

Supporting the actors in an electronic market place

John Debenham

Faculty of Information Technology, University of Technology, Sydney
debenham@it.uts.edu.au

Abstract. An electronic market has been constructed in an on-going collaborative research project between a university and a software house. The way in which actors (buyers, sellers and others) use the market will be influenced by the information available to them, including information drawn from outside the immediate market environment. In this experiment, data mining and filtering techniques are used to distil both individual signals drawn from the markets and signals from the Internet into meaningful advice for the actors. The goal of this experiment is first to learn how actors will use the advice available to them, and second how the market will evolve through entrepreneurial intervention. In this electronic market a multiagent process management system is used to manage all market transactions including those that drive the market evolutionary process.

1. Introduction

The project described here addresses the problem of identifying timely information for e-markets with their rapid, pervasive and massive flows of data. This information is distilled from individual signals in the markets themselves and from signals observed on the unreliable, information-overloaded Internet. Distributed, concurrent, time-constrained data mining methods are managed using business process management technology to extract timely, reliable information from this unreliable environment. The long term goal of this project is to derive fundamental insight into how e-markets evolve. The perturbation of market equilibrium through entrepreneurial action is the essence of market evolution. Entrepreneurship relies both on intuition and information discovery. The term ‘entrepreneur’ is used here in its technical sense [1]. This long term goal includes the provision of timely information to support the market evolutionary process.

A subset of this project is substantially complete; it is called the *basic system* and is described in Sec. 2. The goal of this subset is to identify timely information for traders in an e-market. The traders are the buyers and sellers. This basic system does not address the question of market evolution. The basic system is constructed in two parts:

- the e-market
- the actors’ assistant

The e-market has been constructed by Bullant Australasia Pty Ltd—an Australian software house with a strong interest in business-to-business (B2B) e-business [www.bullant.com]. The e-market is part of their on-going research effort in this area. It has been constructed using Bullant’s proprietary software development tools. The e-market was designed by the author. The actors’ assistant is being constructed in the

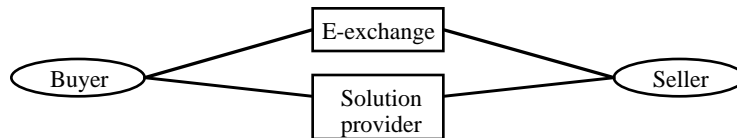


Fig 1. The four actor classes in the basic e-market system

Faculty of Information Technology at the University of Technology, Sydney. It is funded by two Australian Research Council Grants; one awarded to the author, and one awarded to Dr Simeon Simoff.

One feature of the whole project is that every transaction is treated as a business process and is managed by a process management system as described in Sec. 3. In other words, the process management system makes the whole thing work. In the basic system these transactions include simple market transactions such as “buy” and “sell” as well as transactions that assist the actors in buying and selling. For example, “get me information on the financial state of the Sydney Steel Co. by 4.00pm”. The process management system is based on a robust multiagent architecture. The use of multiagent systems is justified first by the distributed nature of e-business, and second by the critical nature of the transactions involved. This second reason means that the system should be able to manage a transaction (eg: buy ten tons of steel by next Tuesday at the best available price) reliably in an essentially unreliable environment. The environment may be unreliable due to the unreliability of the network and components in it, or due to the unreliability of players—for example, a seller may simply renege on a deal.

The overall goal of this project is to investigate the evolution of e-markets. The investigation of e-market evolution will commence at the end of 2001. The overall design framework is already complete and a brief description of it is given in Sec. 4.

2. The basic system

The basic system consists of the e-market and the actors’ assistant. The goal of the basic system is to identify timely information for traders in the e-market. Trading in the e-market either takes place through the ‘e-exchange’ or through a ‘solution provider’. The *e-exchange* is an open exchange where goods and services are offered on a fixed price basis or by some competitive mechanism. The *solution providers* assist in the negotiation of contracts and facilitate the development of business relationships between buyers and sellers. So the actor classes in the basic system are buyer, seller, e-exchange and solution provider. The four actor classes are shown in Fig. 1. Before describing the system itself a justification is given of this choice of four actor classes.

2.1 Actor classes in the basic system

For some while there has been optimism in the role of agents in electronic commerce. “During this next-generation of agent-mediated electronic commerce,... Agents will strategically form and reform coalitions to bid on contracts and leverage economies of scale..... It is in this third-generation of agent-mediated electronic commerce where companies will be at their most agile and markets will approach perfect efficiency.” [2].

