MONOGRAFIES DE L'INSTITUT D'INVESTIGACIÓ EN INTEL·LIGÈNCIA ARTIFICIAL

SEMANTIC ALIGNMENT IN THE CONTEXT OF AGENT INTERACTION

Manuel Atencia

Consell Superior d'Investigacions Científiques

MONOGRAFIES DE L'INSTITUT D'INVESTIGACIÓ
EN INTEL·LIGÈNCIA ARTIFICIAL
Número 46

IIIA
Institut d'Investigació en
Intel·ligència Artificial

GOBIERNO
DE ESPAÑA

MINISTERIO
DE CIENCIA
E INNOVACIÓN

CSIC

Semantic Alignment in the Context of Agent Interaction

Manuel Atencia

Foreword by Marco Schorlemmer and Jaume Agustí

2011 Institut d'Investigació en Intel·ligència Artificial
Consejo Superior de Investigaciones Científicas
Bellaterra, Catalonia, Spain

Series Editor
Institut d'Investigació en Intel·ligència Artificial
Consejo Superior de Investigaciones Científicas

Foreword by Marco Schorlemmer and Jaume Agustí
Institut d'Investigació en Intel·ligència Artificial
Consejo Superior de Investigaciones Científicas

Volume Author
Manuel Atencia
Institut d'Investigació en Intel·ligència Artificial
Consejo Superior de Investigaciones Científicas

ISBN: 978-84-00-09356-3
ISBN (online): 978-84-00-09357-0
NIPO: 472-11-155-0
NIPO (online): 472-11-154-5
D.L: B.33338-2011

Ordering Information: Text orders should be addressed to the Library of the IIIA, Institut d'Investigació en Intel·ligència Artificial, Campus de la Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain.

Contents

List of Figures

List of Tables

Foreword

When embarking on research in Artificial Intelligence —particularly when carried out with a PhD degree in mind— one has generally two options: either to take a clearly defined and well-known open problem in the field and to attempt making a significant concrete contribution towards its resolution, or else to reflect on new avenues of research that provide new perspectives and insights into the field. Due to the enormous pressure posed on young scientists today, demanding them to publish their results in top-ranked journals and specialised conferences to get as many citations as possible, more and more PhD students opt for the first of the two strategies, where research results may have an important and quantifiable immediate impact. For this reason it is very gratifying to see that Manuel Atencia has chosen to follow the second alternative for his research as reported in this book, hence opening new paths in unchartered terrains.

In this book Manuel Atencia combines two approaches to distributed knowledge management and problem solving that have hitherto been largely kept separate: on one hand, the representation of knowledge in distributed systems by means of ontologies —the backbone of current semantic web technologies— and, on the other hand, the techniques of multiagent systems. In particular, Manuel Atencia has studied the manner in which interaction determines the meaning of terms during multiagent communication, putting forward a novel paradigm of semantic alignment of ontologies based on the notion of interaction, which is more in tune with present-day interaction-centred computational paradigms and language semantics. In addition, his research has the formal rigour you would expect from a scientist with a strong mathematical background, but at the same time tackles a long-standing problem in systems interoperability that reaches back to the early days of distributed database design. In this sense, this book also presents a concrete implementation of the proposed techniques together with extensive experimentation that validates their viability.

Either if you are a theoretically-minded researcher interested in fundamental issues concerning semantic technologies, or if you are a more application-oriented engineer looking for novel approaches that tackle the problem of semantic heterogeneity in multiagent systems, you will surely discover in Manuel Atencia's book valuable insights and a fresh view that will stimulate new and interesting research.

Bellaterra, October 2011

Marco Schorlemmer and Jaume Agustí
Institut d'Investigació en Intel·ligència Artificial
Consejo Superior de Investigaciones Científicas

Abstract

In this thesis dissertation we address the problem of semantic heterogeneity in the context of agent communication. We argue that current solutions based on ontologies and ontology matching do not capture completely the complexity of the distributed, dynamic and open-ended nature of multiagent systems, and that they usually do not reckon with the interaction-oriented purpose of communication. Our central thesis is that semantic alignment is also relative to the particular interaction where agents are engaged in, and that in such cases the interaction should be taken into account and brought into the alignment mechanism.

We firstly present a formal model for a semantic alignment procedure that incrementally aligns differing conceptualisations of two or more agents relative to their respective perceptions of the environment or domain where they are acting in. It hence makes the situation in which the alignment occurs explicit in the model. We call this approach Situated Semantic Alignment (or SSA), and we fall back on channel theory, Barwise and Seligman's theory of information flow to carry out the formalisation.

The understanding that semantic alignment is often interaction-dependent is specifically studied in Interaction-Situated Semantic Alignment (I-SSA), which can be seen as a particularisation of the model mentioned above. We also provide a formal foundation for I-SSA, but this time based on a mathematical construct inspired from category theory that we call communication product. In addition, we describe an alignment protocol and a matching mechanism that agents can follow in order to benefit in practice from this approach.

The I-SSA technique is implemented in SICStus Prolog and its viability is proven by means of an exhaustive abstract experimentation and a thorough statistical study through combinations of analyses of variance and Tukey tests. Furthermore, we present a case study about travel reservation that gives us the possibility to put I-SSA within the context of current state-of-the-art techniques. Although a deeper examination is required, this example shows that I-SSA is better suited for semantic alignment when interaction is specially relevant. Also it helps us to highlight the differences between this approach and more standard approaches for semantic alignment.

Agradecimientos

Esta tesis está dedicada a Sofía, quien ha estado a mi lado desde el principio, paciente y comprensiva, brindándome todo su apoyo y cariño en los no pocos malos momentos que precedieron a la consecución de este trabajo. Toda mi vida estaré en deuda con ella.

Mis agradecimientos más sinceros a mi familia. A mi padre, que me dio a conocer el mundo del pensamiento y de los libros, y a quien debo mi condición de intelectual y científico. A mi madre, por su sacrificio y soporte todos estos años, y a mis hermanos, por comprenderme y quererme en cualquier caso.

A mis amigos de Málaga, en especial a Antonio, Urtzi y Jose, con quienes he compartido miles de experiencias y entablado miles de conversaciones, en la cercanía y la distancia, y que bien saben lo que me ha costado llegar hasta aquí.

A todos mis amigos de Barcelona, como a Andrea, Claudio, Pau y Marta. A mis amigos y compañeros de trabajo Adrián y Marco, por haber soportado mis continuas interrupciones en su estudio, bien por una pregunta, una duda, o sin motivo alguno. Y sobre todo a Tito, por su infinita paciencia.

Quiero agradecer a mi director de tesis Marco Schorlemmer el esfuerzo que ha dedicado para que me sienta orgulloso de mi investigación. Trabajar con él ha sido una experiencia tremendamente enriquecedora, y a él le debo todo lo que he conseguido a nivel profesional. También me gustaría mencionar a todo el personal del IIIA, tanto científico como administrativo, por la ayuda que me han prestado durante todo este tiempo.

No puedo dejar de nombrar a Dave Robertson, quien, durante mi estancia en la Universidad de Edimburgo, me ayudó sobremanera con la programación en Prolog y la parte de implementación de esta tesis. Así como a Paolo Besana y Michael Rovatsos, por sus valiosos comentarios y sugerencias. Y a Juan Pane, de la Universidad de Trento, por su atención en los momentos finales.

A todos ellos gracias, todo esto no habría salido adelante sin vosotros.

Este trabajo de investigación ha sido en parte financiado por el proyecto STREP OpenKnowledge (FP6-027253) y el proyecto CONSOLIDER Agreement Technologies (CSD2007-0022, INGENIO 2010).

Bellaterra, Marzo 2010

Manuel Atencia
IIIA, CSIC

Acknowledgements

This thesis is dedicated to Sofía, who has been by my side since the beginning. She has been patient and understanding, supporting me lovingly in all the bad moments that have preceded the attainment of this work. I will be indebted to her the rest of my life.

My most sincere thanks to my family. Thanks to my father, who showed me the world of ideas and books, and to whom I owe my scientific and intellectual condition. Thanks to my mother for her sacrifice and support during all these years, and to my siblings for understanding me and loving me unconditionally.

I would like to thank Antonio, Urtzi and Jose, my friends from Malaga, with whom I have spent myriad of experiences and I have struck up thousands of conversations both in the proximity and distance. They do know how difficult it was for me to get to where I am now.

I would also like to thank my friends from Barcelona, Andrea, Claudio, Pau and Marta. And my friends and workmates Adrián and Marco, who have suffered constant interruptions due to my questions, doubts, or for no reason at all. But above all, thanks to Tito for his endless patience.

I would like to express my gratitude to my supervisor Marco Schorlemmer for his effort devoted to making me feel proud of this thesis. Working with him has been a tremendously enriching experience, and I owe him everything I have accomplished professionally. I would also like to mention all the IIIA staff, both researching and administrative, for the help that they have given throughout all this time.

I cannot leave out Dave Robertson, who during my stay at the University of Edinburgh, greatly helped me with the development in Prolog of the part of implementation of my thesis. I am also indebted to Paolo Besana and Michael Rovatsos for their valuable comments and suggestions, and to Juan Pane, from the University of Trento, for his help in the final moments.

Thanks to all of you. All of this would not have been possible without you.

This research work has been partially funded by the STREP project Open-Knowledge (FP6-027253) and the CONSOLIDER project Agreement Technologies (CSD2007-0022, INGENIO 2010).

Bellaterra, March 2010

Manuel Atencia
IIIA, CSIC

Chapter 1

Introduction

> Tale è la magìa delle umane favelle, che per umano accordo significano spesso, con suoni eguali, cose diverse.
>
> Such is the magic of human languages, which often, by virtue of an agreement between men, the same sounds mean different things.
>
> *Il nome della rosa*
> UMBERTO ECO

1.1 The Semantic Heterogeneity Problem

Initially, software applications, databases and expert systems were all designed and built by a reduced group of software or knowledge engineers, which had overall control of the entire lifecycle of IT artefacts. This time has gone, though, and it has given place to a more decentralised praxis: component-based software engineering, federated databases, distributed knowledge bases.

Surely the most determining factor of these new trends is the arrival and general use of the Internet. The World Wide Web has brought about an unprecedented global distribution of information in the form of inter-linked hypertext documents, online databases, open-source code or Web services, and protocols such as HTTP, FTP, or peer-to-peer (P2P) networks have provided new ways for data transfer and information sharing.

A new scenario has arisen in recent times: the Semantic Web. The Semantic Web was envisioned, at the turn of the century, as a way of "bringing the World Wide Web to its full potential" [Fensel et al., 2003], as an extension of the Web "in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [Berners-Lee et al., 2001]. The emphasis is now on the "semantics", as a guarantee for the Web to be significantly improved.

However, as in many other occasions, the exercise has proven to be trickier than thought at the outset. Though IT component interoperability has achieved a great success at the syntactic level, the same cannot be said of semantics, and this is particularly true for the Web. Standards have been agreed concerning hypertext representation (using HTML), hypertext location (by means of URLs), and data transfer (through HTTP protocols). But since the semantic aspect of information was addressed, achieving interoperability has not been an easy task.

At the heart of the Semantic Web is that systems should be able to exchange information and services with each other in a semantically rich and sound manner. Accordingly, the semantics of one system should be exposed to the environment in such a way that other systems can interpret it correctly and utilise it to achieve interoperability, thus enabling distributed reasoning and supporting applications and services. The problem is that there exist numerous ways of expressing, exposing and understanding semantics, which leads to heterogeneity, or more specifically, to semantic heterogeneity.

This is reminiscent to the problem of identifying and handling inconsistencies in formal specification when developing large and complex systems. But, as stated in [Finkelstein et al., 1994], "maintaining absolute consistency is not always possible", and, moreover, "often this is not even desirable since this can unnecessarily constrain the development process, and can lead to the loss of important information". This suggests that semantic heterogeneity should be seen as what it is, an endemic characteristic of distributed systems, and that, instead of attempting to make it disappear, we should live with it and try to achieve the necessary and sufficient conditions to semantic interoperability, even if this does not mean to resolve the semantic heterogeneity completely.

Now, for two systems to interoperate there has to be a well-established form of communication and the right means to achieve this efficiently and effectively. Ontologies have been advocated as a way to make a shared conceptualisation explicit, thus enabling two systems to share the same ontological commitment. From this viewpoint, we simply need ontologies to act at the protocol to which systems have to adhere in order to establish interoperability.

Ontologies are considered an appropriate answer to the problem of semantic heterogeneity in distributed environments such as those of federated databases, multiagent systems or the Semantic Web. In recent years, many ontologies have come out. This proliferation has been favoured by the speed with which the area of ontology engineering has matured, and the development of off-the-shelf tools for ontology creation. As stated in [Fensel et al., 2001], "ontologies are becoming popular largely because what they promise: a shared and common understanding that reaches across people and application systems".

Although the use of ontologies may facilitate semantic interoperability, this usage firstly relies on the existence of agreed domain ontologies. Furthermore, these domain ontologies have to be as complete and stable as possible, because different versions only introduce more semantic heterogeneity. As a consequence, semantic integration based on *a priori* common domain ontologies may be useful for clearly delimited, close and stable domains, but less so for fully decentralised, open and dynamic environments like the Semantic Web.

Another objection to the ontology-based approach for semantic integration has been stressed by Agustí and Corrêa da Silva: "centralised ontologies [...] promise to bring the control of the organisation back to what was possible under classical management techniques. The problem is that they may also bring back the rigidity of agencies organised under the classical management tenets" [Corrêa da Silva and Agustí, 2003]. Semantic agreements by means of a common ontology may sometimes be not only untenable, but also undesirable.

As a result, when ontology engineers began to apply their products to the Semantic Web with the aim of solving the semantic heterogeneity problem, it became apparent that it would yield a new form of heterogeneity: that of ontology heterogeneity. Now, current approaches mostly tackle the problem by matching ontologies, that is, by finding correspondences between semantically related ontological entities. This research area is very active and has attracted a lot of attention, but is far from being resolved, and this is so to such extent that ontology matching has been called the "Achille's heel of the Semantic Web" [van Harmelen and Antoniou, 2008].

1.2 Motivation of the Thesis

Until recently, most ontology matching mechanisms developed have taken a classical functional approach to the semantic heterogeneity problem, in which ontology matching is seen as a process that takes two or more ontologies as input and produces a semantic alignment of ontological entities as output (see [Euzenat and Shvaiko, 2007]). Furthermore, matching often has been carried out at design-time, before integrating knowledge-based systems or making them interoperate. This may have been successful for clearly delimited and stable domains and for closed distributed systems, but it is untenable and sometimes undesirable for the kind of applications that are currently deployed in open systems. Multiagent communication, P2P information sharing, and web-service composition are all of a decentralised, dynamic, and open-ended nature, and they mostly require ontology matching to be performed locally at run-time. In addition, in many situations peer ontologies are not even open for inspection (*e.g.*, when they are based on commercially sensitive information).

There exist efforts to efficiently match ontological entities at run-time, taking just those ontology fragments that are needed for the task at hand. Nevertheless, the techniques used by these systems to establish the semantic relationships between ontological entities —even though applied at run-time— still exploit *a priori* defined concept taxonomies as they are represented in the graph-based structures of the ontologies to be matched, use previously existing external sources such as thesauri (*e.g.*, WordNet) and upper-level ontologies (*e.g.*, Cyc or SUMO), or fall back on additional background knowledge repositories or shared instances.

We claim that semantic alignment of ontological terminology is ultimately relative to the concrete situation in which the alignment is carried out, and that this situation should be made explicit and brought right into the alignment mechanism. Let us consider the case of multiagent systems. Even two agents with identical conceptualisation capabilities, and making use of exactly the same vocabulary to specify their respective conceptualisations may fail to interoperate in a concrete situation due to their differing perception of the domain. Imagine a situation in which two agents are facing each other in front of a checker board. Agent A_1 may conceptualise a figure on the board as situated on the left margin of the board, while agent A_2 may conceptualise the same figure as situated on the right. Although the conceptualisation of 'left' and 'right' is done in exactly the same manner by both agents, and even if both use the terms *left* and *right* in their communication, they still will need to align their respective vocabularies if they want to successfully communicate to each other actions that change the position of figures on the checker board. Their semantic alignment, however, will only be valid in the scope of their interaction within this particular situation. The same agents situated differently may produce a different alignment.

This appreciation is not exclusive to physical environments. The meaning of certain terms is often very interaction-specific. For instance, the semantic similarity that exists, in the context of an auction, between the Spanish term 'remate' and the English expression 'winning bid' is difficult to establish if we are left to rely solely on syntactic or structural matching techniques, or on external sources such as dictionaries and thesauri. The term 'remate' may have many different senses, and none of them may hint at its meaning as 'winning bid'. But it actually has this very precise meaning when uttered at a particular moment of the interaction happening during an auction. Indeed semantic alignment is also relative to the particular interaction in which agents are engaged, and, more specifically, to the particular state of the interaction. In such cases we believe that the interaction state should be taken into account and brought into the alignment mechanism.

1.3 Contributions

In this thesis dissertation we present two approaches that realise the two ideas explained above: **Situated Semantic Alignment** and **Interaction-Situated Semantic Alignment**.

In Situated Semantic Alignment (or SSA), we consider a scenario with two or more agents situated in an environment. Each agent will have its own viewpoint of the environment so that, if the environment is in a concrete state, both agents may have different perceptions of this state. Because of these differences there may be a mismatch in the meaning of the syntactic entities by which agents describe their perceptions (and that constitute the agents' respective ontologies). We state that these syntactic entities can be related according to the intrinsic semantics provided by the existing relationship between the agents' viewpoints of the environment. The existence of this relationship is precisely justified by the fact that the agents are situated and observe the same environment.

We provide a model that formalises situated semantic alignment as a sequence of information-channel refinements as in channel theory, Barwise and Seligman's theory of information flow (see [Barwise and Seligman, 1997]). This theory is indeed useful for our endeavour since it models the flow of information occurring in distributed systems due to the particular situations —or tokens— that carry information. Analogously, the semantic alignment that will allow information to flow ultimately will be carried by the particular situation in which agents are acting.

In Interaction-Situated Semantic Alignment (or I-SSA), we apply the same ideas to a non-physical environment. We address the case in which two agents need to establish the semantic relationships with terminologies of other agents on the grounds of their communication within an interaction. I-SSA looks at the semantics of messages that are exchanged during an interaction entirely from an interaction-specific viewpoint: messages are deemed semantically related if they trigger compatible interaction state transitions —where compatibility means that the interaction progresses in the same direction for each agent, albeit their partial view of the interaction (their interaction model) may be more simple than the interaction that is actually happening.

We fall back this time on category theory [Mac Lane, 1998] to give a formal foundation of I-SSA. Interaction-situated semantic alignment is computed from a mathematical construct that we call communication product, which is a pullback in a specific category, in other words, a constrained product. We also resort to statistics in order to tackle the practical aspect of this approach. The matching mechanism associates every foreign message with a categorical variable ranging over local messages, such that a variable assignment represents a matching element. The mechanism further computes frequency distributions of all these variables on the basis of past successful interactions. Agents mapping choices are determined by virtue of these distributions.

Although both approaches apparently focus on different aspects when it comes to compute the semantic alignment, SSA and I-SSA are closely related. Actually, I-SSA can be seen as a particular case of SSA, where the interaction state is taken as the environment state. Furthermore, in both approaches agents' terminologies are incrementally aligned as more environment states are covered regarding SSA, or as more interactions are completed concerning I-SSA. In this way, we manage to favour the dynamism of multiagent systems, one of the main motivations of this work.

However, in the case of I-SSA, the formalisation is also accompanied by experimentation and evaluation. We have implemented I-SSA in Prolog, and have conducted an experimentation that proves the viability of this technique. This is based on a thorough statistical study based on combinations of analyses of variance and Tukey tests. In addition, we make a comparison with some current state-of-the-art matching techniques. This comparison is far from being exhaustive, since we simply use it to highlight the differences between the kind of semantic alignment that is computed by I-SSA and those that are produced by more standard approaches.

1.4 Outline of the Thesis

This thesis is organised in seven chapters, including this introduction. Chapters 2, 3 and 4 mostly focus on theoretical aspects, while Chapters 5 and 6 are more related to experimentation and applications. A chapter of conclusions and future work is included, as well as a couple of appendices.

Chapter 2 gives a detailed explanation of ontologies and ontology matching as the two main proposed solutions to the problem of achieving semantic interoperability in open and distributed environments. We start from the beginnings of ontologies as a solely philosophical topic to their irruption in computer science. We review three existing approaches to illustrate the primary aspects of ontologies. Ontology matching is then presented as a way of resolving ontology heterogeneity. Some of the most important matching techniques and matchers are mentioned, and semantic alignment is explained from a formal perspective as well. Finally, the chapter is concluded highlighting on limitations of current approximations for ontology matching, particularly in the context of multiagent systems. We emphasise the interaction-independent nature and the generalised lack of dynamism of state-of-the-art matching techniques. These limitations motivate the work reported in this thesis dissertation.

Chapter 3 presents a novel approach for addressing the problem of semantic heterogeneity in multiagent systems which aims at reducing the limitations of state-of-the-art matching techniques described in Chapter 2. In this chapter we lay out a formal model for a semantic alignment procedure that incrementally aligns differing conceptualisations of two or more agents relative to their respective perceptions of the environment or domain where they are acting. Thus we make the situation in which the alignment occurs explicit in the model. We call our approach Situated Semantic Alignment (or SSA). Our formalisation is based on channel theory, Barwise and Seligman's theory of information flow. This chapter also includes two illustrative examples that support the understanding of the theory: the first about the so-called Magic Box of Ghidini and Giunchiglia, and the second around robots in a typical package distribution scenario.

Chapter 4 continues the line of reasoning of Chapter 3 and applies the same ideas in a non-physical scenario. Interaction-Situated Semantic Alignment (or I-SSA) brings the state of the interaction in which agents are engaged right into the alignment mechanism. In this manner, we manage to make semantics fully dependent on interaction. I-SSA is formalised by means of a mathematical construct inspired from category theory, which we call communication product. Likewise we describe an alignment protocol and a matching mechanism that enable agents to put I-SSA into practice. As in SSA, agents incrementally align their own terminologies by interacting, which favours the dynamism and openness of agent communication. In order to clarify the exposition we repeatedly make use of a running example around the Blackjack game. At the end of the chapter, I-SSA is seen as a specialisation of SSA, as it is presented within the general framework described in Chapter 3.

Chapter 5 addresses the implementation and an experimentation of I-SSA. I-SSA is implemented in SICStus Prolog Release 4.0.7. The evaluation of the experimentation proves the viability of the I-SSA technique. We provide a thorough statistical study consisting of combinations of variance analyses and Tukey tests.

Chapter 6 shows an application of I-SSA in a realistic scenario. Specifically, a case study about travel reservation is presented. This allows us to put I-SSA into context within current state-of-the-art matching techniques. Two travel ontologies are provided, and we compare the resulting semantic alignments of the applications of I-SSA and some matching techniques over these ontologies. Although a more exhaustive comparison is required, this concrete example gives us the possibility to highlight the limitations of standard matching approximations reviewed at the end of Chapter 2, and to show that the I-SSA approach can be very helpful in certain scenarios.

Chapter 7 summarises the main contributions of this thesis and also discusses further directions of research.

Appendix A comprises the definitions and theorems of channel theory used in Chapter 3. No other background is strictly necessary to understand the formalisation given there.

Appendix B includes a fully description of the travel ontologies utilised in Chapter 6. We have chosen the Manchester OWL syntax for the sake of readability.

Every chapter begins with an abstract, and it is closed with a summary and concluding remarks. It is possible to obtain a general understanding of the whole thesis by simply reading these sections.

1.5 Publications

The following publications have been derived from this thesis:

- M. Schorlemmer, Y. Kalfoglou and M. Atencia. A formal foundation for ontology-alignment interaction models. *International Journal on Semantic Web and Information Systems*, 3(2): 50-68, April-June, 2007. Special issue on ontology matching. [ISSN 1552-6283], [eISSN 1552-6291].

- M. Atencia and M. Schorlemmer. Reaching semantic agreements through interaction. In A. Paschke, H. Weigand, W. Berhrendt, K. Tochtermann and T. Pelligrini, editors, *Fourth SigPrag International Pragmatic Web Conference Track, ICPW'09, at the Fifth International Conference on Semantic Systems, i-Semantics'09, 2-4 September, 2009, Messecongress-Graz, Austria, J.UCS Conference Proceedings Series*, pages 726-737. Verlag der Technischen Universität Graz, 2009. [ISBN 978-3-85125-060-2].

- M. Atencia and M. Schorlemmer. I-SSA: Interaction-Situated Semantic Alignment. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008. OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part I*, volume 5331 of *Lecture Notes in Computer Science*, pages 445-455. Springer, 2008. [ISSN 0302-9743].

- M. Atencia and M. Schorlemmer. A formal model for situated semantic alignment. In E. H. Durfee and M. Yokoo, editors, *Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2007, Honolulu, Hawaii, USA, May 14-18, 2007. Proceedings*, pages 1270-1277. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS). [ISBN 978-81-904262-7-5].

- M. Schorlemmer and M. Atencia. Taking interaction ontologically prior to meaning. *Seventh European Conference on Computing and Philosophy, E-CAP 2009, Bellaterra, Barcelona, Spain, July 2-4, 2009. Proceedings.*

- M. Atencia and M. Schorlemmer. Formalising Interaction-Situated Semantic Alignment: the communication product. *Tenth International Symposium on Artificial Intelligence and Mathematics, ISAIM 2008, Fort Lauderdale, USA, January 2-4, 2008. Proceedings.*

- M. Atencia and M. Schorlemmer. Semantic alignment of agent interactions through the communication product. *Fifth European Workshop on Multiagent Systems, EUMAS 2007, Hammamet, Tunisia, December 13-14, 2007. Proceedings.*

- M. Schorlemmer and M. Atencia. Semantic alignment in the context of agent interactions. *Workshop on Formal Approaches to Multi-Agent Systems, FAMAS 2007, Durham, UK, September 6-7, 2007. Part of the Multi-Agents Logics, Languages and Organisations – Federated Workshops, MALLOW 2007. Proceedings.*

- M. Atencia and M. Schorlemmer. Situated semantic alignment. In B. Dunin-Keplicz, A. Omicini and J. A. Padget, editors, *Fourth European Workshop on Multiagent Systems, EUMAS 2006, Lisbon, Portugal, December 14-15, 2006. Proceedings*, volume 223 of *CEUR Workshop Proceedings*, CEUR-WS.org. [ISSN 1613-0073].

Chapter 2

Semantic Interoperability

Abstract. Ontologies have been advocated as a suitable solution to the problem of semantic heterogeneity. In this chapter we start with a survey of ontologies from different angles: a philosophical one, a logic-mathematical perspective, and a view from computer science. The problem of ontology heterogeneity is then revealed. We focus on ontology matching as the most widely used approach for this problem, and we show a number of state-of-the-art matching techniques. We close the chapter stressing some limitations of current approximations for ontology matching, and we analyse the particular case of multiagent systems.

2.1 About Ontologies

We have introduced this pleonasm laying emphasis upon the problem of semantic heterogeneity as one of the core challenges for achieving semantic interoperability, and we have pointed out why and how this problem easily arises in distributed environments, such as federated databases, multiagent systems, or the Semantic Web. Ontologies have been advocated as a way to make a shared conceptualisation explicit, thus enabling two systems to share the same ontological commitment, and, therefore, to interoperate semantically. Ontologies have gained great popularity in recent years, as they are considered one of the primary components destined to play a crucial role for the Semantic Web to be fully realised.

This section is entirely devoted to ontologies. We first give a presentation of ontology from its origins as a philosophical topic (Section 2.1.1) to its present use in computer science (Section 2.1.2). Secondly, a number of formal approaches for ontologies and ontology-based semantic integration are shown (Section 2.1.3). We analyse the evolution of ontologies from classical knowledge representation to the Semantic Web (Section 2.1.4), and, finally, we provide a short explanation of ontology engineering (Section 2.1.5). At the end of this section, we stress how the generalised use of ontologies and its consequent proliferation has lead to a new kind of heterogeneity: that of ontology heterogeneity.

2.1.1 Origins

The word *ontology* comes from the Greek *ontos* (of being) and *logia* (science, study, theory). In philosophy, ontology is the branch of metaphysics dealing with the nature of being. Although the interest on ontological problems goes back to ancient Greece —the philosopher Parmenides already distinguishes the categories of "being" and "non-being" in the 5th century BCE in his poem *On nature*— the word 'ontology' was not coined until the early 17th century within the academic circles of the Protestant Enlightenment [Øhrstrøm et al., 2005].

It was Aristotle who first presented a systematic account of "being" in his works *Metaphysics* and *Categories*. These texts are considered milestones in the history of formal ontology. Aristotle used the notion of category to classify anything that can be said or *predicated* about anything. He made clear that these categories apply to entities by virtue of being. The Greek neo-platonic thinker Porfyrios made important comments on Aristotle's categories in his work *Isagoge*. Porfyrios presented the basis of Aristotle's thought as a tree-like scheme of dichotomous divisions, the so-called Porfyrios' tree (see Figure 2.1).

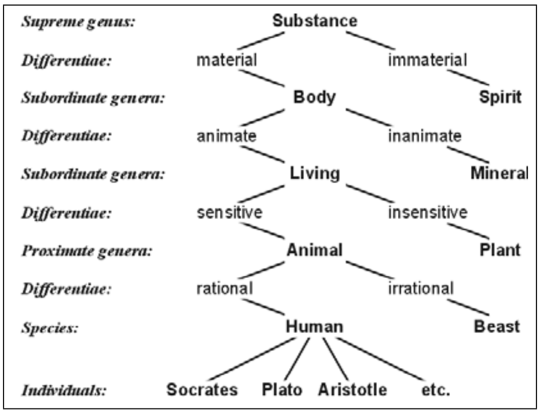


Figure 2.1: The *arbor porphyriana* or Porphyrian tree.

In the Middle Ages, William of Ockham argued against Porfyrios and the realistic conception of the world. In his text *Expositio aurea*, Ockham supports the idea that those universal concepts, used by us and by science, are formal statements about reality, mere conventions of our mind, rather than realities about the world itself. Contemporarily, Ramon Llull developed an alternative to the hierarchic concept systems in his *Ars Magna*. Llull's project was to construct a conceptual system that is independent of natural language, religion and culture. Llull had a great influence on Leibniz and the research on language formalisation and conceptual structures.

The first known appearance of the term 'ontology' was in *Ogdoas scholastica* written by Jacob Lorhard and published in 1606 [Øhrstrøm et al., 2005] (the word appeared on the frontispiece of the book as a synonym of "metaphysica"). Nonetheless, it was Christian Wolff who made the word popular in philosophical circles a century later. In his work *Philosophica prima sive ontologia*, Wolff states that science should be built upon clearly defined concepts, on valid axioms and inferences, and he presents ontology as the study of "being" understood as a "genus". This argument was supported by Jens Kraft in his book *Ontologie* who held that ontology would be a useful background or a useful foundation for any kind of scientific activity. Both for Kraft as for the whole Wolffian tradition, ontology is not a just a technique, but rather a framework of a number of true statements regarding the fundamental structure of reality.

Many other philosophers have contributed to the study of being as such: Thomas Aquinas, Immanuel Kant, Edmund Husserl, Martin Heidegger, William van Orman Quine, among others. It is unquestionable that computer scientists borrowed the term 'ontology' from philosophers, but also that they make use of it in a somewhat different manner.

2.1.2 Ontologies in Computer Science

It was probably McCarthy who first introduced the term 'ontology' in the AI literature. McCarthy used the term in a discussion of what kinds of information should be included in our understanding of the world [McCarthy, 1980]. This interpretation is reminiscent to the perception of ontology due to philosophers. However, Gruber's highly cited definition of ontology as an "explicit specification of a conceptualisation" alludes to a more subjective and variable use of the term.

In computer science, ontologies rarely attempt to account for all the entities that exist or not exist in the real world, but in a particular application domain. Sowa suggests in [Sowa, 2000] the following definition:

> the subject of ontology is the study of the categories of things that exist or may exist in some domain. The product of the study, called 'an ontology' is a catalog of the types of things that are assumed to exist in a domain of interest, D, from the perspective of a person who uses language L for the purpose of talking about D.

Within this view, a theory of ontology has to be seen in relation to a certain domain and it must presuppose a certain perspective. Moreover, ontologies are not only domain-dependent, but also language-dependent. For Guarino, "an ontology refers to an *engineering artefact*, constituted by a specific *vocabulary* used to describe a certain reality, plus a set of explicit assumptions regarding the *intended meaning* of the vocabulary words" [Guarino, 1998].

The fundamental differences between the philosophical and computational approaches for ontologies have been highlighted in [Guarino and Giaretta, 1995] and [Øhrstrøm et al., 2005]. In the philosophical tradition, ontologies are indeed oriented towards making strong claims about the world, and, in this sense, they are singular and domain-independent. In computer science, though, ontologies tend to be more plural and domain-dependent, as they refer to multiple, possibly fragmented domain descriptions relative to some selected perspectives. They also depend on the specific language in which they are written, commonly in a logical formalism.

There exist, however, ontologies that attempt to describe general concepts valid across all domains. These are usually referred to as *upper-level* ontologies, in contrast to the *domain* specific ontologies. Some examples are:

- The Cyc system was designed to accommodate all of human knowledge [Lenat et al., 1990, Lenat, 1995]. The name is taken from the stressed syllable of "encyclopedia". Cyc contains around 60,000 assertions on 6,000 concepts [Cyc, URL].
- The Suggested Upper Merged Ontology (SUMO) was proposed as starter document for the Standard Upperlevel Ontology Working Group (SUO WG) [Niles and Pease, 2001]. SUMO consists of around 4,000 assertions and 1,000 concepts. It is is designed to be relatively small so that assertions and concepts are easy to understand and apply [SUMO, URL].
- WordNet [Miller, 1995, Fellbaum, 1998] is a hierarchy ontology (linguistic resource) widely used for natural language processing. WordNet includes approximately 90,000 word senses [WordNet, URL].

Other examples of upper-level ontologies are the Basic Formal Ontology (BFO) [BFO, URL] and the IFF Ontology [IFF, URL].

Many domain specific ontologies have been developed in recent years. They range from simple thesauri or lightweight ontologies to more sophisticated ones. An example in the domain of genetics is the Gene Ontology [GO, URL]. Two examples in the domain of life sciences are EMTREE, the Life Science Thesaurus of Elsevier, and the Unified Medical Language System (UMLS) [UMLS, URL]. Examples in the domain of commerce and business are the classification United Nations Standard Products and Services (UNSPSC) [UNSPSC, URL], and the Enterprise Ontology [Uschold et al., 1998]. In the domain of Semantic Web, the Semantic Web Research Community (SWRC) ontology [SWRC, URL]. And we give a couple of examples in the domain of mathematics and software engineering: the Engineering Mathematics Ontology [Gruber and Olsen, 1994] and the Software Engineering Body of Knowledge [Bourque et al., 1999].

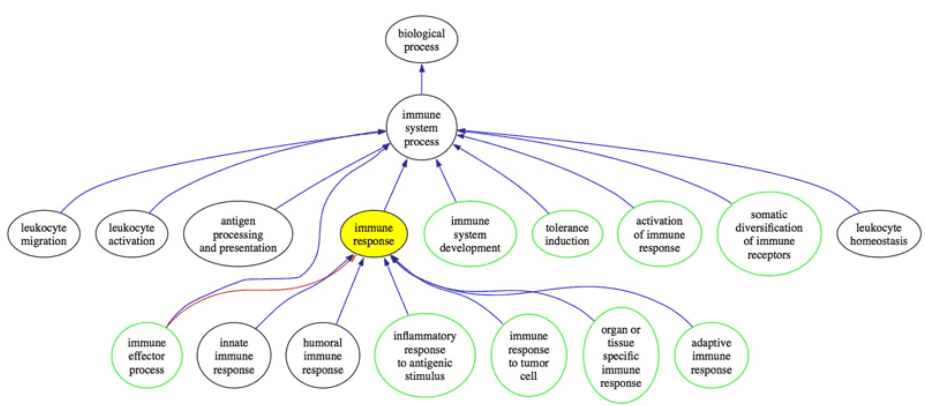


Figure 2.2: Extract of the Gene ontology

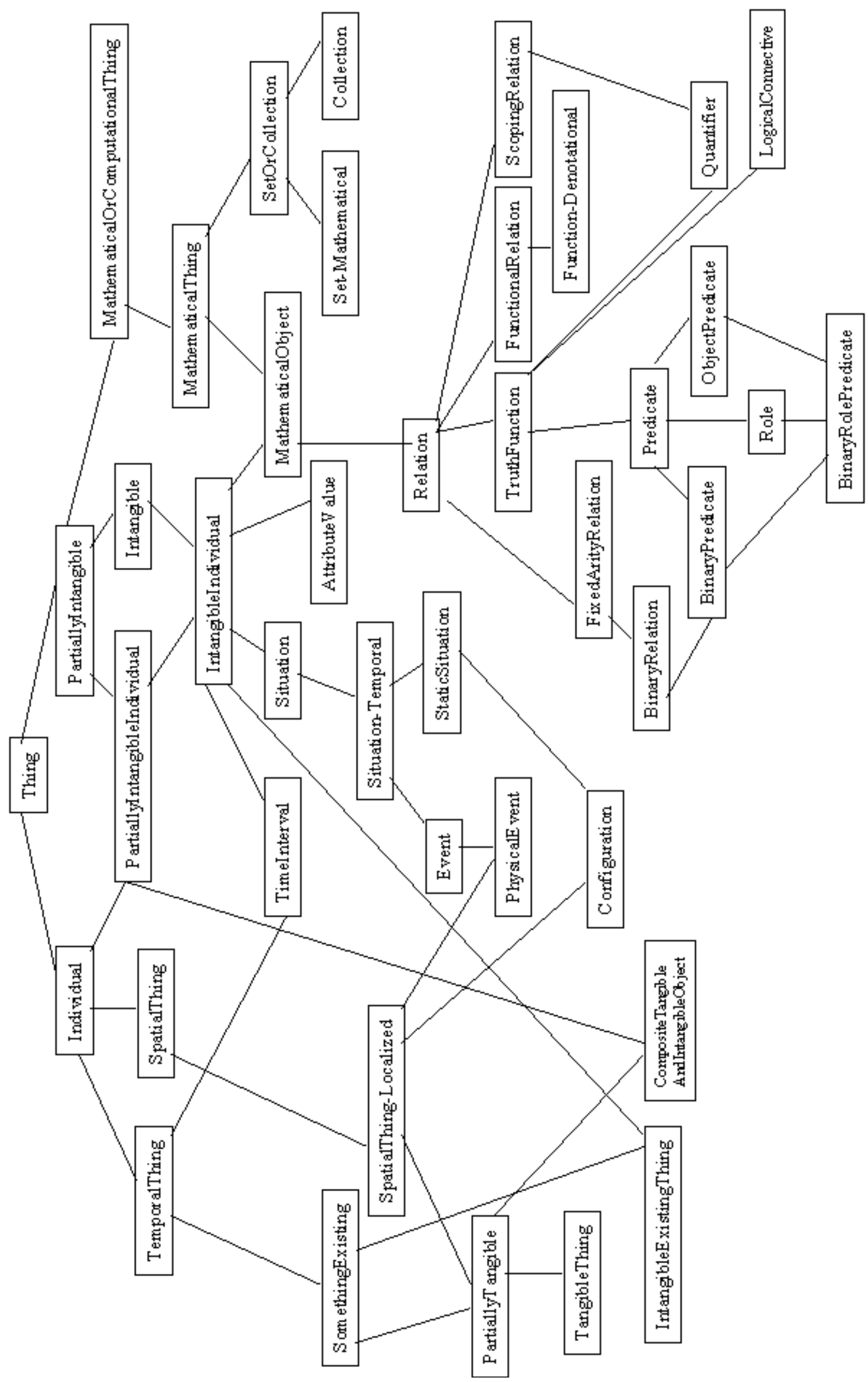


Figure 2.3: Extract of the Cyc ontology

We can check the differences between an upper-level ontology and a domain specific in Figures 2.2 and 2.3. An extract of the Gene Ontology is depicted in Figure 2.2. This ontology contains very specific terms of the domain of genetics, such as 'immune response' or 'tolerance induction'. In contrast to this, the extract of the Cyc ontology depicted in Figure 2.3 shows general domain-independent terms, such as 'individual' or 'spatial thing' or 'configuration'.

2.1.3 Formal Approaches for Ontologies

In the previous section we explained how computer scientists use ontologies, and we presented some informal definitions of an ontology. In this section a number of formal approaches for ontologies are provided. The approximations differ on the kind of formalism that they employ, as well as over the aspects of ontologies that they address.

Conceptualisation, Ontological Commitment and Ontology

In [Guarino, 1998], and in [Guarino et al., 1994, Guarino and Giaretta, 1995], the authors start from a criticism to the classical definition of conceptualisation as given in [Genesereth and Nilsson, 1987]. A conceptualisation is there defined as a first-order structure $S = \langle D, \mathcal{R} \rangle$, where D is non-empty set, usually referred to as *domain of discourse*, and $\mathcal{R} = \{R_i\}_{i \in I}$ is a family of relations on D, that is, $R_i \subseteq D^{n_i}$ (where $n_i \in \mathbb{N}$). The authors argue that these are *extensional* relations that reflect a particular state of affairs, but for a conceptualisation definition to have some sense, we need instead to focus on the "meaning" of these relations independently of the state of affairs. Guarino and Giaretta retrieve an example from [Genesereth and Nilsson, 1987] to illustrate this objection (see Figure 2.4) [Guarino et al., 1994]. A possible conceptualisation is given by the structure $\langle a, b, c, d, e, \{on, above, clear, table\} \rangle$, where, for instance, 'table' is a unary relation that holds of a block as long as this block rests on the table. According to the left side figure, $table = \{a, c\}$, while according to the right side one, $table = \{a, d\}$. The meaning of 'table' should not depend on the particular arrangement of blocks.

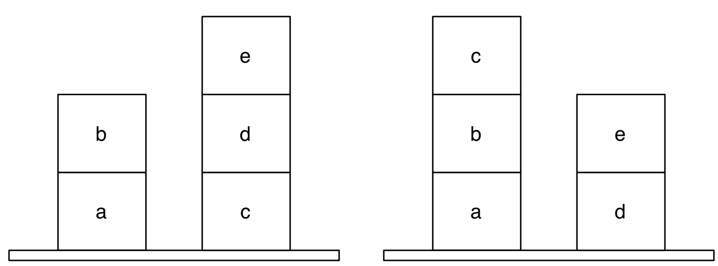


Figure 2.4: Two configurations of blocks on a table, two conceptualisations?

So the authors speak of *intensional* or *conceptual* relations. A conceptual relation is defined as a total function $\rho : W \to 2^{D^n}$, where W is a non-empty set of states of affairs or *possible worlds*. Accordingly, a conceptualisation is seen as a triple $\mathfrak{S} = \langle D, W, \mathcal{R} \rangle$, where $\mathcal{R} = \{\rho_i\}_{i \in I}$ is a family of such conceptual relations. Notice that, given a world $w \in W$, the structure $S^w = \langle D, \mathcal{R}^w \rangle$, where $\mathcal{R}^w = \{\rho(w) : \rho \in \mathcal{R}\}$, is a "classical" conceptualisation as defined before. The family $\{S^w\}_{w \in W}$ is said to be the family of all *intended world structures* of $\mathfrak{S}$.

Now, given a first-order language $L = L(\Sigma)$ over a signature Σ, L *commits* to a conceptualisation $\mathfrak{S}$ by means of how Σ is interpreted into $\mathfrak{S}$, *i.e.*, how constant symbols and predicate symbols are mapped to domain elements and conceptual relations, respectively. If $\mathfrak{I}$ is such an interpretation function, $\mathfrak{M} = \langle \mathfrak{S}, \mathfrak{I} \rangle$ is said to be an *ontological commitment* for L. Again, once fixed a world $w \in W$, $M^w = \langle S^w, I^w \rangle$, where I^w is equal to $\mathfrak{I}$ over constants, and, for each predicate symbol P, $I^w(P) = \rho(w)$ if $\mathfrak{I}(P) = \rho$, is a model of L. The family $\{M^w\}_{w \in W}$ is said to be the family of all *intended models* of L according to $\mathfrak{M}$.

At this point, we are in position to present a more formal definition of an ontology. Given a first-order language L with ontological commitment $\mathfrak{M}$, an *ontology* for L is "a set of axioms designed in a way such that the set of its models approximates as best as possible the set of intended models of L according to $\mathfrak{M}$" [Guarino, 1998]. Figure 2.5 is meant to be a clarification of this definition. To sum up:

> an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, *i.e.*, its ontological commitment to a particular conceptualisation of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (an the underlying conceptualisation) by approximating these intended models.

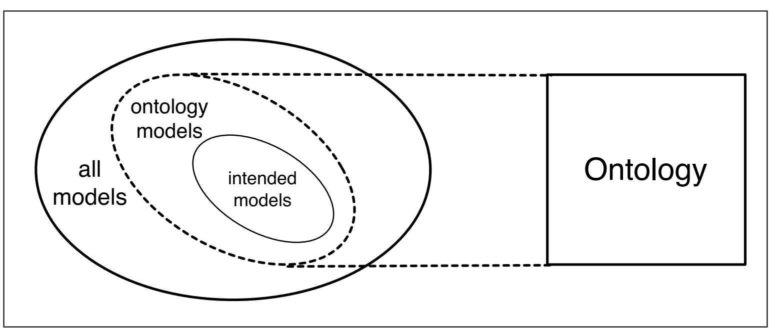


Figure 2.5: Ontology as an approximation of intended models

Description Logics and Ontological Entities

The above definition is useful for clarifying the notions of conceptualisation, ontological commitment and ontology. In practice, an ontology is just a logical theory expressible in the language of first-order logic, and, most commonly, in subsets of first-order logic, such as Description Logics (DLs), for which reasoning has good computational properties [Baader et al., 2003]. Here we give an overview of the formalism of description logics as appropriate ontology languages. This helps us to introduce the ontological entities that will arise in this dissertation, but any more detailed explanation is outside the scope of this thesis.

Description logics are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in structured and formally well-understood ways. Contrary to other knowledge representation formalisms, such as semantic networks and frames, they are equipped with a formal, logic-based semantics. Furthermore, there exists a balance between the expressive power and the efficiency of reasoning that makes description logics suitable as ontology languages.

We shall explain the formalism of description logics over a specific example. Extracts of two ontologies are represented in Figure 2.7. These ontologies specify conceptualisations of a travel reservation domain, and they will appear again in Chapter 6 where a case study around travel reservation is studied. For the task at hand, we can just focus on the ontology on the left-hand side. There we distinguish the main ontological entities: *concepts*, also called classes, and *roles* or properties. Concepts are interpreted as sets of objects in the domain (they are unary predicates); roles are interpreted as sets of pairs of objects (they are binary predicates). For instance, Reservation is a class, and it is interpreted as the set of all reservations; customerDetails, though, is a role, and it associates reservations with the details of the customers that make them. Reservation is an example of an atomic concept. It is also possible to build complex concepts from atomic ones by using concept constructors (see Figure 2.6). These more complex concepts are usually referred to as concept *descriptions*.

