Proceedings K-CAP 2001 Workshop WS2

Knowledge in e-Business

Editors:
Jaap Gordijn. Free University Amsterdam
Hans Akkermans. Free University Amsterdam

Contents

eMarketplace: The Next Generation

Hamada H. Ghenniwa
Cooperative Distributed Systems Engineering Group
Electrical & computer Engineering Department
Faculty of Engineering Science
University of Western Ontario
London, Ontario N6A 5B9
Canada

Abstract
This paper proposes an agent-based architectural framework for cooperative distributed systems in *eBusiness* with a special attention to *eMarketplace*. This business-centric knowledge oriented architecture (BCKOA) describes the framework of an ontology driven *eBusiness* environment that deals with several business and design issues with special attention to supporting many-to-many relationship between the marketplace participants as well as the integration of marketplaces in an open environment. In this work we envision the *eMarketplace* as a seething milieu in which economically motivated software agents cooperatively/competitively interact, find and process information and disseminate it to humans and to other agents. This environment, based on coordinated, intelligent rational agent (CIR-Agent) model for market services, business services, and integration services as well as the user interface. To demonstrate and evaluate the feasibility and the effectiveness of the model as a development tool for *eMarketplace*, a prototype is being developed for *eMarketplace* with two Mutual Fund virtual business entities.

Keywords
E-business, Electronic Marketplace, Agent-oriented Design, Software Architecture

INTRODUCTION

The volume of electronic business (*eBusiness*) is growing rapidly as companies strive to stay competitive by quickly responding to changes in the global marketplace. *eBusiness* is the use of the Internet along with other electronic means and technologies to conduct business, including within-business, business-to-consumer, business-to-business, and business-to-government interactions. Electronic marketplaces (*eMarketplaces*) are the latest and most significant weapon to reshape *eBusiness* relationships, and will soon affect all businesses in one way or another.

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

It will enable consumers to shop around the clock from product suppliers and services providers around the world. Likewise, businesses can reach customers worldwide quickly and at low cost. *eMarketplace*s offer companies the chance to develop and enhance their most important relationships – those with customers and suppliers – while allowing market makers to open new revenue opportunities. Companies can use *eMarketplace*s to strengthen their existing trade relationships, discover and develop new ones, and promote faster and more efficient trading.

This paper presents a novel agent-based architecture for *eMarketplace*. The architectural specification of an *eMarketplace* describes the framework of a cooperative distributed system to deal with several business and systems questions, such as (1) What are the design issues for *eMarketplace*? (2) What is the appropriate model of *eMarketplace* environment? (3) What foundation services an *eMarketplace* should provide? (4) How does agent-orientation enable and support *eMarketplace*?

In this paper we propose and develop architectural agent-based software environment for *eMarketplac*e and *eBusiness* that supports the interaction of economically-motivated software agents, and enables and supports common market and business services as well as cross-enterprise integration in open *eMarketplac*es environment. Section 2 reviews and discusses some of the related work in both the academia and the industrial communities. Section 3 provides a brief analysis for the main architectural design issues for *eMarketplace*. Section 4 presents the main components and architecture of the proposed agent-based BCKOA for *eMarketplace*. Section 5 reports a short description of an ongoing implementation of the proposed model for virtual Mutual Fund *eMarketplace*. Finally, Section 6 summarizes the main contributions of this paper.

RELATED WORK

There have been several attempts to develop *eMarketplace* architectures by the academia and the industrial communities. Due to the limited space, the following represent a few of these attempts and the most appropriate for the work proposed here. For example, the work in [4] proposed a DBMS based architecture for electronic market place (EMP). The objective was to develop a business-to-business system architecture that supports many-to-many relationship between customers and suppliers. EMP view the market in terms of customers, suppliers, and administrators; and it supports three types of hierarchal structured services, namely, market, business and basic services. However, most of the business transactions are realized by the market services, such as registration, authentication, offering, buying, and negotiation. Another approach [13] proposed global electronic market (GEM) system based on a logical market framework and infrastructure. A main objective was to decouple systems related and market related design issues. In GEM framework, the market has been viewed in terms of three main elements, market, traders, and brokers. The market provides trading mechanisms including bids and offers. Traders are market participants and represented as a set of interfaces for human suppliers/consumers or for software traders. Whereas brokers are mediators between traders and market. An alternative approach based on agent technology is Kasbah [5]. Kasbah is a primitive marketplace at which "naive" agents buy and sell goods or services on the behalf of their users. The primary objective of the marketplace was to facilitate interaction between the agents. However, Kasbah's architecture is ad hoc, technology driven and provides few fundamental services, such as common seller-buyer matching (based on ``tents of interest" metaphor) and notification. A more complex architecture for *eMarketplace* is MAGMA [16] with a special focus on the infrastructure required for conducting commerce on the Internet. In MAGMA the *eMarketplace* has been viewed in terms of three main functionalities: traders, advertising and banking. Trader agents are responsible for buying and selling goods. The Advertisement server provides a classified advertisement that includes search and retrieval of ads by category. Whereas Bank provides a set of basic banking services such as checking accounts and electronic cash. Alternatively, OFFER [3] proposed a brokering-based architecture marketplace. In OFFER the *eMarketplace* has been viewed as a collection of suppliers, customers and brokers. A customer can search for a service either directly in the e-catalog of the supplier or use the e-broker to search all the e-catalogs of all the suppliers, which are registered with the broker. In a different approach MOPPET [2] proposed an *eMarketplace* system as agent-based workflows. MOPPET viewed the market as a workflow management system carried out by five types of agents, task, scheduling, facilitator, and recovery.

In the industry arena several companies sprung up to meet the challenge. Many of these have been around for as long as the World Wide Web, others are relatively new. For example, Ariba [1] developed a marketplace based on procurement portals and dynamic exchanges for horizontal marketplace. Ariba Dynamic Trade for instance attempts to provide dynamic trade mechanisms, such as auctions, reverse auctions and bid/ask exchanges and negotiations. SAP AG also developed SAP Service Marketplace [14] as an Internet portal to services for the SAP community. It provides basic online services and information offerings, browse catalogs and order services from SAP and its partners, matchmaking. Other approaches were directed to support vertical marketplaces, like PaperExchange [12] that enables consumers and suppliers to negotiate pricing and directly transact with one another. PaperExchange attempts to provide several supporting services such as logistics and clearing services, industry-specific content including job listings, industry events information, news headlines, and a resource directory. VerticalNet also built a marketplace [17] as a set of several web-based marketplaces for specific industrial segments, such as financial services, healthcare, and energy. Each web-site forms a community of vendors and buyers in a specific area. Vertical trade communities are introduced in segments with a substantial number of buyers and suppliers, a high degree of fragmentation on both the supply and demand sides, and significant on-line access.

Another direction adopted by many major software vendors to develop Internet-based commerce platforms. For example, IBM CommercePOINT [7], Microsoft Site Server Commerce Edition [10], Oracle Internet Commerce Server INTERSHOP [8], and Sun JavaSoft JECF (Java Electronic Commerce Framework) [15]. These attempts focused on providing infrastructure services like security payment directories and catalogs to be integrated with existing systems and the Web, but most of them are proprietary.

The approach taken here to develop a novel software architecture for *eMarketplace* as a cooperative distributed system is based on integrating concepts and design principles from several disciplines and technologies including intelligent software agents, distributed systems, meta-language descriptions, organization theory, and decision theory. However, it is important to note that the results of many of the aforementioned attempts and others were very instrumental in the formulation of the ideas and the work proposed here.

eMARKETPLACE MODELS: DESIGN ISSUES

To develop successful engineering foundation of *eMarketplace*, we need a fully realized solution that accommodates the needs of the customers and suppliers and allows them to extend advanced services to the trading community. To understand the step shift forward that *eMarketplace* should represent, it is important to analyze the nature of the existing *eMarketplace* models.

In the sell-side storefront model, a single seller, typically a distributor, constructs a Web storefront to sell to many consumers. A major shortcoming with this model is the assumption that either a single distributor is responsible to aggregate all the suppliers in a given industry, or the customer remains responsible for comparison-shopping between suppliers. Alternatively, in the buy-side *eProcurement* model, corporate procurement aggregates many supplier catalogs into a single "universal" catalog and allows end-user requisitioning from the desktop, by which facilitating standard procurement for the organization and cutting down on "maverick" purchasing. This model, however, doesn't support dynamic trading. In addition, buying organizations must set-up and maintain catalogs for each of their suppliers, and hence it is too costly and technically demanding.

Clearly, an appropriate model of *eMarketplace* should support "many-to-many" relationship between customers and suppliers. Therefore, both customers and suppliers can leverage economies of scale in their trading relationships and access a more "liquid" marketplace. Sellers find customers for their goods, customers find suppliers with goods to sell. In addition, many-to-many liquidity in the *eMarketplace* allows the use of dynamic pricing models such as auctions and exchanges, further improving the economic efficiency of the market.

Another key aspect of *eMarketplace* model is the ability to support integration in open *eMarketplaces* environment. This is driven by the fact that usually customer needs may go beyond the specialist capabilities of any single *eMarketplace*. To cater to broader buying requirements, *eMarketplaces* might need to support interaction with each other in a way it effectively extending the product range without giving up "control" of the customer. It is also important that the model of the *eMarketplace* allows for incorporating and leveraging customers' and suppliers' legacy environments with minimum overhead. This end-to-end support must take place over a "cooperative distributed system" architecture. This architecture should also be scalable in the sense it is capable of supporting a large number of users in a highly open environment. The *eMarketplace* model should also enable complex business rules, workflow, and relationships, and allow for integration with custom and third-party commerce modules.

BUSINESS-CENTRIC KNOWLEDGE-ORIENTED ARCHITECTURE (BCKOA)

To deal with the aforementioned design issues we propose an agent-based business-centric knowledge-oriented architecture (BCKOA) for *eMarketplace* and *eBusiness*. BCKOA is based on an abstraction-layered architecture, shown in Figure 1. The objective is to develop architectural software environment that is semantically rich and describes the organization and the interconnection amongst software components, business services and business ontologies, as shown in Figure 1.

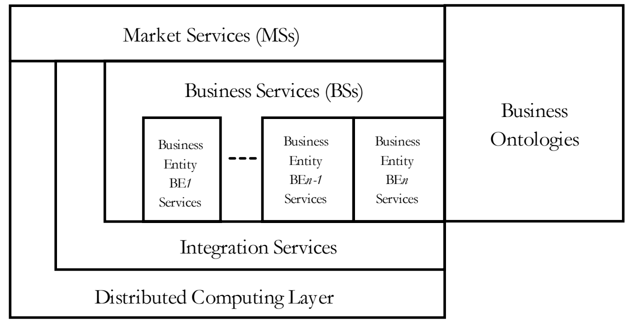


Figure1: BCKOA for *eMarketplace*

The lower layer of BCKOA is the infrastructure that supports both the distributed computing layer and the integration layer. The infrastructure layer might represent one or more physical network-based environment in which *eBusiness* systems can exist. The assumption is that this infrastructure might support various markets, for providing or obtaining specific goods and services. However, each *eMarketplace* may be independent, and may support it's own services, rules, procedures, and protocols as described by the business entities layer and the associated business description in the business ontologies.

An *eMarketplace* may support several business domains as described at the business layer. The business layer provides the integration between the business context of the market and the participant business entities. Note that a business entity may participate in multiple *eMarketplace*s. A Bank, for example, could participate in markets as varied as investment management, mutual fund management, and financial advising service. In this setting, any activities or functions within a specific business entity could participate in an *eMarketplace* dedicated to satisfying its needs as described by the business ontologies of the market. This structure enables a business entity to integrate and describe the types of business services offered within the *eMarketplace*. Examples of business services, such as mutual fund, include catalog browsing, open an account, purchase a fund, and redeem a fund. However, the details of each service type and the required information might vary among business entities. For example, various business entities might provide different service details of purchasing a fund, such as direct purchase, phone purchase and pre-authorized bank withdrawal purchase.

Although the relationships and the interactions among services are captured by the services themselves, the integration services supported by the infrastructure layer, provides the foundations of the interaction within and across *eMarketplaces* in away that is hidden from the service-users and fulfills both the business and the technology requirements. These services include coordination, resource discovery, ontology and semantic integration as well as wrapping services. The interaction mechanisms supported by the *eMarketplace* describe both the pattern and protocol of exchanging massages between the services. Each interaction message may contain one or more types of information needed for or by the services during that interaction.

Agent-Oriented BCKOA

All types of services in *eMarketplaces* environment are usually involving complex and non-deterministic interactions, often producing results that are ambiguous and incomplete. Auctions and negotiations are some examples. In addition, the dynamic nature of the environment requires that the components of the system be able to change configuration to participate in different, often simultaneous roles in *eMarketplaces*. These requirements could not be accomplished using traditional ways of manually configuring software. Add to this the workload scale needed to accommodate profile customer preferences and create communities of like interest, goes beyond humans capabilities to manage the large scale knowledge-based tasks inherent in *eBusiness*.

In this work we strongly believe that agent-orientation is a very promising design paradigm for *eMarketplace*s. In fact, such a paradigm is essential to model open *eMarketplace*s environment, especially considering the multiple dynamic and simultaneous roles a single business-entity may need to participate in given *eMarketplace* sessions (a financial services organization may have representatives acting on its behalf within the context of brokerage, service provision, and marketing simultaneously).

The first principle of agenthood is that an agent should be able to operate as a part of a community of cooperative distributed systems environment, including human users. In our view, an agent is an individual collection of components that provide a focused and related set of capabilities (see Figure 2, depicting Coordinated Rational, Agent (CIR-agent) developed by the author [6]). Each component is associated with a particular functionality supports a specific agent's mental state as related to its goals, [9]. These components include knowledge, problem-solver, pre-interaction, interaction, and communication components, as shown in Figure 2. However, no specific assumptions need to be made on the detailed design of the agent's components. Therefore, the internal structure of the components can be designed and implemented using object oriented or another technology, provided that the developer conceptualizes the specified architecture of the agent as described in Figure 2. In addition, these agents, through the interaction and communication components, are able to coordinate, cooperate and possibly compete with the each other including human beings.

Based on the CIR-Agent model, agent roles within the context of BCKOA-based *eMarketplace* can be categorized as market services, business services, business-entity services and business facilities as well as integration services. In addition this model recognizes that user interface agents as important role to support and collaborate with the market users to achieve the their goals. Figure 3, illustrates the agent-oriented BCKOA for *eMarketplace* based on OMG-CORBA middleware.

At the business layer, business agents are specialists that provide or support a collection of user-services provided by the *eMarketplac*e, such as selling mutual fund products. However, the business-entity agents are the actual service providers. Business-entity agents are representatives of the participant business entity in the *eMarketplac*e and may be based on legacy applications or libraries, such as catalogue web site of mutual fund products. Additionally, business facility agents are specialists that provide a set of generic tasks that can be configured to the requirements of a business domain by which business-entity agents of a specific domain can perform their tasks in the *eMarketplace*. These services belong to the business domain but might not be necessary part of the core business for many business entities in the domain. For example, financial advisory, and legal advisory for mutual fund domain.

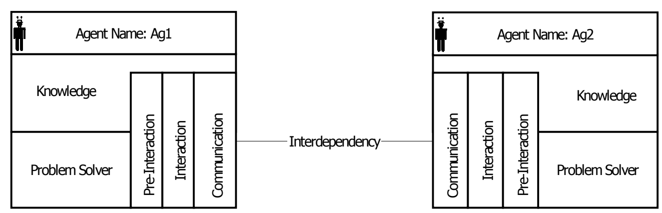


Figure 2: CIR-Agent Architecture

Market agents are specialists that provide collection of functions for the generic eBusiness in an *eMarketplace*, such as market administration, advertisement and registry services by which single entities in the market can perform their tasks regardless of their business domains. Integration agents are also specialists that provide collection of integration functions for the generic cooperative distributed systems environment, such as naming service, resource discovery, and semantic integration by which single entities (agents, components, objects, etc.) can perform their tasks in the environment regardless of their application domain.

