

Agreement Patterns

Carlos A. Iglesias^{1**}, Mercedes Garijo¹,
José I. Fernández-Villamor¹, and José J. Durán²

¹ Depto. Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid

² Centro para las Tecnologías Inteligentes de la Información y sus Aplicaciones (CETINIA), Universidad Rey Juan Carlos

Abstract. Agreement Patterns have been defined for improving communication in software and services development, as well as for providing a practitioner oriented approach for reusing existing agreement technologies. This article presents the notion of agreement patterns, their structure and some of the first identified examples.

1 Introduction

One of the envisioned aspects of Future Internet is the Internet of Services, which is conceived as [1] an open service delivery platform, which goes beyond the client-server model of service delivery.

In an Internet of Services with a vast and changing landscape of service providers and consumers, the fulfillment of agreements for services is a key issue, which involves notions such as trust, coordination or organization. These notions are the basis for the understanding and implementation of artificial social systems [2], and have largely addressed in fields such as multi-agent systems [3,4,5], p2p computing [6], service computing [7,8,9], autonomic computing [10], social psychology [11], sociobiology or social neuroscience.

Agreement technologies [12] (AT) is a recent discipline which collects this multidisciplinary research and can be defined as *the technologies for the practical application of knowledge to the automated fulfillment of agreements*. Agreement technologies do not dictate the underlying technologies (objects, components, agents, services, ...), but are focused on the formalization of knowledge structures, protocols, algorithms and expertise that contribute to the establishing of agreements in an open dynamic environment.

Nevertheless, there has not been defined yet a common vocabulary or patterns for sharing and reusing previous experiences in the application of agreements to social-inspired software integration. This article pretends to bridge the gap through the use of software patterns for agreement analysis and reuse, which are called *agreement patterns*.

The rest of the article is organized as follows. Section 2 reviews some of the related work in patterns in the SOA and multi-agent community. Then, section 3

** The first author has been partially supported by Germinus XXI (Grupo Gesfor) through the project RESULTA.

presents a classification scheme for agreement patterns. Section 4 introduces some examples of identified agreement patterns. Finally, section 5 draws out the conclusions of this research work and the future work.

2 Background and Related Work

Software design patterns [13] can be defined as “*a technique for achieving widespread reuse of software architecture. They capture the static and dynamic structures and collaborations of components in successful solutions to problems that arise when building software in domains like business data processing, telecommunications, graphical user interfaces, databases and distributed communication software. Patterns aid the development of reusable components and frameworks by expressing the structure and collaboration of participants in a software architecture at a level higher than source code or object-oriented design models.*”

Patterns can be represented in an informal way using natural language, UML, pattern languages or ontologies [14]. The informal representation of software patterns usually follows the canonical (or so-called Alexandrian) form [15], which identified the following elements:

- **Name:** a meaningful name that provides a vocabulary for discussing.
- **Alias:** an alternative name to the pattern.
- **Problem:** a statement of the problem and the goals it wants to reach.
- **Context:** the preconditions under which the problem and its solution seem to recur.
- **Forces:** a description of the relevant forces and constraints and how they interact with one another and with the goals. Considerations to be taken into account to select a solution for a problem.
- **Solution:** static relationships and dynamic rules describing how to realized the desired outcome.
- **Examples:** one or more sample applications of the pattern which illustrate its application. Known occurrences of the pattern which help in verifying that the pattern is a proven solution to a recurring problem.
- **Resulting context:** the state or configuration of the system after the pattern has been applied.
- **Rationale:** a justification of the pattern, explaining how and why it works, and why it is “good”.
- **Related patterns:** compatible patterns which can be combined with the described pattern.

Based on this previous research, **Agreement patterns** are defined as *software patterns that facilitate software components coordination through the fulfillment of agreements*. Agreements patterns include all kind of agreements, both explicit ones (e.g. negotiation) and tacit ones (e.g. organization).

There are related works for defining design patterns in the areas of multi-agent systems and Service Oriented Computing (SOC).

