

Flexible decision support in dynamic interorganisational networks

John Collins, Wolfgang Ketter[†], Maria Gini

Department of Computer Science and Engineering, University of Minnesota

[†]Department of Decision and Information Sciences, RSM Erasmus University

wketter@rsm.nl, [jcollins, gini]@cs.umn.edu

Abstract

An effective Decision Support System (DSS) should help its users improve decision-making in complex, information-rich, environments. We present a feature gap analysis that shows that current decision support technologies lack important qualities for a new generation of agile business models that require easy, temporary integration across organisational boundaries. We enumerate these qualities as DSS Desiderata, properties that can contribute both effectiveness and flexibility to users in such environments. To address this gap, we describe a new design approach that enables users to compose decision behaviours from separate, configurable components, and allows dynamic construction of analysis and modelling tools from small, single-purpose evaluator services. The result is what we call an “evaluator service network” that can easily be configured to test hypotheses and analyse the impact of various choices for elements of decision processes. We have implemented and tested this design in an interactive version of the MinneTAC trading agent, an agent designed for the Trading Agent Competition for Supply Chain Management.

Introduction

Organisations in business networks have a growing need for intelligent systems that can assist managers by gathering and analysing information, making recommendations, supporting business decisions, and implementing business workflows. Advanced decision support systems and autonomous software agents promise to address this need by helping to compensate for human cognitive limitations and biases, resulting in better decisions. The recent advent of *Smart Business Networks* (SBN) (van Heck and Vervest, 2007; Vervest et al., 2008) extends traditional business processes and gives rise to new challenges, especially for dynamic and modular business process management, envisioning flexible integration of systems and processes across organisations, and by providing advanced tools to facilitate human managerial decision making and inter-organisational integration.

An example of a Smart Business Network is a Hong Kong trading company that serves retailers around the world with clothing and other products manufactured in Asia, Africa, and the Americas (McFarlan and Young, 2002). Each order requires orchestration of a variety of services, from design and sourcing through production, logistics, quality management, finance, and billing, all within a customised worldwide workflow that may exist only for the duration of that specific order. Their ability to serve customers depends heavily on their ability to assemble information, make decisions, and implement these workflows using a wide range of resources that cross organisational boundaries.

As we see from this example, an important element of the SBN vision requires flexible integration of technical infrastructure on an as-needed basis to support both business processes and managerial decisions. However, it appears to us that the current generation of Decision Support Systems (DSS) is not up to the task in such a dynamic environment, due to complex, non-standard interfaces, lack of clear semantics, and a variety of other reasons. We are interested in finding an approach that provides the flexibility and robustness necessary to construct rich, effective decision support systems in the SBN world.

We motivate the need for our work with a gap analysis between the capabilities of currently available DSS technologies and the capabilities needed to effectively support decision support in smart business networks.

This analysis is based on literature review and discussions with experts in the field. A major element of the SBN vision is the ability of actors, by which we mean people and organisations, to quickly connect to other actors to achieve specific business objectives and then disconnect when a task is finished. To address this need, we describe the design and practical implementation of an “evaluator service network,” a highly configurable and flexible decision support system that dynamically connects nodes of a business network and disconnects them when no longer needed. This design supports goal directed service composition that takes advantage of a constrained service architecture to greatly simplify the semantics of matching, composing, and validating business services. We describe a prototype tool that enables managers to visualise, understand, and validate service network compositions with a graphical representation of the actual network configuration. We describe how an evaluator service network can serve as the foundation of a flexible economic dashboard architecture that can dynamically connect selected nodes in a service network to visualise their real-time status, such as current parts and finished goods inventory positions, risk and reward management. This architecture can greatly empower business managers in their understanding of the overall business network structure and facilitate real-time managerial decision making.

We have developed and demonstrated the effectiveness of evaluator service networks in a supply chain simulation scenario (Collins et al., 2009). Traditionally, supply networks have been created and maintained through the interactions of human representatives of the various enterprises (component suppliers, manufactures, wholesalers/distributors, retailer and customers) involved. However, the recent advent of trading agents opens new possibilities for automating and coordinating the decision making processes within and across organisations. Since experimenting on real world business networks is risky, and the real world is often a poor environment for controlled experiments, we have implemented and tested our evaluator service network on a supply-chain testbed, the Trading Agent Competition for Supply Chain Management (TAC SCM) (Collins et al., 2005). We describe an example implementation of our flexible decision support system and demonstrate its value in an interactive version of the MinneTAC trading agent (Collins et al., 2009), an agent that performs coordinated buying, selling, production scheduling, and inventory management in the context of TAC SCM. We also present results of our network visualiser toolbox, where a manager is able to see and manipulate the current configuration of the network as well as the state of the nodes.

In the following section we review the relevant related literature, focusing on the architectural challenges presented by interorganisational networks. We then present and discuss a feature gap analysis of currently available features in DSS, and our DSS desiderata. Next we describe the evaluator service network – our solution to this gap, evaluate the technology, and list multiple possible applications of it. We then highlight our contributions in different scenarios on an interactive version of the MinneTAC agent. Finally, we summarise our findings in the conclusions and present our research agenda for future work.

Foundations: decision support in smart business networks

This work builds on work in several fields. In computer science, it is related to software engineering, artificial intelligence, autonomous agents, and multi-agent systems, especially architecture, and the semantic web. In economics and information decision sciences, it draws from the framework of design science, smart business networks, and decision theory.

Design science

We use a design-science approach (Hevner et al., 2004) to advance the scope and flexibility of decision support systems. Our ontology-driven decision support system is relevant to existing problems in information systems, because it has the potential to address a variety of important problems that are not well-served by existing approaches. Our presentation of the evaluator service network approach applies guidelines from (Hevner et al., 2004), including design as an artifact, problem relevance, and research contributions.

Decision support systems

Much progress in technology and in business processes comes from automating the elements that do not require human judgement. According to Arnott and Pervan (2007), Decision Support Systems (DSS) are IT-based systems focused on supporting and improving managerial decision-making. They identify seven categories of DSS and, through extensive literature review, eight key issues in the DSS discipline. Clark et al. (2007) discuss the importance of technology base and the gap between available and adopted technology for effectiveness of Management Support Systems, a framework that combines Decision Support, Business Intelligence, Executive Information Systems, and Knowledge Management Systems. We focus on the gap between currently available technology and the needs of advanced DSS.

