The ArguGRID Platform: An Overview

Francesca Toni[1], Mary Grammatikou[2], Stella Kafetzoglou[2], Leonidas Lymberopoulos[2], Symeon Papavassileiou[2], Dorian Gaertner[1], Maxime Morge[3], Stefano Bromuri[4], Jarred McGinnis[4], Kostas Stathis[4], Vasa Curcin[5], Moustafa Ghanem[5], and Li Guo[5]

[1] Imperial College London, Department of Computing, UK
[2] Institute of Communications and Computer Systems, NTUA, Greece
[3] Dipartimento di Informatica, Pisa University, Italy
[4] Department of Computer Science, Royal Holloway, University of London, UK
[5] InforSense Ltd., UK

Abstract. The ArguGRID project aims at supporting service selection and composition in distributed environments, including the Grid and Service-oriented architectures, by means of argumentative agents, an agent environment, a service-composition environment, Peer-to-Peer technology and Grid middleware. Agents are argumentative in that they use argumentation-based decision-making and argumentation-supported negotiation of services and contracts. The integration of all technologies gives rise to the overall ArguGRID platform. In this paper we outline the main components and the overall functionalities of the ARGUGRID platform.

Keywords: Grid Computing, Service-Oriented Computing, e-Business.

1 Introduction

The ArguGRID project[1] aims at developing a Grid/Service-oriented platform populated by rational decision-making agents that are associated with service requesters, service providers and users. The project also aims at using Semantic Web technologies to support semantic integration of services in distributed environments such as the Grid. Within agents, argumentation [5,15,20] is used to support decision making, taking into account (and despite) the frequently conflicting information that these agents have, as well as the preferences of users, service requesters and providers. Argumentation is also intended to support the negotiation between agents [6,14], on behalf of service requesters/providers/users. This negotiation takes place within dynamically formed Virtual Organisations. The agreed combination of services can be seen as a complex service within a service-centric architecture [4]. We intend to validate this overall approach by way of industrial e-business application scenarios [19].

A high-level view of the ArguGRID vision was presented in [4]. In this paper we outline the ArguGRID platform and architecture, by describing at a high-level

[1] `http://www.argugrid.eu`

J. Altmann, D. Neumann, and T. Fahringer (Eds.): GECON 2008, LNCS 5206, pp. 217–225, 2008.

all its components and by showing how these fit together and provide support for user applications. These components include:

- tools for the authoring and execution of workflows, namely combinations of services, to fulfil the requirements (goals) of users;
- argumentation engines to support decision-making and negotiation;
- an agent platform used to support inter-agent interactions;
- a Peer-to-Peer platform for the discovery of Grid services and agents within the platform;
- Grid middleware.

The paper is organized as follows. In Section 2 we give an overview of the project's aims, scenarios and overall methodologies. In Section 3 we summarise the main components of the ArguGRID platform and describe the integration of these components within the ArguGRID platform. In Section 4 we conclude.

2 The ArguGRID Vision

ArguGRID aims to:

- develop argumentation-based foundations for the Grid, populated by rational decision-making agents within Virtual Organisations;
- incorporate argumentation models into a service-centric architecture;
- develop an underlying platform using Peer-to-Peer computing;
- validate the ArguGRID approach by way of industrial application scenarios.

We have chosen a number of e-business application scenarios [19], including

- e-procurement applications and e-Marketplaces,
- e-business for Earth Observation applications.

These scenarios are the outcome of and build upon the extensive field experience of the two industrial partners of the consortium (cosmoONE Hellas Market-site S.A. [2] and GMV S.A. [3], respectively). In [21] we summarise the rationale for the choice of these scenarios to guide the development of and validate the ArguGRID approach to Grid computing and service-oriented architectures.

The envisaged ArguGRID platform is intended to be a multi-layered architecture where: the top layer is about building applications; the middle layer concerns the development of individual agents as well as methodologies for dynamically assembling agents into Virtual Organisations responsible for the negotiation of contracts between service providers and requesters; agents and Virtual Organisations sit on top of the bottom-layer, consisting of Peer-to-Peer and Grid middleware. Each service requester/provider and each user is associated with one or more agents. Agents use argumentation for negotiating on behalf of service

[2] `http://www.cosmo-one.gr/en`
[3] `http://www.gmv.com`

requesters/providers/users. By means of the top-layer, users can provide input to agents, in terms of their objectives (what they expect to achieve from the service composition performed by the agents) and preferences (either for the specific objectives, or, more generally, as a generic profile of the user).

