

Association for Information Systems AIS Electronic Library (AISeL)

ECIS 2007 Proceedings

European Conference on Information Systems
(ECIS)

2007

Knowledge-Driven Selection of Market Mechanisms in E-Procurement

Dirk Neumann

Institute of Information Systems and Management, Universität Karlsruhe, neumann@iism.uni-karlsruhe.de

C. Block

block@iism.uni-karlsruhe.de

Christof Weinhardt

Information Management and Systems, University of Karlsruhe, weinhardt@iism.uni-karlsruhe.de

Y. Karabulut

yuecel.karabulut@sap.com

Follow this and additional works at: <http://aisel.aisnet.org/ecis2007>

Recommended Citation

Neumann, Dirk; Block, C.; Weinhardt, Christof; and Karabulut, Y., "Knowledge-Driven Selection of Market Mechanisms in E-Procurement" (2007). *ECIS 2007 Proceedings*. 92.

<http://aisel.aisnet.org/ecis2007/92>

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

KNOWLEDGE-DRIVEN SELECTION OF MARKET MECHANISMS IN E-PROCUREMENT

Neumann, Dirk, Universität Karlsruhe (TH), Englerstr. 14; 76131 Karlsruhe,
neumann@iism.uni-karlsruhe.de

Block, Carsten, Universität Karlsruhe (TH), Englerstr. 14; 76131 Karlsruhe,
block@iism.uni-karlsruhe.de

Weinhardt, Christof, Universität Karlsruhe (TH), Englerstr. 14; 76131 Karlsruhe,
weinhardt@iism.uni-karlsruhe.de

Karabulut, Yücel, SAP Research Center, Palo Alto, CA 94304,
yuecel.karabulut@sap.com

Abstract

The variety of procurement mechanisms present in the today's e-procurement landscape ranging from electronic catalogue systems over e-negotiations to e-auctions, points at the fact that there exists no single best solution for all sourcing activities. Each mechanism rather has certain advantages and disadvantages. The discipline of economics has traditionally been devoted to the study of markets and market mechanisms. The fundamental lesson learned from economics is that even small changes in the exchange mechanisms can have considerable impact on the outcome. Stated differently, if the market engineer intends to attain a certain mechanism outcome (e.g., efficiency, fairness, revenue maximization), he can define the mechanism in a way that it induces the right incentives for market participants to act as desired in order to achieve the envisioned outcome. Unfortunately, a comprehensive system that combines all relevant design aspects into one single knowledge-based decision support system is missing. The main contribution of this paper is to develop such a system that guides the design of procurement mechanisms by prescribing mechanism formats using an adapted case-based reasoning algorithm. The knowledge-based system, KMS, is implemented in a web application as a proof-of-concept.

Keywords: negotiation, electronic auction, e-auction, decision support, DSS, strategic sourcing, SRM, procurement, knowledge base, expert system.

1 INTRODUCTION

Since the uptake of the Internet, e-markets have become an important component in e-procurement by bringing together demand and supply. E-markets are meeting venues for component suppliers and purchasers, who use exchange mechanisms to electronically support the procurement process. Exchange mechanisms can be conceived as market institutions providing sets of rules, which determine the functioning of the market and the permissible actions such as bidding. Mechanisms vary from online catalogues, where requests and offers are publicly announced, to e-negotiations, where the participants bargain over the conditions of a trade using electronic message exchange and / or decision support platforms, to auctions, where one or two sides automate the process during which participants from the other side compete against each other (Kersten et al., 2006).

The variety of exchange mechanisms present in the today's e-procurement landscape suggests that there exists no single best solution for all sourcing activities; instead each mechanism has certain advantages and disadvantages as well. The discipline of economics has traditionally been devoted to the study of markets and market mechanisms. Especially, the study of auctions and auction design has produced several intriguing insights. Most prominently in this context are examples from the design and implementation of the UMTS auctions (Klemperer, 2001), which exhibit how easily design mistakes can be made. For example, while the British design of an ascending auction resulted in high revenues for the seller, the same auction design flopped in Switzerland.

As a lesson learned from economics, it can be stated that the design of mechanisms is extremely difficult as details matter. Details pertain to the procurement situation - the economic environment - and to the details of the market mechanism as well. Theory typically offers only a very limited comparison of mechanisms but experiments do (e.g. Katok & Roth, 2004; Strecker & Seifert, 2003). A comprehensive framework that explains when to use which mechanism does not exist. This leaves, on the one hand, market operators such as Supply-On or SAP SRM (Casaseca, 2005) alone with the question which mechanisms to offer in their e-markets. On the other hand, suppliers and procurers face a similar decision problem as it is not trivial to decide which of the offered alternatives is most suitable in their specific sourcing context.

