meet2trade: An Electronic Market Platform and Experiment System□

Klaus Kolitz[1], Carsten Block[2], Christof Weinhardt[3]

[1] University of Karlsruhe, Information Systems and Management
Englerstr. 14, 76131 Karlsruhe, Germany
kolitz@iism.uni-karlsruhe.de

[2] University of Karlsruhe, Information Systems and Management
Englerstr. 14, 76131 Karlsruhe, Germany
neumann@iism.uni-karlsruhe.de

[3] University of Karlsruhe, Information Systems and Management
Englerstr. 14, 76131 Karlsruhe, Germany
weinhardt@iism.uni-karlsruhe.de

Abstract
The development of new electronic markets is challenging, since many factors influence the market outcomes and hence the markets' success. Even worse, a fundamental lesson learned from economics is that details matter: small changes in market design can have a significant impact on the market participant's behaviors and thus on the achieved outcomes. Consequently a well structured process for design, implementation, testing and maintenance of markets is required. meet2trade is a software tool suite designed to systematically support each step of such a Market Engineering (ME) process. This paper presents the generic trading platform meet2trade that enables users to individually configure their own electronic markets, to run them on the integrated auction server, and to evaluate them using the built-in full-featured lab experiment system.

Keywords: CAME (Computer Aided Market Engineering), electronic markets, market platform, market modeling language, experimental economics, software engineering

1 Introduction

During the last two decades the progress of information technology has lead to the appearance and the rise of electronic markets. Large corporations negotiate deals with their suppliers using online negotiation platforms, auction sites like eBay play an important role in many people's everyday life, and financial instruments are traded on electronic trading systems like XETRA. The drastically increased importance of the different forms of electronic markets for today's society requires a thorough understanding of the underlying market mechanisms e.g. to be able to anticipate the market outcome in order to evaluate the potential of new markets and to rule out unwanted effects.

In electronic markets the market outcome depends on the market participants and their valuations as well as on the market rules that can stimulate strategic behavior of the participants. Even small changes in the market rules can generate enormous effects on the market outcome. For example, according to Ockenfels and Roth (2005) the bidding behavior in an English auction with a fixed end differentiates from an English auction with soft end. Seifert and Ehrhart (2005) analyze the design of the third generation spectrum auction in the UK and in Germany: The bidder surplus is lower in the German auction but revenue is higher compared to the UK design. Moreover, the Austrian 3G auction, although mechanism wise identical with the German one, yielded prices only little over the reservation value simply because participants had learned their lessons from the earlier German auction and thus adapted their bidding strategies (Klemperer 2002). Taking another example of the Federal Communications Commission (FCC) regional narrow-band auction in 1994, where thirty licenses were offered for sale (c.f. Ayres and Cramton 1996 and Cramton 2004). On ten of the thirty licenses a 40 percent bidding credit was granted and offered to businesses owned by minorities. By subsidizing these designated bidders competition was enhanced and unsubsidized firms were induced to bid higher - this increased the revenue for the government by more than 12%.

Since electronic markets have become essential mechanisms within daily business, the design of innovative mechanisms for new application domains has become an important field of research complementing the study of existing (electronic) markets. Furthermore, increasing competition among operators of electronic markets creates pressure for the functional expansion and refinement of already existing markets. Therefore, innovative market features and improved market design methodologies are required. In this context Weinhardt et al. (2003) have developed the Market Engineering (ME) methodology for the structured development of electronic markets. ME takes into account economic, technical, and juridical aspects of electronic markets. Neumann (2004) proposes Computer Aided Market Engineering (CAME) as a process that consists of four (possibly revolving) phases: (i) *Design*, (ii) *Configuration*, (iii) *Testing*, and (iv) *Operation*. During the design phase a knowledge based system supports the market designer in choosing an appropriate market structure (i.e. business rules for the exchange process). In the next step the electronic market is configured considering the chosen mechanism and the requirements of the application domain. In order to ensure that the newly designed electronic market meets the requirements and leads to the desired results thorough testing needs to be conducted in the third phase of the CAME process. After sufficient testing (and possibly the adaptation of the business rules), the electronic market is ready for operation.

Because each electronic market has its own characteristics determined by specific user requirements and certain environmental conditions, there is a need to tailor each and every market individually to suit these needs. Thus a generic and flexible electronic platform that supports all phases of the market engineering process ranging from the market design and configuration to the testing and operation of the new markets is required. The generic electronic trading platform meet2trade was developed to meet these requirements. meet2trade provides a so called Market Modeling Language (MML) for the easy design and configuration of electronic markets and it is capable of operating markets with different market rules through its generic auction server. Since the impact of market rules and environmental influences on the market outcome are hard to predict, and since the outcome largely depends on the (sometimes irrational) behavior of the market participants, there is a strong need for the evaluation and pre-testing of the individually configured markets. For this type of assessment two important methodologies in Economics are simulations and experiments. Therefore support for both methodologies has been built into meet2trade through two software tools: the agent based simulation environment AMASE (cf. Czernohous 2005) and the meet2trade experimental system MES. In this paper we focus on the description of the meet2trade Experimental System (MES), which allows the easy configuration and execution of laboratory experiments for the study of the market rules' impact on the strategic behavior of the participants and the market outcome.

The remainder of this paper is structured as follows: the next section discusses related work within the area of electronic markets and experimental systems suitable for the examination of e-markets. Section 3 introduces the electronic trading platform meet2trade giving a brief overview on the platform requirements, its architecture and technologies as well as its application domains. Section 4 specifically describes the MES before we conclude this paper with a summary and a brief outlook in section 5.

2 Related Work

The importance of the continuous improvement of existing markets and the design of innovative future markets shows the need for an electronic trading platform that allows the practical testing of markets (cf. Neumann et al. 2005). The crucial features of such a trading platform are the easy configuration of new markets and the provision of the necessary tools for the examination of those markets through experiments and optionally additional other means like simulations.

During the last years the research of electronic markets has focused on generic systems providing various auction or negotiation protocols. The core of meet2trade is an auction server providing a multitude of auction formats. McAfee and McMillan (1987) define an auction as “a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants.” Consequently, the main task in designing and realizing electronic markets within the platform is to implement a single trading process that supports the set of auction rules. Thus, *standardization* of the trading process is the key to a generic as well as to a flexible and reusable platform (c.f. Mäkiö et al. 2004). The key idea of standardization is also suggested by Kersten and Teich (2000), who state that all auctions are standardized negotiations. Since the activities of the negotiation determine this process, the standardization requires the identification of these basic activities including rules and parameters (Kersten et al. 2004).

