SOGOS – A Distributed Meta Level Architecture for the Self-Organizing Grid of Services

C. Beckstein, P. Dittrich, Ch. Erfurth, D. Fey, B. König-Ries, M. Mundhenk, H. Sack
Institut für Informatik
Friedrich-Schiller-Universität Jena, Germany
{beckstein,dittrich,erfurth,fey,koenig,mundhenk,sack}@informatik.uni-jena.de

Abstract

*Handling highly dynamic scenarios as they arise in emergency situations requires lots of semantic information about the situation and an extremely flexible, self-organizing IT infrastructure that provides services that can be used to manage the situation. We show that a distributed meta level architecture is particularly suited for the implementation of such a self-organizing grid of services. This architecture (*SOGOS*) distinguishes between an object level and a meta level. The middle ware processes of the grid are running on the object level. The meta level defines an explicitly and declaratively represented dynamic meta model that provides the semantics for the object level processes. Additionally, this level runs processes that plan, supervise and control mobile agents on the object level. The levels are linked together by reflection processes that ensure that relevant changes on the object level are reflected in the meta model and vice versa. The corresponding reflection principles provide the basis for the implementation of the self-organizing mechanisms that govern the overall system.*

Keywords: Adaptive Grid, Meta Level Architecture, Planning, Mobile Agents, Self-Organization, Semantic Services, System Dynamics, Trust

1. Introduction

The integration of diverse computational devices and services in dynamical environments is an essential prerequisite for any information society. Situations where this integration becomes a matter of death and life arise in times of crisis, when disaster has to be managed. Crises are complex situations where an information technological infrastructure has to be quickly built up in order to gather information and coordinate varieties of mobile workers.

In most major emergencies an appropriate information infrastructure is not available — e.g., due to a lack of technology or because major parts of the present communication infrastructure have been destroyed. But disaster management, nevertheless, first of all needs to collect and aggregate all the data that might be relevant for the situation at hand. This data includes stationary or mobile field sensor data ranging from environmental data like temperature, barometric pressure, humidity, or water level measurements to technical network data like the quality of radio reception or routing information, or data which is directly targeted at human beings like live video or audio streams. The desired result of this aggregation process is a global map of the crisis situation that can be used for decision support to coordinate disaster relief. In return, aggregated data can be distributed back to intermediate nodes providing multi modal information and support for distributed, mobile emergency teams.

For this situation assessment a service infrastructure is needed that amongst others fulfills typical grid objectives: nodes with direct access to sensor data have to communicate, intermediate nodes have to form a reliable transport network, routing nodes have to filter incoming sensor data in order to eliminate duplicates and in order to direct different sensor data to various service requesters, and service aggregation nodes have to merge the incoming sensor data into the desired map of the global situation (cf. figure 1). Finally, all these conditions must be guaranteed for a potentially damaged network of high volatility, where service providers are not necessarily static but mobile and nomadic.

Constructing an emergency response grid requires bridging organizational boundaries. Such a grid has to be built based on computer systems which are possibly heterogeneous both on the side of hardware and operating systems. The available hardware is unreliable and highly dynamic — components are added and removed over time — but the grid nevertheless must be able to continuously provide services that are essential for the management of the crisis — like reliable communication, integration and distribution of situation knowledge, decision support (e.g. for the planning

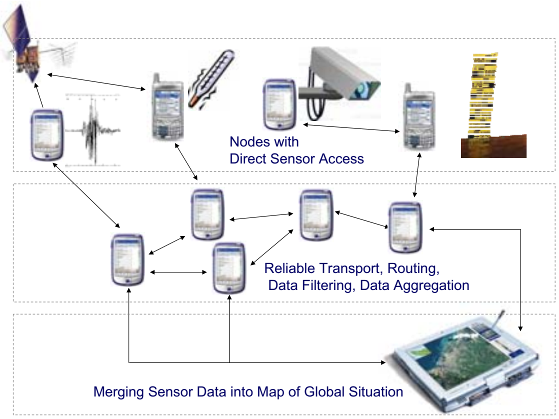

Figure 1. The emergency response grid

of evacuations) or the provision of high-performance computing power to local field workers for real-time disaster assessment.

