IEEE DISTRIBUTED SYSTEMS ONLINE 1541-4922 © 2006 Published by the IEEE Computer Society
Vol. 7, No. 3; March 2006

Works in Progress: The 2nd International Middleware Doctoral Symposium

Edward Curry, *The National University of Ireland, Galway Jacques Mossière*
Jacques Mossière, *Institut National Polytechnique de Grenoble*

Following the successful symposium at Middleware 2004, the 2nd International Middleware Doctoral Symposium took place at Middleware 2005. The symposium is a forum for an invited group of doctoral students; the students get to present their work, obtain guidance from mentors, and meet with their peers. The mentors are senior university or industry researchers, including many current or former members of the Middleware program committee.

The symposium aims to give students constructive criticism before their thesis defense and foster discussions related to future career perspectives. A similar series of doctoral symposia is held in connection with other conferences, including the International Conference on Object-Oriented Programming, Systems, Languages, and Applications; the European Conference on Object-Oriented Programming; the International Conference on Software Engineering; and the International Semantic Web Conference.

The symposium attracted many high-quality submissions, ensuring a competitive selection process. The program committee selected eight doctoral candidates to present their work:

- Abdelkrim Beloued, ENST-Bretagne (Context-Aware Replication and Consistency),
- Vladimir Dyo, University College London (Middleware Design for Integrating Sensor Networks and Mobile Devices),
- Etienne Antoniutti Di Muro, Università degli Studi di Trieste (Translucent Replication: Three Open Questions),
- Michael A. Jaeger, Berlin University of Technology (Self-Organizing Publish/Subscribe),
- Rüdiger Kapitza, University of Erlangen Nürnberg (EDAS: An Environment for Decentralized Adaptive Services),
- Sharath Babu Musunoori, Simula Research Laboratory (Quality-Driven Application Service Planning),
- Trevor Parsons, University College Dublin (Detecting Performance Antipatterns in Component-Based Enterprise Systems), and
- Kurt Schelfthout, Katholieke Universiteit Leuven, Belgium (Middleware That Enables Protocol-Based Coordination in Mobile Networks).

The previous issue of *IEEE Distributed Systems Online* featured summaries of the first four dissertations. Here we feature summaries of the final four.

Acknowledgments

We thank the mentor committee members for their time and effort: Pascal Déchamboux (France Télécom R&D), Thomas Gschwind (IBM Research, Switzerland), Cecilia Mascolo (University College London), David Rosenblum (University College London), Rick Schantz (Bolt, Beranek, and Newman Technologies), and Joe Sventek (University of Glasgow). We also gratefully acknowledge the support of BBN Technologies.

Edward Curry is a PhD candidate at the National University of Ireland, Galway. Contact him at edcurry@acm.org.

Jacques Mossière is a professor at the Institut National Polytechnique de Grenoble. Contact him at jacques.mossiere@inrialpes.fr.

EDAS: An Environment for Decentralized Adaptive Services

Rüdiger Kapitza, *University of Erlangen Nürnberg*
Franz J. Hauck, *University of Ulm*

Infrastructures for grid computing aim to unite a group of computers, servers, and storage systems as one large computing system. Resource management is a key issue, because it helps such systems efficiently and automatically distribute tasks on the grid. Current grid-computing infrastructures are designed to solve compute- and data-intensive tasks with a more or less static set of parameters. Unlike grid computing, EDAS (Environment for Decentralized Adaptive Services) focuses on distributed, user-accessible, community-supported, long-term services.[1,2] Examples of such services are Web servers, name servers, revision-control systems, and project-management services.

EDAS comprises three main components (see figure 1). A *service environment* is a virtual distributed-execution scope, which usually represents a project or service domain. It can host a number of distributed services, so it uses resources from multiple home environments and manages and supports service execution.