One possibility for the market configuration also employed by the meet2trade system is the identification of rules and components to enable a parametric approach towards the configuration of the trading process and therefore of electronic markets. This approach is also argued for by Wurman et al. 1998 and Wurman et al. 2001. The model Wurman et al. suggest is based on parameters that are common to multiple types of auctions such as single-sided or double-sided auctions, as well as multi-commodity auctions, e.g. combinatorial auctions. It identifies various independent parameters within three core activities of auctions: (i) bidding rules, (ii) clearing policy, and (iii) information revelation. These activities span a multi-dimensional auction design space into which most of the well known auction mechanisms can be mapped. This concept is implemented in the *Michigan Internet AuctionBot* (Wurman et al. 1998), a general platform for price-based negotiation providing a flexible approach of decomposition and parameterization of auction mechanisms. The Michigan Internet Auction Bot is able to manage a large number of auctions simultaneously and can be accessed through a browser-based interface. Even though no particular mechanisms for the restriction of the system ensuring controlled laboratory experiments have been implemented, the system can still be used within limits to conduct (internet) experiments or classroom exercises.

Another approach is used by the Invite platform (Kersten et al. 2004). Invite provides a generic run-time environment for negotiations and is designed as a regular web application. The web application design makes it possible to access Invite from any computer running a web browser and therefore in contrast to meet2trade employs the client-pull concept. Consequently an Invite user has to manually reload pages in his web browser in order to see information updates like e.g. incoming offers, while the meet2trade server automatically pushes new information to its java-based clients. Invite uses a three phase process model to represent negotiations: pre-negotiation analysis, conduct of negotiation, and post-settlement analysis. The pre-negotiation phase consists of the analysis of situation, problem and opponent(s), as well as on the elicitation of preferences, reservation levels, and strategy. During the negotiation phase the exchange of messages and offers, the evaluation of offers, and the assessment of the progress of the negotiation take place. Finally, the post-settlement phase concludes the negotiation with the evaluation of the outcomes generated by, and after, the negotiation activity. In order to support a large variety of negotiation protocols and for the easy reconfiguration of protocols, Invite relies on the decomposition of the protocol into so called activities. An activity is a concrete element of a negotiation, e.g. the submission of an offer. On the system level all activities are represented by actions: each activity is associated with at least one action, which are basically code fragments used to (i) retrieve or store user input and data, (ii) to generate output, or (iii) to invoke the execution of external applications. This decomposition approach allows for easy reuse of code segments to design and configure new negotiation protocols. Invite is specifically designed as a platform for negotiation experiments. Therefore it includes a web-based configuration tool for the easy administration of users and negotiation instances. New experiments and protocols can be configured easily, yet it requires a profound understanding of the system and some programming skills, since the configuration process usually includes the definition of web-page sequences to display, the storage of experiment specific data (e.g. questionnaires) into the database, and - depending on the specific experiment - also the programming of HTML, Javascript and Cold Fusion code.

Another market framework implementing common auction formats that can easily be adapted to new application domains and that allow a dynamic configuration is the Global Electronic Market (GEM) (Reich and Ben-Shaul 1998). Its ability to adapt to different settings and domains is based on the decomposition of the market into independent components, or more precisely into parameters used to tailor distinct aspects of market mechanisms. In particular the heart of the market framework consist of (i) a so called *Order Verifier* that determines which orders to accept, (ii) a Market Maker component that solves the allocation problem and determines the price, and (iii) a *Schedule* component that manages the timing of the orders to be executed. Additionally a dynamic configuration of the auction is facilitated by a meta-component called *Builder*, which is able to initialize new components and even replace components during run-time. GEM includes neither dedicated support for executing laboratory experiments, nor tools for user management or administration of market instances. Therefore its use for experiments is limited. Because the system employs a component-based architecture and provides its infrastructure through a well-defined API, it is still possible to implement experiments on using the GEM platform, even though the effort is much larger than with a platform which offers dedicated experiment tools.

The concepts presented above, are only a few examples that are taken from the landscape of electronic market design and electronic market platforms. They focus on the decomposition of the underlying transaction process into core components. To generate new auctions or negotiations, these components can be either recombined or adjusted through parameters. This approach enables the design and configuration of a multiplicity of auction types. The presented approaches are similar to the key idea of the market configuration process in the meet2trade system. Tools for the control of participants behavior in economic experiments and for the administration of users and experiment instances complements some of the described systems and make them more valuable as market engineering tools. The auction platforms presented, i.e. AuctionBot and GEM are comparable with meet2trade as they also facilitate an easy configuration and implementation of various auction types, even during run-time of the auction servers. The following section now describes the distinct features of meet2trade in more detail.

3 The electronic market platform meet2trade

meet2trade was developed as a tool suite for conducting research on electronic markets and as a proof-of-concept for innovative trading concepts introduced throughout the course of the e-FIT project. This section describes the platform, presents the system's architecture and provides further details on the core system’s functions and concepts.

3.1 Platform overview

The main concepts of the meet2trade platform are flexibility from a system point of view and configurability from the user perspective. Flexibility in the system context means

1. the ability to host markets for a large variety of application domains on the same platform (e.g. financial markets, real estate markets, industrial procurement, …),
2. the automatic adoption of the system to the various requirements of different domains (e.g. product structure, order structure),
3. the ability to support single-sided auctions as well as double-sided auctions with seamless integration into the system,
4. the facilitation of fast development and evaluation for new electronic markets, and
5. the implementation of newly developed trading concepts like bundle trading or innovative order types.

The main goal from the user's perspective is to allow for as much configurability and customization of the platform to individual trading needs as possible. This means that users should be able to (i) select (ii) combine, and (iii) configure markets according to their individual preferences. Furthermore, they should be able to customize the graphical user client by setting up a personal workbench. A further comprehensive step towards providing high flexibility to end-users is to maintain a set of auctions the user can select from. Lastly, it is desirable to allow users to submit *one single* order to a combination of markets at once, which then might be executed on *any* of the selected markets but only *once* overall. meet2trade provides all of the aforementioned features including a mechanism design space ranging from simple markets that are executed autonomously, over sequences of markets to a complex market structures with parallel and / or sequentially executed market segments.