As illustrated in Figure 3, an *eMarketplace* customer and a business-entity (BE1) personal are able to participate in the market through their user-interface agent. Similarly, each service offered by the participant business-entities in the *eMarketplace* is represented by an agent. These agents provide thin, intelligent, highly autonomous interfaces for the business-entity services that might represent legacy applications. Each user-interface agent and business-entity agent is registered in the *eMarketplace*. Thus, a user-interface agent is able to benefit from the market and business services by interacting with their representative agents. Each agent representing a business-entity service also register with a registry agent specialized for the corresponding business service, which in turn register with the market registry agent. Each layer, and it's registry services, are intended to provide some aspect of information about the *eBusiness* environment and enable an interested party to obtain information to potentially use offered services, or to join the *eMarketplace* and either provide new services or interact as a trading partner with other business-entities in that *eMarketplace*.

In addition, different agents with different roles might collaborate to achieve a specific business service. These agents might be representatives of the same or different business-entities. For example, a selling agent (for business entity BE*1*, shown in Figure 3) could collaborate with a price forecasting agent, a trading rules agent, and a risk management agent that work for the same BE*1* or out-sourced agents form other business entities, all with the *eMarketplace* data feed and historical task knowledge agents.

Clearly as depicted in Figure 3, agent-oriented approach is appropriate to model and design:

- the foundation components of *eMarketplaces*,
- horizontal services and facilities within *eBusiness* applications,
- vertical services for specific business domain, and
- to optimally lever legacy systems.

In fact, agent-orientation is an adequate paradigm throughout the information architecture of the next generation *eBusiness* systems. In addition, the proposed BCKOA provides an appropriate architecture for the *eMarketplace*. This architecture's form follows *eMarketplace* functions, which are inherent in the real world of market place. They are complex, and non-deterministic. Yet, they are the way of real business environment, and must be the way of *eBusiness*. Luckily that agent technology is rich in the sense of supporting and enabling the automation of complex tasks and developing systems that are reliable and able to take responsibilities of the *eMarketplace* in which they might compete. The components of agent-based BCKOA, namely, *eMarketplace* services, business entities (products, suppliers, customers, etc.) and the integration services that glue them together are essential to building robust many-to-many value chains in the emerging *eBusiness*.

IMPLEMENTATION

A prototype of Agent-based BCKOA for Mutual Fund *eMarketplace* (*MF*) is currently being developed in the Distributed Intelligent Systems Lab at the University of Western Ontario. The current stage of *MF*, as shown in Figure 4, includes two virtual mutual fund companies, *ABC* Bank and *XYZ* Mutual Fund Inc. *MF* supports two basic business services, namely, redemption and purchase of mutual funds. Both *ABC* Bank and *XYZ* Mutual Fund Inc. are supported by BCKOA based on CORBA computation environment and registered with *MF* for both purchase and redemption services. A customer is able to participate in the market through a dedicated user-interface agent assigned by *MF*. Similarly, each participant business entity is assigned to a team of CIR-Agents for the offered services and their representative personal who might need a direct contact with the market as well as with the customers.

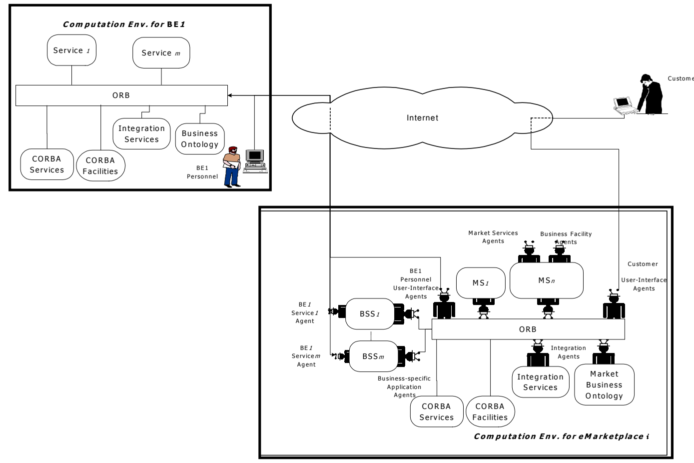


Figure 3: Agent-Oriented BCKOA for *eMarketplace*

An MF-agent's architecture is based on the CIR-Agent model and being implemented using Java. Java features, such as portability, dynamic loading, multi-threading and synchronization support made it appropriate to implement the inherent complexity and concurrency in *MF*. These features were also instrumental for implementing the CIR-Agent components in a parallel fashion. For example, an agent has been designed to be able to negotiate an assignment contact while engaging in resolving a conflict with different agents. The current implementation focuses on the following agent roles user-interface, customer service representatives for purchase and redemption functionalities, market and business registry. Agents communicate with each other using a simple version of KQML with XML message content.

The integration services of *MF* are based on CORBA-middleware commercial product. The current development is focusing on extending CORBA services by two families of services including coordination, ontology and semantic integration. The ontology components are implemented using UML and XML for *MF*, *ABC* Bank and *XYZ* Mutual Fund Inc.

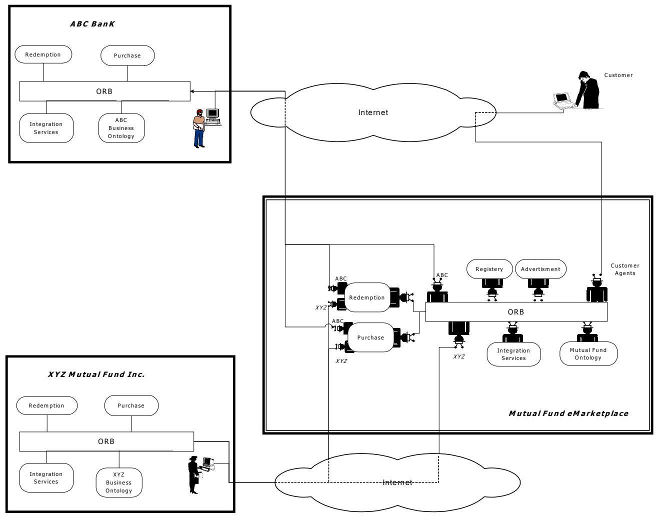


Figure 4: Prototype of MF *eMarketplace*

CONCLUSIONS and FUTURE WORK

This paper reported an ongoing research to developing an agent-based architectural framework for cooperative distributed systems in *eBusiness* with a special attention to *eMarketplace*. Several design issues have been identified for *eMarketplace*, most importantly, to capture and enable "many-to-many" relationship between customers and suppliers, and to support *eMarketplaces* integration in an open market environment. To deal with these issues we proposed agent-based business-centric knowledge-oriented architecture (BCKOA). The CIR-Agent model was proposed as the building blocks for BCKOA, in which six types of agent roles were identified, namely, market services, business- service, business facilities, and business-entity services as well as integration services and user-interfaces. A prototype of BCKOA is currently being developed in the Distributed Intelligent Systems Lab at the University of Western Ontario for mutual fund business with redemption and purchase business functionalities. The current focus is on demonstrating the feasibility and the effectiveness of the model as architectural framework for *eMarketplace*, with a special attention to the foundation components of *eMarketplaces*, services within *eBusiness* applications, fundamental active representatives for specific business services, and levering legacy systems. In continuing the research, the computational elements and effectiveness of the model will be the main concern.

REFERENCES

[1] Ariba Marketplace, Ariba Inc, available at <http://www.ariba.com>

[2] Arpinar, S., Dogac, A., Tatbul, N., "An Open Electronic Marketplace through Agent-based Workflows: MOPPET", Intl. Journal on Digital Libraries, Vol.3, No.1, pp. 36-59, (2000) .

[3] Bichler, M., C. Beam, and A. Segev. OFFER: A broker-centered object framework for electronic requisitioning. In IFIP Conference "Trends in Electronic Commerce (1998).

[4] Boll, S., A. Gruner, A. Haaf, and W. Klas. EMP-a database driven electronic marketplace for business-to-business commerce on the internet. In Journal of Distributed and Parallel Databases, Special Issue on Internet Electronic Commerce, Vol. 7, No. 2, April (1999).

[5] Chavez, A., P. Maes, "Kasbah: An Agent Marketplace for Buying and Selling Goods", Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, (1996).

[6] Ghenniwa, H., and M. Kamel ``Interaction Devices for Coordinating Cooperative Distributed Systems"; Automation and Soft Computing, Vol.6, No.2, pp.173-184, (2000).

[7] IBM Corporation CommercePOINT Payment , available at <http://www.internet.ibm.com.commercepoint.payment>

[8] Intershop Communications, Inc. Intershop 3, (1998), available at <http://www.intershop.com>

[9] KameL, M., and H. Ghenniwa, ``Coordination of Distributed Intelligent Systems", in Soft Computing: Fuzzy Logic, Neural networks, and Distributed Artificial Intelligence; F. Aminzadeh and M. Jamshidi(Eds.), Prentice-Hall, (1994).

[10] Microsoft Corporation. Internet Commerce, (1998), available at <http://www.microsoft.com>

[11] Oracle Corporation. Oracle Internet Commerce Server, available at <http://www.oracle.com/products/asd/ics/ics.html>

[12] Paperexchange Marketplace, available at <http://www.paperexchange.com>

[13] Reich, B., and I. Ben-Shaul. A componentized architecture for dynamic electronic markets, in ACM Sigmod Record (special section on Electronic Commerce), Vol. 27, No. 4, pp. 40-47, December (1998).

[14] SAP Services Marketplace, SAP AG, available at <http://www.sap.com>

[15] Sun Microsystems. Java Electronic Commerce Framework (JECF), available at <http://www.javasoft.com/products/commerce>

[16] Tsvetovatyy, M., M. Gini, B. Mobasher, Z. Wieckowski, "MAGMA: An Agent-Based Virtual Market for Electronic Commerce", Applied Artificial Intelligence, special issue on Intelligent Agents, N. 6, September (1997).

[17] VerticalNet® Marketplaces, available at <http://www.VerticalNet.com>

Incorporating the Knowledge Management Cycle in E-Business

Ying-Lie O
Subdepartment of Information Systems, Department of Computer Science
University of Twente, The Netherlands
Email: ying@cs.utwente.nl

Abstract

In e-business, knowledge can be extracted from the recorded information by intelligent data analysis and then utilised in the business transaction. E-knowledge is a foundation for e-business. E-business can be supported by an intelligent information system that provides intelligent business process support and advanced support of the e-knowledge management cycle. Knowledge is stored as knowledge models that can be updated in the e-knowledge management cycle. As illustrated in examples, the e-knowledge cycle aids in the business decision taking, production management, and costs management.

Keywords

e-business, intelligent data analysis, intelligent information systems, knowledge management.

INTRODUCTION

Knowledge management is an issue for both enterprise organisation and management, as well as the support of advanced information technology. More specifically, enterprise knowledge management [10] entails formally managing knowledge resources in order to facilitate access and reuse of knowledge. Knowledge management is becoming a critical success factor for enterprises [11].

A business model generally describes the way business is performed according to the business goals and strategies. The business process model consists of activities that support the business model. A *business transaction* is a chain of business processes associated with business value, exchange of information between customer, business, and other involved parties, and product handling from the beginning to the end. Knowledge management can be associated to the business transaction by indicating the use and creation of knowledge in the business processes.

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

KNOWLEDGE MANAGEMENT IN E-BUSINESS

In e-business, all business transactions and associated data are recorded. This is an important resource of knowledge about the business that can be used for decision making. Knowledge can be made available using advanced information and communication technology as e-knowledge. It is suggested, that e-knowledge is a foundation for e-business [5]. E-knowledge is not only limited to descriptive knowledge or information, but also includes procedural and reasoning knowledge.

In the following, the term "knowledge" shall refer to knowledge that can be actively used in the generation of new information, that is procedural and reasoning knowledge. Descriptive knowledge, structured documents, associated data sets with meaning, and alike shall be referred to as "information".

The e-Knowledge Management Cycle

In the e-knowledge management cycle, business processes use information and knowledge to create new information [3]. Specific knowledge is extracted from this information by knowledge discovery and then utilised in the business transaction. The resulting knowledge management cycle is depicted in Fig. 1.

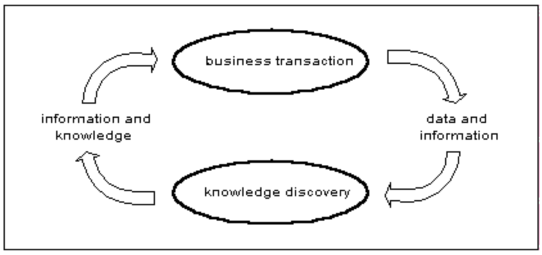


Figure 1: The e-knowledge management cycle

More precisely, all data and information associated to a business transaction are stored. This extends the more common case of only storing the customer and purchase data, thus allowing advanced analysis.

Information about products and production such as product and parts catalogues are included in the *business content* (the "what"). All kinds of knowledge and information for decision support in business is called *business intelligence* (the "how"). In general, business intelligence contains a combination of fixed business rules and adaptive business decision support.

The latter is extracted from stored data using intelligent data analysis. The analysis is applied on subjects of interest such as the group of customers, the set of products, the set of transactions, and so on. Results of analysis may also give raise to changing business policies. This yields flexibility to adapt to rapid changes in the business environment. Specifically, it can be used for knowledge-based marketing [12].

Information and Knowledge Flow

The e-knowledge management cycle is focussed on information and knowledge flow. In Fig. 2, a typical business process model of an e-business transaction is given. The upper part concerns the business processes, while the lower part depicts the specific information and knowledge required for the "e-business intelligence". The arrows between business processes indicate flow control that may include information and knowledge flow. The lines with the bullet heads show the information and knowledge flow.

All data associated to the business transaction are stored. A transaction may be completed successfully, cancelled in one of the processes, or completed with complaints and possible settlements. This is in contrast to conventional business, where only the actual purchases are registered.

These data are useful for generating business reports, and more important, for extracting knowledge by *intelligent data analysis*. This provides an excellent view on the business environment. Conventionally, data analysis is only performed periodically to generate business reports. Beside these periodical runs, intelligent data analysis can be performed after each transaction, or if the situation gives rise to. There are several different analyses that involves information and knowledge flows:

- *Periodic reporting* may give rise to refine, update, or even change the associated knowledge.
- In each *transaction* as a check and detection of specific situations. Small updates can be implemented automatically.
- On *detection* of a specific situation. Especially if it is adverse, it may require more investigation, and as a result changes in policies.

The most important information in the *business transaction* is associated to the *quotation* as it progresses in time.

$$
\begin{array}{l}
\mathit{request\ for\ quotation}: \\
\quad (\ \mathit{requested\ product\ specification},\ \mathit{price\ range}, \\
\qquad \mathit{requested\ delivery},\ \mathit{completed\ or\ cancelled}\ ) \\
\mathit{quotation\ proposal}: \\
\quad (\ \mathit{proposed\ product\ specifications}, \\
\qquad \mathit{proposed\ price\ ranges}, \\
\qquad \mathit{expected\ delivery\ ranges}, \\
\qquad \mathit{completed\ or\ cancelled}\ ) \\
\mathit{ordered\ quotation}: \\
\quad (\ \mathit{product\ specification},\ \mathit{sales\ price}, \\
\qquad \mathit{planned\ delivery},\ \mathit{completed\ or\ cancelled}\ ) \\
\mathit{completed\ transaction}: \\
\quad (\ \mathit{product\ specification},\ \mathit{sales\ price}, \\
\qquad \mathit{actual\ delivery},\ \mathit{payment},\ \mathit{settlement}\ )
\end{array}
$$

The *specification and pricing* part controls the knowledge for the quotation and product specification. The product specification is is related to the production specification. The results of the intelligent data analysis are used to update and refine the required knowledge. Results may also give raise to changes in the production management.