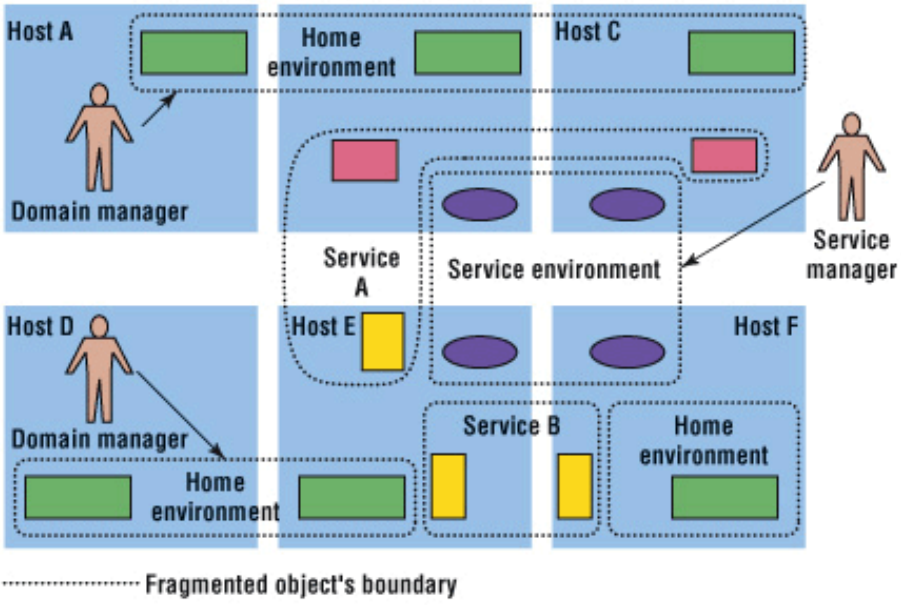

Figure 1. An Environment for Decentralized Adaptive Services. The main components are the home environment, service environment, and services.

A *home environment* is a set of nodes with dedicated resources assigned by the managers of an administrative domain. Thus, a home environment gives a share of resources to a service environment in order to host services.

A *service* can have a single service instance or be decomposed into components that form replicated, partitioned, and hierarchical services as well as peer-to-peer-based structures. Furthermore, a service can dynamically change its internal structure by replacing, removing, and creating components.

Resource management occurs in the home and service environments. EDAS decides on the initial placement and dynamic migration of service components, adapting to changing resource conditions (such as when new nodes are assigned to a home environment, a node crashes, home environments are added, or service resource demands change according to service usage). Because migration operations are expensive, EDAS tries to delay migration as long as possible.

We implemented EDAS on top of our AspectIX middleware system,[3] which is based on the fragmented-object model that Marc Shapiro originally introduced. This model provides objects that are truly distributed over multiple nodes, so that we can implement home environments, service environments, and services as distributed fragmented objects. Our development framework also lets us implement services so they can evolve from a single server instance to an actively replicated server.

Because we use a fragmented-object model, a service's client also gets a local service component (a fragment). So, service developers can influence how clients interact with the service. Our framework also helps service developers introduce client-specific code—for example, clients can transparently interact in a hybrid peer-to-peer architecture such as Napster.

You can email us for more information on EDAS and AspectIX (http://www.aspectix.org).

References

1. 1. R. Kapitza , F.J. Hauck and H. Reiser , "Decentralized, Adaptive Services: The AspectIX Approach for a Flexible and Secure Grid Environment,"*Proc. Int'l Conf. Grid Services Eng. and Management 2004 Conf.,* LNCS 3270, Springer, 2004,pp. 107-118.
2. 2. R. Kapitza and F.J. Hauck , "EDAS—Providing an Environment for Decentralized Adaptive Services,"*Proc. 2nd Int'l Doctoral Symp. Middleware* (DSM 05), ACM Press, 2005, pp. 1-5.
3. 3. H.P. Reiser , et al., "Integrating Fragmented Objects into a CORBA Environment,"*Proc. Net.ObjectDays,* Transit GmbH, 2003.