In meet2trade users have complete control over the market structure as they are able to design their own auction-based markets. To facilitate this market design and configuration process, the market modeling language (MML) was developed. This XML based language provides approximately 100 concepts (i.e. parameters with their respective parameter spaces) that can be used to specify all types of auctions and business rules supported by meet2trade. The innovative platform concepts offered by this system - market configuration and platform flexibility, bundle trading and new order types, just to name a few - provide a good starting point for economic research on auction mechanisms and related questions like e.g. user behavior. The meet2trade system delivers not only the platform to host the markets, but also provides a tool suite for their examination. The tools offered by meet2trade consist of the agent-based simulation environment AMASE and the experiment system MES , which is described in detail in chapter 4.

3.2 System architecture and technologies

The meet2trade suite follows the typical client-server architecture with a central server that provides the runtime environment for all available markets and is responsible for data management (e.g. user data, account data, product information data, protocol data, experiment series data). Clients connected to this central server are responsible for displaying all relevant data to users and to provide them with an interface for the submission of orders and bids. The meet2trade server is programmed in Java using the Enterprise Java Beans (EJB) concept while the client is programmed in Java using Swing for the rendering of the graphical user interface. In particular, the server follows 3-tier architecture as shown in Figure 2:

The communication layer prepares the data for client presentation. It also provides the general communication protocol for the client-server interaction and is used to facilitate the administration of all connected clients.

The business layer consists of the core market environment called ARTE (Auction Runtime Environment). From the technical point of view ARTE can be considered as a runtime-environment for auctions. It is conceptualized in a way that enables the simultaneous execution of an arbitrary number of auctions ARTE supports various auction types and technically provides auctions with multiple services like logging, time based job scheduling, user management or communication. Consequently order processing takes place here as well.

The database layer transparently encapsulates database access from the rest of the system and furthermore provides a logging mechanism for all trading data as well as a user and depot management.

For client-server communication the Java Messaging Service (JMS) is used, which provides a reliable queue-based asynchronous communication mechanism. All data exchanged between client and server is encapsulated in XML messages. The XML format was used because the high degree of configurability and flexibility of the meet2trade system regarding user interface, order structure and domains called for a flexible yet easy to use format. Overall, the meet2trade server provides functionality for the deployment, configuration and operation of various auction mechanisms, the management of user accounts and depots, the processing of orders, and finally, the logging of all market data.

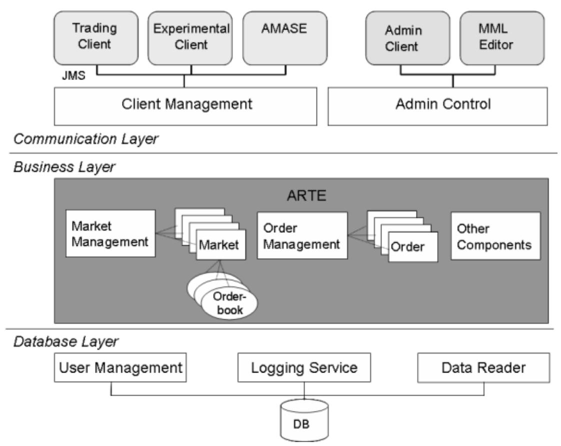

Figure 1: The system architecture of the meet2trade generic trading system

If a user wants to trade on the meet2trade system, he needs to be registered with the system and authenticate himself by logging on via the client. After the *User Management* has checked the permissions a user can configure new markets or participate in already existing markets by submitting buy or sell orders, which are received and forwarded to the specified market(s) by the *Order Management* component. From here orders are able to leave and enter markets according to their own specifications without further user interaction, since they are modeled as autonomous objects.

The *Market Management* administrates all markets running in the system. A market has two different states, active or passive. Orders can only be sent to active markets and are stored within the order book of a market. If two matching orders meet in an order book of a market, a trade can be executed according to the rules of this market. After a successful trade, the user depots which are stored in the database are updated. Thus, the clearing and settlement takes place within the system. Throughout the whole process all events occurring in the system, e.g. the arrival of a new order or the successful execution of a trade, are written into the database by the *Logging Service*.

3.3 Market Configuration through the Market Modeling Language (MML)

Understanding markets as set of rules and components enables a parameterized approach to market configuration. The MML was developed to describe electronic markets in a parametric way (c.f. Mäkiö and Weber 2004) in order to facilitate the construction of electronic auctions. The meet2trade markets follow the generic process that was inspired by the Media Reference model introduced by Schmid and Lindemann 1998 and the Montreal Taxonomy of Ströbel and Weinhardt (2003). The genericity of the transaction process stems from its reusability: one single basic process structure is used for any auction mechanism within meet2trade. Thus, the generic transaction process is implemented in a domain-independent and flexible manner: The basic process is detached from the application domain, from transaction objects, and from specific behavior of market participants. The resulting generic process is adaptable to different domains and to different trading objects within one domain by adapting its parameters. This approach allows a great flexibility in the design and creation of new auction mechanisms and facilitates an easy way of configuring and implementing them.

Using this basic process, users are able to configure single- and double sided auctions. Supported are inter alia the most well-kown sell-side auction types within the single-sided format like English auction, Dutch auction, first-price sealed-bid auction and Vickrey auction. Additionally buy-side auction types with a single buyer and multiple sellers, commonly referred to as “reverse auctions” are provided within the platform as well. Besides the various forms of single-sided auctions, continuous double auctions (CDA) and call markets can be configured and executed as well. In contrast to the former single-sided auctions with one seller and many buyers or vice versa, the latter auctions allow orders from m buyers and n sellers to be submitted. In the continuous double auction each submitted order will be directly executed as soon as a match becomes viable; in a call market all orders are collected and a matching takes place in discrete time intervals.

Even though meet2trade focuses on single-attributive single-unit auctions, meet2trade also supports multi-unit auctions with a generalized uniform pricing rule as well as multi-attribute auctions and combinatorial auctions. Multi-attribute mechanisms are implemented for continuous double auctions and English single-sided auctions. For combinatorial auctions a bundle trading mechanism, i.e. a double-sided auction mechanism, allowing bids on packages or bundles of items, is integrated into the platform (see Grunenberg and Weinhardt 2004).