Agent-Oriented Patterns have been defined for sharing multi-agent system development experiences. Oluyomi [16,17] presents an agent pattern classification scheme based on two dimensions: stages of the agent-oriented software development and tasks in each stage of development. At each stage or level of development (analysis, multi-agent architecture, agent architecture, multi-agent implementation), the framework identifies the attributes of that level of abstraction, in order to classify these patterns. In addition, Oluyomi proposes to refine the canonical pattern form for defining an Agent-Oriented Pattern Template Structure, which adds more granularity depending on the pattern type (agent internal architecture structural, interactional or strategic patterns, etc.). Some of the patterns identified by Oluyomi, whose classification scheme includes other approaches, can be considered agreement patterns. The main differences between her classification and the one proposed in this article is that Oluyomi's classification is agent oriented (agent oriented development phase, agent architecture, etc.), while the one proposed here is independent of the technology to be used, although implementation examples can be presented with different technologies. In addition, agreements are not a key concept in Oluyomi's classification scheme as in our proposal. Future work will provide a mapping of the agreement related patterns classified by Oluyomi onto our classification scheme.

In the area of Service Oriented Architecture (SOA), *SOA patterns* have been defined [18,19,20]. For example, Erl [18] classifies patterns for architecting services, service compositions, service inventories and service oriented enterprise. Rotem-Gal-Oz [19] describes patterns for Message Exchange, Service Interaction, Service Composition, Structural, Security and Management. SOA patterns [20] provide high level architectural patterns, which do not detail yet agreement issues.

Inside the SOC community, the GRAAP Working Group (Grid Resource Allocation and Agreement Protocol WG) has defined the specification Web Services Agreement [8], which is particularly interesting for this research. The purpose of the specification is the definition of a Web Services protocol for establishing agreements defined in XML. The specification covers the specification of agreement schemas, agreement template schemas and a set of port types and operations for managing the agreement life cycle. This specification defines an agreement as *an agreement between a service consumer and a service provider specifies one or more service level objectives both as expressions of requirements of the service consumer and assurances by the service provider on the availability of resources and/or service qualities. An agreement defines a dynamically-established and dynamically-managed relationship between parties. The object of this relationship is the delivery of a service by one of the parties within the context of the agreement. The management of this delivery is achieved by agreeing on the respective roles, rights and obligations of the parties.* An agreement is characterized by its name, context and terms.

The OASIS Reference Architecture for SOA [9] is *an abstract realization of SOA, focusing on the elements and their relationships needed to enable SOA-based systems to be used, realized and owned.* The reference architecture defines

three primary viewpoints: *business via services* that captures what SOA means for people using it to conduct business, *realizing service oriented architectures* deals with the requirements for constructing a SOA; and *owning service oriented architectures* addresses issues involved in owning and managing a SOA. The notion of agreement is included in several ways in the architecture, as an organizational concept (constitution) or as a formalization of a relationship (business agreement and contract).

These two initiatives, OASIS RA and WS-Agreement are compatible and complementary of our proposal, since they provide a modeling reference architecture as well as a language for describing the identified patterns boiling down to the implementation level. This integration will be included in future publications.

3 Classification Scheme for Agreement Patterns

The purpose of this section is an early identification of a multi-dimensional classification scheme for agreement patterns (figure 1) which makes easier for software developers to select the right pattern to apply, as well as the basis for its cataloging.