Smart business networks

During the mid-nineties Goldman et al. (1995) and Sanchez (1995) stressed that in highly dynamic business networks the capability to quickly connect network actors (businesses) is essential to enable fast response times and greater variety when new opportunities arise. The concept of “quick connect” includes a search and select behaviour by the different businesses. Goldman et al. (1995) further argue the need for a “quick disconnect” when the business transaction is over, otherwise open network connections can create undesirable information flows. Our architecture offers a unique way of automatically connecting, disconnecting and communicating with the appropriate actors in the network. One must pay particular attention to the interfaces of the different network actors. Establishing temporary connections between actors is infeasible unless the parties agree on the semantics of the data they share.

Kambil and Short (1994) argue that there is a strong need to construct software tools for business network representation, visualisation, and analysis. These tools can help managers to visualise the different network actors, or roles, and linkage-based strategies of different organisations, enabling analysis of changes in emerging organisational forms. Hoogeweegen et al. (2006) and van Liere et al. (2008) argue that knowledge of the network structure empowers the decision maker, and leads to better business decisions. Our architecture offers unique capabilities for network visualisation, role and linkage analysis. Users can visualise the network structure, and drill down on particular actors to get a detailed picture of specific decision chains.

Creating performance and information dashboards (Eckerson, 2005) is part of the new emerging field of Business intelligence (BI) (Shmueli et al., 2006). BI provides functionalities such as real-time monitoring, performance reporting, and support for exploring solution space with normative models, statistical techniques and visualisation. BI software can crawl the web, mine data, and generate reports customised to user preferences. According to Adam and Pomerol (2002) the layout of an economic dashboard has a direct impact on the understanding derived by managers. They argue that a graphical user interface (GUI) should be leveraged to maximise its visual impact. According to the extensive report on the visual design of dashboards by Few (2006), many software companies, including Microsoft and Oracle, have sold dashboard applications since 2001 (see <http://www.enterprise-dashboard.com>). Our architecture fully supports BI and our dashboards are customisable for individual managers to facilitate managerial decision making.

Semantic web

According to Berners-Lee et al. (2001), the current hypertext document-centric web will evolve to include an infrastructure of machine-readable semantic descriptions that can be understood and acted upon by intelligent agents. In the last seven years we have seen significant progress toward the realization of this vision in the Semantic Web.

Service-oriented architecture has become an important technology for DSS, but there is debate among Information Systems professionals about its influence on managerial decision-making (Demirkan et al., 2008). Our work adds formal semantics to a service-oriented approach, opening up the way for extending its capabilities for flexible decision support.

Our work is complementary to that of Blau et al. (2009). Their approach addresses location, pricing, and provisioning of services in a public network. Our approach adds detailed, domain-oriented semantic

descriptions of composable services and their dependency relationships. Huhns and Singh (2005) describe a process for combining heterogeneous services into solutions, and identify a number of problems with existing technology. Our approach addresses two of the problems they identified, by restricting the form of services and by incorporating formal semantic descriptions.

Universal Description, Discovery and Integration (UDDI) is a platform-independent, XML-based registry for businesses worldwide to list their Web-accessible services on the Internet. UDDI is an open industry initiative, enabling businesses to publish service listings and discover each other and define how the services or software applications interact over the Internet. Major drawbacks include the fact that existing deployments are not for public consumption and the lack of rigorous semantic descriptions of services.

To facilitate the discovery and composition of services we need rigorous semantics based on a formal ontology. We think that usable general-purpose ontologies may be unrealistic, but that organisation-specific and industry-specific ontologies are quite feasible and realistic. The Web Service Modelling Ontology (WSMO) (Roman et al., 2005; Vitvar et al., 2007) is a conceptual model for structuring semantic annotation of services and a step in realizing real world ontologies. The description of services at a semantic level using ontologies allows the use of machine reasoning and the use of implicit information in the process of partner selection.

Scientific workflow systems

Scientific workflow systems (Gil et al., 2010; Ludascher et al., 2006) allow complex data analysis workflows to be composed from collections of data sources and software components, typically using Semantic Web technologies (Korhonen et al., 2003). The goals for this work are not the same as the needs of business users for decision support and process integration. Much of the work in scientific workflows aims to automate large-scale processing of huge, distributed datasets (e.g. genome data, climate models) across large collections of grid-style computing resources. Automation is critical both due of the extreme complexity of the tasks involved, and to the need for repeatable results. In contrast to the approach we present here, the work in scientific workflows does not try to simplify the form of the services, but rather to describe arbitrarily complex services with a degree of semantic precision. Users of these tools need considerable sophistication in mathematics and logic.

Decision Support Systems: A feature gap analysis

Humans have limited cognitive capacity and suffer from cognitive biases, resulting in limited ability to make rational or optimal choices (Simon, 1979; Ariely, 2008). Humans tend to use simple rules called heuristics to guide their decision making (Todd and Gigerenzer, 2001). As a consequence, humans are “satisficers” who typically are willing to accept a non-optimal solution that fits their needs, instead of “maximisers” who scrutinise and evaluate all the options available. Additionally, when humans are faced with large numbers of options, they commonly suffer from the “tyranny of choice,” which leads people to experience negative emotions (Schwartz, 2004) and have difficulty in making choices.

Modern Decision Support Systems (DSS) are designed to compensate for these human limitations, and to reduce the margin of human error. They can be very effective in relatively stable business environments. However, the SBN vision requires a new generation of DSS that can address human decision-support needs in highly agile business situations. This adds new layers of fundamental requirements for effective decision support. These requirements can be categorised into three basic areas: software engineering, SBN support, and decision science.

In the software engineering area, decision support in agile business environments requires a high degree of user-directed adaptability that does not require deep software development skills. To support dynamic business networks, DSS systems must support easy intra- and inter-organisational composition of services, with ready visibility of the structure of the network and of relevant data flowing within the network. Finally, users must be able to play out hypothetical scenarios, and must be able to understand the foundations of aggregated data and decision recommendations. We have summarised these new requirements in the form

Area	Desired Property	Explanation
Software Eng.	1. Appropriate separation of concerns	Elements and tools are a good match to the skills and needs of both technical developers and non-technical business users.
	2. User configurability	System can be easily tailored to specific user preferences.
	3. User-defined abstractions	Users can factor out details so that one can focus on a few concepts at a time.
	4. Easy to experiment and test	Users can experiment with configurations and test the operation of the system at any time. Also called “exploratory development.”
SBN	5. Network structure visualisation	Users can visualise, understand, and validate the designed decision chain with a graphical representation of the actual network.
	6. Quick connect and disconnect	Users can dynamically connect and disconnect nodes of a business network according to current business needs.
	7. Goal directed service composition	Complex business applications can be semi-automatically composed and validated from individual services with formal semantic descriptions.
	8. Flexible dashboard architecture	User interaction tools can be dynamically connected to selected nodes to visualise and manipulate their real-time status.
Decision Sci.	9. Transparency	Users can see foundations and reasoning behind recommendations and decisions. The choice and presentation of information affects trust and confidence of human decision makers.
	10. Hypothetical scenarios	Users can substitute hypothetical for real-world data in order to predict impact of possible future events or courses of action.