Within the middle layer, agents negotiate with one another by using argumentation to support their decision making and communication processes. Negotiation takes place within dynamically created and maintained Virtual Organisations, envisaged as societies of agents whereby interaction is regulated by social norms and/or protocols. The outcome of negotiation results in a contract, understood, at the agent level, as a task allocation (in terms of provision of resources/services) to agents. In particular, this contract may include a workflow description that needs to be appropriately executed (within the bottom layer).

3 The ArguGRID Platform

The ArguGRID platform consists of four interacting components:

– InforSense KDE: this is a commercial software tool developed by InforSense Ltd [4], and that originates from the Discovery Net e-Science project at Imperial College London [17]. This system provides facilities to build end user application as workflows coordinating the execution of remote web services [3] or Grid services [10]. For the needs of ArguGRID, the KDE system is extended to support semantic workflow authoring and composition and to cater for a semantic registry, which holds higher-level and semantic service descriptions, such as information about their functionality, e.g. QoS, cost, etc. This way, abstract workflows representing user needs can be matched partially or be fully instantiated as concrete workflows and be executed and validated within the grid infrastructure.
– GOLEM (Generalized OntoLogical Environments for Multi-agent systems) [5]: this is an agent environment middleware that can be used to create multi-agent system applications. Applications in GOLEM can be specified declaratively, thus making the deployment of cognitive agents of the kinds envisaged by ArguGRID easier in that perceiving the environment amounts to importing parts of a logical theory [1,2,18].
– PLATON (Peer-to-Peer Load Adjusting Tree Overlay Networks) [6]: this is a Peer-to-Peer platform supporting multi-attribute and range queries [11]. It is developed in the Java programming language and supports mechanisms for load-balancing of peer resources. Load-balancing of resources is necessary in order to guarantee logarithmic querying time using any distributed tree-based multi-attribute Peer-to-Peer platform. In its current release, PLATON has implemented the SkipIndex routing framework by Princeton University.

[4] http://www.inforsense.com
[5] http://www.golem.cs.rhul.ac.uk
[6] http://platonp2p.sourceforge.net

– GRIA [7]: this is the Grid middleware that ArguGRID has chosen to use to support its scenarios. The reason for choosing GRIA is that GRIA is a service-oriented infrastructure designed particularly to support Business-to-Business collaborations (such as the ones required by the ArguGRID scenarios) through service provision across organisational boundaries in a secure, interoperable and flexible manner.

The InforSense KDE constitutes the top-layer of the ArguGRID platform, GOLEM supports the middle layer, and the combination of PLATON and GRIA forms the bottom layer.

Within the ArguGRID platform, GOLEM hosts MARGO agents [15,16], running the MARGO argumentative decision-making engine [15], which in turn deploys the CaSAPI general-purpose argumentation engine [7,8,9]:

– MARGO (Multiattribute ARGumentation framework for Opinion explanation) [8], written in Prolog, implements the ArguGRID argumentation framework for practical reasoning about service selection and composition. A logic language is used as a concrete data structure for holding the statements like knowledge, goals, and actions. Different qualitative or quantitative priorities are attached to these items, corresponding to the probability of the knowledge, the preferences between goals, and the expected utilities of alternative actions. MARGO evaluates the possible actions, suggests some solutions, and provides an interactive and intelligible explanation of the choice made. MARGO is built on top of CaSAPI.
– CaSAPI (Credulous and Sceptical Argumentation: Prolog Implementation) [9] is a general-purpose tool for (several types of) assumption-based argumentation. It is written in Prolog. It can support several applications, ranging from decision-making to normative reasoning and goal decision, to e-procurement.

A number of interactions/communications are supported between individual ArguGRID components within the ArguGRID platform, as outlined in Figure 1. Note that this figure should not be interpreted as indicating that the components of the platform are held on a single, local computer. Rather, all components (and the ArguGRID platform itself) will be typically distributed among computer elements residing in distinct locations, connected to a network such as the global Internet.