In order to deliver services reliably on an unreliable, heterogeneous and changing hardware infrastructure, self-organization, which enables autonomy and minimizes maintenance, is crucial. A system can only behave in a self-organized manner if certain architectural prerequisites are fulfilled. In this paper we describe a new architecture called SOGOS (Self-Organizing Grid of Services), which is especially suited to support self-organization.

A test bed for this architecture are catastrophe scenarios of the kind mentioned — they guide the development and evaluation of new methods of self-organized service provision and give rise to the identification and formal specification of new kinds of services that are desirable in emergency situations.

2. The SOGOS Architecture

An effective way of handling highly dynamic scenarios like the specified ones apparently requires lots of semantic information (meta data) — about the involved organizations, the roles, rights and capabilities of the participating agents and about the normal way they interact in problem solving. This information — the meta model — is used to coordinate the activities of the overall system.

Conventional coordination systems (e.g. classical workflow systems) use a fixed and only implicitly defined meta model — a major problem when the system moves into an unexpected state and must be adapted at runtime like in our disaster recovery scenario.

We therefore propose to build the self-organizing grid of services according to a so called meta level architecture [49, 41] discriminating between an object level and a meta level. The middle ware processes of the grid are running on the object level. The meta level defines an explicitly and declaratively represented dynamic meta model that provides the semantics for the object level processes. Additionally, this level runs processes that plan, supervise and control the object level processes [8, 5].

Ideally the object and the meta levels are causally connected i.e. linked together by so called reflection processes that ensure that relevant changes on the object level are reflected in the meta model and vice versa [10, 8].

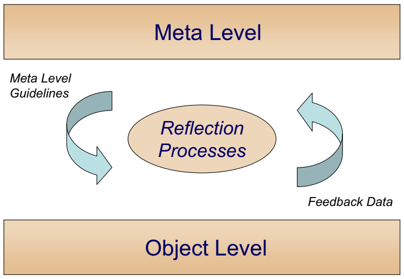

Figure 2. SOGOS Meta Level Architecture

2.1. The Meta Level

We consider a service as being a well defined range of functions in the sense of a general attendance ranging from low level fixed services provided on a local scope up to the compliance of global objectives that can only be achieved by collaboration of the system grid at large.

Primary objective of the meta level is to solve the problem of mapping high level aims to low level services provided by the middle ware layer. To achieve this objective it is of particular importance to map formal domain knowledge into local interaction rules that control the interplay of services for guiding the self-organization process. Heterogeneous services with inconsistent service descriptions have to be aligned, nomadic and volatile services must be scheduled.

Describing Global Aims If global aims are to be mapped to services, the first thing that is needed is a possibility to formulate these aims. To be usable, such a formulation needs to be declarative, i.e., the user should specify what

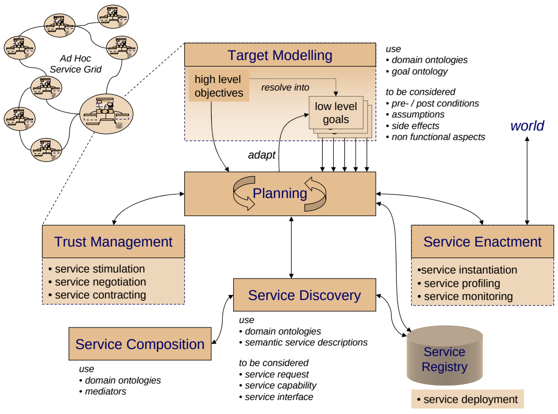

Figure 3. SOGOS **Meta Level**

functionality or information he is interested in not how this functionality/information should be provided.

A first step in this direction is the request formulation offered by our service description language DSD [31]. Here, users specify which effects the service they are looking for should have. Effects are described as fuzzy sets, where the degree of the membership of a certain effect to the set reflects how well this effect matches user preferences. This mechanism allows to fully capture user preferences in a way interpretable by a discovery algorithm. It thus becomes possible to fully automatically match requests and offers [29].

What is still needed are constructs to describe and evaluate non-functional aspects of aims. We believe that these will play an important role in settings like the one described above.

The Planning Process The entire meta level architecture is based on a process of distributed planning and replanning. Starting with the target modeling it has to be considered, how to resolve high level objectives into a number of single goals that can be achieved on a lower level of abstraction. This process has to make use of an appropriate domain ontology describing the area of interest as well as of a general model that describes the process of splitting goals into subordinate targets featuring a lower level of abstraction. A close interaction with the service composition component is needed.