Table 1: DSS areas: Software Engineering, Smart Business Networks, and Decision Sciences.

of 10 specific “DSS Desiderata,” shown in Table 1. We have also numbered them, so that we can refer to them by number in the subsequent discussion.

In the following we highlight the most common technologies that are used for constructing DSS. We are especially interested in DSS that can help their users to make rational choices in complex, information-rich, dynamic environments. In this sense we are primarily interested in Personal DSS, Negotiation Support Systems, and Intelligent DSS as defined by Arnott and Pervan (2007).

<i>Desired property</i>	<i>DSS Technology</i>				
	ERP System	Data Warehouse	Spreadsheet	Expert System	Mash-up
1. Appropriate separation of concerns	+	+	+	-	-
2. User configurability	0	0	++	+	++
3. User-defined abstractions	--	--	-	-	+
4. Easy to experiment and test	-	-	++	0	+
5. Structure visualisation	-	0	-	+	-
6. Quick connect, disconnect	--	--	--	-	+
7. Service composition	--	--	--	0	+
8. Flexible dashboard	0	0	0	0	+
9. Transparency	-	-	-	+	-
10. Hypothetical scenarios	+	--	++	+	--

Table 2: Ratings of popular DSS technologies. Scores range from extremely difficult to implement (--) through neutral (0) to easy (++).

Five technologies are commonly used for building DSS capabilities:

1. *Enterprise resource planning (ERP) Systems* are enterprise-wide computer software system used to manage and coordinate the resources, information, and functions of a business from shared data stores (Esteves and Pastor, 2001).
2. *Data Warehouse Systems* are subject-oriented, integrated, time-varying, non-volatile collections of data and query-based analysis tools, useful to support organisational decision making (Inmon, 2005).
3. *Spreadsheets* are desktop or web-based computer applications that model a rectangular grid of cells. Each cell contains text, a numeric value, or a formula that defines how the contents of that cell is to be calculated from the contents of a combination of other cells whenever one of those cells is updated (Ragsdale, 2004).
4. *Expert Systems* attempt to reproduce the reasoning performance of human experts, most commonly in a specific problem domain by capturing expert knowledge in the form of rules and other formal structures, and providing an inference mechanism to generate consequences (Jackson, 1998).
5. *Mash-ups* are web applications that combine data or functionality from two or more sources into a single integrated application (Ketter et al., 2009; Mulholland et al., 2006).

All these approaches have strengths and weaknesses. In Table 2 we rate these existing DSS technologies according to the desiderata we have identified, based on literature review, discussions with experts in the IS and SBN communities, and our own experience. Clearly, all of these technological approaches lack important features, limiting their usefulness in SBN environments. We have taken the view of a non-technical mainstream business user. Extremely technical users could disagree with some of our ratings, such as the perceived lack of abstractions and transparency in spreadsheet applications. Next, we describe a technological approach that can fill a substantial portion of these gaps.

Filling the gap: Evaluator service networks

We have argued that existing DSS technologies are not sufficiently flexible, transparent, easy to work with, etc. to produce the kinds of ad hoc information and analysis that would truly leverage the skills and experience of their users, compensate for their cognitive limitations, and maximise their effectiveness. Our approach to closing the feature gap provides highly configurable, transparent decision processes that are fully described in terms that both end users and automated systems can understand. In our approach, information, analyses, and decision recommendations are composed from a variety of data views and small, single-purpose analysis modules that can be composed into dataflow networks to produce results with well-defined business meaning.

We call these modules “evaluator services.” A schematic visualisation of such a service is shown in Figure 1. Each evaluator service takes some input data from a variety of sources, performs some transformation on that data, and produces an output.



Figure 1: An evaluator service with its inputs, transform, and output.

Evaluator services (or just “evaluators”) can be composed into arbitrarily complex structures by connecting inputs to compatible outputs. Evaluators refer to each other by name rather than direct reference, and these names are configurable, either through XML configuration files, or through a user interface. This approach preserves independence among evaluator services, and it elevates and makes visible the high-level structure of the decision support processes. The result is that complex networks and feedback loops can be constructed from relatively simple services using metadata.

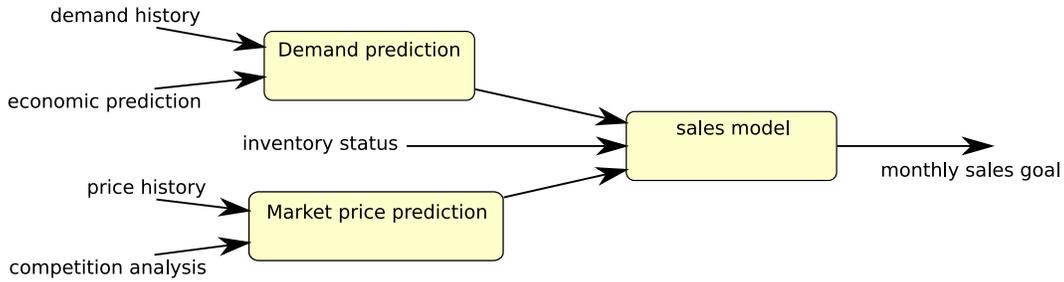


Figure 2: Simple example of an evaluator service network.

Figure 2 shows a simple example of such a network. The goal is to decide how much of a product to attempt to sell in the current market. The Sales model service recommends monthly sales goals, based on inputs including a prediction of future demand, a prediction of future prices, and the current and expected inventory of raw materials and finished goods. The Demand prediction service combines historical demand data with current economic projections to produce predictions of customer demand. The Market price prediction service similarly combines input from multiple sources to predict market prices for the product.

Service composition supported by semantic descriptions

The evaluator service network design addresses the feature gaps we identified in the previous section. It can be implemented as a type of web-service mash-up, and shares the advantages of configurability, scalability, flexibility, and the ability to quickly connect and disconnect service nodes (properties 2 and 6). It is better at separation of concerns, because the simple, restricted form of evaluator services is much easier for non-technical users to understand and compose (property 1) than the more general form of web services commonly used in service-oriented architectures (Vitvar et al., 2007) and scientific workflow systems (Gil et al., 2010). Because of their simple network structure, evaluator service networks are easier to visualise and manipulate than a service mash-up (properties 2 and 5), and the dataflow structure simplifies construction of hypothetical scenarios (property 10). The evaluator service network model allows rich, complex decision-support systems to be built from simple, single-purpose analysis and modelling modules, strung together in ad hoc dataflow networks (properties 6 and 7). In an inter-organisational environment, a security and location infrastructure will be required; we do not address the security implications in this work, and we refer the reader to Blau et al. (2008) for an approach to the location problem.