Rüdiger Kapitza is a PhD student in the Distributed Systems Group at the University of Erlangen-Nürnberg. Contact him at rrkapitz@cs.fau.de.

Franz J. Hauck is a professor in the Distributed Systems Laboratory at the University of Ulm. Contact him at franz.hauck@uni-ulm.de.

Quality-Driven Application Service Planning

Sharath Babu Musunoori, *Simula Research Laboratory, Norway*
Frank Eliassen, *Simula Research Laboratory, Norway*

As advancements in grid computing continue to support a variety of parallel and distributed systems, large-scale application scheduling is becoming a key concern. In particular, there's a growing need to support quality-sensitive applications, because they perform unacceptably if platform resources are scarce or if the deployment isn't carefully configured and tuned for the anticipated load.

We use *application service planning* to determine an application's configuration. The application comprises service components that represent their corresponding functionality. Our work focuses on determining how to configure this service composition in the grid environment such that all services involved have sufficient resources to deliver at least the minimum amount of quality an application requires. Service planning in a grid

environment thus maps a service composition's individual service components onto the underlying grid resources while satisfying the application's specified quality requirements.

To address this issue, we developed an open, reflective, quality-aware component architecture (QuA, http://www.simula.no/QuA) to investigate how to support the component-based application's quality characteristics during deployment and runtime (see figure 2). To find a resource configuration that satisfies the composed application service's quality objectives, we focused on solution techniques that provide a sufficient—but not necessarily optimal—solution. This is because the computational and communication resources in the grid environment can join and leave very often. Such variations in resource availability can destabilize the current running configuration. Fundamentally, this is an NP-hard problem. To configure the application, we use a QuA middleware platform element called a *service planner* (see figure 3). The service planner achieves the specified quality for the application when deployed in a computational environment.

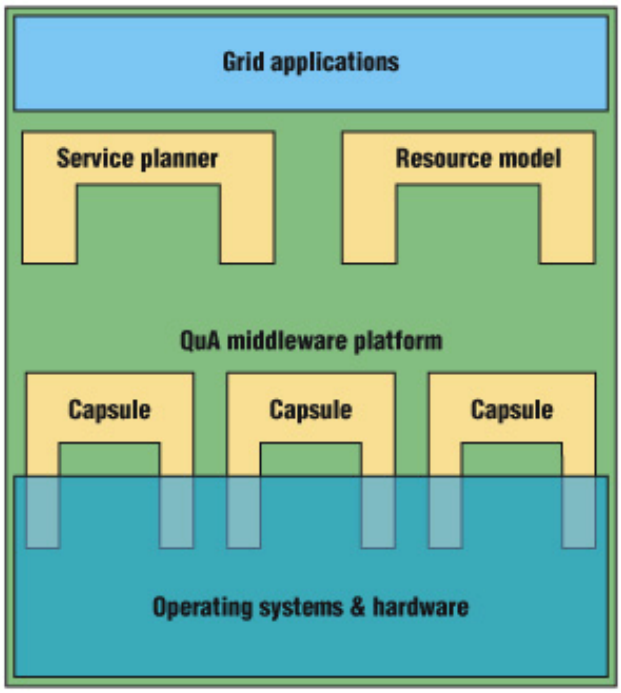

Figure 2. Our quality-aware component architecture (QuA).