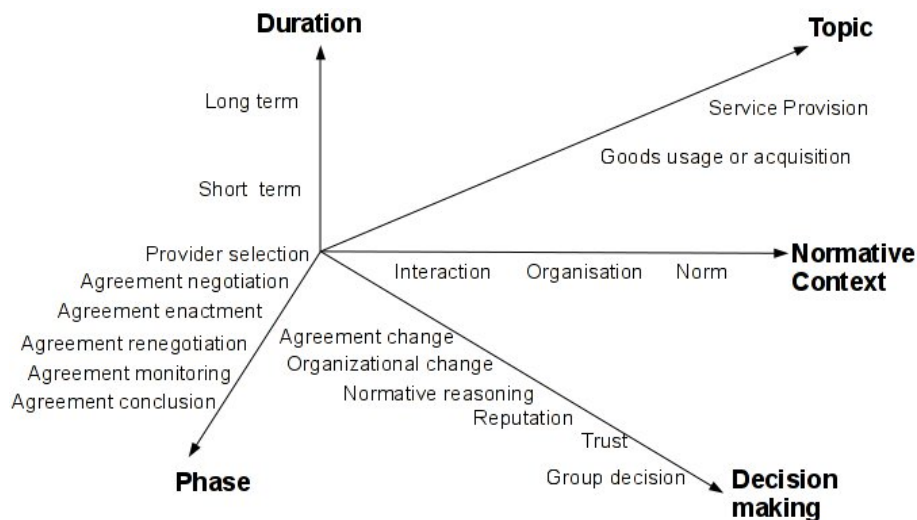


Fig. 1. Classification scheme for Agreement Patterns

The identified dimensions characterize an agreement from several properties of the agreement itself, such as topic (purpose of the agreement), duration, and phase (also so-called state in WS-Agreement); the organizational environment where the agreement takes place, and the decision making cognitive task of the agreement stakeholders.

One pattern can have more than one value for each one of the dimensions, and can be classified only to the relevant dimensions.

The dimensions of the classification scheme are:

- **Duration:** if the agreement is reached for a short or long term. The duration of the agreement has organizational implications, as well as its properties. Pěchouček [21] distinguishes between *alliances* for long-term collaboration agreements and *coalitions* for short-term collaboration agreements. In a similar way, other authors such as Camarinha [22] identifies *long term strategic alliances* and *goal-oriented networks*, based on this dimension.
- **Topic:** the goal of the agreement can be [5], among others, the obtention of goods, the usage of goods or services, the provision of a service or the definition of social rules.
- **Phase:** this dimension defines the agreement phase where the pattern can be applied. The phases considered are provider selection, agreement fulfillment, agreement renegotiation, agreement monitoring and agreement conclusion. *Provider selection* consists of the selection of the best available provider before establishing an agreement, which can be based on trust and reputation techniques and can involve interactions such as bidding or contract net protocol (CNP) [23]. *Agreement negotiation* is the process of defining establishing the conditions of the agreement, respecting the normative context and using negotiation techniques such as argumentation. *Agreement enactment* is the commitments of the agreement are effective. For example, in the call-by-agreement interaction method [24], once the agreement for action is establish, then the actual enactment of the action is requested. In the CNP, the successful bidders are informed that they are now contractors for a task together with this task specification. *Agreement renegotiation* is the phase of changing the conditions of a previously negotiated agreement, because of reasons such as environmental changes, expiration of the agreement or the coming up of a renewal option. *Agreement monitoring* is the process of reviewing the established agreement conditions while the agreement is being enacted. In the area of service computing, it is usually referred as Service Level Agreement monitoring. Finally, *Agreement conclusion* is the process of concluding an agreement by any party, which can be because the agreement is not being enacted according to the negotiated terms, or because one or more parties desires to conclude it.
- **Normative context:** the normative context [24] determines the rules of the game, such as interaction patterns, norms and organization structures.
- **Decision making:** this dimension collects cognitive patterns followed by the parties of the agreement. Some of the main subcategories are *trust and reputation* [25], *group decision* [26], *organizational change* [24], reasoning about the impact of the organization evolution in the current negotiated agreements and in the agreement procedures; *normative reasoning* [24] for interpreting the normative context for a specific case; *agreement change*, reasoning about the improvement of the negotiated agreements as well as its interpretation in specific situations.

The dimensions are interrelated, and one value in one dimension can constraint the values of other dimensions. For example, the topic (goods acquisition) can restrict the valid values of normative context, being interaction (bidding, contract-net protocol, ...) or organizational structure (electronic institution, ...).

4 Examples

This section provides examples of some of the first identified agreement patterns, whose pattern structure follows the canonical form extended with the classification scheme defined in section 3. The decision making process is shown in a reasoning diagram which follows CommonKads notation [27] as illustrated in 2.