ArguGRID distinguishes between (Grid) service requesters and (Grid) service providers. Agents may act as service requesters or as service providers (or both). Figure 1 presents the ArguGRID platform from a service requester's point of view, i.e. from the view point of users using the ArguGRID platform in order to obtain (typically composite) services. Users in Figure 1 can be either human users or agents using the ArguGRID platform to achieve their goals.

[7] `http://www.gria.org`
[8] `http://margo.sourceforge.net`
[9] `http://www.doc.ic.ac.uk/~dg00/casapi.html`

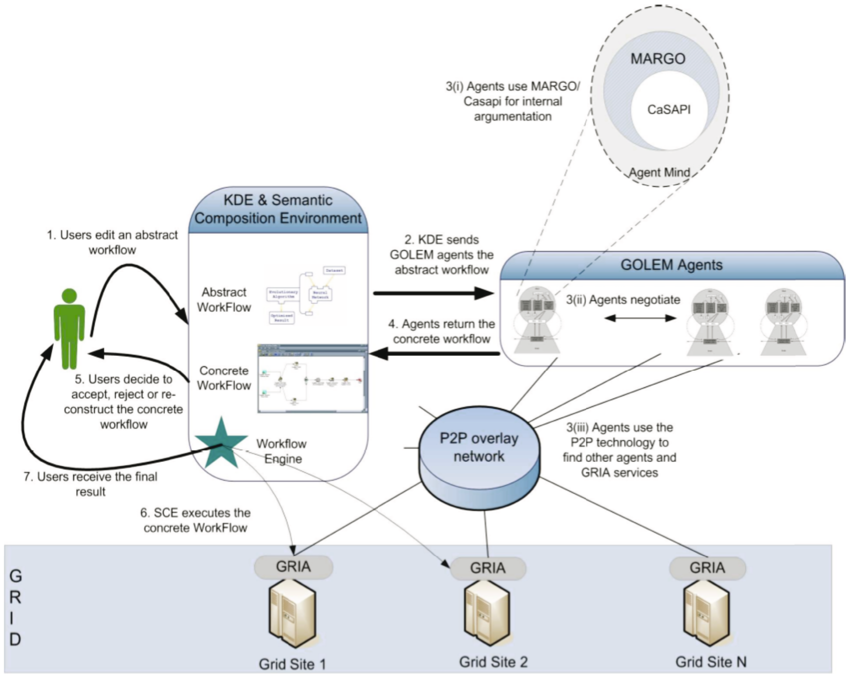

Fig. 1. Global Picture of the ArguGRID Platform

Within the ArguGRID platform, WSML [10] is used to provide semantic descriptions of services in registries available to agents. These desciptions are translated onto a logic-based representation, on demand, so that they can be reasoned upon by GOLEM agents using MARGO.

The main interactions used to support user's requests are the following:

1. Users interact initially with the ArguGRID platform by submitting an abstract workflow to the KDE [11]. This is realised through the KDE workflow editing tool. The abstract workflow reflects, at a high-level, the user requirements. In the next Section, we will give an example of an abstract workflow.
2. In its commercial version, the KDE would involve human interaction in order to derive (having as an input an abstract workflow) a concrete and executable workflow, to be executed on the Grid. In the case of ArguRGID, the KDE is extended so that the process of refining an abstract workflow is delegated to intelligent GOLEM agents. Thus, the KDE communicates the abstract

[10] http://www.wsmo.org/wsml/

[11] Note that this is only one possible entry point to the ArguGRID platform. Indeed, users may also access the platform by interacting directly with a user agent and specifying either an abstract workflow or some high-level goals that agents need to "translate" into workflows. We omit this other view in Figure 1 for the sake of simplicity.

workflow to a GOLEM agent, acting as the agent representing the user within the ArguGRID platform.