$C, D \longrightarrow$	$A \mid$	(atomic concept)
	$\top \mid$	(universal concept)
	$\bot \mid$	(bottom concept)
	$\neg C \mid$	(negation)
	$C \sqcap D \mid$	(conjunction)
	$C \sqcup D \mid$	(disjunction)
	$\forall R.C \mid$	(universal quantification)
	$\exists R.C \mid$	(existential quantification)
	$\leq n\, R \mid$	(number restrictions)

Figure 2.6: Basic concept constructors in DLs

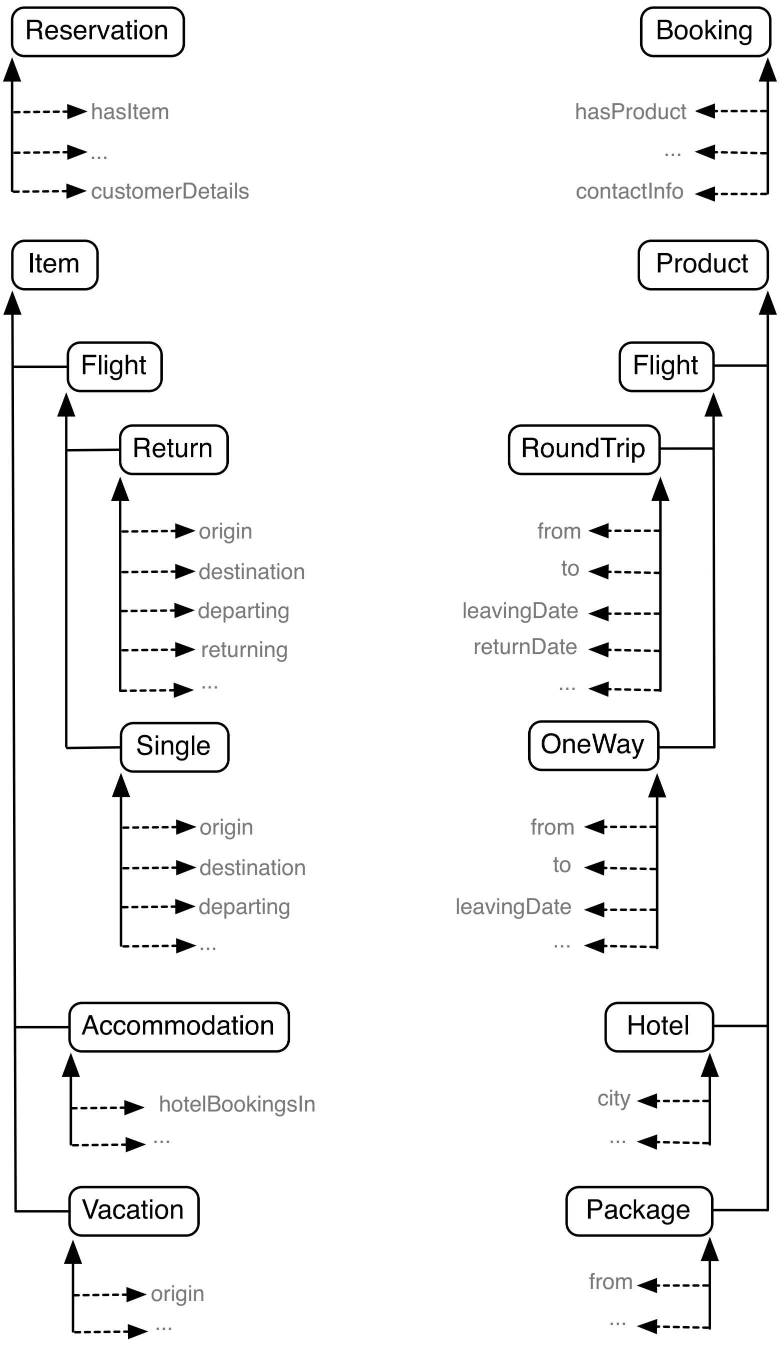


Figure 2.7: Two travel ontologies

For example, the following concept describes flights that have an origin and a destination, and that have less than two passengers:

$$\mathsf{Flight} \sqcap \exists \mathsf{origin.City} \sqcap \exists \mathsf{destination.City} \sqcap \leq 2\ \mathsf{hasPassenger}$$

Figure 2.8 summarises the semantics of the basic concept constructors in DLs.

$$\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \exists b.\ (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \\
(\leq n\, R)^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \sharp\{b \mid (a,b) \in R^{\mathcal{I}}\} \leq n\}
\end{array}$$

Figure 2.8: Semantics of basic concept constructors in description logics, for a given interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$.

The most simple description logic is $\mathcal{AL}$ (named so for attributive language). $\mathcal{AL}$ allows atomic negation, concept intersection and universal quantification, but limited existential quantification ($\exists R.\top$). Others languages can be defined depending on the concept constructions that are permitted. For example, $\mathcal{ALC}$ extends $\mathcal{AL}$ with concept union, full existential quantification and negation of complex concepts.

In addition to this description formalism, DLs are commonly equipped with both a terminological and an assertional formalism. Terminological axioms are typically realised in *equivalences* and *subsumptions*. A concept C is subsumed in a concept D, written $C \sqsubseteq D$, providing that all instances of C are necessarily instances of D, in other words, the first description is always interpreted as a subset of the second description. Now, a concept C is equivalent to a concept D, written $C \equiv D$, as long as $C \sqsubseteq D$ and $D \sqsubseteq C$. For instance, the following states that a flight can be a return flight or a single flight.

$$\mathsf{Flight} \equiv \mathsf{Return} \sqcup \mathsf{Single}$$

The assertional formalism can be used to declare properties of individuals. For instance, the following states that Barcelona is a city:

$$\mathsf{City(Barcelona)}$$

A finite number of terminological axioms is referred to as a TBox, whereas a finite number of assertional axioms is called an ABox. In DLs, an ontology can be described by defining a TBox and an ABox.

The suitability of description logics as ontology languages has been proven by the role that they have played as the foundation for several Web ontology languages. We will talk about this issue later in Section 2.1.4.

Ontology-Based Semantic Integration

Ontologies are helpful for two systems to both syntactically and semantically interoperate. This is commonly achieved by defining each system local language in terms of an ontology vocabulary, so the *semantic integration* is guaranteed. This sort of integration is dubbed "semantic" precisely because it assumes that the ontology consists of a theory $\mathcal{O}$ (and its underlying semantics), and because each system's local language is *interpreted* in the ontology. In this section we present a formal foundation for ontology-based semantic integration.

A number of formalisms have been proposed. Ciocoiu and Nau provided a formal definition of ontology-based semantic integration and translation (see [Ciocoiu and Nau, 2000]). More recently, Menzel ([Menzel, 2002, Menzel, 2005]) and Grüninger ([Grüninger, 2005]) have put forward basic theories of ontology and semantic integration. Bench-Capon, Malcolm and Shave also formalised ontologies and their compatibility (see, *e.g.*, [Bench-Capon and Malcolm, 1999, Bench-Capon et al., 2003]). We (partially) reproduce here the approximation given in [Schorlemmer and Kalfoglou, 2008, Kalfoglou and Schorlemmer, 2010].

Since we are more interested in the structural aspect of ontologies, we will represent an ontology as a logical theory $T = \langle L_T, \vdash_T \rangle$, where L_T is a language —a set of well-formed formulae over a vocabulary— and $\vdash_T$ is a consequence relation over L_T —a subset of $\mathcal{P}(L_T) \times \mathcal{P}(L_T)$ which satisfies certain properties.[1] One usually distinguishes a theory from its presentation. If a language L is infinite (as, for instance, in propositional or first-order languages, where the set of well-formed formulae is infinite, despite of having a finite vocabulary), any consequence relation over L will be infinite as well. Therefore one deals in practice with a finite subset of $\mathcal{P}(L) \times \mathcal{P}(L)$, called a *presentation*, to stand for the smallest consequence relation containing this subset. A presentation may be empty, in which case the smallest consequence relation containing it is called the *trivial theory*. We will write $Tr(L)$ for the trivial theory over a language L. It is easy to prove that, for all $\Gamma, \Delta \subseteq L$, $\Gamma \vdash_{Tr(L)} \Delta$ if, and only if, $\Gamma \cap \Delta \neq \emptyset$.

Theory interpretations capture the relationships between theories. A *theory interpretation* $\iota : T \to T'$ between theories $T = \langle L_T, \vdash_T \rangle$ and $T' = \langle L_{T'}, \vdash_{T'} \rangle$ is a map between the underlying languages that respects the consequence relations, *i.e.*, a map $\iota : L_T \to L'_T$ such that,

$$\text{if } \Gamma \vdash_T \Delta \text{ then } \iota[\Gamma] \vdash_{T'} \iota[\Delta]$$

for all $\Gamma, \Delta \subseteq L_T$ (where, as usual, $\iota[\Gamma]$ and $\iota[\Delta]$ are the sets of direct images of Γ and Δ along ι, respectively).

In this way, we say that two theories T_1 and T_2 are *semantically integrated* with respect to a theory T providing that there exist theory interpretations $\iota_k : T_k \to T$ $(k = 1, 2)$. Accordingly, two languages L_1 and L_2 are semantically integrated with respect to a theory T if the trivial theories $Tr(L_1)$ and $Tr(L_2)$ are. Figure 2.9 shows this situation diagrammatically.

[1] Commonly, those of Identity, Weakening and Global Cut [Dunn and Hardegree, 2001]. Recall that in $\Gamma \vdash_T \Delta$, Γ and Δ are interpreted conjunctively and disjunctively, respectively.

$$\begin{tikzcd} & T & \\ T_1 \arrow[ur, "\iota_1"] & & T_2 \arrow[ul, "\iota_2"'] \end{tikzcd}$$

Figure 2.9: Semantic integration of theories T_1 and T_2 w.r.t. T

Semantic interoperability is therefore formalised in terms of ontology-based consequences as follows. Assume that two theories T_1 and T_2 are semantically integrated by means of theory interpretations $\iota_i : T_i \to T$ $(i = 1, 2)$. Given $\varphi \in L_{T_1}$ and $\Gamma \subseteq L_{T_2}$, φ is said to be an *ontology-based consequence* of Γ if:

$$\iota_2[\Gamma] \vdash_T \iota_1(\varphi)$$

In order to generalise the former definitions and to capture the semantic integration of different logical languages, Schorlemmer and Kalfoglou propose in [Schorlemmer and Kalfoglou, 2008] a formal framework based on the theory of institutions [Goguen and Burstall, 1992, Goguen and Roşu, 2002]. In this framework, the approach of Guarino et al. is specifically captured by taking T as an S5 modal theory, while T_1 and T_2 are first-order theories.

This concludes the section related with formal approaches for ontologies. The approximation of Guarino et al. was used to explain the fundamental notions of conceptualisation, ontological commitment and ontology. We fell back on the formalism of description logics to present the basic ontological entities that will appear in this dissertation. Last but not least, we gave a more categorical approach to explain the ontology-based semantic integration. In the following section, though, we take a completely different direction as we study ontologies in the Semantic Web and how classical formalisms of knowledge representation were adapted to the new trends in the Web.

2.1.4 Ontologies in the Semantic Web

Hendler and Van Harmelen point out in [Hendler and van Harmelen, 2008] some limitations for classical knowledge representation formalisms (first-order logic, conceptual graphs) to bring the World Wide Web to its full potential. Before all else, classical KR systems do not make the most of the linking mechanisms of the Web. And "it is, in fact, exactly this network effect of gaining advantage by linking information created by other people, rather than recreating it locally, that makes the Web so powerful" [Hendler and van Harmelen, 2008].

In many KR systems, the notion of knowledge not directly under the control of a single mechanism is a tricky issue for the design. It can lead to different kinds of inconsistency, not only at syntactic and semantic levels, but also at extra-logical levels that concern implementation details. Therefore a non-logical infrastructure is required in order to achieve the network effect. Furthermore, the linkage demands new kinds of flexibility and addressing issues that KR formalisms have formerly ignored, and the scalability up to the standard of the Web is much larger than traditional in AI work.

These challenges have not been widely explored until recently. Actually, from the perspective of knowledge representation, "designing systems to overcome these challenges, using the Web itself for much of the extra-logical infrastructure, is the very definition of what has come to be known as the Semantic Web" [Hendler and van Harmelen, 2008]. The languages RDF, RDFS and OWL were designed as standards with the aim of resolving the limitations presented so far, and nobody doubts that they already are the most widely used KR languages in history. Let us talk about these languages in more detail.

The basis for the standardised Semantic Web languages is the Resource Description Framework (RDF) [Lassila and Swick, 1999]. RDF is a very simple language designed to make statements about resources (Web resources) in the form of subject-predicate-object expressions referred to as *triples*. The subject denotes the resource, identified with a URI (Uniform Resource Identifier), while the predicate denotes traits or aspects of the resource, as well as it expresses a relationship between the subject and the object. In this sense, RDF predicates correspond to traditional attribute-value pairs. However, RDF does not contain any mechanism for describing predicates, nor does it support description of relationships between predicates and other resources. This is provided by the RDF vocabulary description language, RDF Schema.

RDF Schema (RDFS) [Brickley and Guha, 2004] falls into the category of ontology languages, as it provides basic elements for the description of ontologies (RDF vocabularies), intended to structure RDF resources. RDFS enables the specification of classes and properties, which can be organised in generalisation hierarchies. Moreover, RDFS also allows simple kinds of inferences, such as inferring subclass relations or subproperty relations by transitivity. Nonetheless, RDFS still has a very limited expressive power. For instance, both RDF and RDFS lack any notion of negation or disjunction, and they only have a restricted notion of existencial quantification. This yields, among other drawbacks, to the impossibility of expressing inconsistencies.

The Web Ontology (WebOnt) Working Group of W3C [WebOnt, URL] did identify a number of use cases for the Semantic Web beyond the expressiveness of RDF and RDFS. The jointly effort of DAML+ONT [DAML-ONT, URL] and OIL [Fensel et al., 2001] resulted in DAML+OIL [DAML+OIL, URL] that was taken as the starting point for the Web Ontology Language (OWL), "the language that is aimed to be the standardised and broadly accepted ontology language of the Semantic Web" [van Harmelen and Antoniou, 2008].

OWL [Dean and Schreiber, 2004] builds upon RDF and RDFS. As in RDFS, classes and properties can be defined, but OWL provides the means to create new class descriptions as logical combinations (complements, intersections and unions) of other classes, as well as to define value and cardinality restrictions on properties. Most of the description logic formalisms can be covered by OWL's expressiveness, and some of its representational characteristics resemble those of

DLs. But OWL does fulfil the Web-like requirements described at the beginning of this section. It supports the linking of terms across ontologies making it possible to cross-reference and reuse information, and it has an XML syntax for easy data exchange. It is based on a Web architecture, and, similarly to RDF and RDFS, it uses URIs to unequivocally identify Web resources. And it is the first reasonably expressive ontology language that has become a standard.

The balance between expressive power and efficient reasoning in OWL is realised in a family of three sub-languages. The entire language is called OWL Full and makes use of all OWL language primitives, with the loss of a complete and efficient reasoning support, but gaining upward compatibility with RDFS. OWL DL (for Description Logics) provides the maximum expressiveness possible while retaining computational completeness, decidability and the availability of practical reasoning algorithms. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions. OWL Lite goes further and limits OWL DL to a subset of the language of constructs. In recent years, a new sub-language has arisen: OWL 2. It is more expressive than OWL Lite. In addition, many aspects of OWL have been reengineered in OWL 2, thus producing a robust platform for future development of the language.

2.1.5 Some Words about Ontology Engineering

Ontology Engineering is the discipline dedicated to the design and construction of ontologies. Ontology engineers are concerned with the ontology development process, the ontology life cycle, methodologies for building ontologies and the tool suites and languages that support them [Uschold and Grüninger, 1996, Jarrar and Meersman, 2002, Devedzic, 2002, Gómez-Pérez et al., 2004].

Enumeration of relevant terms, development of a class hierarchy, definition of properties and facets, and creation of instances, they all are among the guidelines for engineering ontologies (see, for instance, [Noy and McGuinness, URL]).

A common and recommended practice is to start from an existing ontology (fortunately, more and more ontologies are available). Now, the resources are of a varying nature: codified bodies of expert knowledge, topic hierarchies, linguistic resources and upper-level ontologies. The mentioned practice does suggest that ontology engineering should focus on decomposability, extensibility, maintainability, modularity and translatability of ontologies.

Although there exist tools for building ontologies (Ontolingua, WebOnto, Protégé, OntoEdit, WebODE) that make the task easier, manual ontology acquisition is still highly skilled, time-consuming and expensive. The emphasis is more on semiautomatic ontology acquisition. Ontology Learning aims at the integration of a multitude of disciplines in order to facilitate the construction of ontologies, in particular, that of machine learning [Maedche, 2002].

In general, the development of off-the-shelf products for designing ontologies, and the maturation of ontology engineering techniques have contributed to the proliferation of ontologies. Since ontologies were advocated as a solution to the problem of semantic heterogeneity, it seems pretty obvious that they have led to another kind of heterogeneity: that of ontology heterogeneity. Ontologies may be useful for closed and stable domains, but less so for fully decentralised, open and dynamic environments like the Semantic Web. In these scenarios, the problem of achieving semantic interoperability largely remains unresolved. Ontology matching is seen as a further step towards its solution.

2.2 Ontology Matching

Ontology matching has been proposed as a solution to ontology heterogeneity. Now, this heterogeneity is of a varying nature. Indeed we may have the same ontology but simply expressed in different ontology languages (for instance, KIF and OWL). It is possible, though, that we have two different sets of axioms that account for the same ontological commitment, or, even worse, that there only exists a partial overlap between the respective intended models. In the example of Section 2.1.3 (see Figure 2.7), we have two ontologies for a travel reservation domain. The concepts Reservation and Booking do refer to the same entity of the world, although they are syntactically dissimilar.

In a nutshell, *matching* is the process of finding relations or correspondences among entities of different ontologies. The outcome of this process is referred to as *semantic alignment* [Euzenat and Shvaiko, 2007]. More specifically, the matching process can be seen as a function that takes two (or more) ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ as input, and it produces a semantic alignment $\mathcal{A}$ as output. Some other factors may play a part in this process, such as an initial alignment $\mathcal{A}_0$ which is to be completed, a number of parameters (weights, thresholds), and external sources like common or background knowledge (see Figure 2.10). This process can be generalised to the case of more than two ontologies.

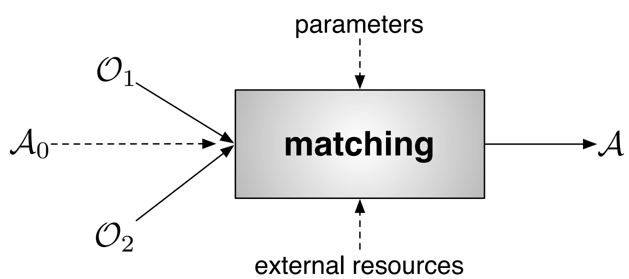


Figure 2.10: The matching process

The matching process results in a semantic alignment usually realised in terms of *matching elements*. A matching element can be represented as a tuple $\langle e_1, R, e_2, t\rangle$, where

- e_i is an entity of ontology $\mathcal{O}_i$ $(i = 1, 2)$,
- R is a relation between e_1 and e_2, and

- t is a confidence measure of the fact of e_1 being related with e_2 by means of the relation R.

The entities e_1 and e_2 are typically ontology classes or properties. Relation R is normally an equivalence ($\equiv$), a subsumption ($\sqsubseteq$) or a disjoint relation ($\perp$). And the confidence measure is often taken as a real number $t \in [0, 1]$, but it can also be a qualitative label (*e.g.*, low, high). For instance, in our example, the semantic alignment of a particular matching could include:

$$\langle \mathsf{Reservation}, \equiv, \mathsf{Booking}, 0.7 \rangle$$

Sometimes we will make use of an alternative notation in order to specify from which ontologies the entities come from:

$$\langle 1, \mathsf{Reservation} \rangle \equiv \langle 2, \mathsf{Booking} \rangle \ [0.7]$$

In the following section we take up again the line of reasoning of Section 2.1.3 about ontology-based semantic integration and we formalise semantic alignment within this framework. As illustration, in Section 2.2.3 we present a particular matching system as an instance of this formal framework.

2.2.1 A Formal Approach for Semantic Alignment

The approach presented here is entirely based on the conceptual framework provided in [Schorlemmer et al., 2006, Kalfoglou and Schorlemmer, 2010]. We also suggest [Zimmermann et al., 2006, Kutz et al., 2008] for a deeper study on formal approaches for semantic alignment.

From a formal perspective, semantic matching is the process that takes two theories T_1 and T_2 as input, called *local theories*, and computes a third theory $T_{1\leftrightarrow 2}$ as output that captures the semantic alignment of T_1 and T_2 languages, called *bridge theory*, and which underlies the semantic integration of T_1 and T_2 with respect to a *reference theory* T.

Assume that T_1 and T_2 are semantically integrated with respect to T by means of theory interpretations $\mathcal{I} = \{\iota_k : T_k \to T\}_{k=1,2}$. The *integration theory* $T_\mathcal{I}$ is the inverse image of T under the sum of interpretations $\iota_1 + \iota_2$, that is,

$$T_\mathcal{I} = (\iota_1 + \iota_2)^{-1}[T]$$

But, of course, we still have to explain what the sum of theory interpretations is, and how the inverse of a theory interpretation is defined.

In general, given two theories $T_1 = \langle L_1, \vdash_1 \rangle$ and $T_2 = \langle L_2, \vdash_2 \rangle$, the sum theory $T_1 + T_2$ is that one over the disjoint union of languages $L_1 \uplus L_2$ which consequence relation is the smallest containing $\vdash_1$ and $\vdash_2$. Therefore $T_1 + T_2$ is a "supertheory" of T_1 and T_2. Given two theory interpretations $\iota_1 : T_1 \to T$ and $\iota_2 : T_2 \to T$, the sum interpretation $\iota_1 + \iota_2$ is just the sum of their underlying maps of languages. Now, as for the inverse of a theory interpretation, $\iota^{-1}[T']$, where $\iota : T \to T'$, is the theory over the language of T such that:

$$\Gamma \vdash_{\iota^{-1}[T']} \Delta \text{ as long as } \iota[\Gamma] \vdash_{T'} \iota[\Delta]$$

And notice that this time $\iota^{-1}[T']$ is a "subtheory" of T.

The integration theory faithfully captures the semantic relationships between sentences in L_1 and L_2 as determined by their respective interpretation into T, but expressed as a theory over the combined language $L_1 \uplus L_2$. The sum of local theories $T_1 + T_2$ is always a subtheory of the integration theory $T_{\mathcal{I}}$. In semantic matching one usually isolates as output to the matching process the bit $T_{1\leftrightarrow 2}$ that makes $T_{\mathcal{I}}$ genuinely a supertheory of $T_1 + T_2$. Strictly speaking, a semantic matching process actually takes two presentations of local theories T_1 and T_2 as input and computes a presentation of the bridge theory $T_{1\leftrightarrow 2}$ as output. However, we have characterised semantic matching in terms of theories themselves for the ease of understanding.

Generally, the reference theory T is not an explicit input to the matching process (not even a representation of it). Instead it should be understood as the background knowledge used by the semantic matcher in order to infer semantic relations between the underlying vocabularies of the respective input theories. For a manual matcher, for instance, the reference theory may be completely dependent on the user input, while a fully automatic matcher would need to rely on automatic services (either internal or external to the matcher) to infer such a reference theory. It is for this reason that we talk of *virtual reference theory*, since it is not explicitly provided to the semantic matcher, but it is implicit in the way internal and external sources are brought into the matching process as background theory for semantic matching.

2.2.2 Ontology Matching Techniques

A wide range of matching techniques and systems have been proposed for some time now (see [Rahm and Bernstein, 2001, Kalfoglou and Schorlemmer, 2003b, Euzenat and Shvaiko, 2007]). It is not the purpose of this section to provide a comprehensive list of these techniques, but to present a number of them so as to show the current trends in this research field. We will follow the classification given in [Euzenat and Shvaiko, 2007] which is indeed built on the one presented in [Rahm and Bernstein, 2001].

Euzenat and Shvaiko identify three dimensions on which any classification of matching approaches should be based. These matching dimensions stem from those elements that take part in the matching process (see Figure 2.10). For this reason, they take into account the input of the algorithm (input dimension), the characteristics of the matching process (process dimension), and the output of the algorithm (output dimension). Parameters, external resources and input alignments are in the background.

Actually, two synthetic classifications are provided by Euzenat and Shvaiko, although only one will be presented here. First of all, matching techniques are all divided into two groups: *element-level* and *structure-level* techniques. The first group comprises all these techniques that analyse ontological entities in isolation, without paying attention to their relations with other entities, whereas techniques from the second group profit from how these entities appear together in a structure. These two groups are then classified regarding the interpretation of the input information: *syntactic*, *external* and *semantic* techniques. In this way, inputs can be interpreted on the basis of the syntax relating to a specific algorithm, some external resource or some formal semantics.

Element-level techniques

The most basic element-level techniques are the **string-based** ones. These are often used for matching names and name descriptions of ontological entities. The underlying assumption is that the more similar two strings are, the more likely they denote the same concepts. String-based techniques range from simple ones that try to find common substrings (prefixes, suffixes), to more sophisticated ones such as *edit distance* or *n-gram*. For example, a string-based technique would have success in matching Flight with Flights, but it would surely fail to match Hotel with Accommodation.

Before applying string-based techniques, names are usually pre-processed so as to achieve better results. One possibility is to treat names as words in some natural language and to make use of Natural Language Processing techniques in order to exploit their morphological properties. Examples of **language-based techniques** are *tokenisation*, *lemmatisation* and *elimination*. Broadly speaking, when "tokenising" one turns a name into a sequence of *tokens* by recognising punctuation, cases, blank characters, digits, etc. Tokens can be morphologically analysed and reduced to their most basic forms or *lemmas*. Furthermore, those tokens that are articles or prepositions are very often eliminated. For example, the following is a typical sequence of word modifications:

$$\mathsf{Flights\_and\_Hotels} \mapsto \langle \mathsf{Flights}, \mathsf{and}, \mathsf{Hotels} \rangle \mapsto \langle \mathsf{Flight}, \mathsf{Hotel} \rangle$$

Description logics are usually integrated with concrete sets such as the real numbers, integers or strings. This supports the modelling of concrete properties of abstract objects. OWL, for instance, gives access to a wide range of datatypes: float, date, anyURI, among others. Something that may lead us to definitely discard matching two names is the fact that they have completely dissimilar datatypes. It is rather unlikely that two names are equivalent when one has date as datatype and the other boolean. There exist other **constraints** to be taken into account as well, such as cardinality and keys.

If we treat ontological entities as words in a specific language, we can resort to **linguistic resources** (like common knowledge or domain specific thesauri) for the task of matching them. These techniques look for linguistic relationships such as synonymity, hyponymy or meronymy, among others. Thus, since Reservation and Booking are synonyms as English words, they would be equivalent.

With the same spirit, **upper-level and domain specific ontologies** (see the end of Section 2.1.2) could be used as external resources too. No matching system, however, has addressed this approximation yet. **Alignment reuse** is another useful technique motivated by the intuition that many ontologies that are to matched do not differ from already matched ontologies, specially if they describe the same application domain.

Structure-level techniques

Because ontologies are structured representations, it makes sense to profit from their structures when matching ontologies. A widely adopted approach is to consider an ontology as a graph whose nodes are ontological entities and whose edges are labelled with relation names. The problem of matching ontologies is therefore translated to a problem of matching graphs or, more precisely, a so-called graph homomorphism problem. **Graph-based techniques** compute (structural) similarity between nodes depending on their positions in the graphs, and hence depending on their neighbours. For instance, the fact that Reservation and Booking are similar should influence the relation between customerDetails and contactInfo, and the other way around. This idea can be realised in many different ways: comparing children, leaves, or comparing entities in the transitive closure, and so on.

Taxonomies are a special kind of graphs where relations reduce to the sole subsumption. Indeed "being a subclass" makes inherit some properties. These techniques exploit paths of sub-concepts or super-concepts so as to find similarities among nodes. As an example, since Return is a Flight in the ontology on the left of Figure 2.7, matching the latter with Flight of the ontology on the right would advise against matching Return with Hotel.

In line with alignment reuse, some advocate for having a **repository of structures** that store ontologies, and ontology fragments, along with pairwise comparisons in terms of similarity measures (normally real numbers in $[0, 1]$). In this way, it is possible to make a selection and identify structures that are worth matching in more detail.

Ontologies come equipped with a precise semantics of the structures that they hold. **Model-based techniques** focus on the semantic interpretation of the inputs, whence they are well grounded deductive methods. Archetypical examples are SAT (propositional satisfiability problem) and description logic reasoning techniques.

Besides logic-based approaches, recently it is **data analysis and statistical techniques** which have captured a lot of attention. The main drawback is that they highly depend on a fully large representative sample of the population from which to draw conclusions.

2.2.3 Matching Systems

The techniques surveyed in Section 2.2.2 are combined in particular matchers, and many matching systems have been developed so far. Some of them are:

- **Similarity Flooding** [Melnik et al., 2002],
- **Artemis** (or Analysis of Requirements: Tool Environment for Multiple Information Systems) [Castano et al., 2001],
- **COMA** (Combination of Matching Algorithms) [Do and Rahm, 2002],

- **Cupid** [Madhavan et al., 2001],
- **NOM** (Naive Ontology Mapping) [Ehrig and Sure, 2004] and its successor **QOM** (Quick Ontology Mapping) [Ehrig and Staab, 2004],
- **OLA** (OWL Lite Aligner) [Euzenat and Valtchev, 2004],
- **GLUE** [Doan et al., 2004],
- **Falcon-OA** [Jian et al., 2005].

Again, for an exhaustive compilation of matching systems we refer the reader to [Euzenat and Shvaiko, 2007]. Here we will particularly study the case of S-Match, developed in the University of Trento [Giunchiglia and Shvaiko, 2003, Giunchiglia et al., 2004, Giunchiglia et al., 2007], and we show how it can be seen as an instance of the general framework described in Section 2.2.1.

S-Match (University of Trento)

The input to S-Match is two labelled acyclic directed graphs $G_1 = \langle N_1, E_1, \ell_1 \rangle$ and $G_2 = \langle N_2, E_2, \ell_2 \rangle$, where N_k is a set of nodes, $E_k \subseteq N_k \times N_k$ is a set of edges, and $\ell_k : N_k \to S_k$ is a function from nodes to labels (where $k = 1, 2$). As hinted in Section 2.2.2, graphs G_1 and G_2 are to be understood as concept taxonomies, so that, given an edge (n, m) of G_k, $\ell_k(n)$ denotes a sub-concept of $\ell_k(m)$. Basically, S-Match reduces the matching problem to a propositional validity problem. Firstly, each label $s \in S_k$ is mapped to a formula $\psi_k(s)$ of propositional description logic whose atomic concepts are WordNet senses (called *concept of label*). In this step, language-based techniques are applied, and WordNet senses of resulting tokens are combined to form complex formulae. Secondly, each node $n \in N_k$ is mapped to a formula $\varphi_k(n)$ in propositional description logic (called *concept of a node*) defined as follows:

$$\varphi_k(n) = \bigsqcap_{m \in \uparrow n} \psi_k(\ell_k(m))$$

where $\uparrow n$ denotes the set of all nodes reachable from n (including itself). These formulae are then converted into equivalent formulae in a propositional logic language with Boolean semantics.

Central to the way S-Match computes the semantic relationships between nodes in N_1 and nodes in N_2 is the background knowledge brought into the matcher. S-Match uses matching element-level techniques which determine a set K of semantic relationships of the form $s\ R\ t$ between labels, $s \in S_1$, $t \in S_2$, and $R \in \{\sqsubseteq, \sqsupseteq, \equiv, \perp\}$. The final output is a collection of semantic relationships of the form $n\ R\ m$ between nodes, with $n \in N_1$, $m \in N_2$, and $R \in \{\sqsubseteq, \sqsupseteq, \equiv, \perp\}$ (called *mapping elements*), such that

- $n \sqsubseteq m$ if, and only if, A_K implies $\varphi_1(n) \to \varphi_2(m)$ in propositional logic
- $n \sqsupseteq m$ if, and only if, A_K implies $\varphi_2(n) \to \varphi_1(m)$ in propositional logic

- $n \equiv m$ if, and only if, A_K implies $\varphi_1(n) \leftrightarrow \varphi_2(m)$ in propositional logic

- $n \perp m$ if, and only if, A_K implies $\neg(\varphi_1(n) \wedge \varphi_2(m))$ in propositional logic

where A_K is the set of propositional axioms determined by the background knowledge K as follows:

$$\begin{array}{rcl}
\psi_1(s) \rightarrow \psi_2(s) \in A_K & \text{iff} & s \sqsubseteq t \\
\psi_2(s) \rightarrow \psi_1(s) \in A_K & \text{iff} & s \sqsupseteq t \\
\psi_1(s) \leftrightarrow \psi_2(s) \in A_K & \text{iff} & s \equiv t \\
\neg(\psi_1(s) \wedge \psi_2(s)) \in A_K & \text{iff} & s \perp t
\end{array}$$

The semantic relationships between nodes are computed and checked using a SAT prover. S-Match is run for any pair of nodes ($|N_1| \cdot |N_2|$ times in total).

Let us show that S-Match is a particular instance of the general framework described in Section 2.2.1. Let L_{WordNet} be the propositional language whose atomic propositions are WordNet senses, and let $\vdash_{A_K}$ be the smallest Boolean consequence relation over L_{WordNet} such that $\emptyset \vdash_{A_K} \{\alpha\}$ if, and only if, $\alpha \in A_K$. As we have seen before, S-Match maps nodes of input graphs into sentences of L_{WordNet}, and the S-Match output relationship between two nodes is determined by the way $\vdash_{A_K}$ relates their associated sentences. Consequently, an application of S-Match, along with its element-based matching techniques, establishes a virtual reference theory $T = \langle L_{\text{WordNet}}, \vdash_{A_K} \rangle$.

Recall that graphs G_1 and G_2 are understood as concept taxonomies. Thus G_1 and G_2 can be characterised as theories $T_k = \langle N_k, \vdash_{E_k} \rangle$, where $\vdash_{E_k}$ is the smallest consequence relation over N_k that includes E_k ($k = 1, 2$), *i.e.*, we take the sets of nodes as languages and the set of edges as theory presentations.

It is straightforward to prove that the induced map $\varphi_k : T_k \rightarrow T$ is a theory interpretation ($k = 1, 2$). Suppose that $\Gamma \vdash_{T_k} \Delta$. It follows from the definition of $\vdash_{T_k}$ that there exist nodes $n \in \Gamma$ and $m \in \Delta$ such that m is reachable from n in G_k, that is, $m \in \uparrow n$. Consequently, by the way φ_k is the defined, $\varphi_k(m)$ is a conjunct of $\varphi_k(n)$. Therefore $\{\varphi_k(n)\} \vdash_T \{\varphi_k(m)\}$, and hence $\varphi_k[\Gamma] \vdash_T \varphi_k[\Delta]$.

We have proven that graph theories T_1 and T_2 are semantically integrated with respect to $T = \langle L_{\text{WordNet}}, \vdash_{A_K} \rangle$ by means of $\mathcal{I} = \{\varphi_k : T_k \rightarrow T\}_{k=1,2}$, but the really interesting point is to show that S-Match output indeed captures the integration theory $T_{\mathcal{I}}$. Actually, we have:

- $n \sqsubseteq m$ if, and only if, $\{n\} \vdash_{T_{\mathcal{I}}} \{m\}$

- $n \sqsupseteq m$ if, and only if, $\{m\} \vdash_{T_{\mathcal{I}}} \{n\}$

- $n \equiv m$ if, and only if, $\{n\} \vdash_{T_{\mathcal{I}}} \{m\}$ and $\{m\} \vdash_{T_{\mathcal{I}}} \{n\}$

- $n \perp m$ if, and only if, $\{n, m\} \vdash_{T_{\mathcal{I}}} \emptyset$

The following proves the first item. The remainder is proven analogously.

$$\begin{array}{lll} n \sqsubseteq m & \text{iff} & A_K \text{ implies } \varphi_1(n) \to \varphi_2(m) \text{ in propositional logic} \\ & \text{iff} & \{\varphi_1(n)\} \vdash_{A_K} \{\varphi_2(m)\} \\ & \text{iff} & \varphi_1[\{n\}] \vdash_{A_K} \varphi_2[\{m\}] \\ & \text{iff} & \varphi_1[\{n\}] \vdash_T \varphi_2[\{m\}] \\ & \text{iff} & \{n\} \vdash_{(\varphi_1+\varphi_2)^{-1}[T]} \{m\} \\ & \text{iff} & \{n\} \vdash_{T_\mathcal{I}} \{m\} \end{array}$$

This concludes the section related to ontology matching. We have provided a formal foundation for semantic alignment and presented a number of current state-of-the-art matching techniques. In the next section we particularly study the problem of semantic heterogeneity in multiagent systems, and we explain how ontology matching is generally applied in this context. This allows us to highlight some limitations of standard approximations for ontology matching.

2.3 The Case of Multiagent Systems

Multiagent systems are considered one of the best ways to characterise or design distributed computing systems [Huhns and Stephens, 1999], and the Semantic Web has not overlooked this appreciation [Hendler, 2001]. In the Semantic Web vision, agents are claimed to assist users to get what they need of the Web, with a prerequisite: Web content must be machine-readable.

In multiagent communication one usually assumes that agents use a shared terminology with the same meaning for message passing. If agents, however, are engineered separately one has to foresee that, when they interact, they will most likely make use of different terminology in their respective messages, and that, if some terms coincide, they may not have the same meaning for all agents participating in an interaction. This is the problem of semantic heterogeneity as it arises in multiagent systems.

As hinted in the beginning of this chapter, ontologies have also been proposed to achieve semantic interoperability in this context [Nodine and Unruh, 1998, Steels, 1998, Chaib-Draa and Dignum, 2002]. As well as ontology matching has been the immediate later step. In multiagent systems, matching is generally addressed as depicted in Figure 2.11. Given two distinct ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ as input, a semantic alignment $\mathcal{A}$ is generated as output of a specific matcher. The resulting alignment $\mathcal{A}$ is the basis for a translator through which agent communication is done: if an agent sends a message to another agent, the latter will receive its translation. Alternatively, bridge axioms can be generated and incorporated to one of the ontologies [Euzenat and Shvaiko, 2007]. There exist many works that deal with ontology matching in the context of multiagent systems [van Eijk et al., 2001, Wiesman et al., 2002].

Dynamism is a primary factor to be considered when matching ontologies. Matching applications can indeed be classified by this criterion, ranging from

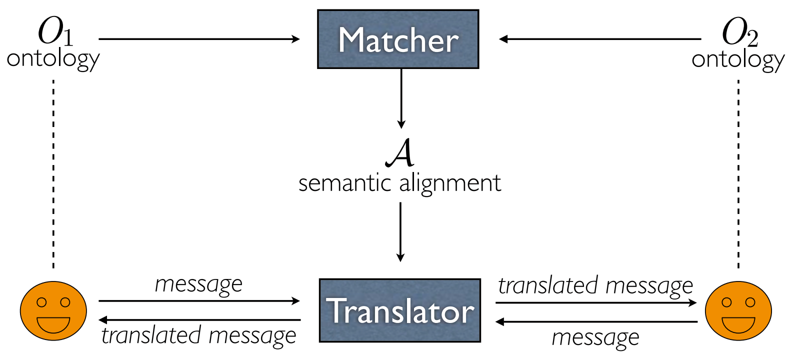


Figure 2.11: Matching in multiagent communication

those of a entirely static nature, such as large business-to-business applications and schema integration, to more dynamic applications, such as query answering, semantic peer-to-peer networks and multiagent systems.

Agents, as peers in a P2P network, have the ability to enter or leave the network, or to change their ontologies at any moment. Matching is thereby required at run-time, rather than at design-time, and taking only those ontology fragments that are necessary for the task at hand. There exist efforts that match ontological entities at run-time [López et al., 2006, McNeill and Bundy, 2007], and approaches that are specifically centered on agent capabilities —reactivity, pro-activity and social ability— and that take advantage of mechanisms for agent coordination, negotitation or argumentation.

In [Bailin and Truszkowski, 2003], the authors propose a protocol that allows agents to discover ontology conflicts, and, through incremental interpretation, clarification and explanation, establish a common basis for communicating with each other. With the same spirit, van Diggelen et al.'s ANEMONE enables agents to gradually build towards a semantically integrated system by creating minimal and effective shared ontologies [van Diggelen et al., 2006]. Laera et al present an argumentation framework for the creation and exchange of arguments that support or reject possible ontological correspondences [Laera et al., 2007]. Wang and Gasser put forward a framework for mutual online concept learning, where agents can collectively design concepts [Wang and Gasser, 2002].

2.4 Summary and Concluding Remarks

Ontologies have been advocated as an appropriate answer to the problem of semantic heterogeneity in open and distributed systems. Nevertheless, the quick proliferation of ontologies has yielded another kind of heterogeneity: that of ontology heterogeneity. Ontology matching is considered a promising approach to resolve this new problem.

In this chapter we have explained in some detail the insights of ontologies and ontology matching, and we have presented a number of state-of-the-art matching techniques. The case of multiagent systems has been specifically studied, and we have pointed at the lack of dynamism as one of the main drawbacks of most current matching approaches.

In the following chapter we continue this line of reasoning and we highlight the often overlooked situation dependence of matching. Our first proposal to overcome these limitations is then presented.

Chapter 3

A Formal Framework for Situated Semantic Alignment

Abstract. In this chapter we present a formal model for a semantic alignment procedure that incrementally aligns differing conceptualisations of two, or more, agents relative to their respective perception of the environment or domain where they are acting. In this way, we make the situation in which the alignment occurs explicit in the model. Our formalisation is founded on channel theory, Barwise and Seligman's theory of information flow. The content of this chapter has been published in [Atencia and Schorlemmer, 2006, Atencia and Schorlemmer, 2007].

3.1 Situated Semantic Alignment

Chapter 2 includes a survey of the most important state-of-the-art matching techniques. In general, all these techniques follow a classical functional approach to the semantic heterogeneity problem, in which ontology matching is seen as a process taking two or more ontologies as input, and producing a semantic alignment of ontological entities as output. Even when these techniques are applied at run-time, they exploit *a priori* defined concept taxonomies as they are represented in the graph-based structures of the ontologies to be matched, use previously existing external sources such as thesauri (*e.g.*, WordNet) and upper-level ontologies (*e.g.*, Cyc or SUMO), or resort to additional background knowledge repositories or shared instances.

We claim, however, that the semantic alignment of ontological terminology is ultimately relative to the specific situation in which the alignment is carried out, and that this situation should be somehow made explicit and brought into the alignment mechanism. Even two agents with identical conceptualisation capabilities, and using exactly the same vocabulary to specify their respective conceptualisations may fail to interoperate in a concrete situation because of their differing perception of the domain. Imagine, for example, a situation in which two agents are facing each other in front of a checker board. Agent A_1 may conceptualise a figure on the board as situated on the left margin of the board, while agent A_2 may conceptualise the same figure as situated on the right. Although the conceptualisation of 'left' and 'right' is done in exactly the same manner by both agents, and even if both use the terms *left* and *right* in their communication, they still will need to align their respective vocabularies if they want to successfully communicate actions to each other that change the position of figures on the checker board. Their semantic alignment, though, will only be valid in the scope of their interaction within this particular situation or environment. The same agents situated differently may produce a different alignment.

This scenario is reminiscent to those in which a group of distributed agents adapt to form an ontology and a shared lexicon in an emergent, bottom-up manner, with only local interactions and no centralised control authority (see [Steels, 1998]). This sort of self-organised emergence of shared meaning is in the end grounded on the physical interaction of agents with the environment. In this chapter, though, we address the case in which agents are already endowed with a top-down engineered ontology (it can even be the same one), which they do not adapt or refine, but for which they want to discover the semantic relationships with separate ontologies of other agents on the grounds of their communication within a specific situation.

We provide a formal model that formalises *situated semantic alignment* as a sequence of information-channel refinements in the sense of channel theory [Barwise and Seligman, 1997]. This theory is particularly useful for our endeavour since it models the flow of information occurring in distributed systems due to the particular situations —or tokens— that carry information. Analogously, the semantic alignment that will allow information to flow ultimately will be carried by the particular situation agents are acting in. We do not assume any knowledge of channel theory. All terms and theorems used along this chapter can be found in Appendix A, but any detailed exposition of the theory is outside the scope of this dissertation.

We shall therefore consider a scenario with two or more agents situated in an environment. Each agent will have its own viewpoint of the environment, so that, if the environment is in a concrete state, both agents may have different perceptions of the state. Because of these differences there may be a mismatch in the meaning of the syntactic entities by which agents describe their perceptions (and which constitute the agents' respective ontologies). These syntactic entities can be related according to the intrinsic semantics provided by the existing relationship between the agents' viewpoint of the environment. The existence of this relationship is justified precisely by the fact that the agents are situated and observe the same environment.

In Section 3.2 we describe any situated semantic alignment as a distributed logic in the sense of Barwise and Seligman's theory. A method by which agents can obtain approximations of this distributed logic is explained in Section 3.3. A couple of illustrative examples are also provided.

3.2 The Logic of Situated Semantic Alignment

Consider a scenario with two agents A_1 and A_2 situated in an environment E (the generalisation to any numerable set of agents is straightforward). We associate a numerable set S of states to E and, at any given instant, we suppose E to be in one of these states. We further assume that each agent is able to observe the environment and has its own perception of it. This ability is faithfully captured by a surjective function $see_i : S \to P_i$, where $i \in \{1, 2\}$, and typically see_1 and see_2 are different.

According to channel theory, information is only viable where there is a systematic way of classifying some range of things as being this way or that, in other words, where there is a classification (see Section A.1). So in order to be within the framework of channel theory, we must associate classifications with the components of our system.

For each $i \in \{1, 2\}$, we consider a classification $\mathbf{A}_i$ that models A_i's point of view of E. First, $tok(\mathbf{A}_i)$ is composed of A_i's perceptions of E states, that is, $tok(\mathbf{A}_i) = P_i$. Second, $typ(\mathbf{A}_i)$ contains the syntactic entities by which A_i describes its perceptions, the ones constituting the ontology of A_i. Finally, $\models_{\mathbf{A}_i}$ synthesises how A_i relates its perceptions with these syntactic entities.

Now, with the aim of associating environment E with a classification $\mathbf{E}$ we choose the *power classification* of S as $\mathbf{E}$, which is the classification whose set of types is equal to 2^S, whose tokens are the elements of S, and for which a token e is of type ε if $e \in \varepsilon$. The reason for taking the power classification is because there are no syntactic entities that may play the role of types for $\mathbf{E}$, since, in general, there is no global conceptualisation of the environment. However, the set of types of the power classification includes all possible token configurations potentially described by types. Thus $tok(\mathbf{E}) = S$, $typ(\mathbf{E}) = 2^S$ and $e \models_{\mathbf{E}} \varepsilon$ if and only if $e \in \varepsilon$.

The notion of channel (see Section A.1) is fundamental in Barwise and Seligman's theory. The information flow among the components of a distributed system is modelled in terms of a channel and the relationships among these components are expressed via infomorphisms (see Section A.1) which provide a way of moving information between them.

The information flow of the scenario under consideration is then accurately described by channel $\mathcal{E} = \{f_i : \mathbf{A}_i \to \mathbf{E}\}_{i\in\{1,2\}}$ defined as follows:

- $f_i^{\to}(\alpha) = \{e \in tok(\mathbf{E}) \mid see_i(e) \models_{\mathbf{A}_i} \alpha\}$ for each $\alpha \in typ(\mathbf{A}_i)$,
- $f_i^{\leftarrow}(e) = see_i(e)$ for each $e \in tok(\mathbf{E})$.

Definition of $f_i^{\leftarrow}$ seems natural while $f_i^{\rightarrow}$ is defined in such a way that the fundamental property of the infomorphisms is fulfilled:

$$\begin{array}{llll} f_i^{\leftarrow}(e) \models_{\mathbf{A}_i} \alpha & \text{iff} & see_i(e) \models_{\mathbf{A}_i} \alpha & \text{(by definition of } f_i^{\leftarrow}) \\ & \text{iff} & e \in f_i^{\rightarrow}(\alpha) & \text{(by definition of } f_i^{\rightarrow}) \\ & \text{iff} & e \models_{\mathbf{E}} f_i^{\rightarrow}(\alpha) & \text{(by definition of } \models_{\mathbf{E}}) \end{array}$$

Consequently, $\mathbf{E}$ is the core of channel $\mathcal{E}$ and a state $e \in tok(\mathbf{E})$ connects agents' perceptions $f_1^{\leftarrow}(e)$ and $f_2^{\leftarrow}(e)$ (see Figure 3.1).

$$\begin{tikzcd}
 & typ(\mathbf{E}) \arrow[dd, dash, dashed, "\models_{\mathbf{E}}"] & \\
typ(\mathbf{A}_1) \arrow[ur, "f_1^{\rightarrow}"] \arrow[dd, dash, dashed, "\models_{\mathbf{A}_1}"'] & & typ(\mathbf{A}_2) \arrow[ul, "f_2^{\rightarrow}"'] \arrow[dd, dash, dashed, "\models_{\mathbf{A}_2}"] \\
 & tok(\mathbf{E}) \arrow[dl, "f_1^{\leftarrow}"'] \arrow[dr, "f_2^{\leftarrow}"] & \\
tok(\mathbf{A}_1) & & tok(\mathbf{A}_2)
\end{tikzcd}$$

Figure 3.1: Channel $\mathcal{E}$

Channel $\mathcal{E}$ explains the information flow of our scenario by virtue of agents A_1 and A_2 being situated and perceiving the same environment E. We want to obtain meaningful relations among agents' syntactic entities, that is, agents' types. We state that meaningfulness must be in accord with $\mathcal{E}$.

