

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher

UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by **Novática** (<http://www.es/novatica/>), journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**

UPGRADE was created in October 2000 by CEPIS and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for UPENET (UPGRADE European NETWORK), the network of CEPIS member societies' publications, that currently includes the following ones:

- **Informatik-Spektrum**, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- **ITNOW**, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
- **Novática**, journal from the Spanish CEPIS society ATI
- **OCG Journal**, journal from the Austrian CEPIS society OCG
- **Pliroforiki**, journal from the Cyprus CEPIS society CCS
- **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

Editorial Team

Chief Editor: Llorenç Pagés-Casas, Spain, pages@ati.es

Associate Editors:

François Louis Nicolet, Switzerland, nicolet@acm.org

Roberto Carniel, Italy, rcarniel@dgt.uniud.it

Zakaria Maamar, Arab Emirates, Zakaria.Maamar@zu.ac.ae

Soraya Kouadri Mostéfaoui, Switzerland, soraya.kouadrimostefaoui@gmail.com

Rafael Fernández Calvo, Spain, rfcvalvo@ati.es

Editorial Board

Prof. Wolfried Stucky, CEPIS Former President

Prof. Nello Scarabottolo, CEPIS Vice President

Fernando Piera Gómez and

Llorenç Pagés-Casas, ATI (Spain)

François Louis Nicolet, SI (Switzerland)

Roberto Carniel, ALSI - Tecnoteca (Italy)

UPENET Advisory Board

Hermann Engesser (Informatik-Spektrum, Germany and Switzerland)

Brian Runciman (ITNOW, United Kingdom)

Franco Filippazzi (Mondo Digitale, Italy)

Llorenç Pagés-Casas (Novática, Spain)

Veith Risak (OCG Journal, Austria)

Panicos Masouras (Pliroforiki, Cyprus)

Andrzej Marciniak (Pro Dialog, Poland)

Rafael Fernández Calvo (Coordination)

English Language Editors: Mike Andersson, Richard Butchart, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Michael Hird, Jim Holder, Alasdair MacLeod, Pat Moody, Adam David Moss, Phil Parkin, Brian Robson

Cover page designed by RFCalvo, © Rafael Fernández Calvo 2006

Layout Design: François Louis Nicolet

Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç.Pagés-Casas@ati.es

Advertising correspondence: novatica@ati.es

UPGRADE Newslist available at

<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

Copyright

© Novática 2006 (for the monograph)

© CEPIS 2006 (for the sections MOSAIC and UPENET)

All rights reserved under otherwise stated. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, contact the Editorial Team

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (December 2006) "Open Document Format (ODF)"

(The full schedule of UPGRADE
is available at our website)



The European Journal for the Informatics Professional
<http://www.upgrade-cepis.org>

Vol. VII, issue No. 5, October 2006

Monograph: Web Services (published jointly with Novática*)

Guest Editors: *José Carlos del Arco-Prieto, Jesús Arias-Fisteus, Óscar Corcho-García, Jorge Cardoso*

- 2 Presentation. Web Services from An Industrial and An Academic Perspective — *Jose Carlos del Arco-Prieto, Jesús Arias-Fisteus, Óscar Corcho-García and Jorge Cardoso*
- 5 Web Services: Introduction and State of The Art — *Óscar Corcho-García, José Carlos del Arco-Prieto and Jesús Arias-Fisteus*
- 11 W3C and Web Services Standardization — *Martín Álvarez-Espinar*
- 16 Integration and Interoperability Experiences in Healthcare — *Antonio García-Landeira*
- 21 Model-Driven Extra-Functional Property Development for Web Services: A Case Study from The Service and Client Side Perspectives — *Guadalupe Ortiz-Bellot and Juan Hernández-Núñez*
- 29 Towards an Automated Trading Process — *Pablo Fernandez-Montes, Manuel Resinas-Arias de Reyna and Rafael Corchuelo-Gil*
- 34 Semantic Web Services with WSMO — *Holger Lausen, Jos de Bruijn, Uwe Keller and Rubén Lara*
- 38 Towards Semantic Service Selection for B2B Integration — *Andreas Friesen and Kioumars Namiri*
- 46 Leveraging E-Marketplace Models for Web Service-Based Application Development — *Abraham Nieva de la Hidalga, Liping Zhao and Pedro R. Falcone-Sampaio*