There is a wealth of material, developed principally by micro-economists, on the behaviour of rational economic agents. The value of that work in describing the behaviour of human agents is limited in part by the inability of humans to necessarily behave in an (economically) rational way, particularly when their (computational) resources are limited. That work provides a firm foundation for describing the behaviour of rational, intelligent software agents whose resource bounds are known, but more work has to be done [3]. Further, new market mechanisms that may be particularly well-suited to markets populated by software agents is now an established area of research [4] [5]. Most electronic business to date has centred on on-line exchanges in which a single issue, usually price, is negotiated through the application of traditional auction-based market mechanisms. Systems for multi-issue negotiation are also being developed [6], also IBM's Silkroad project [7]. The efficient management of multi-issue negotiation towards a possible solution when new issues may be introduced as the negotiation progresses remains a complex problem [8].

Given the optimism in the future of agents in electronic commerce and the body of theoretical work describing the behaviour of rational agents, it is perhaps surprising that the basic structure of the emerging e-business world is far from clear. The majority of Internet e-exchanges are floundering, and it appears that few will survive [9]. There are indications that exchanges may even charge a negative commission to gain business and so too market intelligence [op. cit.]. For example, the Knight Trading Group currently pays on-line brokers for their orders. The rationale for negative commissions is discussed in Sec. 4. One reason for the recent failure of e-exchanges is that the process of competitive bidding to obtain the lowest possible price is not compatible with the development of buyer-seller relations. The preoccupation with a single issue, namely price, can overshadow other attributes such as quality, reliability, availability and customisation. A second reason for the failure Internet e-exchanges is that they deliver little benefit to the seller—few suppliers want to engage in a ruthless bidding war [op. cit.]. The future of electronic commerce must include the negotiation of complex transactions and the development of long-term relationships between buyer and seller as well as the e-exchanges. Support for these complex transactions and relationships is provided here by *solution providers*.

A considerable amount of work has been published on the comparative virtues of open market e-exchanges and solution providers that facilitate direct negotiation. For example, [10] argues that for privately informed traders the 'weak' trader types will systematically migrate from direct negotiations to competitive open markets. Also, for example, see [11] who compare the virtues of auctions and negotiation. Those results are derived in a supply/demand-bounded world into which signals may flow. These signals may be received by one or more of the agents in that world, and so may cause those agents to revise their valuation of the matter at hand. No attempt is made to accommodate measures of the intrinsic validity of those signals, measures of the significance of those signals to the matter at hand, or measures of interdependencies between those signals. That is, those models ignore the general knowledge that is required to make good business decisions. That general knowledge may be drawn from outside the context of the market place. So those results do not necessarily mean that e-exchanges will be preferred over negotiated contracts in the real world. For example, the negotiation of a long term contract for some commodity may be based on an

individual's 'hunch' that future political instability in a foreign country will (partly) cause the price of that commodity to rise.

The issue addressed here is limited to single issue negotiation either in an e-exchange or through a solution provider. This problem is not trivial. For example, if a company has a regular requirement for so-many tons of steel a month then will it be better off for the next 24 months *either* making offers in an e-exchange on a monthly basis *or* negotiating a 24-month contract?

2.2 The e-market

The construction of experimental e-markets is an active area of research. For example, [12] describes work done at IBM's Institute for Advanced Commerce. The basic e-market described here has been designed in the context of the complete system described briefly in Sec. 4. There are two functional components in the *basic e-market*: the e-exchange and a solution provider. The *solution provider* is 'minimal' and simply provides a conduit between buyer and seller through which long term contracts are negotiated. The *solution provider* in its present form does not give third-party support to the negotiation process.

An e-exchange is created for a fixed duration. An *e-exchange* is a virtual space in which a variety of market-type *activities* can take place at specified times. The time is determined by the e-exchange *clock*. Each activity is advertised on a notice *board* which shows the start and stop time for that activity as well as what the activity is and the *regulations* that apply to players who wish to participate in it. A human player works through a PC (or similar) by interacting with a *user agent* which communicates with a *proxy agent* or a solution provider situated in the e-market. The inter-agent communication is discussed in Sec 3. The user agents may be 'dumb', or 'smart' being programmed by the user to make decisions. Each activity has an *activity manager* that ensures that the regulations of that activity are complied with.

When an e-exchange is created, a specification is made of the e-exchange *rules*. These rules will state who is permitted to enter the e-exchange and the roles that they are permitted to play in the e-exchange. These rules are enforced by an *e-exchange manager*. For example, can any player create a sale activity (which could be some sort of auction), or, can any player enter the e-exchange by offering some service, such as advice on what to buy, or by offering 'package deals' of goods derived from different suppliers? A high-level view of the e-market is shown in Fig. 2.