This work aims at supporting these user groups by designing and implementing a decision support system for the selection of procurement mechanisms. In the proposed decision support system results from economic theory, laboratory experiments, numerical analysis and expert interviews are analyzed, structured and subsequently used for the generation of suitable mechanism recommendations.

The purpose of the knowledge-based mechanism design support system (KMS) is twofold:

- Support the analysis of procurement mechanisms and their impact on the market performance (e.g., revenue, efficiency, immediacy, fairness) and
- Provide recommendations for procurement mechanisms dependent on sourcing objectives, supply situation, product characteristics, market conditions and legal or other constraints.

With those two tasks accomplished it is possible to support (a) market operators in their decision making on which particular exchange mechanisms to offer and (b) suppliers and procurers in their e-market selection.

The remainder of the paper is structured as follows. In Section 2, the computer-aided market engineering workbench meet2trade is introduced, which automates all parts of the engineering process. It is shown how the proposed decision support system KMS fits into the workbench. Section 3 describes related work while section 4 and 5 show how knowledge is acquired in practice and how this knowledge is handled in the prototypical implementation of KMS. Section 6 concludes with a summary and an outlook on future work.

2 COMPUTER AIDED MARKET ENGINEERING

Designing e-markets in such a way that a specified objective is attained (e.g. rapid rollout and savings capture or widespread user adoption) is a demanding task involving several activities. Market engineering strives to provide a methodology for designing e-markets in a structured and reproducible way (Neumann, 2004; Weinhardt et al., 2003). Relying on a discursive approach the overall complex design process is decomposed into several phases, each of them being easier to deal with: At the outset of the market engineering process stands the strategic task of defining the segment in which the e-market is intended to operate. Subsequently, the exchange mechanisms that describe the flow of the transaction process are designed and implemented. In the last phase of the process the performance of the e-market is benchmarked against the objectives laid out in the first process phase in order to make sure that the original objectives are met.

It should be noted that market engineering is an inherently interdisciplinary problem comprising tasks from marketing, management, economics and computer science.

The market engineering process alleviates the interface problem of the many disciplines by defining the documents, which are result of the respective phases and the procedure how to develop those documents. The integrated computer-aided market engineering (CAME) workbench meet2trade strives now to automate these procedures beginning with the design of a market mechanism and complementary services and ending with the implementation and testing (Neumann et al., 2005; Weinhardt et al., 2006).

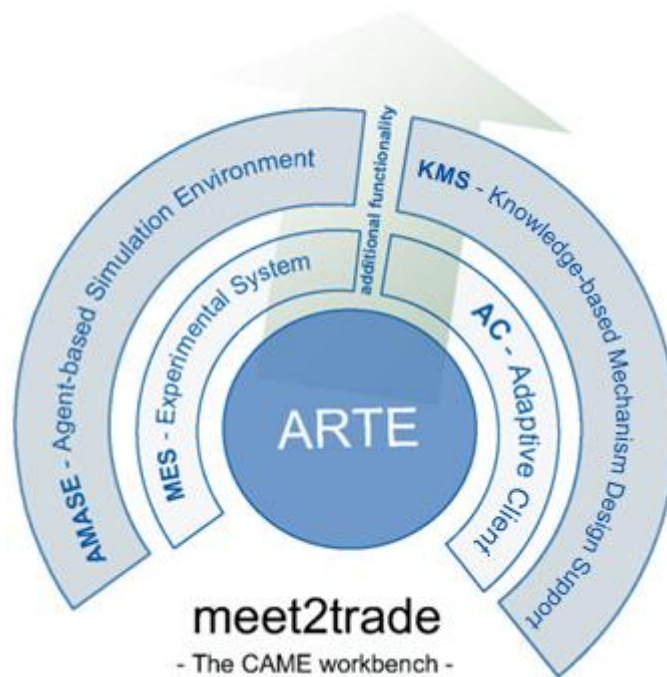


Figure 1: The CAME workbench meet2trade

The computer aided market engineering CAME workbench meet2trade comprises several components as shown in Figure 1.

- **ARTE - Auction run-time environment:** The design of market mechanisms is based on a parameterization approach - i.e. any exchange mechanism (mostly auctions) can be described by a set of parameters representing their specific rules. ARTE is responsible for creating market mechanisms defined by XML instances that contain the required sets of parameters. Hence ARTE is at the core of the computer-aided market engineering workbench. A configuration editor

facilitates the generation of XML instances and also provides a convenient mechanism to upload them into ARTE (Mäkiö & Weber, 2004).