Product Specification and Pricing

In e-business, the average customer should be able to specify the desired products online without the interference of a salesperson or an engineer. In addition, customers expect "online" speed. These requirements limit the complexity of the products. There are three categories of suitable tangible products:

- Off-the-shelf goods are products that do not need any processing before delivery (except packaging). These goods are fully specified by the catalogue of products, and only need intelligent searching. There are already quite a number of e-businesses of this type, such as booksellers.
- Composed products are assembled from ready parts on the customer's specification. The parts are selected from catalogues, there are constraints on connections between parts. Therefore, examples and standard templates are important. These products are built-to-order, and there are some possibilities to change the configuration afterwards. There are a number of e-businesses, for instance PC shops.
- Flexible manufactured products are manufactured according to the customer's specification using highly advanced multi-purpose production machines. Similar products can be manufactured in different ways, therefore the production specification is also needed. Because of its high flexibility, it is necessary to provide examples and templates. These products are customised built-to-order, and can not be changed after the production.

Catalogues of products are part of the *business content*.

Pricing of composed products and flexible manufactured products are elaborate. The price is composed of the estimated cost, additional margins, and profit [6]. Accurate calculation of the estimated costs can be fairly complicated, especially in flexible manufacturing [7]. Costing is usually based on *activity based costing*, thus attributing any expenses directly to the product. Activity based costing can be integrated in the enterprise system [13].

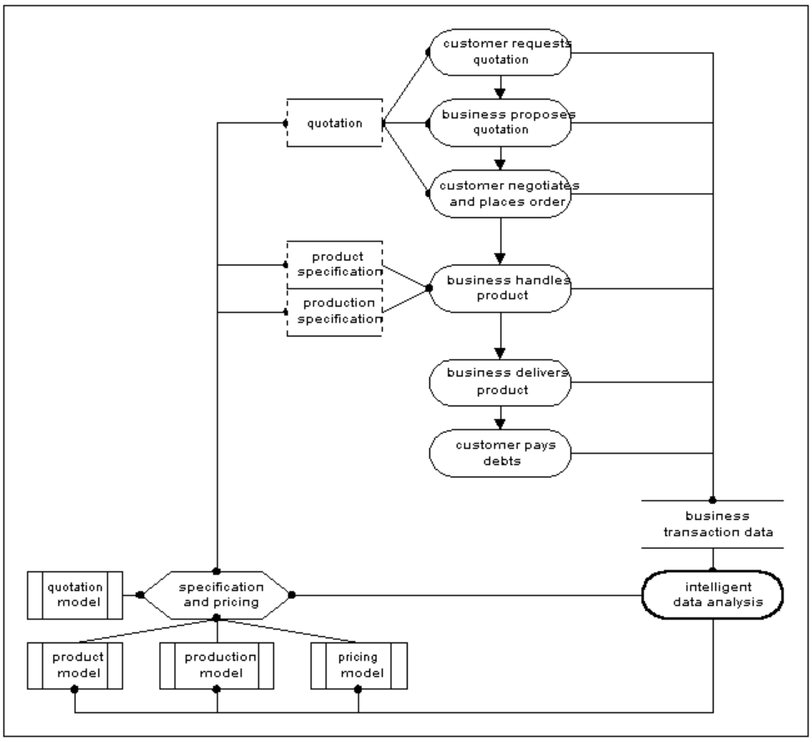


Figure 2: The information and knowledge flow in the e-knowledge management cycle

The following *pricing model* is proposed:

$$\begin{aligned}&calculated\ price = estimated\ costs + \\ &\quad risk\ with\ cost\ variances + \\ &\quad risk\ with\ estimation\ mistakes + \\ &\quad markup\ on\ materials + \\ &\quad markup\ on\ production\ costs + \\ &\quad profit\ margins\ ; \\ &estimated\ costs = estimated\ material\ costs + \\ &\quad estimated\ production\ costs + \\ &\quad estimated\ overhead\ costs\ .\end{aligned}$$

The final *sales price* is determined by the *pricing policy* that may depend on quality, quantity, and delivery; the customer, market situation, and other circumstances. For instance, "good customers" are rewarded by discounts, there may be a contract, potential customers can be attracted by special offers, special or rapid deliveries in a busy period are more expensive. Pricing can be adjusted by a simple decision rule, for example: if $customer\ credits > 10000,$ then $sales\ price\ =\ 90\%\ calculated\ price$. Decisions and the way regarding business is handled is *business intelligence*.

After each transaction, the realised costs can be computed and compared to the estimated costs. The $cost\ difference$ is used to improve the costing, especially for handling inaccuracies, and to recalculate the risks and margins.

$$\begin{aligned}&realised\ costs = material\ costs + \\ &\quad production\ costs + overhead\ costs\ ; \\ &cost\ difference = \\ &\quad estimated\ costs - realised\ costs\ ; \\ &profit = sales\ price - realised\ costs\ .\end{aligned}$$

The realised costs is the *price floor*, the lowest possible price without making losses. The highest price that the customer is willing to pay, the *price ceiling* is determined by the market.

AN INTELLIGENT INFORMATION SYSTEM FOR E-BUSINESS

An *intelligent information system* provides the usual *information system* functionality and *artificial intelligence* functionality in an integrated manner. It is a combination of an information system and an associated *expert system*. The intelligent information system for e-business should provide:

- intelligent business process support, especially in the online interaction;
- business content and business intelligence for adequate support of the quotation;
- advanced support of the e-knowledge management cycle.

The functionality correspond to those indicated in Fig. 2.

Intelligent Business Process Support

The major difference between conventional business and e-business is the online interaction with the customer, and that all information is recorded. Interaction is accomplished through an intelligent electronic form the *e-quotation*, that is generated according to a flexible *quotation model*. The quotation must be fully customised to the customer's needs, and at the same time transparent and clear in its use. To support online negotiations, the proposal should provide several product choices with proposed quantities, delivery times, and prices on a customer's request.

To the customer, the quotation contains a subset of the product specification that is understandable. It should also provide sufficient information on the products that meet the specification and further processing of the order. For further handling, the full product and production specification is needed. For convenience, frequent customers may have their own quotation and product templates. The business may also supply standard templates that may relate to the outcome of data analysis, for instance to promote new products.

The *specification and pricing* part is an intelligent part that provides expert system functionality for the quotation. The *product model* and *production model* are typically a combination of catalogues (business content) and business rules (business intelligence). The structure of a product model is basically hierarchical with interconnections and constraints [8]. Product choices are generated by matching the specified products to the production capabilities. The *pricing model* is generally highly adaptable to the business environment, such as changes in product costs, customer behaviour, and market demand.

The knowledge models consist of

- A qualitative definition part with a fixed structure that can only be changed by remodelling. In the product model, it is represented by an abstract hierarchical structure and interconnections of parts and materials. The production model is a sequence of manufacturing activities. The pricing model is determined by the formulas given earlier.
- A quantitative specification part that can change according to data. A product specification is a product model in which the parts and interconnections are filled out. The same applies to the production specification of this product. In pricing, the variables are replaced by values according to real data.
- An intelligent part that mainly contains decision support functionality, usually as logical rules and facts. In the quotation proposal, several options of similar product specifications are offered. The intelligence handles the matching of the product specification to available production machines in conjunction with additional business policies.

Intelligent Data Analysis

Intelligent data analysis [2] and data mining [4] has become an important issue because of the availability of large amounts of stored data. Data analysis is the process of transforming data into a meaningful form ("information"), generally different from the original format. Intelligent data analysis (IDA) is the analysis of complex data sets using artificial intelligence methods, thus extracting a knowledge model. The knowledge model is then used for further analysis and decision support.

From the viewpoint of application, data mining and IDA can be divided into four categories:

- *Summarisation* gives a compact description, generally by simple statistical analysis and tabulating the results such as mean, median, and standard deviations. Advanced multivariate visualisation at lower dimensions give an understandable overview of expected patterns and concepts. More advanced methods are the discovery of simple patterns that characterise the data as a small number of dominant concepts, relations, and deviations.
- *Dependence* describes significant deterministic or probabilistic dependencies between concepts. It is described in two levels: qualitative specifies which variables are dependent on each other, and quantitative specifies the strength of the dependence as a numerical value. Opposite to dependence is the detection of significant changes in the data from the previous or normative value. This includes the determination of deviations and outliers in a dependent group.
- *Grouping* divides into different homogeneous groups with clear concepts. The groups are fully determined by the concepts and the dependence on the concepts as variables. In *classification*, the groups are disjoint, while in *clustering* the groups may overlap. The number of clusters may be unknown (unsupervised), while the number of classes are fixed (supervised).
- *Prediction* is the estimation of an unknown or future values based on other known values, also referred to as forecasting. Prediction is the primary goal of IDA, on the other hand, the resulting knowledge model of data mining can be used for problem-solving. Grouping can be used for prediction by assigning the case to a group, and using the characteristics of the group to determine the unknown variables.

The main goals of data mining are *description* and *grouping*. Description is generally performed before any other analysis in conjunction with pre-analysis of the data. The primary goal of IDA is problem-solving, in particular *prediction*. In a data-driven situation, the actual IDA is preceded by data mining.

A number of important aspects of the business environment can be analysed:

- *Customer profile* by grouping according to characteristics such as purchases, products, and interaction behaviour. For instance, the groups of customers who visit and do not buy, visit and buy later, or visit and buy immediately can be visualised.

- *Product profile* by grouping and ranking according to popularity and associations to customer profiles. It is also interesting to know which products are purchased in combination.
- *Production method* with regard to efficiency and quality. This may lead to using specialised machines instead of multi-purpose machines for large amounts or special quality.
- *Pricing policy* associated to customer profile, product profile, and production method. Good customers may be rewarded by discounts, while popular products can be priced more accurately.
- *Sales forecast* for different types of products. This may lead to decisions on the product range.

Incorporating the e-Knowledge Management Cycle

Information systems commonly already incorporate small parts of the knowledge management cycle, as periodic reporting (summarisation). Other analyses may be periodical, but are usually invoked manually.

Fully automatic incorporation of the e-knowledge management cycle is not all that simple. In particular, control on updates of the knowledge models should be considered carefully as it may introduce inconsistencies. Changes to the quantitative part can either be implemented directly, or stored first and implemented after approval. Both the reason to, and the change should be verified and validated. In any case, deviations always need additional judgement to find out what the underlying reasons are.

It may thus be better to activate the cycle periodically, for instance in conjunction with the routine back-up. In fact, there are several knowledge management cycles simultaneously, each requires a different approach.

- *Periodic reporting* for customer reporting, product reporting, and production reporting tabulates characteristics of the subject in a period of time. Reports may give raise to business decisions.
- *Profiling* is determined from periodic reporting resulting in the customer profile, product profile, and production method by grouping. Profiles can be updated automatically by assigning each new case to a group. Failure can either mean that this case is a deviation, or that the groups are not appropriate and need to be renewed.
- The *transaction cycle* is activated at each transaction. It controls the most important information associated to the business transaction mentioned earlier and can therefore detect deviations.
- The *recalculation cycle* is invoked after each transaction and periodically. For pricing, it compares the calculated prices to the realised ones. It should signal on adverse situations, such as losses or other deviations. The result may be used to automatically update the variables of the pricing model.
- *Forecast* may be based on the temporal behaviour of the profiles. This may lead to decision on the product variety, production capacity and utilisation.

EXAMPLES

Because of its nature, it is expected that the proposed approach will be most beneficial for *flexible manufacturing*. There are several relevant flexible manufacturing industries: graphic arts, textile printing, plastics and multi-component moulding, packaging, cable industry, and metal industry. Although the above products seem very different from each other, the business characteristics are similar. All products are ordered to customer's specification, allow for mass customisation, are produced on demand by highly advanced machines, and have a short production life cycle. Graphic arts serves as a good example because it is widely available to average customers.

Varnavarni, a Graphic Arts Business

Graphic arts is an example of flexible manufacturing that has been around for a long time. Nowadays, the production machines are highly advanced and multi-purpose. The case presented here is based on PricingOnDemand [14], papers on this subject such as [9, 1], internal reports, examples of orders, and interviews with domain experts.

Varnavarni is a small e-business enterprise in quality printing of artworks such as posters, flyers, and postcards. All interactions with customers are fully online. The artworks are supplied online by the customer in one of the acceptable file formats.

The product characteristics for the quotation are:

- Image resolution: medium to high.
- Colours: black and white or colour in different qualities.
- Paper: a large variety of types, quality, thickness, and size.

The business has 3 types of production machines:

- Machine type A: 3-colour medium resolution, medium quality, high speed printer. There are 3 printers of type A: A1, A2, and A3.
- Machine type B: 3-colour high resolution high quality photo printer. There are 2 printers of type B: B1 and B2.
- Machine type C: 6-colour high resolution super high quality photo printer. There is 1 printer of type C.

A customer places a request for a black and white artwork in a limited number of high quality posters and a large number of medium quality flyers.

Request for quotation
Product specification
High quality

- Image resolution: high.
- Colours: black and white, high quality.
- Paper: glossy quality photo paper, size 600 mm x 900 mm.
- Amount: 13.
- Price range:

Standard
- Image resolution: medium.
- Colours: black and white, medium quality.
- Paper: fine quality, size 200 mm x 300 mm.
- Amount: 100.
- Price range: approximately 200.

Delivery: in 1 day.

Quotation proposal
High quality
- Image resolution: high.
- Colours: black and white, high quality.
- Paper: glossy quality photo paper, size 600 mm x 900 mm.
- choice 1 (type B): amount 13 price 335,
- choice 2 (type B): amount 15 price 355,
- choice 3 (type C): amount 13 price 400,
- choice 4 (type C): amount 15 price 425.

Standard
- Image resolution: medium.
- Colours: black and white, medium quality.
- Paper: fine quality, size 200 mm x 300 mm.
- choice 1 (type A): amount 100 price 185.
- Paper: super quality, size 200 mm x 300 mm.
- choice 2 (type A): amount 100 price 210.

Delivery: in 1 day. No additional fast delivery costs.

An approximate price range yields an interval of ± 10%. This customer has already purchased several times, therefore there are no additional fast delivery costs. The additional choices for the amounts of 15 instead of 13 is to promote the high quality products. Because the image is black and white, it does make sense to choose the 6-colour machine type C. Such irrelevant choices can be avoided by putting more intelligence in the product matching part.

Ordered quotation
High quality
- Image resolution: high.
- Colours: black and white, high quality.
- Paper: glossy quality photo paper, size 600 mm x 900 mm.
- choice 2 (type B): amount 15 price 355.

Standard
- Image resolution: medium.
- Colours: black and white, medium quality.
- Paper: super quality, size 200 mm x 300 mm.
- choice 2 (type A): amount 100 price 210.

Delivery: in 1 day. No additional fast delivery costs.

Examples of Knowledge Models

The products are printed graphics artworks on paper, and the production is the process of applying ink on paper according to the electronic source supplied by the customer. This is flexible manufacturing performed automatically by a single machine. The production method is fully determined by the machine by adjusting the controls at set up. If needed, paper, ink, and machine parts are replaced before adjusting the controls.

The production model is similar to the product model, the qualitative definition part contains the following items:

Product model	*Production model*
artwork in electronic form	
	production machine
paper type, quality, weight	paper type, quality, weight
paper size	paper size
ink colours, quality	ink colours, quality
image resolution	image resolution
	machine set up
	material waste
	machine operation

Paper size is a separate item, because it depends on the machine. The quantitative part gives the actual values in the model for further calculation. For example:
paper size: type = sheet; size = A3
paper size: type = roll; width = 600 mm, length = 9 m
machine operation: machine speed = 5000 standard imprints per hour.
Material waste is caused by misprints given as a percentage of a total of similar products for a machine type during a certain period.
material waste: if production machine = C; ink colours = full colour, high quality; then misprints = 10 %.

The choices of suitable production machines are determined by matching the items in the product model to the available production models.

The pricing model is determined by combining all items in the product model and production model. The estimated material costs is due to paper, ink, machine parts, and material waste. The estimated use of paper can be calculated accurately by matching the size of the product to the size of the paper in the machine.

The use of ink depends on the coverage of colours in the artwork and on the absorption of the paper. The latter can only be estimated from the technical specifications of the ink and paper manufacturers, and by experiment. The deviations of the real values from the initial specifications are unfortunately too significant to be included in the risk with estimation mistakes or markup on materials. These specifications merely give a lower bound.

Examples of the E-knowledge Management Cycle

In this section, a number of aspects common to this industry are mentioned. The application of the e-knowledge cycle, as well as the intelligent data analysis method and the required data are indicated.

