας βασισμένη στην ομαδική
TeamwoRk Enabled Web Service (TrEW)	Σύνθετη ομαδική υπηρεσία ιστού
Teamwork model	Μοντέλο ομαδικής εργασίας
Teamwork role	Ρόλος ομαδικής εργασίας
Teamwork theory	Θεωρία συνεργασίας
Theorem prover	Αποδεικτής θεωρημάτων
Web service	Υπηρεσία ιστού
Weight (priority) model	Μοντέλο προτεραιότητας
Workflow	Ροή εργασίας