All auction types mentioned above can be configured and combined within the meet2trade platform by parametrically describing them using MML. Approximately 50 parameters are available to describe the generic transaction process, e.g. to control matching and execution of submitted orders, to define participation restrictions, to restrict tradable products, to define special events like e.g. volatility breaks, or additional market rules such as bidding credits for weak bidders. The combination of the basic auctions into complex market structures can be defined using further 41 parameters. These parameters permit a market designer to combine simple market mechanisms in to so called meta-market constructs, by scheduling them either sequentially, parallel or both, sequentially and parallel (Czernohous et al. 2003). The *sequential combination* leads to a chain of market structures that are executed one after another. In this case a market takes over the unmatched orders from a predecessor market as initial endowment and hands over unmatched orders at the end of its execution period to the successor. In a *parallel combination* single market structures are running simultaneously. Here, the meta-market structure provides a parallel existence of several simple auction mechanisms. An order is submitted to all parallel running markets. Once it is executed in any of the markets, it will be automatically cancelled in all others in order to avoid double execution. The *sequential and parallel combination* allows arranging moth aforementioned combinations into one single meta market. This meta market concept one of the most significant properties of meet2trade. In combination with the generic market process and the parameterization approach it leads to a powerful auction configuration platform. For a more in depth description of the MML and the meta market concept see Mäkiö 2006.

3.4 Adaptive Client

Professional trading systems rely on real-time market data, like order book or market information and thus provide its users with the opportunity to immediately react to changes in the markets. Since HTML-based web applications cannot display real-time data without employing additional technologies like Macromedia Flash or embedded Java applets, the client was implemented as a Java Swing application allowing full support for the server push mechanism used to distribute real-time market data. But relying on this client concept offers other advantages as well:

- The client's interface configuration is easily adjustable to the user's needs as it allows free adjustment of the different windows (like order book, order entry mask, …) on the screen. Once a user finds a convenient setting he can be save and restore it in subsequent sessions by a mouse click. As these individual screen configurations are stored on the server, users can be provided with the same individualized meet2trade working environments on every client machine they log on.
- The user interface is built on standard windows components, like drop down menus, resizable, movable windows etc. This allows for higher usability than a HTML-based web application.
- The client side user display can be centrally supervised, restricted and monitored by the server. This is especially important when conducting economic experiments with the built-in experimental system of the meet2trade platform.

In order to display real time data in the client an intelligent data push mechanism was implemented in the meet2trade server. This mechanism is encapsulated in the *Client Management* component of the communication layer. Before a client can receive data from the server, it subscribes to this particular data provider (e.g. a specific order book or the trade history) using the *Client Management*. The subscription process is handled automatically by the client that keeps a subscription active as long as a window displaying a particular type of data is open. The subscription is deleted when the last window containing this data is closed.

As aforementioned in Section 3.3 the order attribute settings vary depending on the market. Thus, a mechanism is required for the client to adjust its order entry window according to the auction type and the attributes required by the market. Therefore, each market supplies a XML-based market specific description to the client about its required current parameters. Another challenge is to adapt the client to the several different auction types. Double-sided Auctions are usually visualized by a 2-sided order book containing sell orders on one side and buy orders on the other side. The sorting of the orders on both sides is configurable with the MML - commonly the 'best' order is on top of the order book. Single-sided auctions on the other hand do not require 2-sided order books, since after an auction has been started; only buy orders in the case of a sell-side auction or only sell orders in the case of a buy-side auction can be submitted. Furthermore the order entry mask of single-sided auctions can vary according to the auction type.

This complex client logic is necessary to allow on the one hand for as many kinds of markets as possible and on the other hand still keep the graphical user interface well arranged and thus easy to use for the users. This generic client described above is integrated into the experimental system MES which is presented in chapter 4.

3.5 Simulation Environment (AMASE)

Over the last years computational simulations became a popular methodology for the assessment of complex mechanisms and systems. Modeling different strategies using software agents enables economists to study markets, market participants behaviors and market developments over time at moderate cost. Agents applied in simulations normally use simple decision rules, learning algorithms, or statistical analysis to adapt their strategies. Tesfatsion (2002) provides a detailed overview on ACE research and describes studies of market simulations in electricity and financial markets (for further examples, see also Arthur et al. 1996 and Mizuta and Yamagata 2001).

In order to be able to conduct research on markets using agent-based simulations, AMASE was developed as part of the meet2trade tool suite. AMASE is based on the Java Agent Development Framework (JADE). A *Simulation Control Agent* (SCA) that generates all agents and manages the simulation by sending control messages to the participating agents is the heart of AMASE. Regular JADE agents are extended by a module called *Simulation Agent Control Behavior* (SACB), which enables these agents to receive and reply to simulation control messages from the SCA. The basic settings such as number of rounds, agent types, or endowment are controlled by the SCA and can be adjusted through a graphical user interface (SCA-GUI). Furthermore AMASE supports the automated repetition of simulations and the definition of treatment sequences. All simulation specific rules and methods are implemented within the *Simulation Management Behaviour* (SMB) component which is attached to the SCA and to the SACB as well. A more detailed description of AMASE is presented in Czernohous (2005). Van Dinther 2006 presents a study where AMASE is used to conduct research on adaptive bidding in single-sided auction under uncertainty.

4 The meet2trade experimental system (MES)

4.1 Experiments in Market Engineering

For the evaluation of electronic markets several different methodologies are suitable (cf. Weinhardt et al. 2003): the analytical assessment, where models and mathematical functions are used to predict market results, (agent based) simulations (cf. section 3.3), and laboratory experiments – the focus of this chapter.

Since laboratory experiments are conducted in a controlled environment they allow the isolation of individual influencing factors. To examine the influence of a particular factor, only this one is varied while all other factors are preferably kept constant. This method allows for example to evaluate the impact of particular market parameters on the market outcome. Experiments are an especially important methodology for discovering behavioral patterns which deviate from the predicted rational behavior. Therefore experiments are the preferred method, in particular when “human factors” have to be taken into consideration. Numerous studies have shown that the behavior of human market participants differs significantly from the theoretical predictions. For example bidders in single-sided auctions tend to be vulnerable to the so called “bidding fever”, which leads them to submitting higher bids during the course of an auction than originally intended (Ku 2000 and Heyman et al. 2004). Further factors which can influence bidding behavior include the mood of the participants (Bosman and Riedl 2003) and even the number of previous bids for an auction (Simonsohn and Ariely 2005). Economic experiments with human participants therefore play an important role in the evaluation and testing of newly designed markets. They especially help to estimate the effects of changes in the market design at an early stage. Ross Miller compares laboratory experiments for the assessment of markets to wind tunnels for the automobile development and bridge construction. He argues that with this methodology markets can be optimized continuously, which leads to the emergence of better, more secure and more intuitively usable markets (Miller 2002).