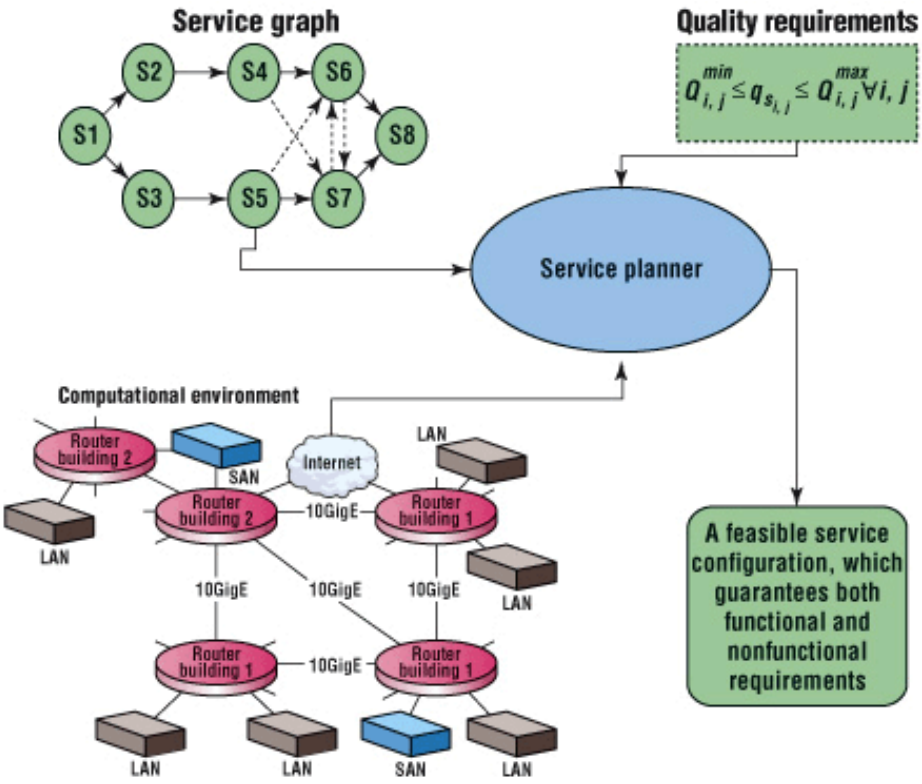

Figure 3. Quality-aware service planning. Here, quality requirements are specified as a simple expression indicating that acquired qualities should at least be equal or higher than minimum-acceptable qualities.

To improve service planning, we used a quality deviation model that minimizes the distance between the achieved and desired service-quality levels. Such a small change in quality objective function can significantly improve service-quality estimation while finding a compromise between the quality objectives. We verified this model using a real-time multimedia-object-tracking application service.[1]

By noting important limitations that result from scalability and multiple quality objectives, we adapted more advanced heuristic- and learning-based solution techniques. Recently, we designed and implemented a learning automata-based solution to find a useful mapping or partition. Using this approach, individual services in the service set interact with the grid environment to make service-configuration decisions. Our proposed learning automata-based solution establishes an autonomous-learning environment for the services.[2] So, as long as the current mapping or partition isn't feasible, the current partition's unsatisfied services move between the capsules.

In addition to performing service configuration at deployment, the resource managers running on grid nodes monitor changes in the underlying resource availability and notify the respective services. Our ongoing work will address how to use the information from the resource managers to help the services move to grid nodes with sufficient resources. Or, we might even enable the services to adapt variations in resource availability to guarantee the application's overall quality.

References

1. 1. S.B. Musunoori and F. Eliassen , "QoS-Aware Application Service Configuration in a Grid Environment,"*Proc. 9th Int'l Conf. Software Eng. and Applications* (SEA 05), ACTA Press, 2005,pp. 363-370.
2. 2. S.B. Musunoori and G. Horn , "A Fixed-Structure Learning Automaton Solution to the Quality Aware Application Service Configuration in a Grid Environment,"*Proc. 17th Int'l Conf. Parallel and Distributed Computing and Systems* (PDCS 05), ACTA Press, 2005,pp. 7-12.

Sharath Babu Musunoori is a PhD student in the Networks and Distributed Systems Department at Simula Research Laboratory, Norway. Contact him at sharath@simula.no.

Frank Eliassen is a professor in the Networks and Distributed Systems Department at Simula Research Laboratory, Norway. Contact him at frank@simula.no.