- **AC - Adaptive client:** The main user interface of the CAME tool suite is the adaptive client (AC). Its graphical user interface is remotely configured by the ARTE core in order to present a suitable user interface for a specific mechanism as defined in the respective XML instance. The adaptability of the client is a key enabler that allows the dynamic rendering of different GUIs according to the needs of specific mechanisms.
- **AMASE - Agent-based market simulation environment:** AMASE is an agent-based simulation environment, which allows the automated testing of market mechanisms. Simple test scenarios can be produced on-the-fly, while more complex scenarios require some coding of the agent behaviour (Czernohous, 2005). AMASE renders predictions about how market mechanisms will perform using simulation techniques that allow valid predictions even about sophisticated market mechanisms.
- **MES - Market experiment shell:** In order to examine specific procurement mechanisms, an experimental system has been added to the meet2trade software suite. The main objective is to conduct experiments on the original system instead of replicating and running the mechanism in experimental software. This approach facilitates experimental studies since the market has to be modelled only once within meet2trade and avoids potential biases from the usage of different user interfaces. For experiments the standard AC client is running in *experimental mode*, which enables more detailed logging of user actions and allows tight control over permitted actions in different stages of the experiments (Kolitz & Weinhardt, 2006).

Currently missing in this set of tools is a decision support system, which gives prescriptions on what mechanism to use in which situation. This gap is filled by the KMS system. It is capable of storing economic design knowledge as well as empirically collected mechanism recommendation e.g. in the field of e-Procurement and providing it to meet2trade users in a consistent manner.

Before the KMS prototype is described in more detail, the main results from mechanism design (a sub-field of economics) and from e-procurement, are summarized to show what kind of knowledge KMS needs to cope with.

3 RELATED WORK

The theory of mechanism design is mainly concerned with the conceptual design of procurement mechanisms on the blackboard (Bichler, 2001). Mechanism design can be characterized as manual craftwork: guided by intuition and experience, a designer claims that a certain mechanism enfold a desirable effect and subsequently he tries to prove this. Alternatively, a designer can determine the "optimal" mechanism by formulating the mechanism design problem as mathematical optimization problem (Myerson, 1981). Since its rise as a discipline initiated by Hurwicz's seminal paper in 1973 (Hurwicz, 1973), mechanism design has produced a small canon of mechanisms, where each of these mechanisms attains a specific desideratum in a certain class of environments:

The most seminal mechanism is called after its inventors "Vickrey-Clarke-Groves" (VCG). The VCG mechanism is attractive for several reasons: It achieves an efficient allocation of resources while it still remains individually rational (i.e. participation does not yield lower utility than non-participation). Also the VCG mechanism does not require payments from the mechanism (Clarke, 1971; Groves, 1973; Vickrey, 1961). Thus the VCG is the only mechanism that achieves those three desiderata (Green & Laffont, 1977; Holmstrom, 1979).

Except VCG, almost all mechanisms crucially depend on common knowledge about private information of the bidders. Common knowledge among the bidders who actually participate in the mechanism is already a strong assumption. However, extending this common knowledge to the mechanism designer is arguably untenable. These rather strong assumptions currently prevent the mechanisms to be applied in practice. But also the VCG mechanism is plagued with severe drawbacks.

Some of them are associated with the computational complexity of the mechanism (Sandholm, 2002), the information it requires from the bidders or the inability to accommodate budget constraints, leaving the VCG mechanism as a (theoretical) benchmark rather than a practical auction.

In addition to this small excerpt from the “*possibility results*”, mechanism design theory has also developed several “*impossibility results*”. Impossibility results state in which settings no mechanism exists that satisfies some desiderata. For example the Myerson-Satterthwaite impossibility theorem states that it is impossible to find a mechanism that allocates goods of the same resource efficiently such that the budget is balanced and the individual rationality condition is satisfied as Bayesian-Nash equilibrium (Myerson & Satterthwaite, 1983). Impossibility theorems like this one are rather powerful, as they generally demonstrate the limitations of mechanisms.

Besides theoretical approaches towards mechanism design, a significant amount of research has also been conducted in the area of applying mechanism knowledge to real life problems, especially with respect to e-procurement. In a field study Beall (2003) found e.g. that English reverse auctions are most appropriate to “*source goods and services that are highly standardized, have sufficient spend volume, can be replicated by a reasonable number of qualified competitors, and have insignificant switching cost*”. Jap (2002) describes the importance of pre-qualifying potential suppliers before running an electronic auction, while Kambil & Sparks (2001) recommend to always use soft-closing rules¹ for procurement auctions. Millet et al. (2004) use regression analysis and machine learning to deduce recommendations from historic e-auction data of a large company. According to their results, procurement auctions are most successful if 5-6 suppliers bid on 2-8 lots of goods in a time window of 2.5 to 5.5 hours. A more conservative approach is taken by Emiliani (2006) who finds electronic auctions extremely counter productive for long-term buyer-supplier relationships and thus argues that avoiding this mechanism and instead cooperatively improving the supply chain is much more fruitful in the end. Still studies like the ones mentioned beforehand have limitations in their descriptive power as well. Most of them were conducted for (a) specific industries only and (b) focusing only on very few mechanisms, e.g. English reverse auction vs. catalogue procurement. Additionally, all these studies define mechanism related terms like “*English reverse auction*” only in an informal manner leaving readers alone with a considerable amount of uncertainty on how the mechanism details might look like.