As argued before, experiments can be a very useful method for market engineering, though the design, setup and realization of experiments can be a very tedious and labor-intensive task. Furthermore, for technical and other reason, oftentimes the reduction of the actual real trading environment to a limited experimental one is necessary. This reduction process implies the risk of an unwanted falsification of outcome-related parameters, especially since the effect of the user interface and experiment environment on the participants is often not negligible (see e.g Schlosser 2003 and also Kersten et al. 2006 and Kolitz and Neumann 2007) and furthermore hard to predict.

4.2 System Description and Implementation

In order to reduce at least the aforementioned technical problems and in order to be able to conduct a large variety of market experiments in an easy and fast manner, the meet2trade experimental system (MES) was developed as a configurable “generic” experimental system. Main goals of the system design were the wide configurability of the system to different domains and situations as well as the ease of use. Thus the technical setup and execution of an experiment has to be possible with only relatively little preparatory work and should particularly not require any programming skills from the experimenters. Therefore, a specific experiment configuration and administration client was developed that allows the creation of experiments with the help of a graphical user interface. To be able to evaluate markets under realistic conditions, the experimental includes an interface that can be reduced in complexity but is still modeled to be similar to a real-life trading system.

Because the meet2trade system includes a configurable trading client that roughly spoken provides the functionality of a real-life trading system client (cf. section 3.4), the experimental functionalities were integrated directly into this client. Therefore the meet2trade client can be used in two ways – as a regular trading client and as a restricted experiment client. If the client is used in the restricted experiment mode, the look of the user interface, the possible user actions and the course of the experiment are controlled by the central meet2trade server instance. For example throughout an experiment a participant is able to trade on the system only as long as an experiment round is running. Furthermore he may solely interact with those markets the experiment administrator has explicitly made accessible for him. The supervision of the client is based on a built-in finite state machine. The different states (*wait for experiment begin, waiting screen, information screen, trading screen, experiment end*) are activated through server commands. Only during the state *trading screen* the trading functionality is available to the experiment participants.

As already mentioned, the meet2trade tool suite comprises a graphical experiment configuration and administration client. This tool facilitates the straightforward creation and execution of laboratory experiments. All parameters entered through this interface are converted into XML documents and transferred to the server, where they are stored in a database.

The most important part of the MES is represented by the central component which controls and manages all parts of the meet2trade system during an experiment. This component called experiment control component (ECC) provides the following functionalities:

- import, analysis and administration of the xml documents
- sequence control during the experiment
- supervision of the markets during the trading periods
- administration of participant data and accounting
- control of the participants’ client instances
- logging of all relevant experiment data

At the beginning of an experiment the ECC reads the xml-based experiment descriptions and analyzes them. After the experiment has been started, the sequence control starts monitoring the meet2trade system, regulates the starting and stopping time as well as the sequence of the different experiment sections and the activation and deactivation of the required markets.

When a new experiment round starts, the ECC sets the accounts (respectively the depots, when goods or stocks are traded) of all participants to their initial values, which are specified in the experiment description. Subsequently, the participant’s experiment clients are initialized. To avoid synchronization problems between the different participants (i.e. one experiment participant starts submitting offers while others still read the experiment instructions), all clients are centrally controlled. At the beginning of each round, the server loads the individual descriptions of the participant’s information and trading screens and sends them to the respective clients. Additionally, centrally controlled operations on the experiment clients are triggered through control commands like "show info screen" or "start trading round". After a round is over, the ECC calculates the monetary success of the participants by analyzing the trades they executed in that round and adds these values to the experiment accounts. The new account balances are subsequently displayed to the users via their experiment clients at the end of each round.

Figure 2 shows an overview over the MES architecture and its integration into meet2trade.

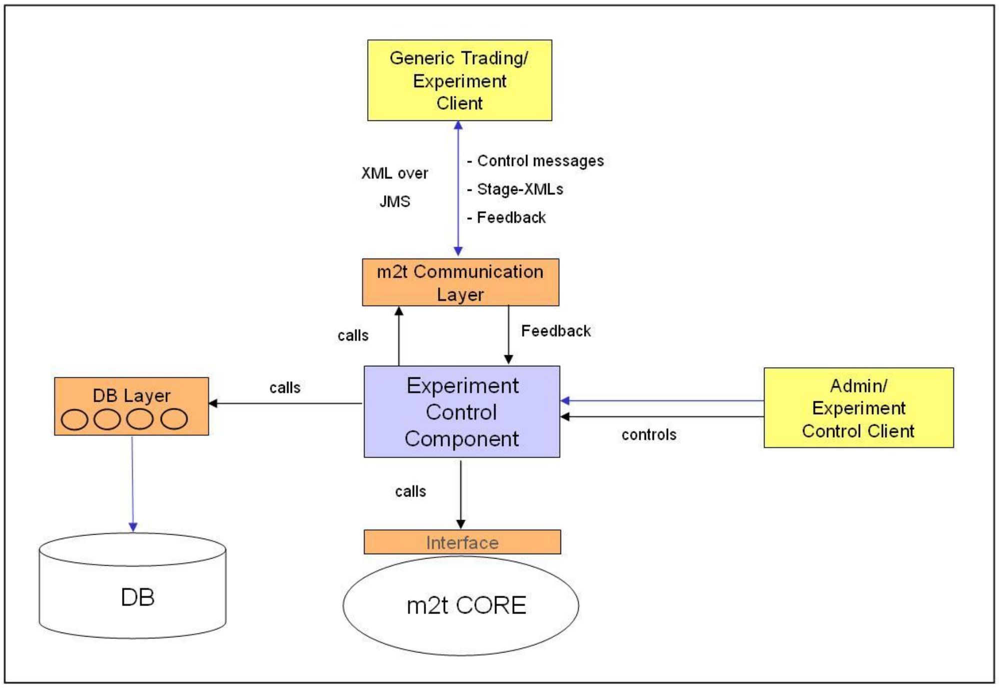

Figure 2: MES system architecture

4.3 Structure and Parameterization of Market Experiments

One of the central ideas during the implementation of the MES was to allow experimenters to setup experiments using the full range of supported market mechanisms, domains and settings without having to change the underlying program code. As for the implementation of the meet2trade runtime environment, also in this case a "generic" process was defined, which could be adapted to particular situations using a set of pre-defined parameters.

For the realization of this *generic experiment process*, an experiment has to be partitioned into organizational units. Consistent with the pertinent literature an experiment is therefore conducted in several *sessions*, where a particular session instance runs at a specific date with a specific group of participants. Sessions consist of several *treatments*, which are in turn divided into *stages*. Each stage represents one trading round. Figure 3 depicts the structure of the described experiment structure. Since in an laboratory experiment, each treatments is usually conducted with several distinct groups of participants to reach a sufficient number of observations, each treatment can be assigned to one or several sessions, without having to be entered again.

