Agent-Based Decision Support: A Case-Study on DSL Access Networks

Karsten Bsufka, Rainer Bye, Joël Chinnow, Stephan Schmidt, and Leonid Batyuk

DAI-Labor, Technische Universität Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
{karsten.bsufka, rainer.bye, joel.chinnow, stephan.schmidt, leonid.batyuk}@dai-labor.de

Abstract

Network management is a complex task involving various challenges, such as the heterogeneity of the infrastructure or the information flood caused by billions of log messages from different systems and operated by different organizational units. All of these messages and systems may contain information relevant to other operational units. For example, in order to ensure reliable DSL connections for IPTV customers, optimal customer traffic path assignments for the current network state and traffic demands need to be evaluated. Currently reassignments are only manually performed during routine maintenance or as a response to reported problems. In this paper we present a decision support system for this task. In addition, the system predicts future possible demands and allows reconfigurations of a DSL access network before congestions may occur.

Introduction

The dynamic nature of computer networks and their complex structure imply huge challenges for capturing a holistic view of the overall system state in a real-time fashion. In the context of network operation and provision, network operators do not only face technical problems but also have to consider the business impact of their decisions. Hence, processes for decision-making are very difficult to automate, but *decision support systems* (DSS) can aid in taking appropriate measures.

A telecommunication operator network is comprised of multiple heterogeneous data sources:

- Standard network appliances (routers, switches …)
- Security suites incorporating Intrusion Detection and Prevention capabilities (IDS, IPS and firewalls)
- General-purpose machines hosting either internal applications or external services for customers (web servers …)

In this context, the processes of data access, processing and exploitation are distributed, and the amount of log data accumulated may be in the range of terabytes. Multiagent systems (MAS) are distributed in nature and suited to these problems via their inherent characteristics, i.e., autonomy, social ability or re-/pro-activeness (Jennings and Wooldridge 1998).

In a long-term research activity supported by a telecommunication provider, we investigate over the course of multiple subprojects the practical applicability of agent technology as an integration environment for decision support in the domain of network operation. In this regard, integration not only considers data sources and their processing, but also exchangeable components for recommendation and testing. In the first subproject, presented in this paper, we targeted a DSS for traffic optimization and early congestion warning in DSL access networks.

In this context, inherent characteristics create a multitude of challenges. For example, traffic path assignments in DSL access networks are static, and the network infrastructure is frequently upgraded to adapt to the increasing traffic demands of TriplePlay (Telephony, Internet, TV via a single line) customers. This requires continuous recomputation of optimal traffic path assignments. Furthermore, congestions caused by an increasing number of customers or larger traffic volumes per subscription are hard to predict. Based on this information, planned reconfigurations must be weighed against the cost of changes to the network and an expected frequency or duration of a corresponding service limitation.

Contribution

The contributions of this paper are twofold: First, we develop an agent-based design of a Decision Support Framework (DSF). This framework is distributed, adapts concepts of the autonomic control loop and provides interfaces for extensions such as recommendation or verification components. Based on this design, we then realize an agent-based DSS for performance management in DSL access networks. In particular, we show how simulation and optimization components can be integrated in the overall solution, detail the optimization algorithms and outline practical results.

Related Work

Shim, et al. have already mentioned the distributed nature of MAS and their relevance for DSS. They also discussed the evolution and future of DSS. They predict that software agents will play a vital role for "Active decision support for the next millennium" (Shim, et al. 2002). Sycara, et al. propose multiagent systems for managing distributed information sources and providing decision support. In this context, the authors introduce different agent types for user interaction, task solving and information access (Sycara and Zeng 1996). In more recent works, DSS agents have been used in the domain of electricity markets (Sueyoshi and Tadiparthi 2008) and for breast cancer research (Siddiqa, et al. 2009).