Detecting Performance Antipatterns in Component-Based Enterprise Systems

Trevor Parsons, *University College Dublin*
John Murphy, *University College Dublin*

Internet-enabled enterprise applications often fail to meet their performance requirements. This can lead to development delays or, even worse, loss of business due to customer dissatisfaction with sluggish or unreliable service. Inefficient enterprise systems are often the result of system developers making incorrect or suboptimal design decisions. Such decisions occur because today's enterprise applications are extremely large and complex, and developers often don't completely understand the entire application. Consequently, they often don't completely understand the performance implications of their design decisions, and as a result, systems frequently fail to meet their performance agreements.

Current performance tools don't help much, because they generally operate by profiling the running application, which can produce a substantial amount of information. Developers must then sift through this data to identify their systems' performance issues. When dealing with large multiuser applications,

where a typical load might be on the order of thousands of users, the amount of information produced during profiling can be unmanageable.

We introduce a framework for automatically detecting common performance design mistakes (antipatterns) in component-based enterprise systems. Our approach builds on performance tools by automatically analyzing the data that profilers collect. This new approach takes the onus away from the developer having to sift through performance data. Instead, our framework makes use of data-mining and statistical-analysis techniques that automatically summarize the data and find patterns of interest, which might reveal suboptimal design choices.

Figure 4 shows our framework architecture,[1] which comprises five main modules: monitoring, analysis, detection, assessment, and presentation. Although we believe we can apply the approach to any contextual composition framework, we're instantiating a prototype for Enterprise JavaBeans technology.

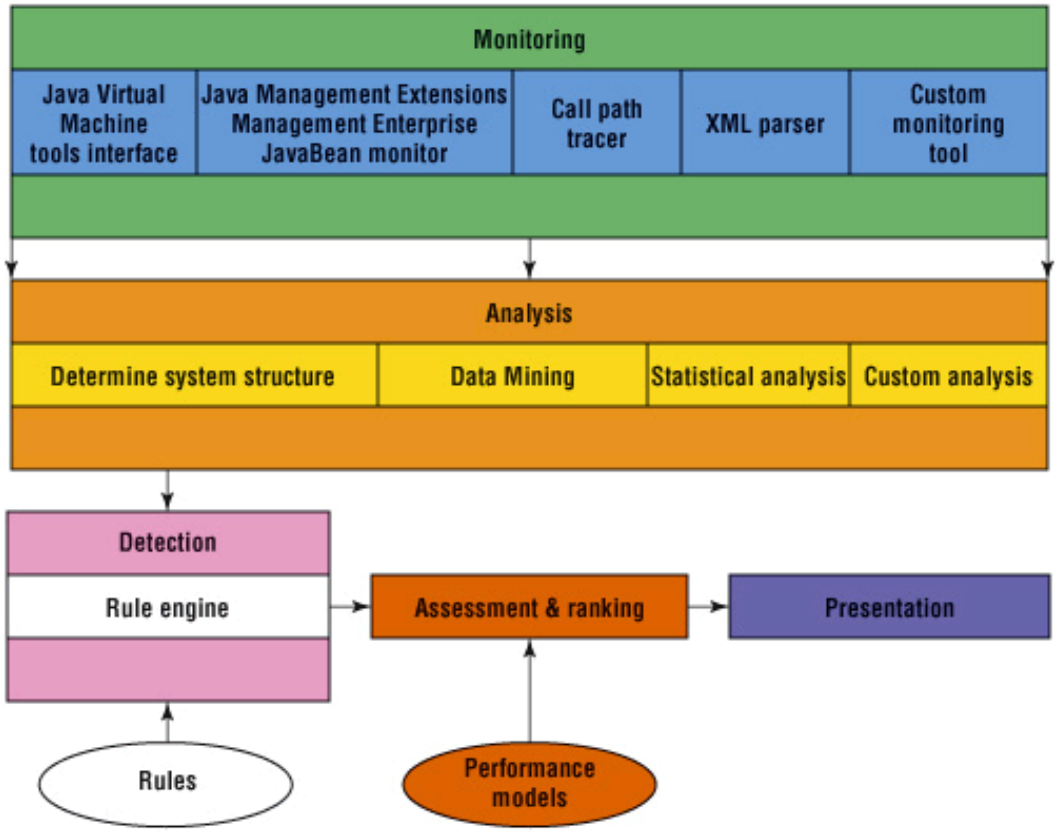

Figure 4. Our framework architecture.