UPENET (UPGRADE European NETWORK)

- 52 From **Novática** (ATI, Spain)
Open Standards
Open Standards, A Renewed Impulse: The Case of Open Document Format — *Miguel A. Amutio-Gómez*
- 59 From **Mondo Digitale** (AICA, Italy)
ICT Security
Critical Information Infrastructure Protection — *Sandro Bologna, Roberto Setola and Salvatore Tucci*
- 68 From **ITNow** (BCS, United Kingdom)
Informatics Profession
Professionalizing IT — *Peter Skyte*

CEPIS NEWS

- 70 Harmonise Project: Presentation

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Towards an Automated Trading Process

Pablo Fernandez-Montes, Manuel Resinas-Arias de Reyna and Rafael Corchuelo-Gil

Many software companies are using SOA (Service-oriented Architectures) as the cornerstone of their business activities. In this context, automated service provision based on the creation of service agreements is gaining importance in both cross-organizational and intra-organizational scenarios because they provide important benefits to both the service consumer and the service provider. On the one hand, service agreements provide consumers with guarantees about how a service will be provided. On the other hand, service agreements allow the providers to deploy automated service provision based on the agreements they have made with their customers. Service agreements are a prominent research field in both academia and industry. In this article, we focus on the service trading process, which is the process of locating, selecting, negotiating, and creating service agreements. This process can be applied to a variety of scenarios and therefore their requirements are also very different.

Keywords: Automated Negotiation, Automated Provision of Services, Service Agreements, Service Trading, SLAs.

1 Introduction

As SOA (*Service-oriented Architectures*) has evolved into a mature paradigm, new challenges have appeared on the horizon. The current standardized stack of protocols make up a solid infrastructure for interoperability. However, in order to meet new business possibilities an extension is needed. In particular, the automatic provision of services is emerging as a promising field that could lead to a new generation of enterprises that adapt "on demand" to rapid changes in their business environment.

In this article, we address a core element of the automatic provision of services: the creation of agreements that describe the rights and obligations of the service consumer and the service provider during the transaction. The terms of the agreement could refer to either functional features (such as the interface of the service) or non-functional features (such as the amount that should be paid for the service). The agreement should also be considered as a first class element [8] of the IT infrastructure of both service consumer and provider organizations.

Generally speaking, service orientation in software development would seem to provide the perfect scenario for short term agreements that can be dynamically created to meet an enterprise's business needs at any given moment (e.g. the cheapest service provider). Automating the creation and management of agreements, so that the human participation in the process is reduced to a minimum, brings benefits such as cutting the cost of reaching an agreement, increasing the speed of the contracting process, and allowing providers to deploy an automated provision of services based on their agreements with customers, leading to a better rationalization [7] of the usage of its resources. Leading on from the above, from the customer's point of view it would be interesting to be able to negotiate the agreement terms in order to obtain better offers. Such a negotiation

process could also be used by the provider to fine tune the final agreement and make concessions or restrictions in order to optimize the current usage level of its resources. Additionally, given that such offers would be closely bound to the resource status at each moment, the decision making infrastructure should also take this information into account as a first level element before entering into new commitments with a customer. However, at this point, it is important to highlight that automation should be understood within the restricted context of the business world in which certain critical strategic decisions can never completely avoid human intervention.

The automatic creation of agreements requires the combined involvement of several research fields to create a holistic approach. This means, among other things: formal lan-

Authors

Pablo Fernandez-Montes is a Lecturer in Computer Science with the Department of Computer Languages and Systems of the *Universidad de Sevilla* (Spain) as part of the Research Group on Distributed Systems. He is also a PhD candidate whose research is focused on automated contract management and service level agreements. < pablofm@us.es >.

Manuel Resinas-Arias de Reyna is a Lecturer in the Department of Computer Languages and Systems of the *Universidad de Sevilla*. In 2004, he obtained an MSc in Computer Science. He is a member of The Distributed Group at the *Universidad de Sevilla* and he is now working on his PhD. His main research interest is the automated negotiation of service agreements in service-oriented scenarios. < resinas@us.es >.

Rafael Corchuelo-Gil is a Reader in Computer Science who is with the Department of Computer Languages and Systems of the *Universidad de Sevilla*, Spain. He received his PhD from this University and he has led its Research Group on Distributed Systems since 1997. His main research interests include instrumentation techniques to support the multiparty interaction model and knowledge extraction. < corchu@us.es >.