The next step is to understand how users can use this approach to construct a real-world decision-support system that can be useful in strategic, tactical, and operational business contexts.

Strategic and tactical users can add and remove services and their interconnections (properties 2 and 6), either individually or by using matching and inference tools to compose sets of services to produce desired results (property 7). Users can also define new abstract services by composing available services into sub-graphs (property 3), to support their own needs or those of subordinates. Network structure is visible and manipulable (properties 5 and 9), and the simple structure of evaluators makes it easy to experiment, and to incrementally build and test a complete system for a specific business purpose (property 4). Users can view and adjust parameters on individual services, such as interest rates or risk tolerance (property 2).

All types of users can dynamically compose dashboard displays by adding data viewers to individual dataflows. Drill-down capability would commonly amount to moving a viewer upstream in the network (property 8). Users can explore hypothetical scenarios by substituting simulated or historical data in place of real-world data on network inputs (property 10). This enables users to evaluate the likely outcomes of alternative decisions or anticipated market decisions, or to replay past events with alternate process models.

We expect non-specialist users to be able to successfully build working evaluator service networks and understand the data that is produced. We also expect that software developers will implement individual services. If the resulting services are easy for business users to understand and compose into effective decision

and process support tools, this will provide effective separation of concerns between software professionals and business users (property 1).

Business users who wish to compose solutions need to clearly understand the format, content, and business meaning of the data and of the transformations that are performed by the services. Much of the capability needed to express this information is provided by elements of the Semantic Web (Berners-Lee et al., 2001), specifically the Resource Description Framework (RDF), the Web Ontology Language (OWL), and associated inference tools (Yu, 2007). RDF represents simple facts (such as descriptions of services and data) as triples of the form [Subject Predicate Object] or alternatively [Resource Property-name Property-value]. For example, we could assert that a service `http://cs.umn.edu/svc/effdemand` provides a daily data element called `effective-demand` in the form of a vector of integers, indexed by product ID, that represents the current demand that could potentially be converted to sales at current prices:

Subject	Predicate	Object
<code>http://cs.umn.edu/svc/effdemand</code>	provides	<code>effective-demand</code>
<code>effective-demand</code>	update-frequency	daily
	datatype	vector1
	instance-of	vector
<code>vector1</code>	element-type	integer
	indexed-by	product-id
	represents	current available demand

and so on. The remainder of this description would detail the inputs and the transformation applied by the `effective-demand` service. A more detailed example is given in the next section.

Storing these facts in a uniform structure allows queries and logical inference to be performed on this dataset. As with web crawlers, an initial query can fan out and extract additional facts, but there is one important difference. Unlike linked web pages, RDF statements require that each fact and link be a URI (e.g., the [provides] relationship is an abbreviation of the URI `http://cs.umn.edu/svc#provides` in the above example). By forcing precise and unique relationships to also be unique URIs, RDF queries can look for relationships between triples that have not been joined together. This simple and repeatable rule of triples can generate a complex, arbitrarily large “system of knowledge”. This semantic “web join” capability allows us to join otherwise unrelated data stored in general web page content.

RDF triples may be stored in structures called “triple stores” analogous to a single relational database table with three columns. Triple stores have their own query language called SPARQL (<http://www.w3.org/TR/rdf-sparql-query>), which facilitates the creation of “mashups” of data. It does this by allowing discovery of nodes in separate semantic graphs that represent the same object. This in turn allows new facts to be generated from the RDF triples.

Semantic annotation of composable services supports key elements of the Smart Business Network vision. Emerging business needs can be addressed by semi-automatic composition of services available from partners, vendors, and internal capabilities. Business users will be able to integrate these purpose-built networks into internal processes as needed, and will be able to control the type and degree of autonomy. The automatic discovery, composition, and invocation of independent services will lead to an increase of the business agility of firms. We will expand on this approach using an example of service annotation and composition in the MinneTAC trading agent.

Building a trading agent with an evaluator service network

Since the inception of TAC SCM in 2002, more than 50 teams have built agents for the competition. These agents represent a variety of approaches to solving the various modelling and decision problems presented by the simulation scenario. Our MinneTAC agent models a flexible organisation using the evaluator service network approach. There are a few top-level decision elements (Procurement, Production Scheduling, Sales) and a large number of evaluator services that act as modelling and analysis modules, supported by a common

data store. A high-level schematic representation of this design is shown in Figure 3. For details and additional examples we refer readers to (Collins et al., 2009).

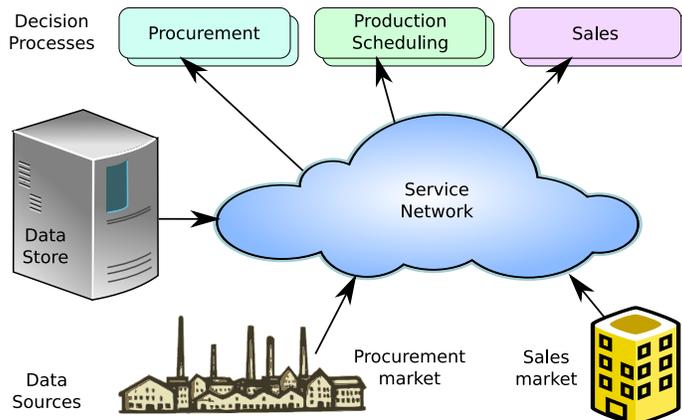


Figure 3: Abstract view of the MinneTAC trading agent architecture.

In Figure 3, the primary decision components are shown across the top. The Procurement service deals with suppliers, attempting to find the parts needed by Production Scheduling at the lowest possible cost. Manufacturing schedules the production facility with assembly tasks that maximise the expected value of its available inventory and production capacity. Sales sets prices and makes customer offers that are expected to maximise profit, given its available resources. These three decision services are supported by a common data store and by a large set of interconnected evaluator services, represented schematically as the cloud in the centre of the diagram. Specific examples of MinneTAC evaluator service networks are shown later in this paper. The evaluators have access to each other and to various internal and external data sources, primarily in the form of periodic market reports that are issued by the simulation, and a data store containing a large body of historical data that has been “digested” by machine learning models, such as the “economic regime” model described by Ketter et al. (2007, 2009).