If a relayed low level goal cannot be achieved directly, a new planning process will be activated in the related sub network, resolving the assigned objective to appropriate lower level goals. This process works closely with the service enactment component.

Service Discovery If a lower level objective cannot be fulfilled with already available resources, the service discovery process is triggered. Guided by appropriate domain ontologies semantic service descriptions have to be evaluated considering service requirements, service capabilities and available service interfaces. If no suitable service can be found the system tries to achieve a service composition that starts from the initial state resulting in the projected target state.

The functionality required here can be provided by the DSD matcher [30, 32]. In DSD, service offers are described as sets of required preconditions and achieved effects. By the introduction of variables, DSD allows to seamlessly integrate the descriptions of which functionality is provided and what the message flow should look like (i.e., which inputs a service expects and which outputs it is able to produce) [33]. DSD descriptions of both offers and requests (see above) are basically directed graphs. The matcher traverses the request graph and tries to find and configure matching offers by implementing a subset test.

As of now, DSD is restricted to rather simple choreographies where a negotiation phase is followed by an atomic service execution (i.e. no further interaction between service offerer and requester is needed during service execution). Here, extensions are clearly needed.

Service Composition Service composition again requires planning. Matching necessary preconditions the estimated outcome of services has to be estimated, postconditions and side effects must be considered, heterogeneous service descriptions have to be aligned. This process continues until the focused objective can be matched.

In addition to the chaining of services described above, other types of service composition need to be considered [35]. Particularly challenging are two cases: A service offers the needed effect, however, additional knowledge on the instance level is needed to invoke the service. Here, specialized knowledge services are needed. The second case is a request that contains several effects that are dependent on one another.

Trust Management Only trustworthy services complying with the prerequisite objectives should be considered during the planning process. To achieve this desirable property, information about which services are trustworthy is needed. Our previous work [44] has shown that is non-trivial to obtain and distribute this information in a dynamic decentralized environment. We have developed an evidence-based reputation system, where recommendations about nodes is backed by non-reputable tokens [42]. Locally, a node uses probabilistic belief revision based on transactional and social evidences [43] context information to estimate the likelihood that another will cooperate in a given transaction. This system ensures that it is more attractive for nodes to cooperate than to defect.

Service deployment has to be stimulated for services previously selected by service discovery and service composition. A service deployment agreement has to be negotiated and contracted.

Service Enactment and Service Registry The services previously selected by service discovery and service composition are passed to the service enactment process to be instantiated. The deployment of services will be recorded by the service registry. Service execution will be monitored to trigger dedicated failure and exception management.

2.2. The Object Level

The task of the object level is to form a basic infrastructure for the meta level. This is characterized by primarily two requirements: First to provide an appropriate hardware infrastructure consisting of static and mobile processing nodes including the set-up of reliable links to connect these nodes. Second to provide typical middle ware services both for the fulfilling of object level internal tasks and for the communication with the meta level. In particular the initialization of the hardware infrastructure is a challenging task because a heterogeneous and possibly unstable environment has to be taken into account. A lot of different and highly dynamical aspects need to be adapted and managed by the grid itself, e.g. locating and managing resources, load balancing, dynamic system adaptation to external context changes or based on internal meta level guidelines, etc. Due to the high fluctuations of nodes, think of a disaster scenario, which concerns in principal all nodes the control for all meta level tasks can not be organized by a central master node. Consequently the only chance to come to a reliable solution despite of a given unstable environment is to realize the grid middle ware based upon a self-organizing approach.

Besides we want to focus that the self-organizing characteristic of the object level middle ware is not only limited to the object level's internal side. The object level gives also feedback to the meta level via the reflection processes whether the defined meta level objectives are reachable or not. It also reports infrastructure changes that will effect the meta level. Finally this results in a self-organizing middle ware for the setup of a context-aware grid structure. The exciting question is how do we want to realize these necessary self-organizing features in detail?