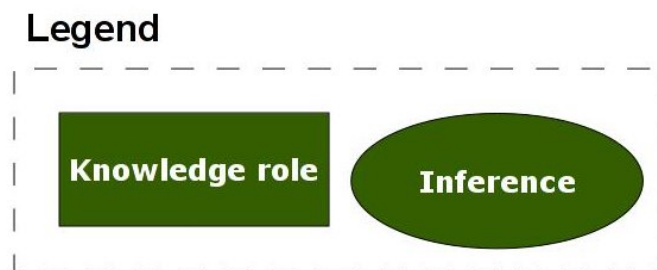


Fig. 2. Inference diagram legend

4.1 ProviderRating Pattern

Selection of the most satisfactory service providers for a specific service demand is known as the *Service Selection* problem [28,29].

This pattern collects the pattern of rating available service providers in order to select the one with highest rating. Rating-based mechanisms are simple and effective [30], if consumers have similar preferences.

Name ProviderRating

Duration Short term (specific project)

Phase Provider Selection

Decision making Reputation, Trust

Problem Choosing a service provider, having access to other users' rating.

Context Public rating of services.

Forces A comparison of the experience-based service-selection techniques can be found in [31].

Solution Service provider selection is based on the rating given to service providers using reputation [29] or trust [32] techniques.

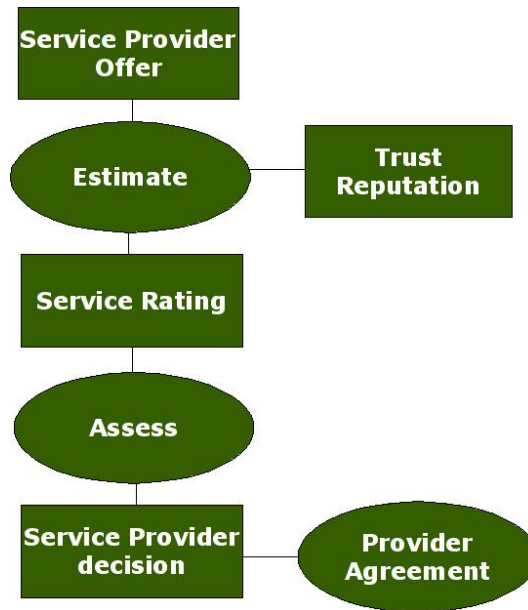


Fig. 3. Provider Rating Reasoning Diagram

Examples E-Commerce [33]

Resulting context An agreement is fulfilled.

Related Patterns Portability

4.2 Portability Pattern

This pattern collect the problematic of changing of provider without service interruption, which involves monitoring the current QoS and the estimation of other Service Provider offer. When the decision of changing is taking, this pattern involves the negotiation of breaking the existing contract and agree a new contract with the new provider.

Usually, service providers try to prevent this portability, defining some *customer retention* policy to improve *customer loyalty*.

Name *Portability*

Alias *Service switching* [34]

Problem A potentially better service offer is available and the customer wants to break the service provision agreement with the current service provider and to establish a new agreement with a new provider, without disruption of the service.

Context There are several available service providers that offer a better service conditions than the current provider.