The decision components are implemented as evaluator services, and different implementations of them can be freely substituted into the network. Evaluators may request inputs from other evaluators, from the data store, and from external sources. They then transform that data in various ways, for example by updating price models, estimating demand trends, or composing and running optimisations to produce sales quotas or procurement recommendations. Results are provided in a common format so they can be used by other evaluators or decision components. Connections among decision components and evaluators are entirely configurable and modifiable at runtime; the only real dependency in this design is on the data store, and on external data sources such as market data and user inputs. This allows individual researchers to encapsulate modelling and decision problems within the bounds of individual evaluator services that have minimal, well-defined interactions among themselves.

Composing an evaluator service network

In this section we describe in some detail the annotation and composition of a small portion of the MinneTAC evaluator network, specifically the portion that determines sales offer prices. In a competitive auction market, an important decision variable is the probability that a customer will buy at a given price. Figure 4 shows schematically an instance of an evaluator that generates transfer functions called “pricers” that calculate a price for a given customer order probability. In other words, given a probability p , we can compute a price as $\text{pricer}(p)$. This is useful if one wishes to maximise profit by offering the highest price that can be expected to sell a target sales volume for a given level of demand.

We can use such a transfer function to set an offer price for a product X (see Figure 5) as follows: given the current demand D_X (current-demand) for product X , determine quantity Q_X (quotas) of product X to

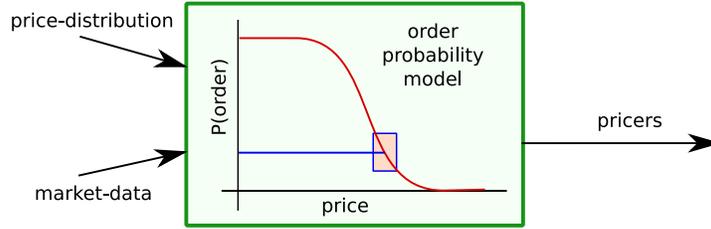


Figure 4: An evaluator service called order-probability that computes prices from an order probability model.

sell and set price (product-prices) $\text{pricer}_X(Q_X/D_X)$ such that when offered on all demand D_X , moves Q_X units in expectation.

The evaluator that performs this function is called simple-price in Figures 7 and 8. Figure 5 is a snapshot of the portion of an evaluator network that computes prices in this way.

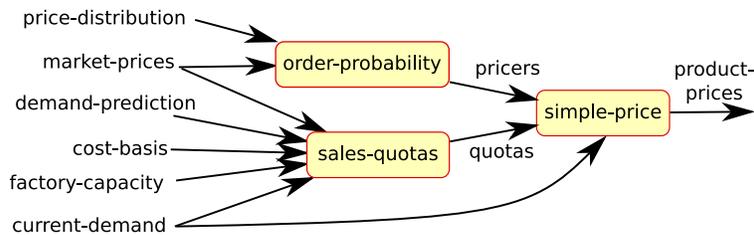


Figure 5: Evaluator example: Dynamic pricing chain.

In order to compose an evaluator network that includes the simple-price evaluator, we need to know what it expects for inputs, what it produces as output, and what function it performs. Knowing what it expects for input data, we can then locate other evaluators that produce that data and make the necessary connections. Figure 6 is a graphical depiction of the semantic description of the simple-price evaluator shown in Figure 5.

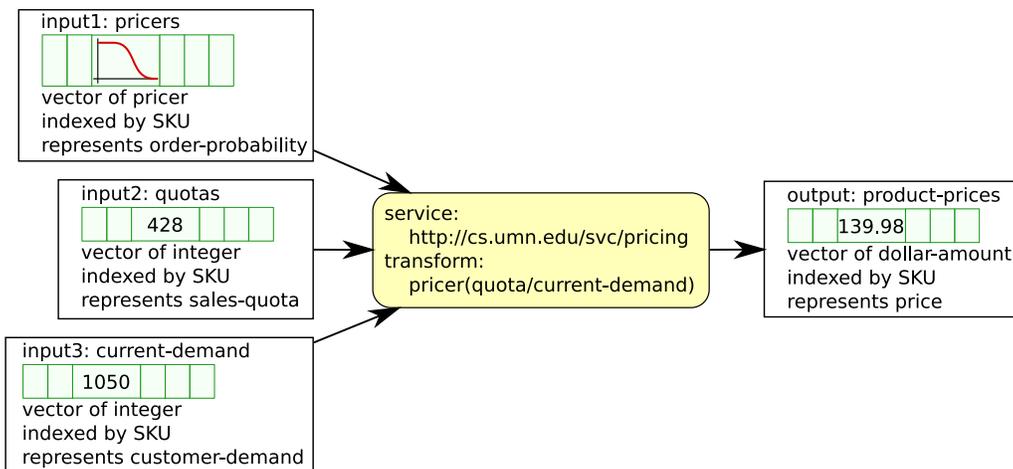


Figure 6: Graphical depiction of simple-price evaluator semantics.

In order to compose this evaluator into a network, it is necessary to find a match between each of the input specifications (pricers, quotas, and current-demand) and the output specifications of candidate upstream

evaluators. Of course, it will frequently be the case that exact matches cannot be found, especially in an open network. This can be addressed by services that adapt formats or by ontology extensions. Note that this relatively simple approach to service composition differs markedly from the more general methods described in the literature, such as (Traverso and Pistore, 2004; Rao and Su, 2004; Gil et al., 2010), because of the constrained dataflow structure of the services we propose. To further illustrate the power of evaluator services, in Figure 7 we show a more detailed example of the evaluation network that is used to set prices in one of the MinneTAC configurations. Each of the cells in this diagram is an evaluator, based on the method of “economic regimes” developed by Ketter (2007).

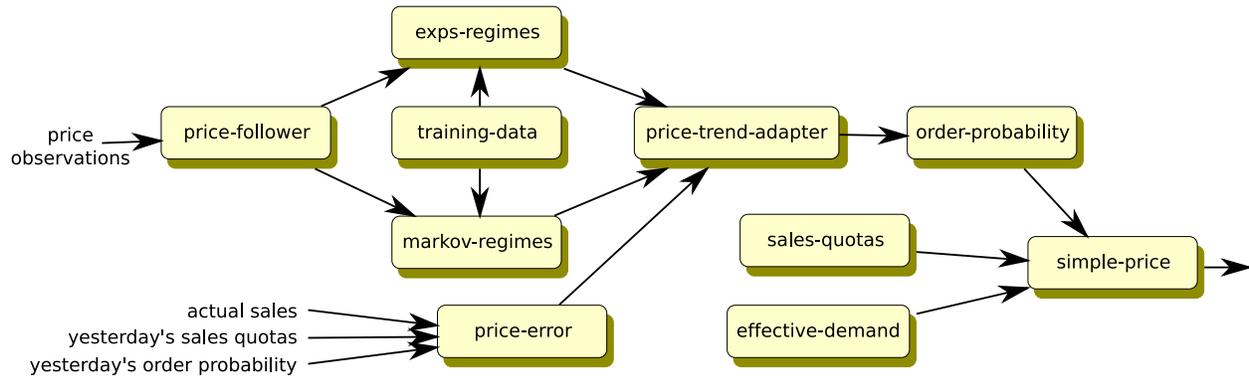


Figure 7: Evaluator chain for a sales manager that uses sales quota and information provided by regimes to determine prices, price trends and order probability.