The activities in the basic e-market are limited to opportunities to buy and sell goods. The regulations for this limited class of activities are called *market mechanisms* [5]. The subject of a negotiation is a *good*, buyers make *bids*, sellers make *asks*. Designing market mechanisms is an active area of research. For example, see optimal auctions [4]. One important feature of a mechanism is the 'optimal' strategy that a player should use, and whether that strategy is "truth revealing" [11]. Mechanisms can be for single-unit (ie a single good) or multi-unit (ie a number of identical goods). Any single-unit mechanism may be extended to a multi-unit mechanism by using it to establish a price for a good, permitting the 'winner' to take as many goods as required at that price (and maybe then permitting other buyers to take further goods at that price),

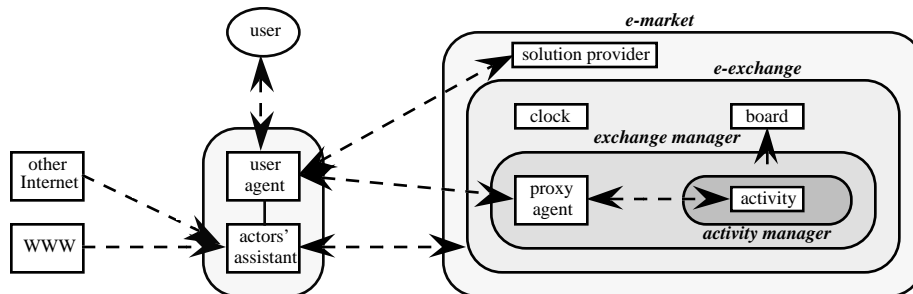


Fig. 2. High-level model of the e-market and user

then putting the remaining goods up for sale to determine another price, and so on, until the process terminates.

A market mechanism is specified by the player who creates an activity and is displayed on the notice board. The market mechanism will determine:

- what is the *starting time* and *closing time* of the activity.
- what is the *subject* of the activity; eg. if the activity is to sell goods then what are the goods for sale.
- the *commission*, if any, payable on any successful transaction—this may be negative, may be a flat fee and/or a percentage of the value of the trade.
- the *reserve price*, if any, below which the good is not for sale—this is usually specified by the seller.
- what *market information* is available—eg. some auctions are ‘open cry’ and some are ‘sealed bid’.
- under what circumstances can a player participate? For example, can a player ‘sell/buy short’. For example, is a player required to make a *deposit* or pay an *entry fee*?
- if a player can ‘sell/buy short’ then what penalty applies if they are unable to deliver.
- what a player has to do to participate—for example, in an auction a buyer has to submit a bid, in which case what does a bid consist of?
- how many bids/asks can a player submit?
- having submitted a bid/ask can a player withdraw or modify that bid/ask?
- what has to happen for somebody to ‘win’?
- if at least one person ‘wins’ then: When does the activity clear? How are the goods distributed amongst the ‘winners’? Including, for example, what happens if there are two equal bids/asks in a sealed-bid multi-unit auction. What is the price for the goods as distributed amongst the ‘winners’? What happens to the unfilled bid(s) of the winner(s), and, if it is a double auction, the ask(s) of the seller(s)?

To enable the activity manager to do its job, each proxy agent reveals in confidence the assets that the proxy agent possesses to the activity manager. The activity manager may ‘ear mark’ some amount of a buying proxy agent’s money if the market mechanism requires that that amount must be set aside to support a bid, or if a buyer’s deposit is required. Each player may withdraw assets from the proxy agent and so from the



Fig. 3. User's screen for an e-exchange that is hooked to the Sydney Stock Exchange.

e-market. Fig. 3 shows a user's screen for an e-exchange that is hooked to the Sydney Stock Exchange.

2.3 The actors' assistant

In 'real' problems, a decision to use an e-exchange or a solution provider will be made on the basis of general knowledge that is external to the e-market place. Having decided to use either an e-exchange or a solution provider, the negotiation strategy used will also depend on general knowledge. Such general knowledge will typically be broadly based and beyond the capacity of modern AI systems whilst remaining within reasonable cost bounds.