Traffic management, the focus of the presented case study, has been identified as a high priority problem in previous work, e.g., (Goth 2008; Feldmann, et al. 2001), and was studied with a strong focus on the IP layer. However, this cannot be applied to prevent traffic congestion in DSL access networks since DSL access networks only employ protocols such as ATM or Ethernet from the link layer, which is located below the IP layer in the TCP/IP protocol stack. Past research, e.g., (Nicklisch, et al. 1998; Marzo, Vilá, and Fabregat 2000) considered the application of agent technology for the management of ATM networks, but current and future DSL access networks will be mostly Ethernet-based.

The cited approaches share a common idea with the concept of a "Knowledge Plane for the Internet", introduced by Clark, et al. This "Knowledge Plane" enables distributed information sharing as well as a certain level of autonomy and intelligence to counteract identified problems in the network (Clark, et al. 2003). In this regard, Bullot, et al. present a detailed description for the realization of such knowledge planes (Bullot, et al. 2008). However, this approach requires the integration of this functionality into all individual network elements. Unfortunately, this is not yet feasible in a contemporary operational infrastructure. Thus, agents cannot be deployed directly on the network elements, but they can interface with monitoring systems.

Case Study: Performance Management in DSL Access Networks

Before we present our DSS for the management of DSL access networks, we first introduce the domain, terminology and the problem addressed by the DSS.

DSL access networks

Figure 1 presents a schematic view of a DSL access network. In DSL access networks, customers are connected via their modem (1) to a DSLAM (Digital Subscriber Line Access Multiplexer, 2). They terminate the local loop and represent the border to a telecom provider network. The DSLAMs denote the entrance to the concentrator network, a layered hierarchy of switches (3a, 3b), where a BRAS (Broadband Remote Access Server, 4) is at the top.

BRAS are the entry points to the Internet, or more precisely to the core network of the telecommunication pro-

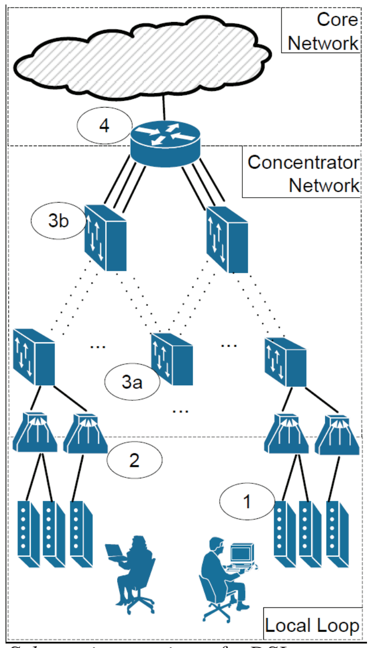


Figure 1: Schematic overview of a DSL access network

vider. Additionally, the BRAS are the endpoints for PPPoE (Point-to-Point-over-Ethernet) connections that are established by the DSL modem of customers. In this regard, they realize the first hop for a customer-initiated IP connection. The hierarchy layers in the concentrator network are connected via parallel links.

As mentioned before the traffic protocols work on the link layer and hence, there are no IP routing or load balancing protocols available. The DSL network administrators need to manually define how traffic is switched on every level in the aggregation hierarchy. Customer traffic is associated to a specific *Virtual LAN* (VLAN), i.e., Ethernet frames labeled with a unique identifier. The atomic unit to be switched is the VLAN group, the aggregation of all customer VLANs connected to a specific DSLAM.

The telecommunication provider uses a performance management (PM) system, which processes inventory data, i.e., the network topology and static VLAN group configuration as well as the consumed bandwidths on the links.

In the scope of this study we realized a DSS to complement the PM system, capable of computing the traffic path assignments, i. e. optimal assignments of VLAN groups to links. Additionally, the DSS incorporates a real-time component for continuous traffic forecast to identify potential congestion situations.

We had access to inventory data of DSL access networks as well as performance data of the monitored links, i.e., upstream and downstream bandwidth consumption as well as the number of customers connected to a DSLAM. To ensure privacy and confidentiality, user-specific data such as access identification and infrastructure-relevant data such as device and link names were encrypted. The performance data was measured in regular 15-minute intervals and used as basis for the optimization and for the simulation model representing DSL customers.