The sum operation (see Section A.1) gives us a way of putting the two agents' classifications of channel $\mathcal{E}$ together into a single classification, namely $\mathbf{A}_1 + \mathbf{A}_2$, and also the two infomorphisms together into a single infomorphism, $f_1 + f_2 : \mathbf{A_1} + \mathbf{A}_2 \rightarrow \mathbf{E}$.

The set $\mathbf{A}_1 + \mathbf{A}_2$ assembles agents' classifications in a very coarse way. $tok(\mathbf{A}_1 + \mathbf{A}_2)$ is the cartesian product of $tok(\mathbf{A}_1)$ and $tok(\mathbf{A}_2)$, that is, $tok(\mathbf{A}_1 + \mathbf{A}_2) = \{\langle p_1, p_2 \rangle \mid p_i \in P_i\}$, so a token of $\mathbf{A}_1 + \mathbf{A}_2$ is a pair of agents' perceptions with no restrictions. The set $typ(\mathbf{A}_1 + \mathbf{A}_2)$ is the disjoint union of $typ(\mathbf{A}_1)$ and $typ(\mathbf{A}_2)$, and $\langle p_1, p_2 \rangle$ is of type $\langle i, \alpha \rangle$ if p_i is of type α. We attach importance to take the disjoint union because A_1 and A_2 could use identical types with the purpose of describing their respective perceptions of E.

Classification $\mathbf{A}_1 + \mathbf{A}_2$ seems to be the natural place in which to search for relations among agents' types. Now, channel theory provides a way to make all these relations explicit in a logical fashion by means of theories and local logics (see Section A.1). The theory generated by the sum classification, $Th(\mathbf{A}_1 + \mathbf{A}_2)$, and hence its logic generated, $Log(\mathbf{A}_1 + \mathbf{A}_2)$, involve all those constraints among agents' types valid according to $\mathbf{A}_1 + \mathbf{A}_2$. Notice though that these constraints are obvious; as stated above, meaningfulness must be in accord with channel $\mathcal{E}$.

Classifications $\mathbf{A}_1 + \mathbf{A}_2$ and $\mathbf{E}$ are connected via the sum infomorphism, $f = f_1 + f_2$ (see Figure 3.2), where:

- $f^{\rightarrow}(\langle i, \alpha \rangle) = f_i^{\rightarrow}(\alpha) = \{e \in tok(\mathbf{E}) \mid see_i(e) \models_{\mathbf{A}_i} \alpha\}$ for each $\langle i, \alpha \rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$,
- $f^{\leftarrow}(e) = \langle f_1^{\leftarrow}(e), f_2^{\leftarrow}(e) \rangle = \langle see_1(e), see_2(e) \rangle$ for each $e \in tok(\mathbf{E})$.

$$\begin{tikzcd}
& \mathbf{E} & \\
\mathbf{A}_1 \arrow[ur, "f_1"] \arrow[r, "\sigma_1"'] & \mathbf{A}_1 + \mathbf{A}_2 \arrow[u, "f"'] & \mathbf{A}_2 \arrow[ul, "f_2"'] \arrow[l, "\sigma_2"]
\end{tikzcd}$$

Figure 3.2: the sum classification (σ_1 and σ_2 are the natural injections).

Meaningful constraints among agents' types are in accord with channel $\mathcal{E}$ because they are computed making use of f as it is shown below.

As important as the notion of channel is the concept of distributed logic (see Section A.1). Given a channel $\mathcal{C}$ and a logic $\mathfrak{L}$ on its core, $DLog_{\mathcal{C}}(\mathfrak{L})$ represents the reasoning about relations among the components of $\mathcal{C}$ justified by $\mathfrak{L}$. If $\mathfrak{L} = Log(\mathbf{C})$, the distributed logic, denoted by $Log(\mathcal{C})$, captures the information flow inherent in the channel in a logical fashion.

In our case, $Log(\mathcal{E})$ explains the relationship between the agents' point of view of the environment in a logical fashion. On the one hand, constraints of $Th(Log(\mathcal{E}))$ are defined by:

$$\Gamma \vdash_{Log(\mathcal{E})} \Delta \quad \text{if} \quad f^{\rightarrow}[\Gamma] \vdash_{Log(\mathbf{E})} f^{\rightarrow}[\Delta] \tag{3.1}$$

where $\Gamma, \Delta \subseteq typ(\mathbf{A}_1 + \mathbf{A}_2)$. On the other hand, the set of normal tokens, $N_{Log(\mathcal{E})}$, is equal to the range of function $f^{\leftarrow}$:

$$\begin{aligned} N_{Log(\mathcal{E})} &= f^{\leftarrow}[tok(\mathbf{E})] \\ &= \{\langle see_1(e), see_2(e) \rangle \mid e \in tok(\mathbf{E})\} \end{aligned}$$

Therefore, a normal token is a pair of agents' perceptions that are restricted by coming from the same environment state (unlike tokens of $\mathbf{A}_1 + \mathbf{A}_2$).

All constraints of $Th(Log(\mathcal{E}))$ are satisfied by all normal tokens (because of being a logic). In this particular case, this condition is also sufficient (the proof is straightforward); as alternative to (3.1) we have:

$$\begin{array}{lll} \Gamma \vdash_{Log(\mathcal{E})} \Delta & \text{iff} & \text{for all } e \in tok(\mathbf{E}), \\ & & \text{if } (\forall \langle i, \gamma \rangle \in \Gamma)[see_i(e) \models_{\mathbf{A}_i} \gamma] \\ & & \text{then } (\exists \langle j, \delta \rangle \in \Delta)[see_j(e) \models_{\mathbf{A}_j} \delta] \end{array} \tag{3.2}$$

where $\Gamma, \Delta \subseteq typ(\mathbf{A}_1 + \mathbf{A}_2)$.

Definition 1 *$Log(\mathcal{E})$ is the* logic of SSA*.*

$Th(Log(\mathcal{E}))$ comprises the most meaningful constraints among agents' types according to channel $\mathcal{E}$. In other words, the logic of SSA contains, and also justifies, the most meaningful relations among those syntactic entities that agents use in order to describe their own environment perceptions.

$Log(\mathcal{E})$ is complete since $Log(\mathbf{E})$ is complete but it is not necessarily sound because although $Log(\mathbf{E})$ is sound, $f^{\leftarrow}$ is not surjective in general (see Section A.2). If $Log(\mathcal{E})$ is also sound then $Log(\mathcal{E}) = Log(\mathbf{A}_1 + \mathbf{A}_2)$ (see Section A.2). That means there is no significant relation between agents' points of view of the environment according to $\mathcal{E}$. It is only the fact that $Log(\mathcal{E})$ is unsound what allows a significant relation between the agents' viewpoints. This relation is expressed at the type level in terms of constraints by $Th(Log(\mathcal{E}))$ and at the token level by $N_{Log(\mathcal{E})}$.

3.2.1 A First Example: the Magic Box

This example is taken from [Ghidini and Giunchiglia, 2001]. There are two observers, $Mr.1$ and $Mr.2$, each having a partial viewpoint of a box. This box consists of six sectors and each sector can enclose a ball. The box is "magic" because observers are not able to distinguish the depth inside it. Figure 3.3 shows the scenario we are describing and it illustrates schematically what $Mr.1$ and $Mr.2$ can observe.

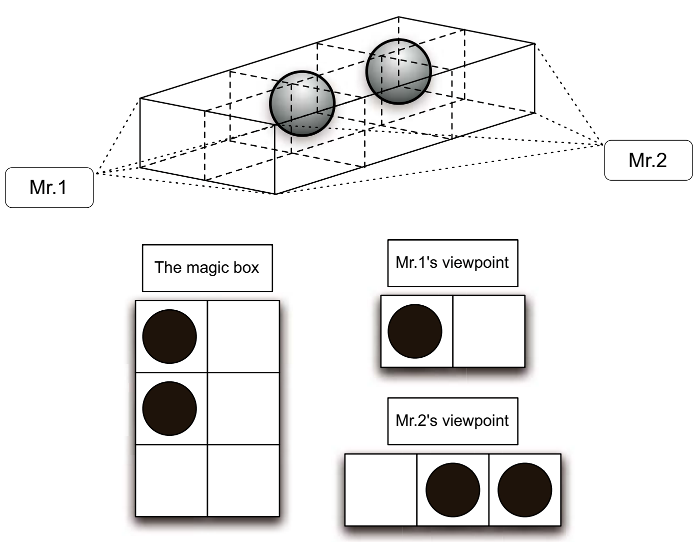


Figure 3.3: The magic box scenario

Now, regarding our approach, the magic box plays the role of E and $Mr.1$ and $Mr.2$ are the agents. The states of E are the possible configurations of the box, so a state e can be represented as a 3×2 binary matrix (e_{ij}). In this way:

$$S = \left\{ \begin{pmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ e_{31} & e_{32} \end{pmatrix} \mid e_{ij} \in \{0,1\} \right\}$$

Intuitively, the state $\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}$, for instance, means that there are only two balls in the box, one in the left-up sector and another in the left centered sector. Notice that $Mr.1$ sees a ball on the left if there is a ball in at least one of the sectors of the left column of the box and sees a ball on the right provided that there is ball in at least one of the sectors of the right column. Then $Mr.1$'s perceptions of the states of E can be represented as two-dimensional binary vectors, and the function see_1 can be defined formally as follows:

$$see_1(e) = \begin{cases} (0\,0) & \text{if} \quad e = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \\ (1\,0) & \text{if} \quad e_{11} + e_{21} + e_{31} \geq 1 \text{ and } e_{12} + e_{22} + e_{32} = 0 \\ (0\,1) & \text{if} \quad e_{11} + e_{21} + e_{31} = 0 \text{ and } e_{12} + e_{22} + e_{32} \geq 1 \\ (1\,1) & \text{if} \quad e_{11} + e_{21} + e_{31} \geq 1 \text{ and } e_{12} + e_{22} + e_{32} \geq 1 \end{cases}$$

where $e \in S$. The function see_2 can be defined analogously on three-dimensional binary vectors as representations of $Mr.2$'s perceptions. Note that $Mr.1$ has the notions of a ball being on the left or on the right and so has $Mr.2$, but this last one has also the notion of a ball being in the center.

After these considerations, $\mathbf{A}_1$ is defined by:

- $tok(\mathbf{A}_1) = \{(1\,1), (1\,0), (0\,1), (0\,0)\}$
- $typ(\mathbf{A}_1) = \{\mathsf{left}, \mathsf{right}\}$
- $\begin{array}{lll} (v_1\,v_2) \models_{\mathbf{A}_1} \mathsf{left} & \text{iff} & v_1 = 1 \\ (v_1\,v_2) \models_{\mathbf{A}_1} \mathsf{right} & \text{iff} & v_2 = 1 \end{array}$

On the other hand, $\mathbf{A}_2$ is defined by:

- $tok(\mathbf{A}_2) = \{(1\,1\,1), (1\,1\,0), (1\,0\,1), (0\,1\,1), (1\,0\,0), (0\,1\,0), (0\,0\,1), (0\,0\,0)\}$
- $typ(\mathbf{A}_2) = \{\mathsf{left}, \mathsf{centre}, \mathsf{right}\}$
- $\begin{array}{lll} (v_1\,v_2\,v_3) \models_{\mathbf{A}_2} \mathsf{left} & \text{iff} & v_1 = 1 \\ (v_1\,v_2\,v_3) \models_{\mathbf{A}_2} \mathsf{centre} & \text{iff} & v_2 = 1 \\ (v_1\,v_2\,v_3) \models_{\mathbf{A}_2} \mathsf{right} & \text{iff} & v_3 = 1 \end{array}$

Channel $\mathcal{E} = \{f_i : \mathbf{A}_i \to \mathbf{E}\}_{i \in \{1,2\}}$ as defined above explains the information flow of this scenario (see Figure 3.4).

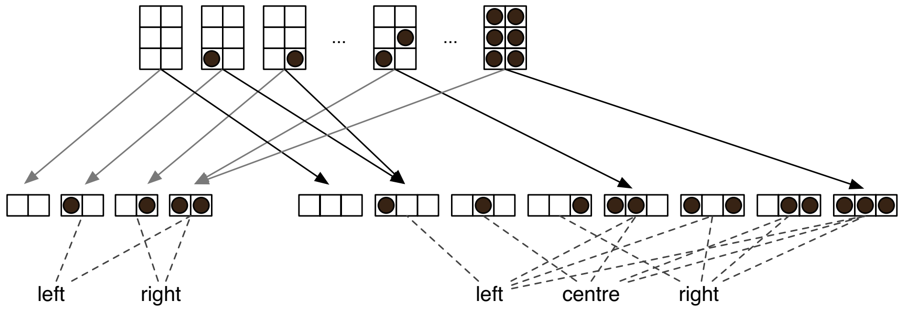


Figure 3.4: The magic box channel

Finally, $Log(\mathcal{E})$ includes among its constraints:

$$\langle 1, \mathsf{left}\rangle \vdash_{Log(\mathcal{E})} \langle 2, \mathsf{left}\rangle, \langle 2, \mathsf{centre}\rangle, \langle 2, \mathsf{right}\rangle$$
$$\langle 1, \mathsf{right}\rangle \vdash_{Log(\mathcal{E})} \langle 2, \mathsf{left}\rangle, \langle 2, \mathsf{centre}\rangle, \langle 2, \mathsf{right}\rangle$$

Let us check the first constraint (the second one can be verified in a similar way). We have to prove:

$$f^{\rightarrow}(\langle 1, \mathsf{left}\rangle) \vdash_{Log(\mathbf{E})} f^{\rightarrow}(\langle 2, \mathsf{left}\rangle), f^{\rightarrow}(\langle 2, \mathsf{centre}\rangle), f^{\rightarrow}(\langle 2, \mathsf{right}\rangle)$$

Let $e = (e_{ij}) \in tok(\mathbf{E})$ and assume that $e \models_{\mathbf{E}} f^{\rightarrow}(\langle 1, \mathsf{left}\rangle)$. Then $e \in f^{\rightarrow}(\langle 1, \mathsf{left}\rangle)$, that is, $see_1(e) \models_{\mathbf{A}_1} \mathsf{left}$. Therefore $e_{11} + e_{21} + e_{31} \geq 1$ and we can ensure that there exists $i \in \{1, 2, 3\}$ such that $e_{i1} = 1$. Hence $e_{i1} + e_{i2} \geq 1$. If $i = 3$ then $e \models_{\mathbf{E}} f^{\rightarrow}(\langle 2, \mathsf{left}\rangle)$, if $i = 2$ then $e \models_{\mathbf{E}} f^{\rightarrow}(\langle 2, \mathsf{centre}\rangle)$ and if $i = 1$ then $e \models_{\mathbf{E}} f^{\rightarrow}(\langle 2, \mathsf{right}\rangle)$.

The necessary and sufficient conditions for a token $\langle a_1, a_2\rangle$ of $\mathbf{A}_1 + \mathbf{A}_2$ being normal are the following:

$$\text{if } a_1 \neq (0\,0) \text{ then } a_2 \neq (0\,0\,0)$$
$$\text{if } a_2 \neq (0\,0\,0) \text{ then } a_1 \neq (0\,0)$$

A normal token corresponds to what the authors define as a *model for the magic box* in [Ghidini and Giunchiglia, 2001]. There contextual reasoning is presented as a combination of two principles: *locality* and *compatibility*. The former expresses the fact that "reasoning uses only part of what is potentially available (*e.g.*, what is known, the available inference procedures)". The part being used while reasoning is the context (of reasoning); but there must be a compatibility among the reasoning performed in different contexts.

In our case, "locality" is represented by agent classifications, along with their associated theories or logics, whereas a channel and its distributed logic account for the existing "compatibility".

3.3 Approaching the logic of SSA

We have dubbed $Log(\mathcal{E})$ the logic of SSA. The theory $Th(Log(\mathcal{E}))$ comprises the most meaningful constraints amongst agents' types according to $\mathcal{E}$. But the problem is that neither agent can make use of this theory because they do not know $\mathcal{E}$ completely. In this section, we will present a method by which agents obtain approximations to $Th(Log(\mathcal{E}))$. Moreover, we prove that these approximations gradually become more reliable as the method is applied.

Agents can obtain approximations to $Th(Log(\mathcal{E}))$ through *communication*. A_1 and A_2 communicate by exchanging information about their perceptions of environment states. Now, this information is expressed in terms of their own classification relations. Specifically, if E is in a concrete state e, agents can convey to each other which types are satisfied by their respective perceptions of e and which are not. This exchange generates a channel $\mathcal{C} = \{f_i : \mathbf{A}_i \to \mathbf{C}\}_{i\in\{1,2\}}$ and $Th(Log(\mathcal{C}))$ contains the constraints among agents' types justified by the fact that agents have observed e. But if E turns to another state e' and agents proceed as before, another channel $\mathcal{C}' = \{f_i' : \mathbf{A}_i \to \mathbf{C}'\}_{i\in\{1,2\}}$ gives account of the new situation considering also the previous information. $Th(Log(\mathcal{C}'))$ comprises the constraints among agents' types justified by the fact that agents have observed e and e'. The significant point is that $\mathcal{C}'$ is a refinement of $\mathcal{C}$ (see Section A.1). Theorem 3.1 below ensures that the refined channel involves more reliable information.

The communication supposedly ends when both agents have observed all the environment states. Again this situation can be modelled by a channel, call it $\mathcal{C}^* = \{f_i^* : \mathbf{A}_i \to \mathbf{C}^*\}_{i\in\{1,2\}}$. Theorem 3.2 states that $Th(Log(\mathcal{C}^*)) = Th(Log(\mathcal{E}))$.

Theorem 3.1 and Theorem 3.2 ensure that applying the method agents can gradually obtain more reliable approximations to $Th(Log(\mathcal{E}))$.

Theorem 3.1 *Let* $\mathcal{C} = \{f_i : \mathbf{A}_i \to \mathbf{C}\}_{i\in\{1,2\}}$ *and* $\mathcal{D} = \{g_i : \mathbf{A}_i \to \mathbf{D}\}_{i\in\{1,2\}}$ *be two channels. If* $\mathcal{D}$ *is a* refinement *of* $\mathcal{C}$ *then:*

1. $Th(Log(\mathcal{D})) \subseteq Th(Log(\mathcal{C}))$
2. $N_{Log(\mathcal{D})} \supseteq N_{Log(\mathcal{C})}$

Proof. Since $\mathcal{D}$ is a refinement of $\mathcal{C}$ then there exists a refinement infomorphism r from $\mathcal{D}$ to $\mathcal{C}$, and thereby $f_i = r \circ g_i$. Let $\mathbf{A} =_{def} \mathbf{A}_1 + \mathbf{A}_2$, $f =_{def} f_1 + f_2$ and $g =_{def} g_1 + g_2$.

1. Let Γ and Δ be subsets of $typ(\mathbf{A})$ and assume that $\Gamma \vdash_{Log(\mathcal{D})} \Delta$, which means $g^{\rightarrow}[\Gamma] \vdash_{\mathbf{D}} g^{\rightarrow}[\Delta]$. We have to prove $\Gamma \vdash_{Log(\mathcal{C})} \Delta$ or, equivalently, $f^{\rightarrow}[\Gamma] \vdash_{\mathbf{C}} f^{\rightarrow}[\Delta]$. We proceed by reductio ad absurdum. Suppose $c \in tok(\mathbf{C})$ does not satisfy the sequent $\langle f^{\rightarrow}[\Gamma], f^{\rightarrow}[\Delta]\rangle$. Then $c \models_{\mathbf{C}} f^{\rightarrow}(\gamma)$ for all $\gamma \in \Gamma$ and $c \not\models_{\mathbf{C}} f^{\rightarrow}(\delta)$ for all $\delta \in \Delta$. Let us choose an arbitrary $\gamma \in \Gamma$. We have that $\gamma = \langle i, \alpha\rangle$ for some $\alpha \in typ(\mathbf{A}_i)$ and $i \in \{1,2\}$. Thus $f^{\rightarrow}(\gamma) = f^{\rightarrow}(\langle i, \alpha\rangle) = f_i^{\rightarrow}(\alpha) = r^{\rightarrow} \circ g_i^{\rightarrow}(\alpha) = r^{\rightarrow}(g_i^{\rightarrow}(\alpha))$. Therefore:

$$\begin{array}{rll} c \models_{\mathbf{C}} f^{\rightarrow}(\gamma) & \text{iff} & c \models_{\mathbf{C}} r^{\rightarrow}(g_i^{\rightarrow}(\alpha)) \\ & \text{iff} & r^{\leftarrow}(c) \models_{\mathbf{D}} g_i^{\rightarrow}(\alpha) \\ & \text{iff} & r^{\leftarrow}(c) \models_{\mathbf{D}} g^{\rightarrow}(\langle i, \alpha \rangle) \\ & \text{iff} & r^{\leftarrow}(c) \models_{\mathbf{D}} g^{\rightarrow}(\gamma) \end{array}$$

Consequently, $r^{\leftarrow}(c) \models_{\mathbf{D}} g^{\rightarrow}(\gamma)$ for all $\gamma \in \Gamma$. Since $g^{\rightarrow}[\Gamma] \vdash_{\mathbf{D}} g^{\rightarrow}[\Delta]$ then there exists $\delta^* \in \Delta$ such that $r^{\leftarrow}(c) \models_{\mathbf{D}} g^{\rightarrow}(\delta^*)$. A sequence of equivalences similar to the above one justifies $c \models_{\mathbf{C}} f^{\rightarrow}(\delta^*)$, contradicting that c is a counterexample to $\langle f^{\rightarrow}[\Gamma], f^{\rightarrow}[\Delta] \rangle$. Hence $\Gamma \vdash_{Log(\mathcal{C})} \Delta$ as we wanted to prove.

2. Let $\langle a_1, a_2 \rangle \in tok(\mathbf{A})$ and assume $\langle a_1, a_2 \rangle \in N_{Log(\mathcal{C})}$. Thus there exists c token in $\mathbf{C}$ such that $\langle a_1, a_2 \rangle = f^{\leftarrow}(c)$. Then we have $a_i = f_i^{\leftarrow}(c) = g_i^{\leftarrow} \circ r^{\leftarrow}(c) = g_i^{\leftarrow}(r^{\leftarrow}(c))$, for $i \in \{1, 2\}$. Hence $\langle a_1, a_2 \rangle = g^{\leftarrow}(r^{\leftarrow}(c))$ and $\langle a_1, a_2 \rangle \in N_{Log(\mathcal{D})}$. Consequently, $N_{Log(\mathcal{D})} \supseteq N_{Log(\mathcal{C})}$ which concludes the proof.

Q.E.D.

Remark Theorem 3.1 asserts that the more refined channel gives more reliable information. Even though its theory has less constraints, it has more normal tokens to which they apply.

In the remainder of the section, this process of communication is explicitly described. We conclude with the proof of Theorem 3.2.

Let us assume that $typ(\mathbf{A}_i)$ is finite for $i \in \{1, 2\}$ and S is countably infinite, though the finite case can be treated in a similar way. We also choose an infinite numerable set of symbols $\{c^n \mid n \in \mathbb{N}\}$.[1]

From here on, we will omit informorphism superscripts when no confusion arises. Types are usually denoted by Greek letters and tokens by Latin letters, so, if f is an infomorphism, $f^{\rightarrow}(\alpha)$ and $f^{\leftarrow}(a)$ will be replaced by $f(\alpha)$ and $f(a)$, respectively.

Agent communication starts from the observation of E. Let us suppose that E is in state $e^1 \in S = tok(\mathbf{E})$. A_1's perception of e^1 is $f_1(e^1)$ and A_2's perception of e^1 is $f_2(e^1)$. We take for granted that A_1 can communicate A_2 those types that are and are not satisfied by $f_1(e^1)$ according to its classification $\mathbf{A}_1$. So can A_2 do. Since both $typ(\mathbf{A}_1)$ and $typ(\mathbf{A}_2)$ are finite, this process eventually finishes. After this communication a channel $\mathcal{C}^1 = \{f_i^1 : \mathbf{A}_i \rightarrow \mathbf{C}^1\}_{i=1,2}$ arises (see Figure 3.5). On the one hand, $\mathbf{C}^1$ is defined by:

- $tok(\mathbf{C}^1) = \{c^1\}$,
- $typ(\mathbf{C}^1) = typ(\mathbf{A}_1 + \mathbf{A}_2)$,
- $c^1 \models_{\mathbf{C}^1} \langle i, \alpha \rangle$ if $f_i(e^1) \models_{\mathbf{A}_i} \alpha$ for every $\langle i, \alpha \rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$.

[1]We write these symbols with superscripts to limit the use of subscripts for what concerns to agents. Note that this set is chosen to have the same cardinality as that of S.

On the other hand, f_i^1, with $i \in \{1,2\}$, is defined by:

- $f_i^1(\alpha) = \langle i, \alpha \rangle$ for every $\alpha \in typ(\mathbf{A}_i)$,
- $f_i^1(c^1) = f_i(e^1)$.

$$
\begin{tikzcd}
 & \mathbf{C}^1 & \\
\mathbf{A}_1 \arrow[ur, "f_1^1"] & & \mathbf{A}_2 \arrow[ul, "f_2^1"']
\end{tikzcd}
$$

Figure 3.5: First stage of communication

$Log(\mathcal{C}^1)$ represents the reasoning about the first stage of communication. It is easy to prove that $Th(Log(\mathcal{C}^1)) = Th(\mathbf{C}^1)$. The significant point is that both agents know $\mathbf{C}^1$ as the result of the communication. Therefore they are able to compute theory $Th(\mathbf{C}^1) = \langle typ(\mathbf{C}^1), \vdash_{\mathbf{C}^1} \rangle$ separately, which contains the constraints among agents' types justified by the fact that agents have observed state e^1.

Now, let us assume that E turns to a new state e^2. Agents can proceed as before, exchanging this time information about their perceptions of e^2. Another channel $\mathcal{C}^2 = \{f_i^2 : \mathbf{A}_i \to \mathbf{C}^2\}_{i \in \{1,2\}}$ comes up. We define $\mathcal{C}^2$ so as to take also into account the information provided by the previous stage of communication. On the one hand, $\mathbf{C}^2$ is defined by:

- $tok(\mathbf{C}^2) = \{c^1, c^2\}$,
- $typ(\mathbf{C}^2) = typ(\mathbf{A}_1 + \mathbf{A}_2)$,
- $c^k \models_{\mathbf{C}^2} \langle i, \alpha \rangle$ if $f_i(e^k) \models_{\mathbf{A}_i} \alpha$ for each $k \in \{1,2\}$ and $\langle i, \alpha \rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$.

On the other hand, f_i^2, with $i \in \{1,2\}$, is defined by:

- $f_i^2(\alpha) = \langle i, \alpha \rangle$ for every $\alpha \in typ(\mathbf{A}_i)$,
- $f_i^2(c^k) = f_i(e^k)$ for every $k \in \{1,2\}$.

$Log(\mathcal{C}^2)$ represents the reasoning about the former and later communication stages. $Th(Log(\mathcal{C}^2))$ is equal to $Th(\mathbf{C}^2) = \langle typ(\mathbf{C}^2), \vdash_{\mathbf{C}^2} \rangle$, so it comprises constraints among agents' types justified by the fact that agents have observed e^1 and e^2. Since A_1 and A_2 knows $\mathbf{C}^2$, they can make use of these constraints. The key point is that channel $\mathcal{C}^2$ is a refinement of $\mathcal{C}^1$. It is easy to check that f^1 defined as the identity function on types and the inclusion function on tokens is a refinement infomorphism (see Figure 3.6). By virtue of Theorem 3.1, $\mathbf{C}^2$ constraints are more reliable than $\mathbf{C}^1$ constraints.

In the general situation, once the states $e^1, e^2, \ldots, e^{n-1}$ $(n \geq 2)$ have been observed and a new state e^n appears, channel $\mathcal{C}^n = \{f_i^n : \mathbf{A}_i \to \mathbf{C}^n\}_{i \in \{1,2\}}$ accounts for agent communication up to that moment. The definition of $\mathcal{C}^n$ is

$$\begin{tikzcd}
 & \mathbf{C}^2 \arrow[d, "f^1"] & \\
\mathbf{A}_1 \arrow[ur, "f_1^2"] \arrow[r, "f_1^1"'] & \mathbf{C}^1 & \mathbf{A}_2 \arrow[ul, "f_2^2"'] \arrow[l, "f_2^1"]
\end{tikzcd}$$

Figure 3.6: Second stage of communication

similar to the previous ones and analogous remarks can be made (see at the top of Figure 3.7). Theory $Th(Log(\mathcal{C}^n)) = Th(\mathbf{C}^n) = \langle typ(\mathbf{C}^n), \vdash_{\mathbf{C}^n} \rangle$ comprises constraints among agents' types justified by the fact that agents have observed $e^1, e^2, \ldots, e^n$.

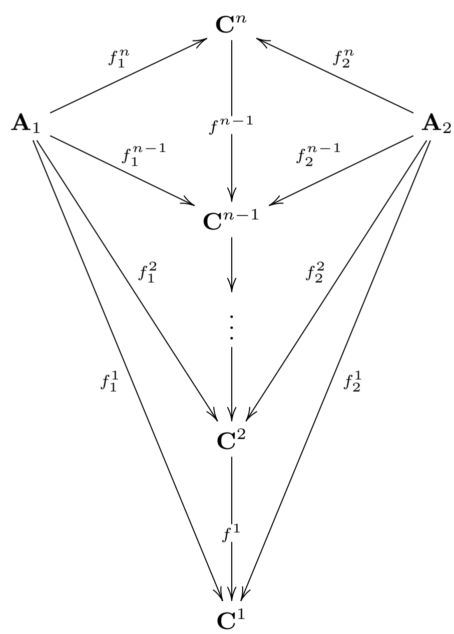


Figure 3.7: Agent communication

Remember that S is an infinite numerable set of states. Consequently, agents are not capable of observing all environment states. It is up to them to decide when to stop communicating, given a justified evidence that the information gathered so far is good enough for the task at hand. But the study of possible termination criteria is outside the scope of this thesis and left for future work. From a theoretical point of view, however, we can consider the channel $\mathcal{C}^* = \{f_i^* : \mathbf{A}_i \to \mathbf{C}^*\}_{i \in \{1,2\}}$ which accounts for the end of the communication when all environment states have been observed. On the one hand, $\mathbf{C}^*$ is defined by:

- $tok(\mathbf{C}^*) = \{c^n \mid n \in \mathbb{N}\}$,
- $typ(\mathbf{C}^*) = typ(\mathbf{A}_1 + \mathbf{A}_2)$,
- $c^n \models_{\mathbf{C}^*} \langle i, \alpha\rangle$ if $f_i(e^n) \models_{\mathbf{A}_i} \alpha$ for $n \in \mathbb{N}$ and $\langle i, \alpha\rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$.

On the other hand, f_i^*, with $i \in \{1, 2\}$, is defined by:

- $f_i^*(\alpha) = \langle i, \alpha\rangle$ for $\alpha \in typ(\mathbf{A}_i)$,
- $f_i^*(c^n) = f_i(e^n)$ for $n \in \mathbb{N}$.

The theorem below constitutes the cornerstone of the model exposed in this chapter. It ensures, together with Theorem 3.1, that at each communication stage agents obtain a theory that approximates more closely to the theory generated by the logic of SSA.

Theorem 3.2 *The following statements hold:*

1. *For all $n \in \mathbb{N}$, $\mathcal{C}^*$ is a refinement of $\mathcal{C}^n$.*
2. *$Th(Log(\mathcal{E})) = Th(\mathbf{C}^*) = Th(Log(\mathcal{C}^*))$.*

Proof.

1. It is easy to prove that for each $n \in \mathbb{N}$, g^n defined as the identity function on types and the inclusion function on tokens is a refinement infomorphism from $\mathcal{C}^*$ to $\mathcal{C}^n$.

2. The second equality is straightforward; the first one follows directly from:

$$
\begin{array}{lll}
c^n \models_{\mathbf{C}^*} \langle i, \alpha\rangle & \text{iff} & f_i^{\leftarrow}(e^n) \models_{\mathbf{A}_i} \alpha \\
 & & \text{(by definition of } \models_{\mathbf{C}^*}) \\
 & \text{iff} & e^n \models_{\mathbf{E}} f_i^{\rightarrow}(\alpha) \\
 & & \text{(because } f_i \text{ is infomorphim)} \\
 & \text{iff} & e^n \models_{\mathbf{E}} f^{\rightarrow}(\langle i, \alpha\rangle) \\
 & & \text{(by definition of } f^{\rightarrow})
\end{array}
$$

$$
\begin{tikzcd}
 & \mathbf{E} & \\
 & \mathbf{C}^* \arrow[dd, "g^n"] & \\
\mathbf{A}_1 \arrow[uur, "f_1"] \arrow[ur, "f_1^*"'] \arrow[dr, "f_1^n"'] & & \mathbf{A}_2 \arrow[uul, "f_2"'] \arrow[ul, "f_2^*"] \arrow[dl, "f_2^n"] \\
 & \mathbf{C}^n &
\end{tikzcd}
$$

Q.E.D.

3.3.1 An Example with Robots

Let us reflect on a system consisting of robots located in a two-dimensional grid looking for packages with the aim of moving them to a certain destination (Figure 3.8). Robots can carry only one package at a time and they can not move through a package.

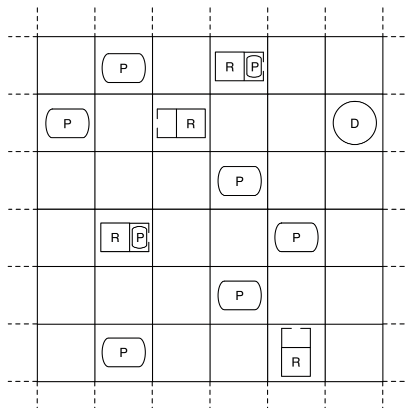


Figure 3.8: Robots (R) carrying packages (P) to a destination (D)

Robots have a partial view of the domain and there exist two kinds of robots according to the visual field they have. Some robots are capable of observing the eight adjoining squares, but others just observe the three squares they have in front (see Figure 3.9). We call them URDL robots (shortened form of Up-Right-Down-Left) and LCR robots (abbreviation for Left-Center-Right), respectively.

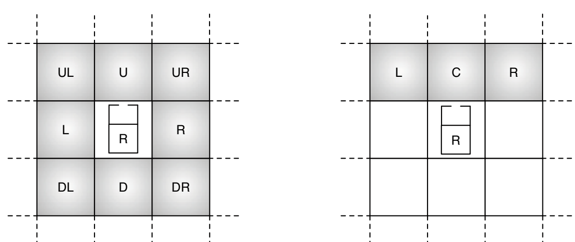


Figure 3.9: Robot visual fields

Describing the environment states as well as the robots' perception functions is rather tedious and even unnecessary. We assume that the reader has all those descriptions in mind.

All robots in the system must be able to solve package distribution problems cooperatively by communicating their intentions to each other. In order to communicate, agents send messages using some ontology. In our scenario, there coexist two ontologies, the UDRL and LCR ontologies. Both of them are very simple and are just confined to describing what robots observe.

When a robot carrying a package finds another package obstructing its way, it can either go around it or, if there is another robot in its visual field, ask it for assistance. Let us suppose two URDL robots are in a situation like the one depicted in Figure 3.10. Robot1 (the one carrying a package) decides to ask Robot2 for assistance and sends a request. This request is written as a KQML message [Labrou and Finin, 1997]:

(**request**

:sender	Robot1
:receiver	Robot2
:language	Package-distribution language
:ontology	URDL-ontology
:content	(PICK UP $\mathsf{U(Package)}$ BECAUSE $\mathsf{UR(Robot2)}$)

It should be read as "Robot2, pick up the package located in my 'up' square since you are located in my 'up-right' square". Robot2 understands the content of this request and makes use of a rule represented by the following constraint:

$$\langle 1, \mathsf{UR(Robot2)}\rangle, \langle 2, \mathsf{UL(Robot1)}\rangle, \langle 1, \mathsf{U(Package)}\rangle \vdash \langle 2, \mathsf{U(Package)}\rangle$$

This constraint should be read as "if Robot2 is situated in Robot1's 'up-right' square, Robot1 is situated in Robot2's 'up-left' square and a package is located in Robot1's 'up' square, then a package is located in Robot2's 'up' square".

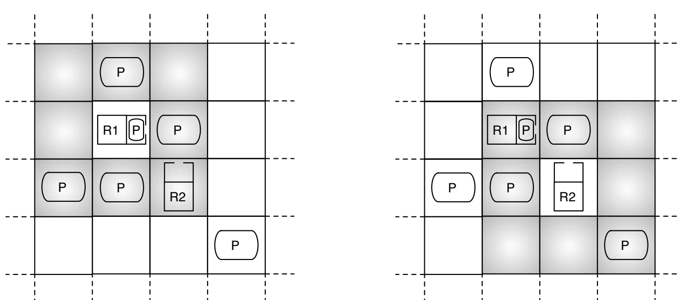


Figure 3.10: Robot assistance

Now, problems arise as soon as a LCR robot and an URDL robot meet and try to interoperate (see Figure 3.11). Robot1 sends a request of the form:

(**request**

:sender	Robot1
:receiver	Robot2
:language	Package-distribution language
:ontology	LCR-ontology
:content	(PICK UP $\mathsf{R}(\mathsf{Robot2})$ BECAUSE $\mathsf{C}(\mathsf{Package})$))

Robot2 does not understand the content of the request but they decide to begin an alignment process —corresponding to channel $\mathcal{C}^1$. Once finished, Robot2 searches in $Th(\mathbf{C}^1)$ for constraints similar to the expected one, that is, those ones of the form:

$$\langle 1, \mathsf{R}(\mathsf{Robot2})\rangle, \langle 2, \mathsf{UL}(\mathsf{Robot1})\rangle, \langle 1, \mathsf{C}(\mathsf{Package})\rangle \vdash_{\mathbf{C}^1} \langle 2, \lambda(\mathsf{Package})\rangle$$

where $\lambda \in \{\mathsf{U}, \mathsf{R}, \mathsf{D}, \mathsf{L}, \mathsf{UR}, \mathsf{DR}, \mathsf{DL}, \mathsf{UL}\}$. From these ones, only the following are plausible according to $\mathcal{C}^1$:

$$\begin{array}{lll}
\langle 1, \mathsf{R}(\mathsf{Robot2})\rangle, \langle 2, \mathsf{UL}(\mathsf{Robot1})\rangle, \langle 1, \mathsf{C}(\mathsf{Package})\rangle & \vdash_{\mathbf{C}^1} & \langle 2, \mathsf{U}(\mathsf{Package})\rangle \\
\langle 1, \mathsf{R}(\mathsf{Robot2})\rangle, \langle 2, \mathsf{UL}(\mathsf{Robot1})\rangle, \langle 1, \mathsf{C}(\mathsf{Package})\rangle & \vdash_{\mathbf{C}^1} & \langle 2, \mathsf{L}(\mathsf{Package})\rangle \\
\langle 1, \mathsf{R}(\mathsf{Robot2})\rangle, \langle 2, \mathsf{UL}(\mathsf{Robot1})\rangle, \langle 1, \mathsf{C}(\mathsf{Package})\rangle & \vdash_{\mathbf{C}^1} & \langle 2, \mathsf{DR}(\mathsf{Package})\rangle
\end{array}$$

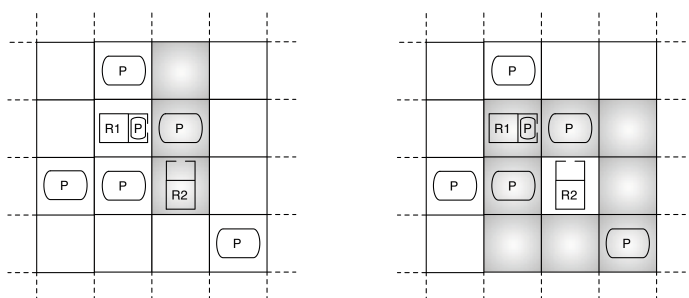


Figure 3.11: Ontology mismatching

If both robots, adopting the same roles, take part in a situation like the one depicted in Figure 3.12, a new alignment process —corresponding to channel $\mathcal{C}^2$— takes place. $\mathcal{C}^2$ also considers the previous information and refines $\mathcal{C}^1$. The only constraint from the above ones that remains plausible according to $\mathcal{C}^2$ is:

$$\langle 1, \mathsf{R}(\mathsf{Robot2})\rangle, \langle 2, \mathsf{UL}(\mathsf{Robot1})\rangle, \langle 1, \mathsf{C}(\mathsf{Package})\rangle \vdash_{\mathbf{C}^2} \langle 2, \mathsf{U}(\mathsf{Package})\rangle$$

Notice that this constraint is an element of the theory of the distributed logic. Agents communicate in order to cooperate successfully and success is guaranteed using constrains of the distributed logic.

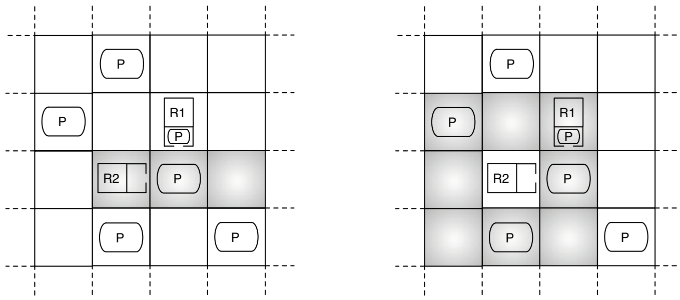


Figure 3.12: Refinement

3.4 Summary and Concluding Remarks

In this chapter we have described a formal framework for semantic alignment as a sequence of information-channel refinements that are relative to the concrete states of the environment in which two agents communicate and align their respective state conceptualisations.

Before us, Kalfoglou and Schorlemmer [Kalfoglou and Schorlemmer, 2003a, Schorlemmer and Kalfoglou, 2005] and Kent [Kent, 2005] have also applied channel theory to formalise semantic alignment using Barwise and Seligman's insight to focus on tokens as the enablers of information flow. Their approach to semantic alignment, though, like most ontology matching mechanisms developed to date (regardless of whether they follow a standard functional and design-time-based approach, or an interaction-based, run-time-based approach), still defines semantic alignment in terms of *a priori* design decisions such as the concept taxonomy of the ontologies or the external sources brought into the alignment process. Instead the model we have presented in this chapter makes explicit the particular states of the environment in which agents are situated and are attempting to gradually align their ontological entities.

Also similar to the work of this chapter is [De Saeger and Shimojima, 2007], where de Saeger and Shimojima put forward a channel theoretic account for modelling agent reasoning in context.

In the following chapter, we study the case of agents that need to align their respective ontologies in the context of a specific interaction. Within the same spirit, we make semantic alignment interaction-dependent, as it is based on the state of the interaction in which agents are engaged.

Chapter 4

I-SSA: Interaction-Situated Semantic Alignment

Abstract. In this chapter we address the problem of semantic heterogeneity in multiagent communication by looking at semantics related to interaction. This approach is called Interaction-Situated Semantic Alignment (or I-SSA), as it takes the state of the interaction agents are engaged in as the basis on which the semantic alignment rests. We provide a formal foundation of I-SSA by means of a mathematical object inspired from category theory called the communication product. We present an alignment protocol and several matching mechanisms for agents to benefit from this technique in practice. The core aspects of this chapter have been published in [Atencia and Schorlemmer, 2008a] and [Atencia and Schorlemmer, 2008b].

4.1 Introduction

At the beginning of Chapter 3 we pointed out a number of limitations of current approaches to tackle semantic heterogeneity in multiagent communication, and we presented Situated Semantic Alignment (SSA) as an attempt to overcome these limitations. SSA makes the situation in which the alignment occurs explicit, as the semantic alignment is just based on agents' perceptions of the states of the environment where the communication takes place. In this way, we manage to avoid dependency on *a priori* semantic agreements, a common characteristic of most of the state-of-the-art matching techniques as we have seen in Chapter 2. Furthermore, SSA proposes a procedure by which agents can incrementally align their terminologies, and thereby matching is dynamically performed, which is another desired feature when addressing semantic heterogeneity in open multiagent communication.

Chapter 3 describes a formal framework for SSA which is exemplified with a couple of examples around agents situated in a physical environment. Recall that, in both examples, agents' terms are about objects, positions and actions in that environment. SSA is helpful precisely because agents' state perceptions are potentially enough to discriminate between all agents' terms. However, it is still unclear what these perceptions are in a scenario where agents are situated in a non-physical environment.

In this chapter, we shall address the case in which agents need to establish the semantic relationships with terminologies of other agents on the grounds of their communication within a specific interaction. In line with SSA, we claim that semantic alignment is ultimately relative to the particular interaction in which agents are engaged, and, more precisely, to the specific state of the interaction. For this reason, we propose Interaction-Situated Semantic Alignment, or I-SSA, for short. I-SSA looks at the semantics of messages that are exchanged during an interaction entirely from an interaction-specific viewpoint: messages are deemed semantically related if they trigger compatible interaction state transitions — where compatibility means that the interaction progresses in the same direction for each agent, albeit their partial view of the interaction (their interaction model) may be more simple than the interaction that is actually happening.

The insights of I-SSA are synthesised in a number of principles that are listed in Section 4.4.1. We provide a formal foundation of the I-SSA principles by means of a mathematical construct inspired from category theory: the communication product. This formalisation takes up all of Section 4.5. After this pure theoretical work, we show in Section 4.6 how the I-SSA technique can be put into practice through the combination of an alignment protocol (Section 4.6.1) along with a matching mechanism (Section 4.6.2). I-SSA is presented as a particular case of SSA in Section 4.7. A summary and concluding remarks close the chapter.

In order to make all these formal ideas clear we will make reference to an example. This running example is introduced next.

4.2 A Running Example: the Blackjack Game

In what follows we show the basic insights of the Blackjack game and justify why we have chosen it as a running example for this chapter. Readers already familiar with the Blackjack rules can skip Section 4.2.1 and go directly to Section 4.2.2. Blackjack rules, though, are important to fully understand both Figures 4.1 and 4.2 depicted in Section 4.3.

4.2.1 Blackjack Rules

We are solely concerned with the communication acts that arise during a Blackjack interaction, as well as anything that directly influences the communication itself, everything else regarding Blackjack has been discarded.

The Blackjack game unfolds by players trying to beat a dealer separately. Regardless the number of players, it is always a one-to-one interaction. Since each player has an independent game with the dealer, it is possible for the dealer to lose to some players but still beat the others in the same round.

The hand with the highest total wins as long as it does not exceed 21, in which case the hand is said to be *bust* or *too many*. The total calculation is made regarding these card values: cards 2 through 10 are worth their face value, court cards are all worth 10, and an ace's value is 11 unless it results in a bust hand, in which case it is worth 1. Each player's goal is to beat the dealer by having the higher, unbusted hand. Note that if a player busts then she loses, even if the dealer also busts. When both player and dealer have the same point value, this is called a *push*, and neither player nor dealer wins the hand. A two-card hand of 21 (an ace plus a ten-value card) is called a *blackjack* or *natural*, and it is an automatic winner unless the dealer has blackjack as well, in which case the hand is a push.

At the beginning of a new round, all players are required to place a bet. Once all players place their bets, the dealer starts to hand cards. He will make two passes around the table so that the players and the dealer end up with two cards each. Then, the dealer flips one of his cards over, exposing its value. The player's standard choices for playing a hand are enumerated bellow:

1. *Hit*: draw another card.

2. *Stand*: take no more cards, also known as *stick* or *stay*.

3. *Double down*: this can only be done with a two card hand which is not worth 21 and before another card has been drawn. Doubling down allows you to increase the bet to a maximum of double the original bet and receive one, and only one, additional card to the hand.

4. *Split*: in case that the first two cards are worth the same, a player has the possibility to split them into two separate hands by placing an additional wager, equal to the original one, on the second hand.

5. *Surrender*: just after the deal, players usually have the option to surrender half their wager and forfeit the hand.

6. *Insurance*: when the dealer's face-up card is an ace, players will be offered an insurance against a possible blackjack (if the hidden card is a ten-value card). Players who wish to take this option can bet an amount up to half their original bets, and if the dealer does have a blackjack, insurance wagers are paid at odds of 2:1. Obviously, all players lose their initial bets (except players who also have blackjack, who push). If the dealer ends up not having a blackjack, the game is continued, but the insurance bet is forfeited.