3. Having received the abstract workflow, the GOLEM agent representing the user will start finding which GRIA services should be used in order to derive a concrete workflow, to be executed on the Grid. In order to accoGOLEM agent uses the following capabilities:

   (a) A GOLEM agent uses the MARGO argumentation engine for decision-making, which in turn uses the CaSAPI general-purpose argumentation engine. These are implemented within the mind of every GOLEM agent and work in a way to reason about services and make decisions, aiding the refinement process of the abstract workflow.
   (b) A GOLEM agent can negotiate with other GOLEM agents, sign contracts with them and form Virtual Organisations (VOs) [13]. The latter follow the basic philosophy of the Grid, where VOs are formed in order to solve a common problem or task. In our case, the common problem is the problem of providing a solution to the requirements of the user application, i.e. finding a concrete workflow whose execution will satisfy the application requirements, as stated in the abstract workflow. Interactions amongst GOLEM agents are provided by means of dialectical protocols, using special language structures for agent communication.
   (c) To find out an appropriate GOLEM agent or a GRIA Grid service, GOLEM agents are given the capability to use the Peer-to-Peer platform, linking all available GOLEM agents and GRIA services in a virtual registry that can be queried. Implementation of this virtual registry containing all agents and GRIA services is realised using PLATON. Three types of registries exist: GOLEM registries within the GOLEM platform, Grid registries within PLATON and Semantic registries with the Environment of the KDE.

   Note that all interactions described above as cases a,b,c can be realised in parallel, i.e. we do not imply that there is a strict sequence of interactions. Which interaction to use is a choice determined by the mind of the GOLEM agent, while CaSAPI and MARGO are running.

4. Having carried out its mission, the GOLEM agent representing the user (i.e. the initial agent that received the abstract workflow from the KDE) will return back to the KDE the concrete workflow, constituted by a set of GRIA services to be executed in a certain manner/sequence.
5. At this point, a concrete workflow is provided to the KDE. The user is informed of this solution and is given the choice of either accepting the concrete workflow or rejecting it or deciding to modify the abstract workflow, in order to get a better solution. In the latter case, the abstract workflow will be given again as input to the KDE, repeating steps 1, 2, 3, 4 and 5, until the user either accepts or rejects the ArguGRID concrete workflow solution. In the case of acceptance, the system will follow step 6 below.

6. The workflow engine within the KDE will use its workflow execution service to send the concrete workflow for execution on the Grid infrastructure, running the GRIA middleware.
7. Upon successful execution of the concrete workflow, the user is informed and the execution results/data are returned back to the user.

Of course, more than one user will be able to use the ArguGRID platform at the same time, as the ArguGRID platform follows the philosophy of the underlying Grid, having a distributed nature with multiple service providers and service clients, all using the shared Grid infrastructure at the same time, each client trying to achieve his/hers own goals.

4 Conclusions

We have outlined the main components of the ArguGRID platform and their integration to support user-driven applications. We are currently testing components and their integration to support the ArguGRID scenarios of e-procurement and Eearth observation. Preliminary results for e-procurement are described in [16,12].

Overall, the ArguGRID platform affords solutions to problems within these scenarios with the following features: agents automate the process of identifying orchestrations of services (workflows); users and services cooperate (via the agents 'representing' them within the ArguGRID platform) and can negotiate orchestrations of services that require the agents' goals to be flexible; users and services exist within a dynamic and open environment.

Other projects have considered the automated construction of workflows, for example K-Wf Grid [12], which uses agents to support users in authoring workflows. The focus of ArguGRID is the automatisation of the *negotiation* of workflows and contracts amongst agents 'representing' services. Agents are equipped with knowledge, goals and preferences, given to them by users (requesting or providing services), and need to take decisions under 'qualitative' uncertainty. They also use argumentation to 'influence' one another.

To fully support step 3 in Figure 1, we are currently exploring the negotiation of contracts (including SLAs) between users and services (again via the agents 'representing' them within the ArguGRID platform) [6]. We are also studying interaction protocols and strategies amongst agents to support automatic negotiation of workflows and contracts, and the evaluation and use of the trustworthiness of agents (and the services they represent) in order to render these protocols and strategies more effective.

Acknowledgements. This work was funded by the Sixth Framework IST programme of the EC, under the 035200 ArguGRID project. Many thanks to anonymous referees for helpful suggestions and to all participants in the ArguGRID consortium for stimulating challenges to the platform design described in this paper.