In summary, theoretical mechanism design provides apt mechanisms for only very restricted settings. If those settings are slightly changed, the mechanism may lose its properties. The number of analyzed restricted settings is in total relatively small, such that mechanism design can only provide little guidance for practical design. Besides theoretical approaches towards mechanism design, also a considerable amount of empirical literature exists and can be used as a source for deducing mechanism recommendations. Unfortunately this literature is mostly limited in scope, mechanism description and coverage; furthermore the two fields of research are not very consistent in their results and even e.g. within the descriptive literature different (opposing) opinions and lines of argumentations have to be considered and harmonized.

Overall neither the theoretical nor the empirical literature provides a systematic methodology for engineering procurement mechanisms. Thus, a new approach is proposed that is capable of using the knowledge accrued by mechanism design, experimental economics, management literature and expert interviews. This approach needs to combine different results (interpreted as economic effects) in the form of cases, which describe context (economic environment), mechanism and outcome and may be collected from various sources, e.g. experiments or literature. To cope with these requirements KMS² uses a case-based reasoning approach for the generation of mechanism recommendations. This approach is also used in similar applications like e.g. the SAGE Solvent Alternative Guide (SAGE,

¹ A soft closing rule is a bidding time extension that is executed whenever a bid occurs within the last minutes of an auction in order to ensure that competing bidders have sufficient time for a reaction (Ockenfels, 2002).

² A Prototype of KMS is available online on <http://www.anegom.de>

2005). Though, to the author's best knowledge, there is no other system so far aimed at providing automated and systematic support to market engineers designing market mechanisms.

4 KNOWLEDGE ACQUISITION, STORAGE AND EVALUATION

Before describing the knowledge acquisition and its subsequent processing, we will shortly sketch up what "knowledge" means in our context. Davenport & Prusak (2000) define knowledge as "*a fluid mix of framed experience, contextual information, values and expert insight that provides a framework for evaluating and incorporating new experiences and information.*" Thus the KMS system has to accomplish both, providing (i) a storage facility for *contextual information, values and expert insight* and (ii) a mechanism that allows knowledge retrieval in a context of *new experiences and information*. For fulfilling the first part, KMS offers several different fields to store e.g. verbal recommendations (e.g. "use an English reverse auction"), literature and other references as well as a variable number of parameters describing the preconditions for which the recommendation holds (e.g. "at most low probability of collusion among bidders" for the recommendation to use an English Auction). Figure 2 shows an example screenshot of KMS during data entry of new knowledge.

The second part is implemented by an inference mechanism that takes a set of parameters describing a situation the user seeks advice for, and computes similarities to those cases (situations) already stored in the knowledge base. Sufficiently similar cases, which consist of a set of preconditions describing the procurement situation as well as suitable recommendations, are returned to the user conveying knowledge on how to proceed best in the respective setting.

As with all knowledge based systems, the most crucial task for the KMS project is the acquisition, adaptation, verification and maintenance of the underlying knowledge base. Especially challenging in this case is the fact that normative literature on auction design could be a possible source for providing knowledge, as can be empirical literature, structured interviews e.g. with procurement experts from industry, or even common sense (c.f. section 3).

For the acquisition of knowledge we followed a twofold approach: On the one hand we took recommendations from existing literature, identified their respective prerequisites, condensed them into a parametric format and stored them into the knowledge base. On the other hand we conducted interviews with procurement experts from several different industries trying to confirm that the findings from literature are in line with business practice in today's industry sourcing. For the interviews we chose a semi-structured format (Drever, 2003) which allowed us to collect structured data (e.g. type and size of the companies, industry the expert comes from, type of products mostly procured) but also left room for exploiting topics that were raised throughout the course of the discussions. Experts e.g. oftentimes expressed difficulties when asked to provide advice on how to proceed best in stylized procurement situations that were described only on an abstract level. In these cases, specific examples drawn from the business domain of the expert helped clarifying the issues.

Additionally a specific feature of the knowledge base proved especially valuable when eliciting advice during interviews: While oftentimes unable to give a definite recommendation on which mechanism to choose best, experts were still quite clear on which mechanism *not* to choose. E.g. in a case where strong bidder asymmetries occur it is easy to predict that an English auction will lead to an inefficient outcome, while it is not clear if e.g. a Dutch auction or an electronic negotiation might be the more favourable alternatives instead. In such a case, the KMS user still receives the warning (i.e. negative recommendations) not to choose an English auction, which increases his awareness and helps him avoid stepping into a "trap" of severe design failures.