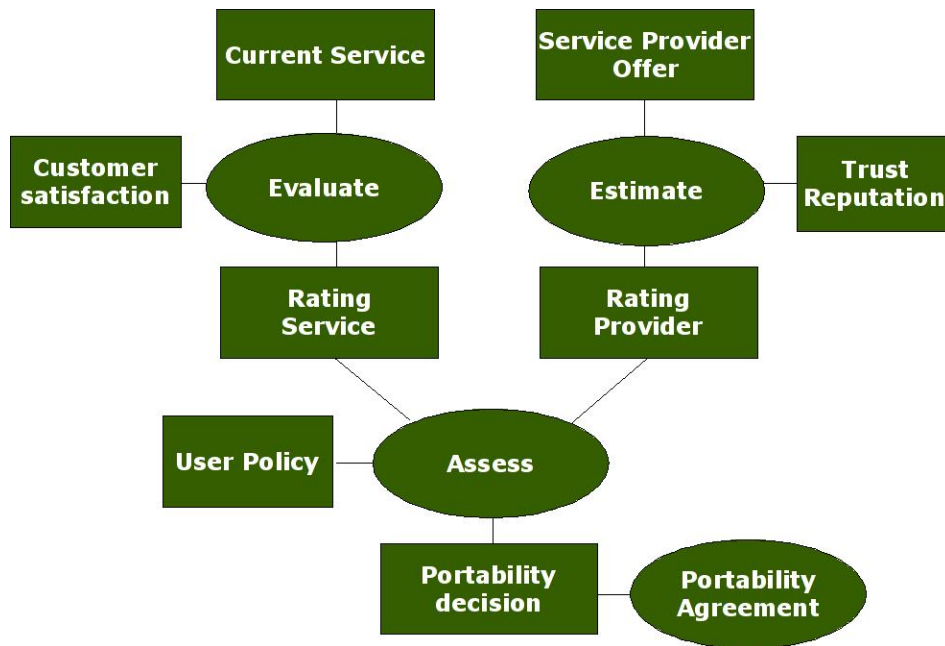


Fig. 4. Portability pattern reasoning diagram

Forces This pattern is applicable when there are more than one stakeholder in the development of the new service, and the consumer do not have strong penalties or commitments with the current provider.

Solution Based on the QoS evaluation of the current service provider, the estimation of the QoS of the available service providers, and the consumer policies and commitments, the decision to change of provider is taken, which involves requesting portability (breaking the current agreement with the service provider and establishing an agreement with the new provider).

Examples This pattern is frequently found in several domains. For example, it is a common pattern in telephony customers or mobile network environments.

Resulting context The consequences of the portability can be penalties due to commitments with the initial service provider, and the new commitment with the new service provider.

Duration Short term (specific project)

Topic Service Provision

Phase Agreement conclusion, Provider selection, Agreement negotiation

Decision making Agreement change

Related Patterns –

Known uses Wireless networks [35], professional services [36]

The context and resulting context of this pattern can be easily formalized as preconditions and postconditions as follows. Let be d a decisor, p_c the current

provider, p_i the available provider i , and the relation $provider(decisor, provider)$ used for representing that $decisor$ has an agreement in effect with the $provider$.

Let be U_i^d the utility function of a decisor d for a provider i .

In order to be applied this pattern, the following pre and postconditions should be fulfilled.

$$\text{Preconditions} \begin{cases} provider(d, c) \wedge \\ \neg provider(d, i) \forall i, \quad i \neq c \wedge \\ \exists i, \quad U_i^d \geq U_c^d \quad | \quad i \neq c \end{cases}$$

$$\text{Postconditions} \quad provider(d, i) \wedge \neg provider(d, c)$$

4.3 Intermediary Pattern

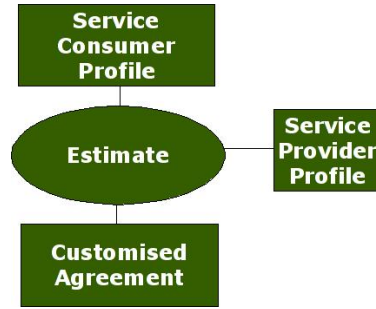


Fig. 5. Intermediary Pattern

Name *Intermediary*

Alias *Service Broker*

Duration Short term (specific project)

Topic Service Provision

Phase Agreement Negotiation

Problem Selection of a service provider that fits the QoS requirements of the consumer.

Context There are several available service providers that fit the functional requirements.

Forces Some of the considerations to be taken into account if the trust of the service consumer on the intermediary.

Solution The intermediary selects one or more providers and adapts the service provider offer to the service consumer preferences, providing an added value.

Examples Service provider selection in wireless networks [35], tourism and finance [37],

Resulting context the intermediary acts on behalf of the user based on an agreement between user and intermediary. This pattern refines the *service mediator* entity defined within SOA-RM [9].

Related Patterns Portability. Portability could be a decision of the intermediary.

5 Conclusions and Future work

This article presents preliminary results of the research on defining a software methodology for *agreement technology*. The task is challenging, since *agreement technology* is an emerging area.