Once all players' turns are over, the dealer's hidden card is revealed. The dealer must hit until she has at least 17, regardless of what players have. If the dealer busts then all remaining players win. Bets are normally paid out at the odds of 1:1. Players who push with the dealer receive their original bet back.

4.2.2 Why Blackjack?

So far we have only explained the basic Blackjack rules. It is time to look at this example from a more scientific perspective. There are two main reasons that led us to choose Blackjack as a running example for this chapter.

A wide amount of versions. There exist over one hundred variations of this game, and each variation has its own rules and strategies. American casinos, for example, offer the insurance option, whereas European casino players do not have this choice. In some casinos, the dealer wins ties (which certainly is very unfavourable to the players). There also exist casinos where players automatically win when five cards have been drawn without busting (this rule is commonly known as *charlie* or *five card trick*). And so forth. Of course, in a real casino, players are advised to take a look at the rules before playing, so everyone is aware of the Blackjack version they will play.

A very specific and varying terminology. At this point, the reader will probably agree that the Blackjack terminology is rather particular. Terms like "hit", "push" or "bust" are attached to very specific meanings, and it seems an arduous task to figure out these meanings outside this context. Additionally, the Blackjack terminology varies considerably. We have been showing this varying nature in Section 4.2.1: "bust" is also known as "too many", "stand" as "stick" or "stay", and "blackjack" as "natural", among others.

These two aspects make Blackjack especially useful for our endeavour. In the proposed running example, we shall consider a scenario in which two software agents try to play Blackjack, one as dealer and another as player, but following different specifications. These specifications may refer to dissimilar Blackjack versions, or, though related to the same version, contain distinct terminologies. Both cases are plausible for the reasons stated above. We will describe the general case of this problem in Section 4.4 when talking about the I-SSA insights, but before that the notion of interaction model is to presented.

4.3 Interaction Models

When dealing with software agents, communication is the result of a message-passing process. For instance, if a player agent in a Blackjack interaction decides to stand, it can send a message with "stand" as content to the dealer agent. But this message should contain more information. To begin with, it should specify by whom, and to whom, this message is addressed. In a multiagent system, it is assumed that each agent is associated with an *agent identifier*, but this identification should be also accompanied by the *roles* the message sender or receiver are playing. In fact, an agent may play more than one role during the same interaction, and, depending on the role that it is playing, certain messages may not be addressed to it. In addition, the content of a message is often attached to a particle that gives information about its illocutionary force, in order to ensure that there is no doubt about the message type. Indeed a stand decision entails the commitment not to receive any other card, and to wait for the hand final result. All things considered, if an agent, identified with id_1 and in the role of a player, decides to stand, it can send the following message to the dealer agent, possibly identified with id_2:

$$\langle commit, (id_1 : player), (id_2 : dealer), \texttt{stand}\rangle \tag{4.1}$$

The expression in (4.1) is often referred to as *illocution*, and its ingredients are the *illocutionary particle* (in this case, *commit*), the *sender* and *receiver* ($(id_1 : player)$ and $(id_2 : dealer)$, respectively), and the illocution *content* (there `stand`). The term "message" usually alludes to the illocution content, though in this work we will use it to refer both the illocution and its content if no misunderstanding arises (as we have already done above).

So far we have described what is called a *communication protocol*. Multiagent systems conform to communication protocols so as to enable agents to exchange and understand messages. Besides all the elements already presented above, a communication protocol also specifies the language in which the messages are expressed (KIF [Genesereth and Fikes, 1992], Prolog, LISP, SQL), as well as an ontology —the vocabulary of the "words" in the messages. This information is indeed missing above. On the one hand, we assume that all agents agree on the language used during the communication. On the other hand, we accept the existence of more than one ontology, and, paradoxically, ontologies will not play a crucial role in this work when resolving semantic heterogeneity, since, as revealed in Section 4.1, I-SSA looks at the semantics of messages that are exchanged during an interaction entirely from an interaction-specific point of view. Examples of languages for specifying communication protocols are KQML [Labrou and Finin, 1997] and FIPA ACL [O'Brien and Nicol, 1998].

Communication protocols are mechanisms for agents to communicate single messages, while *interaction protocols* enable agents to have conversations, that is, structured exchanges of messages. Now, when dealing with software agents, interactions can be specified in many ways. One possibility is by means of finite automata, which is the formalism that we will be using throughout all this document. Finite state machines are the basis of more complex interaction-modelling formalisms, such as the well-known Petri nets [Cost et al., 2000], or electronic institutions [Arcos et al., 2005].

Figure 4.1 illustrates the message passing between a dealer and a player in a Blackjack game —specified according to the rules explained earlier in Section 4.2.1. We call this kind of automaton an *interaction model* (a formal definition is given in Section 4.5.1). It describes when the dealer and player agents are supposed to send, or to receive, particular messages; but it does not specify what reasons lead the agents to choose a concrete action. These actions depend on, among other things, the information conveyed through illocutions, but we assume that agents hold reasoning mechanisms aside from, but connected with, the interaction model.

Transitions between states of an interaction model may be labelled by either an *illocution scheme* containing variables, or a *special transition* (as λ in Figure 4.1), the latter denoting state transitions not caused by message passing. An arc labelled with $v \mid w$ replaces two arcs. During an interaction, the variables in illocution schemata are bound to the values of the uttered illocutions. Variables are written in uppercase letters and get their values in those illocutions in which they occur preceded by a question mark (?), and these values are subsequently used in those illocutions in which the corresponding variable occurs preceded by an exclamation mark (!).

4.4 I-SSA Insights

We consider a scenario in which two or more agents participate in an interaction following different interaction models. It is assumed that the interaction models are indeed about the same kind of interaction (*e.g.*, an auction or a bargaining process). We will not go into the discussion of how this can actually be ensured. Indeed, this is not an easy task and there exist complex research issues to be tackled in this regard, but they are outside the scope of this dissertation.

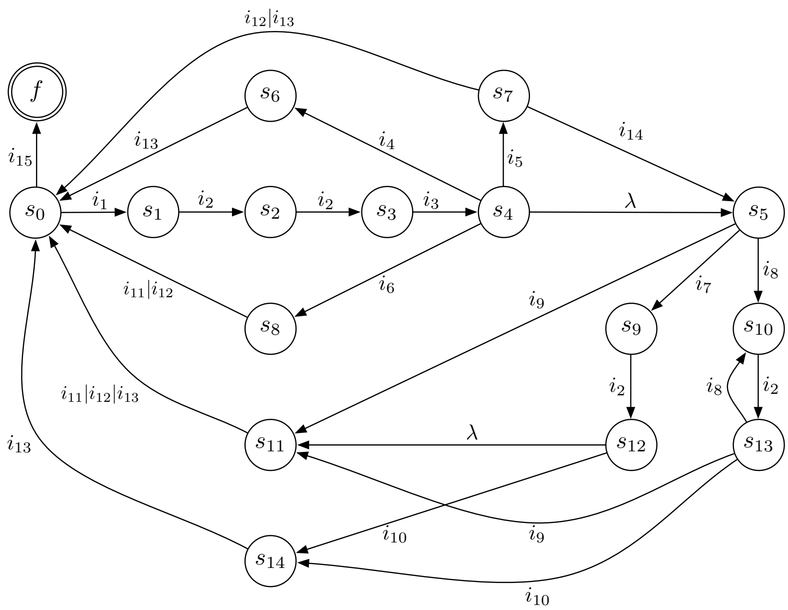
In this scenario, agents may misunderstand each other because they do not share the same ontology, or, even in the case that they make use of one single ontology, agents may be programmed to send or receive messages in different orders. Imagine, for instance, that in the Blackjack scenario the dealer follows the interaction model depicted in Figure 4.1, whereas the player follows that one depicted in Figure 4.2. Despite the use of dissimilar terminologies (`dealer_card` and `bust` against `face_up_card` and `too_many`, among others), according to the dealer's interaction model, the player can surrender and get an insurance, while these options are not available in the player's interaction model. Thus the player may want to send a surrender message to the dealer, while she is not programmed to receive it. Note also that the dealer wins ties in her interaction model, and that the charlie option is active.

In contrast to most of the state-of-the-art matching techniques, I-SSA looks at the semantics of messages that are exchanged during an interaction entirely from an interaction-specific point of view. Let us explain the I-SSA insights on the basis of the Blackjack scenario.

According to Figure 4.1, the dealer initially expects to receive a "new_hand" message or a "walk_away" message from the player. But these messages contain much more information, as they arise within illocutions:

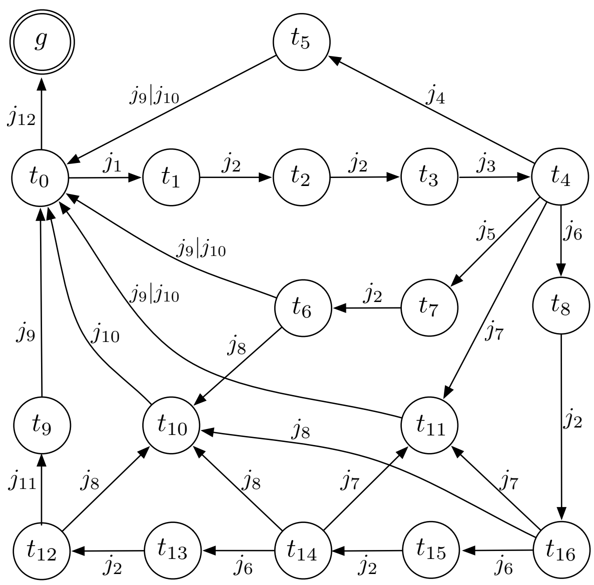
$$i_1 = \langle commit, (?A : player), (?B : dealer), \texttt{new\_hand}(?W)\rangle$$
$$i_{15} = \langle commit, (?A : player), (?B : dealer), \texttt{walk\_away}\rangle$$

Thereby the player has to *commit* to one of these options: to start a new hand for a particular bet or to directly conclude the game.



$$
\begin{aligned}
i_1 &= \langle \mathit{commit}, (?A : \mathit{player}), (?B : \mathit{dealer}), \texttt{new\_hand}(?W)\rangle \\
i_2 &= \langle \mathit{inform}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{card}(?C)\rangle \\
i_3 &= \langle \mathit{inform}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{dealer\_card}(?D)\rangle \\
i_4 &= \langle \mathit{commit}, (!A : \mathit{player}), (!B : \mathit{dealer}), \texttt{surrender}\rangle \\
i_5 &= \langle \mathit{commit}, (!A : \mathit{player}), (!B : \mathit{dealer}), \texttt{insurance}\rangle \\
i_6 &= \langle \mathit{inform}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{blackjack}\rangle \\
i_7 &= \langle \mathit{commit}, (!A : \mathit{player}), (!B : \mathit{dealer}), \texttt{double\_down}\rangle \\
i_8 &= \langle \mathit{commit}, (!A : \mathit{player}), (!B : \mathit{dealer}), \texttt{hit}\rangle \\
i_9 &= \langle \mathit{commit}, (!A : \mathit{player}), (!B : \mathit{dealer}), \texttt{stand}\rangle \\
i_{10} &= \langle \mathit{inform}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{bust}\rangle \\
i_{11} &= \langle \mathit{inform}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{win}(?X)\rangle \\
i_{12} &= \langle \mathit{inform}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{push}\rangle \\
i_{13} &= \langle \mathit{demand}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{lose}(?Y)\rangle \\
i_{14} &= \langle \mathit{demand}, (!B : \mathit{dealer}), (!A : \mathit{player}), \texttt{loss}(?Z)\rangle \\
i_{15} &= \langle \mathit{commit}, (?A : \mathit{player}), (?B : \mathit{dealer}), \texttt{walk\_away}\rangle
\end{aligned}
$$

Figure 4.1: Dealer's Blackjack interaction model



$$
\begin{aligned}
j_1 &= \langle commit, (?A : player), (?B : dealer), \texttt{wager}(?W)\rangle \\
j_2 &= \langle inform, (!B : dealer), (!A : player), \texttt{card}(?C)\rangle \\
j_3 &= \langle inform, (!B : dealer), (!A : player), \texttt{face\_up\_card}(?D)\rangle \\
j_4 &= \langle inform, (!B : dealer), (!A : player), \texttt{natural}\rangle \\
j_5 &= \langle commit, (!A : player), (!B : dealer), \texttt{stick}\rangle \\
j_6 &= \langle commit, (!A : player), (!B : dealer), \texttt{draw\_another\_card}\rangle \\
j_7 &= \langle commit, (!A : player), (!B : dealer), \texttt{no\_more\_cards}\rangle \\
j_8 &= \langle inform, (!B : dealer), (!A : player), \texttt{too\_many}\rangle \\
j_9 &= \langle inform, (!B : dealer), (!A : player), \texttt{profit}(?X)\rangle \\
j_{10} &= \langle demand, (!B : dealer), (!A : player), \texttt{debt}(?Y)\rangle \\
j_{11} &= \langle inform, (!B : dealer), (!A : player), \texttt{charlie}\rangle \\
j_{12} &= \langle inform, (?A : player), (?B : dealer), \texttt{finish}\rangle
\end{aligned}
$$

Figure 4.2: Player's Blackjack interaction model

Nevertheless, according to Figure 4.2, the player is supposed to send either a "wager" message or a "finish" message to the dealer:

$$j_1 = \langle commit, (?A : player), (?B : dealer), \texttt{wager}(?W) \rangle$$
$$j_{12} = \langle commit, (?A : player), (?B : dealer), \texttt{finish} \rangle$$

Assume that the player finally chooses to send the following to the dealer:

$$\langle commit, (a : player), (b : dealer), \texttt{wager}(w) \rangle \tag{4.2}$$

where a and b are agent identifiers, and w is a numerical value. The dealer does not understand `wager` because it does not belong to its vocabulary. Any foreign term can be matched with any local one. What makes `wager` to be matched with `new_hand` is a justified evidence that these two terms are *semantically related*. Broadly speaking, a standard matching approach would solve this problem by checking the syntactic similarity of these two terms, or looking at some external source which establishes a semantic relation between `wager` and `new_hand`, or combining those facts. I-SSA, though, takes a pragmatical stand on this issue.

4.4.1 I-SSA Principles

The I-SSA approach is founded on a number of principles. Though a formal explanation is given in Section 4.5, we prefer to give an informal account of these principles first.

Principle 1 *Whether to match a foreign term with a local one depends on the particular interaction state where the former is received.*

This principle stresses the fact that, when an agent receives a message, it is received in a particular interaction state, and, regardless of the size of the agent's vocabulary, the foreign message is to be matched with one of the local messages that the agent expects to receive at that state. In our example, the dealer agent receives $\langle commit, (a : player), (b : dealer), \texttt{wager}(w) \rangle$ at the beginning state s_0. The dealer can only receive two messages at state s_0: `new_hand` or `walk_away`. So, according to Principle 1, the dealer's matching decision comes down to these two options.

Principle 2 *Whether to match a foreign term with a local one depends on the illocutionary force with which the former is uttered.*

Messages come with performatives that inform about their illocutionary force, and Principle 2 states that, when matching two terms, their performatives must be equal. Performatives are usually realised in terms of speech act verbs such as "inform", "commit" or "demand". Unfortunately, this principle is not very helpful to discriminate between `new_hand` or `walk_away` —both come with the performative *commit*— but it will be useful as the interaction unfolds.

These first two principles rule an agent's step-by-step matching decisions. Principle 3, however, takes a global perspective.

Principle 3 *For two terms to be semantically related it is required that, if matched, the interaction eventually ends successfully.*

This principle is the most arguable as it is the most ambiguous. It certainly depends on when an interaction is dubbed "successful". One possibility is to consider an interaction to be successful as long as both agents jointly reach a final state. We therefore work with a much more specific principle:

Principle 3* *For two terms to be semantically related it is required that, if matched, the agents eventually jointly reach a final state.*

If the dealer matched `wager` with `walk_away`, the dealer would get a final state (from s_0 to s_1 in Figure 4.1), though the player would unawarely continue the interaction (changing its state from t_0 to t_3 onwards in Figure 4.2). This reason is convincing enough to opt for matching `wager` with `new_hand` instead of `walk_away`, but only subsequent events will disclose if both terms are really semantically related, that is, if agents concurrently get a final state.

We can think of reaching simultaneously a final state as a way of safely accepting that an interaction has successfully finished. If the dealer and player get final states according to their respective interaction models, the game is finished, and, if all matching decisions were valid, why not to think that they will be so in the future?

For the purpose of this work, Principle 3* is appropriate. However, the reader can find another version of Principle 3 in Chapter 7, where we present several future research lines.

These three (or four) principles are the basis of the I-SSA approach. The remainder of this section is devoted to draw conclusions from these principles.

4.4.2 Global Interaction

The term "global" has been mentioned earlier: I-SSA's third principle looks at agent interactions from a global perspective. Indeed, if the dealer and player interact by message passing, an interaction unfolds which contains more detail than the ones specified in Figures 4.1 or 4.2. These interaction models capture only a partial view of the actual *global interaction*, namely, only the view from the perspective of the dealer or the player, respectively. A global interaction model matches all messages occurring in compatible illocutions of agent interaction models, where compatibility just depends on the I-SSA principles.

Actually, neither agent needs to be aware of the model followed by the other for the interaction to unfold correctly in its totality. In general, two (or more) agents are capable of interacting following separate interaction models if their states are assumed to be projections of states of a global interaction —which, in general, is not known to each of the agents— and each state transition that separate agents follow when an illocution is uttered has a corresponding state transition in the global interaction. The key point is to provide agents with the proper mechanism to handle this situation. For this reason, an alignment protocol is presented later in Section 4.6.1. But the global interaction model itself, as a purely theoretical model, is also interesting, since it is the right place where to find the "ideal" semantic alignment. We will give a formal definition of "global interaction model" through the idea of a product of interaction models, which we will call the communication product.

All these ideas are reminiscent of those ones explained in Chapter 3. We will come back later to this issue in Section 4.7, when presenting the I-SSA semantic alignment within the general framework described there.

4.4.3 What is Shared?

Open multiagent systems tend to be as flexible as possible about communication, since, the more rigid a system is, the less agents will take part in the interactions. However, in order to communicate, something must be shared. An ontology provides a shared meaning, a content language brings a shared syntax, and protocols of communication and interaction afford the means to communicate and interact. All of this certainly puts constraints on the openness, but if these constraints are weakened, semantic heterogeneity becomes an issue. So the challenge is to get the right balance between the openness of a system and its communication rules. In the particular case of I-SSA, agents share:

A common language of performatives. Agents must agree on a common language of illocutionary particles, otherwise Principle 2 becomes useless. Now, performatives are usually realised in terms of speech act verbs. Wierzbicka's book [Wierzbicka, 1989] is a remarkable effort to define a semantic dictionary of English speech act verbs. Wierzbicka's dictionary contains definitions (or explanations) of around 250 speech act verbs. More technically, KQML contains around 35 performatives [Labrou, 1996]. These are not high numbers, so the assumption of agents sharing a language of performatives seems reasonable.

A family of roles. Senders and recipients of messages are to be identified. This is usually done by means of agent identifiers and roles, and, for this reason, agents must share a collection of roles that may even be structured in an ontology [Cáceres et al., 2006]. In this sense, no role heterogeneity is assumed.

A content language. Messages are expressed in content languages of varying complexity. Although agents typically agree on a content language of first-order expressiveness, we shall treat messages as propositions, that is, as grounded atomic sentences, leaving the generalisation to first-order sentences for future work. Let us make clear that with this assumption we mean that agents share the same language syntax, but not the same ontology.

An alignment protocol. The alignment protocol was mentioned both in the Introduction and in Section 4.4.2 when we talked about the global interaction, and it will be explained in detail in Section 4.6. This protocol helps agents to resolve semantic mismatches, and it uses a minimal set of terms the semantics of which is also supposed to be harmonised.

4.4.4 What is not Shared? I-SSA Assumptions and Goals

So what is not shared? Agents do not share any ontology. I-SSA tries to explore how we can reduce the semantic heterogeneity by assuming:

- no local ontologies for each agent,
- no global ontologies or external sources (such as upper-level ontologies, dictionaries or thesauri).

Thus we let agents agree on a language of performatives, a family of roles, a content language, and an alignment protocol, but we assume that no ontologies are shared. But if it is so, where can we find the shared commodity on which semantic alignment is based? In the interaction. In the agent's usage of terms. This interaction-situated semantic alignment contrasts with most of the current matching techniques, where semantic alignment is generally computed prior to agent interaction. These techniques, as explained at the end of Chapter 2, follow a classical functional approach, taking two or more ontologies as input and producing a semantic alignment as output. This limits the dynamism and openness of the interaction, as only agents with previously matched ontologies can participate in it. Likewise it keeps matching out of the context of the interaction. Semantic correspondences are established in an interaction-independent fashion, *e.g.*, by means of external sources such as WordNet, where semantic relatioships such as synonymy, among others, were determined prior to interaction and independently from it.

Although recent approaches apply ontology matching at interaction-time and only among those fragments of ontologies that are deemed relevant to the interaction at hand —allowing for increased openness and dynamism— such dynamic ontology matching techniques still follow a functional approach: when a mismatch occurs, semantic heterogeneity is solved by applying state-of-the-art ontology matching techniques, albeit for only a fragment. Again, although done at interaction-time, matching is still done separately from the interaction.

However, the meaning of certain terms is often very interaction-specific. The semantic similarity that exists in the context of a Blackjack game between the term "stand" and the expression "no more cards" is difficult to establish if we rely solely on syntactic-based or structural matching techniques, or even on external sources such as dictionaries and thesauri. Their meaning arises when uttered at a particular moment of the interaction happening during a Blackjack game. The I-SSA approach brings the interaction state straight into the alignment mechanism.

4.5 I-SSA Formalisation

We model a multiagent system as a set MAS of agents. Each agent in MAS has a unique identifier and may take one (or more) roles in the context of an interaction. Let *Role* be the set of roles and Id the set of agent identifiers. We write $(id : r)$, with $r \in Role$ and $id \in Id$, for the agent in MAS with identifier id playing role r.

Each agent is able to communicate by sending messages from a set M, which is local to the agent. We assume that a set $\mathfrak{I}_{\mathrm{P}}$ of *illocutionary particles* is shared by all agents.

Definition 2 *Given a non-empty set M of messages, the set of* illocutions *generated by M, denoted by $\mathfrak{I}(M)$, is the set of all tuples $\langle \iota, (id : r), (id' : r'), m\rangle$ with $\iota \in \mathfrak{I}_{\mathrm{P}}$, $m \in M$, and $(id : r), (id' : r')$ agents such that $id \neq id'$.*

If $\varphi = \langle \iota, (id : r), (id' : r'), m\rangle$ is an illocution then $(id : r)$ is the sender *of φ and $(id' : r')$ is the* receiver *of φ. In addition, $h = \langle \iota, (id : r), (id' : r')\rangle$ and m are called the* head *and* content *of φ, respectively.*

4.5.1 Interaction Models Revisited

We model an interaction model as a (partial) deterministic finite-state machine whose transitions are labelled either with illocutions, or with special transitions such as, for instance, timeouts, or null transitions (λ-transitions). Recall that state transitions prompt state changes without message passing.

Definition 3 *An* interaction model *is a tuple* $\mathrm{IM} = \langle Q, q^0, F, M, C, \delta\rangle$ *where:*

- *Q is a finite set of* states*,*
- *$q^0 \in Q$ is a distinguished element of Q called the* initial state*,*
- *F is a non-empty subset of Q the elements of which are called* final states*,*
- *M is a finite non-empty set of* messages*,*
- *C is a finite set of* special transitions*, and*
- *δ is a partial function from $Q \times (\mathfrak{I}(M) \cup C)$ to Q called the* transition function*.*

Remark Although not explicitly stated in Definition 3, for theoretical reasons we will take for granted that every interaction model contains a special transition ε such that $\delta(q, \varepsilon) = q$ for all $q \in Q$.

Given an interaction model $\mathrm{IM} = \langle Q, q^0, F, M, C, \delta\rangle$, we denote by $\mathfrak{I}_{\mathrm{IM}}$ or $\mathfrak{I}$ the subset of $\mathfrak{I}(M)$ made up of all those illocutions that appear in elements of the domain of δ. IM is associated with an automaton, $Aut(\mathrm{IM}) = \langle Q, q^0, F, \Sigma, \delta\rangle$, where $\Sigma = \mathfrak{I} \cup C$. We can then consider the language generated by $Aut(\mathrm{IM})$. The notion of history associated to an interaction model presented below is very similar to a string accepted for an automaton. The clear difference is that the former one takes the states explicitly into account.

Definition 4 *Let* IM *be an interaction model, where* IM $= \langle Q, q^0, F, M, C, \delta\rangle$*. An* IM-history *or* history *associated with* IM *is a finite sequence:*

$$h = q^0, \sigma^1, q^1, \ldots, q^{k-1}, \sigma^k, \ldots, q^{n-1}, \sigma^n, q^n$$

where $q^n \in F$ *and for each* k*:*

- $q^k \in Q$,
- $\sigma^k \in \Sigma = \mathfrak{I} \cup C$, *and*
- $\delta(q^{k-1}, \sigma^k) = q^k$.

Example *As hinted before, all messages will be treated as grounded atomic sentences. So if we replace all illocutions in Figure 4.2 with the ones bellow, we obtain an interaction model as defined in Definition 3; and similarly for Figure 4.1. These variations will be the automata under consideration in the remainder of the chapter.*

$$\begin{aligned}
j_1 &= \langle \mathit{commit}, (a : \mathit{player}), (b : \mathit{dealer}), \texttt{wager}\rangle \\
j_2 &= \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{card}\rangle \\
j_3 &= \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{face_up_card}\rangle \\
j_4 &= \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{natural}\rangle \\
j_5 &= \langle \mathit{commit}, (a : \mathit{player}), (b : \mathit{dealer}), \texttt{stick}\rangle \\
j_6 &= \langle \mathit{commit}, (a : \mathit{player}), (b : \mathit{dealer}), \texttt{draw_another_card}\rangle \\
j_7 &= \langle \mathit{commit}, (a : \mathit{player}), (b : \mathit{dealer}), \texttt{no_more_cards}\rangle \\
j_8 &= \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{too_many}\rangle \\
j_9 &= \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{profit}\rangle \\
j_{10} &= \langle \mathit{demand}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{debt}\rangle \\
j_{11} &= \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{charlie}\rangle \\
j_{12} &= \langle \mathit{inform}, (a : \mathit{player}), (b : \mathit{dealer}), \texttt{finish}\rangle
\end{aligned}$$

4.5.2 The Communication Product

We shall use the algebraic product of two interaction models in order to capture all possible interactions between agents. In general, a product of two objects is the natural algebraic construction that represents all possible behaviours of the combination of those two objects. The *communication product* (or CP) defined below, thus, captures the global interaction with respect to the message-passing behaviour of agents with two interaction models. It is not an unconstrained product, since it takes into account the compatibility of illocutions and special transitions in terms of illocutionary particles, senders, and receivers. Now, in Category Theory [Mac Lane, 1998], a constrained product is called a *pullback*. Theorem 4.1 states that the communication product is indeed a pullback in the natural category of interaction models.

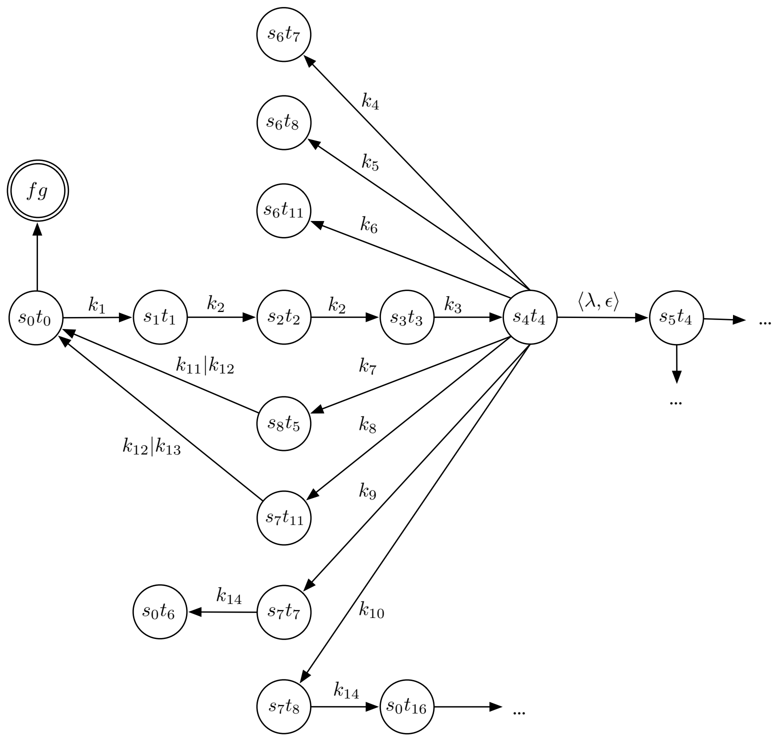
Definition 5 *Let* $\mathrm{IM}_i = \langle Q_i, q_i^0, F_i, M_i, C_i, \delta_i \rangle$ $(i = 1, 2)$ *be two interaction models. The* communication product *of* IM_1 *and* IM_2*, denoted by* $\mathrm{IM}_1 \otimes \mathrm{IM}_2$*, is the interaction model* $\langle Q, q^0, F, M, C, \delta \rangle$ *where:*

- Q *is the Cartesian product of* Q_1 *and* Q_2*, in other words, the states in* Q *are all possible ordered pairs* $\langle q_1, q_2 \rangle$ *with* $q_1 \in Q_1$ *and* $q_2 \in Q_2$*,*
- *the initial state* q^0 *is the pair* $\langle q_1^0, q_2^0 \rangle$*,*
- F *is the Cartesian product of* F_1 *and* F_2*,*
- M *is the Cartesian product of* M_1 *and* M_2*,*
- C *is the Cartesian product of* C_1 *and* C_2*,*
- δ *is defined as follows:* $\langle q_1', q_2' \rangle = \delta(\langle q_1, q_2 \rangle, \sigma)$ *if*
  - σ *is illocution* $\langle \iota, (id : r), (id' : r'), \langle m_1, m_2 \rangle \rangle$ *and, for every* i*,* $q_i' = \delta_i(q_i, \langle \iota, (id : r), (id' : r'), m_i \rangle)$*, or*
  - $\sigma = (c_1, c_2)$ *and* $q_i' = \delta_i(q_i, c_i)$ *for every* i*.*

Remark Notice that, according to the above definition of δ, $\varepsilon = \langle \varepsilon_1, \varepsilon_2 \rangle$ is such that $\delta(\langle q_1, q_2 \rangle, \varepsilon) = \langle q_1, q_2 \rangle$ for all $\langle q_1, q_2 \rangle \in Q_1 \times Q_2$. In addition, the special transitions of IM_i become also paired with the ε_j symbol of IM_j $(i \neq j)$. In this way, we capture the idea that, though the global interaction state may change, this cannot be the case for one agent interaction model.

Example *The communication product of interaction models for the dealer and player roles is partially depicted in Figure 4.3 (the total communication product is too big to be represented in a figure).*

Let us briefly explain this construction. For example, there is an arc labelled with $k_7 = \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \langle \texttt{blackjack}, \texttt{natural} \rangle \rangle$ *from state* $\langle s_4, t_4 \rangle$ *to state* $\langle s_8, t_5 \rangle$*. This is due to the fact that there is an arc labelled with* $i_6 = \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{blackjack} \rangle$ *between the states* s_4 *and* s_8 *in the dealer interaction model, there is also an arc labelled with* $j_4 = \langle \mathit{inform}, (b : \mathit{dealer}), (a : \mathit{player}), \texttt{natural} \rangle$ *between the states* t_4 *and* t_5 *in the player interaction model, and both illocution heads match up. Nevertheless, for instance, since the heads of the illocutions* i_{13} *and* j_2 *are not equal, there is no arc between states* $\langle s_6, t_7 \rangle$ *and* $\langle s_0, t_6 \rangle$ *(the latter does not even appear in the figure). Furthermore, states* $\langle s_4, t_4 \rangle$ *and* $\langle s_5, t_4 \rangle$ *are linked with* $\langle \lambda, \varepsilon \rangle$*, simply because the special transition* λ *labels an arc between* s_4 *and* s_5 *in the dealer interaction model.*



$$
\begin{aligned}
k_1 &= \langle \textit{commit}, (a : \textit{player}), (b : \textit{dealer}), \langle \texttt{new\_hand}, \texttt{wager}\rangle\rangle \\
k_2 &= \langle \textit{inform}, (b : \textit{dealer}), (a : \textit{player}), \langle \texttt{card}, \texttt{card}\rangle\rangle \\
k_3 &= \langle \textit{inform}, (b : \textit{dealer}), (a : \textit{player}), \langle \texttt{dealer\_card}, \texttt{face\_up\_card}\rangle\rangle \\
k_4 &= \langle \textit{commit}, (a : \textit{player}), (b : \textit{dealer}), \langle \texttt{surrender}, \texttt{stick}\rangle\rangle \\
k_5 &= \langle \textit{commit}, (a : \textit{player}), (b : \textit{dealer}), \langle \texttt{surrender}, \texttt{draw\_another\_card}\rangle\rangle \\
k_6 &= \langle \textit{commit}, (a : \textit{player}), (b : \textit{dealer}), \langle \texttt{surrender}, \texttt{no\_more\_cards}\rangle\rangle \\
k_7 &= \langle \textit{inform}, (b : \textit{dealer}), (a : \textit{player}), \langle \texttt{blackjack}, \texttt{natural}\rangle\rangle \\
k_8 &= \langle \textit{commit}, (a : \textit{player}), (b : \textit{dealer}), \langle \texttt{insurance}, \texttt{no\_more\_cards}\rangle\rangle \\
k_9 &= \langle \textit{commit}, (a : \textit{player}), (b : \textit{dealer}), \langle \texttt{insurance}, \texttt{stick}\rangle\rangle \\
k_{10} &= \langle \textit{commit}, (a : \textit{player}), (b : \textit{dealer}), \langle \texttt{insurance}, \texttt{draw\_another\_card}\rangle\rangle \\
k_{11} &= \langle \textit{inform}, (b : \textit{dealer}), (a : \textit{player}), \langle \texttt{win}, \texttt{profit}\rangle\rangle \\
k_{12} &= \langle \textit{inform}, (b : \textit{dealer}), (a : \textit{player}), \langle \texttt{push}, \texttt{profit}\rangle\rangle \\
k_{13} &= \langle \textit{demand}, (b : \textit{dealer}), (a : \textit{player}), \langle \texttt{loss}, \texttt{debt}\rangle\rangle \\
k_{14} &= \langle \textit{inform}, (b : \textit{dealer}), (a : \textit{player}), \langle \texttt{push}, \texttt{card}\rangle\rangle
\end{aligned}
$$

Figure 4.3: (incomplete) description of the communication product.

Definition 6 *Let* $\mathrm{IM}_i = \langle Q_i, q_i^0, F_i, M_i, C_i, \delta_i \rangle$ $(i = 1, 2)$ *be two interaction models. A* morphism of interaction models $f : \mathrm{IM}_1 \to \mathrm{IM}_2$ *is a pair of functions* $f = \langle g, h \rangle$*, where* $g : Q_1 \to Q_2$ *and* $h : \Sigma_1 \to \Sigma_2$*, such that:*

- $g(q_1^0) = q_2^0$ *and* $g(F_1) \subseteq F_2$*,*
- $h(\mathfrak{I}_1) \subseteq \mathfrak{I}_2$*,* $h(C_1) \subseteq C_2$ *and* $h(\varepsilon_1) = \varepsilon_2$*,*
- $g(\delta_1(q_1, \sigma_1)) = \delta_2(g(q_1), h(\sigma_1))$ *for all* $q_1 \in Q_1$ *and* $\sigma_1 \in \Sigma_1$*.*

From here on, if $f = \langle g, h \rangle$ is a morphism of interaction models, we will make use of the letter f both applying on states and transitions, as long as no confusion arises. Hence $f(q)$ and $f(\sigma)$ will replace $g(q)$ and $h(\sigma)$, respectively.

Definition 7 *The category of interaction models* **IM** *has interaction models as objects and morphisms of interaction models as arrows. Both composition operator and identity arrow are defined in the natural way.*

Theorem 4.1 *The communication product is a pullback in* **IM***.*

Proof. Let IM_* be the interaction model with q_* and m_* as the only state and message, respectively, and transition function δ_* defined as follows:

$$\delta_*(q_*, \langle \iota, (id : r), (id' : r'), m_* \rangle) = q_*$$

for all $\langle \iota, (id : r), (id' : r'), m_* \rangle \in \mathfrak{I}(\{m_*\})$.

Let $f_i : \mathrm{IM}_i \to \mathrm{IM}_*$ $(i = 1, 2)$ be constant to q_* on states, and be defined on transitions by $f_i(\langle \iota, (id : r), (id' : r'), m_i \rangle) = \langle \iota, (id : r), (id' : r'), m_* \rangle$ for every $\langle \iota, (id : r), (id' : r'), m_i \rangle \in \mathfrak{I}_i$, and $f_i(c_i) = \varepsilon_*$ for every $c_i \in C_i$. It is straightforward to prove that f_i is a morphism of interaction models $(i = 1, 2)$. In the remainder of the proof we show that $\mathrm{IM}_1 \otimes \mathrm{IM}_2$ is a pullback of arrows f_1 and f_2 (see Figure 4.4).

Let $\theta_i : \mathrm{IM}_1 \otimes \mathrm{IM}_2 \to \mathrm{IM}_i$ $(i = 1, 2)$ be the projection on states, and be defined by $\theta_i(\langle \iota, (id : r), (id' : r'), \langle m_1, m_2 \rangle \rangle) = \langle \iota, (id : r), (id' : r'), m_i \rangle$, whereas $\theta_i(\langle c_1, c_2 \rangle) = c_i$. θ_i is a morphism of interaction models and $f_1\theta_1 = f_2\theta_2$.

Assume that there exist two morphisms ϑ_1 and ϑ_2, $\vartheta_i : \mathrm{IM} \to \mathrm{IM}_i$ $(i = 1, 2)$, such that $f_1\vartheta_1 = f_2\vartheta_2$. We must prove that there exists a unique morphism $\xi : \mathrm{IM} \to \mathrm{IM}_1 \otimes \mathrm{IM}_2$ such that $\vartheta_i = \theta_i\xi$. Firstly, we define $\xi(q) = \langle \vartheta_1(q), \vartheta_2(q) \rangle$ on IM states. Secondly, given φ an arbitrary IM illocution, the fact that $f_1\vartheta_1 = f_2\vartheta_2$ ensures that $\vartheta_1(\varphi)$ and $\vartheta_2(\varphi)$ have the same illocution head. Let us write $\vartheta_i(\varphi) = \langle \iota, (id : r), (id' : r'), m_i \rangle$. Accordingly, we define $\xi(\varphi)$ to be equal to $\langle \iota, (id : r), (id' : r'), \langle m_1, m_2 \rangle \rangle$. And, predictably, $\xi(c) = \langle \vartheta_1(c), \vartheta_2(c) \rangle$ on special transitions. It is straightforward to prove that ξ is a morphism of interaction models. Furthermore, it is also obvious that ξ is the unique morphism in such conditions. Q.E.D.

$$\begin{tikzcd}
\mathrm{IM} \arrow[dr, dashed, "\xi"] \arrow[drr, "\vartheta_2"] \arrow[ddr, "\vartheta_1"'] & & \\
& \mathrm{IM}_1 \otimes \mathrm{IM}_2 \arrow[r, "\theta_2"] \arrow[d, "\theta_1"'] & \mathrm{IM}_2 \arrow[d, "f_2"] \\
& \mathrm{IM}_1 \arrow[r, "f_1"] & \mathrm{IM}_*
\end{tikzcd}$$

Figure 4.4: Pullback diagram

4.5.3 Semantic Alignment through the Communication Product

Being a model of all compatible interactions of varying interaction models, the communication product is the place to look for the semantic relations between messages. From a theoretical point of view, in order to establish these relations, we look at the language generated by the communication product. This formally synthesises the three I-SSA principles explained in Section 4.4.1. Messages of different interaction models are semantically related if they are paired in illocutions whose utterance make the interaction reach a final state (*i.e.*, make the interaction succeed) according to the global interaction determined by the communication product. This is formally given below. We use '$\sqsubseteq$' to denote semantic subsumption of messages, and use '$\sqcup$' to denote disjunction. Semantic equivalence between messages, denoted with '$\equiv$', arises when they subsume each other. We also pair messages with natural numbers to keep syntactically equivalent messages separate, as they may not be semantically equivalent.

Definition 8 *Let* $\mathrm{IM}_i = \langle Q_i, q_i^0, F_i, M_i, C_i, \delta_i \rangle$ $(i = 1, 2)$ *be two interaction models. Let* $m \in M_1$ *and* $m^1, \ldots, m^n \in M_2$*. We write:*

$$\langle 1, m \rangle \sqsubseteq \langle 2, m^1 \rangle \sqcup \cdots \sqcup \langle 2, m^n \rangle$$

if for all strings x *accepted by the communication product* $\mathrm{IM}_1 \otimes \mathrm{IM}_2$*, if the illocution* $\langle \iota, (id : r), (id' : r'), \langle m, m' \rangle \rangle$ *appears in* x *then* $m' = m^k$ *for some* $k \in \{1, \ldots, n\}$*. If such* $m^1, \ldots, m^n \in M_2$ *do not exist, we simply write:*

$$\langle 1, m \rangle \sqsubseteq \langle 2, \bot \rangle$$

Analogously, we define:

$$\langle 2, m \rangle \sqsubseteq \langle 1, m^1 \rangle \sqcup \cdots \sqcup \langle 1, m^n \rangle$$

We can also establish relationships among messages with regard to a specific illocution particle.

Definition 9 *Let* $\mathrm{IM}_i = \langle Q_i, q_i^0, F_i, M_i, C_i, \delta_i \rangle$ $(i = 1, 2)$ *be two interaction models. Let* $m \in M_1$ *and* $m^1, \ldots, m^n \in M_2$*. Let* ι_0 *be a fixed illocution particle. We write:*

$$\langle 1, m \rangle \sqsubseteq_{\iota_0} \langle 2, m^1 \rangle \sqcup \cdots \sqcup \langle 2, m^n \rangle$$

if for all strings x *accepted by the communication product* $\mathrm{IM}_1 \otimes \mathrm{IM}_2$*, if the illocution* $\langle \iota_0, (id : r), (id : r'), \langle m, m' \rangle \rangle$ *appears in* x *then* $m' = m^k$ *for some* $k \in \{1, \ldots, n\}$*. If such* $m^1, \ldots, m^n \in M_2$ *do not exist, we simply write:*

$$\langle 1, m \rangle \sqsubseteq_{\iota_0} \langle 2, \bot \rangle$$

Analogously, it is defined:

$$\langle 2, m \rangle \sqsubseteq_{\iota_0} \langle 1, m^1 \rangle \sqcup \cdots \sqcup \langle 1, m^n \rangle$$

The *semantic alignment* is made up of all these expressions (as in Definition 8 or Definition 9), and it represents the formal synthesis of the I-SSA principles.

Example *The semantic relationships among the dealer's and player's messages are listed below (only the semantic alignment that conforms to Definition 8 is shown, as the other is similar).*

$$
\begin{array}{rcl}
\langle 1, \texttt{new\_hand} \rangle & \equiv & \langle 2, \texttt{wager} \rangle \\
\langle 1, \texttt{walk\_away} \rangle & \equiv & \langle 2, \texttt{finish} \rangle \\
\langle 1, \texttt{card} \rangle & \equiv & \langle 2, \texttt{card} \rangle \\
\langle 1, \texttt{dealer\_card} \rangle & \equiv & \langle 2, \texttt{face\_up\_card} \rangle \\
\langle 1, \texttt{surrender} \rangle & \sqsubseteq & \langle 2, \texttt{no\_more\_cards} \rangle \\
\langle 1, \texttt{insurance} \rangle & \sqsubseteq & \langle 2, \texttt{no\_more\_cards} \rangle \\
\langle 1, \texttt{stand} \rangle & \sqsubseteq & \langle 2, \texttt{no\_more\_cards} \rangle \\
\langle 2, \texttt{no\_more\_cards} \rangle & \sqsubseteq & \langle 1, \texttt{surrender} \rangle \sqcup \langle 1, \texttt{insurance} \rangle \sqcup \langle 1, \texttt{stand} \rangle \\
\langle 1, \texttt{blackjack} \rangle & \equiv & \langle 2, \texttt{natural} \rangle \\
\langle 1, \texttt{double\_down} \rangle & \sqsubseteq & \langle 2, \texttt{stick} \rangle \sqcup \langle 2, \texttt{draw\_another\_card} \rangle \\
\langle 1, \texttt{hit} \rangle & \sqsubseteq & \langle 2, \texttt{stick} \rangle \sqcup \langle 2, \texttt{draw\_another\_card} \rangle \\
\langle 2, \texttt{stick} \rangle & \sqsubseteq & \langle 1, \texttt{double\_down} \rangle \sqcup \langle 1, \texttt{hit} \rangle \\
\langle 2, \texttt{draw\_another\_card} \rangle & \sqsubseteq & \langle 1, \texttt{double\_down} \rangle \sqcup \langle 1, \texttt{hit} \rangle \\
\langle 1, \texttt{bust} \rangle & \equiv & \langle 2, \texttt{too\_many} \rangle \\
\langle 1, \texttt{lose} \rangle & \equiv & \langle 2, \texttt{debt} \rangle \\
\langle 1, \texttt{loss} \rangle & \sqsubseteq & \langle 2, \bot \rangle \\
\langle 1, \texttt{push} \rangle & \sqsubseteq & \langle 2, \texttt{profit} \rangle \\
\langle 1, \texttt{win} \rangle & \sqsubseteq & \langle 2, \texttt{profit} \rangle \\
\langle 2, \texttt{profit} \rangle & \sqsubseteq & \langle 1, \texttt{win} \rangle \sqcup \langle 1, \texttt{push} \rangle \\
\langle 2, \texttt{charlie} \rangle & \sqsubseteq & \langle 1, \bot \rangle
\end{array}
$$

Let us have a look at these semantic relationships. First of all, `new_hand` *and* `wager` *are equivalent. If we look up these words in dictionaries or thesauri, however, we can see that they have different meanings. It is the fact that both terms trigger a commitment to play a new Blackjack game what makes them to be equivalent. Similarly,* `walk_away` *and* `finish` *are equivalent because they trigger commitments to abandon the game.*

Now, neither `surrender` *nor* `insurance` *have a counterpart in the player's interaction model (these decisions are not allowed), but they can be seen as a kind of* `no_more_cards` *decision which, as was to be expected, also subsumes* `stand`*. Similar considerations apply to* `double_down`*,* `hit`*, and their counterparts.*

There is no relation between `loss` *and any dealer's message, while* `lose` *is equivalent to* `debt`*. This is due to the fact that a* `loss` *message only follows an insurance action, which is not reflected in the player's interaction model unless it leads to a hand final. Although the dealer wins ties according to the player's interaction model,* `push` *is subsumed in* `profit`*, as well as* `win`*. Finally, there is no relation between* `charlie` *and any dealer's message.*

4.6 I-SSA Dynamics

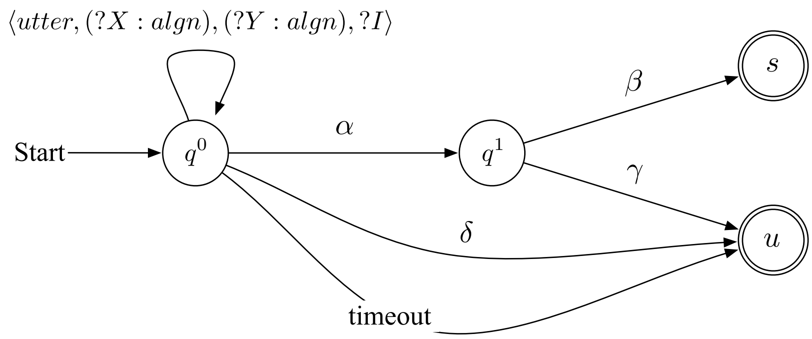
As said before, interaction models specify the space of interactions that are allowed, and their communication product captures the entire space of actual interactions when combining particular ones. The above semantic relationships are, thus, those justified by the entire space of actual interactions. This product, however, may not be accessible to agents. This is the case when interaction models are not completely open for inspection, because, for example, they are based on commercially confidential information, so agents are only aware of their local ones. Furthermore, interaction models could be of a size that makes the product computation infeasible (our running example has hinted at this).