With a growing number of recommendations entered into the knowledge base, data consistency becomes an important issue. As users are allowed to define their own parameters, rules, and recommendations, an automated approach for consistency checks is hard to implement. Thus our current solution to this problem is to give users of KMS the possibility to manually check at the time of entering new recommendations into the system, which other existing recommendations also match

their specific set of preconditions (c.f. button "Show all matching recommendations" in Figure 2). Furthermore we implemented a rating system that enables other users to judge recommendations and like this provides qualitative information on goodness of the knowledge stored.

Figure 2 shows a screenshot of KMS that displays the administration page used to enter or adjust recommendations. Basically a procurement expert wishing to enter knowledge into KMS needs to specify the type of recommendation that should be stored (e.g. "Recommendation", "Warning",...). Subsequently he fills in short and long descriptions of his recommendation (e.g. "Use English Reverse Auction") and lastly he may add References to related resources in order to increase the credibility of this recommendation.

Left in this state, the recommendation would be generally valid and thus always displayed to users using KMS no matter which search parameters they specify. In order to limit the applicability or scope of a recommendation one can add an arbitrary number of preconditions. KMS then only returns the recommendation as a search result if the preconditions can be matched³ with the search parameters.

For specifying a precondition, the expert first has to choose a parameter (e.g. "Switching Cost" of a product) from an (extensible) list of parameters provided by KMS. In the case of "Switching Cost", the parameter is specified as an enumeration⁴, which basically means that for this parameter a predefined, ordered set of parameter values is given. After having selected the parameter, the expert needs to determine the parameter value (e.g. "Medium") and an operator (e.g. "LessOrEqual"⁵) in order to finish adding the precondition. Overall, the expert specified in this case that "Use English Reverse Auction" is a valid recommendation iff product switching cost are at most medium.

Edit Recommendation

Please specify the recommendation details below.

Description

RecommendationId: 330 Rating: 3.5

Type: Recommendation

Short Description: Use English Reverse Auction

Description: In this case you might use an English Reverse auction.

Reference: Beall et al. (2003)

Preconditions

Id	Name	Type	Predicate	Value	Is knock-out?	Options
43	Switching Cost	Enumeration	LessOrEqual	Medium	false	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
49	Complexity	Enumeration	LessOrEqual	Medium	false	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
44	Direct cost reduction	Enumeration	GreaterOrEqual	Important	false	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
45	Collusion probability	Enumeration	LessOrEqual	Low	true	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
41	Market Power Distribution	Enumeration	Equal	Buyer's market	true	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
48	Long-term relationship	Enumeration	LessOrEqual	Unimportant	false	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 2: Storing a recommendation in KMS

³ A recommendation is matched if its preconditions are "sufficiently similar" to a user's search parameters (c.f. Section 5.2).

⁴ Supported parameter types in KMS are *String*, *Boolean*, *Number*, *Decimal*, and *Enumeration*

⁵ Different operators are provided for different parameter type as e.g. LessOrEqual is not meaningful for string parameters.

In the long term, the system's success will heavily depend on the participation of users and their contribution of knowledge. There are two scenarios intended to ensure the sufficient supply and verification of recommendations for KMS. Firstly the system can be offered as an add-on to configurable marketplace solutions (e.g. SAP, Moai, SupplyOn) where the knowledge acquisition is accomplished by the commercial vendor. In this scenario an integration into the existing software landscape of the respective company is desirable, where KMS might receive parameters from Business warehouses and return responses directly to the market platforms avoiding user interaction. Like this the market platforms can be automatically pre-configured according to the KMS recommendation and thus provide their users with sensible and automatically adjusted defaults.

Secondly the introduction of *public* and *private* recommendations is considered. While in this scenario the system still searches the complete knowledge base, only *public* recommendations are directly returned as results. For private recommendations a disclosure request can be send to the anonymous holder of the private information giving this person the possibility to freely decide if or if not to share his knowledge without him fearing any threats. This approach might be especially useful within company internal networks, where otherwise people might deny entering their knowledge for fear of losing their competitive advantage over colleagues and thus worsen their position in future labour negotiations.

5 DESIGN & IMPLEMENTATION OF THE SYSTEM

In this paragraph we focus on the implementation of the KMS system as described in section 3. First, the system architecture is introduced to establish a common notion of the domain model and its interaction with the system. Subsequently, the case-based reasoning for the recommendation retrieval is shown.

5.1 System Design

Following the typical Separation of Concern (SoC) pattern, our knowledge based system is divided into five distinct application tiers (Alur et al., 2003), user Interface, controller, service, persistence and domain model. Each of these layers encapsulates its specific tasks and logic from the other layers in order to achieve a maximum code decoupling and like this a high system stability, maintainability and adaptability.