State-of-the art is that current grid structures and middle ware have more in the focus to set-up so-called virtual organizations to overcome administrative organizations or barriers. The situation of a highly fluctuation of the resources is to our knowledge not considered in depth so far. The research work summarized with the words "Next Generation Grids" [17, 51] addresses exactly this problem. Nevertheless current middle ware solutions offer a large potential for providing basic services like certification (GSI module in GLOBUS [20]), monitoring or standardized access to infrastructure data (MDS module in GLOBUS).

Therefore, for the realization of the meta level middle ware, we base on a combination and enhancement of existing de-facto standard toolkits for grids like GLOBUS [19] and techniques used in CONDOR [48] with new technologies, e.g. mobile agents [12]. We favor mobile agent technology, a new paradigm for distributed programming, to meet the dynamic features of our grid environment. As we mentioned already above current grid toolkits deliver the prerequisites to set-up more static grid structures to overcome administrational barriers. We want to investigate in detail which of these mechanisms can be used, and which have to be supplemented by us with agents to meet the requirements necessary for the creation of self-organizing adaptive grids. To combine current technology with new mechanisms we orientate to the CONDOR-G concept [22], an adaptation for GLOBUS to build up CONDOR pools based on underlying GLOBUS Grids. We define SOGOS-G, an adaptation for GLOBUS for self-organizing ad-hoc Grids based on mobile agents as core of our self-organizing middle ware. By this we can use services from existing technologies, which are also useful for our purposes in SOGOS, and enhance them to meet the mentioned highly dynamic aspects. Mobile agents will be integrated with these standards to provide additional capabilities, especially to support platform mobility, decentralization, the autonomic formation of ad-hoc organizational structures, etc. This adds a certain learning capability to the system and provides for the study of emerging grid behavior in specific environmental situations.

The resulting structure of the object level is an agent-managed grid (see figure 4). The task of the mobile agents (Agent Layer) is to continuously discover "grid-enabled" resources (single computer devices or grids that wish to join), to interconnect them regarding meta level directives, to manage infrastructure changes, and to provide basic infrastructure services for the meta level, e.g. transaction management or service discovery. For the Agent Layer we utilize the mobile agent toolkit TRACY, which is a result of our research in that area [11, 18] and which reached product status by a small start-up company.

The abilities of such a grid depend on the abilities and possibilities of agents. One goal is to achieve a generic grid architecture which is customizable for certain general needs. The target architecture is a result of ability selection or module selection made for a special application scenario of the grid middle ware. With this generic grid architecture we are able to establish a system family for grids.

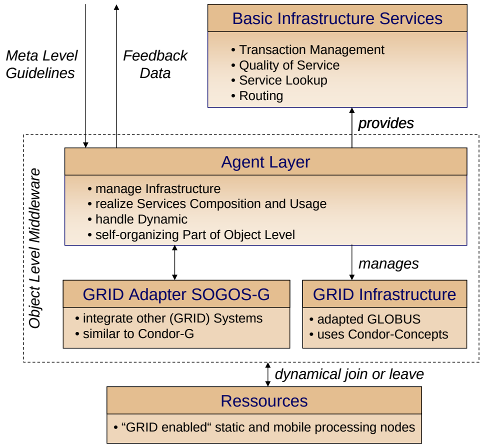

Figure 4. SOGOS Object Level: Agent-managed Grid

Looking at the most interesting and typical scenarios at meta level and the resulting, situation adequate grid architectures at object level we try to identify patterns. These patterns will be used to specify meta level guidelines which provide the "essence" and prerequisites for these already successful grid architectures in a "precompiled" manner. This will enable the end user to select such a pattern in a specific situation and the grid-based system to build its architecture faster and with higher reliability. Thus, the aspect of self-organization remains active in the grid (patterns are still generic), but is supported and guided, leading to higher reliability and performance. This is especially important in real-world scenarios dealing with disaster management and other critical situations.

3. Open Problems / Research Challenges

An important aspect of our scenario is that the dynamics is only partially under the control of SOGOS-processes. Even more, there is no central control nor an instance with guaranteed global knowledge. Therefore planning on the meta level can not be perfect but must rely also on the (self-organizing) dynamics on the object level. The challenge here is to bring together the symbolic planning on the meta level an the self-organizing dynamics of the underlying distributed system.

In order to control the self-organizing dynamics of the grid, the system must have a representation of itself, which can be used for reflection and reasoning purposes. Thus, the meta model must be explicitly and declaratively represented. In addition, it must be designed in a way that allows a practical implementation, because it serves as the central data structure for the components that plan and control the dynamics of the system.