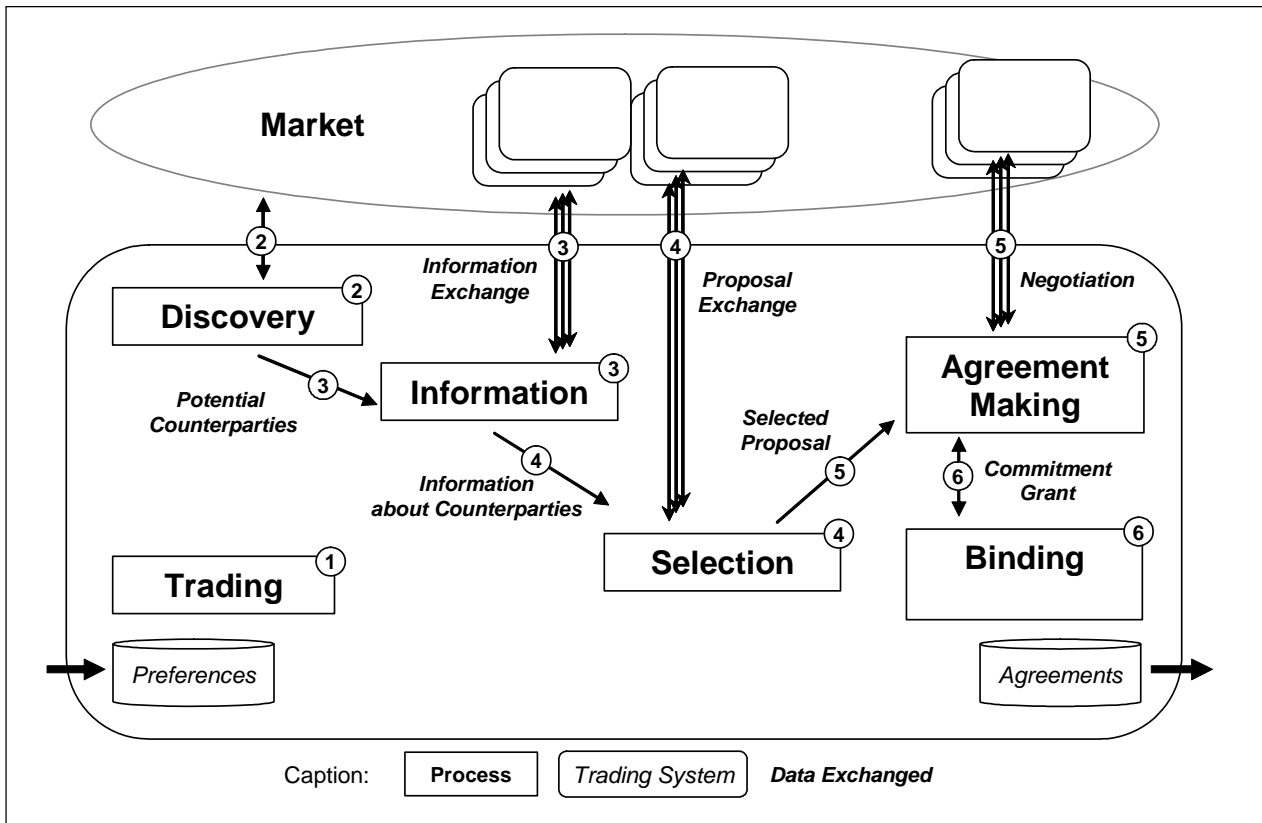


Figure 1: Process View.

languages for the expression of requests, offers, and agreements; the search and discovery of potential partners; the evaluation of agreement proposals; decision making algorithms and strategies; provision capacity estimation; agreement negotiation; or agreement signing.

The steps of an automated contracting process have already been identified in the relevant literature [6]: The first step involves the creation of the offer by the service provider and the analysis of its functional and non-functional requirements by the consumer. This step may be considered as a preliminary stage of the contracting process. In the contracting process itself, the first step (the information phase) is to match service providers with potential consumers and vice versa. In the negotiation phase, a service agreement is created between service consumer and service provider. During the deployment phase, both service provider and service consumer set up a deployment plan to meet all the terms established in the service agreement. Finally, the fulfilment phase involves the fulfilment of the obligations established in the agreement and the monitoring of the entire process in order to ensure that both parties observe the agreement properly.