The periodic product reporting revealed that there are approximately equal amounts of colour prints and black and white prints. It is therefore decided to use printer B1 for black and white prints, and printer B2 for colour prints. This decreases the machine set up time. Most reporting is generated by statistical summarisation on quotation data.

In the transaction cycle, it is found that an order has made losses. This could be corrected for by directly increasing the profit margin. But the reason for this were some unexpected problems during the set up which significantly had increased the set up time, and thus the job costs. In the intelligent data analysis, this is classified as an outlier.

The amount of misprints and the ink use have to be computed in each recalculation cycle. These analyses also require production management data such as job times, actual produced amounts, actual material use, etc.

In the forecast analysis, it is found that there is a decrease in cancelled quotations of high quality products. Forecast analysis on the grouping shows that this group is steadily growing. Forecast by itself generally only requires quotation data, but the determination of the associated capacities needs production management data.

CONCLUDING REMARKS

As illustrated in the examples, the e-knowledge cycle aids in the business decision taking, production management, and costs management. The e-business processes and the above features must be supported by an intelligent information system. The design of the intelligent information system requires thorough understanding of the specific industries and the associated production management.

Mostly, business transaction data are only retained for financial purposes, thus limited to completed transactions. Production management data and production data are usually registered separately in each production machine in different formats. Most of the data is not in a suitable format for analysis. This means, that the integration of data and interfaces to existing information systems and the production machines is a significant part of the design.

Further research will be dedicated on design and modelling. It is certainly a challenge to define generic knowledge models that can be utilised for different industries. In addition, the control of updates on these models according to the knowledge management cycle calls for thorough investigation as it may introduce inconsistencies.

ACKNOWLEDGEMENTS

The author is indebted to Huub van't Hart, Sylvia de Vries-Wennink, and Paul Gimberg, all with FrontEnd BV, for providing valuable domain expertise. Paul Gimberg has also given detailed comments. This research is jointly supported by the Centre for Telematics and Information Technology, the chair of Information Systems, Department of Computer Science, University of Twente, and FrontEnd BV.

REFERENCES

1. S. Almutawa and Y. Moon. The development of a connectionist expert system for compansation of color deviation in offset lithographic printing. *Artificial Intelligence in Engineering*, 13:427–434, 1999.
2. M. Berthold and D. Hand, editors. *Intelligent Data Analysis*. Springer, 1999.
3. U. Borghoff and R. Pareschi, editors. *Information Technology for Knowledge Management*. Springer, 1998.
4. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / MIT Press, 1996.
5. C. Holsapple and M. Singh. Toward a unified view of electronic commerce, electronic business, and collaborative commerce: A knowledge management approach. *Knowledge and Process Management*, 37:151–164, 2000.
6. B. Kingsman and A. de Souza. A knowledge-based decision support system for cost estimation and pricing decisions in versatile manufacturing companies. *Int. J. Production Economics*, 53:119–139, 1997.
7. T. Koltai, S. Lozano, F. Guerrero, and L. Onieva. A flexible costing system for flexible manufacturing systems using activity based costing. *Int. J. Production Research*, 38(7):1615–1630, 2000.
8. T. Männistö , H. Peltonen, T. Soininen, and R. Sulonen. Multiple abstraction levels in modelling product structures. *Data & Knowledge Engineering*, 36:55–78, 2001.
9. K. Musselman. Complex scheduling of a printing process. *Computer & Industrial Engineering*, 39:273–291, 2001.
10. D. O'Leary. Enterprise knowledge management. *IEEE Computer*, 9:54–60, 1998.
11. D. O'Leary and R. Studer. Knowledge management: An interdisciplinary approach (editorial). *IEEE Intelligent Systems*, 16(1):24–25, 2001.
12. M. Shaw, C. Subramanian, G. Tan, and M. Welge. Knowledge management and data mining for marketing. *Decision Support Systems*, 31:127–137, 2001.
13. I. Tatsiopoulos and N. Panayiotou. The integration of activity based costing and enterprise modeling for reengineering purposes. *Int. J. Production Economics*, 66:33–44, 2000.
14. H. van 't Hart and S. de Vries-Wennink. Pricing on Demand. FrontEnd BV, www.frontend.nl, 2000.

Capturing the knowledge about the user for e-commerce applications

Rony Abi-Aad Alina Andreevskaia T. Radhakrishnan
Department of Computer Science, Concordia University
Montreal, Quebec, Canada
{abiaad, andreev, krishnan} @cs.concordia.ca

Abstract
Software agents are becoming more and more popular in electronic commerce. In this case, one of the major components of agent's knowledge base is the information about the user, or the user profile. Capturing the information about the user is a challenging task that requires specific tools. A new user model is proposed in this paper and its potential use in e-commerce applications is discussed. Based on this model, a software tool to capture the knowledge about users is currently being implemented. Different methods for capturing such knowledge are presented.

Keywords
E-commerce; Agent; Knowledge acquisition; User profile; User modeling

SOFTWARE AGENTS IN E-COMMERCE

Electronic commerce is a fast developing area. One of the recent studies found that "business-to-consumer e-commerce has reached $33.1 billion last year" [17]. Agent technology is very popular in this area. There are hundreds of software agents already operating in the virtual marketplace. The main effort in creating agents for electronic commerce has gone in two major directions: agents that serve as virtual sales representatives explicitly or implicitly representing the vendor, and price comparison "bots" [1], [8] [6]. Currently an end-user who goes shopping on the Internet has rather poor choice of tools that can assist him/her. Most e-malls offer some questionnaires or fill-in forms that are supposed to help the user in information retrieval, but these tools lack flexibility (for example, while selecting a language of a book during search on amazon.com the user can only choose between a book in any language or a book in Spanish). Results of "personalized" search are often frustrating. The list of hits is usually too long and contains large numbers of items that do not really correspond to user's requirements.

If we look at the current state of e-commerce web-sites and the kind of assistance they offer to consumers we see that certain aspects of shopper's behavior are underserved. In this context, we identified three different types of shopping behaviors [10]:

- *Comparative shopping* when the user has a clear idea about product category that he/she is looking for, has limited time to make his/her buying decision, and wants to compare similar products to find the best match for his/her requirements, preferences and constraints.
- *Planned shopping* is very similar to comparative shopping but the time constraint in fulfilling the shopping need is much more relaxed.
- *Browsing-based shopping* occurs when a person is just gathering information and/or socializing with other people while browsing.

Of these three, comparative shopping is more complex and requires more sophisticated tools than price comparison "bots" available now. If a customer wants to include constraints other than price range he/she gets no software support. Meanwhile, time is crucial in the case of comparative shopping and software tools that can save user's time (for example, agents or sophisticated search tools) are especially relevant. They allow the buyer to make more informed decision in the available time and thereby can create more user satisfaction.

Whichever type of shopping behavior we consider the shopping process can be subdivided in a number of stages. The consumer buying behavior (CBB) model [5], [16] is one of the popular approaches to the analysis of phases of the buying process. Six major stages are defined in the CBB model:

1. Need identification.
2. Product brokering.
3. Merchant brokering.
4. Negotiation.
5. Purchase and Delivery.
6. Service and Evaluation.

The major focus of applications of agent technology to electronic commerce has been on the negotiation stage

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

(on-line auctions of different kinds), and to some extent on product and merchant brokering [16]. Currently, as can be seen from business reports, the main effort in the area of electronic commerce is in: (a) targeting products and services to selected list of specific consumer groups; (b) retaining the customer base. We believe that in retail e-commerce, product and merchant brokering should be coupled with special attention to the post-purchase stage. The companies should not only care about being known but also about knowing the customers and their personalized needs. Gathering customer's feedback after the purchase is also crucial for other purposes. All this information will help the merchant to understand the relative importance of product and service features (that range from how many buyers would prefer to have a deferred payment option to what color or style of T-shirts is the most popular this season). On the other hand, the consumer should have an incentive to fill in all the on-line questionnaires before and after the purchase and to give personal information to the third party.[1] We suggest a user agent that will acquire information about the user and use it to help the buyer in his/her shopping. The same agent works with multiple vendors (in our prototype it is used in a virtual mall with several boutiques). The agent is perceived by the user as his/her personal adaptive shopping assistant. From user's perspective the two main goals of the agent are facilitation of electronic shopping and monitoring events relevant to the user (sales, promotions, new products, etc.). The vendor or his/her authorized software sales agent may be permitted to share the information possessed by a user agent.

KNOWLEDGE ACQUISITION FOR SOFTWARE AGENTS

Software agents can be viewed as knowledge-based systems. They have to contain knowledge about the task domain (domain model) and knowledge about the user (user model). Domain knowledge acquisition has been at the center of research for more than two decades and important research advances have been made and practical applications have been created in this area. Recently the trend has been towards end-user programming and creating applications that assist in knowledge acquisition (PROTÉGÉ-2000, WebGrid, recent interfaces to EXPECT).

Acquisition of the user model has received less attention as a separate problem. In the same time, "modeling and adapting to user's preferences seems to be the most successful approach by now" [7]. A new user model that we propose is discussed in the following.

[1] Privacy and trust are important issues but they are not addressed here.

What Parameters to Acquire?

The set of parameters that form the user profile can be categorized as follows:

1. Related to the task domain (on-line shopping context means that the consumer is interested in prices, warranty, return policy, similar to traditional shopping, but some other parameters should be added, for example, shipping prices or time of delivery, or even weight of the item);
2. Related to product domain (parameters used in shopping for apparel would be different from ones applicable to toys);
3. Related to information about merchants (there are sets of common characteristics used by merchants to describe products; the agent should know what significance to assign to them and how important they are).

Based on these three categories a set of initial parameters is defined for our prototype. After the core user profile is created new parameters can be added based on user's actions, new information gathered and so on.

First, we focus on a model of the user. Our approach to user modeling [2] was inspired by Elaine Rich's work on Grundy system [13]. Grundy has a user model based on personality traits of users. The model was proposed in 1979 by Elaine Rich [13] for recommending novels. The personality traits are elements of a set that does not form any tree or graph-like structure.

We have proposed in [2] a user model based on a directed acyclic graph (DAG) structure that consists of nodes which we call PIE (Preference Indication by Example). Associated with each node are: value denoting the degree of user's interest in that PIE and some features or attributes that can take attribute values. Figure 1 shows one of such graphs designating some user preferences. An arrow from A to B indicates that B is more specific than A.

How to Capture Parameters for User Model?

There exist three main techniques for knowledge acquisition:

1. The acquisition can be done by the knowledge engineer based on interviews. This approach is the most reliable but also the most expensive one.
2. The knowledge can be captured via a dialog between the system and the user (for example, user or domain expert). This technique requires a considerable effort from the user and careful dialog design. The advantage of this approach is increased user's trust and direct user control.
3. Machine learning techniques can also be employed. Such application is becoming more and more reliable, but there still is a relatively high percentage of errors associated with this approach.

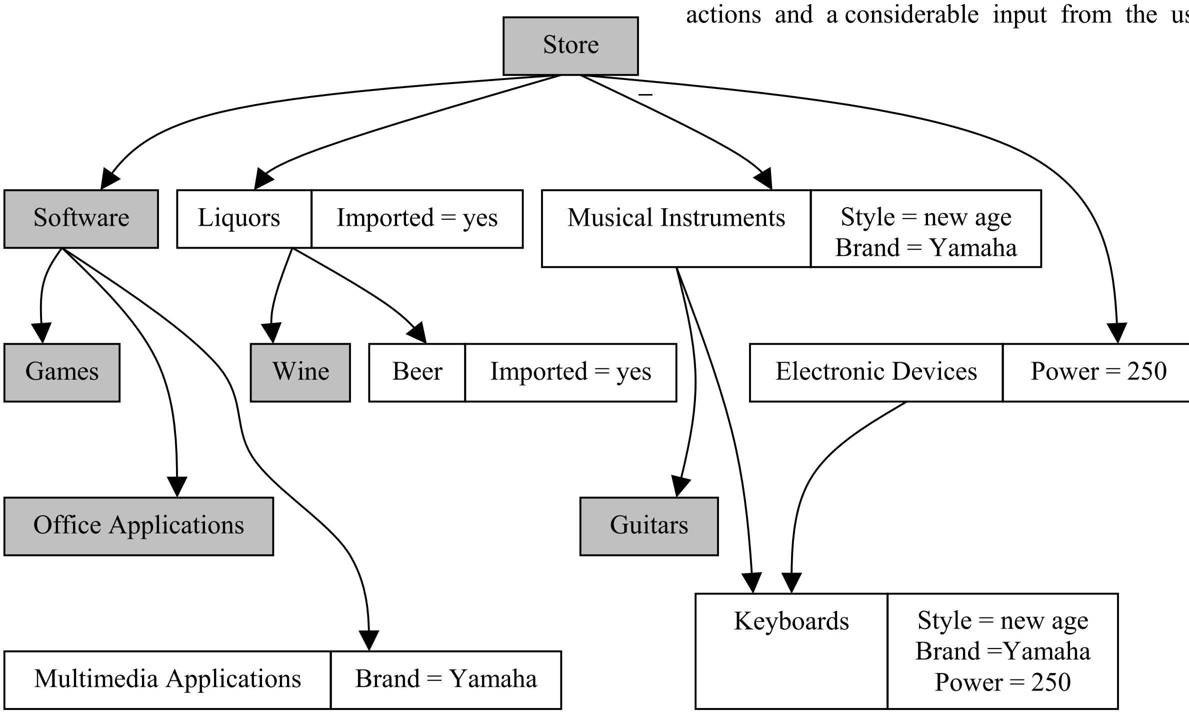


Figure 1. Example of PIE hierarchy designating user's preferences.

Various combinations of these techniques are also popular and seem to be the best solution.

These techniques have been successfully applied to domain knowledge acquisition and can be also used for creating user profiles. However, capturing the knowledge about the user presents various challenges:

– It is not possible to employ professional knowledge engineers for capturing knowledge from individuals – it will be too expensive, cumbersome, and might be unsuitable in some cases.

– Introspection seems to be a natural way to acquire the knowledge about individual's preferences and requirements but it is very error prone. Good design of knowledge acquisition dialogue and/or use of reliable techniques (such as Repertory Grid [18]) can help to reduce the level of errors but at the same time it will make more demands on user's time and effort.

– Observation ("looking over the shoulder" in terms of Pattie Maes [11]) is often employed when a personalized agent is created. It is a valuable technique but it can not be used alone because it is easy to make wrong induction based on user's actions and a considerable input from the user is required in order to make the system's reasoning worthwhile.

We suggest an approach that will combine observation (keeping a log of user actions – his/her "shopping history") with mixed-initiative dialogs between the agent and the user that can be used to clarify ambiguities and avoid incorrect inferences.

First a set of nodes of the PIE DAG is defined based on the knowledge about common shopping mall organization. This activity can benefit from product classifications[2] combined with intuitive hierarchy (based on ontology built using PROTÉGÉ 2000) that would make navigation easier for a user.

Category	Preference values	Features
CAR	5	Color = blue, red HP = 150 Brand = Mazda

[2] Parameters relevant for the product domain can also be selected based on existing product classifications (such as UPC).

Figure 2. Example of PIE node

A node in PIE has a structure as shown in Figure 2. The preference value (specific to a user) is applicable at the node level and each feature is associated with a feature-value. Some of these entries can be NULL. The initial values for PIEs are obtained when the user first registers on the e-mall's web-site. From the information gathered through fill-in form at first log-in initial values are assigned to PIEs. Stereotypes can be used at this stage (for example, an elderly lady living in the country might be interested in gardening; a teenager can be expected to buy music CDs, etc.). Of course, such inferences lead to very generalized user portrait. This information should be checked against actual user's preferences in order to fine-tune the user model. There is also a need to reflect the dynamics of user's interests that evolve over time. To ensure that the user model is personalized and up-to-date we use machine learning techniques and question-answering sessions between the system and the user for verification or classification. Further updates are made either proactively by the agent (based upon user's shopping history) or on user's initiative. The user can start the dialog with the agent in a restricted subset of natural language giving instructions to the software agent (for example, "change values assigned to cars to zero").