The monitoring module monitors the running application and obtains a wide range of information including executed call paths (collected using the Compas (http://compas.sourceforge.net) monitoring tool), memory and CPU statistics, server resource usage, and structural and behavioral information.

The analysis module analyzes the information, collected during monitoring, offline to

- determine the overall system structure,
- apply different data-mining algorithms (such as association rule mining, clustering, and sequential rule mining) to find patterns of interest, and
- apply statistical analysis techniques to summarize the data.

The framework uses a rule engine to determine if any antipatterns exist in the system being tested. The data produced by the analysis module is loaded into a rule engine. Developers can easily write rules describing the performance antipatterns. The rule engine uses such rules to automatically analyze the data and detect antipatterns in the system. The antipatterns detected are assessed (and ranked) in terms of their performance impact using performance models and information collected during monitoring. Ranking the antipatterns lets developers focus on the ones that most affect performance. Finally, the antipatterns and their solutions are highlighted for the developer. Figure 5 shows the overall system structure of an application highlighted with an EJB antipattern. Developers can zoom in on particular antipatterns to view the antipattern details and corresponding solution.

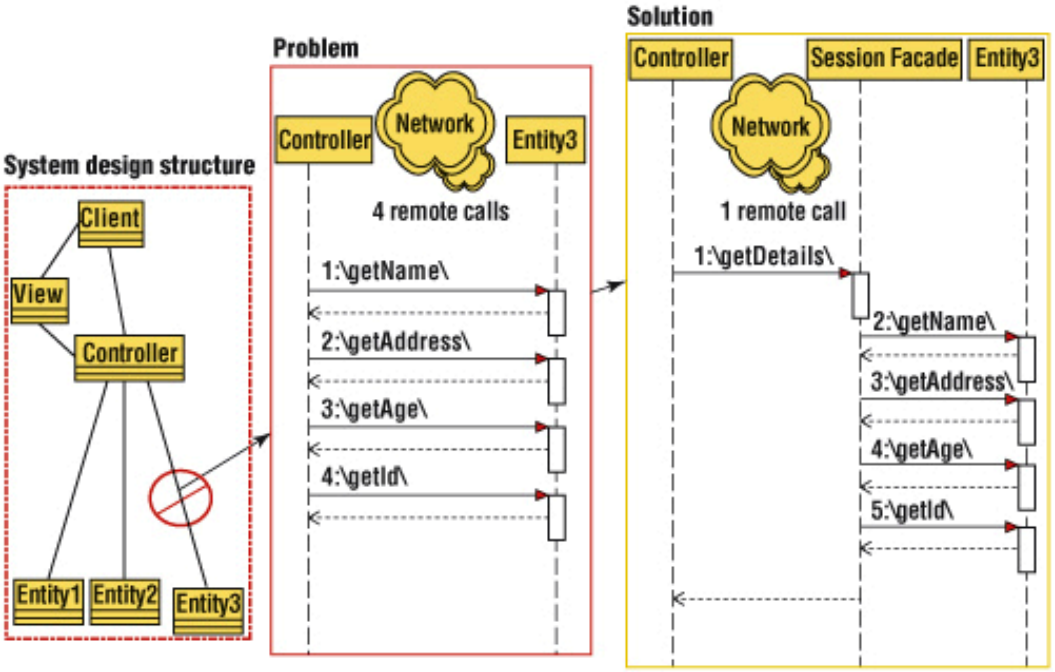

Figure 5. Overall system structure with antipatterns highlighted.