Planning Planning means to devise directives for the agents and services how to act. Our agents act concurrently and locally distributed in a changing and partially unknown world. Therefore, a multi-agent temporal planning framework has been developed that supports concurrent execution, non-deterministic activities, just in time completion and dependency directed replanning [4, 28, 40].

Since no component of the system has complete knowledge about the system at large, planning is done in a distributed way and on the basis of explicitly represented reasonable assumptions about the missing knowledge. The resulting local plans are then executed independently of each other and readjusted when conflicts arise or critical assumptions prove to be false [3, 7, 6].

This framework along with a suitable declarative representation of the relevant object level aspects plays a key role in the implementation of the self-organization principles that govern the overall system [9].

Controlling the Dynamics In order to control the system's dynamics we must be able to monitor the system's behavior and evaluate its performance. Non-linear dynamical systems theory [25] offers a set of concepts that can characterize dynamical properties of the grid. The (asymptotic) stability of an operational mode, for example, is defined as the ability to compensate *small* perturbations of the state of the system. The concept of robustness captures *structural* changes of the system [27]. Measures like those for stability and robustness will not only be able to monitor the running system, but can also be taken to investigate its scalability [16] and adaptivity in a quantitative manner.

In passing we note that it is far from trivial to setup local planning rules that lead to a globally robust behavior, which is defined by the users on the symbolic level (i.e. meta level) in an appropriate modal or description logic. One promising way to achieve this is to apply learning or optimization techniques inspired by biology or sociology [39].

Integration For each of the relevant system aspects, such as time, goals, dependencies, terminologies, actions, system dynamics, and trust we have either developed or adapted special purpose modules, which can handle just this single aspect. For our application these single aspect modules alone are not sufficient, but have to be integrated into a hybrid system respecting all aspects at the same time. In order to achieve this, we have to design protocols for information exchange among participating modules, criteria that trigger information exchange, and rules for the integration of information received from other modules.

4. Related Work

Self-organization, service oriented computing, and the grid are very active research areas. It is widely recognized that a combination of the techniques developed by these three communities would be of great benefit. The large number of workshops and conferences (e.g., the SOAS, IEEE Web Intelligence, GCC and GSEM conference series, the SelfMan, SGT and SeNS workshop series) aiming to bridge the areas bears witness to this desire. However, a look at the publications produced by these and other events shows that the research community is far from reaching this goal: Most research papers do not seriously attempt to overcome the boundaries between the communities; rather, they present results from merely one of the areas. The question how approaches from the different communities can be combined for more efficient, robust and flexible solutions remains mostly unanswered.

Of course, first steps towards such an integration are already taken — see e.g., ASG, the EU-funded Adaptive Services Grid [2] or the KW-F Grid [36]. Another one is AKOGRIMO [1, 50], an European funded project aiming to architect and prototype a Next Generation Grid based on Open Services Grid Architecture (OSGA [21]) which exploits and closely co-operates with evolving Mobile Internet infrastructures. The concept of the project is to evaluate the derived Mobile Grid through testbeds, among others chosen from applications from the domain of crisis management, e-learning, and e-health.

The mentioned approaches, however, are typically rather pragmatic in nature and lack a fundamental investigation of the methods and mechanisms required. A pragmatic approach, of course, is a valuable first step towards a successful combination of these areas, but it is not enough to fully exploit the synergies arising from a more integrated view. In fact, a more basic approach is needed. This is also one of the conclusions drawn at a recent Dagstuhl seminar titled ”Semantic Grid — Convergence of Technologies” attended by researchers from all three areas [52]. At this seminar, a number of important areas that need fundamental work have been identified. Among these are trust and security, adding semantics to the grid via semantic services and supporting self-organization to obtain more flexible and heterogeneous grids:

Self-organization Currently, a plethora of projects address self-x properties of computer systems from a number of different angles, see e.g the UKCRC Computing Grand Challenge Manifesto [13] for a discussion on why self-organization is needed and what the challenges involved are. SOGOS is developed at ICS, the Jena competence center on the self-organized integration of computing and information systems [47] that bundles a number of projects addressing issues in this area ranging from mobile agents [12], incentive schemes to biologically inspired methods and architectures for organic computing. The ICS is currently performing active research on further aspects that are highly relevant to SOGOS, like semantics of dynamical systems and temporal and multi-agent planning.