Each stage consists of two phases: in the first phase (pre-stage phase) an information screen is presented, which can contain an experiment description and / or the current account balance. In the second phase (trading phase) the actual trading takes place. Each session in an experiment follows the same basic sequence: the treatments contained in each session are executed in their pre-specified order. Within each treatment the pre-specified stages are executed one after another. At the beginning and at the end of every treatment, an optional questionnaire can be specified, e.g. to test the participants' understanding of the experiment case (quiz) or to collect further information like e.g. subjective impressions. An experiment session ends when all treatments have been executed successfully.

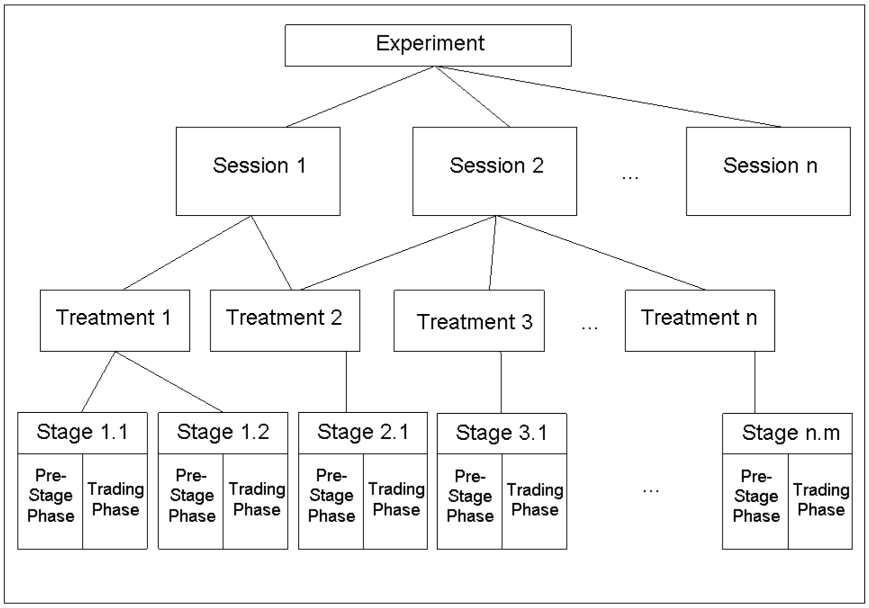

Figure 3: Experiment Structure

For each level in the outlined experiment hierarchy an individual range of parameters is available. Parameters for a certain level also hold in all of its sublevels, i.e. the session level contains a different set of parameters than the treatment level, but the session level parameters are valid for the treatments levels and all other sublevels as well. In order to avoid redundancy the parameters of different levels are non-overlapping thus each parameter is only available at one specific level. If a parameter has to be varied between two treatments (while it remains unchanged throughout the stages of a single treatment), it will be available only on the treatment level. If a differentiation is necessary on a lower level, say the stage level, the parameter will appear only on this level. For example, the post and pre questionnaires of a treatment are specified on the treatment level while the stage duration and market running times are specified on the stage level.

To be able to specify parameters in a more fine grained way, two more levels below the stage level had to be introduced, namely the market level and the user level. While the user level allows the user-specific variation of certain parameters like the user interface or their endowment of goods and currency, the markets level provides the possibility of overriding certain market parameters specified by the market designer for a part of the participants, stages or treatments. Figure 4 provides an excerpt of an experiment configuration showing several different experiment parameters on the different levels.

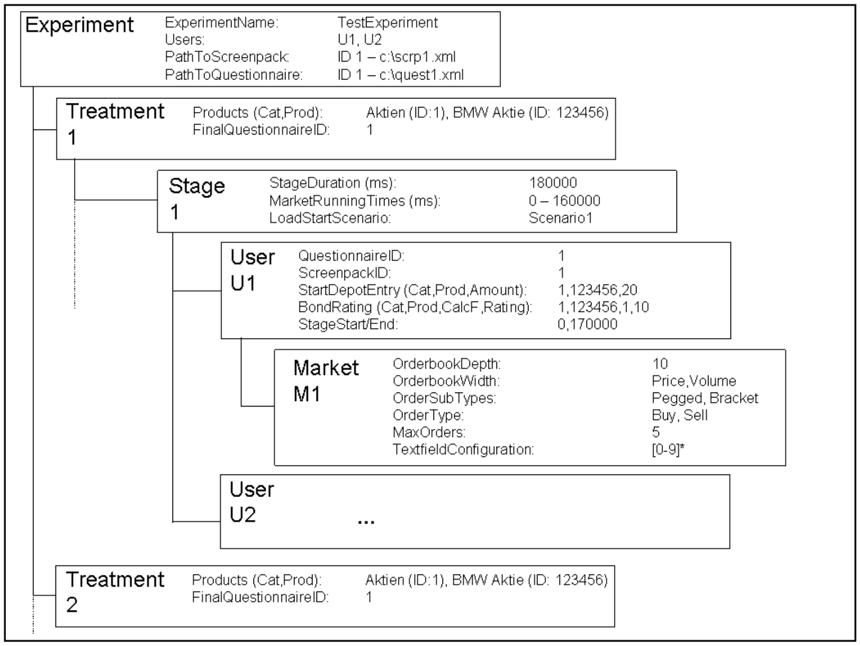

Figure 4: Excerpt of an Experiment Configuration

Since the overall number of adjustable experiment parameters is rather large and complex, they have to be specified in several XML documents, so a graphical administration tool was designed to support the experiment administrator in accomplishing this task. The interface displays the parameters in a tree-like view to represent the hierarchical structure of the experiment. By navigating through the experiment tree and by filling in the required parameters into the respective fields, all necessary values for the specification of an experiment can be entered easily. Besides the collection of experiment parameter values, this administration tool also offers functionality for manually adjusting (restricting) the trading client interface, for generating questionnaires, and for other administrative functionalities. Figure 5 displays a screenshot of the experiment administration tool.