Agent-based Design of a Decision Support Framework

Since the previously described use case of a DSL access network was a trial for other decision support systems in the context of telecommunication operator networks, we start by describing the general Decision Support Framework (DSF), before we go into the details about the case study. Figure 2 presents a high-level view of the DSF. For realizing the DSF we needed an agent framework offering support for component-based building of agents and embedding them in real-world environments. These requirements are fulfilled by *JIAC V* (Hirsch, Konnerth, and

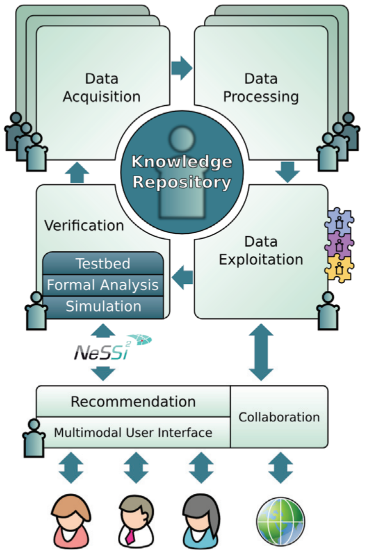


Figure 2: High-level overview of an agent-based decision support framework.

Heßler 2009). An additional advantage of *JIAC V* is that it is used by the network security simulator *NeSSi*^{2} (Schmidt, et al. 2009). Simulation capabilities were a key factor, since one main aspect of our DSS is to use simulation to evaluate the effects of decisions and to use it to forecast future system states, which in turn affect decision proposals.

NeSSi^2 is an open source network simulator originally focused on evaluating security solutions for defending infrastructures against network-related attacks. It provides simulation capabilities on graph-based representations of real-life networks. The simulation models are defined by using the Eclipse Modeling Framework. *NeSSi*^{2} uses *JIAC V* agents for the management and execution of simulations, which offer the ability to annotate tasks an agent performs as public services that can be invoked by other agents. This means a DSS computes the parameters for a simulation scenario at runtime, requests to launch one or more simulations, and evaluates the results upon termination.

The depicted DSF essentially implements a stereotypical autonomic control loop; see for example (Dobson, et al. 2006). We facilitated the public services of *JIAC V* agents and their ability to either automatically publish or manually wrap these services using other service-oriented technologies. For example, services can be automatically published as web services or manually exposed as Java Management Extensions (JMX) services. This is especially important for agents that need to interact with the outside world, be it for collecting data, cooperating with other systems or recommending actions to the operator. We will highlight this in the following, when we present the building blocks of the DSF and describe the agents for the DSS for DSL access network optimization for each building block.

Figure 3 gives an architectural overview of the realized multiagent system. It consists of two agent execution environments (nodes). One node hosts the DSF agents and the second node the *NeSSi*^{2} agents.

The following describes the agent groups of the DSF and implementation of our specific DSS in more detail.

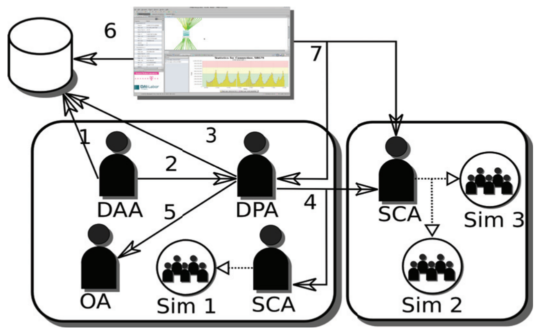


Figure 3: Multiagent system for observing DSL access network performance measurements, create network optimization proposals and forecasting of future network behavior.

Data Acquisition. This group is comprised of agents that are solely responsible for accessing data from existing applications or from sensors on devices. Their single task is to collect data from various sources and store it in a format that can be processed by other agents.