It is therefore necessary to provide agents with a mechanism to somehow discover the above semantic relationships while interactions unfold —in the kind of manner intuitively described for our example above— assuming that for all agents participating in the interaction, the state they perceive stems from the actual global state (*i.e.*, their locally managed states are projections of the actual global state), and that occurs throughout the entire interaction.

4.6.1 The Alignment Protocol

Let us consider a scenario where two agents A_1 and A_2, identified with id_1 and id_2, try to interact following (possibly distinct) interaction models IM_1 and IM_2, respectively. Let us also assume that no other agents will take part in the interaction according to IM_1 and IM_2. A conversation among n agents can always be split into at most $n(n-1)$ conversations between two agents, as long as no broadcast messages are involved (this is called a *binary protocol* [Huhns and Stephens, 1999]).

With agents knowing that they follow different interaction models and that semantic mismatches are likely to occur, communication requires to be processed at another level. For this reason, we define an *alignment protocol* that links agent interaction models. This protocol is seen as a meta-protocol through which the communication is carried out: any communication act regarding the lower level becomes ineffective and has an effective counterpart according to the meta-level. The alignment protocol (from here on AP) is depicted in Figure 4.5. Let us explain it in detail.



$$
\begin{array}{rcl}
\alpha & = & \langle \textit{inform}, (?X : \textit{algn}), (?Y : \textit{algn}), \texttt{final\_state}\rangle \\
\beta & = & \langle \textit{confirm}, (!Y : \textit{algn}), (!X : \textit{algn}), \texttt{final\_state}\rangle \\
\gamma & = & \langle \textit{deny}, (!Y : \textit{algn}), (!X : \textit{algn}), \texttt{final\_state}\rangle \\
\delta & = & \langle \textit{inform}, (?X : \textit{algn}), (?Y : \textit{algn}), \texttt{failure}\rangle
\end{array}
$$

Figure 4.5: The alignment protocol

There are four states: the initial state q^0, and intermediate state q^1, and two final states by name of letters s and u. These last ones are the initial letters of the words *successful* and *unsuccessful*. If the meta-level state s is reached, whatever path is followed, the object-level interaction is considered successful, otherwise it is considered unsuccessful. In this sense, we distinguish for the moment only between two kinds of interactions.

Regarding transitions, all of them are listed below the figure except one that has a special status. Notice that agents can adopt only one role, namely, the role of "aligner" or *algn* in short. There are two sorts of messages: `failure` and `final_state`. In addition, the former can be tagged with the illocutionary particle $inform$, and the latter with *inform*, *confirm* and *deny*.

The next illocution scheme establishes the link between the agent interaction models and the alignment protocol:

$$\langle \textit{utter}, (?X : \textit{algn}), (?Y : \textit{algn}), ?I\rangle \tag{4.3}$$

Let us have a look at its ingredients: X and Y are agent identifier variables, and I is an illocution variable. Therefore, (4.3) can be seen as a meta-illocution, since its content, in turn, is also an illocution. It can be grounded with illocutions of the form $\langle utter, (id_i : algn), (id_j : algn), \varphi\rangle$, where $\varphi = \langle \iota, (id_i : r), (id_j : r'), v\rangle\rangle$ is an illocution of IM_i. Note that the sender and receiver of φ must be equal to the instantiations of X and Y, respectively. Furthermore, let us stress that φ has to come from the interaction model associated with the instantiation of X. Consequently, the choice of "utter" as illocutionary particle seems natural. This performative expresses the sender attitude with respect to its own interaction model: if A_j receives $\langle utter, (id_i : algn), (id_j : algn), \varphi\rangle$, she can assume that A_i has decided to utter φ according to IM_i.

At this point the alignment protocol dynamics and matching mechanisms come into play.

Alignment Protocol Dynamics

Each agent is guided by both the alignment protocol and its own interaction model, whilst effective communication is done through the former as described above. When agents agree to initiate an interaction, both of them are in state q^0 wrt AP. In addition, agent A_i is in state q_i^0 wrt IM_i $(i = 1, 2)$.

Imagine that agent A_i is in state q_i, where q_i is an arbitrary element of Q_i. There can be several possibilities:

AP.1 A_i decides to utter $\varphi = \langle \iota, (id_i : r), (id_j : r'), v\rangle\rangle$ in the IM_i context, where $\varphi \in \delta_i(q_i, \cdot)$.[1] The communication act must be carried out via AP so agent A_i has to send the meta-illocution $\langle utter, (id_i : algn), (id_j : algn), \varphi\rangle$ to A_j. The state thereby remains the same in the AP context, whereas q_i turns to $q_i' = \delta_i(q_i, \varphi)$ in the IM_i context.

AP.2 A_i prompts a state change by a special transition $c_i \in C_i$ in the IM_i context. Thus q_i turns to $q_i' = \delta_i(q_i, c_i)$. This action is not reflected in AP since it does not entail any communication act.

AP.3 A_i receives $\langle utter, (id_j : algn), (id_i : algn), \varphi\rangle$ in the meta-level AP, where $\varphi = \langle \iota, (id_j : r), (id_i : r'), v\rangle\rangle$. Recall that from A_i's viewpoint v is a foreign message, and, for this reason, it is considered semantically different from all local ones.

Now, the **key issue** is that v is to be mapped with one of those messages that A_i expects to receive at state q_i in the IM_i context (Principle 1 stated in Section 4.4.1). Moreover, we can make a selection and just consider those messages encased in illocutions the head of which is equal to that of φ (Principle 2). In this way, A_i is to choose an element from the following set:

$$D = \{w \mid \langle \iota, (id_j : r), (id_i : r'), w\rangle \in dom(\delta_i(q_i, \cdot))\}$$

[1] $\delta_i(q_i, \cdot)$ is the function defined from $\Sigma_i = \Im_i \cup C_i$ to Q_i in the natural way.

There can be two possibilites: D is empty or not.

(a) As long as D is not empty, A_i can select an element w of D by making use of the **matching mechanism** explained further below. So q_i turns to $q'_i = \delta_i(q_i, \psi)$ where $\psi = \langle \iota, (id_j : r), (id_i : r'), w) \rangle$.

(b) In case D is empty then no mapping is possible. The interaction is considered unsuccessful. A_i is to send a failure message to A_j by uttering $\langle inform, (id_i : algn), (id_j : algn), \texttt{failure} \rangle$, which matches with the illocution scheme δ. Thus q^0 turns to u in the AP context.

AP.4 If q_i is a final state and A_i assumes the interaction to be finished, A_i can send the illocution $\langle inform, (id_i : algn), (id_j : algn), \texttt{final\_state} \rangle$ to A_j, which matches with the illocution scheme α. In this way, q^0 turns to q^1, and A_j is supposed to ground β or γ, either confirming or denying the completion of the interaction, respectively. Grounding β makes agents to reach the final state s, and the interaction is considered successful; γ, however, leads to an unsuccessful interaction.

AP.5 Finally, we have to take into account the possibility of a deadlock. This is the case when, for example, successive mappings have led the agents to states where both of them only await messages. In order to avoid deadlocks, the special transition *timeout* is linked to the initial state q^0 in AP. When a specific period of time is exceeded, this transition leads agents to finish the interaction, which is then considered unsuccessful.

4.6.2 The Matching Mechanism

As mentioned above, the matching mechanism is called whenever a message is received. In a nutshell, it is based on three assertions:

- every foreign message is associated with a categorical variable ranging over local messages; likewise a variable assignment represents a matching element.
- The matching mechanism computes frequency distributions of all these variables on the basis of past successful interactions.
- Agents' matching decisions are determined by virtue of these distributions.

Past Information: Histories and Frequency Distributions

Whenever an interaction is successfully performed, agents are to record relevant information that will be useful in future interactions. This information is revealed in terms of *histories* that gather all past matching decisions. These histories increasingly enlarge the population on which a statistical reasoning for forthcoming matching decisions will be based. In this section, we explain both statistical updating and matching decisions in detail.

Agents build histories while interacting with the alignment protocol. Specifically, a *history* is a sequence of the form:

$$h = q_i^0, \sigma_i^1, q_i^1, \ldots, q_i^{k-1}, \sigma_i^k, \ldots, q_i^{n-1}, \sigma_i^n, q_i^n$$

computed recursively as follows:

- q_i^0 is the initial state of IM_i, and
- if A_i is in case AP.1, then $[\varphi, q_i']$ is queued in h,
- if A_i is in case AP.2, then $[c_i, q_i']$ is queued in h,
- if A_i is in case AP.3.a, $[\langle \iota, (id_j : r), (id_i : r'), [v/w] \rangle, q_i']$ is queued in h,
- q_i^n is a final state of IM_i.

Notice that unsuccessful interactions are not considered. The problem with this is that it is not so easy to find out which particular matching was responsible of a failure, or if we should blame one agent or another for a wrong matching decision. We will come back to this issue when discussing the future work in Chapter 7.

In order to make the notation clearer, we will dispense with subscripts. So we have two agents A and B, identified with a and b, and associated with interaction models IM_A and IM_B.

Let $\mathcal{H} = \{h^k\}_{k=1}^n$ be the sequence of all past successful histories reported by agent A so far. Note that it may happen that $h^k = h^l$ for $1 \leq k, l \leq n$ and $k \neq l$. If this is the case, as far as agent A is concerned, there is no other distinction between h^k and h^l but time occurrence. Now, from all information contained in these histories, we will particularly pay attention to those pairs of the form:

$$p = \langle q, \langle \iota, (b : r), (a : r'), [v/w] \rangle \rangle \tag{4.4}$$

where $\langle \iota, (b : r), (a : r'), [v/w] \rangle$ comes straight after the state q in (at least) one history of $\mathcal{H}$. The reader should think of p as follows: at some point in the past and having received illocution $\langle \iota, (b : r), (a : r'), v \rangle$ at state q, agent A decided to match message v with the local message w.

Forthcoming matching decisions will be based on successful past matching decisions, represented by pairs as p in (4.4). From here on, we will refer to these pairs with the abbreviation *pmd* (past matching decision), or *pmd on v* if we want to specify the matched message.

Assume that agent A received v in the past. Let us consider the multiset (or bag) $\mathcal{P}_v$ of all pmd on v that appear in $\mathcal{H}$ (indeed, there may be more than one occurrence of the same pmd). $\mathcal{P}_v = \langle P_v, \pi_v \rangle$ where P_v is the underlying set of elements and $\pi_v : P_v \to \mathbb{N}$ is the multiplicity function. For the task at hand, message v will be treated as a (qualitative) statistical variable $V : P_v \to M_A$, where M_A is the set of A's local messages and V is defined in the natural way.

If v turned out to be matched with $w \in M_A$ (in other words, w is a member of the range of V), the frequency associated with w is:

$$F(V = w) = \frac{\sum_{V(p)=w} \pi_v(p)}{\sum \pi_v(p)} \in [0,1]$$

where summations range by default over $p \in P_v$. But there are other attributes of the elements of $\mathcal{P}_v$ that are worth studying. If $H : P_v \to H$ is the *head* variable defined in the natural way,

$$F(V = w | H = h) = \frac{\sum_{V(p)=w, H(p)=h} \pi_v(p)}{\sum_{H(p)=h} \pi_v(p)}$$

An analogous formula holds for $F(V = w | Q = q, H = h)$, where $Q : P_v \to Q$ is what we can call the *state* variable.

We will use the symbol $\mathcal{F}_v$ when referring to this frequency distribution. In this way, if $\mathcal{H} = \{h^k\}_{k=1}^n$ is the resulting history recording of $n \geq 1$ interactions with agent B, $\mathcal{H}$ generates a family of frequency distributions $\mathcal{F} = \{\mathcal{F}_v\}_{v \in \Omega_n}$, where Ω_n is the set of all B's messages received by A so far. At whatever time a new interaction is successfully completed, frequency distributions have to be updated.

Matching Decisions

In this part of the chapter we will explain the reasoning followed by an agent when facing a matching decision. So the backdrop is an agent A, identified with a having received an illocution $\varphi = \langle \iota, (b : r'), (a : r), v_0 \rangle$ from another agent B identified with b during a specific interaction. Thus we take up again the story-line that we momentarily left when explaining case AP.3.a.

In principle, A could match the received message v_0 with any $w \in D$, where D, written with the new notation for agent identifiers, is:

$$D = \{w \mid \langle \iota, (b : r'), (a : r), w \rangle \in dom(\delta_A(q, \cdot))\}$$

but this reasoning can be refined. Let us distinguish between two cases: agent A has information about successful past interactions with B that involved v_0 or not.

If agent A has information about former successful interactions, this will become available in terms of frequency distributions, $\mathcal{F} = \{\mathcal{F}_v\}$, as we have already explained above. If it so happens that $\mathcal{F}_{v_0} \in \mathcal{F}$ then A can benefit from this information when making a matching decision on v_0. Let us study this case and we will come back later to the case of no information.

One first idea is to choose a local message w_0 such that

$$F(V_0 = w) \leq F(V_0 = w_0)$$

for all $w \in D$, in other words, to choose w_0 if no other matching element $[v_0/w]$ was more successful in the past. But we also want to follow the I-SSA principles in this regard. We should take into account in which state $q \in Q_A$ the message is actually received and its illocutionary force, exhibited by the head h of the illocution φ, specifically, $h = \langle \iota, (a : r), (b : r') \rangle$. Thus, the second idea is to "truncate" the previous frequency values so as to choose $w_0 \in M_A$ with

$$F(V_0 = w | Q = q, H = h) \leq F(V_0 = w_0 | Q = q, H = h)$$

for all $w \in D$. However, this state condition seems to be very restrictive. It is true that I-SSA makes the semantic alignment conditional on the states, but also that the same message may be received in many different states, and we want to exploit this. A more sensible option is to replace the event $\{Q = q\}$ with $\{V_0 \in D\}$ in the frequency values above.

Now, in the case of no information, all D messages are equiprobable. Since IM_A is a finite automaton, D is a finite set, and, if $|D| = n < \infty$, we can write $D = \{w_1, \ldots, w_n\}$. Agent A can thereby choose $w_0 \in D$ with probability $p = \frac{1}{n}$. This leads us to our first matching criterion:

First Matching Criterion (maximal frequency criterion)

if $\mathcal{F}_{v_0} \in \mathcal{F}$ **then**
choose $w_0 \in M_A$ such that
$F(V_0 = w | V_0 \in D, H = h) \leq F(V_0 = w_0 | V_0 \in D, H = h)$ for all $w \in M_A$
else
choose $w_0 \in D$ with probability $p = \frac{1}{n}$
end if

The maximal frequency criterion highly depends on how rich the frequency distributions are. If there is not much information about former interactions, it makes sense not to fully rely on a matching element with maximal frequency. Hence another matching criterion is required. For this reason, we consider the probability distribution $\{(w_i, p_i)\}_{i=1}^{n}$, where $p_i = F(V_0 = w_i | V_0 \in D, H = h)$, and we "contaminate" it with a uniform distribution. The contamination parameter $s \in (0, 1)$ is usually a real number close to 1.

Second Matching Criterion (contaminated probability criterion)

Require: $s \in (0, 1)$, $s \approx 1$
if $\mathcal{F}_{v_0} \in \mathcal{F}$ **then**
choose $w_0 \in M_A$ with probability
$$p = s \cdot F(V_0 = w | V_0 \in D, H = h) + (1 - s) \cdot \frac{1}{n}$$
else
choose $w_0 \in D$ with probability $p = \frac{1}{n}$
end if

As an alternative to both criteria, we present an intermediate solution. The following criterion corresponds with the previous one when $s = 1$.

Third Matching Criterion (probability criterion)

if $\mathcal{F}_{v_0} \in \mathcal{F}$ **then**
  choose $w_0 \in M_A$ with probability $p = F(V_0 = w_0 | V_0 \in D, H = h)$
else
  choose $w_0 \in D$ with probability $p = \frac{1}{n}$
end if

We argue for a combination of the three preceding criteria in practice. As we have already explained above, the first criterion seems to be the best option when agents have rich information (statistical population) of previous interactions. However, the second matching criterion should be followed until this happens. The question of when to change the criteria is not an easy problem as it also depends on each particular domain. In Chapter 5 we present experiments that combine the matching criteria presented in this section, but a deeper analysis has been left for future work.

4.6.3 Semantic Alignment through the Matching Mechanism

The matching mechanism described above helps agents to interact successfully. Note that agent messages are related as more interactions are completed. In what follows, we pin down these semantic relationships in a logical fashion, and we finally compare them with the ones deduced from the communication product as stated in Definition 8.

Let us assume that agent A_i $(i = 1, 2)$ has generated a family $\mathcal{F} = \{\mathcal{F}_v\}_{v \in \Omega}$ of frequency distributions using the matching mechanism. Let $v \in \Omega$ and V its associated statistical variable. If $F(V = w^k) \neq 0$ for $w^k \in M_i$ and $1 \leq k \leq n$, whereas $F(V = w') = 0$ for $w' \in M_i$ and $w' \neq w^k$ for $1 \leq k \leq n$, then:

$$\langle j, v \rangle \sqsubseteq \langle i, w^1 \rangle \sqcup \ldots \sqcup \langle i, w^n \rangle$$

where $j = 1, 2$ and $j \neq i$. The idea behind this subsumption is that, regarding $\mathcal{F}_v$, only the matching elements $[v/w^1], \ldots, [v/w^n]$ triggered states transitions making the interaction eventually reach a final state.

In contrast with Definition 8, this time it is also possible to discriminate between disjunction members. If $F(V = w) = t \in (0, 1]$, it holds:

$$\langle j, v \rangle \sqsubseteq \langle i, w \rangle [t]$$

The real number t expresses the *confidence degree* of the matching element $[v/w]$. Finally, the *semantic alignment* is made up of the set of all these expressions.

Once at this point the natural step is to compare the semantic alignment computed via the matching mechanism with the semantic alignment deduced from the communication product. One expects that the former is “bounded” by the latter, so everything that agents can obtain is limited by the communication product. This is further formalised in Theorem 4.2, but before that a definition needs to be introduced.

Definition 10 *Let* IM_1 *and* IM_2 *be two interaction models and* $\mathrm{IM} = \mathrm{IM}_1 \otimes \mathrm{IM}_2$*. Given an* IM*-history* h *written as in Definition 4, the* projection *of* h *onto* IM_1 *or* IM_1-projection *of* h *is the finite sequence:*

$$\pi_1(h) = q_1^0, \sigma_1^1, q_1^1, \ldots, q_1^{k-1}, \sigma_1^k, \ldots, q_1^{n-1}, \sigma_1^n, q_1^n$$

where q_1^k *is the first component of* q^k $(k = 1, \ldots, n)$*, and*

- $\sigma_1^k = \langle \iota, (id_1 : r), (id_2 : r'), m \rangle$ *if* $\sigma^k = \langle \iota, (id_1 : r), (id_2 : r'), \langle m, m' \rangle \rangle$*,*
- $\sigma_1^k = \langle \iota, (id_2 : r), (id_1 : r'), [m'/m] \rangle$ *if* $\sigma^k = \langle \iota, (id_2 : r), (id_1 : r'), \langle m, m' \rangle \rangle$*,*
- $\sigma_1^k = c_1$ *if* $\sigma^k = \langle c_1, c_2 \rangle$*.*

The projection *of* h *onto* IM_2 *or* IM_2-projection *of* h *is analogously defined.*

Theorem 4.2 *If* $\langle j, v \rangle \sqsubseteq \langle i, w^1 \rangle \sqcup \ldots \sqcup \langle i, w^n \rangle$ *is a subsumption computed using the matching mechanism, and* $\langle j, v \rangle \sqsubseteq \langle i, m^1 \rangle \sqcup \ldots \sqcup \langle i, m^r \rangle$ *is a subsumption deduced from the communication product, then* $n \leq r$ *and there exist indices* $s_1, \ldots, s_n$ *with* $1 \leq s_k \leq r$ *such that* $w^k = m^{s_k}$ *for* $k = 1, \ldots, n$*.*

Proof. We simply give a sketch of the proof. If h_i is a history built by agent A_i using the matching mechanism then there exists an IM-history h such that $h_i = \pi_i(h)$. The result is thus a direct application of the definitions. Q.E.D.

Example *With the help of the matching mechanism, the dealer agent is able to deduce the following relationships:*

$$\begin{aligned}
\langle 2, \texttt{no_more_cards} \rangle &\sqsubseteq \langle 1, \texttt{surrender} \rangle\, [r_1] \\
\langle 2, \texttt{no_more_cards} \rangle &\sqsubseteq \langle 1, \texttt{insurance} \rangle\, [r_2] \\
\langle 2, \texttt{no_more_cards} \rangle &\sqsubseteq \langle 1, \texttt{stand} \rangle\, [r_3] \\
\langle 2, \texttt{stick} \rangle &\sqsubseteq \langle 1, \texttt{double_down} \rangle\, [s_1] \\
\langle 2, \texttt{stick} \rangle &\sqsubseteq \langle 1, \texttt{hit} \rangle\, [s_2] \\
\langle 2, \texttt{draw_another_card} \rangle &\sqsubseteq \langle 1, \texttt{double_down} \rangle\, [t_1] \\
\langle 2, \texttt{draw_another_card} \rangle &\sqsubseteq \langle 1, \texttt{hit} \rangle\, [t_2]
\end{aligned}$$

where $r_1 + r_2 + r_3 = 1$*,* $s_1 + s_2 = 1$ *and* $t_1 + t_2 = 1$*.*

Similarly, the player agent can deduce the following ones:

$$\begin{aligned}
\langle 1, \texttt{new_hand} \rangle &\sqsubseteq \langle 2, \texttt{wager} \rangle\, [1.0] \\
\langle 1, \texttt{walk_away} \rangle &\sqsubseteq \langle 2, \texttt{finish} \rangle\, [1.0] \\
\langle 1, \texttt{card} \rangle &\sqsubseteq \langle 2, \texttt{card} \rangle\, [1.0] \\
\langle 1, \texttt{dealer_card} \rangle &\sqsubseteq \langle id_2, \texttt{face_up_card} \rangle\, [1.0] \\
\langle 1, \texttt{blackjack} \rangle &\sqsubseteq \langle id_2, \texttt{natural} \rangle\, [1.0] \\
\langle 1, \texttt{bust} \rangle &\sqsubseteq \langle id_2, \texttt{too_many} \rangle\, [1.0] \\
\langle 1, \texttt{lose} \rangle &\sqsubseteq \langle id_2, \texttt{debt} \rangle\, [1.0] \\
\langle 1, \texttt{push} \rangle &\sqsubseteq \langle id_2, \texttt{profit} \rangle\, [1.0] \\
\langle 1, \texttt{win} \rangle &\sqsubseteq \langle id_2, \texttt{profit} \rangle\, [1.0]
\end{aligned}$$

4.7 I-SSA as a Particular Case of SSA

In Chapter 3 we described an alignment process by which two agents establish the semantic relationship among terms of their respective vocabularies based on the assumption that mismatching terms describe a partial perspective of a shared physical environment state, a state that is not accessible (*i.e.*, completely and faithfully perceived) to any of the two agents. As agents go through more and more environment states, the semantic alignment between their vocabularies is further and further refined.

In the scenario illustrated in this chapter agents do not share a physical environment, but they share the same interaction. Hence their "environment" is captured by the communication product that represents the entire space of actual interactions, but which is not accessible to agents in general. An uttered illocution, though, provides a "description" of the interaction state, because its utterance "means" that it was allowed in the current interaction state according to the partial perspective of the uttering agent. An agent receiving the illocution can now establish a semantic alignment based on the assumption that both agents where sharing the same interaction state.

More formally, given two interaction models IM_1 and IM_2, an *interaction state* can be seen as a pair $e = \langle h, k\rangle$, where h is a history of the communication product $\mathrm{IM}_1 \otimes \mathrm{IM}_2$ and $1 \leq k \leq n$, where n is the length of h. What agent A_i $(i = 1, 2)$ perceives from e is $p_i = see_i(h, k) = \langle h_i, k_i\rangle$, where $h_i = \pi_i(h)$ is the projection of h onto IM_i (as defined in Definition 10) and k_i is the position in h_i of the projection of $h[k]$ (note that it may happen that $k \neq k_i$).[2] Now, agents' types are agents' messages. Thus $\langle h_i, k_i\rangle \models_i m$ if $h_i[k_i]$ is an illocution whose content is m. This leads us to a channel $\mathcal{E}$ whose distributed logic $Log(\mathcal{E})$ is what we have agreed on referring to as the logic of SSA. The key point is that this logic comprises the I-SSA semantic alignment. Imagine indeed that the following expression belongs to the latter:

$$\langle 1, m\rangle \sqsubseteq \langle 2, m^1\rangle \sqcup \ldots \sqcup \langle 2, m^n\rangle$$

Let $e = \langle h, k\rangle$ be an arbitrary interaction state. Let us assume that $see_1(e) = \langle h_1, k_1\rangle \models_1 m$. Therefore, m is the content of illocution $h_1[k_1]$. Since h_1 is the projection of h onto IM_1, the content of $h[k]$ must be equal to $\langle m, m'\rangle$ for some $m' \in M_2$. But the above expression belongs to the I-SSA semantic alignment, so $m' = m^k$ for some $k = 1, \ldots, n$. The content of $h[k]$ is then equal to $\langle m, m^k\rangle$. If h_2 is the projection of h onto IM_2 and k_2 is the position in h_2 of the projection of $h[k]$, we can ensure that $see_2(e) = \langle h_2, k_2\rangle \models_2 m^k$. This proves that:

$$\langle 1, m\rangle \vdash \langle 2, m^1\rangle, \ldots, \langle 2, m^n\rangle$$

So we end up with the following:

Theorem 4.3 I-SSA *is a particular case of* SSA. *More specifically, each* I-SSA *subsumption is a constraint of a* SSA *logic.*

[2] $h[k]$ denotes the k-th element of the sequence h.

In Chapter 3, the SSA channel was denoted by $\mathcal{E}$, while $\mathcal{C}^*$ represented the theoretical end of a communication process where all environment states are observed. Note that, in this case, since the environment is represented by the communication itself, channel $\mathcal{E}$ matches $\mathcal{C}^*$. The communication process is much more refined that the one described in Chapter 3, where a simple exchanging of types is performed. Here the agent communication is ruled by the combination of the alignment protocol and mechanism.

4.8 Summary and Concluding Remarks

In this chapter we have laid the formal foundations for a novel approach to tackle the problem of semantic heterogeneity in multi-agent communication. We look at the semantics of messages from an interaction-based point of view, as it arises in the context of interaction models. Messages are deemed semantically related if they trigger compatible interaction state transitions, where compatibility here means that the interaction progresses in the same direction for each agent, albeit their interaction views (that is, their interaction models) may be more constrained than the interaction that is actually happening.

One advantage of this approach is that it takes into account meaning that is very interaction-specific and which cannot be derived from sources that are external to the interaction. In this sense we see it as a complement to current state-of-the-art matching techniques as it may provide valuable information for pruning the search space or disambiguating the results of candidate semantic alignments computed with today's ontology-matching technology.

Other approaches share with ours the insight that semantics is interaction-specific. In [Besana and Robertson, 2007] the authors opt to attach probabilities to meanings of terms that are determined by earlier, similar interactions, and these probabilities are used to predict the set of possible meanings of a message. Meaning is also defined relative to a particular interaction, but the authors aim at reducing the search space of possible a priori mappings (in a classical sense), namely by assessing those ones with highest probability in the context of an interaction.

In [Rovatsos, 2007] a dynamic semantics for agent communication languages (ACLs) is proposed. With the same spirit, Rovatsos bases his notion of dynamic semantics on the idea of defining alternatives for the meaning of individual speech acts in an ACL semantics specification, and transition rules between semantic states (collections of variants for different speech acts) that describe the current meaning of the ACL. One of our initial premises leads to an ACL to be shared by all agents. We believe that to agree on a pre-defined ACL is not a big assumption that can significantly help to solve the semantic heterogeneity brought by the existence of different ontologies.

In line with the previous work, Bravo and Velázquez present an approach for discovering pragmatic similarity relations among agent interaction protocols [Bravo and Velázquez, 2008]. Besides the objection already explained above, the authors do not take into account state histories when measuring their notion of pragmatic similarity, but separate state transitions. This certainly leaves out relations among messages that may be crucial in certain scenarios.

In the following chapter we prove the viability of our approach with experiments and their evaluation.

Chapter 5

I-SSA Implementation and Experimentation

Abstract. In this chapter we describe an implementation and experiments with the I-SSA approach. The evaluation of this experimentation proves the viability of this technique. We also provide a thorough statistical study consisting of combinations of variance analyses and Tukey tests. Part of the content of this chapter has been published in [Atencia and Schorlemmer, 2009].

5.1 Introduction

Chapter 4 contains the I-SSA formalisation, as well as an approximation to put this approach into practice, namely, by means of an alignment protocol and an alignment mechanism. These facets are here complemented with the analysis of experimental results.

We set out to answer two Research Questions:

RQ.1 Is there a gain in communication accuracy —measured in the number of successful interactions, that is, interactions reaching a final state— by repeated semantic alignment through a meta-level alignment protocol and the use of a matching mechanism?

RQ.2 If so, how many repeated interactions between two agents are needed in order to get sufficiently good alignments —measured in the probability of a successful interaction?

The experiment design opens the chapter, followed by a presentation of its execution and evaluation. A thorough statistical analysis completes its.

5.2 Experiment Design

In this section the experiment design is explained. The alignment protocol and all matching mechanisms are implemented in SICStus Prolog Release 4.0.7 [Carlsson, 2009] and random operations are executed with the SICStus Prolog random library.

In our simulations only two agents are considered. This assumption is not very restrictive, since, as stated in Chapter 4, it is always possible to split an interaction among several agents into several interactions between two agents, provided that no broadcast messages are involved.

To overcome the lack of sufficiently complex examples on which to run our implementation and experiments, we have used the FSA utilities toolbox [van Noord, 1996] as follows. First, an abstract alphabet made up of arbitrary illocutions and special transitions is generated. Second, a regular expression is built upon this alphabet and prefixed numbers of Kleene star, concatenation and alternation operators. Finally, the regular expression is compiled into an automaton that is not necessarily minimal using the FSA library (Figure 5.1).

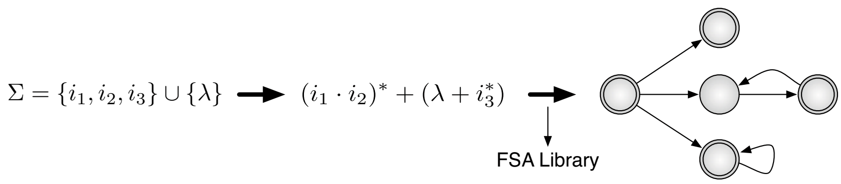


Figure 5.1: Process of generating abstract interaction models

Table 5.1 shows all variables considered in this process and the range of values they may take (as usual, $\mathbb{N}$ stands for the set of all positive integers, while $\mathbb{N}_0$ stands for the set of non-negative integers, *i.e.*, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$).

Name	**Variable**	**Range**
Number of illocutions	N_{ill}	$\mathbb{N}$
Number of illocutionary particles	N_{ip}	$\mathbb{N}$
Number of roles	N_{role}	$\mathbb{N}$
Number of messages	N_{msg}	$\mathbb{N}$
Number of special transitions	N_{spt}	$\mathbb{N}_0$
Number of Kleene star operators	N_{star}	$\mathbb{N}_0$
Number of concatenation operators	N_{con}	$\mathbb{N}_0$
Number of alternation operators	N_{alt}	$\mathbb{N}_0$

Table 5.1: Relevant variables when generating abstract interaction models

So, before all else, a variable grounding is required. In practice, the ranges of these variables are bounded. Firstly, one expects N_{ip} not to be much greater than 15. Although the sum of shared performatives may add up to three or four times this number, we take for granted that no interaction model makes use of all of them. Secondly, a sensible upper bound for N_{role} is 15, and our experience within the project OpenKnowledge has confirmed this (see, for example, [Marchese et al., 2008] where an eResponse interaction model with no more than 10 roles is defined). Though ontologies vary in size from a few hundred classes to tens of thousands of classes, and the number of instances varies from hundreds to hundreds of thousands, these amounts reduce when limited to appear in specific interactions. For this reason, interaction models with more than 100 different messages are not considered.

Notice that all variables examined so far are illocution components. It is straightforward to prove that the number of illocutions N_{ill} has a lower bound:

$$N_{ill} \geq \max\left\{N_{ip}, N_{msg}, \left[\frac{N_{role}}{2}\right] + 1\right\} \tag{5.1}$$

Indeed we should have more illocutions than illocutionary particles, otherwise some of them would be discarded, ditto messages and roles (remember that each illocution involves two roles, namely, the sender and receiver roles).

Regarding the operators of regular expressions, we first need to explain how regular expressions are constructed. It is easy to check that an expression of n binary operators —concatenations and alternations— has no more than $n + 1$ distinct alphabet symbols (notice that the number of Kleene star operators is not relevant here). In our case, these symbols may be either illocutions or special transitions. If n_{con} and n_{alt} are the number of concatenation and alternation operators included in a regular expression r, respectively, then there exist $n_{con} + n_{alt} + 1$ placeholders in r to be filled with alphabet symbols. These placeholders can be seen as leaves in a tree representation of the regular expression (see Figure 5.3). In our implementation, operators are randomly chosen, and placeholders are randomly filled with either illocutions or special transitions.

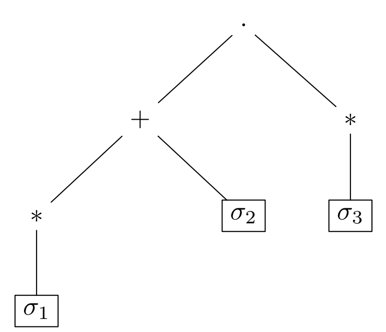


Figure 5.2: Tree representation of $r = (\sigma_1^* + \sigma_2) \cdot \sigma_3^*$

Let us represent by N_{leaf} the number of leaves of a regular expression built with our implementation. Since $N_{leaf} = N_{con} + N_{alt} + 1$, we have:

$$N_{ill} + N_{spt} \leq N_{con} + N_{alt} + 1 \tag{5.2}$$

while putting together (5.1) and (5.4):

$$\max\left\{N_{ip}, N_{msg}, \left\lceil\frac{N_{role}}{2}\right\rceil + 1\right\} \leq N_{ill} + N_{spt} \leq N_{con} + N_{alt} + 1 \tag{5.3}$$

Consequently, N_{ill} has both a lower and upper bound (as long as we give a specific value for N_{spt}). For instance, if $N_{ip} = 2$, $N_{role} = 3$ and $N_{msg} = 30$, and $N_{con} = 20$ and $N_{alt} = 25$, then $30 \leq N_{ill} + N_{spt} \leq 46$. In case $N_{spt} = 1$, N_{ill} can take any value from 29 to 45.

To conclude this section, let us say that the number of special transitions is no likely to be greater than 5. Also, experience within the OpenKnowledge project has shown that the complexity of an interaction model does not go over the complexity entailed by a few hundreds of operators. Table 5.3 summarises the variable boundaries explained so far (clearly, all shown intervals are integer intervals). Notice that N_{ill} does not appear this time.

Name	**Variable**	**Range**
Number of illocutionary particles	N_{ip}	$[1, 15]$
Number of roles	N_{role}	$[1, 15]$
Number of messages	N_{msg}	$[1, 100]$
Number of special transitions	N_{spt}	$[0, 5]$
Number of Kleene star operators	N_{star}	$[0, 100]$
Number of concatenation operators	N_{con}	$[0, 100]$
Number of alternation operators	N_{alt}	$[0, 100]$

Table 5.2: Ranges of relevant variables when generating interaction models

Regarding the operators of regular expressions, we first need to explain how regular expressions are constructed. It is easy to check that an expression of n binary operators —concatenations and alternations— has no more than $n + 1$ distinct alphabet symbols (notice that the number of Kleene star operators is not relevant here). In our case, these symbols may be either illocutions or special transitions. If n_{con} and n_{alt} are the number of concatenation and alternation operators included in a regular expression r, respectively, then there exist $n_{con} + n_{alt} + 1$ placeholders in r to be filled with alphabet symbols. These placeholders can be seen as leaves in a tree representation of the regular expression (see Figure 5.3). In our implementation, operators are randomly chosen, and placeholders are randomly filled with either illocutions or special transitions.

Let us represent by N_{leaf} the number of leaves of a regular expression built with our implementation. Since $N_{leaf} = N_{con} + N_{alt} + 1$, we have:

$$N_{ill} + N_{spt} \leq N_{con} + N_{alt} + 1 \tag{5.4}$$

while putting together (5.1) and (5.4):

$$\max\left\{N_{ip}, N_{msg}, \left\lceil\frac{N_{role}}{2}\right\rceil + 1\right\} \leq N_{ill} + N_{spt} \leq N_{con} + N_{alt} + 1 \tag{5.5}$$

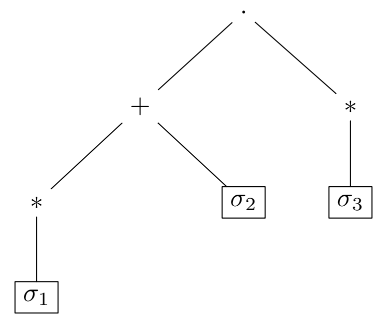


Figure 5.3: Tree representation of $r = (\sigma_1^* + \sigma_2) \cdot \sigma_3^*$

Consequently, N_{ill} has both a lower and upper bound (as long as we give a specific value for N_{spt}). For instance, if $N_{ip} = 2$, $N_{role} = 3$ and $N_{msg} = 30$, and $N_{con} = 20$ and $N_{alt} = 25$, then $30 \leq N_{ill} + N_{spt} \leq 46$. In case $N_{spt} = 1$, N_{ill} can take any value from 29 to 45.

To conclude this section, let us say that the number of special transitions is no likely to be greater than 5. Also, experience within the OpenKnowledge project has shown that the complexity of an interaction model does not go over the complexity entailed by a few hundreds of operators. Table 5.3 summarises the variable boundaries explained so far (clearly, all shown intervals are integer intervals). Notice that N_{ill} does not appear this time.

Name	**Variable**	**Range**
Number of illocutionary particles	N_{ip}	$[1, 15]$
Number of roles	N_{role}	$[1, 15]$
Number of messages	N_{msg}	$[1, 100]$
Number of special transitions	N_{spt}	$[0, 5]$
Number of Kleene star operators	N_{star}	$[0, 100]$
Number of concatenation operators	N_{con}	$[0, 100]$
Number of alternation operators	N_{alt}	$[0, 100]$

Table 5.3: Ranges of relevant variables when generating interaction models

5.3 Execution and Evaluation

Remember that in our model agents consider all foreign messages semantically different *a priori*, even when they match syntactically any local ones. This fact justifies our decision to let agents follow the same interaction model, since agents will deal the situation as if they conform to disparate models.

In total three experiments were performed. In the first one, we simulated two agents interacting through the meta-level alignment protocol and taking advantage of the matching mechanism (Experiment 1). Matching choices were based on the second matching criterion (see Chapter 4), in other words, on the contaminated probability criterion with contamination factor equal to 0.1. In the second experiment, agents made use of the meta-level protocol but no updating was ever carried out (Experiment 2). Now, some series of interactions were simulated. Specifically, we ran both implementations in series of $N = 2^n$ interactions, where $n = 1, 2, \ldots, 12$ (thus we let agents interact at most 4096 times). Each batch of interactions was performed 50 times recording the average of failures $F(N)$ (or F). In order to compare both experiments we computed the ratio of failures to interactions, that is, $R = \frac{F}{N}$.

Experiments 1 and 2 were performed on the basis of five interaction models of varying complexity. Table 5.4 shows their corresponding parameter choices.

	N_{ill}	N_{ip}	N_{role}	N_{msg}	N_{spt}	N_{star}	N_{con}	N_{alt}
imodel1	15	1	1	5	0	2	10	15
imodel2	20	1	2	10	0	5	15	10
imodel3	30	2	3	15	2	10	20	25
imodel4	50	1	1	40	0	15	30	25
imodel5	100	4	5	80	2	20	50	80

Table 5.4: Generated interaction models

Figure 5.4 presents the results of experimenting with *imodel4*. Now, it is straightforward to check that when using the alignment mechanism the number of failures decreases considerably, while the alignment protocol alone yields a high and almost constant number of failures. Similar results were obtained with the rest of interaction models (Figure 5.5). This supports a positive answer to the research question RQ.1. We will come back to this experiment in Section 5.4.

In the third experiment (Experiment 3), we first let agents interact as in Experiment 1 to compute an alignment, again in series of $N = 2^n$ interactions, where $n = 1, 2, \ldots, 12$. This alignment was then used by the agents to interact 50 times, with no frequency updating and matching decisions based on the first matching criterion (maximal frequency criterion). We recorded this time the ratio of successes to interactions, *i.e.*, $R = \frac{S}{50}$. Figure 5.6 shows the results with the five interaction models. In all cases R approaches 1. Actually, no more than 256 interactions are needed to achieve an alignment that ensures a probability close to 0.8 to interact successfully. This answers research question RQ.2.

5.4 Statistical Analysis

When looking at the experiment results presented before, some natural questions come up: do all parameters have an influence on the final result?, and, if so, what is their influence? Which values do better for the parameters?

In order to be in position to give answers to the previous questions, we have realised a factorial execution of Experiment 1 along with a statistical analysis

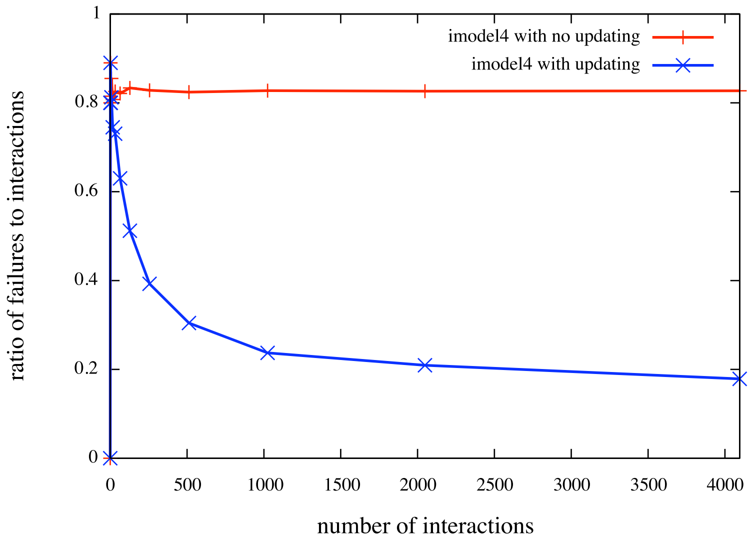


Figure 5.4: Experiments 1 and 2 with *imodel4*

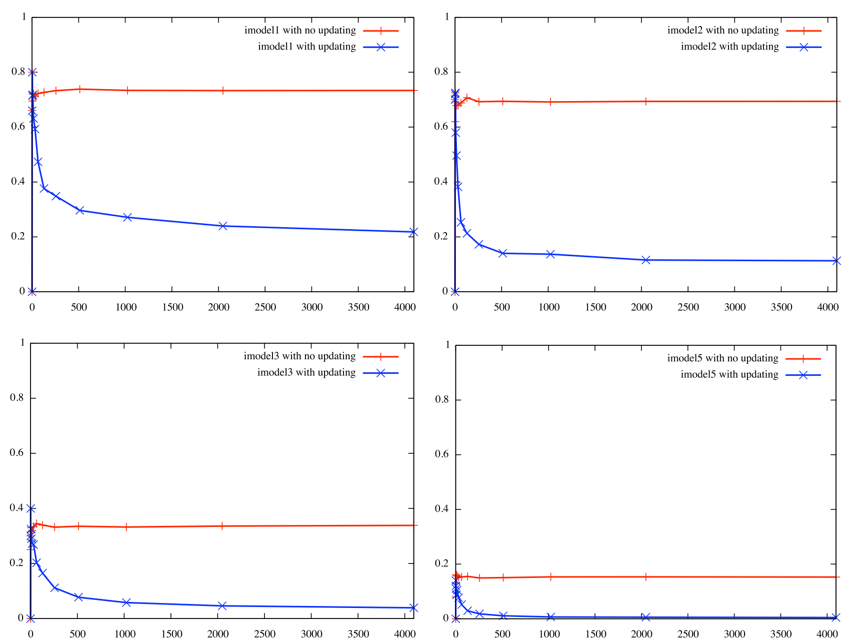


Figure 5.5: Experiment results with the rest of interaction models

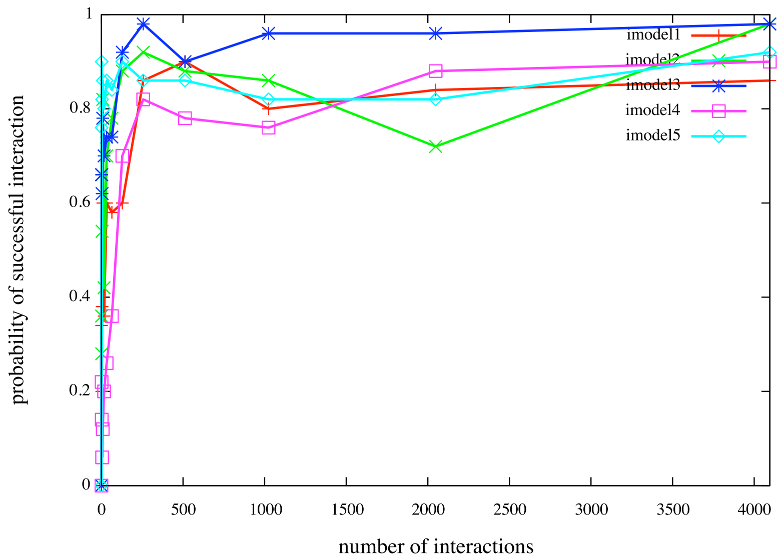


Figure 5.6: Experiment 3 with all generated interaction models

of the resulting experimental data, combining an analysis of variance (ANOVA) with post-hoc comparisons using the so-called Tukey test [Cohen, 1995]. The former helped us elucidate whether there was or not a significant relationship between the independent variables —the parameters in the simulation— and the dependent variable —ratio of failures to interactions. The latter verified which values did better for each of the independent variables. But before presenting the statistical results, let us explain how we proceeded in this experimentation.

For ANOVA test results to be reliable, a number of preconditions must be satisfied. One refers to independence in the sample. This led us to modify our input variables, since restrictions (5.1) and (5.4) explained in Section 5.2 obviously violate the required independence. One possible first step is to discard the number of special transitions N_{spt}. This is not a great loss, since we are more interested in studying the effect of the illocution ingredients and the structure of the interaction model. In this way, $N_{leaf} = N_{ill}$, so that an ANOVA test is run for each specific value of N_{ill}. Specifically, the following have been selected:

$$N_{ill} = 8, 16, 32, 64, 128$$

The number of alternation operators was not considered, since, as already seen, $N_{alt} = N_{ill} - N_{con} - 1$, so any statement about N_{con} has a counterpart statement about N_{alt}. We also replaced variables N_{ip} and N_{role} with a unifying variable N_{head} which accounts for the number of illocution heads.

Once a particular value of N_{ill} is chosen, an upper bound for N_{head} and N_{msg} is laid down: $1 \leq N_{head}, N_{msg} \leq N_{ill}$. However, since an interaction model in which there are no repeated illocution heads is not interesting for the task at hand (agents would always be able to distinguish the correct message between all incoming ones, and, hence, they would not fail at all), $\frac{1}{2}N_{ill}$ is a much more effective upper bound for N_{head}. Interaction models with only one message were not taken into account either. Concerning operators, we have decided to generate regular expressions with at least one operator of each type. In this way, $1 \leq N_{con} \leq N_{ill} - 2$. Kleene star operators, though, were left to be in any case lower than $\frac{1}{2}N_{ill}$. Series of interactions were again powers of two, but this time limited to 512.

Name	**Selected Values**
Number of heads	1,2,4,8,16
Number of messages	2,4,8,16,32
Number of Kleene star operators	1,2,4,8,16
Number of concatenation operators	1,2,4,8,16,30
Number of interactions	1,2,4,8,16,32,64,128,256,512

Table 5.5: Simulation variables for $N_{ill} = 32$

In this chapter, only the case $N_{ill} = 32$ will be examined, but the rest of cases are similar. Table 5.5 registers all selected values chosen for $N_{ill} = 32$. Before executing the ANOVA tests we verified that the resulting data had a normal distribution through a Quantile-Quantile test, which is another precondition for the results to be reliable.