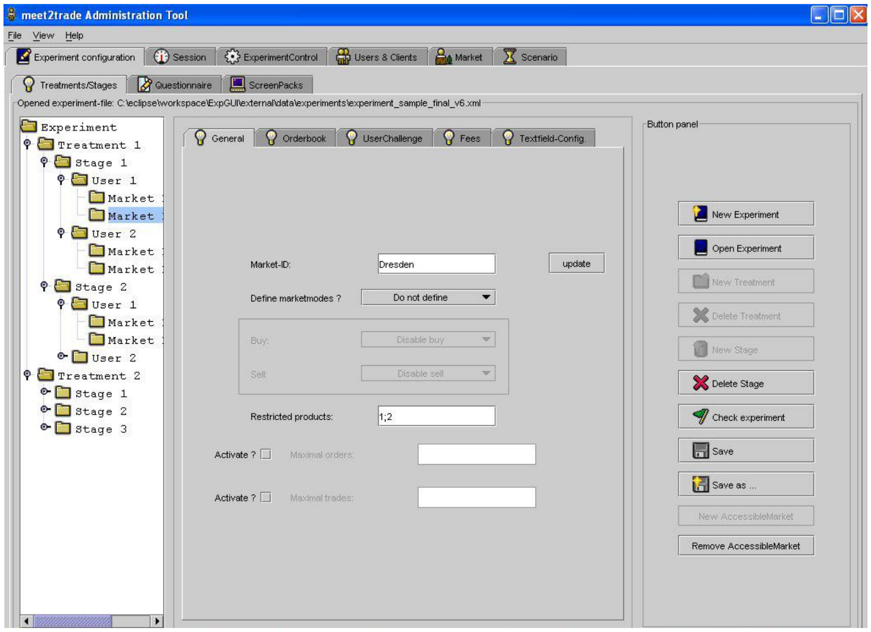

Figure 5: Screenshot of the experiment administration tool

4.4 Limitations

Even though the applicability of the meet2trade system for market research has been proven through several studies and experiments (see chapter 5), the chosen approach for the realization of a generic experimental system implies certain limitations. As described in sections 3.3 and 4.3, a generic process has been defined for the market and the experiment configuration, which is adapted to different domains and situations through a multitude of parameters. While this approach results in a fast and easy setup and configuration of new experiments without any coding, the limitation is naturally the number of parameters. Only a limited number of common parameters can be supported by the system and if a relevant parameter for a required functionality has not been included during the system design phase, then the system cannot provide that functionality. With almost 100 parameters available for specifying market mechanisms and another 100 parameters for the experiment configuration, a considerable design space is already provided to the users of meet2trade. Future experiments and studies conducted on and facilitated by the meet2trade tool suite will generate further knowledge on which additional functionalities might be desirable for auction and market mechanism related research. This knowledge can then be used as a basis for future, enhanced versions of this software.

5 Conclusion

Today's requirements for electronic markets are manifold, and innovative features that distinguish a market from those of its competitors have become an important success factor. At the same time the knowledge about interdependencies between market structure and market outcome is still limited to some few effects. To facilitate the evaluation of electronic markets and to provide a high degree of flexibility, we have presented the electronic trading platform meet2trade and the experimental system MES, which is based on this platform. meet2trade is a generic, flexible trading platform that facilitates the design, creation, testing and execution of auction based markets. Thus meet2trade is a proof-of-concept implementation of a tool suite capable of supporting the whole computer aided market engineering process.

The research field of Market Engineering analyzes all facets of electronic trading taking into account the economic, technical and juridical perspective. It involves the structured, customer-oriented process of designing, implementing and running electronic markets and considers the refinement and the quality assurance of these markets within their life cycles as well. The CAME process is defined as a life cycle which consists of the four phases (i) *Design*, (ii) *Configuration*, (iii) *Testing*, and (iv) *Operation*. The design phase requires knowledge about the mechanisms and how these mechanisms work. A knowledge database assists the market designer to guide and support him during the design phase. Using the *market modeling language* a market can be designed, created and set-up on the meet2trade platform. The *MML* supports the *configuration* of electronic markets as well as the *combination* of markets. Hence, it is a powerful tool to support the designer in creating auctions ranging from single-sided over double-sided to even multi-attribute auctions. The performance of markets can be *tested* in laboratory experiments and simulations under controlled conditions. Both methodologies are supported by the meet2trade tool suite through MES and AMASE, two subsystems linked to the core platform in order conduct agent-based and / or lab experiments to assess the performance of previously designed markets. Results from these tests, e.g. impacts on human behavior or market outcome, are used to redesign the trading mechanism and can be stored within knowledge-based system.

So far meet2trade was tested and used in various use cases and experimental studies. For sports events like the *Tour de France 2004* the *Olympic Games 2004* or the *FIFA world championships 2006* prediction markets based on auction mechanisms were configured and implemented allowing market participants to trade stocks thus “betting” on the occurrence of future events. Furthermore, meet2trade was used in several lectures to give students a practical insight into the operation of real world auction systems. Besides these activities, several simulation studies in the financial and other sectors have been carried out, e.g. on innovative order types (Kunzelmann and Mäkiö 2005), on bundle trading (Grunenberg and Weinhardt 2004) and on adaptive bidding in single-sided auction under uncertainty (van Dinther 2006). In addition to the simulations, meet2trade and in particular MES was used for a variety of experimental studies. In one study the influence of a first-bidder discount on the outcome of second price single-sided auctions was examined with the help of a laboratory experiment (see Weber 2006). Another application area for meet2trade and the MES is an Australian study of emissions trading (cf. Stößer 2006). Only recently MES was used within the NorA (Negotiations or Auctions) project at the Concordia University in Montreal, Canada to carry out a comparison study between market mechanisms (negotiations and auctions) and trading systems (meet2trade and Invite) (Kolitz and Neumann 2007).

However, there are many other research question for further studies. Thus, due to its flexibility and reusability the platform has a huge potential for innovative market design in various application domains. The platform meet2trade is a powerful tool facilitating research in the field of electronic ME.

References

Arthur, W., B. Holland, J. H., LeBaron, B., Palmer, R., and Tayler, P. (1996). “Asset pricing under endogenous expectations in an artificial stock market”, Technical report, Santa Fe Institute

Ayres, I. and Cramton, P. (1996). “Deficit reduction through diversity: How affirmative action at the fcc increased auction competition”, *Stanford Law Review* 48, pp. 761-815.