The *Data Acquistion Agent (DAA)* of the DSS accesses performance measurement data exported from monitors in the DSL access network. During the research project, this data was not forwarded directly to the DAA; as mentioned before this data was first pseudonymized.

After receiving this pseudonymized data, the task of the DAA is to transfer the performance measurements to a database (step 1 in Figure 3). Finally, the DAA notifies the DPA (see below) about the new raw data (2).

Data Processing. Agents in this group convert raw data provided by data acquisition agents into more suitable knowledge representation formats. They also perform integrity checks and condense data by filtering and simple aggregation. At this stage of the decision it is possible to automatically trigger reaction mechanisms based on predefined rules on matched incoming data.

In the case study, a *Data Processing Agent (DPA)* is responsible for translating the raw data into information, which can be used for visualization, optimization and forecasting, and storing it in the dedicated database (3). Based on the available performance data, different calculations will be performed, e.g., average link utilization or busy hour identification. This metadata is added to the raw topology data. The available performance measurements are then translated to events that can be interpreted and visualized by *NeSSi*^{2} as graphical reports for inspecting network behavior and state (3). During the calculation of statistics and the translation to simulation events, the performance measurements are also checked for congestion indicators. When deemed necessary, forecast simulations and optimizations can afterwards be requested by the DPA (4, 5). If the previous tasks detected problems, they are written to a log file and optionally reported by e-mail. For this we use an e-mail agent, which is part of a collection of available general-purpose *JIAC V* agents. This agent exposes *remote agent actions* for sending e-mails to other agents, so a DPA simply needs to look up the “Send E-Mail” functionality in the agent node directory service and then invoke it.

In *JIAC V* services are externally offered as web service or Java Management Extension (JMX) services. Internally these services are also published as remote agent actions to other agents. These actions are registered with a directory service and can be invoked by other *JIAC V* agents. The communication between *JIAC V* agents uses the Java Message Service (JMS).

Data Exploitation. Agents in this group use the previously generated knowledge, apply further analysis algorithms (indicated by the pieces of a puzzle in Figure 2) and in so doing, generate new knowledge. This new knowledge is either based on insights generated during data analyses or during comparison of the observed system state with a desired system state. Based on this, they may schedule additional actions that alter a system or generate feedback to operators. The main design aspect for this agent group is that the actual implementation of approaches is not realized within the DSF but delegated to a DSS.

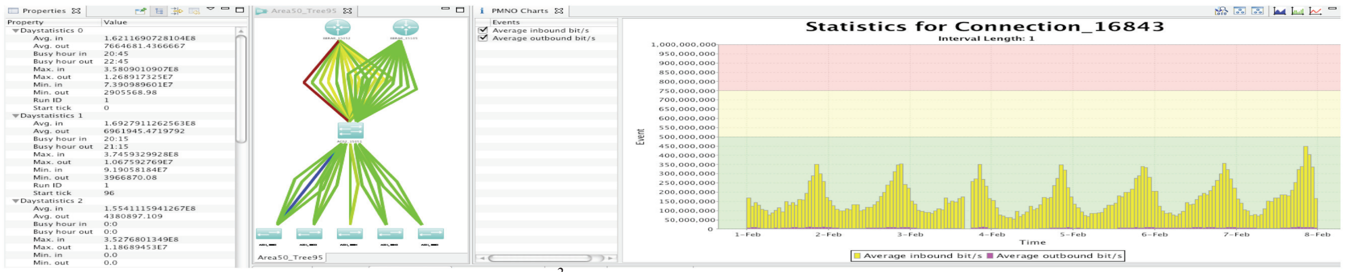
For the case study only an *Optimization Agent (OA)* was realized. Optimization refers to the redistribution of customer traffic on parallel links. An important constraint is that all customer traffic originating from one DSLAM (VLAN group) cannot be split onto different paths. The optimization problem is not a fractional but an integer optimization problem, since DSLAM customer traffic is composed of discrete units of flow, which cannot be divided due to modeling requirements. The optimization problem can be modeled as a bin-packing problem, where the links represent the bins, the capacity of the bin is the capacity of the physical link, and the items are the customer traffic originating from one DSLAM. The size of an item naturally corresponds to the aggregated amount of traffic in the respective VLAN group. Since the bin-packing problem is NP-hard and we are dealing with a potentially large number of algorithm invocations depending on network size, we employ a heuristic to redistribute DSLAM traffic. To this end, several heuristics are available (Vazirani 2004) and were implemented; in test runs the worst-fit decreasing heuristic was selected, which was deemed most suited to achieve a balanced distribution over the links. The DSS is designed to automatically and manually invoke the OA, due to requested operational constraints it is only manually invoked (see section *Interface)*.