We ran an ANOVA test with the software environment R [Dalgaard, 2008]. The analysis of variance does demonstrate that all the independent variables were statistically significant with a p-value much lower than 0.05, which is the standard threshold for statistical significance tests. This value is shown in Figure 5.7 among other information. Degrees of freedom (Df) indicate the number of quantities that must be estimated to define the effect of the variable. The sum of squared error (Sum Sq) is a measure of how much of the variance in the response is explained by the variable. Mean Sq is just the result of dividing Sum Sq by Df. The F value relates the Sum Sq for a variable to the total amount of variation of response (incorporating the Df information).

```
                  Df  Sum Sq Mean Sq F value    Pr(>F)
Heads              4  13.625   3.406 150.403 < 2.2e-16 ***
Messages           4   3.943   0.986  43.524 < 2.2e-16 ***
Stars              4   3.269   0.817  36.081 < 2.2e-16 ***
Concatenations     5  48.277   9.655 426.333 < 2.2e-16 ***
Interactions       9   9.579   1.064  46.994 < 2.2e-16 ***
Residuals       7472 169.224   0.023
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 5.7: ANOVA results for $N_{ill} = 32$

Therefore, the ANOVA test verified, or, better said, *did not refute*:

Hypothesis 1 *The following factors affect the number of failures:*

1. *the variety of illocution heads,*
2. *the amount of local messages,*
3. *the structure of interaction models, and*
4. *the number of interactions.*

But one also expects a hypothesis of this kind to be confirmed:

Hypothesis 2 *The following imply a lower number of failures:*

1. *a higher number of illocution heads,*
2. *a lower number of messages, and*
3. *a higher number of interactions.*

For this reason, we ran post-hoc comparisons using the Tukey test, which did not refute Hypothesis 2. Figures 5.8, 5.9 and 5.10 include the results for the number of illocution heads, messages and interactions, respectively. These results are both graphically and numerically represented. In the tables, 'diff' stands for the difference in the observed means, 'lwr' represents the lower end point of the interval, 'upr' gives the upper end point and 'p adj' the p-value after adjustment for the multiple comparisons.

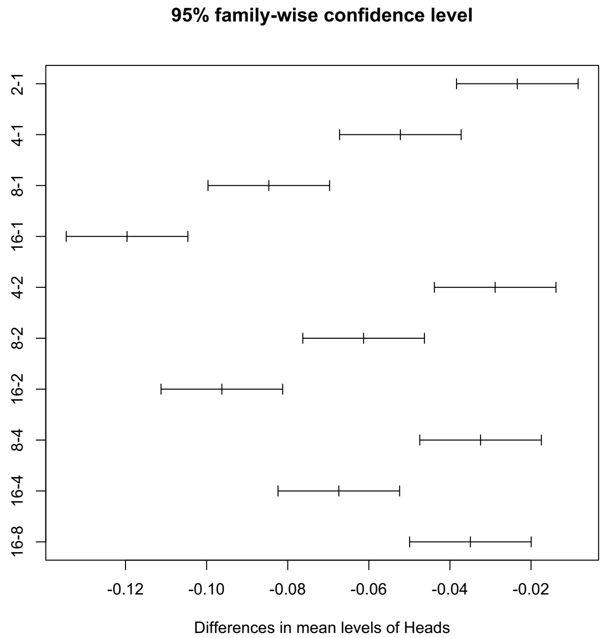
But, what about the operators? Well, concerning concatenation operators (and, thereby, alternation operators), the following can be confirmed (Figure 5.11):

Hypothesis 3 *A lower number of concatenations implies a lower number of failures. Alternatively, a higher number of alternations implies a lower number of failures.*

This hypothesis deserves some comments. Remember that our approach highly depends on the criterion followed when it comes to classify an interaction as successful or unsuccessful (see I-SSA third principle in Chapter 4). In this dissertation, an interaction is dubbed successful as long as agents jointly reach a final state. In general, the more alternation operators a regular expression has, the more paths leading to final states in the interaction model, and, thus, the more chances for the agents to interact successfully. Nevertheless, different notions of success —as a consequence of, for example, more expressiveness in the interaction models (see the concluding remarks of Chapter 4)— would result in a different hypothesis.

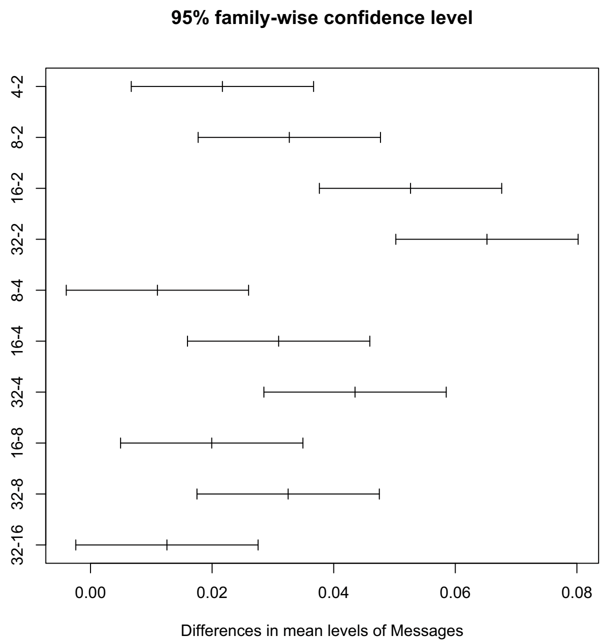
Regarding Kleene star operators, the Tukey test provided some confusing results. Neither a higher nor a lower number of star operators imply a lower number of failures. This fact made us think that this may not be the proper parameter to be studied. Another option is to look into the complexity of an interaction model measured, for instance, with the concept of *star-height*

[Eggan, 1963, Hopcroft and Ullman, 1979]. Actually, the three operators that we have considered in this work are involved in this new parameter. However, a preliminary experimentation suggested us that this is not the right way either. It seems to be necessary to work on a specific notion of complexity appropriate for the case at hand. Unfortunately, this task has not been addressed in this dissertation and has been left for future work.