Semantic web services Most pertinent to this area are the US-based OWL-S project [46] and the work done by the EU-funded WSMO working group [53]. Through participation in the latter and a number of own projects, ICS is at the forefront of research in this field. One of the active research aspects in this area related to SOGOS are complex service choreographies and smart service distribution.

Security and Trust in dynamic, open systems This is an area that has only recently gained a lot of interest. See, e.g., the work done by the member of the IST iTrust Working Group [26] for an overview of issues addressed. Mechanisms for trust formation in such systems have been extensively studied by members of ICS [15]. SOGOS-related research in this area currently focuses related security issues [45, 37].

Grid Computing For an excellent overview of ongoing work on grid computing, see [24]. On a smaller scale, ICS possesses experience with the issues addressed there. Current work concentrates on the integration of existing standards implementations [23]. E.g. small-sized Grid structures consisting of Linux Beowulf cluster and a symmetric multiprocessor machine were used for the parallel simulation of cellular automaton (see [34]).

In our opinion, the key to an approach combining all four areas is reflection based on autonomous planning that guides the self-organization of the system and thus enables efficient service provisioning in dynamic environments. Here, our approach differs considerably from existing attempts and promises tremendous potential for innovation.

5. Conclusions

We have shown how a distributed meta level architecture can be used as a basis for a self-organizing service grid. This architecture distinguishes between an object level and a meta level. The object level is composed of a grid infrastructure based on GLOBUS. This infrastructure supports small and static hardware clusters that provide virtual computation services. An emergency grid however must reflect the complex dynamics of mobile, heterogeneous, and highly volatile groups of autonomous agents. For this reason we currently push the integration of our grid and agent platform in order to obtain a sound hybrid middle ware for the proposed self-organized service architecture.

The meta level is responsible for the mapping of high level aims to low level services provided by the middle ware layer. For the description of global aims we have developed the semantic service description language DSD. DSD already facilitates service matching and service discovery. It also provides all language facilities necessary for service composition. We also have developed a distributed and assumption based multi agent planner, which is currently adapted for the composition of heterogeneous services. A permanently changing system environment requires trustworthy services. For this purpose we have developed an evidence based reputation system using probabilistic belief revision. This reputation system currently is being integrated with the planning component.

Once we have accomplished the horizontal integration on both levels of the meta level architecture, the essential research objective we are currently working on is the vertical integration, the design and implementation of the reflection mechanisms that connect the object and the meta level.

Only if both, the horizontal and the vertical integration are successfully accomplished, a system will result that can self-adapt to a rapidly changing environment. When disaster strikes, self-organization becomes essential. A self-organized grid of services like SOGOS will then be a sound basis for any kind of disaster relief.

References

[1] AKOGRIMO. Access to knowledge through the grid in a mobile world, `http://www.akogrimo.org`.

[2] ASG. Adaptive service grid, 2005, `http://asg-platform.org/cgi-bin/twiki/view/Public`.

[3] C. Beckstein, R. Fuhge, and G. K. Kraetzschmar. Supporting Assumption-based Reasoning in a Distributed Environment. In K. P. Sycara, editor, *Proc. of the Twelfth Workshop on Distributed Artificial Intelligence, Hidden Valley Ressort, Pennsylvania*, pages 3–17, May 1993.

[4] C. Beckstein and T. Geisler. An Application-independent Support system for Assumption-based Temporal Reasoning. In *Proc. of the 1994 Workshop on Temporal Representation and Reasoning (TIME-94), Florida AI Workshops and Symposiums (FLAIRS-94), Pensacola Beach, Florida*, May 1994.

[5] C. Beckstein and J. Klausner. A meta level architecture for workflow management. *Journal of Integrated Design and Process Science*, 3(1):15–26, 1999.

[6] C. Beckstein and J. Klausner. A planning framework for workflow management. In *Proceedings of the International Workshop on Intelligent Workflow and Process Management: The New Frontier for AI in Business (held as part of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI-99)*, Stockholm, 1999.