E-markets reside on the Internet alongside the vast resources of the World Wide Web. In the experiments described here, the general knowledge available is restricted to that which can be gleaned from the e-markets themselves and that which can be extracted from the Internet in general—including the World Wide Web. The actors' assistant is a workbench that provides a suite of tools to *assist* a buyer or seller in the e-market. [Sec. 4 describes how the actors' assistant will assist other market actors as well.] The actors' assistant does *not* attempt to replace buyers and sellers. For example, there is no attempt to automate 'speculation' in any sense. Web-mining tools assist the players in the market to make informed decisions. One of the issues in operating in an e-market place is coping with the rapidly-changing signals in it. These signals include: product and assortment attributes (if the site offers multiple products), promotions shown, visit attributes (sequences within the site, counts, click-streams) and business agent attributes. Combinations of these signals may be vital information to an actor. A new generation of data analysis and supporting techniques—collectively labelled as *data mining* methods—are now applied to stock market analysis, predictions and other financial and

market analysis applications [13]. The application of data mining methods in e-business to date has predominantly been within the B2C framework, where data is mined at an on-line business site, resulting in the derivation of various behavioural metrics of site visitors and customers.

The basic steps in providing assistance to actors in this project are:

- identifying potentially relevant signals;
- evaluating the reliability of those signals;
- estimating the significance of those signals to the matter at hand;
- combining a (possibly large) number of signals into coherent advice, and
- providing digestible explanations for the advice given.

For example, the identification of potentially relevant signals includes scanning news feeds. In the first pass the text is stripped of any HTML or other formatting comments—this is *preprocessing*. Then keyword matches are performed by *scanners*. Provided the number of ‘hits’ has not been too large, these two steps alone produce useful information. For example, in response to a query “find anything you can on Sydney Steel in the international press during the past 24 hours’ is represented simply as:

“...takeover bid for Sydney Steel” (New York Times)

“Sydney Steel board announce...” (CNN)

Here the search phrase is Sydney Steel, and the references quoted give n (=2 here) significant additional words in the sentence in which the search phrase was found. The underlining in these examples are clickable and take the user to the source material. Or, if the source material has been pre-fetched, to a small window containing the immediate surrounding text. This is not ‘rocket science’ but it does show that useful information may be presented simply by employing existing news bots through a common interface. Bots that have been built into the system include: News Hub [www.newshub.com], NewsTrawler [www.newstrawler.com] and CompanySleuth [www.companysleuth.com]. Following the preprocessing and scanning steps, an *assessment* is made of the overall reliability of the source. At present this is simply a measure of the overall reputation that the source has for accuracy. In addition, *watchers* detect changes to material on the Web. Here URLyWarning [www.urlywarning.com] and other watcher bots are used to identify pages in which designated information *may* have changed; they may be used to trigger a detailed search of other pages.

The estimation of the significance of a signal to a matter at hand is complicated by the fact that one person may place more faith in the relevance of a particular signal than others. So this estimation can only be performed on a personal basis. This work does *not*, for example, attempt to use a signal to predict whether the US dollar will rise against the UK pound. What it *does* attempt to do is to predict the value that an actor will place on a signal [14]. So the feedback here is provided by the user in the form of a rating of the material used. A five point scale runs from ‘totally useless’ to ‘very useful’. Having identified the signals that a user has faith in, “classical” data mining methods [15] are then applied to combine these signals into succinct advice again using a five point scale. This feedback is used to ‘tweak’ the weights in Bayesian networks and as feedback to neural networks [16]. Bayesian networks are preferred when some confidence can be placed in a set of initial values for the weights. The system is able to

raise an alarm automatically and quickly when a pre-specified compound event occurs such as: four members of the board of our principal supplier “Good Co” have resigned, the share price has dropped unexpectedly and there are rumours that our previous supplier “Bad Co” is taking over “Good Co”.

The actors’ assistant integrates two different approaches in data mining — the data driven and the hypothesis-driven approach. In the data-driven approach the assistant is just “absorbing” the information discovered by the scanners. It only specifies broad parameters to constrain the material scanned. For example, in the text analysis of the news files a text miner observes the frequencies of word occurrences and co-occurrences that appear to be relevant to a keyword such as ‘steel prices’. The result of this process is an initial representative vocabulary for that news file. In the hypothesis-driven approach, the actors’ assistant specifies precisely what it is looking for, for example, it formulates a hypothesis that a fall in the price of steel is likely within a month. The combination of data-driven and hypothesis driven approaches aims to provide a mechanism for meeting tight time constraints. Managing and synchronising the actors’ assistant is handled by process management plans in the user agents. For example, a request is made for the best information on the Sydney Steel Co to be delivered by 4.00pm. This request triggers a business process. Things can go wrong with this process, for example a server may be down, in which case the process management plans activate less-preferred but nevertheless useful ways of obtaining the required information by the required time.

3. Managing e-market transactions with a multiagent system

A multiagent business process management system manages *all* e-market transactions and so it makes the e-market operate.