```
$Heads
             diff         lwr          upr     p adj
2-1   -0.02337885 -0.03837214 -0.008385572 0.0002058
4-1   -0.05222260 -0.06721589 -0.037229322 0.0000000
8-1   -0.08466807 -0.09966386 -0.069672290 0.0000000
16-1  -0.11962161 -0.13461490 -0.104628332 0.0000000
4-2   -0.02884375 -0.04383703 -0.013850468 0.0000016
8-2   -0.06128922 -0.07628500 -0.046293436 0.0000000
16-2  -0.09624276 -0.11123604 -0.081249478 0.0000000
8-4   -0.03244547 -0.04744125 -0.017449686 0.0000000
16-4  -0.06739901 -0.08239229 -0.052405728 0.0000000
16-8  -0.03495354 -0.04994932 -0.019957759 0.0000000
```

Figure 5.8: Tukey's test results for parameter Heads



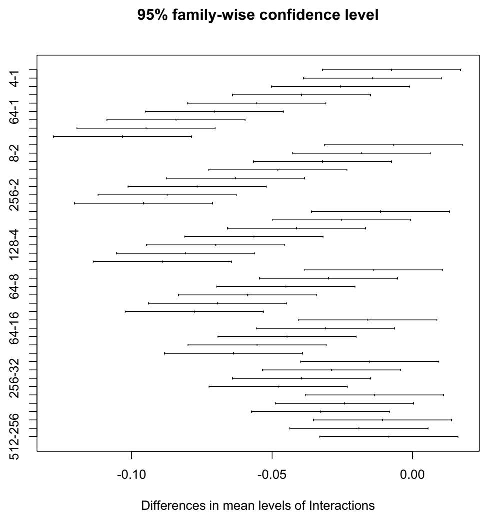
$Messages

	diff	lwr	upr	p adj
4-2	0.02169549	0.006702212	0.03668878	0.0007573
8-2	0.03269964	0.017706353	0.04769292	0.0000000
16-2	0.05263656	0.037643280	0.06762984	0.0000000
32-2	0.06520561	0.050209830	0.08020139	0.0000000
8-4	0.01100414	-0.003989142	0.02599742	0.2647751
16-4	0.03094107	0.015947785	0.04593435	0.0000002
32-4	0.04351012	0.028514335	0.05850590	0.0000000
16-8	0.01993693	0.004943645	0.03493021	0.0026615
32-8	0.03250598	0.017510194	0.04750176	0.0000000
32-16	0.01256905	-0.002426733	0.02756483	0.1491461

Figure 5.9: Tukey's test results for parameter Messages

5.5 Summary and Concluding Remarks

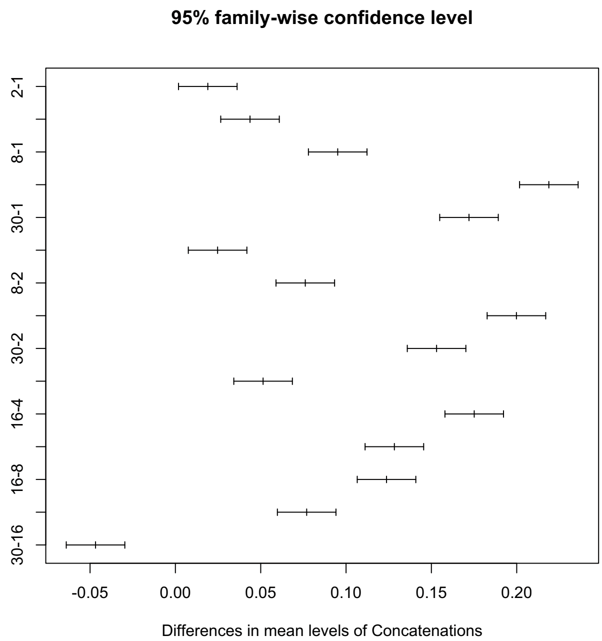
This chapter includes some experimental results around the I-SSA approach explained from a theoretical angle in Chapter 4. In a first experimentation, it is shown how the number of unsuccessful agent interactions decreases with the use of the I-SSA technique, as well as an estimation of the probability of success, on the basis of a family of abstract interaction models. A second experimentation allows us to perform a thorough statistical analysis from which some remarkable



$Interactions

	diff	lwr	upr	p adj
2-1	-0.007506667	-0.03210028	0.0170869448	0.9940556
4-1	-0.014160000	-0.03875361	0.0104336115	0.7212705
8-1	-0.025546667	-0.05014028	-0.0009530552	0.0342098
16-1	-0.039540000	-0.06413361	-0.0149463885	0.0000164
32-1	-0.055429167	-0.08002278	-0.0308355552	0.0000000
64-1	-0.070613750	-0.09520736	-0.0460201385	0.0000000
128-1	-0.084219375	-0.10881299	-0.0596257635	0.0000000
256-1	-0.094892221	-0.11948583	-0.0702986094	0.0000000
512-1	-0.103288045	-0.12788986	-0.0786862259	0.0000000
4-2	-0.006653333	-0.03124694	0.0179402781	0.9976201
8-2	-0.018040000	-0.04263361	0.0065536115	0.3750204
16-2	-0.032033333	-0.05662694	-0.0074397219	0.0015588
32-2	-0.047922500	-0.07251611	-0.0233288885	0.0000000
64-2	-0.063107083	-0.08770069	-0.0385134719	0.0000000
128-2	-0.076712708	-0.10130632	-0.0521190969	0.0000000
256-2	-0.087385554	-0.11197917	-0.0627919427	0.0000000
512-2	-0.095781378	-0.12038320	-0.0711795592	0.0000000
8-4	-0.011386667	-0.03598028	0.0132069448	0.9058037
16-4	-0.025380000	-0.04997361	-0.0007863885	0.0366108
32-4	-0.041269167	-0.06586278	-0.0166755552	0.0000050
64-4	-0.056453750	-0.08104736	-0.0318601385	0.0000000
128-4	-0.070059375	-0.09465299	-0.0454657635	0.0000000
256-4	-0.080732221	-0.10532583	-0.0561386094	0.0000000
512-4	-0.089128045	-0.11372986	-0.0645262259	0.0000000
16-8	-0.013993333	-0.03858694	0.0106002781	0.7350656
32-8	-0.029882500	-0.05447611	-0.0052888885	0.0047640
64-8	-0.045067083	-0.06966069	-0.0204734719	0.0000003
128-8	-0.058672708	-0.08326632	-0.0340790969	0.0000000
256-8	-0.069345554	-0.09393917	-0.0447519427	0.0000000
512-8	-0.077741378	-0.10234320	-0.0531395592	0.0000000
32-16	-0.015889167	-0.04048278	0.0087044448	0.5671861
64-16	-0.031073750	-0.05566736	-0.0064801385	0.0025941
128-16	-0.044679375	-0.06927299	-0.0200857635	0.0000004
256-16	-0.055352221	-0.07994583	-0.0307586094	0.0000000
512-16	-0.063748045	-0.08834986	-0.0391462259	0.0000000
64-32	-0.015184583	-0.03977819	0.0094090281	0.6316870
128-32	-0.028790208	-0.05338382	-0.0041965969	0.0081179
256-32	-0.039463054	-0.06405667	-0.0148694427	0.0000172
512-32	-0.047858878	-0.07246070	-0.0232570592	0.0000000
128-64	-0.013605625	-0.03819924	0.0109879865	0.7660639
256-64	-0.024278471	-0.04887208	0.0003151406	0.0564299
512-64	-0.032674295	-0.05727611	-0.0080724759	0.0011053
256-128	-0.010672846	-0.03526646	0.0139207656	0.9354277
512-128	-0.019068670	-0.04367049	0.0055331491	0.2946187
512-256	-0.008395824	-0.03299764	0.0162059950	0.9866600

Figure 5.10: Tukey's test results for parameter Interactions



$Concatenations

	diff	lwr	upr	p adj
2-1	0.01899503	0.001836279	0.03615378	0.0200042
4-1	0.04372453	0.026565779	0.06088328	0.0000000
8-1	0.09510998	0.077947795	0.11227217	0.0000000
16-1	0.21882363	0.201664873	0.23598238	0.0000000
30-1	0.17203550	0.154876748	0.18919425	0.0000000
4-2	0.02472950	0.007570748	0.04188825	0.0005753
8-2	0.07611495	0.058952764	0.09327714	0.0000000
16-2	0.19982859	0.182669842	0.21698735	0.0000000
30-2	0.15304047	0.135881717	0.17019922	0.0000000
8-4	0.05138545	0.034223264	0.06854764	0.0000000
16-4	0.17509909	0.157940342	0.19225785	0.0000000
30-4	0.12831097	0.111152217	0.14546972	0.0000000
16-8	0.12371364	0.106551457	0.14087583	0.0000000
30-8	0.07692552	0.059763332	0.09408770	0.0000000
30-16	-0.04678813	-0.063946877	-0.02962937	0.0000000

Figure 5.11: Tukey's test results for parameter Concatenations

conclusions are drawn. We can confirm that the variation in illocution heads, the amount of local messages, the structure of interaction models, and the number of interactions have an influence on the number of failures. Furthermore, we verified that a higher number of illocution heads implies a lower number of failures, and so a lower number of messages and more interactions do.

No comparisons with other alignment methods or matching techniques are shown, though. This aspect is undertaken in the following chapter, where a case study is presented.

Chapter 6

A Case Study: Travel Reservation

Abstract. In this chapter we compare the I-SSA approach with a number of state-of-the-art ontology matching techniques on the basis of an example about travel reservation. Specifically, we contrast the semantic alignments produced by all these techniques, and show how the I-SSA results fit better in this particular scenario. Although a comparison over a richer library of examples is required, we believe that this simple study already highlights the potential of I-SSA.

6.1 Introduction

Blackjack was chosen in Chapter 4 as a didactical running example with which to explain the I-SSA technique and to show its potential for semantic alignment. Later in Chapter 5 the viability of this technique was exhibited through a thorough abstract experimentation. Nevertheless, we still lack an illustration by means of a realistic scenario that gives us the possibility to put I-SSA in the context of current state-of-the-art matching techniques.

In this chapter, we specifically study the case of a customer who contacts an online travel agency to make a reservation. This is a typical Web service use case, but we will look at it from the perspective of multiagent systems, and under the assumption that participants may make use of different terminologies. We compare the I-SSA technique with certain matching techniques on the basis of this travel reservation scenario. The results show that I-SSA is better suited for semantic alignment where the context of interaction is very relevant. Although a deeper study is required for a comparison to be fully reliable, we believe that this example is good enough to convince the reader of the potential of I-SSA.

The chapter is structured as follows. Firstly, we explain the usage of Semantic Web technologies in the travel and tourism industry by surveying a number of approaches. Secondly, the travel reservation scenario is presented. Thereupon two interaction models are provided, one for the customer and one for the travel agent, and two ontologies on which the interaction models are based are given. Thirdly, the I-SSA approach is applied both from a theoretical angle, in terms of the communication product, and from a practical point of view, letting agents interact through a meta-level alignment protocol and make use of the matching mechanism. Finally, we show how state-of-the-art techniques would deal with the semantic heterogeneity for this particular example, and we compare it with the proposed I-SSA solution.

6.2 Online Travel Reservation

The Web has evidently transformed many human day-to-day activities, and this is definitely true for travel reservation. Few people hesitate these days to surf on transportation carrier Web sites and online travel portals when looking for a flight, hotel or vacation package, instead of going to the travel agency in person. The Web has certainly made this task much easier, but online travel reservation, as one of so many other B2C (business-to-consumer) e-commerce applications, suffers from a number of limitations.

Online travel portals, all provided with travel search engines, are claimed to be the best alternative when trying to make the most appropriate travel reservation. Ideally, a user would only need to use one of these portals, but the situation is rather different. Every portal has its special characteristics (tariffs, expenses) and its search is conditioned to a group of companies. Most users end up consulting manually a considerable number of Web sites, collecting information (prices, availability or reservation terms) by themselves, and in many cases taking a premature final decision.

The Semantic Web project envisions a scenario where human users delegate this kind of tedious task to software agents. Human users would only have to feed their personal agents with required information for making the most suitable reservation regarding their preferences. But, of course, this requires a comprehensive markup effort, adding metadata processable by machines and thereby capturing the meaning of data. Metadata will be used to identify and extract information from Web sources, and ontologies will be used to assist in Web searches, to interpret retrieved information, and to communicate with other agents. Logic will be employed for processing retrieved information and for drawing conclusions [van Harmelen and Antoniou, 2008]. The tourism community has kept up with these new trends.

Tourism, e-Commerce and the Semantic Web

Since the 60s the travel and tourism industry has been an active field of IT and network applications —computerised reservation systems (CRS) or global distribution systems (GDS) such as AMADEUS, SABRE, Worldspan or Galileo [Werthner and Klein, 1999]. This dependency on IT is increasing due to the emergence of online marketing and information services on the Internet. Whereas other industries are displaying a stronger hold on to traditional processes, the tourism industry is witnessing an acceptance of e-commerce technologies to the extent that the entire industry structure is changing [Werthner and Ricci, 2004]. These technologies vary from techniques for information extraction, integration and presentation, to mobile applications and recommender systems, and the Semantic Web vision provides a unifying view of all of them.

The travel and tourism industry can particularly benefit from the Semantic Web due to the significant heterogeneity of the market and information sources, and the high volume in online transactions. Contrariwise, the industry also provides a challenging test bed for Semantic Web services and agent technologies. The Travel Agent Game in Agentcities (TAGA) [TAGA, URL] is an agent framework for simulating a global travel market in the Agentcities environment. TAGA is used for demonstrating Agentcities and Semantic Web technologies. All agents are FIPA compliant and FIPA interaction protocols rule agent communications. Besides the FIPA OWL content language, several OWL ontologies are specified (a travel business ontology or an auction ontology, among others).

Harmonise [Dell'Erba et al., 2002, Fodor and Werthner, 2005] is an attempt at ontology-mediated integration of tourism systems. It proposes a platform that allows distinct tourism organisations to keep their proprietary data format and to exchange information based on such an ontology by means of a mediator module. This tool is RDF-based for what concerns information representation and exchange, and ontologies are used for knowledge and content management. It is claimed that the next generation of e-tourism will be powered by Semantic Web technologies (resulting in an e-tourism Semantic Web portal which will connect customers and virtual travel agents from anywhere at anytime). One of the main milestones of Harmonise was to put together an enlarged consortium. This consortium involved major tourism standard organisations like OTA (the so-called Open Travel Alliance) [OTA, URL]. OTA has been working on defining a common language for travel-related terminology, along with a mechanism to enable the exchange of information among travel industry members.

Other tourism domain ontologies have been designed. Mondeca, a company specialised in taxonomy management, semantic representation of knowledge and business repositories, devised for one of their clients a thesaurus inspired by the World Tourism Organisation (WTO) [Mondeca, URL]. On a small scale, other travel ontologies can be found in the DAML language portal (for instance, the Itinerary-ont or the Trip report ontology) [DAML, URL].

In [Vukmirovic et al., 2007] an agent-based system for selling airline tickets is presented. The system is a combination of a model agent-based e-commerce system with an agent-based travel support system. For this merging to be fully accomplished, the authors realised that an air travel ontology was to be designed. They have started to develop this ontology on the basis of IATA (International Air Transport Association) manuals and OTA messaging system. Once it is completed, it should be capable of being used to interface the travel support system with an actual GDS. This is not entirely novel, since AMADEUS is a GDS that supports communication using OTA messaging. It aims at lowering cost of operations and it offers OTA messaging as a way of distributing airline inventory to external travel sites and dynamic package providers.

All approaches presented so far tackle the problem of semantic heterogeneity by defining either standards or ontologies. But the tourism domain has come up against the problem of ontology heterogeneity too, as this new form of tourism is of a clear open, dynamic and highly distributed nature. OTA's efforts for standardising may work for travel industry members, but global standardisation at the level of the Semantic Web, where agents are claimed to act on their own for the benefit of users, seems, if not impossible, arduous to be accomplished. For this reason, ontology matching has been proposed as an alternative solution.

In what follows, we present what we think is a realistic scenario in the domain of e-tourism. We will apply the I-SSA technique in this scenario and compare its result with the results produced by a number of state-of-the-art ontology matching techniques.

6.3 The Travel Reservation Scenario

Imagine a travel agency that offers to its customers facilities to either book flights, accommodations or vacations. Consider also that this travel agency is up to date with the new technologies so that it has delegated to a software agent the task of making reservations. This software agent is thus programmed for interacting with customers, whether they are human or software agents, in order to satisfy all their requests. Here we will particularly study the scenario where two software agents, one as travel agent and the other as customer, participate in a travel reservation interaction.

6.3.1 Interaction Models and Ontologies for Travel Reservation

The notion of interaction model has been proposed in this work as an abstract model for the specification of agent interactions. Since they are based on the concept of finite state machine, interaction models are general enough to capture a wide range of applications. In this particular scenario, the customer and the travel agent will also interact according to an interaction model.

Assume, however, that these agents follow different interaction models, and that messages to be exchanged while interacting belong to dissimilar ontologies. Specifically, agent identified with a will play the role of customer in interaction model IM_1, which is divided into sections in Figures 6.2, 6.3, 6.4 and 6.5, whereas agent identified with b will play the role of travel agent in interaction model IM_2, depicted into parts in Figures 6.7, 6.8, 6.9 and 6.10. There exist significant differences between messages of IM_1 and IM_2. Actually, these two interaction models conform to two distinct ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ (a formal specification using the Manchester OWL syntax is given in Appendix B). In what follows, a common description of both the interaction models and ontologies is provided.

Let us start by studying interaction model IM_1 and its relation with ontology $\mathcal{O}_1$. At a first glance, IM_1 seems to be structurally divided into three, let us say, "sub-interaction models". Now, looking at it more carefully, it is easy to realise that these sub-interaction models indeed represent three types of interactions: a flight reservation, an accommodation reservation and a vacation reservation. This fact is reflected in ontology $\mathcal{O}_1$, since there exists an upper class Reservation associated with a class Item by means of the property hasItem, and the class Item is equivalent to the disjunction of three subclasses: Flight, Accommodation and Vacation (see Figure 6.1). Note also that these subclasses are pairwise disjoint. But terms in IM_1 they are also wrapped up in illocutions.

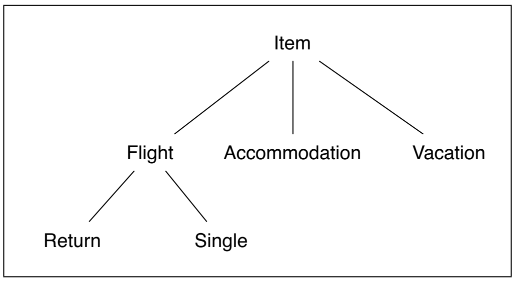


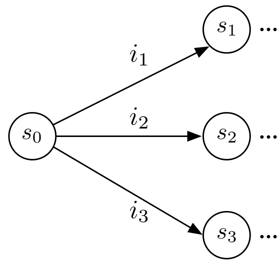
Figure 6.1: Sketch of the customer's travel ontology

There exist a number of illocutionary particles that meaningfully enrich those messages to be exchanged while interacting according to IM_1: *request*, *inform*, *commit*, *suggest*, *accept* and *reject*. All agents are supposed to agree on the semantics of these performatives (see Section 4.4.3 of Chapter 4). Wierzbicka's book [Wierzbicka, 1989] is a remarkable effort to define a semantic dictionary of English speech act verbs. The author tries to overcome a fundamental flaw of all traditional dictionaries: their circularity. Wierzbicka's dictionary contains definitions (explanations) of around 250 speech act verbs, written in first person perspective and on the basis of a controlled (both minimal and standardised) vocabulary. This vocabulary is made up of 150 words (and, in a sense, not much more than 50 words) which recur in different combinations ('I', 'you', 'want', 'know', 'do') and with only one undefined speech act verb ('say'). As an example, let us take the case of *request* and show a piece of its definition:

Meaning of *request*:

I say: I want Y to happen.
I know that Y cannot happen if someone (X) doesn't do something to cause it to happen.
I say this because I want to cause X to cause Y to happen
I don't want to say that X has to do it.
I assume that X will understand that I have a reason to say that I want Y to happen.
I assume that X will cause Y to happen.

The travel reservation interaction models have been designed in a way that the involved illocutionary particles conform to Wierzbicka's semantics.



$i_1 = \langle \mathit{request}, (?A : \mathit{customer}), (?B : \mathit{travel\_agent}), \texttt{flight}\rangle$
$i_2 = \langle \mathit{request}, (?A : \mathit{customer}), (?B : \mathit{travel\_agent}), \texttt{accommodation}\rangle$
$i_3 = \langle \mathit{request}, (?A : \mathit{customer}), (?B : \mathit{travel\_agent}), \texttt{vacation}\rangle$

Figure 6.2: Interaction model for the customer agent (part I).

At the starting state s_0 (initial state), the customer is supposed to send a message to the travel agent requesting either a flight (through illocution i_1), an accommodation (illocution i_2) or a vacation (i_3). Each choice triggers a distinct interaction. Assume that the customer asks the travel agent for making a flight reservation, or, in our terms, that the customer sends the following illocution:

$$i_1 = \langle \mathit{request}, (a : \mathit{customer}), (b : \mathit{travel\_agent}), \texttt{flight}\rangle$$

The choice of *request* as illocutionary particle seems very suitable regarding the definition presented above.

A flight trip may be a return trip or a single trip, and this is something the customer must be specific about firstly (through sending illocutions i_4 and i_5, respectively). Once this is done, the customer is supposed to send some required information about the desired flight, but this information certainly depends on the previous choice. In fact, if the customer agent asks for a single flight, it only needs to provide information about the departure, but if it asks for a return flight then it needs to give information about the return too. This required information varies: origin and destination (illocutions i_6 and i_7), dates (i_8 and i_{10}), times (i_9 and i_{11}), and number of passengers (i_{12}). In all of these illocutions the customer is "informing" the travel agent. Let us have a look to Wierzbicka's definition of *inform*:

Meaning of ***inform***:

I assume that you want to know things about X.
I know something about X that I think you should know.
I assume I should cause you to know it
I say: (...).
I say this because I want to cause you to know it I assume that you will understand that this is not something that could be untrue.
I assume that I will cause you to know it by saying this.

In ontological terms (see the customer's travel ontology in Appendix B), the travel agent is to build a *concept description* of the flight that the customer is looking for with all this information; something like:

$$\begin{aligned}\mathsf{Flight} \quad \sqcap \quad & \mathsf{Return} \sqcap \\ & \mathsf{origin} : \mathrm{Barcelona} \sqcap \\ & \mathsf{destination} : \mathrm{Malaga} \sqcap \\ & \mathsf{departing} : \text{2009-12-23} \sqcap \\ & \mathsf{outboundTime} : \text{18:00} \sqcap \\ & \mathsf{returning} : \text{2010-01-05} \sqcap \\ & \mathsf{inboundTime} : \text{20:00} \sqcap \\ & \mathsf{numberOfPassengers} : 2\end{aligned}$$

(as long as variable X in i_6 is grounded with 'Barcelona', so does Y in i_7 with 'Malaga'...). But this requires a clarification. There is no need for the ontology language and the interaction model content language to be exactly the same. In this particular example, the former is a description logic language, but the latter is a first-order language (remember that the I-SSA implementation is done in Prolog). Nevertheless, there must exist a one-to-one correspondence between predicate names and ontology entities (concept or role names). This translation should also be semantics-preserving. For the task at hand, we can skip the formal details and just assume that this translation bridges the two languages, so that the ontology is used for retrieving information (reservations) and the interaction model content language for the communication itself. However, we shall provide a DL counterpart of all communication messages for the sake of homogeneity of presentation. The key point is that we will be able to compare the semantic alignment produced by any ontology matching method with the I-SSA solution.

Once the above concept description is built, the travel agent is supposed to collect all instances that satisfy it with the help of the ontology. Note that $\mathcal{O}_1$ contains the role carrier which does not appear in IM_1 (this interaction model does not consider this facet significant for the search). All search results must agree with the customer requirements, but they may differ over the operating carrier. Let $[a_1, \ldots, a_n]$ be the search results written as a list. We can think of a_k as a URI that identifies a flight. The travel agent is to inform the customer by sending the following:

$$i_{13} = \langle \mathit{inform}, (b : \mathit{travel\_agent}), (a : \mathit{customer}), \mathtt{results}([a_1, \ldots, a_n]) \rangle$$

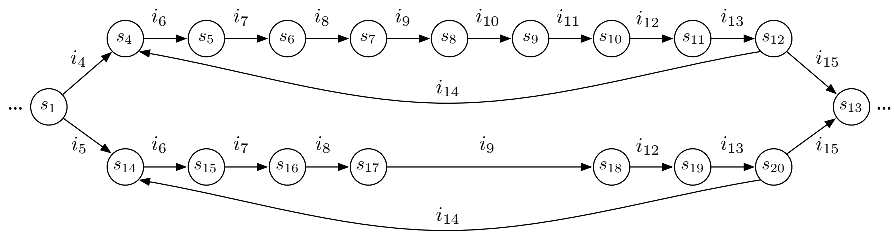
Actually, this can be seen as a logical axiom:

$$\mathsf{Result} \equiv \{a_1, \ldots, a_n\}$$

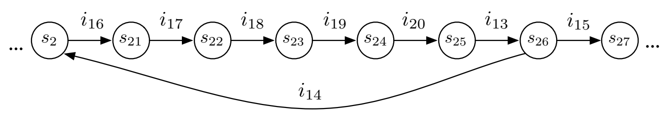
where $a_1, \ldots, a_n$ are individuals that represent flights.

Now, one of these flights may be to the customer's liking. If so, the customer is supposed to send:

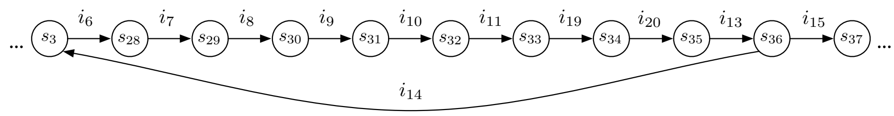
$$i_{15} = \langle \mathit{inform}, (a : \mathit{customer}), (b : \mathit{travel\_agent}), \mathtt{choice}(a_k) \rangle$$



(a) Selecting a flight



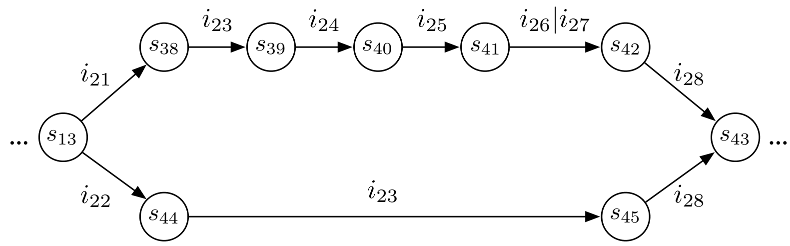
(b) Selecting an accommodation



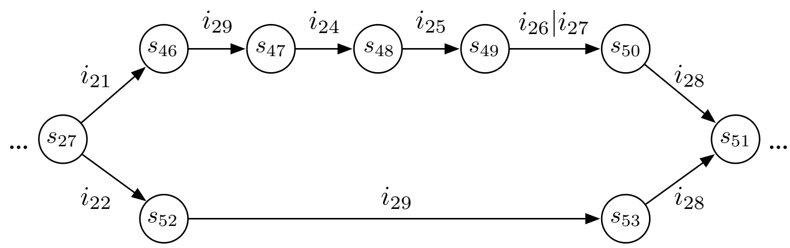
(c) Selecting a vacation

$$
\begin{aligned}
i_4 &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{return} \rangle \\
i_5 &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{single} \rangle \\
i_6 &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{origin}(?X) \rangle \\
i_7 &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{destination}(?Y) \rangle \\
i_8 &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{departing}(?V) \rangle \\
i_9 &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{outboundTime}(?OT) \rangle \\
i_{10} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{returning}(?W) \rangle \\
i_{11} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{inboundTime}(?IT) \rangle \\
i_{12} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{numberOfPassengers}(?NP) \rangle \\
i_{13} &= \langle \mathit{inform}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{results}(?R) \rangle \\
i_{14} &= \langle \mathit{request}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{search} \rangle \\
i_{15} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{choice}(?C) \rangle \\
i_{16} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{hotelBookingsIn}(?X) \rangle \\
i_{17} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{signIn}(?V) \rangle \\
i_{18} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{signOut}(?W) \rangle \\
i_{19} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{numberOfRooms}(?NR) \rangle \\
i_{20} &= \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{numberOfGuests}(?NG) \rangle
\end{aligned}
$$

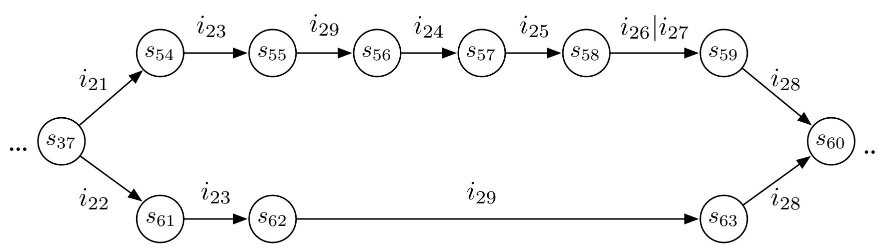
Figure 6.3: Interaction model for the customer agent (part II)



(a) Flight info



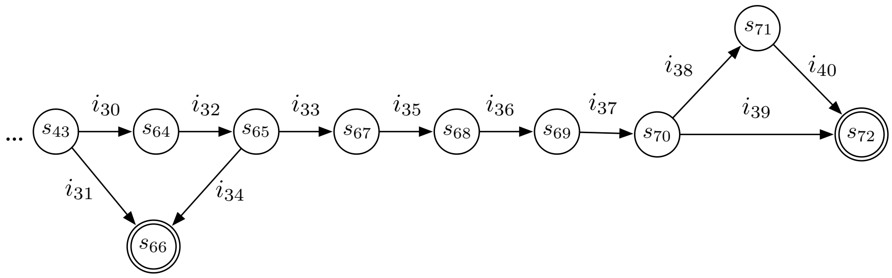
(b) Accommodation info



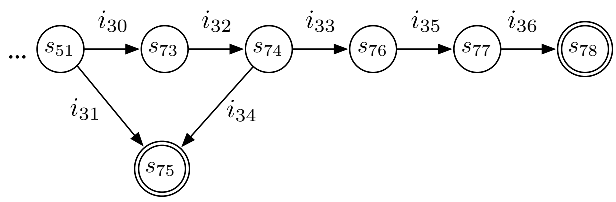
(c) Vacation info

$i_{21} = \langle \textit{inform}, (!A : \textit{customer}), (!B : \textit{travel\_agent}), \texttt{unregisteredCustomer}\rangle$
$i_{22} = \langle \textit{inform}, (!A : \textit{customer}), (!B : \textit{travel\_agent}), \texttt{registeredCustomer}\rangle$
$i_{23} = \langle \textit{inform}, (!A : \textit{customer}), (!B : \textit{travel\_agent}), \texttt{passengerDetails}(?PD)\rangle$
$i_{24} = \langle \textit{inform}, (!A : \textit{customer}), (!B : \textit{travel\_agent}), \texttt{customerDetails}(?CD)\rangle$
$i_{25} = \langle \textit{suggest}, (!B : \textit{travel\_agent}), (!A : \textit{customer}), \texttt{registration}\rangle$
$i_{26} = \langle \textit{accept}, (!A : \textit{customer}), (!B : \textit{travel\_agent}), \texttt{registration}\rangle$
$i_{27} = \langle \textit{reject}, (!A : \textit{customer}), (!B : \textit{travel\_agent}), \texttt{registration}\rangle$
$i_{28} = \langle \textit{inform}, (!B : \textit{travel\_agent}), (!A : \textit{customer}), \texttt{reservationTerms}(?RT)\rangle$
$i_{29} = \langle \textit{inform}, (!A : \textit{customer}), (!B : \textit{travel\_agent}), \texttt{guestDetails}(?GD)\rangle$

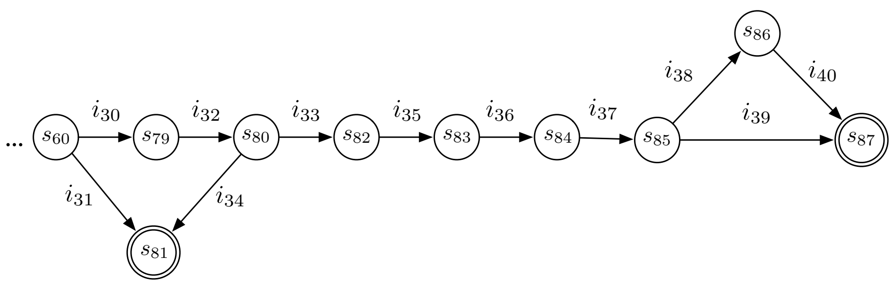
Figure 6.4: Interaction model for the customer agent (part III)



(a) Flight purchase



(b) Accommodation purchase



(c) Vacation purchase

$i_{30} = \langle accept, (!A : customer), (!B : travel\_agent), \texttt{reservationTerms}(!RT)\rangle$
$i_{31} = \langle reject, (!A : customer), (!B : travel\_agent), \texttt{reservationTerms}(!RT)\rangle$
$i_{32} = \langle inform, (!B : travel\_agent), (!A : customer), \texttt{totalAmountToPay}(?TA)\rangle$
$i_{33} = \langle commit, (!A : customer), (!B : travel\_agent), \texttt{totalAmountToPay}(!TA)\rangle$
$i_{34} = \langle reject, (!A : customer), (!B : travel\_agent), \texttt{totalAmountToPay}(!TA)\rangle$
$i_{35} = \langle inform, (!A : customer), (!B : travel\_agent), \texttt{paymentInfo}(?PI)\rangle$
$i_{36} = \langle inform, (!B : travel\_agent), (!A : customer), \texttt{reservationSummary}(?RS)\rangle$
$i_{37} = \langle suggest, (!B : travel\_agent), (!A : customer), \texttt{checkIn}\rangle$
$i_{38} = \langle accept, (!A : customer), (!B : travel\_agent), \texttt{checkIn}\rangle$
$i_{39} = \langle reject, (!A : customer), (!B : travel\_agent), \texttt{checkIn}\rangle$
$i_{40} = \langle inform, (!B : travel\_agent), (!A : customer), \texttt{boardingPass}(?BP)\rangle$

Figure 6.5: Interaction model for the customer agent (part IV)

where a_k is the chosen flight, and this is reflected in the ontology as an equality of individuals:

$$\mathsf{choice} = a_k$$

If not, the following illocution restarts the search:

$$i_{14} = \langle \mathit{request}, (a : \mathit{customer}), (b : \mathit{travel\_agent}), \mathtt{search} \rangle$$

which corresponds this time to the fact that the desired flight is an instance of the class Search.

Let us assume that the customer has chosen a_k. At this point, some other information is required for the reservation to be completed, or, in ontological terms, for building a complete description of the reservation that the customer is looking for. Now, the amount of information depends on whether the customer is registered or not (illocutions i_{21} and i_{22}). If the customer is already registered then only the passenger details are required (i_{23}), otherwise the customer details (i_{24}) are required too. In case that the customer is not registered, a registration is suggested (i_{25}) which may be accepted (i_{26}) or rejected (i_{27}).

The travel agent is to inform the customer of the reservation terms (i_{28}), and nothing prevents the customer from rejecting them (i_{31}). This option would lead both agents to reach a final state, so that the interaction would be finished. If the customer, though, accepts these reservation terms (i_{30}), then she will be informed of the total amount to pay for the flight (i_{32}). Again the customer may accept this price and continue the interaction (i_{33}), or reject it and reach a final state (i_{34}). Once the customer is committed to pay this price, the payment info is required (i_{35}). After the reservation summary has been sent (i_{36}), the early check-in is suggested (i_{37}), and, once more, it may be accepted (i_{38}) or rejected (i_{39}). The sending of a boarding pass closes the interaction (i_{40}). The remainder of the customer's interaction model, that is, those parts corresponding to vacation and accommodation reservations can be similarly explained.

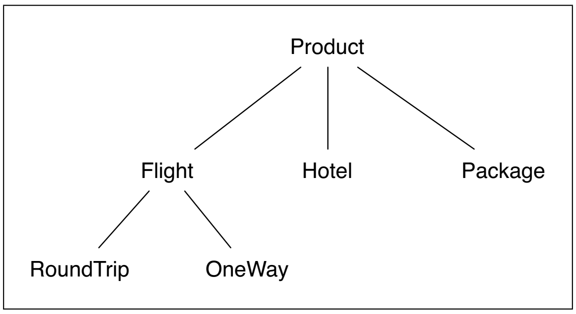
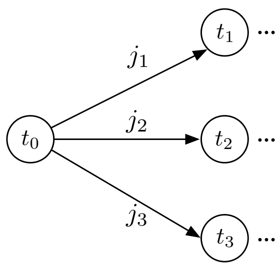


Figure 6.6: Sketch of the travel agent's ontology

The problem is that travel agent b will follow interaction model IM_2 when interacting with customer agent a, and there exist significant differences between interaction models IM_1 and IM_2. Actually, we are assuming that IM_2 conforms to another ontology that we denote by $\mathcal{O}_2$ (see the travel agent's ontology in Appendix B). Among other things, $\mathcal{O}_2$ has an upper class named Booking, instead of Reservation, and a class Product, instead of Item, that is equivalent to the disjunction of three subclasses: Flight, Hotel and Package (see Figure 6.6 above). Regarding the interaction models, IM_2 allows the customer agent to be "flexible on dates" (illocution j_{11}) when requesting flights and travel packages, an option that is not available in IM_1. In short: the semantic interoperability is not guaranteed. In the following section, the I-SSA solution is presented.



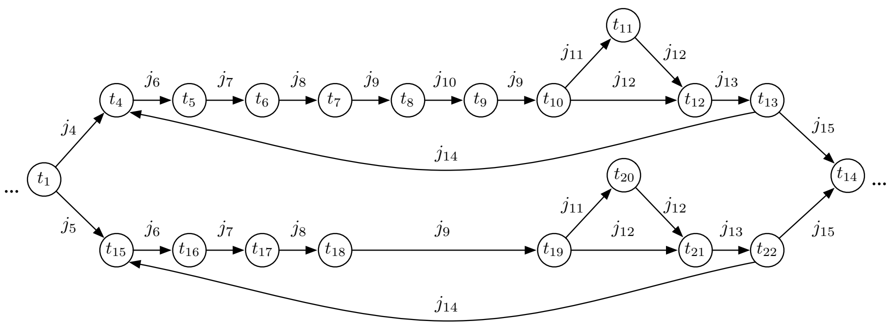
$$j_1 = \langle request, (?A : customer), (?B : travel\_agent), \texttt{flight}\rangle$$
$$j_2 = \langle request, (?A : customer), (?B : travel\_agent), \texttt{hotel}\rangle$$
$$j_3 = \langle request, (?A : customer), (?B : travel\_agent), \texttt{package}\rangle$$

Figure 6.7: Interaction model for the travel agent (part I).

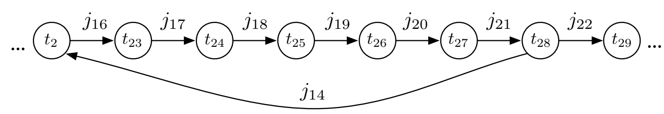
6.4 I-SSA Solution

We have applied the I-SSA technique in the travel reservation scenario from both a theoretical angle, with the computation of the semantic alignment generated by the communication product, and a more practical perspective, letting agents interact through a meta-level alignment protocol and make use of the matching mechanism. Let us start with the latter approach. Recall that in the current version of I-SSA messages are treated as propositional terms, so all variables in IM_1 and IM_2 were not taken into account in the experimentation.

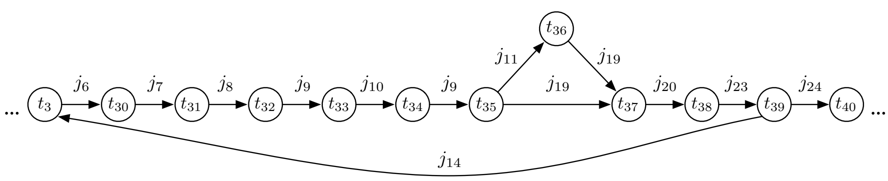
We have run Experiments 1, 2 and 3 as in Section 5.3 of the preceding chapter, but this time with the two distinct interaction models of our travel reservation scenario, namely, IM_1 and IM_2. Remember that with Experiments 1 and 2, we are in position to compare the use of the matching criterion (specifically, the 2nd matching criterion in Section 4.6.2) when updating frequencies or not. With Experiment 3, though, we can estimate the number of interactions necessary to achieve a semantic alignment that ensures a successful interaction with a given probability value. Figures 6.11 and 6.12 show the results of this experimentation. It is clear that the I-SSA technique works well in our travel reservation scenario. Actually, regarding Experiment 3, only 64 interactions are enough to get a probability of success greater than 0.8.



(a) Selecting a flight



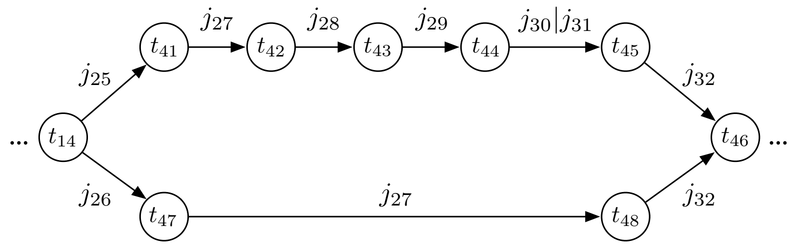
(b) Selecting a hotel



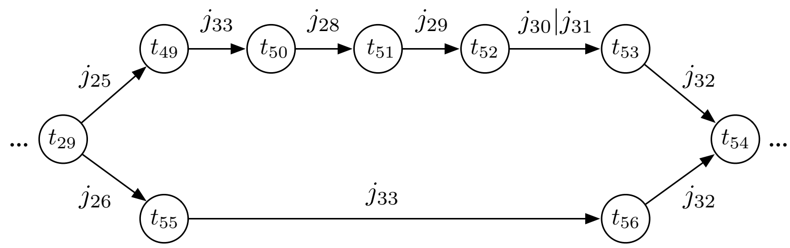
(c) Selecting a package

$$
\begin{aligned}
j_4 &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{roundTrip}\rangle \\
j_5 &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{oneWay}\rangle \\
j_6 &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{from}(?X)\rangle \\
j_7 &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{to}(?Y)\rangle \\
j_8 &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{leavingDate}(?V)\rangle \\
j_9 &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{time}(?T)\rangle \\
j_{10} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{returnDate}(?W)\rangle \\
j_{11} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{flexibleOnDates}\rangle \\
j_{12} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{passengers}(?P)\rangle \\
j_{13} &= \langle inform, (!B : travel\_agent), (!A : customer), \texttt{flightOutcome}(?FL)\rangle \\
j_{14} &= \langle request, (!A : customer), (!B : travel\_agent), \texttt{search}\rangle \\
j_{15} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{myFlight}(?F)\rangle \\
j_{16} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{city}(?X)\rangle \\
j_{17} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{register}(?V)\rangle \\
j_{18} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{nights}(?N)\rangle \\
j_{19} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{rooms}(?R)\rangle \\
j_{20} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{lodgers}(?L)\rangle \\
j_{21} &= \langle inform, (!B : travel\_agent), (!A : customer), \texttt{hotelOutcome}(?HL)\rangle \\
j_{22} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{myHotel}(?H)\rangle \\
j_{23} &= \langle inform, (!B : travel\_agent), (!A : customer), \texttt{packageOutcome}(?PL)\rangle \\
j_{24} &= \langle inform, (!A : customer), (!B : travel\_agent), \texttt{myPackage}(?P)\rangle
\end{aligned}
$$

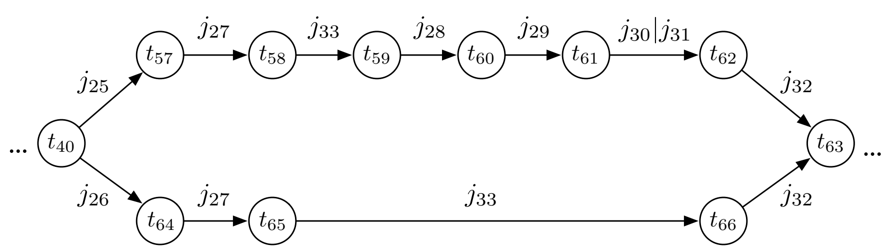
Figure 6.8: Interaction model for the travel agent (part II)



(a) Flight info



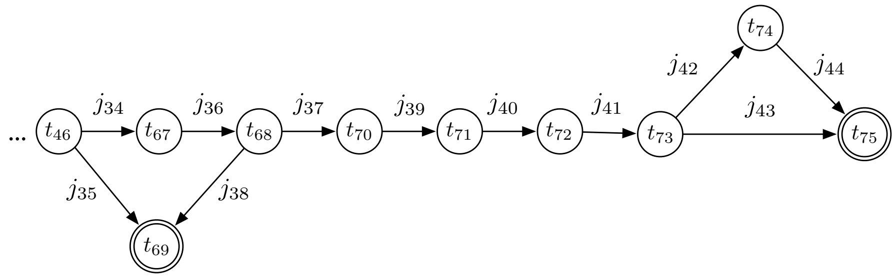
(b) Hotel info



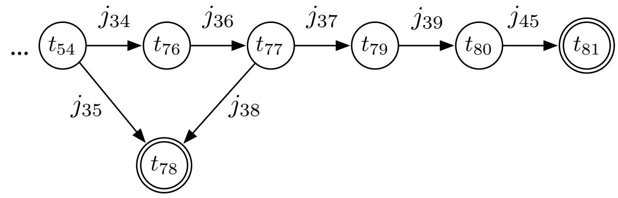
(c) Package info

$j_{25} = \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{noUser}\rangle$
$j_{26} = \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{user}\rangle$
$j_{27} = \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{passengerData}(?PD)\rangle$
$j_{28} = \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{contactInfo}(?CI)\rangle$
$j_{29} = \langle \mathit{suggest}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{account}\rangle$
$j_{30} = \langle \mathit{accept}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{account}\rangle$
$j_{31} = \langle \mathit{reject}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{account}\rangle$
$j_{32} = \langle \mathit{inform}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{rulesAndRestrictions}(?RR)\rangle$
$j_{33} = \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{lodgerData}(?LD)\rangle$

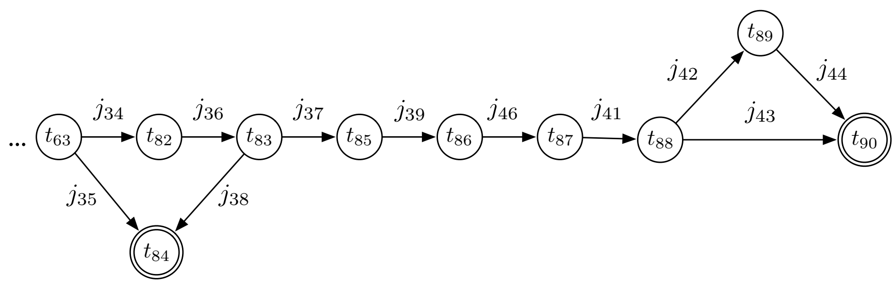
Figure 6.9: Interaction model for the travel agent (part III)



(a) Flight purchase



(b) Hotel purchase



(c) Package purchase

$j_{34} = \langle \mathit{accept}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{rulesAndRestrictions}(!RR)\rangle$
$j_{35} = \langle \mathit{reject}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{rulesAndRestrictions}(!RR)\rangle$
$j_{36} = \langle \mathit{inform}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{totalPrice}(?TP)\rangle$
$j_{37} = \langle \mathit{commit}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{totalPrice}(!TP)\rangle$
$j_{38} = \langle \mathit{reject}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{totalPrice}(!TP)\rangle$
$j_{39} = \langle \mathit{inform}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{payingInfo}(?PI)\rangle$
$j_{40} = \langle \mathit{inform}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{flightSummary}(?FS)\rangle$
$j_{41} = \langle \mathit{suggest}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{checkIn}\rangle$
$j_{42} = \langle \mathit{accept}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{checkIn}\rangle$
$j_{43} = \langle \mathit{reject}, (!A : \mathit{customer}), (!B : \mathit{travel\_agent}), \texttt{checkIn}\rangle$
$j_{44} = \langle \mathit{inform}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{boardingPass}(?BP)\rangle$
$j_{45} = \langle \mathit{inform}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{hotelSummary}(?HS)\rangle$
$j_{46} = \langle \mathit{inform}, (!B : \mathit{travel\_agent}), (!A : \mathit{customer}), \texttt{packageSummary}(?HS)\rangle$

Figure 6.10: Interaction model for the travel agent (part IV)

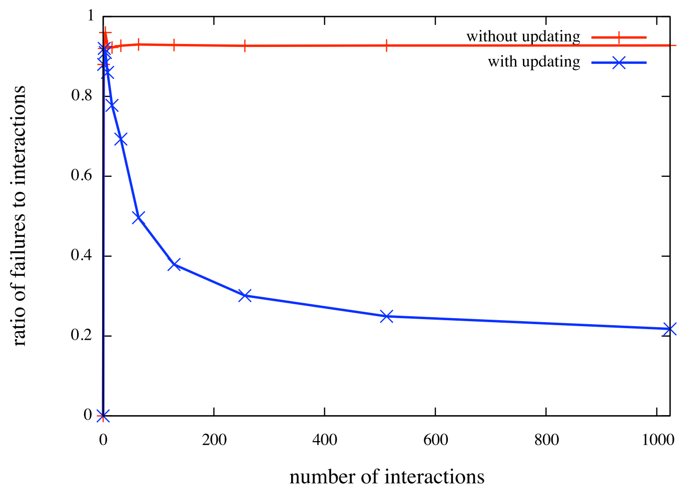


Figure 6.11: Experiments 1 and 2 in the travel reservation scenario

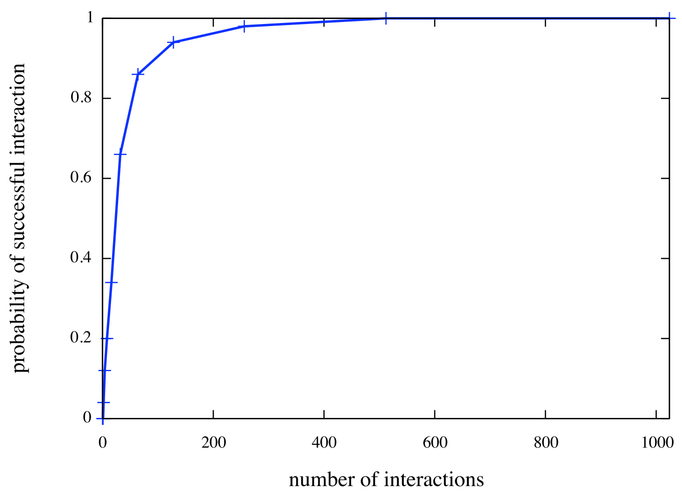


Figure 6.12: Experiment 3 in the travel reservation scenario

$$
\begin{aligned}
\langle id_1, \texttt{flight}\rangle &\equiv \langle id_2, \texttt{flight}\rangle\\
\langle id_1, \texttt{accommodation}\rangle &\equiv \langle id_2, \texttt{hotel}\rangle\\
\langle id_1, \texttt{vacation}\rangle &\equiv \langle id_2, \texttt{package}\rangle\\
\langle id_1, \texttt{return}\rangle &\equiv \langle id_2, \texttt{roundTrip}\rangle\\
\langle id_1, \texttt{single}\rangle &\equiv \langle id_2, \texttt{oneWay}\rangle\\
\langle id_1, \texttt{origin}\rangle &\equiv \langle id_2, \texttt{from}\rangle\\
\langle id_1, \texttt{destination}\rangle &\equiv \langle id_2, \texttt{to}\rangle\\
\langle id_1, \texttt{departing}\rangle &\equiv \langle id_2, \texttt{leavingDate}\rangle\\
\langle id_1, \texttt{returning}\rangle &\equiv \langle id_2, \texttt{returnDate}\rangle\\
\langle id_1, \texttt{outboundTime}\rangle &\sqsubseteq \langle id_2, \texttt{time}\rangle\\
\langle id_1, \texttt{inboundTime}\rangle &\sqsubseteq \langle id_2, \texttt{time}\rangle\\
\langle id_1, \texttt{numberOfPassengers}\rangle &\equiv \langle id_2, \texttt{passengers}\rangle\\
\langle id_1, \texttt{hotelBookingsIn}\rangle &\equiv \langle id_2, \texttt{city}\rangle\\
\langle id_1, \texttt{signIn}\rangle &\equiv \langle id_2, \texttt{register}\rangle\\
\langle id_1, \texttt{signOut}\rangle &\equiv \langle id_2, \texttt{nights}\rangle\\
\langle id_1, \texttt{numberOfRooms}\rangle &\equiv \langle id_2, \texttt{rooms}\rangle\\
\langle id_1, \texttt{numberOfGuests}\rangle &\equiv \langle id_2, \texttt{lodgers}\rangle\\
\langle id_1, \texttt{result}\rangle &\sqsubseteq \langle id_2, \texttt{flightOutcome}\rangle \sqcup\\
&\quad\ \langle id_2, \texttt{hotelOutcome}\rangle \sqcup\\
&\quad\ \langle id_2, \texttt{packageOutcome}\rangle\\
\langle id_1, \texttt{choice}\rangle &\sqsubseteq \langle id_2, \texttt{myFlight}\rangle \sqcup\\
&\quad\ \langle id_2, \texttt{myHotel}\rangle \sqcup\\
&\quad\ \langle id_2, \texttt{myPackage}\rangle\\
\langle id_1, \texttt{search}\rangle &\equiv \langle id_2, \texttt{search}\rangle\\
\langle id_1, \texttt{unregisteredCustomer}\rangle &\equiv \langle id_2, \texttt{noUser}\rangle\\
\langle id_1, \texttt{registeredCustomer}\rangle &\equiv \langle id_2, \texttt{user}\rangle\\
\langle id_1, \texttt{passengerDetails}\rangle &\equiv \langle id_2, \texttt{passengerData}\rangle\\
\langle id_1, \texttt{guestDetails}\rangle &\equiv \langle id_2, \texttt{lodgerData}\rangle\\
\langle id_1, \texttt{customerDetails}\rangle &\equiv \langle id_2, \texttt{contactInfo}\rangle\\
\langle id_1, \texttt{registration}\rangle &\equiv \langle id_2, \texttt{account}\rangle\\
\langle id_1, \texttt{reservationTerms}\rangle &\equiv \langle id_2, \texttt{rulesAndRestrictions}\rangle\\
\langle id_1, \texttt{totalAmountToPay}\rangle &\equiv \langle id_2, \texttt{totalPrice}\rangle\\
\langle id_1, \texttt{paymentInfo}\rangle &\equiv \langle id_2, \texttt{payingData}\rangle\\
\langle id_1, \texttt{reservation\_summary}\rangle &\sqsubseteq \langle id_2, \texttt{flightSummary}\rangle \sqcup\\
&\quad\ \langle id_2, \texttt{hotelSummary}\rangle \sqcup\\
&\quad\ \langle id_2, \texttt{packageSummary}\rangle\\
\langle id_1, \texttt{checkIn}\rangle &\equiv \langle id_2, \texttt{checkIn}\rangle\\
\langle id_1, \texttt{boardingPass}\rangle &\equiv \langle id_2, \texttt{boardingPass}\rangle
\end{aligned}
$$

Figure 6.13: I-SSA semantic alignment in the travel reservation scenario

Figure 6.13 shows the semantic alignment generated by the communication product of IM_1 and IM_2. We provide a detailed explanation of this solution in the following section, where we compare it with the solutions given by other matching techniques.

6.5 Comparison with other Techniques

We have launched three state-of-the-art matchers with the ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, specifically with COMA++, Falcon-OA and OLA (see Section 2.2.3). Matching results are given in Tables 6.1, 6.3 and 6.2. The overall impression is that I-SSA is better suited for this travel reservation scenario, but the purpose of this section is to highlight the differences between the semantic alignments produced by I-SSA and the rest of the matchers.

The three selected matchers use element-level and structure-level techniques to find similarities between ontological entities. Both COMA++ and OLA also resort to external sources, such as auxiliary thesauri, to discover synonymity or hypernymity relations. Let us start talking about the I-SSA semantic alignment and we will compare it with the results provided by the other matchers.

According to the interaction model IM_1, the customer is supposed to request either a flight, an accommodation or a vacation to the travel agent at the initial state s_0. But according to IM_2, the travel agent is supposed to receive a request of either a flight, a hotel or package at the initial state t_0. Now, I-SSA finds the following equivalences (which certainly are the expected ones):

$$\langle id_1, \texttt{flight} \rangle \equiv \langle id_2, \texttt{flight} \rangle \tag{6.1}$$
$$\langle id_1, \texttt{accommodation} \rangle \equiv \langle id_2, \texttt{hotel} \rangle \tag{6.2}$$
$$\langle id_1, \texttt{vacation} \rangle \equiv \langle id_2, \texttt{package} \rangle \tag{6.3}$$

These semantic relationships are deduced from the communication product. Since there are no disjunctions, each equivalence, if computed in practice, must have 1.0 of confidence measure. The fact that the probability of success is 100% after a number of interactions in Experiment 3 ensures that all these equivalences are indeed computed by the travel agent, which is the recipient of the messages.

Nevertheless, COMA++, Falcon-OA and OLA returned weaker results. Let us focus on their attempts to match the classes Accommodation and Hotel. To begin with, OLA found the relation Accommodation $\equiv$ Account [0.38]. This equivalence is partially based on the apparent syntactic similarity between the two terms. This relationship, though, could never be returned by I-SSA. Observe that, according to the interaction model IM_2, the term `account` can be conveyed for the first time in states t_{43}, t_{51} or t_{60}, where the travel agent suggests making an account for the customer. Furthermore, as already mentioned, an account is *suggested*, whereas an accommodation is *requested*. These facts violate principles 1 and 2 of I-SSA (see Section 4.4.1), respectively. COMA++ with a node strategy (that is, without taking into account any structural aspect of the ontologies) returned Accommodation $\equiv$ Account [0.23], and Falcon-OA did not output any relation involving neither Accommodation nor Account.

The relationship Item ≡ Hotel [0.4] was also discovered by OLA. For I-SSA, however, semantic alignment is deliberately limited to the terms that may arise during an interaction. Note that the class Item has no counterpart message in the interaction model IM_1. It is used in ontology $\mathcal{O}_1$ to gather the three kinds of items on which the travel agent make reservations, and it is related to the class Reservation by means of the property hasItem, so that a reservation whose item falls into the class Flight is a flight reservation, ditto Accommodation and Vacation. The class Item does not appear in IM_1 because it is not necessary for a reservation description to be completed. So I-SSA could never find a semantic relationship that involves Item. Analogously, carrier ≡ city [0.55], also returned by OLA, could not be discovered by I-SSA. The property carrier is not taken into account in IM_1 either. In fact, this is the reason why the customer is presented with more that one result once she has described her desired flight: flights may differ in the specific carrier that operates them. Neither COMA++ with a node strategy nor Falcon-OA output any relation involving Hotel.

OLA resorts to WordNet to find semantic relationships between ontological terms. It is worth checking the WordNet synsets of 'accommodation' and 'hotel':

> S: (n) adjustment, **accommodation**, fitting (making or becoming suitable; adjusting to circumstances)
>
> S: (n) **accommodation** (a settlement of differences) "*they reached an accommodation with Japan*"
>
> S: (n) **accommodation** (in the theories of Jean Piaget: the modification of internal representations in order to accommodate a changing knowledge of reality)
>
> S: (n) **accommodation** (living quarters provided for public convenience) "*overnight accommodations are available*"
>
> S: (n) **accommodation** (the act of providing something (lodging or seat or food) to meet a need)
>
> S: (n) **accommodation** ((physiology) the automatic adjustment in focal length of the natural lens of the eye)

and here we have the sole synset of 'hotel':

> S: (n) **hotel** (a building where travellers can pay for lodging and meals and other services)

As OLA, S-Match also falls back on WordNet (see Section 2.2.3), but it only takes class hierarchies as inputs, and our ontologies are relatively rich in terms of properties. Instead we launched S-Match with the two ontology fragments depicted in Figures 6.1 and 6.6. No relationship was returned at all. The fact that 'accommodation' and 'hotel' are semantically related in I-SSA is motivated by their usage in the context of a travel reservation, albeit they may be entirely unrelated in a fully non-contextual external source

In I-SSA, whether to consider the matching element [`accommodation/hotel`] valid or not is ultimately relative to the success of the agents in an interaction where this matching decision is made (see Principle 3 in Section 4.4.1). This means that the validity of a matching element depends on the validity of forthcoming matching elements during an interaction. Now, COMA++ (with a strategy of context), Falcon-OA and OLA, they all consider structural aspects of ontologies when producing a semantic alignment. Broadly speaking, they look at the node neighbours in the graph-like structures of ontologies (children, leaves), so that whether to match two nodes or not also depends on matching their respective graph neighbours. Matching, thus, takes into account the paths in the ontology graphs. Analogously, I-SSA just considers the interaction paths delimited by the communication product of interaction models. For example, the fact that the property hotelBookingsIn is related with the class Accommodation in $\mathcal{O}_1$ is reflected in the fact that the term `hotelBookingsIn` is to be sent after the term `accommodation` according to IM_1. Whether to match `accommodation` with `hotel` or not does influence the relation between `hotelBookingsIn` with `city`, and this is specified as interaction paths in the communication product of IM_1 and IM_2.

To conclude this section, let us say that both COMA++ and OLA returned Flight $\equiv$ Flight [0.7], while Flight $\equiv$ Customer [0.01] for Falcon-OA. Regarding the classes Vacation and Package, COMA++ did not output any relationship between them, whereas OLA output Vacation $\equiv$ Package [0.18], and Falcon-OA discovered the relation Return $\equiv$ Package [0.41].

6.6 Summary and Concluding Remarks

In this chapter we have applied the I-SSA technique to a realistic scenario about travel reservation. This has given us the possibility to put the I-SSA approach into context with current state-of-the-art matching techniques.

We have compared I-SSA with COMA++, Falcon-OA and OLA on the basis of the travel reservation scenario. I-SSA proved to be better suited for semantic alignment in an example where the interaction context is very relevant. Although a deeper comparison is obviously required, this first attempt allowed us to highlight the main differences between the I-SSA approach and more standard techniques for ontology matching.

Ontology1	Ontology2	Similarity	Relation
Accommodation	Account	0.23	=
boardingPass	boardingPass	1.0	=
carrier	city	0.39	=
checkIn	checkIn	0.7	=
CurrentSearch	CurrentSearch	0.7	=
Customer	Customer	0.7	=
departingDate	leavingDate	0.46	=
Flight	Flight	0.7	=
hasCustomer	hasCustomer	0.7	=
hasIdentification	hasIdentification	1.0	=
hasItem	hasProduct	0.63	=
inboundTime	time	0.76	=
Item	Product	0.56	=
numberOfGuests	nights	0.4	=
numberOfPassengers	passengers	0.65	=
numberOfRooms	rooms	0.67	=
outboundTime	time	0.76	=
passengerDetails	passengerData	0.69	=
paymentInfo	payingInfo	0.77	=
reservationSummary	flightSummary	0.65	=
reservationSummary	hotelSummary	0.65	=
reservationSummary	packageSummary	0.65	=
returning	returnDate	0.59	=
Search	Search	0.7	=
totalAmountToPay	totalPrice	0.63	=

Table 6.1: COMA++ results with a node strategy

Ontology1	**Ontology2**	**Similarity**	**Relation**
Return	Package	0.41	=
Single	RoundTrip	0.19	=
Customer	Booking	0.05	=
RegisteredCustomer	User	0.04	=
UnregisteredCustomer	OneWay	0.03	=
Result	Outcome	0.01	=
Item	Product	0.01	=
CurrentSearch	NonUser	0.01	=
Search	Search	0.01	=
Flight	Customer	0.01	=
boardingPass	boardingPass	1.0	=
destination	airlineCompany	0.99	=
carrier	to	0.99	=
numberOfPassengers	passengers	0.99	=
departing	leavingDate	0.99	=
origin	from	0.99	=
hasCustomer	hasCustomer	0.98	=
outboundTime	flexibleOnDates	0.86	=
returning	returnDate	0.76	=
hasItem	hasProduct	0.74	=
inboundTime	lodgerData	0.63	=
guestDetails	passengerData	0.56	=
numberOfRooms	rooms	0.56	=
rate	stars	0.56	=
passengerDetails	contactInfo	0.56	=
checkIn	checkIn	0.52	=
customerDetails	rulesAndRestrictions	0.39	=
reservationTerms	payingInfo	0.30	=
totalAmountToPay	totalPrice	0.30	=
paymentInfo	hotelSummary	0.30	=
reservationSummary	packageSummary	0.30	=
hotelBookingsIn	city	0.30	=
signOut	register	0.30	=
numberOfGuests	nights	0.30	=
hasIdentification	hasIdentification	0.30	=
signIn	flightSummary	0.28	=
hasResult	hasOutcome	0.11	=

Table 6.2: Falcon-AO results

Ontology1	Ontology2	Similarity	Relation
departing	leavingDate	0.58	=
Registration	Booking	0.08	=
customerDetails	lodgerData	0.57	=
numberOfGuests	nights	0.47	=
hotelBookingsIn	airlineCompany	0.49	=
reservationTerms	rulesAndRestrictions	0.55	=
outboundTime	flexibleOnDates	0.25	=
carrier	city	0.55	=
CurrentSearch	CurrentSearch	1.0	=
numberOfPassengers	passengers	0.61	=
passengerDetails	passengerData	0.80	=
Result	Product	0.43	=
Reservation	User	0.27	=
hasItem	hasOutcome	0.35	=
reservationSummary	flightSummary	0.68	=
Flight	Flight	0.70	=
RegisteredCustomer	HotelOutcome	0.39	=
rate	stars	0.51	=
Search	Search	1.0	=
Return	RoundTrip	0.33	=
Customer	Customer	1.0	=
hasIdentification	hasIdentification	1.0	=
Item	Hotel	0.4	=
Item	Product	0.0	=
inboundTime	time	0.66	=
paymentInfo	payingInfo	0.80	=
returning	returnDate	0.60	=
signOut	register	0.44	=
totalAmountToPay	totalPrice	0.56	=
Accommodation	Account	0.38	=
hasCustomer	hasCustomer	0.91	=
origin	from	0.53	=
boardingPass	boardingPass	0.91	=
Vacation	Package	0.18	=
signIn	contactInfo	0.18	=
numberOfRooms	rooms	0.58	=
destination	too	0.53	=
hasResult	hasProduct	0.36	=
checkIn	checkIn	0.79	=
Single	OneWay	0.33	=
guestDetails	hotelSummary	0.53	=
UnregisteredCustomer	NonUser	0.50	=

Table 6.3: OLA results

Chapter 7

General Conclusions and Further Work

Until recently, most ontology matching mechanisms developed have taken a classical functional approach to the semantic heterogeneity problem, in which ontology matching is seen as a process taking two or more ontologies as input and producing a semantic alignment of ontological entities as output. During this process, matchers typically analyse the syntactic form of ontological entities, examine the structure of ontologies, and fall back on external sources and background knowledge, such as dictionaries, thesauri and upper-level ontologies, in order to find the semantic relationships between ontological entities. Furthermore, in most of these systems matching is performed at design-time, prior to integration or interaction.

This generalised approach of matching involves, however, several drawbacks. On the one hand, it limits the dynamism and openness, when many matching applications —multiagent communication, peer-to-peer information sharing, or web-service composition— are of a clear decentralised, dynamic and open-ended nature, and they require matching to be locally performed at run-time. On the other hand, it keeps matching outside the context of the interaction where it is actually needed. Semantic similarity of ontological terms is established in an interaction-independent fashion, for example, by means of a external source like WordNet, where relations as synonymy, among others, was determined prior to interaction and independently from it.

Although there exist efforts to apply ontology matching at run-time and only amongst those fragments that are deemed relevant for the task at hand or the current interaction, proposed dynamic ontology matching still follows a classical functional approach: when a mismatch occurs, semantic heterogeneity is resolved by applying current state-of-the-art ontology matching techniques, albeit only for a fragment and at run-time.

In this thesis we have presented two approaches for semantic alignment in multiagent communication with the aim of complementing the solutions applied so far, and that try to overcome the two limitations remarked above. Situated Semantic Alignment and Interaction-Situated Semantic Alignment specifically bring the situation or interaction in which agents are engaged into the alignment mechanism to avoid dependency on *a priori* semantic agreements. Additionally, in both approaches agent terminologies are incrementally aligned, which favours the dynamism and openness of multiagent systems.

The implications of our research can be studied from three different angles: a theoretical one, a practical one, and a philosophical one.

7.1 Theoretical Implications

In Situated Semantic Alignment we consider a scenario with two or more agents situated in an environment. It is assumed that each agent has its own viewpoint of the environment so that, if the environment is in a concrete state, both agents have different perceptions of this state. Because of these differences there may be a mismatch in the meaning of the syntactic entities by which agents describe their perceptions (and which constitutes the agents' ontologies), and if so, syntactic entities are related according to the intrinsic semantics provided by the existing relationship between the agents' viewpoints of the environment.

We have provided a general framework for situated semantic alignment based on channel theory, Barwise and Seligman's theory of information flow. Semantic alignment is formalised in terms of a distributed logic referred to as SSA logic, which agents can approximate through communication, and this communication process is modelled as a sequence of information-channel refinements.

In this framework we deliberately do not commit on the kind of environment in which agents are situated, or what these perceive of it. Actually, the latter is just captured by a function from the set of environment states to a set the elements of which are called state perceptions. This level of generality becomes apparent when presenting Interaction-Situated Semantic Alignment as a particular case of Situated Semantic Alignment. Interaction states pass for environment states, where the environment is realised in the communication product. Nonetheless, we do believe that this formal framework is general enough to capture different approximations.

In the particular case of interaction-situated semantic alignment, we have given a formalisation in terms of the communication product, which is proven to be a pullback —constrained product— in the category of interaction models. Although not addressed in this dissertation, this formalisation paves the way to apply results of category theory in an *a priori* unrelated topic as that one of semantic alignment or ontology matching, in the same manner as, for instance, Zimmermann et al in [Zimmermann et al., 2006].

Interaction-situated semantic alignment, though, was primarily intended to tackle the pragmatical aspects of situated semantic alignment. In the following section we stress the practical implications of this approach, specifically relating to ontology matching and how current techniques can benefit from it.

7.2 Practical Implications

The viability of the I-SSA approach has been evinced through an exhaustive abstract experimentation, as well as a thorough statistical study. Through the combination of analyses of variance and Tukey tests we have been able to identify which factors —number of illocution heads, messages and interactions— have an influence on the total amount of failed interactions, and which values do better for each of the independent variables.

Besides showing the viability of I-SSA, we have also applied it in a realistic scenario about travel reservation. This allowed us to put I-SSA in the context of current state-of-the-art matching techniques. We have designed two interaction models that conform to two different ontologies, and launched the former with I-SSA and the latter with the matchers COMA++, Falcon-OA and OLA. The results showed that I-SSA is better suited for semantic alignment where the interaction context is specially relevant. A deeper study is clearly required for a comparison to be fully reliable, but this example is useful to highlight some of the main differences between I-SSA and standard matching approaches, and to convince the reader of the potential of this technique.

Nonetheless, it is still a challenge to find out how the I-SSA technique can be applied at a greater scale. Before all else, though, some extensions are to be made, such as more expressiveness in content languages or a more sophisticated matching mechanism. We talk about these issues in detail below in the section related with further work. In any case, the way we look at our approach is as a complement, rather than as a replacement to current state-of-the-art matching techniques, as a manner of bringing the interaction context into the alignment mechanism.

In the following section we give some philosophical remarks concerning the I-SSA approach [Schorlemmer and Atencia, 2009].

7.3 Philosophical Implications

Modern hermeneutics, as initiated by Heidegger and Gadamer, has shown that language is listened in a background, and that interpretation is not independent of the interpreter. Meaning, thus, is always re-created in the context of the purposes, expectations, and commitments the interpreter attaches to its usage or utterance. Meaning is ultimately interaction-dependent and relative to an implicit background, which cannot be completely de-contextualised.

The core of this thesis is the appreciation that most of the current approaches for semantic alignment do not tackle the interaction-dependence of meaning in depth.

Semantic Alignment as a Wittgenstein Language Game

We investigated how software agents may establish the semantic relationships between their respective terminologies on the grounds of their communication within a specific interaction by *taking interaction ontologically prior to meaning.* As with Wittgenstein's language games [Wittgenstein, 1967], the meaning of those terms uttered by each agent arises by how the agents actually *make use* of them in the interaction, which, in some respect, can be seen as a simple language game. We assume that agents follow certain interaction models, or protocols (the game rules), according to which they are allowed to make certain utterances at certain interaction states. These utterances are in the form of illocutionary speech acts whose content are the words of the game language (such as "stand" or "no more cards"). When an agent listens to an utterance whose content it does not understand, it does so against the background of a particular interaction state. It will have to guess among the possible alternatives regarding its own view of the interaction, assuming that all agents are in the same or compatible interaction state.

The "meaning" an agent attaches to a term, then, is the interaction state transition it believes is the result from its utterance in a speech act, according to the agent's view of the interaction and of the current interaction state. As with a language game, the guesses of what the meanings of the words are may be wrong, which will eventually lead to a breakdown of the communication: the interaction has not progressed in the direction foreseen by the interaction models of each agent. Agents can be aware of such a breakdown if they are capable of communicating about the interactions themselves.

In our model agents follow both their own interaction protocol and also an alignment protocol in parallel. This alignment protocol is seen as a meta-protocol through which the actual communication is carried out. In addition, agents are endowed with an alignment mechanism used to perform the actual matching. Now, matching elements are reinforced as many interactions are completed and this reinforcement is based on statistical reasoning. Eventually, terms are deemed semantically related when they trigger compatible interaction state transitions, where compatibility means that the interaction progresses in the same direction for each agent —albeit their interaction views (that is, their own interaction models) may be more constrained than the interaction that is actually happening.

7.4 Further Work

Situated Semantic Alignment and Interaction-Situated Semantic Alignment are novel approaches to tackle semantic heterogeneity in multiagent systems. We do believe that they already represent an advancement in its current state, but also that many potential research lines can be envisaged. We plan to extend both the theoretical and practical aspects of these approaches.

Modularity and interaction. Modularity is a fundamental issue in ontology engineering since it allows for a better understanding of ontologies, as well as it facilitates ontology maintaining and reasoning. Several frameworks for ontology modularity have been proposed, such as, *e.g.*, local model semantics and multi-context systems [Ghidini and Giunchiglia, 2001], or distributed description logics [Borgida and Serafini, 2003], or the so-called $\mathcal{E}$-connections [Kutz et al., 2004], to name a few. Modularity is commonly seen as an aspect that is separate from the situation where ontologies are actually needed. The framework for situated semantic alignment that we have described in this thesis is, in our opinion, appropriate for looking into modularity from an interaction point of view. We believe that channel theory as a theory for reasoning about the information flow in distributed systems can give us the theoretical tools that are needed for this endeavour.

More expressiveness in content languages. In this work, all messages of interaction models are treated as propositions, that is, as grounded atomic sentences. But for interaction models to capture any kind of application, at least we have to consider variables. So first-order expressiveness in interaction models is an expected enhancement. In this sense, we think that Besana and Robertson's work on mapping predictions can be helpful [Besana and Robertson, 2007]. In this work, the authors opt to attach probabilities to meanings of terms that are determined by earlier, similar interactions, and then these probabilities are used for predicting the set of possible meanings of a message. Meaning is also defined relative to a particular interaction, but the authors aim at reducing the search space of possible a priori mappings (in a classical sense), namely by assessing those ones with highest probability in the context of an interaction.

New versions of I-SSA Principle 3. Recall that I-SSA Principle 3* stated in Section 4.4.1 has been put forward as a particular case of the more general one Principle 3. However, other principles can be proposed. One possible extension is to take into account commitments made by interaction model players while interacting. So for an interaction to be considered successful it is necessary that all players' commitments are ultimately satisfied. In fact, commitments enable to have checkpoints in mid-interactions, and, thus, they allow agents to detect failures earlier, as well as to achieve a more accurate alignment.

Past unsuccessful interactions. Current version of the matching mechanism only keeps track of past successful interactions, and unsuccessful ones are simply discarded. Clearly, this is a great loss, since agents could also learn from past matching mistakes. The problem is that it is not straightforward to figure out which matching was responsible of a failure, or if we should blame one agent or another. Again, a probabilistic approach seems to be appropriate for this matter, attaching values to matching elements that vary as more interactions are completed, regardless of whether they are successful or unsuccessful. This should considerably improve the matching mechanism in terms of learning speed.

Ontological information. One of the main characteristics of I-SSA is that it is not fully aware of ontological information. I-SSA semantic alignment is more conformed to agents' use of messages while interacting, though ontological information is actually implicit in this usage. I-SSA assumes that agents' ontologies are not open for inspection, but nothing prevents agents from taking advantage of their own ontological information. Indeed an agent could reason about the relations between their own messages when matching a received one. The basic example refers to disjointness. Imagine that an agent matched message m with a in the past, and that this matching led him to interact successfully. The fact that m and m' are disjoint according to his own ontology should stop him from matching a with m' in future interactions.

Comparison with other matching techniques. Chapter 6 was an effort to compare state-of-the-art matchers with the I-SSA approach, but a deeper comparison study is required. The main drawback is the lack of sufficiently complex interaction models with which to experiment, since current trends for ontology alignment evaluation do not take into account the interactions in which ontologies are deployed.

Combination with other matching techniques. I-SSA was initially motivated by the fact that most of the current state-of-the-art matchers ignore pragmatics. But more than a replacement, we believe that I-SSA is a good complement for these matchers, and it is in our mind to work on this line in the future. Besides a comparison with other matching techniques as stated above, we plan to experiment in order to evaluate the performance of matchers in conjunction with I-SSA.

Some of the future research lines that we have proposed above require to redefine interaction models. We also plan to take formalisms more sophisticated than finite automata such as Electronic Institutions [Arcos et al., 2005], Petri Nets or LCC [Robertson, 2004]. In addition, the notions of communication product and alignment protocol should be rethought.

Appendix A

Channel Theory

This appendix includes the definitions of all the terms and theorems of channel theory that are used in Chapter 3.

A.1 Channel Theory Terms

Classification: is a tuple $\mathbf{A} = \langle tok(\mathbf{A}), typ(\mathbf{A}), \models_{\mathbf{A}} \rangle$ where $tok(\mathbf{A})$ is a set of *tokens*, $typ(\mathbf{A})$ is a set of *types* and $\models_{\mathbf{A}}$ is a binary relation between $tok(\mathbf{A})$ and $typ(\mathbf{A})$. If $a \models_{\mathbf{A}} \alpha$ then a is said to be *of type* α.

Infomorphism: $f : \mathbf{A} \to \mathbf{B}$ from classifications $\mathbf{A}$ to $\mathbf{B}$ is a contravariant pair of functions $f = \langle f^{\to}, f^{\leftarrow} \rangle$, where $f^{\to} : typ(\mathbf{A}) \to typ(\mathbf{B})$ and $f^{\leftarrow} : tok(\mathbf{B}) \to tok(\mathbf{A})$, satisfying the following fundamental property:

$$f^{\leftarrow}(b) \models_{\mathbf{A}} \alpha \quad \text{iff} \quad b \models_{\mathbf{B}} f^{\to}(\alpha)$$

for each token $b \in tok(\mathbf{B})$ and each type $\alpha \in typ(\mathbf{A})$.

Strictly speaking, an infomorphism f comprehends on the one hand a pair of classifications $\langle \mathbf{A}, \mathbf{B} \rangle$ and on the other hand a contravariant pair of functions $\langle f^{\to}, f^{\leftarrow} \rangle$ defined as above.

Channel: consists of two infomorphisms $\mathcal{C} = \{f_i : \mathbf{A}_i \to \mathbf{C}\}_{i \in \{1,2\}}$ with a common codomain $\mathbf{C}$, called the *core* of $\mathcal{C}$. $\mathbf{C}$ tokens are called *connections* and a connection c is said to *connect* tokens $f_1^{\leftarrow}(c)$ and $f_2^{\leftarrow}(c)$.

In fact, this is the definition of a binary channel. A channel can be defined with an arbitrary index set.

Sum: given classifications $\mathbf{A}$ and $\mathbf{B}$, the *sum* of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} + \mathbf{B}$, is the classification with $tok(\mathbf{A} + \mathbf{B}) = tok(\mathbf{A}) \times tok(\mathbf{B}) = \{\langle a, b \rangle \mid a \in tok(\mathbf{A}) \text{ and } b \in tok(\mathbf{B})\}$, $typ(\mathbf{A} + \mathbf{B}) = typ(\mathbf{A}) \uplus typ(\mathbf{B}) = \{\langle i, \gamma \rangle \mid i = 1 \text{ and } \gamma \in typ(\mathbf{A}) \text{ or } i = 2 \text{ and } \gamma \in typ(\mathbf{B})\}$ and relation $\models_{\mathbf{A}+\mathbf{B}}$ defined as follows:

$$\begin{array}{lll} \langle a,b\rangle \models_{\mathbf{A}+\mathbf{B}} \langle 1,\alpha\rangle & \text{if} & a \models_{\mathbf{A}} \alpha \\ \langle a,b\rangle \models_{\mathbf{A}+\mathbf{B}} \langle 2,\beta\rangle & \text{if} & b \models_{\mathbf{B}} \beta \end{array}$$

Given infomorphisms $f : \mathbf{A} \to \mathbf{C}$ and $g : \mathbf{B} \to \mathbf{C}$, the sum $f+g : \mathbf{A}+\mathbf{B} \to \mathbf{C}$ is defined on types by $(f+g)^{\to}(\langle 1,\alpha\rangle) = f^{\to}(\alpha)$ and $(f+g)^{\to}(\langle 2,\beta\rangle) = g^{\to}(\beta)$, and on tokens by $(f+g)^{\leftarrow}(c) = \langle f^{\leftarrow}(c), g^{\leftarrow}(c)\rangle$.

Theory: given a set Σ, a *sequent* of Σ is a pair $\langle \Gamma, \Delta\rangle$ of subsets of Σ. A binary relation $\vdash$ between subsets of Σ is called a *consequence relation* on Σ. A *theory* is a pair $T = \langle \Sigma, \vdash\rangle$ where $\vdash$ is a consequence relation on Σ. A sequent $\langle \Gamma, \Delta\rangle$ of Σ for which $\Gamma \vdash \Delta$ is called a *constraint* of the theory T. T is *regular* if it satisfies:

1. *Identity:* $\alpha \vdash \alpha$
2. *Weakening:* if $\Gamma \vdash \Delta$, then $\Gamma, \Gamma' \vdash \Delta, \Delta'$
3. *Global Cut:* if $\Gamma, \Pi_0 \vdash \Delta, \Pi_1$ for each partition $\langle \Pi_0, \Pi_1\rangle$ of Π (*i.e.*, $\Pi_0 \cup \Pi_1 = \Pi$ and $\Pi_0 \cap \Pi_1 = \emptyset$), then $\Gamma \vdash \Delta$

for all $\alpha \in \Sigma$ and all $\Gamma, \Gamma', \Delta, \Delta', \Pi \subseteq \Sigma$.

All theories considered in this dissertation are regular.

Theory generated by a classification: let $\mathbf{A}$ be a classification. A token $a \in tok(\mathbf{A})$ *satisfies* a sequent $\langle \Gamma, \Delta\rangle$ of $typ(\mathbf{A})$ provided that if a is of every type in Γ then it is of some type in Δ. The *theory generated* by $\mathbf{A}$, denoted by $Th(\mathbf{A})$, is the theory $\langle typ(\mathbf{A}), \vdash_{\mathbf{A}}\rangle$ where $\Gamma \vdash_{\mathbf{A}} \Delta$ if every token in $\mathbf{A}$ satisfies $\langle \Gamma, \Delta\rangle$.

Local logic: is a tuple $\mathfrak{L} = \langle tok(\mathfrak{L}), typ(\mathfrak{L}), \models_{\mathfrak{L}}, \vdash_{\mathfrak{L}}, N_{\mathfrak{L}}\rangle$ where:

1. $\langle tok(\mathfrak{L}), typ(\mathfrak{L}), \models_{\mathfrak{L}}\rangle$ is a classification denoted by $Cla(\mathfrak{L})$,
2. $\langle typ(\mathfrak{L}), \vdash_{\mathfrak{L}}\rangle$ is a regular theory denoted by $Th(\mathfrak{L})$,
3. $N_{\mathfrak{L}}$ is a subset of $tok(\mathfrak{L})$, called the *normal tokens* of $\mathfrak{L}$, which satisfy all constraints of $Th(\mathfrak{L})$.

A local logic $\mathfrak{L}$ is *sound* if every token in $Cla(\mathfrak{L})$ is normal, that is, $N_{\mathfrak{L}} = tok(\mathfrak{L})$. $\mathfrak{L}$ is *complete* if every sequent of $typ(\mathfrak{L})$ satisfied by every normal token is a constraint of $Th(\mathfrak{L})$.

Local logic generated by a classification: the *local logic generated* by $\mathbf{A}$, written $Log(\mathbf{A})$, is the local logic on $\mathbf{A}$ (*i.e.*, $Cla(Log(\mathbf{A})) = \mathbf{A}$), with $Th(Log(\mathbf{A})) = Th(\mathbf{A})$ and such that all its tokens are normal, that is, $N_{Log(\mathbf{A})} = tok(\mathbf{A})$.

Inverse image: given an infomorphism $f : \mathbf{A} \to \mathbf{B}$ and a local logic $\mathfrak{L}$ on $\mathbf{B}$, the *inverse image* of $\mathfrak{L}$ under f, denoted $f^{-1}[\mathfrak{L}]$, is the local logic on $\mathbf{A}$ such that $\Gamma \vdash_{f^{-1}[\mathfrak{L}]} \Delta$ if $f^{\to}[\Gamma] \vdash_{\mathfrak{L}} f^{\to}[\Delta]$ and $N_{f^{-1}[\mathfrak{L}]} = f^{\leftarrow}[N_{\mathfrak{L}}] = \{a \in tok(\mathbf{A}) \mid a = f^{\leftarrow}(b) \text{ for some } b \in N_{\mathfrak{L}}\}$.

Distributed logic: let $\mathcal{C} = \{f_i : \mathbf{A}_i \to \mathbf{C}\}_{i\in\{1,2\}}$ be a channel and $\mathfrak{L}$ a local logic on its core $\mathbf{C}$, the *distributed logic* of $\mathcal{C}$ generated by $\mathfrak{L}$, written $DLog_{\mathcal{C}}(\mathfrak{L})$, is the inverse image of $\mathfrak{L}$ under the sum $f_1 + f_2$.

Refinement: let $\mathcal{C} = \{f_i : \mathbf{A}_i \to \mathbf{C}\}_{i\in\{1,2\}}$ and $\mathcal{D} = \{g_i : \mathbf{A}_i \to \mathbf{D}\}_{i\in\{1,2\}}$ be two channels with the same component classifications $\mathbf{A}_1$ and $\mathbf{A}_2$. A *refinement* infomorphism from $\mathcal{D}$ to $\mathcal{C}$ is an infomorphism $r : \mathbf{D} \to \mathbf{C}$ such that for each $i \in \{1,2\}$, $f_i = rg_i$ (*i.e.*, $f_i^{\to} = r^{\to}g_i^{\to}$ and $f_i^{\leftarrow} = g_i^{\leftarrow}r^{\leftarrow}$). Channel $\mathcal{D}$ is a *refinement* of $\mathcal{C}$ if there exists a refinement infomorphism r from $\mathcal{D}$ to $\mathcal{C}$.

A.2 Channel Theory Theorems

Theorem A.1 *The logic generated by a classification is sound and complete. Furthermore, given a classification* $\mathbf{A}$ *and a logic* $\mathfrak{L}$ *on* $\mathbf{A}$, $\mathfrak{L}$ *is sound and complete if and only if* $\mathfrak{L} = Log(\mathbf{A})$.

Theorem A.2 *Let* $\mathfrak{L}$ *be a logic on a classification* $\mathbf{B}$ *and* $f : \mathbf{A} \to \mathbf{B}$ *an infomorphism.*

1. *If* $\mathfrak{L}$ *is complete then* $f^{-1}[\mathfrak{L}]$ *is complete.*
2. *If* $\mathfrak{L}$ *is sound and* $f^{\leftarrow}$ *is surjective then* $f^{-1}[\mathfrak{L}]$ *is sound.*

Appendix B

Travel Ontologies

In this appendix we provide the specifications of the two ontologies used in the travel reservation scenario described in Chapter 6. The ontologies are written in the Manchester OWL syntax, and their expressiveness is $\mathcal{ALCIF}(\mathcal{D})$, which is covered by OWL Lite.

Customer's Ontology

In the customer's travel ontology, customers make reservations of items, and an item can be either a flight, an accommodation or a vacation.

```
Class: Item
    EquivalentTo: Accommodation or Flight or Vacation

DisjointClasses: Flight, Accommodation, Vacation
```

Flights, in turn, can be either return flights or single flights:

```
Class: Flight
    EquivalentTo: Return or Single
    SubClassOf: Item
```

Now, the following is the definition of a return flight:

```
Class: Return
    EquivalentTo: (carrier some string)
        and (departing some date)
        and (destination some string)
        and (inboundTime some time)
        and (numberOfPassengers some positiveInteger)
        and (origin some string)
        and (outboundTime some time)
        and (returning some date)
        and (carrier only string)
        and (departing only date)
        and (destination only string)
```

```
        and (inboundTime only time)
        and (numberOfPassengers only positiveInteger)
        and (origin only string)
        and (outboundTime only time)
        and (returning only date)
    SubClassOf: Flight
    DisjointWith: Single
```

And here we have the definition of a single flight:

```
Class: Single
    EquivalentTo: (not (inboundTime some time))
        and (not (returning some date))
        and (carrier some string)
        and (departing some date)
        and (destination some string)
        and (numberOfPassengers some positiveInteger)
        and (origin some string)
        and (outboundTime some time)
        and (carrier only string)
        and (departing only date)
        and (destination only string)
        and (numberOfPassengers only positiveInteger)
        and (origin only string)
        and (outboundTime only time)
    SubClassOf: Flight
    DisjointWith: Return
```

The following is the definition of an accommodation:

```
Class: Accommodation
    EquivalentTo: (hotelBookingsIn some string)
        and (numberOfGuests some positiveInteger)
        and (numberOfRooms some positiveInteger)
        and (rate some positiveInteger)
        and (signIn some date)
        and (signOut some date)
        and (hotelBookingsIn only string)
        and (numberOfGuests only positiveInteger)
        and (numberOfRooms only positiveInteger)
        and (rate only positiveInteger)
        and (signIn only date)
        and (signOut only date)
    SubClassOf: Item
```

And here we have the definition of a vacation:

```
Class: Vacation
    EquivalentTo: (carrier some string)
        and (departing some date)
        and (destination some string)
        and (inboundTime some time)
```

```
        and (numberOfPassengers some positiveInteger)
        and (numberOfRooms some positiveInteger)
        and (origin some string)
        and (outboundTime some time)
        and (rate some positiveInteger)
        and (returning some date)
        and (carrier only string)
        and (departing only date)
        and (destination only string)
        and (inboundTime only time)
        and (numberOfPassengers only positiveInteger)
        and (numberOfRooms only positiveInteger)
        and (origin only string)
        and (outboundTime only time)
        and (rate only positiveInteger)
        and (returning only date)
    SubClassOf: Item
```

Customers are to be identified, and they can be registered or unregistered.