The research has been based on the following premises:

- *Applicability*. The results of agreement technologies should be applied by any software component which requires to fulfill an agreement. In particular, we have not restricted the applicability to the agent community.
- *Simplicity*. Agile methodologies have shown their potential for fitting in every day task. In particular, design patterns and refactoring have succeed in its adoption. The results of our approach try to be easy to adopt.
- *Reuse*. In order to provide technology transfer from agreement technologies to other areas, it is needed to organize and provide facilities for the comprehension of the research results. Patterns provide a suitable mechanism for this technology transfer and a principled way.

The main conclusions of our research is that *agreement patterns* can be a useful concept for providing a unifying vocabulary in agreement technology, and providing a catalogue of methods for its implementation.

In this article we have proposed a first structure and some examples of application with some agreement patterns we have identified. Nevertheless, these are initial results, and we are now working on a more definitive structure of the patterns, as long as its formalization and operationalization.

Acknowledgements

This research has been partly funded by the Spanish Ministry of Science and Innovation through the projects Ingenio Consolider2010 AT *Agreement Technologies* (CSD2007-0022) and T2C2 (TIN2008-06739-C04-03/TSI) as well as the Spanish Ministry of Industry, Tourism and Trade through the project RESULTA (TSI-020301-2009-31).

We would like to express our gratitude to Sascha Ossowski and Alberto Fernández for involving us in the project AT as well as the rest of the URJC research group for their support within the project. We would like also to thank the rest of research groups of AT, IIIA CSIC and UPV for their comments, and specially to Vicente Botti and Vicente Julián for their support, motivation and patience along the project.

References

1. Working Group on Future Internet Infrastructure FP7 ICT Advisory Group: Future internet infrastructure. Technical Report version 8, FP7 ICT Advisory Group (January 2008)
2. Sierra, C., Botti, V., Ossowski, S.: Agreement technologies. Available at <http://www.agreement-technologies.org> (2008)
3. Jennings, N.R., Faratin, P., Norman, T.J., O'Brien, P., Wiegand, M.E., Voudouris, C., Alty, J.L., Miah, T., Mamdani, E.H.: ADEPT: Managing business processes using intelligent agents. In: Proceedings of the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems (ISIP Track), Cambridge, UK (1996) 5–23
4. Huhns, M.N., Singh, M.P., Burstein, M., Decker, K., Durfee, E., Finin, T., Gasser, L., Goradia, H., Jennings, N., Lakkaraju, K., Nakashima, H., Parunak, V., Rosenschein, J.S., Ruvinsky, A., Sukthankar, G., Swarup, S., Sycara, K., Tambe, M., Wagner, T., Zavala, L.: Research directions for service-oriented multiagent systems. *IEEE Internet Computing* **9**(6) (2005) 65–70
5. Lomuscio, A.R., Wooldridge, M., Jennings, N.R.: A classification scheme for negotiation in electronic commerce. *Lecture Notes in Computer Science* **1991** (2001)
6. Greco, G., Scarcello, F.: On the complexity of computing peer agreements for consistent query answering in peer-to-peer data integration systems. In: *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, New York, NY, USA, ACM (2005) 36–43
7. Papazoglou, M.P.: Service -oriented computing: Concepts, characteristics and directions. *Web Information Systems Engineering, International Conference on* **0** (2003) 3
8. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web services agreement specification (WS-Agreement). Technical report, Grid Resource Allocation Agreement Protocol (GRAAP) Working Group (2007)
9. McCabe, F.G.: Reference architecture for service oriented architecture. Technical report, OASIS (April 2008)
10. Parashar, M., Hariri, S.: *Autonomic Computing*. CRC Press, Inc., Boca Raton, FL, USA (2006)
11. Carnevale, P.: *Psychology of Agreement*. Psychology Press (2009)
12. Jennings, N.: Agreement technologies. *Intelligent Agent Technology, IEEE / WIC / ACM International Conference on* **0** (2005) 17
13. Schmidt, D.C.: Using design patterns to develop reusable object-oriented communication software. *Communications of the ACM* **38**(10) (1995) 65–74
14. Rosengard, J.M., Ursu, M.F., marc Rosengard, J., Ursu, M.F.: Ontological representations of software patterns. In: *Proc. KES'04, Lecture Notes in Computer Science*, Springer Verlag (2004) 31–38
15. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc. (1996)
16. Oluyomi, A., Karunasekera, S., Sterling, L.: A comprehensive view of agent-oriented patterns. *Autonomous Agents and Multi-Agent Systems* **15**(3) (2007) 337–377
17. Oluyomi, A.O.: *Patterns and Protocols for Agent-Oriented Software Development*. PhD thesis, Faculty of Engineering. University of Melbourne, Australia. (November 01 2006)