[12] http://www.dps.uibk.ac.at/projects/kwfgrid/

References

1. Bromuri, S., Stathis, K.: Situating cognitive agents in golem. In: Proceedings EEMMAS 2007, Dresden, Germany (October 2007)
2. Bromuri, S., Urovi, V., Conteras, P., Stathis, K.: A virtual e-retailing environment in golem. In: Proceedings of Intelligent Environments (IE08), Seattle, US (July 2008)
3. Curcin, V., Ghanem, M., Guo, Y.: Web services in life sciences. Drug Discovery Today 10(12), 865–871 (2005)
4. Curcin, V., Ghanem, M., Guo, Y., Stathis, K., Toni, F.: Building next generation service-oriented architectures using argumentation agents. In: Proc. 3rd International Conference on Grid Services Engineering and Management (GSEM 2006) (September 2006)
5. Dung, P., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artificial Intelligence - Special Issue on Argumentation in Artificial Intelligence 171(10-15), 642–674 (2007)
6. Dung, P., Thang, P., Toni, F.: Towards argumentation-based contract negotiation. In: Proceedings of the Second International Conference on Computational Models of Argument (COMMA'08), Toulouse, France. IOS Press, Amsterdam (2008)
7. Gaertner, D., Toni, F.: CaSAPI: a system for credulous and skeptical argumentation. In: Proc. LPNMR Workshop on Argumentation for Non-monotonic Reasoning, pp. 80–95 (2007)
8. Gaertner, D., Toni, F.: Computing arguments and attacks in assumption-based argumentation. IEEE Intelligent Systems (November/December) (2007)
9. Gaertner, D., Toni, F.: Hybrid argumentation and its properties. In: Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA 2008). IOS Press, Amsterdam (2008)
10. Ghanem, M., Azam, N., Boniface, M., Ferris, J.: Grid-enabled workflows for industrial product design. In: Proc. Second IEEE International Conference on e- Science and Grid Computing (e-Science 2006), December 2006. IEEE Computer Society, Los Alamitos (2006)
11. Lymberopoulos, L., Papavassiliou, S., Maglaris, V.: A novel load balancing mechanism for p2p networking. In: Proceedings of ACM sponsored Conference GridNets, Lyon, France (2007)
12. Matt, P., Toni, F., Stournaras, T., Dimitrelos, D.: Argumentation-based agents for eprocurement. In: Berger, Burg, Nishiyama (eds.) Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008) - Industry and Applications Track, Estoril, Portugal (May 2008)
13. McGinnis, J., Stathis, K., Toni, F.: Virtual organisations as agent societies: Phases. ArguGRID deliverable D.3.2. Technical report, ArguGRID (2008)
14. Miller, T., McBurney, P., McGinnis, J., Stathis, K.: First-class protocols for agent-based coordination of scientific instruments. In: Proceedings of the 5th International Workshop on Agent-based Computing for Enterprise Collaboration: Agent-Oriented Workflows and Services (ACEC 2007), Paris, France (2007)
15. Morge, M., Mancarella, P.: The hedgehog and the fox. An argumentation-based decision support system. In: Proc. of the Fourth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS), pp. 55–68 (2007)
16. Morge, M., McGinnis, J., Bromuri, S., Toni, F., Mancarella, P., Stathis, K.: Toward a modular architecture of argumentative agents to compose services. In: Proc. of the Fifth European Workshop on Multi-Agent Systems (EUMAS), Hammamet, Tunisia, December 2007, pp. 1–15 (2007)

17. Rowe, A., Kalaitzopolous, D., Osmond, M., Ghanem, M., Guo, Y.: The discovery net system for high throughput bioinformatics. Bioinformatics 19, 225–231 (2003)
18. Stathis, K., Kafetzoglou, S., Pappavasiliou, S., Bromuri, S.: Sensor network grids: Agent environment combined with qos in wireless sensor networks. In: Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems (ICAS 2007), Athens, Greece (2007)
19. Stournaras, T., Dimitrelos, D., Tabasco, A., Barba, J., Pedrazzani, D., Yagüe, M., An, T., Dung, P., Hung, N., Khoi, V.D., Thang, P.M.: e-Business application scenarios. ArguGRID deliverable D.1.2. Technical report, ArguGRID (2007)
20. Toni, F.: Assumption-based argumentation for selection and composition of services. In: Proceedings of the 8th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA VIII) (2007)
21. Toni, F.: E-Business in ArguGRID. In: Veit, D.J., Altmann, J. (eds.) GECON 2007. LNCS, vol. 4685, pp. 164–169. Springer, Heidelberg (2007)