Prototype

Our approach is being implemented as a part of CITR project "Enabling technologies for electronic commerce" [4].

In the prototype, a person is allowed to use the system as an anonymous user. He/she can decide to use the system without registering or logging in. Obviously there will be no personalization in this case. During the login stage, first the user is given the choice to use the system anonymously, login if he/she is already known to the system, or register in the case he/she is a new user. After deciding on this, the user can see a list of links to many virtual malls, and the links to existing shops inside each mall. The user can choose between clicking on the link to a mall or to a specific shop. If he/she decides to go to a mall, he/she can then explore the 3-D environment of this mall, chat with other users inside the mall and go from one shop to another by moving a pointer. If the user decides to click on a specific shop link, he/she will have the proper browsing and searching interface for each shop. The personalization is currently implemented at the level of each shop. The user has also the possibility to alternate from the 3-D virtual mall environment to the specific vendor-managed environment also known as "store front".

The design of the architecture of the prototype included two major components:

- The database design.
- The interface and internal classes.

The database design includes two distinct databases:

i. *User database*: responsible for storing the user model.
ii. *Product database*: responsible for storing the different products in the department store.

There are eight main objects in this prototype:

i. *Login Form*: to allow registered user logging in and using the service.
ii. *Registration Form*: to allow new user registering and taking advantage of the personalized service.
iii. *Browser*: an interface to allow browsing among different departments, categories and products.
iv. *Query Form*: to query the database for specific products. Several forms are supplied for different categories of products.
v. *Proactive Notification*: a window that pops up to inform the user of good opportunities.
vi. *Product Order Form*: to allow the user submitting an order.
vii. *Database*: an object that handles all interactions between other objects and the database.
viii. *Update Entry*: an object that takes care of updating the user model based on messages from other GUI objects.

The prototype was developed in Java Swing using the Symantec Java! JustInTime Compiler Version 3.10.093(i) for JDK 1.1.x. The products-database was implemented in Access 2000.

The GUI of the Prototype

The application allows the users to query and browse the content of the database with or without registering. In order to take advantage of the user model, users must register. This is a way to illustrate the difference between personalised and non-personalised service. Registered users can log in by typing the id and password. New users can register by clicking on the "Register" button at the top of the form. The "Query" and "Browse" buttons allow anonymous user to query the database or browse its content.

Upon registration, the user is assigned an id and an initial model is stored as a PIE DAG. The initial user model is created based on the information supplied with this form.

The user can always browse by moving up or down the hierarchy or stay on the same level. This means the user can go from a department to a shelf (or category) to a specific product, or from a product level, hierarchically up to the department level. He/she is also able to go from one department to another, or from one product to another, or from one category to another (staying on the same level). While browsing, the user model is

constantly queried in order to decide how to display in the graphical form items of most interest to the user. This decision is made based on the ratings given to the nodes of the DAG at the top and middle levels. At the bottom level (leaves), the features are used to decide which products to show first.
The user profile stored in the agent's knowledge base can change and evolve over time. Therefore, in our approach, a specialized module of the system is made responsible for maintaining and updating the knowledge about the user. It can be done in dual mode – the dialogue can be initiated by the system or by the user. In system-driven dialogue the user mostly answers questions about his/her preferences, goals, constraints that the system asks in order to complete the knowledge base or to adjust it to changes in the user's behavior.

In user-driven mode the user initiates the "conversation" by making an assertion (for example, "Don't check the Z.com while looking for appliances"). A given assertion can result in different scenarios:

S1. The information is new for the system, it does not contradict any previously formulated rules or earlier acquired facts. In this case the agent will incorporate the knowledge, for example, by creating the appropriate rule in the knowledge base.

S2. The information contradicts some other rule in the knowledge base. For example, before the user asked to always go to Z.com while shopping for appliances. In such case the agent has to check the shopping history to see if the new rule is a result of some past experience. If the user recently bought a washer from Z.com and had to send it back at his/her own cost because of technical problems, the agent will add the rule and replace the older one. If there is no indication about user's reasons for the sudden dislike of Z.com the agent initiates the dialogue and poses some questions in order to be sure that the user did not make any mistake (may be he/she meant "W.com"?) and does not want to make his/her new rule less rigid (e.g., by assigning a lowest rating to the shop but not taking it completely off the list).

Usage of User Profile

The resulting user profile can be used in various ways. For example, querying can be personalized (a) either by filling some fields of a query-form in anticipation, or (b) by filtering the results of the query. In the case (a), the user can see the anticipated filling before submitting the query. In the case (b) we filter the results of a query when the response-set is too large to present in a transparent way for the user. In both cases, refining the information is based on the features in the user model. The choice of when to fill in blanks and when to filter query results is dependent on the significance of the fields used in the query. The significance can be determined in two ways. We can either assign a significance value to each field, or we can assign a significance value to each feature in the individual user model. The latter method is more complex because it requires a way of deducing what factors are important for each different user. In the current version of our prototype, we assigned a constant significance factor to each field. For example, we considered that the brand of a car is more significant than its color. Thus, if the brand field is left empty in the query, we fill it before submitting the query. But if the color field is left empty, we ignore it until we get our query results. If the number of matches is more than a threshold (set also according to user's preferences), then we show the first ten with the color that is specified in the user model. When displaying the results of the query, the "additional information" in the user model, is used to personalize the display of the information. For example, we can show the details of the car engine to an expert in cars, and we show only the car external options for a person who is not an expert.
Another important use of personalization is the proactive notification whenever the agent finds that the user might otherwise miss a good opportunity. Such agent may act on <u>the vendor's behalf</u> or on <u>the customer's behalf</u>: it can either inform the user about what is important and interesting for him/her or about what the vendor wants customers to know. For example, a vendor may announce a sale, but from a particular person's point of view prices do not change significantly enough and he/she does not really need this advertisement. Our focus is on the customer's perspective. This kind of interaction could be in the form of a pop-up window that displays a message like: "Did you know that product X is available now at the price of Y?" or "Did you know that product X has a feature Y?" Proactive notification is used during any type of shopping behavior. It can be initiated by a change in the database, a change in the user model, or an external event, like an important date (Christmas, Valentine's Day, or a birthday) [10]. We included this kind of proactive notification in our prototype. The message displayed is based on the knowledge stored in the user model, on the content of the database and on the user's additional or personal information (such as address or date of birth). The user additional or personal information, combined with an external event can trigger a proactive notification message.

Several different query forms are implemented for different types of items. Here again the user model is exploited by either filtering the results of the query when the set is too large, or by filling in missing fields' values in the form. Both services are based on the features in the user model. The query also helps updating the user model by acquiring features that were explicitly specified by the user.

Finally, the proactive notification window informs the user about suitable opportunities. Usually proactive notification is initiated when there is a match between the features of a lower level node with a high rating in the user model (product node) and an existing item in the product database. Search for matches is done once at the beginning of each session, and after that the user model is updated so that the system knows that this user has been notified so it does not keep repeating the "stale" information.

SUMMARY

The user model based on PIE can play an important role in the personalization [3], [15] and in the development of user interface agents. These agents are discussed elsewhere [2]. The PIE incorporates a user preference value at the node level and attribute values at the feature level. Capturing this knowledge about the user is a non-trivial task. In the system under implementation we use a combination of techniques:

(a) Direct user input at registration time;
(b) User input in response to a clarification dialog initiated by the user interface;
(c) Dynamic updating by the user or a user interface software agent;
(d) Memory-based learning techniques as used by P.Maes in [9].

The implementation work is not completed at this stage. After implementation, we intend to devise methods for understanding the dynamic changes made in the captured knowledge.

REFERENCES

[1] Aardsma, T. L. It pays to shop around, *Inside the Internet,* elementk Journals, August 2000, available at <http://www.elementkjournals.com/int/0008/int0081.htm>

[2] Abi-Aad, R. "A New User Model to Support Electronic Commerce," M. Comp. Sc. Thesis, Department of Computer Science, Concordia University, Montreal (2001).

[3] Adomavicius, G., and A.Tuzhilin. User Profiling in Personalization Applications through Rule Discovery and Validation *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 377-381 (1999).

[4] Enabling Technologies for Electronic Commerce (CITR), available at < http://www.citr.ece.mcgill.ca/english/enabling_tech n.html>

[5] Guttman R. H., et al. Agents as Mediators in Electronic Commerce *Electronic Markets,* vol. 8, No. 1, pp. 22-27, available at <http://www.e-global.es/011_gutmann_agents.pdf>

[6] Guttman R. H., et al. Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective, available at <http://lcs.www.media.mit.edu/people/moux/papers/icec98.pdf>

[7] Kaasinen, E. Usability Issues in Agent Applications: What Should the Designer be Aware of *VTT Information Technology FINLAND,* available at <http://atwww.hhi.de/USINACTS/agent.htm#user>

[8] Kasbah (MIT Media Lab), available at <http://www.kasbah.media.mit.edu/>

[9] Kozierok, R., and P. Maes. A learning Interface Agent for Scheduling Meetings. *Proceedings of the international workshop on Intelligent user interfaces*, 1993, pp.81-88, available at <http://www.acm.org/pubs/articles/proceedings/uist/169891/p81-kozierok/p81-kozierok.pdf>

[10] Lu, J. "User Interface Agents in Electronic Commerce Applications," M. Comp. Sc. Thesis, Department of Computer Science, Concordia University, Montreal (1999).

[11] Maes, P. Agents that Reduce Work and Information Overload, *Communications of the ACM*, Vol. 37, No.7, pp. 31-40, (1994) available at <http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.pl.htm>

[12] Muller, M. W. Managing Dialogue in a Statistical Expert Assistant with a Cluster-Based User Model, available at <http://link.springer.de/link/service/series/0558/bibs/1280/12800029.htm>

[13] Rich, E.A. User Modeling via Stereotypes, *Cognitive Science*, 3, pp.329-341 (1979).

[14] Rich, E.A. Users Are Individuals: Individualizing User Models, *International J. Man-Machine Studies*, vol. 18, pp. 199-214 (1983).

[15] Riecken, R. Growth in Personalization and Business, *Communications of the ACM*, vol. 43, issue 8, pp. 32, Aug. 2000.

[16] Roeland, D. Trading Agents, Caput Technical Computer Science, available at <http://www.cs.rug.nl/~ben/agents/trading-agents.html>

[17] Rosencrance, L. Not all e-commerce sites are losing money, availbale at

<http://www.cnn.com/2000/TECH/computing/05/15/e.commerce.makes.money.idg/>

[18] Stuart, V. Business Applications of Repertory Grid, available at <http://www.EnquireWithin.co.nz/business.htm>

Alice: An ontology based architecture for supporting online shopping

Maria De Carvalho, John Domingue
The Knowledge Media Institute
The Open University
Milton Keynes, UK
{M.C.M.De-Carvalho, J.B.Domingue}@open.ac.uk

Helgi Petursson
INNN hf.
Laugavegi 26
101 Reykjavik Iceland
H.Petursson@open.ac.uk

Abstract

This paper presents initial results from the Alice project, which aims to use enterprise knowledge to support personal shopping agents. We describe the types of knowledge which occur within retail based enterprises and describe an architecture which represents explicit knowledge within an ontology and incorporates machine learning techniques.

Key-words: personalization, e-commerce, knowledge modelling, ontology.

1 INTRODUCTION

Shopping on the internet is not always a pleasant experience. Shoppers are frequently faced with web pages containing a lot of information, the majority of which is not relevant to their current task. Often these web pages take a significant amount of time to download. These e-commerce web pages seemed to have been developed on the principle that the more the customer sees, the more he or she will buy. Moreover, these sites indicate that a significant proportion of marketing strategists and web developers are building web pages without taking into account how shoppers behave and are consequently failing to engage their customers online.

Even though companies have the necessary knowledge to build systems which meet the needs to customers, they often fail to do so – thereby losing an opportunity to offer a high quality service and gain customers' trust.

Studies of consumer behaviour in the physical world categorise shoppers into several types, each type with their own information needs. For example, experienced customers often require more information than first time customers but will make a buy/do-not-buy decision relatively quickly. Detecting expert consumers can enhance personalisation. Companies may wish to give special treatment to these consumers as they are often opinion leaders.

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

Within the Alice project we build on the above and on recent studies of online customers which indicate that:

a) it is possible to predict their behaviour patterns [1]

b) contrary to what was thought, price might be just one of many factors which influence a customers decision to buy a product [2].

We also draw upon recent work within the agent community [3] where intelligent agents which embed knowledge about customer behaviour can act as intermediaries.

This paper gives an overview of preliminary work on the Alice project where knowledge involved in commercial transactions is used to improve the shopping experience. At this stage, a demo has been developed using artificial data.

The structure of the paper is as follows. In the next section we describe how knowledge within a company can be used to make e-commerce web sites more effective. Section 3 outlines how we have characterised the main types of corporate knowledge within a set of ontologies and section 4 gives an overview of the clustering technology that we are using. This is followed by an outline of the Alice architecture and conclusions.

2 COMPANY'S SYSTEMS AND KNOWLEDGE

The main functions of a manufacturing company are production, marketing, financing, human resources, and logistics. Eventually, these functions become company's departments. Therefore, people working in Marketing Department aim to fulfil marketing functions. This is a type of departmentalisation used in most companies world-wide, however variations do exist according to the company's main business. Information systems within the company have the hard task to keep departments informed about whatever important facts happen in the company and in other departments. Each department operates an intensive exchange of information with both other departments (within the company and with other companies' department peers), and with the external world. Systems theory[i] developed during the 60's tried to explain this information chain as in the sketch shown in Figure 1.

This operational framework allows the company to work towards its main objectives. When there is no synchronisation between the company's departments to get feedback and improve each other performance, the company's results might not be achieved in a satisfactory way. Nowadays, globalisation also requires that companies work in a synchronous way in order to keep costs down, and increase profits. Furthermore, some virtual companies rely entirely on the excellence of their supply chain. In such a way the knowledge behind the transactions is also spread throughout the companies belonging to the chain.

Figure 1 – Departmental Information System

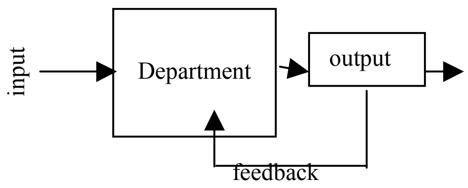
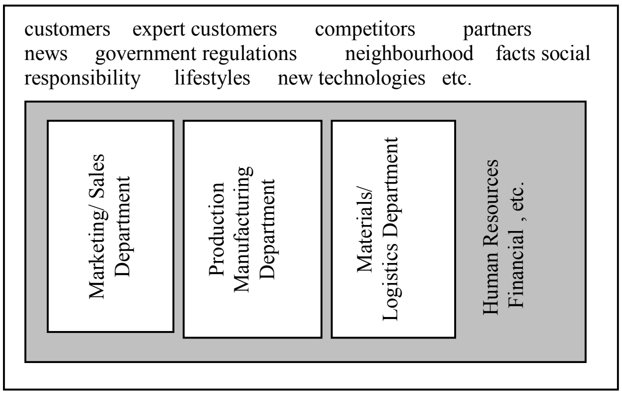


Figure 2 represents the knowledge context in which a company operates. The shadowed box indicates knowledge within the company. Outside the company (the white area) there is a huge amount of information that can influence the way the company works over which it has no power. The best a company can do is to keep updated with the changes and try to foresee how they can influence its results. The company has to figure out how this information-net works and then take advantage of it.

Figure 2 - Company interface with other systems and organisations (stakeholders)



The mismanagement of information from these stakeholders can be disastrous. For instance, an ill trained sales person might have problems identifying the customer needs and would show a customer the wrong product.

The most typical mismanagement of e-commerce companies is the lack of integration between their sellers (the web-page designer), and the other company departments such as production, materials handling, logistics, etc. The information is available, but it is not used in a proper way. Toys-r-us is one of dot.com cases that did not have the appropriate information to manage the e-commerce web-site efficiently at the beginning. As a result, after launching it and expending resources in warehousing, web page creation, etc. they could not deliver their products as promised by the site. Customers became disappointed and stayed away from their web site[ii].