This configuration, based on the method of “economic regimes” developed by Ketter (2007), includes different economic regime identification and prediction methods which are encapsulated in the `markov-regimes` evaluator and in the `exps-regimes` evaluator. Both of these evaluators depend on training data, distilled from a large number of past simulations. The `training-data` evaluator supplies this data from an external data source. The analysis was developed using machine learning methods, as described by Ketter et al. (2007, 2009). These evaluators can dynamically select the most appropriate portions of the training data for a given market situation. In a real business network setting we would train the system on historical transaction data, and update it at regular intervals.

Visualising evaluator service networks

As we can see from Figure 7, evaluator chains can become very complex. One result is that a design that was intended to make a complex agent design easy to understand has its own complexities. The original implementation requires users to configure an agent through a pair of XML files. One maps implementations of the principal decision processes to their roles in the agent, and the other specifies the evaluators and their interconnections, along with parameters that control aspects of their operation. This is a major problem for two reasons. First, XML is not a particularly readable language for most people. Second, the network structure is not immediately evident from reading a configuration file. This leads to serious usability problems.

To ease the burden of creating and understanding these networks, we have built a visualiser and graphical editor for agent configuration files. Figure 8 is a screen shot of its user interface (the configuration of a complete MinneTAC agent is much larger than this, typically including nearly 100 interconnected evaluator services). Users can visualise the entire network or portions of it, add or remove evaluators, and set parameters on the evaluators. The selected “price-error” evaluator uses the difference between sales quotas and actual sales to fine-tune sales prices.

The ability to view and edit the evaluator network configuration is very helpful, but agent designers, like business managers, need more than that in order to gain a clear understanding of the status of the business.

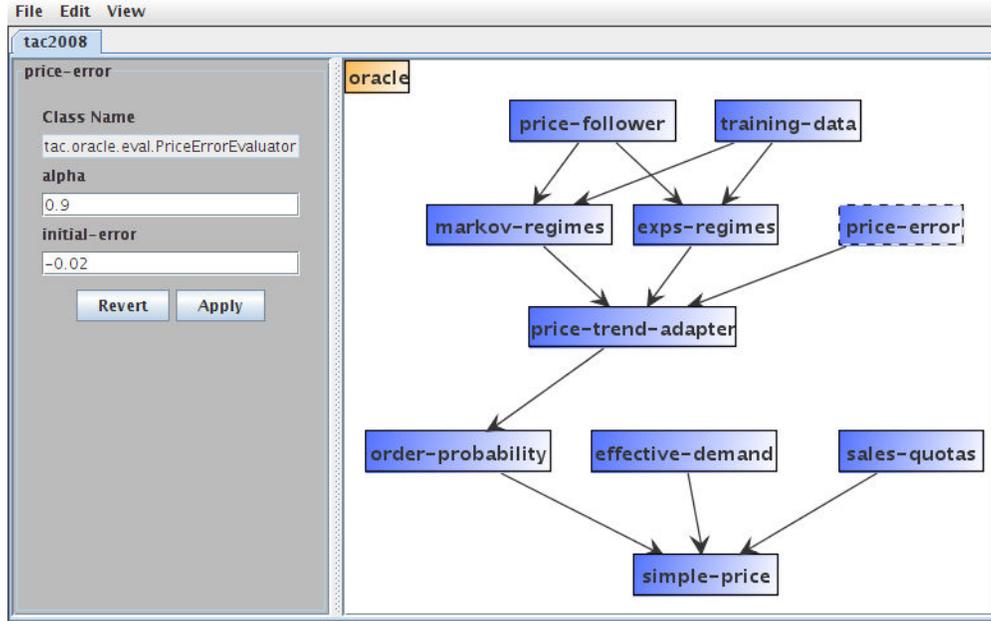


Figure 8: Network detail view of the agent configuration editor and visualisation tool.

Advanced business systems address this need by providing the ability to configure and display “dashboards” that can show summary and detail views of various quantifiable aspects of a business operation. The MinneTAC design allows a user to dynamically compose such “dashboard” displays by connecting a variety of graphing and plotting widgets to the outputs of evaluators. This can be done “on the fly”, while the system is running, because the composition of services (Sirin et al., 2003; Wu et al., 2003) and visualisations is entirely dynamic.

In an evaluator service network, virtually all of the quantitative information that drives decisions flows through connections between evaluators. This means that business-intelligence dashboards can be largely composed of relatively simple graphical viewers attached to these connections. For example, Figure 9 show a strategic dashboard with a sample price distribution prediction and historically observed prices that are used by the *order-probability* evaluator. In the left panel, the dashed curve is the predicted price distribution for the current day, the solid thick line for 20 days in the future, and the thin lines for intermediate days. The shift of the price distribution towards the left over its 20-day horizon indicates decreasing future prices, and the increasing spread of the distribution reflects increasing uncertainty over time. The right panel shows a history of minimum and maximum prices for a specific product, along with three different statistical time series.

In addition to composing and viewing the decision network and its data, a business user will need the ability to view and possibly override the decisions it recommends. In Figure 10 we see a prototype interface that presents a user with recommended product prices for the current period, and gives the user the option of changing them.