3.1 The process management problem

Fig 2. may give the false impression that all the process management system does is to support communication between the user agents and their corresponding proxy agents. All transactions are managed as business processes, including a simple ‘buy order’, and a complex request for information placed with an actor’s assistant. For example, a simple ‘buy order’ placed by a user agent may cause its proxy agent to place a bid in an activity which may then provoke interaction with a number of user agents who are trying to sell the appropriate goods, and so on. Alternatively, a user agent may seek out other agents through a solution provider. In either case, any buy or sell order will be partly responsible for triggering a reaction along the entire supply (or value) chain. Building e-business process management systems is business process reengineering on a massive scale, it often named *industry process reengineering* [17]. This can lead to considerable problems unless there is an agreed basis for transacting business. The majority of market transactions are constrained by time (“I need it before Tuesday”), or more complex constraints (“I only need the engine if I also have a chassis and as long as the total cost is less than..). The majority of transactions are *critical* in that they must be dealt with and can’t be forgotten or mislaid. Or at least it is an awful nuisance if they

are. So this means that a system for managing them is required that can handle complex constraints and that attempts to prevent process failure.

E-market processes will typically be *goal-directed* in the sense that it may be known *what* goals have to be achieved, but not necessarily *how* to achieve those goals today. A goal-directed process may be modelled as a (possibly conditional) sequence of goals. Alternatively a process may be *emergent* in the sense that the person who triggers the process may not have any particular goal in mind and may be on a ‘fishing expedition’ [18]—this is particularly common in the entrepreneurial processes described briefly in Sec. 4. There has been little work on the management of emergent processes [19]. There a multiagent process management system is described that is based on a three-layer, BDI, hybrid architecture. That system ‘works with’ the user as emergent processes unfold. It also manages goal-directed processes in a fairly conventional way using single-entry quadruple-exit plans that give almost-failure-proof operation. Those plans can represent constraints of the type referred to above, and so it is a candidate for managing the operation of the system described in Sec. 2, and elaborated in Sec. 4.

One tension in e-business is that the transactions tend to be critical and the environment in which they are being processed is inherently unreliable. On one level sellers and buyers may renege on a deal, or may take an unpredictable amount of time to come to a conclusion. On another level the whole e-business environment, including the computer systems on it and the Internet itself, can not be relied on to perform on time, and, on occasions, may not operate at all. So this means that the system which manages the transactions must be able to adapt intelligently as the performance characteristics of the various parts of its environment are more or less satisfactory and more or less predictable. A more subtle complicating factor is that whereas much valuable information may be available free of charge on the World Wide Web it is not necessarily current. For example, the Sydney Stock Exchange publishes its quoted prices free of charge but with a time delay of several minutes—current information costs money. So, while working in an unreliable environment, the system for managing processes should balance time and cost tradeoffs in sub-processes whilst attempting to keep the whole process on time and within budget, and whilst delivering results of an acceptable standard.

3.2 The multiagent system

Multiagent technology is an attractive basis for industry process re-engineering [20] [21]. A multiagent system consists of autonomous components that negotiate with one another. The scalability issue of industry process reengineering is “solved”—in theory—by establishing a common understanding for inter-agent communication and interaction. Standard XML-based ontologies will enable data to be communicated freely [22] but much work has yet to be done on standards for communicating expertise [23]. Results in ontological analysis and engineering [24] [23] is a potential source for formal communication languages which supports information exchange between the actors in an e-market place. Systems such as CommerceNet’s Eco [www.commerce.net] and Rosettanet [www.rosettanet.org] are attempting to establish common languages and frameworks for business transactions and negotiations. Specifying an agent interaction

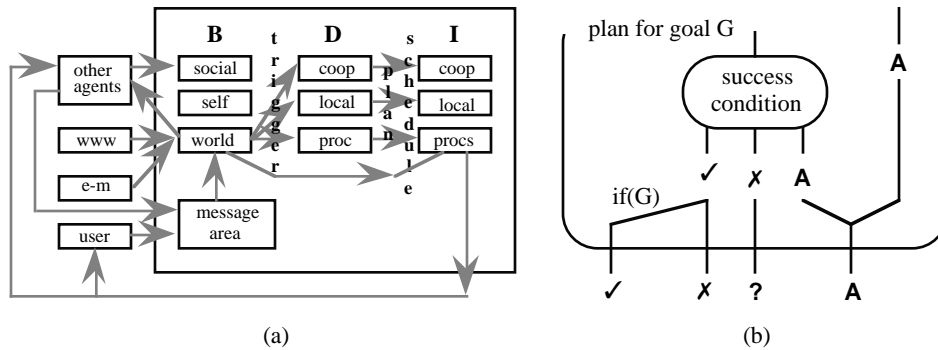


Fig 4. (a) conceptual architecture, (b) the four plan exits

protocol is complex as it in effect specifies the common understanding of the basis on which the whole system will operate.