Verification. This group of agents is responsible for the verification of results, which can be realized with different strategies. An option is to prove the assumptions with formal analysis. Our DSS also supports simulations to verify assumptions too complex for formal analysis. Finally, it is also possible to integrate an agent platform into a testbed and verify estimations under realistic conditions.

While it is nice to have the ability to compute routing reconfigurations for resolving congestions on links, it is preferable to prevent them in the first place. To this end, we conduct forecast simulations. A forecast simulation is automatically triggered by the system. For this purpose the aforementioned DPA checks the current link utilization each time it receives new data. In case the utilization surpasses predefined thresholds, it initiates a forecast simulation, which includes the link in question.

This is done by contacting the *Simulation Control Agent (SCA)***.** The SCA is a default *NeSSi*^{2} agent and offers services for starting arbitrary simulations. To be able to handle non-IP access networks, we used a new simulation model. Details about this can be found in a paper to be presented at SimuTools 2010 (Bye, et al. 2010). A SCA can start multiple simulations. Furthermore, it is possible to use several nodes with a SCA on a node to increase the simulation performance. The system notifies previously assigned persons from the network operation center via e-mail when congestion is deemed very likely to occur.

Knowledge Repository. Each agent owns a local knowledge repository, called memory. An agent uses its memory to store facts. These facts are either obtained by observing an agent’s environment or by communication with other



Property	Value
Daystatistics 0	
Avg. in	1.6211690728104E8
Avg. out	7664681.4366667
Busy hour in	20:45
Busy hour out	22:45
Max. in	3.5809010907E8
Max. out	1.268917325E7
Min. in	7.390989601E7
Min. out	2905568.98
Run ID	1
Start tick	0
Daystatistics 1	
Avg. in	1.6927911262563E8
Avg. out	6961945.4719792
Busy hour in	20:15
Busy hour out	21:15
Max. in	3.7459329928E8
Max. out	1.067592769E7
Min. in	9.19058184E7
Min. out	3966870.08
Run ID	1
Start tick	96
Daystatistics 2	
Avg. in	1.5541115941267E8
Avg. out	4380897.109
Busy hour in	0:0
Busy hour out	0:0
Max. in	3.5276801349E8
Max. out	1.18689453E7
Min. in	0.0
Min. out	0.0

Figure 4: Replay of performance measurements in NeSSi^2 for one week in February 2009.

agents. Changes to an agent's memory can be observed and if they match a defined pattern can trigger actions, which will be performed by the agent.

In case an agent needs persistent knowledge or to share this persistent knowledge with other agents, the facts are stored in a database. Facts are either stored directly in the database or pass through a persistency layer to facilitate better database storage and access.

For the implemented DSS a database is used, which stores network topologies and current performance measurements.

Interfaces. The group depicted in the bottom of Figure 2 describes two types of interfaces, between several DSS (collaboration; right) as well as between DSS and user (recommendation, multimodal UI; left). In the first case, data is exchanged in a multi-DSS architecture, for example collaborative network intrusion detection systems. In the second case, the actual support part of our DSS, data leaves the DSS domain and is presented to the end user. The responsible DSS agents either actively inform users of the system, e.g., by sending e-mail, or passively by publishing web services other applications can use to query an agent for decision proposals.