The domain model is implemented in a relational database as displayed in Figure 3. The main entity is called *recommendation* which stores instances of recommendations, warnings and so on. For each recommendation to be valid, 0..* prerequisites must hold. These prerequisites are specific values (or value ranges) from different parameters, stored in the database. If a recommendation is true, not only a verbal description as stored in the *recommendation* entity but also a structured (parameterized) recommendation stored in *Mechanism* and *MechanismParam* may be returned. These mechanism parameter sets could be parsed into several formats (like e.g. XML, property files, ...) that afterwards might be used to automatically pre-configure market platforms like meet2trade.

The relational database storing the domain model is accessed from the KMS application using a distinct *persistence layer*, which allows the manipulation of the data using the data access objects (DAO) pattern. Like this, the underlying storage technology could be appended or switched with minimal impact on the program itself. Above the persistence layer, a *service layer* implements the more complex business logic like e.g. the case-based reasoning algorithm. Separating this logic from the DAO on the one hand, and from the *application workflow* on the other, ensures that e.g. different recommendation retrieval mechanisms could be implemented without changing the principal workflow. The last distinct layer is introduced between *application workflow* and *view layer*, the fifth

layer of KMS. This separation allows different front-ends like a HTML interface and a web services interface to be implemented transparently using the same application logic.

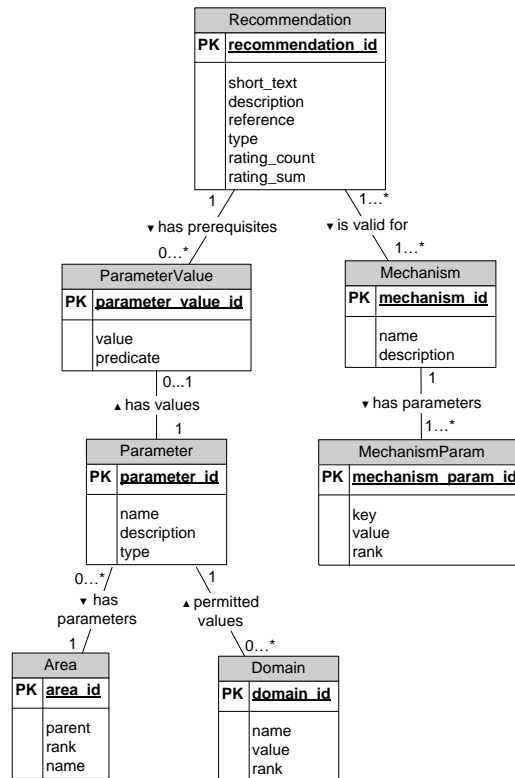


Figure 3: ER-Diagram of the KMS Knowledge Base

5.2 System Workflow for Recommendation Retrieval

For the implementation of recommendation retrieval algorithms, several approaches already exist. Forgy (1982) introduces RETE, an algorithm for matching many patterns on many objects, which is oftentimes used in rule based expert systems. Many alternatives have been proposed since then, the most notable ones being TREAT (Miranker, 1987) and LEAPS (Batory, 1994). The main problem with this group of algorithms is of technical nature: Existing implementations of these rule engines rely on proprietary storage formats that do not cope well with traditional DBMS. To the authors' knowledge only one (quite complex) approach exist that adapts the RETE algorithm to directly work on a database (Jin et al., 2005).

For our system a database for storage and retrieval of recommendations was more advantageous as it provides a convenient way to store verbal recommendations along with structured information and allows easy manipulation of the stored data. Thus we adapted an alternative approach for the recommendation retrieval which stems from the research on recommender-systems. In this area, case-based reasoning is oftentimes used to compute similarities between a new case and existing (historic) cases (Chi & Kiang, 1991; Porter et al., 1993).

We implemented a case based reasoning algorithm that compares a new case (recommendation request) to cases (recommendations) already stored in KMS. Like this the task of finding a suitable mechanism recommendation can be reduced to comparing parameter lists with each other and returning one list if the number of matches between the list elements exceeds a certain predefined threshold value.

The first list (Figure 4) consists of parameters a user enters into the system in order to describe the procurement situation he seeks advice for. The second list contains parameters from the same parameter domain as the first list but in this case the parameters are prerequisites that must be fulfilled for the recommendation to be valid. Figure 5 shows schematic examples of these lists. For the recommendation retrieval, the input parameter list is compared with each recommendation prerequisite list stored in the knowledge base. For each comparison cycle the similarity between all list items from both compared lists is computed on a per attribute basis. If an attribute is found in the input parameter list but not in the respective recommendation prerequisite list, the parameter is counted as *relaxation*, as it is not necessary for the current recommendation to be valid. If a parameter on the other hand is only found in the recommendation prerequisite list, it is counted as *restriction* as the parameter was not specified by the user but is required for the recommendation to be valid.