A variety of architectures have been described for autonomous agents. A fundamental distinction is the extent to which an architecture exhibits deliberative (feed forward, planning) reasoning and reactive (feed back) reasoning. If an agent architecture combines these two forms of reasoning it is a *hybrid architecture*. One well reported class of hybrid architectures is the three-layer, BDI agent architectures. One member of this class is the INTERRAP architecture [25], which has its origins in the work of [26]. A multiagent system to manage “goal-driven” processes is described in [19]. In that system each human user is assisted by an agent which is based on a generic three-layer, BDI hybrid agent architecture similar to the INTERRAP architecture. That system has been extended to support emergent processes and so to support and the full range of industry processes. That conceptual architecture is adapted slightly for use here; see Fig 4(a). Each agent receives messages from other agents (and, if it is a personal agent, from its user) in its message area. The world beliefs are derived from reading messages, observing the e-market and from the World Wide Web (as accessed by an actor’s assistant).

Deliberative reasoning is effected by the non-deterministic procedure: “on the basis of current *beliefs*—identify the current *options*, on the basis of current options and existing commitments—select the current commitments (called the agent’s *goals* or *desires*), for each newly-committed goal choose a *plan* for that goal, from the selected plans choose a consistent set of things to do next (called the agent’s *intentions*)”. A *plan* for a goal is a conditional sequence of sub-goals that may include iterative or recursive structures. If the current options do not include a current commitment then that commitment is dropped. In outline, the reactive reasoning mechanism employs triggers that observe the agent’s beliefs and are ‘hot wired’ back to the procedural intentions. If those triggers fire then they take precedence over the agent’s deliberative reasoning. The environment is intrinsically unreliable. In particular plans can not necessarily be relied upon to achieve their goal. So at the end of every plan there is a *success condition* which tests whether that plan’s goal has been achieved; see Fig 4(b). That success condition is itself a procedure which can succeed (✓), fail (✗) or be aborted (A). So this leads to each plan having four possible exits: success (✓), failure (✗),

aborted (**A**) and unknown (?). In practice these four exists do not necessarily have to lead to different sub-goals, and so the growth in the size of plan with depth is not quite as bad as could be expected.

KQML (Knowledge Query and Manipulation Language) is used for inter-agent communication [27]. Each process agent has a *message area*. If agent A wishes to tell something to agent B then it does so by posting a message to agent B's message area. Each agent has a *message manager* whose role is to look after that agent's message area. Each message contains an instruction for the message manager. Two such instructions are:

- post message and remove on condition—the sender is asking the receiving agent's message manager to display the message in the receiving agent's message area *until* the stated condition is satisfied, and
- remove message—the sender is asking the receiving agent's message manager to remove an existing message from the receiving agent's message area.

The system operates in an environment whose performance and reliability are unreliable and unpredictable. Further, choices may have to be made that balance reliability with cost. To apply the deliberative reasoning procedure requires a mechanism for *identifying* options, for *selecting* goals, for *choosing* plans and for *scheduling* intentions. A plan may perform well or badly. The process management system takes account of the "process knowledge" and the "performance knowledge". *Process knowledge* is the wisdom that has been accumulated, particularly that which is relevant to the process instance at hand. *Performance knowledge* is knowledge of how effective agents, people, systems, methods and plans are at achieving various things. A plan's *performance* is defined in terms of: the likelihood that the plan will succeed, the expected cost and time to execute the plan, the expected value added to the process by the plan, or some combination of these measures. If each agent knows how well the choices that it has made have performed in the past then it can be expected to make decisions reasonably well as long as plan performance remains reasonably stable. One mechanism for achieving this form of adaptivity is reinforcement learning. An alternative approach based on probability is described in [19]. In addition, an agent may know things about the system environment, and may have some idea of the *reason why* one choice lead to failure. An agent's belief in these reasons may result from communication with other agents. Such beliefs may be used to revise the "historical" estimates to give an *informed* estimate of plan performance that takes into account the *reasons why* a plan behaved the way that it did [op. cit.].