To issue DSS-specific commands and display case study specific information, a *NeSSi*^{2} UI extension was developed. This extension allows reviewing the networks and listing all DSL access network topologies. After selecting a topology, the graph representation with additional selection-based node and link information is loaded and displayed. A replay with the measured data can now be started, viewed and individual points in the last measurement interval can be inspected (6). Figure 4 shows this for a selected tree and a measurement interval of one week in February 2009. The user can now supply different traffic generation parameters and start a simulation or optimize the active tree (7). It is also possible to combine both, e.g., simulate optimized trees as well as answer *what-if* questions in regard to customer increase. By switching to another database, users can examine the conducted simulation from continuous decision support, which is described in the next section.

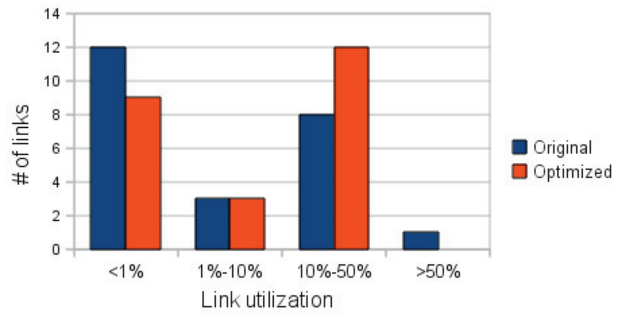


Figure 5: Utilization of a network between 20:15 and 20:30

Evaluation and Discussion

The evaluation of our system addresses two different areas. First, we look at performance issues of the system in relationship to continuous operation. Second, we address the question whether optimizations have the anticipated effect on link congestion.

We tested our continuous DSS with a set of performance-metric data collected in German DSL access networks operated by the data provider. We used a MacBook Pro (Dual Core 2.4 GHz, 4GB RAM) for this purpose. The import of a 15-minutes-interval dataset and 117 forecast simulations for the different access networks took 40 minutes. This means it would not be possible to operate the system in intervals of 15 minutes. After closer examination, we found out that write operations to the database were a limiting factor in our system; in particular this is a result of storing captured performance data persistently. Here, we see optimization opportunities: on the one hand, it is possible to distribute the load on different machines (e.g., three, each handling 39 access trees). On the other hand, a more powerful computer with enough memory to store the database or an SSD RAID may be an option, too. Besides those improvements, it would also make sense to operate the system individually for each DSL access network, and not with one instance for all DSL access networks.

Link utilizations of around 80% will cause packet drops on the affected link and thus decrease for example the picture quality in IPTV services. To account for traffic peaks in the 15-minute intervals, we selected 50% link utilization as an upper bound. Figure 5 shows the link utilization between 20:15 and 20:30, a time where congestion is likely to occur. With the original traffic path assignment, half of the links are unused. However, one link was utilized more than 50% percent. After optimization, several unused links where utilized and the congested one relieved. There are still unused links left, as only parallel links are optimized while unused BRAS are ignored (see Figure 4). Because it isn't possible to split VLAN groups, parallel links on top of aggregation switches with only one DSLAM can't be optimized either.

We achieved the goals defined for our project: we realized a prototype MAS that fulfills the necessary requirements for giving network operators decision support. We were able to cope with the information flood and our framework enabled us to run in a distributed fashion via the *JIAC V* nodes. Our DSS can detect possible network bottlenecks and notifies the user with a more useful VLAN configuration proposed by the optimization agent.

Conclusions and Outlook

We presented an agent-based Decision Support Framework (DSF) targeted at the needs of decision support activities in the domain of network operation. In the context of a case study, we realized an application for performance management in DSL access networks. The system is able to find VLAN group assignments for the current network state and traffic demands, which will reduce the likelihood of traffic congestion and is also able to provide recommendations for future assignments based on forecast simulations. Additionally, the system has been tested for real-time continuous decision support.