If a parameter is found in both lists, the similarity between both parameter values will be computed. It is counted as a *match* if the similarity exceeds an (adjustable) threshold level. Finally, after all comparisons, three measures are available indicating the matching quality of a recommendation:

- # restrictions
- # relaxations
- matching quality := # matching parameters / total # parameters matched

A recommendation is returned to the user if (a) its matching quality exceeds a predefined threshold, (b) the number of restrictions does not exceed a predefined threshold, and (c) none of the "restriction" parameter was marked as "knock-out" criterion. For convenience, the results are sorted by matching quality in descending order. The number of relaxations and the number of restrictions are also displayed to the user as further indicators.

Input Parameter List		
Name	Value	Type
Objective	Reduce buy price	String
Current Process	Manual orders	String
Prod. Type	Good	String
Prod. LC	Commodity	String
Prod. Quantity	100.000	Integer
# Negotiable Attributes	1	Integer
# Sellers	10	Integer
# qualified Sellers	3	Integer

Figure 4: Input Parameter List

Recommendation Prerequisite List			
Name	Predicate	Value	Type
Objective	=	Reduce buy price	String
Current Process	≠	Auctions	String
Prod. Type	=	Good	String
Prod. LC	=	Commodity	String
# Negotiable Attributes	=	1	Integer
# Sellers	near	7	Integer
Recommendation: Use English Reverse Auction			
Mechanism Parameter Name		Value	
Order visibility:		Best order only	
Allow Order Revocation		false	

Figure 5: Recommendation Prerequisite Lists

6 SUMMARY & OUTLOOK

The paper at hand proposes the conception and implementation of a decision support system for selecting procurement mechanisms. The reasoning component is realized by means of a case-based reasoning approach. In contrast to other approaches, such as (manual) mechanism design, the proposed knowledge based approach is capable of generating recommendations by combining several effects and patterns. The peculiarity of the approach is that it can make recommendations in cases, which are

hitherto not studied and other sources are silent. The proposed approach is shown to work by a prototypical implementation.

This paper is a step towards understanding the effect and strength of different procurement mechanisms in different scenarios. Contributions include the definition of an extensible default domain model and the integrated case based reasoning approach. The prototype KMS is intended to support practical design by making reasonable recommendations.

Future research needs to further investigate possibilities for providing incentives to users to actively contribute knowledge to the system avoiding free rider phenomena known from p2p systems. Future research as well will be the confrontation of KMS with practical market engineering. To gear up KMS for such a purpose, the knowledge base needs to be extended; additional recommendations need to be extracted from theory and experiments. Once the knowledge base contains a critical mass of recommendations, a field experiment with the entire CAME tool suite will demonstrate the usefulness of the approach.

Overall, KMS is a useful system that supports industrial sourcing managers with systematic decision support on which procurement mechanism to choose best in which procurement situation. Thus, it can be offered as add-on to procurement systems. However, KMS is not limited to procurement scenarios and could potentially also be used for C2C auctions such as eBay. If an auction house allowed configuring several mechanisms, the private user can become a market designer by means of the KMS. In this case, although useful and usable on its own, the full advantage of this system will unfold especially in combination with other CAME tools that allows the use of KMS recommendations to automatically configure and launch recommended market mechanism instances reducing manual user effort as much as possible.

References

- Alur, D., Malks, D. and Crupi, J. (2003) *Core j2ee patterns: Best practices and design strategies*. Prentice Hall.
- Batory, D. (1994) The leaps algorithm. University of Texas at Austin, Austin, TX, USA.
- Beall, S., Carter, C., Carter, P. L., Germer, T., Hendrick, T., Jap, S., Kaufmann, L., Maciejewski, D., Monczka, R. and Petersen, K. (2003) The role of reverse auctions in strategic sourcing. CAPS Center for Strategig Supply Research.
- Bichler, M. (2001) *The future of e-markets: Multidimensional market mechanisms*. Cambridge University Press, New York and Melbourne.
- Casaseca, J. P. V. (2005) Manufacturers of e-marketplace solutions. eMarketServices, Spain.
- Chi, R. H. and Kiang, M. Y. (1991) An integrated approach of rule-based and case-based reasoning for decision support. In *CSC '91: Proceedings of the 19th annual conference on Computer Science*, pp 255--267, ACM Press, New York, NY, USA.
- Clarke, E. H. (1971) Multipart pricing of public goods. *Public Choice* V11 (1), 17--33.
- Czernohous, C. (2005) Simulations for evaluating electronic markets: An agent-based environment. In *SAINT-W '05: Proceedings of the 2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops)*, pp 392--395, IEEE Computer Society, Washington, DC, USA.
- Davenport, T. H. and Prusak, L. (2000) *Working knowledge*. Harvard Business School Press.
- Drever, E. (2003) *Using semi-structured interviews in small-scale research. A teacher's guide*. The SCRE Centre.
- Emiliani, M. L. (2006) Executive decision-making traps and b2b online reverse auctions. *Supply Chain Management: An International Journal* 11 (1), 6-9.
- Forgy, C. L. (1982) Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence* 19 (1), 17-37.
- Green, J. and Laffont, J.-J. (1977) Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica* 45 (2), 427--438.