18. Erl, T.: SOA Design Patterns. Prentice-Hall (2008)
19. Rotem Gal Oz, A.: SOA Patterns. Manning (2009)
20. Zdun, U., Hentrich, C., Aalst, W.M.P.V.D.: A survey of patterns for service oriented architectures. *Int. J. Internet Protoc. Technol.* **1**(3) (2006) 132–143
21. Pěchouček, M., Mařík, Bárta, J.: A knowledge-based approach to coalition formation. *IEEE Intelligent Systems* **17**(3) (2002) 17–25
22. Camarinha-Matos, L.M., Afsarmanesh, H.: Collaborative networks. value creation in a knowledge society. In: Proceedings of PROLAMAT'06, Shanghai, China (2007)
23. Smith, R.G.: The contract net protocol: High-level communication and control in a distributed problem solver. *Transactions on Computers* **C-29**(12) (1980) 1104–1113
24. Ossowski, S.: Coordination in multiagent systems – towards a technology of agreement. In: MATES-2008. Volume 5244., Springer Verlag, Springer Verlag (2008) 2–12
25. Mui, L.: Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks. PhD thesis, MIT (2002)
26. Kersten, G.: Support for group decisions and negotiations. an overview. In Climaco, J., ed.: *Multicriteria Analysis*, Springer Verlag (1997) 332–346
27. Schreiber, G., Akkermans, H., Anjewierden, A., Dehoog, R., Shadbolt, N., Van de Velde, W., Wielinga, B.: *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press (December 1999)
28. Maximilien, E.M., Singh, M.P.: A framework and ontology for dynamic web services selection. *IEEE Internet Computing* **8**(5) (2004) 84–93
29. Şensoy, M.: A Flexible Approach For Context-Aware Service Selection In Agent-Mediated E-Commerce. PhD thesis, Boğaziçi University (2008)
30. Şensoy, M., Yolum, P.: On choosing an efficient service selection mechanism in dynamic environments. In: Proceedings of the 9th International Workshop on Agent-Mediated Electronic Commerce (AMEC IX), 2007. Volume 13. (2007) 105–118
31. Şensoy, M., Yolum, P.: A comparative study of reasoning techniques for service selection. In: Proceedings of the 5th International Workshop on Agents and Peer-to-Peer Computing (AP2PC'06). Volume 123–134. (2006) 91–102
32. Billhardt, H., Hermoso, R., Ossowski, S., Centeno, R.: Trust-based service provider selection in open environments. In: SAC '07: Proceedings of the 2007 ACM symposium on Applied computing, New York, NY, USA, ACM (2007) 1375–1380
33. Sierra, C., Faratin, P., Jennings, N.R.: A service-oriented negotiation model between autonomous agents, Springer-Verlag (1997) 17–35
34. Keaveney, S.: Customer switching behaviour in service industries: an exploratory study. *Journal of Marketing* **59** (1995) 71–82
35. Lee, E.G., Lee, G., Faratin, P., Bauer, S., Wroclawski, J.: Automatic service selection in dynamic wireless network. Technical report, First ACM International Workshop WMASH'03 (2003)
36. Kugytė, R., Sliburyte, L.: A standardized model of service provider selection criteria for different service types: a consumer oriented approach. *Engineering Economics. Commerce of Engineering Decisions* **43**(3) (2005) 56–63
37. Yu, C.C.: A web-based consumer-oriented intelligent decision support system for personalized e-services. In: ICEC '04: Proceedings of the 6th international conference on Electronic commerce, New York, NY, USA, ACM (2004) 429–437