Reference

1. 1. T. Parsons , "A Framework for Detecting Performance Design and Deployment Antipatterns in Component Based Enterprise Systems,"*Proc. 2nd Int'l Middleware Doctoral Symp.* (MDS 05), ACM Press, 2005, art. no. 7.

Trevor Parsons is a PhD student in the School of Computer Science and Informatics at University College Dublin and a member of the University's Performance Engineering Laboratory. Contact him at trevor.parsons@ucd.ie.

John Murphy is a senior lecturer in the School of Computer Science and Informatics at University College Dublin and a member of the University's Performance Engineering Laboratory. Contact him at j.murphy@ucd.ie.

Middleware That Enables Protocol-Based Coordination in Mobile Networks

Kurt Schelfthout, *Katholieke Universiteit Leuven, Belgium*
Tom Holvoet , *Katholieke Universiteit Leuven, Belgium*

A major concern for application development in mobile networks is coordination between application components spread over multiple nodes. Coordination involves managing dependencies between activities, generally by executing interaction protocols among the application components that are performing the activities. This interaction is especially difficult in mobile networks owing to the computing nodes' physical mobility: interaction partners can appear or disappear at any time. For example, to avoid collisions, automatic vehicles must execute a mutual-exclusion protocol, but new vehicles can come into collision range at any moment.

Current coordination middleware for mobile networks focuses on efficient and reliable mechanisms for information exchange by adapting well-known middleware approaches from "traditional" distributed systems. For example, Ludger Fiege, Felix Gartner, Oliver Kasten, and Andreas Zeidler have proposed location-aware subscriptions for publish-subscribe middleware in mobile networks.[1] Gruia-Catalin Roman, Christine Julien, and Qingfend Huang propose a network abstraction to gather information from neighboring nodes given a declarative specification based on (among other things) location.[2] If an interaction protocol is necessary to

coordinate between application components, the application developer can use such existing middleware as a declarative mechanism to discover the necessary interaction partners for a given application component. However, the application developer must handle the ensuing protocol session without support, relying on a general-purpose network infrastructure that offers no support for dealing with mobility. The application developer must then solve issues related to mobility (such as changes in interaction partners) in an ad hoc and time-consuming way.

We developed a middleware for mobile networks that supports *roles* to encapsulate one partner's behavior in an interaction protocol. The middleware offers two services. First, application components can set up a new interaction session with components on other nodes by declaring that they will play a role in the session. The *role-activation service* then activates the necessary roles on the other nodes in the network given a declarative specification—for example, based on location. In the collision-avoidance example, a vehicle wanting to cross an intersection activates a "request-to-cross" role and asks to activate a new session with "give-permission" roles on other vehicles near the intersection (see figure 6a and 6b).

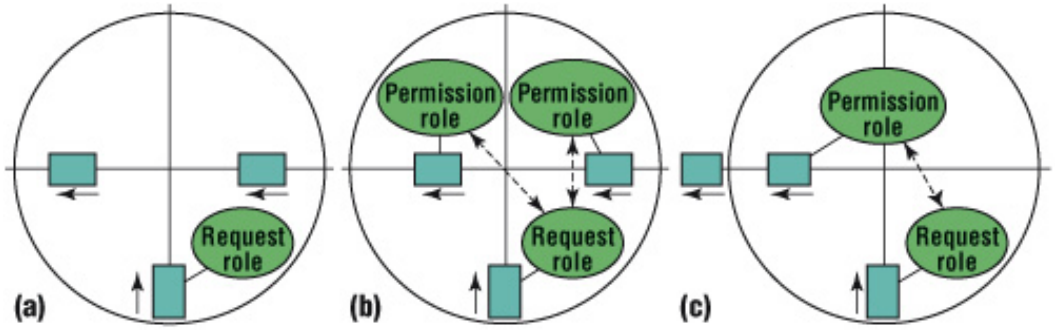

Figure 6. A scenario with three vehicles avoiding collisions: (a) A vehicle wanting to cross an intersection activates a "request-to-cross" role and asks to activate a new session with "give-permission" roles on other vehicles near the intersection; (b) the middleware activates "give-permission" roles on vehicles near the intersection, and the activated roles execute the protocol; (c) the middleware notifies the "request-to-cross" role if a vehicles leaves the intersection.