- Groves, T. (1973) Incentives in teams. *Econometrica* 41 (4), 617--631.
- Holmstrom, B. (1979) Groves' scheme on restricted domains. *Econometrica* 47 (5), 1137--1144.
- Hurwicz, L. (1973) The design of mechanisms for resource allocation. *American Economic Review* 63 (2), 1-30.
- Jap, S. (2002) Online reverse auctions: Issues, themes, and prospects for the future. *Journal of the Academy of Marketing Science* 30 (4), 506-525.
- Jin, C., Carbonell, J. and Hayes, P. (2005) Argus: Rete + dbms = efficient persistent profile matching on large-volume data streams. In *Foundations of Intelligent Systems* (Hacid, M.-S. and Ras, Z. W. and Tsumoto, S., Eds), pp 142-152, Springer.
- Kambil, A. and Sparks, M. S. (2001) Seizing the value of e-procurement auctions.
- Katok, E. and Roth, A. E. (2004) Auctions of homogeneous goods with increasing returns: Experimental comparison of alternative dutch auctions. *Management Science* 50 (8), 1044--1063.
- Kersten, G., Neumann, D., Vahidov, R. and Chen, E. H. (2006) A framework for e-market assessment: The case of online auctions and e-negotiations.
- Klemperer, P. (2001) How (not) to run auctions: The european 3g telecom auctions. Economics Group, Nuffield College, University of Oxford.
- Kolitz, K. and Weinhardt, C. (2006) Mes - ein experimentalsystem zur untersuchung elektronischer markte. In *Service Oriented Electronic Commerce*, pp 103-118.
- Mäkiö, J. and Weber, I. (2004) Component-based specification and composition of market structures. In *Coordination and agent technology in value networks* (Bichler, M. and Holtmann, C., Eds), Gito Verlag.
- Millet, I., Parente, D. H., Fizel, J. L. and Venkataraman, R. R. (2004) Metrics for managing online procurement auctions. *Interfaces* 34 (3), 171-179.
- Miranker, D. P. (1987) Treat: A better match algorithm for ai production system matching. In *AAAI*, pp 42-47.
- Myerson, R. (1981) Optimal auction design. *Mathematics of Operations Research* 6 (1), 58--73.
- Myerson, R. B. and Satterthwaite, M. A. (1983) Efficient mechanisms for bilateral trading. *Journal of Economic Theory* 29 (2), 265-281.
- Neumann, D. (2004) Market engineering - a structured design process for electronic markets. University of Karlsruhe, Germany.
- Neumann, D., Mäkiö, J. and Weinhardt, C. (2005) Came - a tool set for configuring electronic markets. In *ECIS*.
- Ockenfels, A. and Roth, A. E. (2002) Last-minute bidding and the rules for ending second-price auctions: Evidence from ebay and amazon on the internet. *American Economic Review* 92 (4), 1093-1103.
- Porter, B. W., Bareiss, R. and Holte, R. C. (1993) Readings in knowledge acquisition and learning: Automating the construction and improvement of expert systems. (Buchanan, B. G. and Wilkins, D. C., Eds), pp 741--758, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Sage (2005) Sage solvent alternatives guide. RTI International and U.S. EPA Air Pollution Prevention and Control Division, pp <http://www.clean.rti.org/es/index.cfm>.
- Sandholm, T. (2002) Algorithm for optimal winner determination in combinatorial auctions. *Artif. Intell.* 135 (1-2), 1--54.
- Strecker, S. and Seifert, S. (2003) Preference revelation in multi-attribute bidding procedures: An experimental analysis. In *Proceedings of the 24th International Conference on Information Systems (ICIS)*, pp 850-854, Seattle, WA.
- Vickrey, W. (1961) Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance* 16 (1), 8--37.
- Weinhardt, C., Holtmann, C. and Neumann, D. (2003) Market engineering. *Wirtschaftsinformatik* 45 (6), 635-640.
- Weinhardt, C., Neumann, D. and Holtmann, C. (2006) Computer-aided market engineering. *Communications of the ACM* 49 (7), 79.