Conclusions and future work

We believe that current approaches to building Decision Support Systems are not well-matched to the needs of a future environment where business agility will depend on the ability to build and manage business processes quickly, across organisations. We have identified a set of ten “DSS Desiderata” that are important for building systems in organisations that wish to implement the Smart Business Network vision. We have

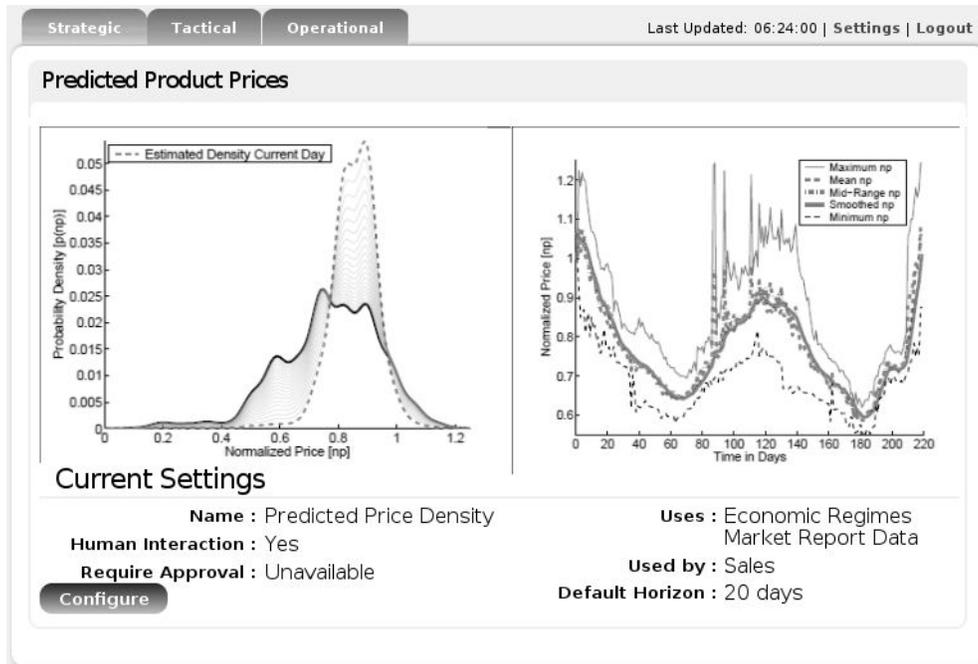


Figure 9: Dashboard showing current and predicted price distribution, and price history.

Please approve the recommended prices below.

SKU	Description	Price (\$)
1	Pintel 2GHz, 1Gb Memory, 300 Gb Hard disk	1650
2	Pintel 2GHz, 1Gb Memory, 500 Gb Hard disk	1750
3	Pintel 2GHz, 2Gb Memory, 300 Gb Hard disk	1750
4	Pintel 2GHz, 2Gb Memory, 500 Gb Hard disk	1850
5	Pintel 5GHz, 1Gb Memory, 300 Gb Hard disk	2150
6	Pintel 5GHz, 1Gb Memory, 500 Gb Hard disk	2250
7	Pintel 5GHz, 2Gb Memory, 300 Gb Hard disk	2250
8	Pintel 5GHz, 2Gb Memory, 500 Gb Hard disk	2350

Approve

Figure 10: Interface for user approval of recommended prices.

shown that none of the current technologies for building DSS have all of these desirable properties.

We introduced an approach to closing this feature gap, in the form of evaluator service networks. This approach combines a restricted form of web services with a combination of technical and domain-oriented semantic description and inference. These evaluator services can be composed into dataflow networks to accomplish arbitrary monitoring, analysis, and decision support tasks ranging from simple data monitoring to fully-autonomous intelligent agents. Because an evaluator service network is composed of small, easy-

to-understand components, and because the network itself is visible and manipulable, it is reasonable to expect that a non-technical business user will be able to understand, modify, and even create them to satisfy immediate business needs. Networks can be composed quickly from services drawn from the local technical environment, within an organisation, or across organisations, and they can be easily disconnected when no longer needed. The combination of domain-oriented and technical semantic descriptions and inference tools will allow many kinds of problems to be solved by automatic or semi-automatic composition of evaluator networks. Because all the important data flows through identifiable network connections, it is easy to compose dashboards by connecting viewers to the outputs of individual evaluators. By routing a dataflow through a user interface element, recommendations can be presented to the user for approval with possible modification.

To demonstrate the effectiveness of the evaluator service network design approach, we have used it to implement a trading agent called MinneTAC that competes in the Trading Agent Competition for Supply Chain Management (TAC SCM). The MinneTAC agent consists of an adapter that connects the agent to the simulation environment, a data store, and a large evaluator network of 60 to 100 evaluators. The TAC SCM environment is a severe test of software design and implementation, requiring hundreds of sales, procurement, and inventory-management decisions every 15 seconds. MinneTAC, which obtained the third place in the 2009 competition, is competitive with the best agents that have been implemented for this scenario.

In complex economic scenarios such as TAC SCM, the desired design qualities include clean separation of infrastructure from decision processes, ease of implementation of multiple decision processes, clean separation of different decision processes from each other, and controllable generation of experimental data. In a competition environment, the ability to compose multiple agents with different combinations of decision process implementations makes it possible to quickly test hypotheses about the effectiveness of competing decision models.

The basic ideas behind the evaluator service network approach can be directly applied in a web service environment. The services could be implemented as RESTful services (Richardson and Ruby, 2007), supported by semantic descriptions expressed with OWL-DL (the description-logic version of the Web Ontology Language). Location and security services are required to complete the environment; their form and requirements will depend on the details of the application.

In the future, we plan to extend these ideas into an agent-assisted collaborative work environment in which a service network that supports business processes can be “tapped into” by multiple users to support their own roles and preferences. We also plan to support a rich capability for setting up and analysing hypothetical scenarios with an array of simulation and statistical tools, implemented as evaluator services.

Acknowledgements

A preliminary version of this paper (Collins et al., 2008) was presented at the Smart Business Network conference in Beijing, May 2008, and won the best paper award.

Partial support for Maria Gini is gratefully acknowledged from the National Science Foundation under award NSF/IIS-0414466.

References

- ADAM F and POMEROL JC (2002). Critical factors in the development of executive systems – leveraging the dashboard approach. In *Decision Making Support Systems: Achievements and Challenges for the New Decade*, M. Mora, G. Forgionne, and J. Gupta, Eds. Idea Group Inc, 305–330.
- ARIELY D (2008). *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. HarperCollins.
- ARNOTT D and PERVAN G (2007). Eight key issues for the decision support systems discipline. *Decision Support Systems* 44, 657–672.
- BERNERS-LEE T, HENDLER J, and LASSILA O (2001). Semantic web. *Scientific American Magazine* 284(5), 34–43.