[7] C. Beckstein, G. K. Kraetzschmar, and J. Schneeberger. Distributed plan maintenance for scheduling and execution. In C. Bäckström and E. Sandewall, editors, *Current Trends in AI Planning, EWSP'93 — Second European Workshop on Planning, Frontiers in Artificial Intelligence and Applications Series*, pages 74–86, Amsterdam, Oxford, Washington D.C., Tokyo, 1994. IOS Press.

[8] C. Beckstein, R. Stolle, and G. Tobermann. Metaprogramming for generalized horn clause logic. In *Proceedings of the Workshop on Metaprogramming and Metareasoning in Logic, META-96 (held in association with the 1996 Joint International Conference and Symposium on Logic Programming, JICSLP-96)*, pages 27–42, Bonn, 1996.

[9] C. Beckstein, G. Tobermann, and R. Stolle. Declarative meta level control for logic programs. In *Proceedings des Ersten Russisch-Deutschen Symposiums zu Intelligenten Informationstechnologien und Expertensystemen (anläßlich des Internationalen Forums für Informatisierung, IFI-95)*, Moskau, November 23rd–29th 1995.

[10] K. A. Bowen and R. A. Kowalski. Amalgamating language and metalanguage in logic programming. In K. L. Clark and S. A. Tärnlund, editors, *Logic Programming*, pages 153–172. Academic Press, London, 1982.

[11] P. Braun, J. Eismann, C. Erfurth, and W. R. Rossak. Tracy - a prototype of an architected middleware to support mobile agents. In *Proceedings of the 8th Annual IEEE Conference and Workshop on the Engineering of Computer Based Systems (ECBS)*, Washington D.C., 2001.

[12] P. Braun and W. R. Rossak. *Mobile Agents: Basic Concepts, Mobility Models, and the Tracy Toolkit*. Morgan Kaufmann, 2004.

[13] CGC Manifesto. UKCRC Computing Grand Challenge Manifesto, `http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/Manifesto/manifesto.pdf`.

[14] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 2000.

[15] DIANE. DIANE Project, `http://hnsp.inf-bb.uni-jena.de/DIANE`.

[16] P. Dittrich, T. Kron, and W. Banzhaf. On the formation of social order — modeling the problem of double and multi contingency following Luhmann. *Journal of Artifical Societies and Social Simulation*, 6(1), 2003.

[17] Special theme: Grids, the next generation. www.ercim.org, ERCIM News No. 59, Dec. 2005.

[18] C. Erfurth. *Proaktive autonome Navigation für mobile Agenten*. PhD thesis, Friedrich-Schiller-Universität Jena, Fakultät für Mathematik und Informatik, 2004.

[19] I. Foster. Globus toolkit version 4: Software for service-oriented systems. In *IFIP International Conference on Network and Parallel Computing*, number 3779 in LNCS, pages 2–13. Springer-Verlag, 2005.

[20] I. Foster, C. Kessekan, G. Tsudik, and S. Tueckel. A Security Architecture for Computational Grids. In *Proceedings of 5th ACM Conference Computer and Communications Security*, pages 83–92, 1998.

[21] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration, June 28 2002.

[22] J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke. Condor-G: A computation management agent for multi-institutional grids. *Cluster Computing*, 5:237–246, 2002.

[23] Globus Grid. Globus Grid, `http://www.globus.org/`.

[24] GridCoord. GridCoord, Survey of Activities in Universitíes and Research Labs, Deliverable D.3.1.2. of GridCoord, November, 2005, `http://www.gridcoord.org/`.

[25] C. Heij, A. C. Ran, and F. v. Schagen. *Introduction to Mathematical Systems Theory: Linear Systems, Identification and Control*. Birkhäuser, 2006.

[26] IST Working Group. iTrust Working Group, `http://www.itrust.uoc.gr/`.

[27] G. Jetschke. *Mathematik der Selbstorganisation*. Vieweg, Braunschweig, 1989.

[28] J. Klausner. *Planen und intelligentes Workflowmanagement*. PhD thesis, Universität Jena, Institut für Informatik, 2001.

[29] M. Klein and B. König-Ries. Combining query and preference - an approach to fully automatize dynamic service binding. In *Short Paper at IEEE International Conference on Web Services*, San Diego, CA, USA, July 2004.