3 TOWARDS ONTOLOGY

Ontology is an explicit representation of a domain of discourse [4] - a set of objects, containing relations to other objects, predicates, and functions. An ontology is usually composed of human-readable text describing a set of terms and formal axioms that constrain the interpretation and well-formed use of these terms[iii].

Within Alice, ontologies are used to represent knowledge concerning the domain of commercial transactions between companies and customers, where a commercial company-customer transaction involves the following:

i. satisfying a customer's needs through a product
ii. the company functions, lead time, information exchange, environment, personnel.
iii. the customers' geographical location related to use of product
iv. the information process between raw-material acquisition, production and delivery
v. the product (e.g. quality standards, storage, life, competitors)
vi. outside world rules to the product (e.g. government taxes, expert customers' opinions, non-governmental organisations

Correspondingly, ontologies in Alice encode knowledge about products, customers tasks (the reason of buying certain product or even the shopping task), external contexts (where the shopping, the customer, the product, and the company are acting), and media. A summary of these main types of knowledge is contained in *Table 1*.

Table 1 – The ontologies used to support e-commerce shopping

Ontology	Type of knowledge
Customers	Shopping and browsing behaviour, available finances, number of dependents
Products	Use, components, supplementary products, complementary products, geographic origin, etc.
Tasks	A monthly shop for household essentials, shopping for an evening meal
Contexts	News of social events, relevant personalities, relevant local groups
Media	Online multimedia resources

An e-commerce knowledge-based system encompasses decision-making processes that use knowledge to reasoning about what to present to the e-shopper and the specific task.

Alice uses those sources of knowledge in an integrated way so as to provide contextual and meaningful answers to customer's queries. Links to related products, past information (other sessions), and the recent past information (what has being bought in the session) within a given context (the task) provide knowledge necessary for personalization. The reasoning behind the system narrows the list of products presented to the e-shopper.

Figure 3 shows a snapshot of the ontology related to the product's knowledge (see *Table 1*), developed to implement the demo application, using OCML language (Motta, 1999).

In the demo, the class <productsdemo> inherits slots from the class <product> in the upper Alice-products ontology. Follow is part of the program for the product class in Alice-products ontology.

```
(def-class product( )
 ((has-product-name :string)
  (has-product-type :type string)
  (has-brand-name :type string)
  (has-product-description :type string)
  (has-barcode :type string)
  (has-transportation-requirements :type transport-mode)
  (has-packaging :type packaging)
  (has-validity-and-life-time :type string)
 (has-appearance :type URL)
  (has-manufacturer :type company)
  (has-country-of-origin :type country)))
```

For instance, the class <packaging> has detailed information about the way the product is packed. This can be used by the delivery department to avoid product damage.

4 PERSONALIZATION

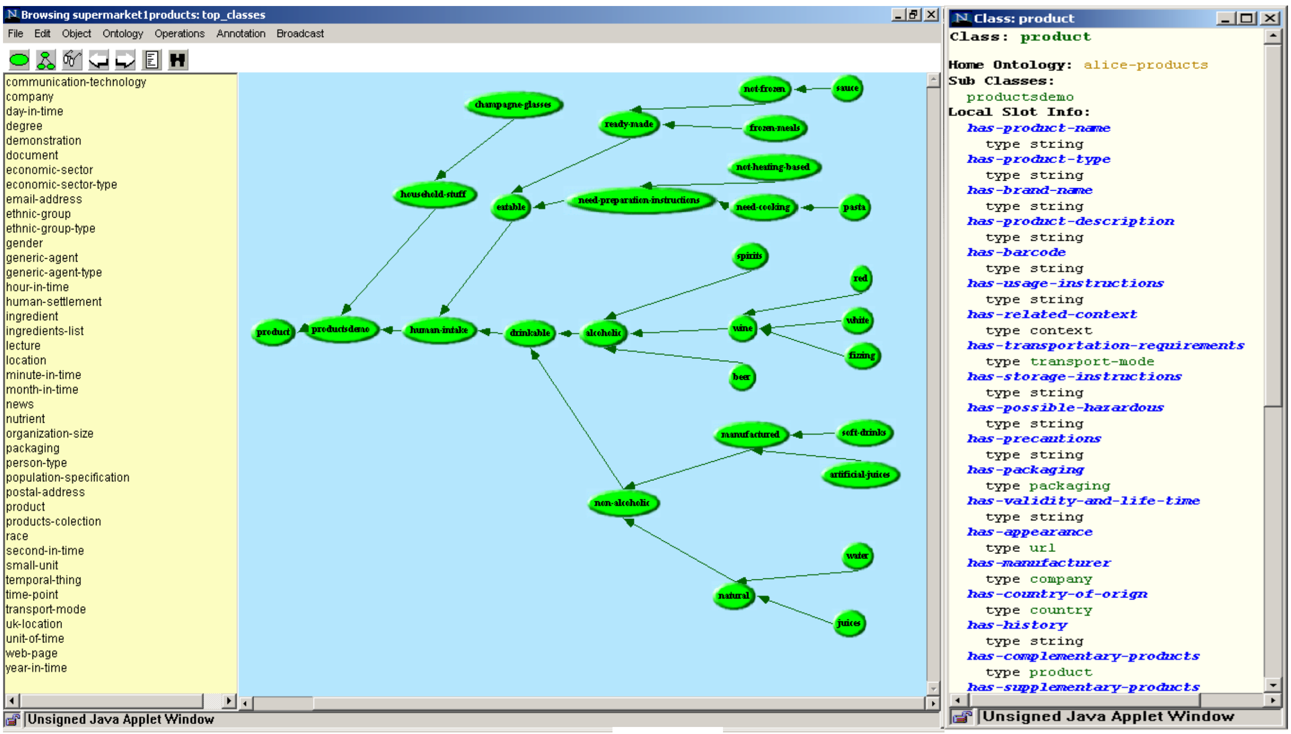
Personalization on web sites aims to present only information that is relevant to the customer. The understanding of the customer behaviour would allow one to build user models that could make the personalization task easier. Once the model is built and validated, it could be possible to forecast the customer next action, and then show what he or she wants. The main idea behind personalization is that predictive power can be obtained by capturing naturally occurring regularities in human characteristics [5]. Brynjolfsson and Smith [2] have shown that econometric models of discrete choice can also be applied to e-commerce business.

Personalisation works by filtering a candidate set of items through some representation of a personal profile or stereotype. Web sites currently use two types of filtering: content-based and/or collaborative.

- Content-based filtering recommends items based on their similarity to what the customer has bought in the past. An example of this approach is the Intelligent Personalised TV Guides [6].

- Collaborative filtering makes recommendations based on the preferences of customers from the same group. Users are compared based on how similar their ratings are, and they are recommended items favoured by other people with similar interests. A well known example of collaborative filtering is www.amazon.com. ALEXA (http://www.alexa.com) is a web browser that recommends related links based in part on other people's web surfing habits. The Microsoft Site Server also has several Collaborative Filtering tools that have been used in multiple sites[iv].

Figure 3 - Products ontology used in the Demo



The main problem in some types of business is the lack of information about customers' habits. Customers do not want to fill forms about themselves, unless they can clearly see the advantage of doing so (for instance, credit card companies often offer a prize draw for filling in a survey). Thus, it is difficult to fully understand their shopping behaviour. An alternative approach to solve this problem is to cluster customers according to their buying pattern (the shopping basket), and extract the rules from each customer cluster. In this way, the rules encode consumption patterns and act as stereotypes, which are used for personalization. A similar approach was adopted by Lawrence, et al. [7] to identify groups of shoppers with similar spending histories.

Stereotypes [8] assume that facts about people are not statistically independent. This suggests that facts can be clustered into groups that frequently co-occur. Thus, a user model built with stereotypes adds a whole cluster of user facts at once, as soon as some evidence that is known to be a predictor of the cluster is observed. Therefore, it might be possible to make predictions about the behaviour of users on the basis of an amount of evidence – which can be acquired before an action is performed. The role of these predictions is to provide a basis for an action until specific knowledge becomes available.

In this project, we are using Unsupervised Clustering [9] to build the knowledge about customers. Information used for the task is related to the customer shopping and browsing history, and the customer identification. Additional cluster stereotype features might result in changes to the ontology by the Sales/Marketing department using a customised interface. In the long run, unsupervised clustering and machine learning procedures are used to identify customers' clusters, and to extract the rules for each cluster. Afterwards, the rules are transformed into relations within the customer's ontology. The coherence of the cluster determines the reliability of the rules within the cluster. Thereafter, old customers can be instantly associated with their cluster through the newly acquired relations.

5 THE ALICE ARCHITECTURE

In this section we describe Alice architecture for supporting customers shopping on the web (See Figure 4)The architecture is composed of a server (the Alice server) built on top of LispWeb [10]. In addition to implementing the standard HTTP protocol, LispWeb server offers a library of high-level Lisp functions to dynamically generate HTML pages, a facility for dynamically creating image maps, and a server-to-server communication method.

Within the server the ontologies are represented in OCML [11] which is somewhat similar to Ontolingua [4] but has operational semantics and a built-in interpreter. As we describe in section 3, there are five main components to the ontology representing: customers, products, shopping tasks, the shopping context and related media. These components are organised within the library in three levels. At the top there are retail wide ontologies. Below there are versions of the ontologies customised for specific retail sectors. For example, within the 'white goods' sector products often have an associated insurance policy, whereas within supermarkets shelf-life is an important attribute, a characteristic that might determine the window-time for delivery. Similarly, there will be differences in the tasks, contexts and customer behaviour between retail sectors.

At the most specific level there are ontologies related to specific companies. These ontologies will support a company's specific view of retail and interoperability between Alice and the company's existing databases. The database interface module will use mappings defined within the corporate knowledge model to enable appropriate database entries, such as for products, to be accessed and modified.

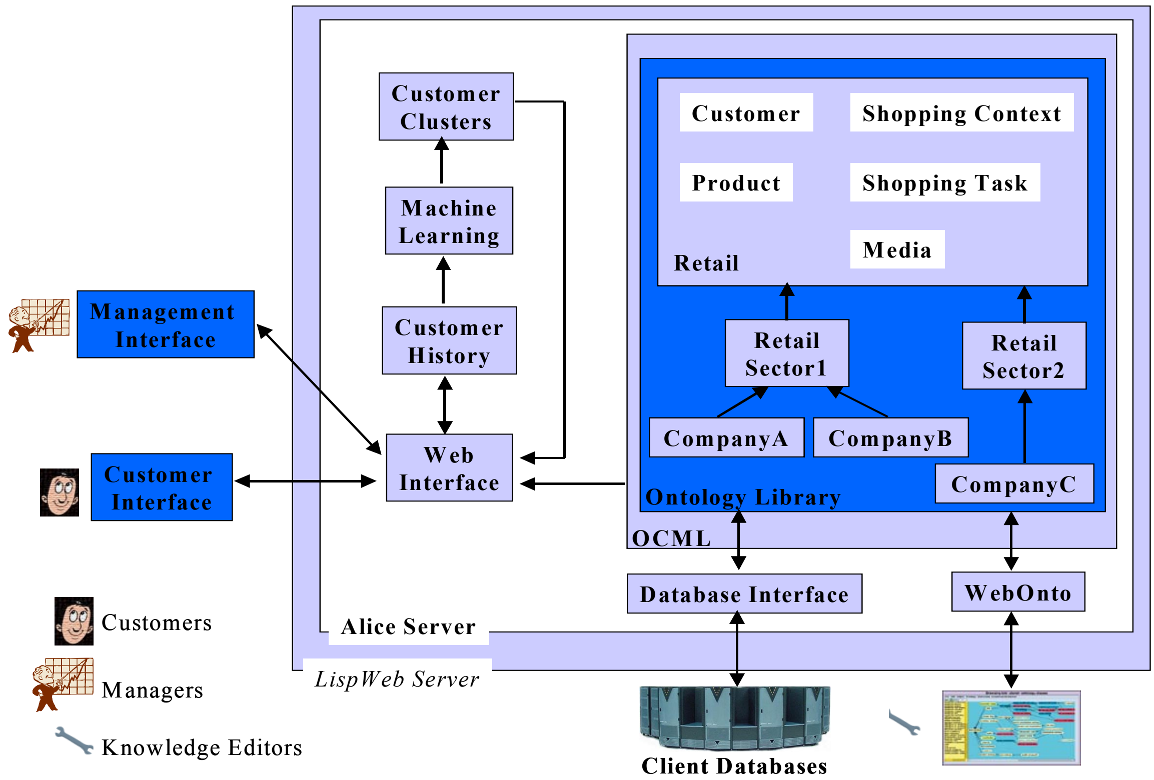
WebOnto [12] enables knowledge engineers to collaboratively browse and edit the knowledge models using a standard web browser. It has been used publicly for nearly two years and was formally evaluated [13] and found to be the most easy-to-use web based ontology editing environment.

The web interface module serves two roles. Firstly, the module uses the ontologies, the customer history and the customer clusters to generate web pages for customers and managers. The customer pages will typically contain a set of relevant products. The manager pages will facilitate the editing of specific parts of the company's knowledge model and provide visualizations of the customer clusters. Secondly, all the user interface actions are collected and used to create a customer history consisting of a time-stamped sequence of different types of events. The event types will include browsing events and purchase events. The machine learning module will create clusters of products and customers according to the customer history.

6 CONCLUSION

An architecture for developing a knowledge-based e-commerce personalised web site has been described. Ontologies are used to represent the knowledge from enterprise functions that are relevant to a commercial transaction, specifically knowledge about: customers, products, tasks and contexts. Customers' shopping histories are transformed into knowledge using unsupervised clustering and machine learning tools. Preliminary results using synthetic data have shown that our architecture can give highly relevant choices to the e-shopper. We are confident that the combination of knowledge used within Alice will enable us to create effective customer oriented online shopping experiences.

Figure 4- Alice architecture



7 REFERENCES

[1] Lau, T. and E. Horvitz. Patterns of search: analysing and modelling web query refinement. *UM99 USER MODELING*. Proceedings of the seventh international conference. Banff, Canada, June 20-24, 1999. SpringerWienNewYork (1999)

[2] Brynjolfsson E. and M. D. Smith. *The Great Equalizer? Consumer Choice Behavior at Internet Shopbots*. MIT Sloan School of Management. Cambridge,Massachusetts (2000*)*. Available at http://ecommerce.mit.edu/forum/

[3] Tewari, G. P. Maes. *Design and Implementation of an Agent-based intermediary infrastructure for electronic markets*. MIT Media Lab (2000).

[4] Gruber, T. R. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2) (1993).

[5] Pennock, D. M.; E. Horvitz; C. L. Giles. Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering. Proceedings of the AAAI-2000 Seventeenth National Conference on Artificial Intelligence, Austin, Texas, USA (2000).

[6] Cotter, P. and B. Smyth. PTV: Intelligent Personalised TV Guides. Smart Media Institute, Dep. of Computer Science, University College Dublin (2001). Available at *http://www.ptv.ie*

[7] Lawrence, R. D.; G. S. Almasi; V. Kotlyar; M. S. Viveros; S. S. Duri. Personalization of Supermarket product recommendations. *Data Mining and Knowledge Discovery,* 5, 11-32 (2001).

[8] Rich, E. Stereotypes and User Modeling. In *User's model in Dialog Systems* Kobsa, A.; W. Wahlster (eds.). Spring-Verlag (1985).

[9] Cheeseman, P. and J. Stutz. Bayesian Classification (AutoClass): Theory and Results. In Fayyad, M. et al. [eds], *Advances in knowledge discovery and data mining*. MIT Press (1996).

[10] Riva, A. and Ramoni, M. LispWeb: a Specialized HTTP Server for Distributed AI Applications, *Proceedings of the Fifth International World Wide Web Conference*- May 6-10, Paris, France (1996).

[11] Motta, E. *Reusable Components for Knowledge Modelling: case studies in parametric design problem solving*. IOS Press, Netherlands (1999).