In this article, we focus on the establishment of the agreement as a subprocess of the more general contracting process. Specifically, this subprocess can be broken down into four different stages: the location, selection, negotiation, and creation of the agreement. We refer to this as the service trading process. In the rest of the paper we outline the vari-

ous challenges involved in each stage of the service trading process and review the most important proposals.

2 Trading Process

The complexity of the trading process lies in the fact that it can vary greatly depending on the scenario: how the parties (service consumers and service providers) discover each other or how each party negotiates an agreement offer. Also, time constraints may vary from one scenario to another depending on the type of parties involved: In some scenarios with short fixed timeframes in which to reach an agreement, it may be necessary to carry out a quick and efficient (but not optimal) selection of the agreement from a set of proposals; on the other hand, with more relaxed timeframes the trading process can be more focused on developing complex negotiations to achieve an optimal agreement.

To address the variable nature of the trading process, we propose the *Trading protocol* concept as a means of orchestrating the various elements in a system while switching the choreography of the process between different systems to ensure an interaction that respects the trading process requirements of all parties. In order to clarify the idea of *trading protocol* perhaps we should look at a real-life example: A public auction in which an institution is looking for a service provider and devises a trading protocol that consists of the following stages: *the announcement of the auction, a deadline for the submission of proposals, a pe-*

riod of resolution and, finally, the communication of results. At an abstract level, the trading protocol is defined as a set of stages (e.g. "announcement, proposal submission, negotiation, resolution, etc...") cross-linked according to some time constraints and bounded to some choreographies. In general, these elements are expressed in two sets of constraints:

- A set of potential choreographies for each stage. In some situations, this means a list of optional choreographies but, in other cases, a unique choreography can be established at a given stage. An example of this type of constraint could be "In the Negotiation stage, auction or bargaining protocols are allowed" (i.e. choreographies implementing these protocols).
- Time restrictions specify a set of constraints about the life-cycle of the trading process. This restrictions could vary from simple fixed temporal points (e.g. "The trading process should end by March 14 at 14h") or can be complex relationships between the durations of some stages (e.g. "The information stage should start in the middle of the discovery stage").

From a structural point of view, the whole trading process (see Figure 1) is composed of six subprocesses (depicted as squares) that interact to reach a specific agreement. The fulfilment of the set of processes make up trading systems (depicted as rounded squares); at an abstract level, these systems represent different parties in the trading process: i.e. some of the trading systems would be acting on behalf of service consumers while others would act on behalf of the service provider. It is important to stress that this abstraction does not imply a specific deployment in a possible implementation of the concrete system but only creates a separation of concerns at a conceptual level.

Following, we identify the behaviour of the whole process (Figure 1) taking into account the different stages involved:

1. The main goal of the trading management process is to create and specify the trading protocol that will regulate how the whole process is carried out; that is, it establishes the timing for when to start the search for parties, when to submit offers, when to wait for responses, when a negotiation must start, or when a binding offer can be sent. Thus, this process focuses on the global behaviour from a temporal point of view. Its goal is the coordination of the rest of the processes so as to implement a trading protocol.

2. The discovery process is concerned with locating potential parties requesting (or supplying) a service that other

party provides (or needs). Complementarily, this process is responsible for accessing the market and disseminating the events generated by the internal active trading processes in the system. In this context, the market should be seen both as an abstract concept that includes the set of organizations looking for business relationships, and the infrastructure that acts as a facilitator for the communication between them. In [12], an analysis of the discovery problem for web services is discussed. Finally it is important to point out that the discovery process can locate potential providers or consumers according to a number of functional or non-functional requirements.

3. Discovered candidates are then passed to the information process in order to gather detailed information about the characteristics and preferences of each potential party. The aim of this process is to manage public information about user preferences and the potential candidates found during the discovery process. The amount and type of information collected from each candidate may be different; however, at a conceptual level the information should include, as a minimum, the publicly available characteristics of the service requested/supplied. In addition, some information can be harvested from external sources (e.g. information about the reputation of the candidate).

4. The data harvested in the information process is used by the selection process to create and select a set of promising agreement proposals with other parties. The aim of this process is to choose a set of candidate parties with whom a negotiation process can be started or to whom an agreement proposal can be submitted. The selection starts with a set of information about potential parties from several sources: information provided during the information process after an active search, agreement proposals received from other parties. In [3] a selection based on constraint programming is described.