[30] M. Klein and B. König-Ries. Coupled signature and specification matching for automatic service binding. In *Proc. of the European Conference on Web Services (ECOWS 2004)*, Erfurt, Germany, September 2004.

[31] M. Klein and B. König-Ries. Integrating preferences into service requests to automate service usage. In *First AKT Workshop on Semantic Web Services*, Milton Keynes, UK, Dezember 2004.

[32] M. Klein, B. König-Ries, and M. Müssig. What is needed for semantic service descriptions - a proposal for suitable language constructs. *International Journal on Web and Grid Services (IJWGS)*, 1(2), 2005.

[33] M. Klein, B. König-Ries, and P. Obreiter. Stepwise refinable service descriptions: Adapting DAML-S to staged service trading. In *Proc. of the First Intl. Conference on Service Oriented Computing*, pages 178–193, Trento, Italy, December 2003.

[34] M. Komann, C. Kauhaus, and D. Fey. Calculation of single-file diffusion using grid-enabled parallel generic cellular automata simulation. In Martino et al. [38], pages 528–535.

[35] U. Küster, M. Stern, and B. König-Ries. A classification of issues and approaches in automatic service composition. In *Intl. Workshop WESC 05*, 2005.

[36] KW-F. KW-F grid project, `http://www.kwfgrid.net/main.asp`.

[37] Manet. Mobile Ad hoc Network Security, `http://csrc.nist.gov/manet/`.

[38] B. D. Martino, D. Kranzlmüller, and J. Dongarra, editors. *Recent Advances in Parallel Virtual Machine and Message Passing Interface, 12th European PVM/MPI Users' Group Meeting, Sorrento, Italy, September 18-21, 2005, Proceedings*, volume 3666 of *Lecture Notes in Computer Science*. Springer, 2005.

[39] H. J. Müller, T. Malsch, and I. Schulz-Schaeffer. Socionics. Introduction and Potential. *Journal of Artificial Societies and Social Simulation*, 1(3):., 1998.

[40] M. Mundhenk, J. Goldsmith, C. Lusena, and E. Allender. Complexity results for finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681–720, 2000.

[41] J. P. Nytun and A. Prinnz. Metalevel representation and philosophical ontology. In *Workshop on Philosophy, Ontology, and Information Systems (held as part of the Eighteenth European Conference on Object Oriented Ppogramming, ECOOP-04)*, Oslo, 2004.

[42] P. Obreiter. A case for evidence-aware distributed reputation systems. In *Second International Conference on Trust Management (iTrust'04)*, pages 33–47, Oxford, UK, 2004. Springer LNCS 2995.

[43] P. Obreiter, S. Fähnrich, and J. Nimis. How social structure improves distributed reputation systems - three hypotheses. In *Third Intl. Workshop on Agents and Peer-to-Peer Computing (AP2PC'04)*, New York, USA, 2004.

[44] P. Obreiter, B. König-Ries, and M. Klein. Stimulating cooperative behavior of autonomous devices - an analysis of requirements and existing approaches. In *Proceedings of the Second International Workshop on Wireless Information Systems (WIS2003)*, pages 71–82, Angers, France, 2003.

[45] OS Grid. Open Science Grid: Grid Security Incident Handling and Response Guide, `http://osg-docdb.opensciencegrid.org/0000/000019/002/OSG_incident_handling_v1.0.pdf`.

[46] OWL-S Working Group. OWL-S Working Group, `http://www.daml.org/services/owl-s/`.

[47] SOVIS. ICS Competence Center "Self Organisation", `http://hnsp.inf-bb.uni-jena.de/sos/`.

[48] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.

[49] F. van Harmelen. *Meta-level Inference Systems*. Research Notes in AI. Pitmann, Morgan Kaufmann Publishers, London, 1991.

[50] J. Wedwik, B. Viken, S. Wesner, R. Piotter, I. Mueller, T. Dimitrakos, G. Laria, C. Morariu, N. Inacio, P. Mandic, R. del Campo, and S. F. Gonzales. The state of the art of mobile grids. Technical report, 2005.

[51] S. Wesner. Towards an Architecture for the Mobile Grid. *information technology*, 47:336–342, Dec. 2005.

[52] WSMO. Web Service Modeling Ontology, `http://www.wsmo.org/`.

[53] WSMO. WSMO Working Group, `http://www.wsmo.org/`.