4. Future work: investigating e-market evolution

E-markets, with their rapid, extensive, pervasive and visible flows of information provide an ideal context in which (a) to study market evolution in a context that is undergoing rapid evolution and innovation, and (b) to build systems that assist entrepreneurial intervention in actual market situations. The term 'entrepreneur' is used here in its technical sense [1]. Entrepreneurial transactions are typically more complex than the basic e-market transactions described in Sec. 2. Work on this part of the project will commence in the latter half of 2001, and the outline design is complete.

One entrepreneurial transaction is a request for information. For example, “look for Sydney-based manufacturing companies with a high asset backing that have been subject of at least one failed takeover bid in the past two years”. Such a request triggers an instance of an *information process* in the user agent that will invoke information discovery and data mining methods, and perhaps hand-crafted analytical tools, in that agent’s actors’ assistant. These methods and tools are applied to either the e-market data or to Internet data. So there are four basic problems here. First, determining where to access the individual signals. Second, assessing some measure of confidence in the validity of those signals. Third combining those signals—which may be invalid and which may have been observed at different times and different places—into reliable advice. Fourth to do all this within tight time and cost constraints. Another entrepreneurial transaction is *entrepreneurial intervention* itself. For example, “I am offering the following unique combination of products and services with a novel payment structure.....”. Such a transaction also triggers a business process that sets up the required structures within the e-market.

In the full e-market system, additional actor classes are introduced to support market evolution. *E-speculators* take short term positions in the e-exchange. This introduces the possibility of negative commission charges for others using an e-exchange. *Sell-side asset exchanges* exchange or share assets between sellers. *Content aggregators*, who act as forward aggregators, coordinate and package goods and services from various sellers. *Specialist originators*, who act as reverse aggregators, coordinate and package orders for goods and services from various buyers. The specialist originators, content aggregators and e-speculators represent their presence in the e-exchange by creating activities. This finally gives justification to the use of the term ‘activity’ to describe what happens in the e-exchange rather than ‘sale’ which might be more appropriate to the limited range of transactions in the basic system described in Sec. 2. The eight actor classes are illustrated in Fig. 5. They are a super-set of the actor classes in the basic system.

The full system will be used to study market evolution. E-markets are a fruitful arena for hunting entrepreneurial opportunities—witness the proliferation of new market forms and players [28], [9] and the characterisation of the new network economy as an “opportunity factory”. The rich flows of information in and around e-markets can be tapped to facilitate the discovery of potentially profitable opportunities and thereby stimulate the innovation and evolution process.

There are valuable studies on the cognitive processes and factors driving alertness to opportunities [29]. There is also a wealth of established work in economic theory, that provides a basis for work in e-markets, including: the theory of auctions, theory of bargaining, and theory of contracts. [8] presents mechanisms for negotiation between computational agents. That work describes tactics for the manipulation of the utility of deals, trade-off mechanisms that manipulate the value, rather than the overall utility, of an offer, and manipulation mechanisms that add and remove issues from the negotiation set. Much has to be done before this established body of work may form a practical basis for an investigation into electronic market evolution [3]. Little is known on how entrepreneurs will operate in electronic market places, although the capacity of the vast amount of information that will reside in those market places, and on the Internet generally, to assist the market evolution process is self-evident. The long term aim of this project is to understand the e-market evolutionary process.

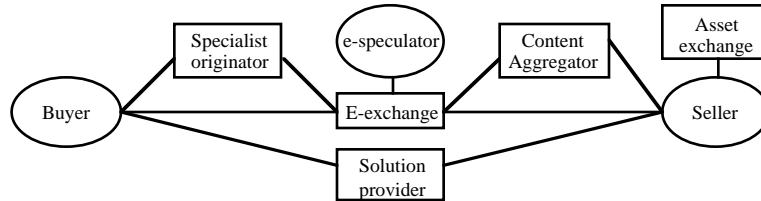


Fig 5. The eight actor classes in the full e-market system

5. Conclusion

One of the innovations in this project is the development of a coherent environment for e-market places, a comprehensive set of actor classes and the use of a powerful multiagent process management system to make the whole thing work. The use of a powerful business process management system to drive all the electronic market transactions unifies the whole market operation. The development of computational models of the basic market transactions, deploying those models in the e-market place, and including them as part of the building blocks for creating a complete e-market place provides a practical instrument for continued research and development in electronic markets.

Acknowledgment

The work described herein was completed whilst the author was a visitor at the CSIRO Joint Research Centre for Advanced Systems Engineering, Macquarie University, Sydney. The contribution to the work by the members of that Centre is gratefully acknowledged.