5. The instructions about proposal handling are delegated to the agreement making process which is responsible for actually negotiating or proposing the agreement to the other party and for creating and signing the final agreement. Thus, the aim of agreement making is to provide a mechanism to create agreements, possibly through an automated negotiation process, that are acceptable to all parties involved. The result of the process is therefore an agreement that specifies the terms under which the service is to be executed. [10] shows a conceptual framework for automated negotiations of agreements. An example of a simple agreement making protocol is described in [1].

Discovery	Information	Agreement Making	Trading
UDDI	WS-MetadataExchange	WS-Agreement	WS-CDL
WS-Notification	WS-InspectionLanguage	FIPA Protocols	BPEL
WS-Addressing	WS-Agreement		

Table 1: Related Standards.

6. During this procedure, the agreement making process interacts with the binding process by asking for approval to make or accept binding offers. The main concern is to determine when a binding offer must be submitted and whether a binding offer that has been received should be accepted. This process must also establish when these decisions are to be made. For example, one option is to make the decision as the offers are received; another possibility is to make the decisions at some specific points in time that have been previously decided. These points may be dynamically selected, depending on changing conditions of the environment such as the frequency of arrival of offers, or statically determined based on time constraints imposed by the *trading protocol*, or a combination of the two. An interesting infrastructure can be found in [7] that carries out decision-making processes based on the agreements available and the amount of resources available to the service provider.

3 Architectures and Technologies

A number of abstract architectures have been proposed [2][4][5] which carry out some subset of the stages in the service trading process. An analysis [9] of the trading process support in these architectures leads us to several conclusions: (i) the discovery process is well supported by the majority of architectures, most of which provide for knowledge adaptation; (ii) these architectures do not generally include elements to support advanced decision-making; (iii) there is little support for the advanced features of service trading such as the decommitment of agreements or trading protocols. Due to these shortcomings, some complex service trading scenarios cannot be fully automated.

Several standards have emerged to enrich the basic web service stack. Table 1 shows the standards that can be used in the various subprocesses.

With regard to the discovery process, there are three specifications that can be used to implement its requirements: (i) UDDI (*Universal Description, Discovery and Integration*) [13] can be used as a flexible repository that can be used to store the access points of elements and the taxonomies used by the discovery process. (ii) WS-Notification can be used to subscribe and broker notification events. (iii) Lastly, WS-Addressing [13] provides a specification of the references/locations of web services by means of a standardization of the concept of endpoint references.

There are a number of standards that deal with the exchange of service descriptions, from both a functional and a non-functional point of view and they can be used in the implementation of the information process. For instance: WS-MetadataExchange [13] or WS-InspectionLanguage. Alternatively, WS-Agreement [1] uses a template-driven procedure, and those templates can be seen as a mean of expressing the preferences of a given party.

The most significant specification that covers most aspects included in the agreement making process is WS-Agreement [1]. It allows us to specify the structure of an agreement document, so that it must be used together with

one or several domain-specific vocabularies to give the proper semantics to the terms of the agreement. Furthermore, it defines a protocol and a web service-based interface to create, represent, and allow the monitoring of agreements.

However, WS-Agreement only defines a take-it-or-leave-it protocol. To use more complex negotiation protocols, other specifications must be implemented. For instance, WS-AgreementNegotiation, which builds on WS-Agreement and specifies a bilateral negotiation protocol, or the negotiation protocols defined by FIPA (*Foundation for Intelligent Physical Agents*).

With regard to the trading management process different approaches are possible depending on the complexity of the trading protocol used. For complex co-ordinations, there are workflow standards such as BPEL (*Business Process Execution Language*) [1] or choreography languages such as WS-CDL (*Web Services Choreography Description Language*). For simple cases, an alternative to implement Trading Protocols would be the specification of ad-hoc elements in the concrete architecture built upon the conceptual framework.

4 Conclusions

In recent years the software industry has been moving from a vision of software as a product to a vision of software as a service. This change has become even more apparent with the appearance and popularization of SOA and web services.

This new vision opens up new possibilities and challenges for the software industry. In this article, we focus on the automated provision of services and, more particularly, we centre on the trading process, that is, the process of establishing a service agreement between two parties. The trading process is as a core element in the development of automated service provision.