The second service the middleware provides is a *group maintenance service*. While the roles are executing the interaction protocol, the middleware monitors the network and notifies the roles of any changes in interaction partners. For example, it notifies the "request-to-cross" role if a new vehicle approaches the intersection or if a vehicle leaves the intersection (see figure 6c).

We applied the middleware in a real-world application (http://emc2.egemin.com) involving automatic-guided-vehicle control in for warehouses. More information appears elsewhere (http://www.cs.kuleuven.be/~kurts).[3]

References

1. 1. L. Fiege , et al., "Supporting Mobility in Content-Based Publish/Subscribe Middleware,"*Proc. ACM/IFIP/Usenix Int'l Middleware Conf.,* LNCS 2672, Springer, 2003, pp. 103-122.
2. 2. G.-C. Roman , C. Julien and Q. Huang , "Network Abstractions for Context-Aware Mobile Computing,"*Proc. 24th Int'l Conf. Software Eng.,* ACM Press, 2002,pp. 363-373.

3. 3. K. Schelfthout , D. Weyns and T. Holvoet , "Middleware for Protocol-Based Coordination in Dynamic Networks,"*Proc. 3rd Int'l Workshop Middleware for Pervasive and Ad-hoc Computing* (MPAC 05), ACM Press, 2005,pp. 1-8.

Kurt Schelfthout is a PhD student in the DistriNet research group in the Department of Computer Science at Katholieke Universiteit Leuven. Contact him at kurt.schelfthout@cs.kuleuven.be.

Tom Holvoet is a professor in the DistriNet research group in the Department of Computer Science at Katholieke Universiteit Leuven. Contact him at tom.holvoet@cs.kuleuven.be.

Related Links

- DS Online's Middleware Community, http://dsonline.computer.org/ portal/site/dsonline/index.jsp?pageID=dso_level1_ home& path=dsonline/topics/middleware& file=index.xml& xsl=generic.xsl

- The 2nd International Middleware Doctoral Symposium, cms:/dsonline/2006/02/o2003.xml

Cite this article: Edward Curry and Jacques Mossière, "The 2nd International Middleware Doctoral Symposium," *IEEE Distributed Systems Online*, vol. 7, no. 3, 2006, art. no. 0603-o3004.

Rüdiger Kapitza and Franz J. Hauck, "The 2nd International Middleware Doctoral Symposium: EDAS: An Environment for Decentralized Adaptive Services," *IEEE Distributed Systems Online*, vol. 7, no. 3, 2006, art. no. 0603-o3004.

Sharath Babu Musunoori and Frank Eliassen, "The 2nd International Middleware Doctoral Symposium: Quality-Driven Application Service Planning," *IEEE Distributed Systems Online*, vol. 7, no. 3, 2006, art. no. 0603-o3004.

Trevor Parsons and John Murphy, "The 2nd International Middleware Doctoral Symposium: Detecting Performance Antipatterns in Component-Based Enterprise Systems," *IEEE Distributed Systems Online*, vol. 7, no. 3, 2006, art. no. 0603-o3004.

Kurt Schelfthout and Tom Holvoet, "The 2nd International Middleware Doctoral Symposium: Middleware That Enables Protocol-Based Coordination in Mobile Networks," *IEEE Distributed Systems Online*, vol. 7, no. 3, 2006, art. no. 0603-o3004.