Bosman, R. and Riedl, A. (2003). Emotions and Economic Shocks in a First-Price Auction: An Experimental Study. Tinbergen Institute Discussion Paper No. 2003-056

Cramton, P. (2004). “Simultaneous ascending auction”, Technical report

Czernohous, C. (2005). “Simulation for evaluating electronic markets - an agent-based environment”, In *Proceedings of the 2005 International Symposium on Applications and the Internet (SAINT)*. IEEE

Czernohous, C., Kolitz, K., Mäkiö, J., Weber, I., and Weinhardt, C. (2003). “Integrating electronic market models - problems and solutions of parallel markets”, In U. Lechner (Ed.), *10th Research Symposium on Emerging Electronic Markets (RSEEM 2003), E-Marketplaces - Between Market and Cooperation*, Bremen, pp. 93-104.

van Dinther, C. (2006) “Adaptive Bidding in Single Sided Auctions under Uncertainty”, PhD thesis, Information Management and Systems (IISM), Universität Karlsruhe (TH), Karlsruhe, Germany

Grunenberg, M., Veit, D., and Weinhardt, C. (2004). “Elektronische Finanzmärkte und bundle trading”, In *Abstract Proceedings of the 66th Annual Conference of the Association of University Professors of Management*, Universität Graz, pp. 310-313.

Heyman, J. E., Orhun, Y., Ariely, D. (2004). Auction Fever: The Effect of Opponents and Quasi-Endowment on Product Valuations. Journal of Interactive Marketing Volume 18 / Number 4, 2004

Kersten, G., Chen E., et al. (2006). On Comparison of Mechanisms of Economic and Social Ex-changes: The Times Model. Dagstuhl Seminar Proceedings 06461 Negotiation and Market Engineering, Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany.

Kersten, G. E., Strecker, S. E., and Law, K. P. (2004). “Protocols for electronic negotiation systems: Theoretical foundations and design issues”, In K. Bauknecht, M. Bichler, and B. Pröll (Eds.), *E-Commerce and Web Technologies, 5th International Conference EC-Web 2004*, Berlin, pp. 106-115.

Kersten, G. E., and Teich, J. (2000). “Are all e-commerce negotiations auctions?”, *In Proceedings of the Fourth International Conference on the Design of Cooperative Systems,* Sophia-Antipolis, France

Klemperer, P. (2002). *"How (Not) to Run Auctions: the European 3G Telecom Auctions."* European Economic Review 46(4-5), pp. 829-845.

Kolitz, K. and Neumann, D. (2007). “The Impact of Information System Design on eMarketplaces – An Experimental Analysis“, Working Paper

Ku, G. (2000). "Auctions and Auction Fever: Explanations from Competitive Arousal and Framing." Kellogg Journal of Organization Behavior

Kunzelmann, M. and Mäkiö, J. (2005). “Pegged and bracket order as a success factor in stock exchange competition”, In *Proceedings of the 2nd Conference FinanceCom05*. IEEE

Mäkiö, J., and Weber, I. (2004). “Component-based specification and composition of market structures”, In M. Bichler (Ed.), *Coordination and Agent Technology in Value Networks*, Berlin, pp. 127-137.

Mäkiö, J., Weber, I., and Weinhardt, C. (2004). “Electronic negotiations - a generic approach with auction systems”, In K. Bauknecht, M. Bichler, and B. Pröll (Eds.), *Ecommerce and Web Technologies, 5th International Conference EC-Web 2004*, Berlin, pp. 135-143

Mäkiö, J. (2006) “The Parametric Design of Electronic Markets”, PhD thesis, Information Management and Systems (IISM), Universität Karlsruhe (TH), Karlsruhe, Germany

McAfee, R. P., and McMillan, J (1987). “Auctions and bidding”, *Journal of Economic Literature 25*, pp. 699-738

Miller, R. M. (2002). Paving Wall Street: Experimental Economics & the Quest for the Perfect Market

Mizuta, H., and Yamagata, Y. (2001). “Agent-based simulation and greenhouse gas emissions trading”, In B. Peters, J. Smith, D. Medeiros, and M. Rohrer (Eds.), *Proceedings of the 2001 Winter Simulation Conference*, pp. 535-540

Neumann, D. (2004). *Market Engineering - A Structured Design Process for Electronic Markets*, Ph. D. thesis, Universität Karlsruhe (TH), Karlsruhe, Germany

Neumann, D., Mäkiö, J. and Weinhardt, C. (2005). ”CAME - a toolset for configuring electronic Markets”, Proceedings of the ECIS 2005, Regensburg.

Ockenfels, A., and Roth, A. E. (2005). “Late bidding in second price internet auctions: Theory and evidence concerning different rules for ending an auction”. *Games and Economic Behavior forthcoming*

Reich, B. and Ben-Shaul, I. (1998). “A componentized architecture for dynamic electronic markets”, *SIGMOD Record* 27(4), pp. 40-47

Schlosser, A. E. (2003). Computers as Situational Cues: Implications for Consumer Product Cognitions and Attitudes. Journal of Consumer Psychology, 13 (1&2): 103-112

Schmid, B., and Lindemann, M. (1998). “Elements of a reference model for electronic markets”, In E. Sprague (Ed.), *31st Hawaiian International Conference on System Sciences (HICSS)*

Seifert, S. and Ehrhart, K.-M. (2005). “Design of the 3G spectrum auctions in the UK and Germany: An experimental investigation”, *German Economic Review* 6(2), pp. 229-248.

Simonsohn, U. and Ariely, D. (2005). Non-Rational Herding in Online Auctions. http://ssrn.com/abstract=722484.

Stößer, J. (2006) “Generic Market Platforms: Adaption and Application of meet2trade to Emissions Trading”, Thesis, Information Management and Systems (IISM), Universität Karlsruhe (TH), Karlsruhe, Germany

Ströbel, M., and Weinhardt, C. (2003). “The montreal taxonomy for electronic negotiations”, *Journal of Group Decision and Negotiation* 12(2), pp. 143-164.

Tesfatsion, L. (2002) “Agent-based computational economics: Growing economies from the bottom-up”, ISU Economics Working Paper No. 1. http://www.econ.iastate.edu/tesfatsi/

Weber, I. (2006) “Discounts in Auctions - Theoretical and Experimental Analysis”, PhD thesis, Information Management and Systems (IISM), Universität Karlsruhe (TH), Karlsruhe, Germany

Weinhardt, C., Holtmann, C., and Neumann, D. (2003). “Market engineering”, *Wirtschaftsinformatik* 45(6), pp. 635-640.

Wurman, P. R., Wellman, M. P., and Walsh, W. E. (1998). “The michigan internet auctionbot: A configurable auction server for human and software agents”, In K. P. Sycara and M.Wooldridge (Eds.), *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, New York, pp. 301-308

Wurman, P. R., Wellman, M. P., and Walsh, W. E. (2001). “A parametrization of the auction design space”, *Games and Economic Behavior* 35, 2001, pp. 304-338.