```
Class: Customer
    EquivalentTo: (hasIdentification some ID)
        and (hasIdentification only ID),
        RegisteredCustomer
        or UnregisteredCustomer

Class: RegisteredCustomer
    SubClassOf: Customer
    DisjointWith: UnregisteredCustomer

Class: UnregisteredCustomer
    EquivalentTo: Customer
        and (not (RegisteredCustomer))
    SubClassOf: Customer
    DisjointWith: RegisteredCustomer
```

Finally, we give the definition of a reservation:

```
Class: Reservation
    EquivalentTo:
        (((((not (boardingPass some anyURI))
        and (checkIn only owl:Nothing))
        or ((checkIn some owl:Thing)
        and (boardingPass some anyURI)
        and (boardingPass only anyURI)))
        and (hasItem some Flight)
        and (passengerDetails some anyURI)
        and (passengerDetails only anyURI))
        or ((((not (boardingPass some anyURI))
        and (checkIn only owl:Nothing))
        or ((checkIn some owl:Thing)
```

```
        and (boardingPass some anyURI)
        and (boardingPass only anyURI)))
        and (hasItem some Vacation)
        and (guestDetails some anyURI)
        and (passengerDetails some anyURI)
        and (guestDetails only anyURI)
        and (passengerDetails only anyURI))
        or ((hasItem some Accommodation)
        and (guestDetails some anyURI)
        and (guestDetails only anyURI)))
        and (((hasCustomer some RegisteredCustomer)
        and (hasCustomer only RegisteredCustomer))
        or ((hasCustomer some UnregisteredCustomer)
        and (hasCustomer only UnregisteredCustomer)
        and (customerDetails some anyURI)
        and (customerDetails only anyURI)))
        and (hasItem some Item)
        and (hasItem only Item)
        and (paymentInfo some anyURI)
        and (reservationSummary some anyURI)
        and (reservationTerms some anyURI)
        and (totalAmountToPay some decimal)
        and (paymentInfo only anyURI)
        and (reservationSummary only anyURI)
        and (reservationTerms only anyURI)
        and (totalAmountToPay only decimal)
```

And the definition of a registration:

```
Class: Registration
    EquivalentTo: (hasCustomer some RegisteredCustomer)
        and (hasCustomer only RegisteredCustomer)
        and (customerDetails some anyURI)
        and (customerDetails only anyURI)
```

Additionally, this ontology describes the part of the domain related with searches and their results.

```
Class: CurrentSearch
    SubClassOf: Search

Class: Result
    EquivalentTo: inv(hasResult) some CurrentSearch

Class: Search
    SubClassOf: owl:Thing

Individual: choice
    Types: Item, owl:Thing
```

The following are the object properties of the customer's ontology:

```
ObjectProperty: hasItem
    Characteristics: Functional
    Domain: Reservation
    Range: Item

ObjectProperty: checkIn
    Characteristics: Functional
    Domain: Reservation

ObjectProperty: hasCustomer
    Characteristics: Functional
    Domain: Registration or Reservation
    Range: Customer

ObjectProperty: hasResult
    Domain: Search
    Range: Item
```

And here we have the datatype properties:

```
DataProperty: destination
    Characteristics: Functional
    Domain: Flight or Vacation
    Range: string

DataProperty: inboundTime
    Characteristics: Functional
    Domain: Return or Vacation
    Range: time

DataProperty: hotelBookingsIn
    Characteristics: Functional
    Domain: Accommodation
    Range: string

DataProperty: returning
    Characteristics: Functional
    Domain: Return or Vacation
    Range: date

DataProperty: signOut
    Characteristics: Functional
    Domain: Accommodation
    Range: date

DataProperty: passengerDetails
    Characteristics: Functional
    Domain: Flight or Vacation
    Range: anyURI

DataProperty: totalAmountToPay
    Characteristics: Functional
    Domain: Reservation
    Range: decimal

DataProperty: numberOfGuests
    Characteristics: Functional
    Domain: Accommodation
    Range: positiveInteger

DataProperty: carrier
    Characteristics: Functional
    Domain: Flight or Vacation
    Range: string

DataProperty: hasIdentification
    Characteristics: Functional
    Domain: Customer
    Range: ID

DataProperty: origin
    Characteristics: Functional
    Domain: Flight or Vacation
    Range: string

DataProperty: departing
    Characteristics: Functional
    Domain: Flight or Vacation
    Range: date

DataProperty: reservationTerms
    Characteristics: Functional
    Domain: Reservation
    Range: anyURI

DataProperty: signIn
    Characteristics: Functional
    Domain: Accommodation
    Range: date

DataProperty: paymentInfo
    Characteristics: Functional
    Domain: Reservation
    Range: anyURI

DataProperty: customerDetails
    Characteristics: Functional
    Domain: Registration or Reservation
    Range: anyURI
```

```
DataProperty: numberOfPassengers
    Characteristics: Functional
    Domain: Flight or Vacation
    Range: positiveInteger

DataProperty: outboundTime
    Characteristics: Functional
    Domain: Flight or Vacation
    Range: time

DataProperty: numberOfRooms
    Characteristics: Functional
    Domain: Accommodation or Vacation
    Range: positiveInteger

DataProperty: rate
    Characteristics: Functional
    Domain: Accommodation or Vacation
    Range: positiveInteger

DataProperty: reservationSummary
    Characteristics: Functional
    Domain: Reservation
    Range: anyURI

DataProperty: guestDetails
    Characteristics: Functional
    Domain: Accommodation
    Range: anyURI

DataProperty: boardingPass
    Characteristics: Functional
    Domain: Reservation
    Range: anyURI
```

Travel Agent's Ontology

In the travel agent's ontology, customers make reservations of flights, hotels or travel packages.

```
Class: Product
    EquivalentTo: Flight or Hotel or Package

DisjointClasses: Flight, Hotel, Package
```

Flights can be round-trip flights or one-way flights:

```
Class: Flight
    EquivalentTo: OneWay or RoundTrip
    SubClassOf: Product
```

Now, the following is the definition of a round-trip flight:

```
Class: RoundTrip
    EquivalentTo: ((flexibleOnDates some owl:Thing)
        or (flexibleOnDates only owl:Nothing))
        and (airlineCompany some string)
        and (from some string)
        and (leavingDate some date)
        and (passengers some positiveInteger)
        and (returnDate some date)
        and (time some time)
        and (to some string)
        and (airlineCompany only string)
        and (from only string)
        and (leavingDate only date)
        and (passengers only positiveInteger)
        and (returnDate only date)
        and (to only string)
```

```
    SubClassOf: Flight
    DisjointWith: OneWay
```

And here we have the definition of a one-way flight:

```
Class: OneWay
    EquivalentTo: ((flexibleOnDates some owl:Thing)
        or (flexibleOnDates only owl:Nothing))
        and (not (returnDate some date))
        and (airlineCompany some string)
        and (from some string)
        and (leavingDate some date)
        and (passengers some positiveInteger)
        and (time some time)
        and (to some string)
        and (airlineCompany only string)
        and (from only string)
        and (leavingDate only date)
        and (passengers only positiveInteger)
        and (to only string)
    SubClassOf: Flight
    DisjointWith: RoundTrip
```

The following are the definitions of a hotel and a travel package:

```
Class: Hotel
    EquivalentTo: (city some string)
        and (lodgers some positiveInteger)
        and (nights some positiveInteger)
        and (register some date)
        and (rooms some positiveInteger)
        and (stars some positiveInteger)
        and (city only string)
        and (lodgers only positiveInteger)
        and (nights only positiveInteger)
        and (register only date)
        and (rooms only positiveInteger)
        and (stars only positiveInteger)
    SubClassOf: Product

Class: Package

EquivalentTo: ((flexibleOnDates some owl:Thing)
        or (flexibleOnDates only owl:Nothing))
        and (airlineCompany some string)
        and (from some string)
        and (leavingDate some date)
        and (passengers some positiveInteger)
        and (returnDate some date)
        and (rooms some positiveInteger)
        and (stars some positiveInteger)
```

```
        and (time some time)
        and (to some string)
        and (airlineCompany only string)
        and (from only string)
        and (leavingDate only date)
        and (passengers only positiveInteger)
        and (returnDate only date)
        and (rooms only positiveInteger)
        and (stars only positiveInteger)
        and (to only string)
    SubClassOf: Product
```

Here we have the definition of a booking:

```
Class: Booking
    EquivalentTo: (((((not (boardingPass some anyURI))
        and (checkIn only owl:Nothing))
        or ((checkIn some owl:Thing)
        and (boardingPass some anyURI)
        and (boardingPass only anyURI)))
        and (hasProduct some Flight)
        and (flightSummary some anyURI)
        and (passengerData some anyURI)
        and (flightSummary only anyURI)
        and (passengerData only anyURI))
        or ((((not (boardingPass some anyURI))
        and (checkIn only owl:Nothing))
        or ((checkIn some owl:Thing)
        and (boardingPass some anyURI)
        and (boardingPass only anyURI)))
        and (hasProduct some Package)
        and (lodgerData some anyURI)
        and (packageSummary some anyURI)
        and (passengerData some anyURI)
        and (lodgerData only anyURI)
        and (packageSummary only anyURI)
        and (passengerData only anyURI))
        or ((hasProduct some Hotel)
        and (lodgerData some anyURI)
        and (lodgerData only anyURI)))
        and (((hasCustomer some NonUser)
        and (hasCustomer only NonUser)
        and (contactInfo some anyURI)
        and (contactInfo only anyURI))
        or ((hasCustomer some User)
        and (hasCustomer only User)))
        and (hasProduct some Product)
        and (hasProduct only Product)
        and (hotelSummary some anyURI)
        and (payingInfo some anyURI)
```

```
        and (rulesAndRestrictions some anyURI)
        and (totalPrice some decimal)
        and (hotelSummary only anyURI)
        and (payingInfo only anyURI)
        and (rulesAndRestrictions only anyURI)
        and (totalPrice only decimal)
```

Customer are to be identified and they can be users or non-users:

```
Class: Customer
    EquivalentTo:
        (hasIdentification some ID)
        and (hasIdentification only ID),
        NonUser or User
Class: User
    SubClassOf: Customer
    DisjointWith: NonUser

Class: NonUser
    EquivalentTo: Customer and (not (User))
    SubClassOf: Customer
    DisjointWith: User
```

Finally, we give the definition of an account:

```
Class: Account
    EquivalentTo:
        (hasCustomer some User)
        and (hasCustomer only User)
        and (contactInfo some anyURI)
        and (contactInfo only anyURI)
```

Additionally, this ontology describes the part of the domain related with searches and their results:

```
Class: Outcome
    EquivalentTo:
        FlightOutcome
        or HotelOutcome
        or PackageOutcome,
         inv(hasOutcome) some CurrentSearch

Class: CurrentSearch
    SubClassOf: Search

Class: Search

Class: HotelOutcome
    SubClassOf:
        Outcome
```

```
Class: FlightOutcome
    SubClassOf:
        Outcome

Class: PackageOutcome
    SubClassOf:
        Outcome

Individual: myFlight
    Types: Flight, owl:Thing

Individual: myHotel
    Types: Hotel, owl:Thing

Individual: myPackage
    Types: Package, owl:Thing
```

The following are the object properties of the customer's ontology:

```
ObjectProperty: flexibleOnDates
    Domain: Flight or Package

ObjectProperty: hasCustomer
    Characteristics: Functional
    Domain: Account or Booking
    Range: Customer

ObjectProperty: hasProduct
    Characteristics: Functional
    Domain: Booking
    Range: Product

ObjectProperty: hasOutcome
    Domain: Search
    Range: Product

ObjectProperty: checkIn
    Characteristics: Functional
    Domain: Booking
```

And here we have the datatype properties:

```
DataProperty: leavingDate
    Characteristics: Functional
    Domain: Flight or Package
    Range: date

DataProperty: returnDate
    Characteristics: Functional
    Domain: Package or RoundTrip
    Range: date

DataProperty: nights
    Characteristics: Functional
    Domain: Hotel
    Range: positiveInteger

DataProperty: lodgers
    Characteristics: Functional
    Domain: Hotel
    Range: positiveInteger

DataProperty: contactInfo
    Characteristics: Functional
    Domain: Account or Booking
    Range: anyURI

DataProperty: rooms
    Characteristics: Functional
    Domain: Hotel or Package
    Range: positiveInteger

DataProperty: airlineCompany
    Characteristics: Functional
    Domain: Flight or Package
    Range: string

DataProperty: stars
    Characteristics: Functional
    Domain: Hotel or Package
    Range: positiveInteger
```

```
DataProperty: packageSummary
    Characteristics: Functional
    Domain: Booking
    Range: anyURI

DataProperty: totalPrice
    Characteristics: Functional
    Domain: Booking
    Range: decimal

DataProperty: city
    Characteristics: Functional
    Domain: Hotel
    Range: string

DataProperty: lodgerData
    Characteristics: Functional
    Domain: Hotel
    Range: anyURI

DataProperty: time
    Domain: Flight or Package
    Range: time

DataProperty: hotelSummary
    Characteristics: Functional
    Domain: Booking
    Range: anyURI

DataProperty: passengerData
    Characteristics: Functional
    Domain: Flight or Package
    Range: anyURI

DataProperty: rulesAndRestrictions
    Characteristics: Functional
    Domain: Booking
    Range: anyURI

DataProperty: from
    Characteristics: Functional
    Domain: Flight or Package
    Range: string

DataProperty: passengers
    Characteristics: Functional
    Domain: Flight or Package
    Range: positiveInteger

DataProperty: register
    Characteristics: Functional
    Domain: Hotel
    Range: date

DataProperty: hasIdentification
    Characteristics: Functional
    Domain: Customer
    Range: ID

DataProperty: to
    Characteristics: Functional
    Domain: Flight or Package
    Range: string

DataProperty: payingInfo
    Characteristics: Functional
    Domain: Booking
    Range: anyURI

DataProperty: boardingPass
    Characteristics: Functional
    Domain: Booking
    Range: anyURI

DataProperty: flightSummary
    Characteristics: Functional
    Domain: Booking
    Range: anyURI
```

Bibliography

[Arcos et al., 2005] Arcos, J. L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J. A., and Sierra, C. (2005). Engineering open evironments with electronic institutions. *Engineering Applications of Artificial Intelligence*, 18(2):191–204.

[Atencia and Schorlemmer, 2006] Atencia, M. and Schorlemmer, M. (2006). Situated semantic alignment. In Dunin-Keplicz, B., Omicini, A., and Padget, J. A., editors, *Proceedings of the 4th European Workshop on Multi-Agent Systems, EUMAS'06, Lisbon, Portugal, December 14-15, 2006*, volume 223 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Atencia and Schorlemmer, 2007] Atencia, M. and Schorlemmer, M. (2007). A formal model for Situated Semantic Alignment. In Durfee, E. H., Yokoo, M., Huhns, M. N., and Shehory, O., editors, *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007. Proceedings*, pages 1270–1277. IFAAMAS.

[Atencia and Schorlemmer, 2008a] Atencia, M. and Schorlemmer, M. (2008a). Formalising Interaction-Situated Semantic Alignment: the communication product. In *Proceedings of the 10th International Symposium on Artificial Intelligence and Mathematics, ISAIM'08, Fort Lauderdale, Florida, USA. January 2-4, 2008.*

[Atencia and Schorlemmer, 2008b] Atencia, M. and Schorlemmer, M. (2008b). I-SSA: Interaction-Situated Semantic Alignment. In Meersman, R. and Tari, Z., editors, *On the Move to Meaningful Internet Systems: OTM 2008. OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part I*, volume 5331 of *Lecture Notes in Computer Science*, pages 445–455. Springer.

[Atencia and Schorlemmer, 2009] Atencia, M. and Schorlemmer, M. (2009). Reaching semantic agreements through interaction. In Paschke, A., Weigand, H., Berhrendt, W., Tochtermann, K., and Pelligrini, T., editors, *Fourth AIS SigPrag International Pragmatic Web Conference Track, ICPW'09, at the Fifth International Conference on Semantic Systems, i-Semantics'09, Graz, Austria, 2-4 September, 2009*, J.UCS Conference Proceedings Series, pages 726–737. Verlag der Technischen Universität Graz.

[Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The description logic handbook. Theory, implementation and applications*. Cambridge University Press.

[Bailin and Truszkowski, 2003] Bailin, S. C. and Truszkowski, W. (2003). Ontology negotiation: how agents can really get to know each other. In Truszkowski, W., Rouff, C., and Hinchey, M., editors, *Innovative Concepts for Agent-Based Systems, First International Workshop on Radical Agent Concepts, WRAC 2002, McLean, VA, USA, January 16-18, 2002 Revised Papers*, volume 2564 of *Lecture Notes in Computer Science*, pages 320–334. Springer.

[Barwise and Seligman, 1997] Barwise, J. and Seligman, J. (1997). *Information flow: the logic of distributed systems*. Cambridge University Press.

[Bench-Capon and Malcolm, 1999] Bench-Capon, T. and Malcolm, G. (1999). Formalising ontologies and their relations. In *Database and Expert Systems Applications. 10th International Conference, DEXA'99 Florence, Italy, August 30 - September 3, 1999 Proceedings*, volume 1677 of *Lecture Notes in Computer Science*, pages 250–259. Springer.

[Bench-Capon et al., 2003] Bench-Capon, T., Malcolm, G., and Shave, M. (2003). Semantics for interoperability - relating ontologies and schemata. In *Database and Expert Systems Applications, 14th International Conference, DEXA 2003, Prague, Czech Republic, September 1-5, 2003, Proceedings*, volume 2736 of *Lectures Notes in Computer Science*, pages 703–712. Springer.

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

[Besana and Robertson, 2007] Besana, P. and Robertson, D. (2007). How service choreography statistics reduce the ontology mapping problem. In Aberer, K., Choi, K.-S., Noy, N. F., Allemang, D., Lee, K.-I., Nixon, L. J. B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P., editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15 2007, Proceedings*, volume 4825 of *Lecture Notes in Computer Science*, pages 44–57. Springer.

[BFO, URL] BFO (URL). <http://www.ifomis.org/bfo>.

[Borgida and Serafini, 2003] Borgida, A. and Serafini, L. (2003). Distributed description logics: assimilating information from peer sources. *Journal on Data Semantics*, I:153–184.

[Bourque et al., 1999] Bourque, P., Dupuis, R., Abran, A., Moore, J. W., and Tripp, L. (1999). The guide to the software engineering body of knowledge. *IEEE Software*, 16(6):35–44.

[Bravo and Velázquez, 2008] Bravo, M. and Velázquez, J. (2008). Discovering pragmatic similarity relations between agent interaction protocols. In Meersman, R., Tari, Z., and Herrero, P., editors, *On the Move to Meaningful Internet Systems: OTM 2008 Workshops, OTM Confederated International Workshops and Posters, ADI, AWeSoMe, COMBEK, EI2N, IWSSA, MONET, OnToContent + QSI, ORM, PerSys, RDDS, SEMELS, and SWWS 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings*, volume 5333 of *Lecture Notes in Computer Science*, pages 128–137. Springer.

[Brickley and Guha, 2004] Brickley, D. and Guha, R. (2004). RDF vocabulary description language 1.0: RDF Schema. Technical report, World Wide Web Consortium (W3C). <http://www.w3.org/tr/rdf-schema/>.

[Cáceres et al., 2006] Cáceres, C., Fernández, A., Ossowski, S., and Vasirani, M. (2006). Agent-based semantic service discovery for healthcare: an organizational approach. *IEEE Intelligent Systems*, 21(6):11–20.

[Carlsson, 2009] Carlsson, M. (2009). SICStus prolog user's manual. Release 4.0.7. Technical report, Swedish Institute of Computer Science, PO Box 1263, SE-164 29 Kitsa, Sweden.

[Castano et al., 2001] Castano, S., de Antonellis, V., and de Capitani di Vimercati, S. (2001). Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):277–297.

[Chaib-Draa and Dignum, 2002] Chaib-Draa, B. and Dignum, F. (2002). Trends in agent communication language. *Computational Intelligence*, 18(2):89–101.

[Ciocoiu and Nau, 2000] Ciocoiu, M. and Nau, D. (2000). Ontology-based semantics. In Cohn, A. G., Giunchiglia, F., and Selman, B., editors, *KR 2000, Principles of Knowledge Representation and Reasoning, Proceedings of the Seventh International Conference, Breckenridge, Colorado, USA, April 11-15, 2000*, pages 539–546. Morgan Kaufmann Publishers Inc.

[Cohen, 1995] Cohen, P. R. (1995). *Empirical methods for Artificial Intelligence*. The MIT Press.

[Corrêa da Silva and Agustí, 2003] Corrêa da Silva, F. S. and Agustí, J. (2003). *Knowledge coordination*. Wiley.

[Cost et al., 2000] Cost, R. S., Chen, Y., Finin, T., Labrou, Y., and Peng, Y. (2000). Using colored petri nets for conversations modelling. In Dignum, F. and Greaves, M., editors, *Issues in agent communication*, volume 1916/2000 of *Lecture Notes in Computer Science*, pages 178–192. Springer.

[Cyc, URL] Cyc (URL). <http://www.opencyc.org/>.

[Dalgaard, 2008] Dalgaard, P. (2008). *Introductory Statistics with R*. Statistics and Computing. Springer, second edition.

[DAML, URL] DAML (URL). <http://www.daml.org>.

[DAML-ONT, URL] DAML-ONT (URL). <http://www.daml.org/2000/10/daml-ont.html>.

[DAML+OIL, URL] DAML+OIL (URL). <http://www.daml.org/2001/03/daml+oil-index.html>.

[De Saeger and Shimojima, 2007] De Saeger, S. and Shimojima, A. (2007). Contextual reasoning in agent systems. In Inoue, K., Satoh, K., and Toni, F., editors, *Computational Logic in Multi-Agent Systems, 7th International Workshop, CLIMA VII, Hakodate, Japan, May 8-9, 2006, Revised Selected and Invited Papers*, volume 4371 of *Lecture Notes in Computer Science*, pages 171–190. Springer.

[Dean and Schreiber, 2004] Dean, M. and Schreiber, G. E. (2004). OWL Web Ontology Language reference, W3C recommendation. Technical report, World Wide Web Consortium (W3C). <http://www.w3.org/TR/owl-ref/>.

[Dell'Erba et al., 2002] Dell'Erba, M., Fodor, O., Ricci, F., and Werthner, H. (2002). Harmonise: a solution for data interoperability. In Monteiro, J. L., Swatman, P. M. C., and Tavares, L. V., editors, *Towards the Knowledge Society: eCommerce, eBusiness and eGovernment. The Second IFIP Conference on E-Commerce, E-Business and E-Government (I3E 2002), October 7-9, 2002, Lisbon, Portugal*, volume 233, pages 433–445.

[Devedzic, 2002] Devedzic, V. (2002). Understanding ontological engineering. *Communications of the ACM*, 45(4):136–144.

[Do and Rahm, 2002] Do, H. H. and Rahm, E. (2002). COMA - a system for flexible combination of schema matching approaches. In *VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China*, pages 610–621. Morgan Kaufmann Publishers.

[Doan et al., 2004] Doan, A., Madhavan, J., Domingos, P., and Halevy, A. Y. (2004). Ontology matching: a machine learning approach. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, chapter 18, pages 385–404. Springer.

[Dunn and Hardegree, 2001] Dunn, J. M. and Hardegree, G. M. (2001). *Algebraic methods in Philosophical Logic*. Oxford University Press.

[Eggan, 1963] Eggan, L. C. (1963). Transition graphs and the star-height of regular events. *Michigan Mathematical Journal*, 10(4):385–397.

[Ehrig and Staab, 2004] Ehrig, M. and Staab, S. (2004). QOM - Quick Ontology Mapping. In McIlraith, S. A., Plexousakis, D., and van Harmelen, F., editors, *The Semantic Web - ISWC 2004: Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 of *Lecture Notes in Computer Science*, pages 683–697. Springer.

[Ehrig and Sure, 2004] Ehrig, M. and Sure, Y. (2004). Ontology mapping - an integrated approach. In Bussler, C., Davies, J., Fensel, D., and Studer, R., editors, *The Semantic Web: Research and Applications, First European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece, May 10-12, 2004, Proceedings*, volume 3053 of *Lecture Notes in Computer Science*, pages 76–91. Springer.

[Euzenat and Shvaiko, 2007] Euzenat, J. and Shvaiko, P. (2007). *Ontology matching*. Springer.

[Euzenat and Valtchev, 2004] Euzenat, J. and Valtchev, P. (2004). Similarity-based ontology alignment in OWL-Lite. In de Mántaras, R. L. and Saitta, L., editors, *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, pages 333–337. IOS Press.

[Fellbaum, 1998] Fellbaum, C., editor (1998). *WordNet: an electronic lexical database*. The MIT Press.

[Fensel et al., 2003] Fensel, D., Hendler, J., Lieberman, H., and Wahlster, W., editors (2003). *Spinning the Semantic Web. Bringing the World Wide Web to its full potential*. The MIT Press.

[Fensel et al., 2001] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., and Patel-Schneider, P. F. (2001). OIL: an ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 12(3):38–45.

[Finkelstein et al., 1994] Finkelstein, A., Gabbay, D. M., Hunter, A., Kramer, J., and Nuseibeh, B. (1994). Inconsistency handling in multperspective specifications. *IEEE Transactions on Software Engineering*, 20(8):569–578.

[Fodor and Werthner, 2005] Fodor, O. and Werthner, H. (2005). Harmonise: a step towards an interoperable e-tourism marketplace. *International Journal of Electronic Commerce*, 9(2):11–39.

[Genesereth and Fikes, 1992] Genesereth, M. R. and Fikes, R. E. (1992). Knowledge interchange format, version 3.0 reference manual. Technical report, Computer Science Department, Stanford University.

[Genesereth and Nilsson, 1987] Genesereth, M. R. and Nilsson, N. J. (1987). *Logical foundations of artificial intelligence*. Morgan Kaufmann publishers, Inc.

[Ghidini and Giunchiglia, 2001] Ghidini, C. and Giunchiglia, F. (2001). Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259.

[Giunchiglia and Shvaiko, 2003] Giunchiglia, F. and Shvaiko, P. (2003). Semantic matching. *The Knowledge Engineering Review*, 18(3):265–280.

[Giunchiglia et al., 2004] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2004). S-Match: an algorithm and an implementation of semantic matching. In Bussler, C., Davies, J., Fensel, D., and Studer, R., editors, *The Semantic Web: Research and Applications, First European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece, May 10-12, 2004, Proceedings*, volume 3053 of *Lecture Notes in Computer Science*, pages 61–75. Springer.

[Giunchiglia et al., 2007] Giunchiglia, F., Yatskevich, M., and Shvaiko, P. (2007). Semantic matching: algorithms and implementation. *Journal on Data Semantics*, IX:1–38.

[GO, URL] GO (URL). <http://www.geneontology.org/>.

[Goguen and Burstall, 1992] Goguen, J. and Burstall, R. (1992). Institutions: abstract model theory for specification and programming. *Journal of the ACM*, 309(1):95–146.

[Goguen and Roşu, 2002] Goguen, J. and Roşu, G. (2002). Institution morphisms. *Formal Aspects of Computing*, 13(3-5):274–307.

[Gómez-Pérez et al., 2004] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2004). *Ontological engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web.* Springer.

[Gruber and Olsen, 1994] Gruber, T. R. and Olsen, G. R. (1994). An ontology for engineering mathematics. In Doyle, J., Sandewall, E., and Torasso, P., editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94), Bonn, Germany, May 24-27, 1994*, pages 258–269. Morgan Kaufmann.

[Grüninger, 2005] Grüninger, M. (2005). Model-theoretic approaches to semantic integration. In Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., and Uschold, M., editors, *Semantic Interoperability and Integration*, volume 04931 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, Germany.

[Guarino, 1998] Guarino, N. (1998). Formal ontology and information systems. In Guarino, N., editor, *Formal Ontology in Information Systems, First International Conference, FOIS, Trento, Italy, June 6-8, 1998, Proceedings*, volume 46 of *Frontiers in Artificial Intelligence and Applications*, pages 3–15. IOS Press.

[Guarino et al., 1994] Guarino, N., Carrara, M., and Giaretta, P. (1994). Formalizing ontological commitments. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'94), Seattle, Washington, USA, July 31 - August 4, 1994*, volume 1, pages 560–567. AAAI Press.

[Guarino and Giaretta, 1995] Guarino, N. and Giaretta, P. (1995). Ontologies and knowledge bases: towards a terminological clarification. In Mars, N. J. I., editor, *Towards Very Large Knowledge Bases, Knowledge Building and Knowledge Sharing*, pages 25–32. IOS Press.

[Hendler and van Harmelen, 2008] Hendler, J. and van Harmelen, F. (2008). The Semantic Web: webizing knowledge representation. In van Harmelen, F., Lifschitz, V., and Porter, B., editors, *Handbook of Knowledge Representation*, chapter 21, pages 821–839. Elsevier.

[Hendler, 2001] Hendler, J. A. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems*, 16(2):30–37.

[Hopcroft and Ullman, 1979] Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to automata theory, languages and computation*. Addison-Wesley Series in Computer Science. Addison-Wesley Publishing Company, Inc.

[Huhns and Stephens, 1999] Huhns, M. N. and Stephens, L. M. (1999). Multiagent systems and societies of agents. In Weiss, G., editor, *Multiagent systems. A modern approach to distributed artificial intelligence*, chapter 2, pages 79–120. The MIT Press.

[IFF, URL] IFF (URL). <http://suo.ieee.org/iff/>.

[Jarrar and Meersman, 2002] Jarrar, M. and Meersman, R. (2002). Formal ontology engineering in the DOGMA approach. In Meersman, R. and Tari, Z., editors, *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE. Confederated International Conferences CoopIS, DOA, and ODBASE 2002 Proceedings*, volume 2519 of *Lecture Notes in Computer Science*, pages 1238–1254. Springer.

[Jian et al., 2005] Jian, N., Hu, W., Cheng, G., and Qu, Y. (2005). Falcon-AO: aligning ontologies with Falcon. In Ashpole, B., Ehrig, M., Euzenat, J., and Stuckenschmidt, H., editors, *Integrating Ontologies 2005, Proceedings of the K-CAP 2005, Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005*, volume 156 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Kalfoglou and Schorlemmer, 2003a] Kalfoglou, Y. and Schorlemmer, M. (2003a). IF-Map: an ontology-mapping method based on Information-Flow Theory. *Journal on Data Semantics*, I:98–127.

[Kalfoglou and Schorlemmer, 2003b] Kalfoglou, Y. and Schorlemmer, M. (2003b). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31.

[Kalfoglou and Schorlemmer, 2010] Kalfoglou, Y. and Schorlemmer, M. (2010). The information-flow approach for ontology-based semantic integration. In Poli, R., Healey, M., and Kameas, A., editors, *Theory and Applications of Ontology: Computer Applications*, volume 2, chapter 4. Springer.

[Kent, 2005] Kent, R. E. (2005). Semantic integration in the Information Flow framework. In Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., and Uschold, M., editors, *Semantic Interoperability and Integration*, volume 04391 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, Germany.

[Kutz et al., 2004] Kutz, O., Lutz, C., Wolter, F., and Zakharyaschev, M. (2004). $\mathcal{E}$–connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73.

[Kutz et al., 2008] Kutz, O., Mossakowski, T., and Codescu, M. (2008). Shapes of alignments - Construction, combination and computation. In Sattler, U. and Tamilin, A., editors, *Proceedings of the Workshop on Ontologies: Reasoning and Modularity (WORM-08), Tenerife, Spain, June 2, 2008*, volume 348. CEUR-Workshop Proceedings.

[Labrou, 1996] Labrou, Y. (1996). *Semantics for an agent communication language*. PhD thesis, Computer Science and Electrical Engineering Department (CSEE), University of Maryland Graduate School, Baltimore, Maryland, USA.

[Labrou and Finin, 1997] Labrou, Y. and Finin, T. (1997). Semantics and conversations for an agent communication language. In Pollack, M. E., editor, *Proceedings of the 15th International Joint Conference on Artificial Intelligence, IJCAI'97*, pages 584–591, Nagoya, Japan. Morgan Kaufmann publishers Inc. : San Mateo, CA, USA.

[Laera et al., 2007] Laera, L., Blacoe, I., Tamma, V., Payne, T. R., Euzenat, J., and Bench-Capon, T. (2007). Argumentation over ontology correspondences in MAS. In Durfee, E. H., Yokoo, M., Huhns, M. N., and Shehory, O., editors, *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*, pages 1285–1292. IFAAMAS.

[Lassila and Swick, 1999] Lassila, O. and Swick, R. (1999). Resource Description Framework (RDF) model and syntax specification. Technical report, World Wide Web Consortium (W3C). <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>.

[Lenat, 1995] Lenat, D. B. (1995). CYC: a large-scale investment in knowledge infrastructure. *Communication of the ACM*, 38(11):33–38.

[Lenat et al., 1990] Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). Cyc: towards programs with common sense. *Communication of the ACM*, 33(8):30–49.

[López et al., 2006] López, V., Sabou, M., and Motta, E. (2006). PowerMap: mapping the real Semantic Web on the fly. In Cruz, I. F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L., editors, *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC'06, Athens, GA, USA, November 5-9, 2006. Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 414–427. Springer.

[Mac Lane, 1998] Mac Lane, S. (1998). *Categories for the working mathematician*. Springer, second edition.

[Madhavan et al., 2001] Madhavan, J., Bernstein, P. A., and Rahm, E. (2001). Generic schema matching with Cupid. In Apers, P. M. G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., and Snodgrass, R. T., editors, *VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases, September 11-14, 2001, Roma, Italy*, pages 49–58. Morgan Kaufmann Publishers Inc.

[Maedche, 2002] Maedche, A. (2002). *Ontology learning for the Semantic Web.* Kluwer Academic Publishers.

[Marchese et al., 2008] Marchese, M., Vaccari, L., Trecarichi, G., Osman, N., and McNeill, F. (2008). Interaction models to support peer coordination in crisis management. In Fiedrich, F. and de Walle, B. V., editors, *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management, ISCRAM'08, Washington, DC, USA, May 4-7, 2008*, pages 230–242.

[McCarthy, 1980] McCarthy, J. (1980). Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39.

[McNeill and Bundy, 2007] McNeill, F. and Bundy, A. (2007). Dynamic, automatic, first-order ontology repair by diagnosis of failed plan execution. *International Journal of Semantic Web and Information Systems*, 3(3):1–35.

[Melnik et al., 2002] Melnik, S., Garcia-Molina, H., and Rahm, E. (2002). Similarity Flooding: a versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering, 26 February - 1 March, 2002, San Jose, CA, USA*, pages 117–128. IEEE Computer Society.

[Menzel, 2002] Menzel, C. (2002). Ontology theory. In Euzenat, J., Gomez-Perez, A., Guarino, N., and Stuckenschmidt, H., editors, *Proceedings of the ECAI-02 Workshop on Ontologies and Semantic Interoperability, Lyon, France, July 22, 2002.*, volume 64 of *CEUR Workshop Proceedings*.

[Menzel, 2005] Menzel, C. (2005). Basic semantic integration. In Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., and Uschold, M., editors, *Semantic Interoperability and Integration*, volume 04391 of *Dagstuhl Seminar Proceedings*. IBFI, Schloss Dagstuhl, Germany.

[Miller, 1995] Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

[Mondeca, URL] Mondeca (URL). <http://www.mondeca.com>.

[Niles and Pease, 2001] Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In Welty, C. and Smith, B., editors, *The 2nd International Conference on Formal Ontology in Information Systems (FOIS'01), Ogunquit, Maine, USA, October 17-19, 2001, Proceedings*, volume 2001 of *ACM Press book series*, pages 2–9. ACM Press.

[Nodine and Unruh, 1998] Nodine, M. H. and Unruh, A. (1998). Facilitating open communication in agent systems: the InfoSleuth infrastructure. In Singh, M. P., Rao, A. S., and Wooldridge, M., editors, *Intelligent Agents IV, Agent Theories, Architectures, and Languages, 4th International Workshop, ATAL'97, Providence, Rhode Island, USA, July 24-26, 1997, Proceedings*, volume 1365 of *Lecture Notes in Computer Science*, pages 281–295. Springer.

[Noy and McGuinness, URL] Noy, N. F. and McGuinness, D. L. (URL). Ontology development 101: a guide to creating your first ontology. <http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101 -noy-mcguinness.html>.

[O'Brien and Nicol, 1998] O'Brien, P. D. and Nicol, R. C. (1998). FIPA – towards a standard for software agents. *BT Technology Journal*, 16(3):51–59.

[Øhrstrøm et al., 2005] Øhrstrøm, P., Andersen, J., and Schärfe, H. (2005). What has happened to ontology. In Dau, F., Mugnier, M.-L., and Stumme, G., editors, *Conceptual Structures: Common Semantics for Sharing Knowledge, 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17-22, 2005, Proceedings*, volume 3596 of *Lecture Notes in Computer Science*, pages 425–438. Springer.

[OTA, URL] OTA (URL). <http://www.opentravel.org/>.

[Rahm and Bernstein, 2001] Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350.

[Robertson, 2004] Robertson, D. (2004). Multi-agent coordination as distributed logic programming. In Demoen, B. and Lifschitz, V., editors, *Logic Programming. 20th International Conference, ICLP'04, Saint-Malo, France, September 6-10 2004. Proceedings*, volume 3132 of *Lecture Notes in Computer Science*, pages 416–430. Springer.

[Rovatsos, 2007] Rovatsos, M. (2007). Dynamic semantics for agent communication languages. In Durfee, E. H., Yokoo, M., Huhns, M. N., and Shehory, O., editors, *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007, Proceedings*. IFAAMAS.

[Schorlemmer and Atencia, 2009] Schorlemmer, M. and Atencia, M. (2009). Taking interaction ontologically prior to meaning. In Vallverdú, J., editor, *Proceedings of the 7th European Conference on Computing and Philosophy, E-CAP'09, Bellaterra (Barcelona), Spain, July 2-4 2009.*, pages 160–161.

[Schorlemmer et al., 2006] Schorlemmer, M., Atencia, M., Bundy, A., Giunchiglia, F., López, V., and McNeill, F. (2006). OpenKnowledge deliverable D3.2. Ontology handling framework. A formal conceptual framework for semantic matching, alignment, and refinement in P2P information systems. Technical report, School of Informatics, The University of Edinburgh.

[Schorlemmer and Kalfoglou, 2005] Schorlemmer, M. and Kalfoglou, Y. (2005). Progressive ontology alignment for meaning coordination: an information-theoretic foundation. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., and Wooldridge, M., editors, *4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 737–744. ACM Press.

[Schorlemmer and Kalfoglou, 2008] Schorlemmer, M. and Kalfoglou, Y. (2008). Institutionalising ontology-based semantic integration. *Applied Ontology*, 3(3):131–150.

[Sowa, 2000] Sowa, J. F. (2000). *Knowledge representation. Logical, philosophical and computational foundations*. Brooks Cole Publishing Co.

[Steels, 1998] Steels, L. (1998). The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194.

[SUMO, URL] SUMO (URL). <http://suo.ieee.org/>.

[SWRC, URL] SWRC (URL). <http://www.ontoware.org/>.

[TAGA, URL] TAGA (URL). <http://taga.sourceforge.net/>.

[UMLS, URL] UMLS (URL). <http://www.nlm.nih.gov/research/umls/>.

[UNSPSC, URL] UNSPSC (URL). <http://www.unspsc.org/>.

[Uschold and Grüninger, 1996] Uschold, M. and Grüninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155.

[Uschold et al., 1998] Uschold, M., King, M., Moralee, S., and Zorgios, Y. (1998). The Enterprise Ontology. *The Knowledge Engineering Review*, 13(1):31–89.

[van Diggelen et al., 2006] van Diggelen, J., Beun, R.-J., Dignum, F., van Eijk, R. M., and Meyer, J.-J. (2006). ANEMONE: an effective minimal ontology negotiation environment. In Nakashima, H., Wellman, M. P., Weiss, G., and Stone, P., editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 899–906. ACM Press.

[van Eijk et al., 2001] van Eijk, R. M., de Boer, F. S., van der Hoek, W., and Meyer, J.-J. C. (2001). On dynamically generated ontology translators in agent communication. *International Journal of Intelligent Systems*, 16(5):587–607.

[van Harmelen and Antoniou, 2008] van Harmelen, F. and Antoniou, G. (2008). *A Semantic Web primer*. The MIT Press, second edition.

[van Noord, 1996] van Noord, G. (1996). FSA utilities: a toolbox to manipulate finite-state automata. In Raymond, D., Wood, D., and Yu, S., editors, *Automata Implementation. First International Workshop on Implementing Automata, WIA'96, London, Canada, August 29-31, 1996, Revised Papers*, volume 1260 of *Lecture Notes in Computer Science*, pages 87–108. Springer.

[Vukmirovic et al., 2007] Vukmirovic, M., Szymczak, M., Ganzha, M., and Paprzycki, M. (2007). Designing new ways for selling airline tickets. *Informatica*, 31(1):93–104.

[Wang and Gasser, 2002] Wang, J. and Gasser, L. (2002). Mutual online concept learning for multiple agents. In Castelfranchi, C. and Johnson, W. L., editors, *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings*, pages 362–369. ACM Press.

[WebOnt, URL] WebOnt (URL). <http://www.w3.org/2001/sw/webont/>.

[Werthner and Klein, 1999] Werthner, H. and Klein, S. (1999). *Information technology and tourism: a challenging relationship*. Springer-Verlag Wien New York.

[Werthner and Ricci, 2004] Werthner, H. and Ricci, F. (2004). E-commerce and tourism. *Communications of the ACM*, 47(12):101–105.

[Wierzbicka, 1989] Wierzbicka, A. (1989). *English speech act verbs: a semantic dictionary*. Academic Press.

[Wiesman et al., 2002] Wiesman, F., Roos, N., and Vogt, P. (2002). Automatic ontology mapping for agent communication. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings*, pages 563–564. ACM Press.

[Wittgenstein, 1967] Wittgenstein, L. (1967). *Philosophische untersuchungen*. Suhrkamp Verlag.

[WordNet, URL] WordNet (URL). <http://wordnet.princeton.edu/>.

[Zimmermann et al., 2006] Zimmermann, A., Krötzsch, M., Euzenat, J., and Hitzler, P. (2006). Formalizing ontology alignment and its operations with category theory. In Bennett, B. and Fellbaum, C., editors, *Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, FOIS 2006, Baltimore, Maryland, USA, November 9-11, 2006*, volume 150 of *Frontiers in Artificial Intelligence and Applications*, pages 277–288. IOS Press.

Monografies de l'Institut d'Investigació en Intel·ligència Artificial

Núm. 1	J. Puyol, *MILORD II: A Language for Knowledge–Based Systems.*
Núm. 2	J. Levy, *The Calculus of Refinements, a Formal Specification Model Based on Inclusions.*
Núm. 3	Ll. Vila, *On Temporal Representation and Reasoning in Knowledge–Based Systems.*
Núm. 4	M. Domingo, *An Expert System Architecture for Identification in Biology.*
Núm. 5	E. Armengol, *A Framework for Integrating Learning and Problem Solving.*
Núm. 6	J. Ll. Arcos, *The Noos Representation Language.*
Núm. 7	J. Larrosa, *Algorithms and Heuristics for Total and Partial Constraint Satisfaction.*
Núm. 8	P. Noriega, *Agent Mediated Auctions: The Fishmarket Metaphor.*
Núm. 9	F. Manyà, *Proof Procedures for Multiple-Valued Propositional Logics.*
Núm. 10	W. M. Schorlemmer, *On Specifying and Reasoning with Special Relations.*
Núm. 11	M. López-Sánchez, *Approaches to Map Generation by means of Collaborative Autonomous Robots.*
Núm. 12	D. Robertson, *Pragmatics in the Synthesis of Logic Programs.*
Núm. 13	P. Faratin, *Automated Service Negotiation between Autonomous Computational Agents.*
Núm. 14	J. A. Rodríguez, *On the Design and Construction of Agent-mediated Electronis Institutions .*
Núm. 15	T. Alsinet, *Logic Programming with Fuzzy Unification and Imprecise Constants: Possibilistic Semantics and Automated Deduction.*
Núm. 16	A. Zapico, *On Axiomatic Foundations for Qualitative Decision Theory - A Possibilistic Approach.*
Núm. 17	A. Valls, *ClusDM: A multiple criteria decision method for heterogeneous data sets.*
Núm. 18	D. Busquets, *A Multiagent Approach to Qualitative Navigation in Robotics.*
Núm. 19	M. Esteva, *Electronic Institutions: from specification to development.*
Núm. 20	J. Sabater, *Trust and Reputation for Agent Societies.*

Núm. 21 J. Cerquides, *Improving Algorithms for Learning Bayesian Network Classifiers.*
Núm. 22 M. Villaret, *On Some Variants of Second-Order Unification.*
Núm. 23 M. Gómez, *Open, Reusable and Configurable Multi-Agent Systems: A Knowledge Modelling Approach.*
Núm. 24 S. Ramchurn, *Multi-Agent Negotiation Using Trust and Persuasion.*
Núm. 25 S. Ontañón, *Ensemble Case-Based Learning for Multi-Agent Systems.*
Núm. 26 M. Sánchez, *Contributions to Search and Inference Algorithms for CSP and Weighted CSP.*
Núm. 27 C. Noguera, *Algebraic Study of Axiomatic Extensions of Triangular Norm Based Fuzzy Logics.*
Núm. 28 E. Marchioni, *Functional Definability Issues in Logics Based on Triangular Norms.*
Núm. 29 M. Grachten, *Expressivity-Aware Tempo Transformations of Music Performances Using Case Based Reasoning.*
Núm. 30 I. Brito, *Distributed Constraint Satisfaction.*
Núm. 31 E. Altamirano, *On Non-clausal Horn-like Satisfiability Problems.*
Núm. 32 A. Giovannucci, *Computationally Manageable Combinatorial Auctions for Supply Chain Automation.*
Núm. 33 R. Ros, *Action Selection in Cooperative Robot Soccer using Case-Based Reasoning.*
Núm. 34 A. García-Cerdaña, *On some Implication-free Fragments of Substructural and Fuzzy Logics.*
Núm. 35 A. García-Camino, *Normative Regulation of Open Multi-agent Systems.*
Núm. 36 A. Ramisa Ayats, *Localization and Object Recognition for Mobile Robots.*
Núm. 37 C.G. Baccigalupo, *Poolcasting: an intelligent technique to customise music programmes for their audience.*
Núm. 38 J. Planes, *Design and Implementation of Exact MAX-SAT Solvers.*
Núm. 39 A. Bogdanovych, *Virtual Institutions.*
Núm. 40 J. Nin, *Contributions to Record Linkage for Disclosure Risk Assessment.*
Núm. 41 J. Argelich Romà, *Max-SAT Formalisms with Hard and Soft Constraints.*
Núm. 42 A. Casali, *On Intentional and Social Agents with Graded Attitudes.*
Núm. 43 A. Perreau de Pinnick Bas, *Decentralised Enforcement in Multiagent Networks.*
Núm. 44 I. Pinyol Catadau, *Milking the Reputation Cow: Argumentation, Reasoning and Cognitive Agents.*
Núm. 45 S. Joseph, *Coherence-based Computational Agency.*
Núm. 46 M. Atencia, *Semantic Alignment in the Context of Agent Interaction.*