[12] Domingue, J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In B. Gaines and M. Musen (editors), *Proceedings of the*

11th Knowledge Acquisition for Knowledge-Based Systems Workshop, April 18th-23th, Banff, Canada (1998). Available at http://kmi.open.ac.uk/people/domingue/banff98-paper/domingue.html.

[13] Duineveld, A;, Stoter, R., Weiden, M., Kenepa, B., and Benjamins, V. R. Wondertools? A comparative study of ontological engineering tools. *International Journal of Human computer Studies*, 52, 1111-1133 (2000).

---

[i] http://faculty.ncwc.edu/toconnor/417/417lect06.htm

[ii] www.manufacturing.net/mmh/index.asp?layout=articleWebzine&stt=000&articleid=CA82289&pubdate=5/15/01

[iii] www-ksl.stanford.edu/kst/what-is-an-ontology.html

[iv] www.microsoft.com/DirectAccess/products/sscommerce)

A Conceptual Value Modeling Approach for e-Business Development

Jaap Gordijn
Vrije Universiteit - Vuture.net (Centre for e-business research)
Cisco Systems - Internet Business Solutions Group
De Boelelaan 1081a
1081 HV Amsterdam, The Netherlands
gordijn@cs.vu.nl

Hans Akkermans
Vrije Universiteit - Vuture.net (Centre for e-business research)
AKMC Knowledge Management
De Boelelaan 1081a
1081 HV Amsterdam, The Netherlands
hansakkermans@compuserve.com

Abstract

e-Business models show why an e-business case might work from an economic perspective. They are often represented using informal textual representations. This inhibits a clear understanding of an e-business model by all stakeholders involved. We propose a more formal, conceptual approach for representing the business model, which we call a *value* model, to enhance the common understanding amongst stakeholders, but also to allow for an asssesment of an e-business model. We illustrate our modeling approach by a state of the art project in the free Internet arena.

INTRODUCTION

Successful e-business information systems often are characterized by *innovative* ways of doing business at their time of introduction. Such a way of doing business is called the *e-business model*, and because of its newness, an important design topic in e-business projects.

Currently the *e-business model* concept is overused and has many interpretations. In most cases, an e-business model is represented by a rough, textual, outline of the service to be delivered. This rather vague notion of a *business model* results in time-consuming- and mis-communications between stakeholders involved, not in the least between business oriented people, and IT people.

We propose to use a *conceptual modeling* approach with clearly defined modeling constructs, called e^3-*value* , to design and to reach common understanding about an e-business model. Our approach offers a number of interrelated core concepts, also called an ontology, which are be used to build a semi-formal conceptual e-business model. Our approach is unique because we focus on the concept of *economic value* as a central modeling construct. Consequently, we define an e-business model as a *conceptual, economic value* model, which shows objects of *value*, which are created, exchanged and consumed in a multi-actor network.

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

A conceptual value modeling approach has the advantage that it it facilitates, by clearly articulating the value proposition, in reaching a better shared understanding and agreement between actors on a service to be offered, Secondly, our technique allows for evaluation of an e-business value model. Evaluation assesses whether the e-business model is profitable, or increases economic utility, for all stakeholders involved.

We illustrate the use of e^3-*value* with one of the e-business projects where we successfully applied our approach. The project at hand is about the provisioning of a value add news service. With respect to such a service, a regular newspaper called the *Amsterdam Times* wants to offer to all its subscribers on the regular newspaper the service to read articles online using the Internet. However, *Amsterdam Times* does not want to make any additional costs for offering this online service. Therefore, we finance the execution of the business idea by telephone connection revenues, which are paid by the reader who has to set up a telephone connection for Internet connectivity.

In this article, we demonstrate how we used our e^3-*value* technique to explore this e-business value model, first by showing how we represent an e-business model, and by discussing what stakeholders can conclude from such a model, and second by illustrating evaluation as a confidence builder in the e-business idea at hand.

REPRESENTING AN E-BUSINESS VALUE MODEL

To represent an e-business value model, we use a lightweight ontology consisting of interrelated core concepts, and we utilize a well known lightweight scenario technique, called Use Case Maps [1]. A lightweight ontology contains a limited set of concepts and relations [7]. This allows us to communicate our ontology easily to intended users such as business consultants, and CxO's. Moreover, the agility of e-business projects (the need to define, explore, and execute a business idea fast [6]) asks for a lightweight approach. For the same reasons, we use a light weight scenario technique as well.

Below, we discuss the ontological concepts and the UCM scenario concepts briefly (see also [5], [1]). Throughout the text, we refer to an e-business value model example, presented in Fig. 1.

The e^3-*value* ontology

Actor. An actor is perceived by its environment as an independent economic (and often also legal) entity. By doing *value activities* (see below) an actor makes profit or increases its utility. In a sound, viable, business value model *every* actor should be capable of making profit.

Example. Actors are represented as rectangles. For instance, *Last Mile* is a telecommunication company operating the local loop (the last mile from a telephone switch to an end-user).

Value Object. Actors exchange value objects. A value object is a service, product or even a consumer experience. The important point here is that a value object is *of value* for one or more actors.

Example. Value objects are shows as text aside arrows (the value exchanges, see below). Most of the objects speak for themselves, but *Termination* as a value object needs further explanation. In the vocabulary of the telecommunication industry, telephone calls have to be terminated. This means, that if someone tries to set up a telephone connection, another actor must pick up the phone, that is *terminate* the connection. If someone is willing to cause termination of a large quantity of telephone calls, telecommunication operators are willing to pay such an actor for that. *Amsterdam Times* causes these terminations by offering its existing, large subscriber base (on a regualar newspaper) an *article online* service, which requires a telephone and internet connection.

Value Port. An actor uses a value port to show to its environment that it wants to provide or request value objects. The concept of port is important, because it enables to abstract away from the internal business processes, and to focus only on how external actors and other components of the e-business value model can be 'plugged in'.

Example. Ports are shows as small black circles. For instance, *Amsterdam Times* has an outgoing port for *articles online*.

Value Interface. Actors have one or more value interfaces. A value interface groups individual value ports offering or requesting value objects. It shows the value object(s) an actor is willing to exchange *in return for* other value object(s). Such willingness is expressed by a decision function on the value interfaces, which shows on what conditions an actor wants to exchange a value object for another value object. The exchange of value objects is atomic at the level of the value interface. Either *all* exchanges occur as specified by the value interface or *none* at all. The value interface says nothing about the time ordering of objects to be exchanged on its ports. It simply states which value object is available, in return for some another value object.

Example. A value interface is shown by a rounded box, connected to an actor. A *reader* has a value interface, which says that s/he wants to give its environment a fee for a telephone connection, and a *termination*, but wants an article online and a telephone connection *in return* for that.

Value Exchange. A value exchange is used to connect two value ports with each other. A value exchange represents one or more *potential* trades of value objects between value ports. As such, it is a prototype for actual trades between actors. The Enterprise Ontology [9] would call a value exchange a potential sale. It shows which actors are willing to exchange value objects with each other.

Example. A value exchange is shown by an arrow. Ports of *Amsterdam Times* and a *Reader* are connected by a value exchange, to exchange *articles online* and *terminations*.

Value Offering. A value offering is a set of value exchanges. It shows which value objects are exchanged via value exchanges *in return for* other value objects. If a value offering occurs, *all* value exchanges part of it should occur, or *none* at all. A value offering should obey the semantics of the connected value interfaces: that is values are exchanged via a value interface on *all* its ports, or *none* at all.

Example. The four value exchanges between *reader*, *Amsterdam Times*, and *Last Mile* are all part of *one* offering, because the value interface of the *reader* prescribes that either *all* these exchanges should occur, or *none* at all.

Market segment. In the marketing literature [8], a market segment is defined as a concept that breaks a market (consisting of actors) into segments that share common properties. Accordingly, our concept *market segment* shows a set of actors that share for value interfaces an equal decision function. Note that actors who are in a segment may also have in-similar value interfaces, because it is the actor-value interface combination, which build up a market segment.

Example. A market segment is shown by stacking actors graphically. A *reader* is an example of a market segment.

Composite actor. An actor is perceived by its environment as an independent economic (and often also legal) entity. However, for providing a particular service, a number of actors may decide to work together, and to offer objects of value jointly to their environment. Such actors decide on one or more common value interfaces to their environment. We call such a group of actors a composite actor.

Note that both composite actors and market segments, internally consist of multiple actors. A composite actor has a *common/shared* value interface for its internal actors, while,

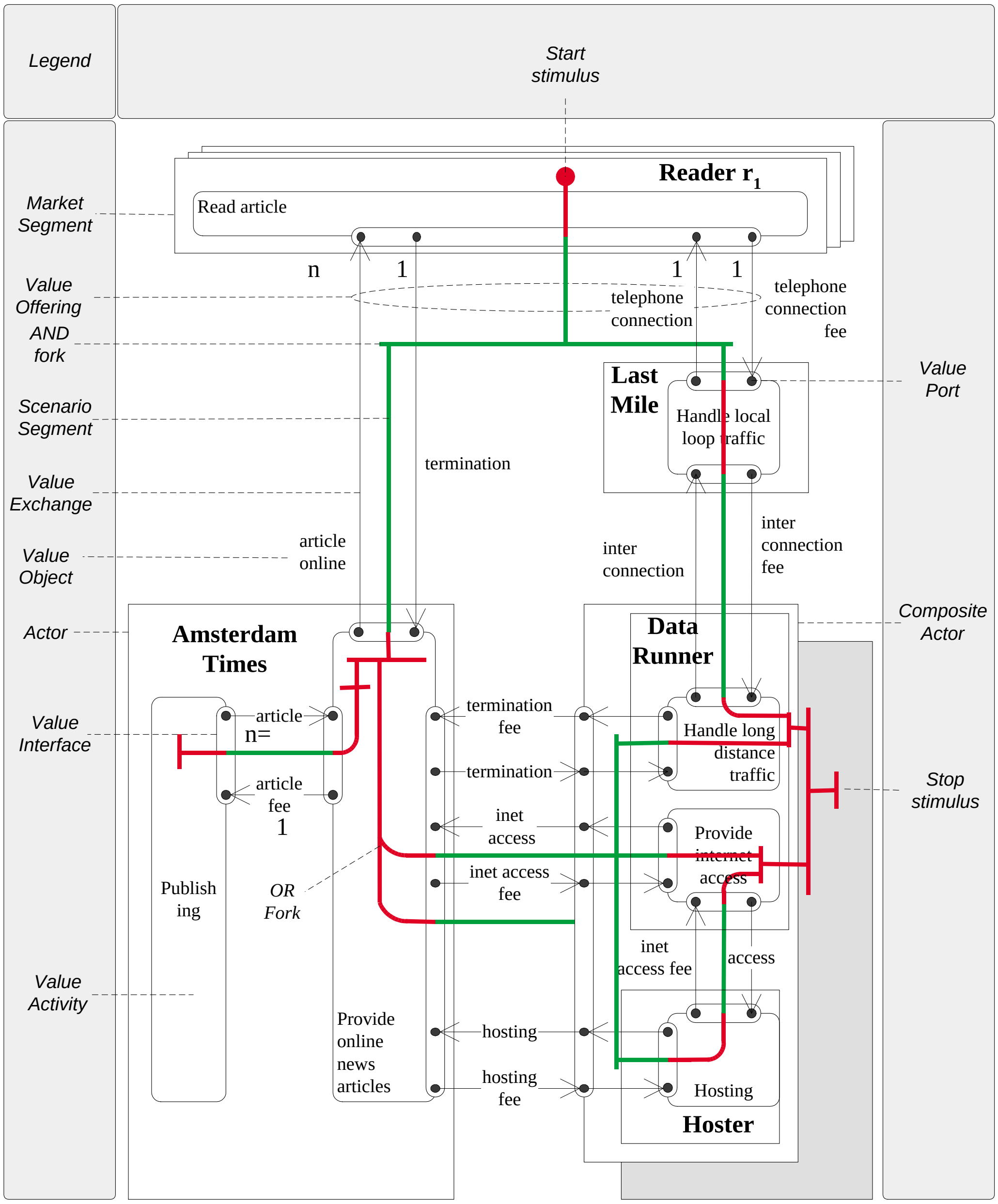


Figure 1: An e-business value model for the newspaper case. The *reader* pays a local operator, *Last Mile*, a fee for a telephone connection. *Last Mile* uses a long distance carrier, *Data Runner* to set up the connection if the callee is located beyond the geographical area *Last Mile* serves, and pays *Data Runner* for this (the interconnection fee). Because *Amsterdam Times* offers an article online service to its existing subscriber base, a large number of telephone calls are expected. Therefore, *Data Runner* is willing to pay *Amsterdam Times* a termination fee for the generated traffic. Furthermore, *Amsterdam Times* outsources the internet service provisioning (hosting and internet access) aspects of offering an article online and pays a fee for this. *Last Mile* and *Hoster* form a composite actor (a partnership) to offer hosting, access and call termination as one service to *Amsterdam Times*. There is a second composite actor (grey shaded) making the same offering also. *Amsterdam Times* can choose on a per scenario execution basis, which composite actor to use for service provisioning.

in contrast, a market segment is seen by its environment as a number of fully *independent* actors, for which we assume for a particular set of value interfaces an equal decision function.

Example. The topmost partnership consists of *Data Runner*, a telecommunication company and *Hoster*, an internet service provider. Both these companies decide to offer hosting and internet access jointly as a bundle, under certain special conditions. A special condition can be the price, which might be cheaper for *Amsterdam Times* than an alternative, such as obtaining the objects of value from other actors separately. In this specific case, *Data Runner* and *Hoster* can offer services jointly cheaper, because they co-locate technical equipment such as a telephone switch, internet access servers, and web servers at one physical site, thus saving costly wide area connections to interconnect all these components.

Figure 1 shows two of such partnerships, who offer comparable services. Note that these two partnerships are not a market segment, because their decision function may differ. For instance, the first partnership may offer the same services (access, hosting) for lower prices than the second partnership. The second partnership is shown as a gray box, to prevent unnecessary cluttering of the diagram.

Value Activity. A value activity is *performed by* an actor and increases profit or utility *for* such an actor. The value activity is included in the ontology to discuss the *assignment* of value activities to actors. As such, we are interested in the collection of activities which can be assigned as a whole to actors, and as a consequence, such an activity should be profitable or increase utility to be interesting to perform. Value activities can be decomposed into smaller value activities, but these still should be profitable or increase utility for the performing actor. This gives a decomposition stop rule, which is by the way clearly different from business process or work flow decomposition [3].

Example. Provide Internet access is an example of a profitable value activity, while *Read Article* is a utility increasing activity for the *reader*.

Use Case Maps

Scenario path. A scenario path consists of one or more segments, related by connection elements and start- and stop stimuli. It represents via *which* value interfaces objects of value must be exchanged, as a result of a start stimulus, *or* as result of exchanges via *other* value interfaces. Thus a scenario path shows causal relations between value interfaces.

Stimulus. A scenario path starts with a **start stimulus**. A start stimulus represents an event, possibly caused by an actor. If an actor causes an event, the start stimulus is drawn within the box representing the actor. The last segment(s) of a scenario path is connected to a **stop stimulus**. A stop stimulus indicates that the scenario path ends.

Example. A start stimulus is the desire to read an article online by a *reader*.

Segment. A scenario path has one or more segments. Segments are used to relate value interfaces with each other, possibly via connection elements, to show that an exchange on one value interface causes an exchange on another value interface. Using connection elements, sophisticated causal relations can be represented.

Connection. Connections are used to relate individual segments. An **AND fork** splits a scenario path into two or more sub path, while the **AND join** collapses sub path into one path. An **OR fork** models a continuation of the scenario path into one direction, to be chosen from a number of alternatives. The **OR join** merges two or path into on path. Finally, the **direct** connection interconnects two individual segments. Note that a scenario path must obey the semantics of the value interface: either objects are exchanged on *all* its ports or *none* at all.

Example. The AND-fork between the *reader*, *Last Mile*, and *Amsterdam Times* is important: it ensures, that if values are exchanged via the *reader*'s value interface, values must be exchanged via the value interfaces and of the *Amsterdam Times* and *Last Mile* respectively.