References

- [1] Israel M. Kirzner Entrepreneurial Discovery and the Competitive Market Process: An Austrian Approach” Journal of Economic Literature XXXV (March) 1997 60-85.
- [2] R. Guttman, A. Moukas, and P. Maes. Agent-mediated Electronic Commerce: A Survey. Knowledge Engineering Review, June 1998.
- [3] Moshe Tennenholtz. Electronic Commerce: From Economic and Game-Theoretic Models to Working Protocols. Invited paper. Proceedings Sixteenth International Joint Conference on Artificial Intelligence, IJCAI’99, Stockholm, Sweden.
- [4] Milgrom, P. Auction Theory for Privatization. Cambridge Univ Press (2001).
- [5] Bichler, M. The Future of E-Commerce: Multi-Dimensional Market Mechanisms. Cambridge University Press (2001).
- [6] Sandholm, T. Agents in Electronic Commerce: Component Technologies for Automated Negotiation and Coalition Formation. Autonomous Agents and Multi-Agent Systems, 3(1), 73-96.
- [7] Ströbel, M. Design of Roles and Protocols for Electronic Negotiations. Electronic Commerce Research Journal, Special Issue on Market Design 2001.

- [8] Peyman Faratin. Automated Service Negotiation Between Autonomous Computational Agents. PhD dissertation, University of London (Dec 2000).
- [9] R. Wise & D. Morrison. Beyond the Exchange; The Future of B2B. Harvard Business review Nov-Dec 2000, pp86-96.
- [10] Neeman, Z. & Vulkan, N. Markets Versus Negotiations. The Hebrew University of Jerusalem Discussion Paper 239. (February 2001).
- [11] Bulow, J. & Klemperer, P. Auctions Versus Negotiations. American Economic Review, 1996.
- [12] Kumar, M. & Feldman, S.I. Business Negotiations on the Internet. Proceedings INET'98 Internet Summit, Geneva, July 21-24, 1998.
- [13] B. Kovalerchuk & E. Vityaev. Data Mining in Finance: Advances in Relational and Hybrid Methods. Kluwer, 2000.
- [14] J. Han, L.V.S. Lakshmanan & R.T. Ng. Constraint-based multidimensional data mining. IEEE Computer, 8, 46-50, 1999.
- [15] Han, J. & Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann (2000).
- [16] Chen, Z. Computational Intelligence for Decision Support. CRC Press, Boca Raton, 2000.
- [17] Feldman, S. Technology Trends and Drivers and a Vision of the Future of e-business. Proceedings 4th International Enterprise Distributed Object Computing Conference, September 25-28, 2000, Makuhari, Japan.
- [18] Fischer, L. (Ed). Workflow Handbook 2001. Future Strategies, 2000.
- [19] Debenham, J.K.. Supporting knowledge-driven processes in a multiagent process management system. Proceedings Twentieth International Conference on Knowledge Based Systems and Applied Artificial Intelligence, ES'2000: Research and Development in Intelligent Systems XVII, Cambridge UK, December 2000, pp273-286.
- [20] Jain, A.K., Aparicio, M. and Singh, M.P. "Agents for Process Coherence in Virtual Enterprises" in Communications of the ACM, Volume 42, No 3, March 1999, pp62—69.
- [21] Jennings, N.R., Faratin, P., Norman, T.J., O'Brien, P. & Odgers, B. Autonomous Agents for Business Process Management. Int. Journal of Applied Artificial Intelligence 14 (2) 145—189, 2000.
- [22] Robert Skinstad, R. "Business process integration through XML". In proceedings XML Europe 2000, Paris, 12-16 June 2000.
- [23] Guarino N., Masolo C., and Vetere G., OntoSeek: Content-Based Access to the Web, IEEE Intelligent Systems 14(3), May/June 1999, pp. 70-80
- [24] Uschold, M. and Gruninger, M.: 1996, Ontologies: principles, methods and applications. Knowledge Engineering Review, 11(2), 1996.
- [25] Müller, J.P. "The Design of Intelligent Agents" Springer-Verlag, 1996.
- [26] Rao, A.S. and Georgeff, M.P. "BDI Agents: From Theory to Practice", in proceedings First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA, pp 312—319.
- [27] Finin, F. Labrou, Y., and Mayfield, J. "KQML as an agent communication language." In Jeff Bradshaw (Ed.) Software Agents. MIT Press (1997).
- [28] Kaplan, Steven and Sawhney, Mohanbir. E-Hubs: The New B2B Marketplace. Harvard Business Review 78 May-June 2000 97-103.
- [29] Shane, Scott. Prior knowledge and the discovery of entrepreneurial opportunities. Organization Science 11 (July-August), 2000, 448-469.