In our future research activities we follow two directions. First, we will optimize the implemented system for the real-time case to perform faster. To this end, a live interface to the telecom operator has been realized, providing updated performance data in intervals of 15 minutes. Second, we target misuse detection in an upcoming case study, in particular anomalous behavior of compromised customer machines. Here, we plan to apply the general DSF with a focus on the collaboration of different instances of the system and on anomaly detection algorithms incorporated in the exploitation agent. In particular, distributing not only the individual components of a decision support system, but even instances of it will be a focal point of this research activity.

Acknowledgements

This research was supported and funded by Deutsche Telekom. We thank the reviewers of this paper for their helpful comments.

References

Bullot, T.; Khatoun, R.; Hugues, L.; Gaïti, D.; and Merghem-Boulahia, L. 2008. A situatedness-based knowledge plane for autonomic networking. *Int. J. Network Mgmt*. 18(2):171–193.

Bye, R.; Chinnow, J.; Clausen, J.; Bsufka, K.; Albayrak, S. 2010. Optimization and Early-Warning in DSL Access Networks based on Simulation. *Proceedings SimuTools 2010*.

Clark, D. D.; Partridge, C.; Ramming, J. C.; and Wroclawski, J. T. 2003. A Knowledge Plane for the Internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 3–10.

Dobson, S.; Denazis, S.; Fernández, A.; Gaïti, D.; Gelenbe, E.; Massacci, F.; Nixon, P.; Saffre, F.; Schmidt, N.; and Zambonelli, F. 2006. A survey of autonomic communications. *ACM Trans. Auto. Adap. Syst*. 1(2):223–259.

Feldmann, A.; Greenberg, A.; Lund, C.; Reingold, N.; Rexford, J.; and True, F. 2001. Deriving traffic demands for operational IP networks: methodology and experience. *IEEE-ACM Trans. Netw*. 9(3):265 – 279.

Goth, G. 2008. Traffic management becoming high-priority problem. *IEEE Internet Comput*. 12(6):6 – 8.

Hirsch, B.; Konnerth, T.; and Heßler, A. 2009. Merging agents and services — the JIAC agent platform. In *Multi-Agent Programming: Languages, Tools and Applications*. Springer. 159–185.

Jennings, N. R., and Wooldridge, M. J., eds. 1998. *Agent Technology: Foundations, Applications, and Markets*. Springer.

Marzo, J.-L.; Vilà, P.; and Fabregat, R. 2000. ATM network management based on distributed artificial intelligence architecture. In *Proceedings of the 4th International Conference on Autonomous Agents*, 171–172.

Nicklisch, J.; Quittek, J.; Kind, A.; and Arao, S. 1998. INCA: An agent-based network control architecture. In *Intelligent Agents for Telecommunication Applications*, volume 1437 of Lecture Notes in Computer Science (LNCS), 142–155.

Schmidt, S.; Bye, R.; Chinnow, J.; Bsufka, K.; Camtepe, A.; and Albayrak, S. 2009. Application-level simulation for network security. SIMULATION first published on August 4, 2009 as doi:10.1177/0037549709340730.

Shim, J. P.; Courtney, M. W. J. F.; Power, D. J.; Sharda, R.; and Carlsson, C. 2002. Past, present, and future of decision support technology. *Decis. Support Syst*. 3(2):111–126.

Siddiqa, A.; Niazi, M.; Mustafa, F.; Bokhari, H.; Hussain, A.; Akram, N.; Shaheen, S.; Ahmed, F.; and Iqbal, S. 2009. A new hybrid agent-based modeling & simulation decision support system for breast cancer data analysis. In *International Conference on Information and Communication Technologies (ICICT), 2009*, 134–139.

Sueyoshi, T., and Tadiparthi, G. R. 2008. An agent-based decision support system for wholesale electricity market. *Decis. Support Syst*. 44(2):425–446.

Sycara, K., and Zeng, D. 1996. Multi-agent integration of information gathering and decision support. In *Proceedings of the 12th European Conference on Artificial Intelligence*. John Wiley & Sons, Ltd.

Vazirani, V. 2004. *Approximation algorithms*. Springer. 74–83.