Responsibility point. Scenario segments may hit value interfaces. Such a hit is called a responsibility point. By following a scenario path and by finding the responsibility points along the path, we construct a profit/utility sheet (see Sec.). By valuing the objects in this sheet, and by making assumptions on the number of executions per time frame of the scenario path, we obtain a basic idea about the profitability or utility increase for each actor.

DISCUSSING AN E-BUSINESS VALUE MODEL

While the previous section discussed what an e-business value model looks like in terms of elementary concepts and scenarios, some other characteristics of an e-business value model can be seen from the pictures we draw.

Causality of revenue streams. The most obvious observation, which can made by looking at a conceptual e-business value model, is the causality of revenue streams, in reaction on a stimulus. Simply by following the scenario path, it can be seen which exchanges of values via value interfaces result in other value exchanges.

Example. In the e-business value model at hand the initial money flow is between the *reader*, and *Last Mile*. All other money flows are generated from the money earned by *Last Mile*.

Bundling of value ports- and objects. For several reasons, actors may decide to offer or request objects of value only *in* *combination*. For instance, objects are only of value for actor, if they are obtained in combination; or an actor may believe he earns more money if objects are sold in combination rather then separately. This is shown by grouping ports offering and requesting such objects into one value interface

Example. Article online, *telephone connection*, *termination*, and *telephone connection fee* are *bundled* into one value interface from a *reader*'s perspective. This models that a *reader* only values an *article online*, and a *telephone connection* in combination. Because we bundle these value objects, and because a *reader* needs to offer a compensation for these objects, *termination*, and *telephone connection fee* must also be part of the same value interface.

Customer ownership. In the situation that a customer (e.g. a *reader*) buys a product from only one seller regularly, such a seller start ti build up a relation with that customer, and owns the customer. Owning a customer is important, because it allows an actor for instance to build a profile of a customer, which can used to offer the customer new products or services in the future. Whether an actor *solely* ownes a customer can be seen by examining value offerings. If an offering is between two actors (a seller and a buyer), the seller 'owns' the buyer. However, if an offering contains more than one seller, customer owbership will be partitioned.

Example. Originally, the reader was a full customer of *Amsterdam Times*, because the reader is part of *Amsterdam Times*' regular subscriber database. However, for the online service, the reader now has to exchange values with *Amsterdam Times, and Last Mile*. The latter is even the party who receives the only payment for the delivered service This can be seen as a shift in customer ownership from *Amsterdam Times* to *Last Mile*, which might be an unwanted situation from *Amsterdam Times*' point of view.

Power: Price setting. An important aspect of business power is the ability to determine the price for products or services to be delivered. By examining value interfaces and value exchanges, at least actors playing a role in pricing can be seen.

Example. The *reader* pays for the *article online* via a telephone connection fee to *Last Mile*. Unfortunately, no one, except *Last Mile* and perhaps a market regulation authority, can influence the pricing. Consequently, the success of the business value model depends largely on *Last Mile*. Also, the *Amsterdam Times* is unable to set a price itself for its *article online* offered to a *reader*.

Power: Selection. If a buyer is able to make a selection of a set of potential sellers, his selection power increases, due to competition. This can be seen if a buyer is connected to multiple sellers by multiple offerings.

Example. The *reader must* use *Last Mile* for local loop access. At the time the project was carried out, there was only one actor available controlling the local loop to subscribers. This can be concluded from figure 1, because only one actor for local loop access is drawn, and consequently only one offering between *Last Mile* and *reader* exists. Again, this makes the business value model very critical to the behavior of *Last Mile*.

Example. The business idea at hand has a special trick to enlarge the selection power of *Amsterdam Times* with respect to the telecommunication and internet service providers. The *Amsterdam Times* can choose from two different composite actors to actually deliver the article online (from an access and hosting perspective), and this selection can be done on a per scenario execution base. The reason for this is that the *Amsterdam Times* does not want to be dependent on one provider for access and hosting. By distributing the amount of traffic over these two (composite) providers, the *Amsterdam Times* controls the distribution of revenues for the two composite actors, and motivates both partnerships to deliver a high level quality of service. This is graphically shown using an OR-fork in the scenario path, which models the service selection by *Amsterdam Times*.

Duplication of assets against nearly zero marginal cost. An important property of digital assets, such as news articles, is the ability to reproduce the asset against nearly zero marginal cost [2]. This is graphically shown by placing cardinalities near value exchanges connected to ports.

Example. The *Amsterdam Times*, and especially the value activity *Provide Online News Articles*, buys an article from another value activity *Publishing* (the activity necessary to produce articles for a regular newspaper anyway) if it is requested by the *reader*. However, it only buys this article once, so if multiple readers ask for the same article, the *Provide online news articles* only pays once the *Publishing* activity. Fig. 1 shows this by the cardinalities on the value exchanges near the value interface of value activity *Publishing*: for *n* equal articles, only *one* fee has to be paid.

EVALUATING AN E-BUSINESS VALUE MODEL

We use a conceptual e-business value model, and scenario paths not only to create a common understanding of the e-business idea amongst stakeholders, but also to evaluate the economical feasibility of an e-business idea. We call this the *evaluation* of the business idea, because it is based on an assessment of the value of objects by actors. Feasibility is in reach if *all* actors involved are able to make profit or increase their economic utility with an e-business idea. Again, our technique for determining feasibility is a light weight approach, and focusses on building *confidence* that an e-business idea is of interest for the actors involved, rather than offering a precise calculation of all profits. Our evaluation approach consists of the following steps:

1. Creation of profit/utility sheets for all actors, and/or value activities;
2. Determination of a valuation scheme for each actor;
3. Evaluation of *what-if* scenarios.

Creation of a profit/utility sheet. Profit/utility sheets are created on an actor or value activity level. The actor level is of interest to create confidence in an e-business idea, while profits sheets on the value activity level are useful to evaluate whether activities are really profitable. Due to lack of space, we concentrate on the actor profit sheets only.

Table 1 shows a reduced profit/utility sheet for the scenario *read article online*. We create this sheet by following the scenario path, starting at the start-stimulus, and each time the path crosses a value interface of an actor, we update the sheet with value objects flowing in- and out the actor. In Table 1, we have two composite actors. For brevity, we only show for composite actor 1 the profit sheets of its actors (*Data Runner* and *Hoster*).

We *reduce* the profit sheet by removing for each actor all value objects, which are (1) not money streams, *and* (2) which are entering the actor and leaving the actor (possibly in an enriched form, as the result of performing an activity) on the same scenario path. For example, we remove *telephone connection* and *interconnection* from the actor *Last Mile*, because the *telephone connection* is an enriched *interconnection*. *Last Mile* enriches the *interconnection* by exploiting a district telephone switch and a list mile of copper or fibre optics.

In some cases, it is not possible to remove all value objects, which are not money streams. This is especially the case for end-customers such as a *reader*. In such a situation, delivering or receiving such objects changes the economic utility for that actor. Due to lack of space, we do not discuss the valuation of such objects, but refer to [4], where we propose an approach for valuation of these objects. Therefore, we ommit the emphreader from the profit/utility sheets.

Determination of a valuation scheme. After reduction of the profti/utility sheet, the remaining value objects in the profit sheet have to be assigned a value by actors, expressed in monetary units (e.g. dollars). Table 1 shows such a valuation scheme, which we explain briefly:

Telephone connection fee. The telephone connection fee per scenario occurrence is based on a start tariff and a connection-time dependent tariff. To calculate the total monthly fees, the telephone connection fee is multiplied with the actual number of scenario occurrences.

Interconnection fee. The interconnection fee per scenario occurrence is based on a fraction (the interconnection factor) of the telephone connection fee, and on the physical distance *Data Runner* bridges.

Termination fee. The termination fee *Amsterdam Times* receives, is calculated analogously to the interconnection fee, only now we use a revenue sharing factor rather than an interconnection factor. Typically, the revenue sharing factor is smaller than the interconnection factor. Note that by valuing this way, we are capable of analysing the effects of a decreasing interconnection factor (e.g. influenced by a market regulator), while the revenue sharing factor keeps the same. This models a situation where *Data Runner* takes the risk of a decreasing interconnection factor solely.

Internet access fee for Amsterdam Times. *Data Runner* charges *Amsterdam Times* an internet access fee in return for giving *readers* access. This fee is based on an access tariff per second. We want to account for the situation that internet access equipment is a very scarce resource; *Data Runner* wants to have the opportunity to assign unused access ports to others. Therefore, *Amsterdam Times* is asked to forecast the number of scenario occurrences on a monthly basis, including the average connection duration. *Data Runner* then allocates access ports based on this forecast, and can allocate remaining ports to others. To motivate *Amsterdam Times* for good forecasting, the following valuation is used: If the actual scenario occurrences drop below 75 % of the forecasted occurrences, we use 75 % of the forecasted occurrences for the calculation of the monthly internet access fee. Otherwise, we use the actual, realized number of scenario occurrences.

Internet access fee for Hoster. To make hosted internet applications available for the end-user, *Hoster* needs also internet access. The internet access fee to be paid by *Hoster* is entirely based on the forecasted number of scenario occurences. Based on this, we calculate a estimate of the required bandwidth, and calculate the price for this.

Internet hosting fee. *Hoster* uses a forecast of *Amsterdam Times* of the number of concurrent page views, which in turn is based on an average number of page views per forecasted scenario occurrence. This results in a fixed fee per month for hosting.

Evaluation of *what-if* scenarios. Using the valuation in Table 1, we e-valuate several scenarios, which model expected changes in the future regarding valuation. As an example, Table 2 shows a number of scenarios.

Null scenario. The *null* scenario is the situation at this moment. Observe that *all* actors make profit to cover additional costs.

Bad forecasting. What happens if the *Amsterdam Times* is not a good forecaster of scenario occurrences. It can be seen

Table 1: Profit/utility sheets and valuation schemes for the scenario *Read article online*.

Scenario	Read article online	
Actor	*Value Object In*	*Value Object Out*
Last Mile	*tel. connect. fees* = (*tel. start tariff* + (*tel. connect. tariff* ∗ *duration*)) ∗ *actual occ.*	*interconnect. fees*$_{composite1}$ = *tel. connect. fee* ∗ *distance factor*$_{composite1}$ ∗ *interconn. factor* ∗ *actual occ.* ∗ *p*
Amsterdam Times	*termination fees*$_{composite1}$ = *tel. connect. fee* ∗ *revenue sharing factor*$_{composite1}$ ∗ *distance factor* ∗ *actual occ.* ∗ *p*	*inet access fees*$_{composite1}$ = *see Data Runner* *hosting fee*$_{composite1}$ = *see Hoster*
Composite 1		
Data Runner	*interconnect. fees* = *see Last Mile* *inet access fee*$_{AmsterdamTimes}$ = *inet. connect. tariff* ∗ *duration* ∗ *actual-forecast occ.* ∗ *p* *inet access fee*$_{Hoster}$ = *occurences*$_{forecast}$ ∗ *p* → *needed bandwidth* → *fixed fee/month*	*termination fees* = *see Amsterdam Times*
Hoster	*hosting fee* = *concurrent occ.*$_{forecast}$ ∗ *p* → *concurrent pageviews* → *fixed fee/month*	*inet access fee* = *see Data Runner*
Composite 2	...	...

Table 2: Different valuation scenarios. The null-scenario uses the valuation parameters as we see them most likely for now. A second scenario assumes that *Amsterdam Times* forecasts inaccurately. A decrease in the interconnection is expected to occur, especially of competition between telecommunication actors increases (see the third scenario). The fourth scenario supposes a drop in the revenue sharing factor between *Data Runner* and *Amsterdam Times*.

	Profit			
Scenarios	*Amsterdam Times*	*Last Mile*	*Data Runner*	*Hoster*
Null-scenario	164,400	102,000	113,800	8,000
Forecast(1,500,000) >> *Actual*(150,000)	-28,560	10,200	26,680	8,000
Decrease in interconn. factor (1.0 to 0.4)	164,400	346,800	-8,600	8,000
Decrease in revenue sharing factor (0.5 to 0.1)	-19,200	102,000	205,600	8,000

that *Amsterdam Times* will not make profit. For *Last Mile* and *Data Runner* there is still profit to cover costs. *Hoster* is insensitive to bad forecasts, because it is not dependent on the actual scenario occurrences.

Interconnection fee decreases. A decrease in the interconnection factor is reasonable to expect after a couple of month, because this factor is now high to stimulate competition between telecommunication operators. As soon as this competition works, the factor will decrease. *Amsterdam Times* does not feel such a decrease, but *Data Runner* will.

Revenue sharing fee decreases. Data Runner may decide to decrease its revenue sharing factor. As can be seen, this will harm *Amsterdam Times*.

In conclusion, by valueing the objects for each actor, and by making assumptions about the number of (forecasted) scenario accurrences, we can do a sensitivity analysis for the business idea hand. This sensitivity analysis is in most cases of more interest, than the valuation itself.

CONCLUSIONS

In this paper, we proposed a conceptual modeling approach for the development and representation of e-business models. The notion of *economic value*, and how objects are created, exchanged and consumed in a multi-actor network is a central theme in our ontology for e-business models.

We showed how a non-trivial business idea can be represented using our e^3*-value* technique. Besides the causality of revenue streams, our e^3*-value* models also show other aspects, such as bundling, customer ownership, power in pricing and actor selection, and partnership.

Using well-known scenario techniques, we constructed profit sheets, which in turn were used to get a first impression of the profitability of the business idea for each actor. These profit sheets should not be seen as absolute profit predictors, but more as calculation schemes with which a sensitivity analysis can be done. Using *what if* scenarios, which focus on expected changes in the future, even more confidence can be built.

For the project at hand, our approach was especially useful to articulate the business idea precisely, thereby creating a common understanding amongst stakeholders. The e- business model appeared to be too complicated, to communicate in natural language. Moreover, the valuation scheme and the what-if scenarios were of use during contract negotiations, especially to enhance transparency and to evaluate future developments and risks for all actors involved quickly.

Acknowledgement. This work has been partly sponsored by the Stichting voor Technische Wetenschappen (STW), project nr VWI.4949. Also, we thank *De PersCombinatie* for their approval to make this paper public.

REFERENCES

1. R.J.A. Buhr. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24(12):1131–1155, 1998.

2. Soon-Yong Choi, Dale O Stahl, and Andrew B. Whinston. *The economics of doing business in the electronic marketplace*. Macmillan Technical Publishing, Indianapolis, 1997.

3. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Business modelling is not process modelling. In Stephen W. Liddle and Heinrich C. Mayr, editors, *Conceptual Modeling for E-Business and the Web, Salt Lake City, Utah, 9-12 October*, volume LNCS 1921, pages 40–51. Springer Verlag, 2000. Also available from http://www.cs.vu.nl/˜gordijn.

4. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Selling bits: A matter of creating consumer value. In Kurt Bauknecht, Sanjay Kumar Madria, and Gunther Pernul, editors, *First International Conference on Electronic Commerce and Web Technologies EC-Web 2000, London, UK, 4-6 September*, volume LNCS 1875, pages 48–62, Berlin, 2000. Springer Verlag. Also available from http://www.cs.vu.nl/˜gordijn.

5. J. Gordijn, J.M. Akkermans, and J.C. van Vliet. What's in an electronic business model. In R. Dieng and O. Corb, editors, *Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th International Conference, EKAW 2000, Juan-les-Prins, France, October 2000*, volume LNAI 1937, pages 257–273. Springer Verlag, 2000. Also available from http://www.cs.vu.nl/˜gordijn.

6. Amir Hartman, John Sifonis, and John Kador. *Net Ready - Strategies for Success in the E-conomy*. McGraw-Hill, New York, 2000.

7. R. Jasper and M. Uschold. A framework for understanding and classifying ontology applications. In B. Gaines, R. Cremer, and M. Musen, editors, *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW'99 (16-21 October 1999, Banff, Alberta, Canada)*, volume I, pages 4–9–1 — 4–9–20, Calgary, 1999. University of Calgary, SRDG Publications.

8. P. Kotler. *Marketing management: analysis, planning, implementation and control*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

9. M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The knowledge engineering review*, 13, 1998.