- BLAU B, NEUMANN D, WEINHARDT C, and MICHALK W (2008). Provisioning of service mashup topologies. In *Proceedings of the 16th European Conference on Information Systems, ECIS 2008*.
- BLAU B, VAN DINTHER C, CONTE T, XU Y, and WEINHARDT C (2009). How to coordinate value generation in service networks: A mechanism design approach. *Business and Information Systems Engineering* 1(5), 343–356.
- CLARK TD, JONES MC, and ARMSTRONG CP (2007). The dynamic structure of management support systems: theory development, research focus, and direction. *MIS Quarterly* 31(3), 579–615.
- COLLINS J, ARUNACHALAM R, SADEH N, ERICSSON J, FINNE N, and JANSON S (2005). The supply chain management game for the 2006 trading agent competition. Tech. Rep. CMU-ISRI-05-132, Carnegie Mellon University, Pittsburgh, PA. November.
- COLLINS J, KETTER W, and GINI M (2008). Flexible decision support in a dynamic business network. In *The Network Experience – New Value from Smart Business Networks*, P. Vervest, D. van Liere, and L. Zheng, Eds. Springer Verlag, 233–246.
- COLLINS J, KETTER W, and GINI M (2009). Flexible decision control in an autonomous trading agent. *Electronic Commerce Research and Applications* 8(2), 91–105.
- DEMIRKAN H, KAUFFMAN RJ, VAYGHAN JA, FILL HG, KARAGIANNIS D, and MAGLIO PP (2008). Service-oriented technology and management: Perspectives on research and practice for the coming decade. *Electronic Commerce Research and Applications* 7(4), 356 – 376.
- ECKERSON WW (2005). *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. Wiley.
- ESTEVEZ J and PASTOR J (2001). Enterprise resource planning systems research: an annotated bibliography. *Communications of the Association for Information Systems* 7(8), 1–52.
- FEW S (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O’Reilly Media, Inc.
- GIL Y, RATNAKAR V, KIM J, GONZÁLEZ PA, GROTH P, MOODY J, and DEELMAN E (2010). WINGS: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems* forthcoming.
- GOLDMAN S, NAGEL R, and PREISS K (1995). *Agile competitors and virtual organizations*. Van Nostrand Reinhold New York.
- HEVNER A, MARCH S, PARK J, and RAM S (2004). Design science in information systems research. *Management Information Systems Quarterly* 28(1), 75–106.
- HOOGEWEEGEN M, VAN LIERE D, VERVEST P, VAN DER MELJDEN L, and DE LEPPER I (2006). Strategizing for mass customization by playing the business networking game. *Decision Support Systems* 42(3), 1402–1412.
- HUHNS MN and SINGH MP (2005). Service-oriented computing: Key concepts and principles. *IEEE Internet Computing* 9(1), 75–81.
- INMON W (2005). *Building the data warehouse*. Wiley.
- JACKSON P (1998). *Introduction to Expert Systems (3rd Edition)*. Addison Wesley.
- KAMBIL A and SHORT J (1994). Electronic integration and business network redesign: a roles-linkage perspective. *Journal of Management Information Systems* 10(4), 59–83.

- KETTER W (2007). Identification and prediction of economic regimes to guide decision making in multi-agent marketplaces. Ph.D. thesis, University of Minnesota, Twin-Cities, USA.
- KETTER W, BANJANIN M, GUIKERS R, and KAYSER. A (2009). Introducing an agile method for enterprise mash-up development. In *Proc. of the IEEE Conf. on Commerce and Enterprise Computing*. Vienna, Austria.
- KETTER W, COLLINS J, GINI M, GUPTA A, and SCHRATER P (2007). A predictive empirical model for pricing and resource allocation decisions. In *Proc. of 9th Int'l Conf. on Electronic Commerce*. ACM, Minneapolis, Minnesota, USA, 449–458.
- KETTER W, COLLINS J, GINI M, GUPTA A, and SCHRATER P (2009). Detecting and Forecasting Economic Regimes in Multi-Agent Automated Exchanges. *Decision Support Systems* 47(4), 307–318.
- KORHONEN J, PAJUNEN L, and PUUSTJARVI J (2003). Automatic composition of web service workflows using a semantic agent. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03)*. IEEE Computer Society, 566–569.
- LUDASCHER B, ALTINTAS I, BERKLEY C, HIGGINS D, JAEGER E, JONES M, LEE EA, TAO J, and ZHAO Y (2006). Scientific workflow management and the kepler system. *Concurrency and Computation* 18(10), 1039.
- MCFARLAN FW and YOUNG F (2002). Li & Fung (A): Internet issues. *Harvard Business School case*, 9–301.
- MULHOLLAND A, THOMAS C, KURCHINA P, and WOODS D (2006). *Mashup Corporations: The End of Business as Usual*. Evolved Technologist Press.
- RAGSDALE C (2004). *Spreadsheet modeling and decision analysis*. Thomson/South-Western Mason, Ohio.
- RAO J and SU X (2004). A survey of automated web service composition methods. In *Proc. of the First Int'l Workshop on Semantic Web Services and Web Process Composition*. Springer.
- RICHARDSON L and RUBY S (2007). *Restful web services*. O'Reilly.
- ROMAN D, KELLER U, LAUSEN H, DE BRUIJN J, LARA R, STOLLBERG M, POLLERES A, FEIER C, BUSSLER C, and FENSEL D (2005). Web service modeling ontology. *Applied Ontology* 1(1), 77–106.
- SANCHEZ R (1995). Strategic flexibility in product competition. *Strategic Management Journal* 16, 135–159.
- SCHWARTZ B (2004). The tyranny of choice. *Scientific American* 290(4), 70–75.
- SHMUELI G, PATEL N, and BRUCE P (2006). *Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner*. Wiley-Interscience.
- SIMON HA (1979). Rational decision making in business organizations. *The American Economic Review* 69(4), 493–513.
- SIRIN E, HENDLER J, and PARSIA B (2003). Semi-automatic composition of web services using semantic descriptions. In *Web Services: Modeling, Architecture and Infrastructure at ICEIS*. Angers, France.
- TODD PM and GIGERENZER G (2001). Précis of simple heuristics that make us smart. *Behavioral and Brain Sciences* 23(05), 727–741.
- TRAVERSO P and PISTORE M (2004). Automated composition of semantic web services into executable processes. *Lecture Notes in Computer Science*, 380–394.

- VAN HECK E and VERVEST P (2007). Smart business networks: how the network wins. *Commun. ACM* 50(6), 28–37.
- VAN LIERE DW, KOPPIUS OR, and VERVEST PHM (2008). Network horizon: An information-based view on the dynamics of bridging positions. *Advances in Strategic Management* 25, 595–639.
- VERVEST P, VAN LIERE D, and ZHENG L (2008). *The Network Experience – New Value from Smart Business Networks*. Springer Verlag, Berlin, Germany.
- VITVAR T, MOCAN A, KERRIGAN M, MORAN M, CIMPIAN M, HASELWANTER T, and FENSEL D (2007). Semantically-enabled service oriented architecture: concepts, technology and application. *Service Oriented Computing and Applications* 1(2), 129–154.
- WU D, PARSIA B, SIRIN E, HENDLER J, and NAU D (2003). Automating DAML-S web services composition using SHOP2. *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*.
- YU L (2007). *Semantic Web and Semantic Web Services*. Chapman&Hall/CRC.