As we have discussed, the trading process is complex and involves many different interactions between the parties involved. To deal with this complexity, we have broken it down into several subprocesses: discovery, information, selection, agreement making, binding, and trading. Then, we have analysed the main challenges posed by each subprocess and we have described different approaches and technologies that have been developed to solve each problem.

In addition, when reaching an agreement, there are explicit temporal constraints that regulate how the subprocesses that compose the trading process must be coordinated. These constraints are defined in a trading protocol, as we have discussed previously. From this analysis we conclude that it is necessary to integrate several different approaches and technologies in order to build a system capable of handling the entire trading process. Today, this integration can be carried out in an ad-hoc way. However, there are a number of different trading scenarios, and each scenario has different requirements [9]. There is therefore no general solution for each problem, but each solution de-

depends on the specific trading scenario in question. In addition, the time constraints of the trading process create a great many difficulties in the coordination of the various parts of the trading system.

Consequently, the idea of developing an integration framework to integrate and coordinate all these partial solutions is very appealing, and forms the basis of our current research [11].

The main goal of the framework is to provide infrastructure services which are common to several trading scenarios and that enable the integration and coordination of different proposals and technologies. This will enable the parts of the framework that may be useful for different scenarios (e.g. the discovery of potential service providers) to be reused. Furthermore, this framework would also facilitate the application in real scenarios of research proposals that deal with parts of the trading process, because it would allow us to focus on the specific problem only, leaving all other details to the framework.

Translated by the authors

References

- [1] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke and M. Xu. WS-Agreement Specification, 2004.
- [2] M. Burstein, C. Bussler, M. Zaremba, T. Finin, M. N. Huhns, M. Paolucci, A. P. Sheth, S. Williams. A Semantic Web Services Architecture. *IEEE Internet Computing*, 9(5):72-81, 2005.
- [3] A. Ruiz-Cortés, O. Martín-Díaz, A. D. Toro and M. Toro. Improving the automatic procurement of web services using constraint programming. *Int. J. Cooperative Inf. Syst.*, 14(4):439-468, 2005.
- [4] Global Grid Forum. Open Grid Service Architecture. <<http://www.ggf.org/documents/GFD.30.pdf>>, 2005.
- [5] G. Laures and K. Jank. Adaptive Service Grid Reference Architecture. <<http://www.asg-platform.org>>, July 2005.
- [6] H. Ludwig. A Conceptual Framework For Building E-Contracting Infrastructure. In R. Corchuelo, R. Wrembel, and A. Ruiz-Cortes, editors, *Technologies Supporting Business Solutions*, chapter 1. Nova Publishing, 2003.
- [7] H. Ludwig, H. Gimpel, A. Dan, and R. Kearney. Template-Based Automated Service Provisioning - Supporting the Agreement-Driven Service Life-Cycle. In *Proceedings of ICSOC 2005*, pp 283-295, 2005.
- [8] C. Molina-Jimenez, J. Pruyne, and A. van Moorsel. The Role of Agreements in IT Management Software. In R. de Lemos, C. Gacek, and A. Romanovsky, editors, *Architecting Dependable Systems III*, volume 3549, pp 36-58. Springer-Verlag GmbH, Sept. 2005.
- [9] M. Resinas, P. Fernandez, and R. Corchuelo. An analysis of service trading architectures. In *Proceedings of EC-Web 2006*, volume 4082 of *Lecture Notes in Computer Science*, pp 204-213. Springer-Verlag GmbH, 2006.
- [10] M. Resinas, P. Fernandez, and R. Corchuelo. A conceptual framework for automated negotiation systems. In *Proceedings of International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2006)*, volume 4224 of *Lecture Notes in Computer Science*, pp 1250-1258. Springer-Verlag GmbH, 2006.
- [11] M. Resinas, P. Fernandez, and R. Corchuelo. Towards automated service trading. In *Proceedings of ICE-B 2006*, to appear. Springer, 2006.
- [12] K. Sycara, M. Paolucci, J. Soudry, and N. Srinivasan. Dynamic Discovery and Coordination of Agent-Based Semantic Web Services. *IEEE Internet Computing*, 8(3):66-73, 5, 2004.
- [13] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson. *Web Services Platform Architecture*. Prentice Hall, 2005. ISBN: 0-131-48874-0.