

Civil agent societies: Tools for inventing open agent-mediated electronic marketplaces

Chrysanthos Dellarocas

Mark Klein

Sloan School of Management
Massachusetts Institute of Technology
Room E53-315, Cambridge, MA 02139
{dell, m_klein}@mit.edu

Abstract. In the emerging model of 21st century electronic commerce, a variety of open agent marketplaces will be competing with one another for participants. The most successful marketplaces will be those that provide the best “quality of service” guarantees (in terms of security, fairness, efficiency, etc.), while meeting such challenges as agent heterogeneity, limited trust, and potential for systemic dysfunctions. Civil human societies provide a useful model for designing the infrastructure needed to achieve these guarantees. Successful civil human societies build on well-designed “social contracts”, i.e. agreed-upon constraints on agent behavior made in exchange for quality of service assurances backed up by social institutions. Civil Agent Societies can be defined in an analogous way. The objective of our work is to provide tools that help developers systematically explore the space of possible Civil Agent Societies, helping them invent the electronic marketplaces that work best for their intended purposes. We present a framework that captures the fundamental elements and processes of Civil Agent Societies and a methodology for designing, prototyping and evaluating a wide range of “civil” open marketplaces. We also discuss how these ideas are currently being applied to the design of open marketplaces of contract net agents, a useful abstraction of agent-mediated business-to-business e-commerce.

1 Introduction

Software agent technologies promise substantial increases in productivity by automating several of the most time-consuming stages of electronic commerce processes. Agents are software systems, which are capable of interacting with other agents in a flexible and autonomous way, in order to meet the design objectives of their creators [9]. In the context of electronic commerce, we can already point to several examples of agents used to compare information about products, buy products, sell products, etc. [15].

Electronic agent marketplaces are formed by collections of software agents, which interact with one another in order to automatically trade products and services through the Internet. For example, one vision for the future of business-to-business electronic commerce consists of electronic marketplaces, where sets of contractor and subcontractor agents connect with one another and form virtual supply chains for providing goods and services [7].

In the emerging model of 21st century electronic commerce, a variety of open electronic marketplaces will be competing with one another for participants. Independently developed agents will be entering and leaving marketplaces at will, in pretty much the same way that human investors enter and leave different financial markets today. The stakeholders of electronic marketplaces will, therefore, have an interest in making them as attractive to prospective “customers” as possible. One expects that the most successful marketplaces will be the ones that have the lowest barriers to entry (in terms of required agent sophistication) and provide the best “quality of service” guarantees (in terms of security, fairness, efficiency, etc.). The proper design of open electronic marketplaces thus emerges as an important research and practical question.

A lot of the early work on the design of agent marketplaces focused on agent *mechanism design*, that is, on the design of “optimal” rules of behavior to be followed by individual agents [23]. The underlying assumption behind this line of work is that if all agents follow the “right” mechanism, the emerging society will exhibit stable and efficient behavior.

Such research typically assumes that agents will be homogeneous and rational, that their infrastructure will be reliable, and therefore that their relatively simple and “optimistic” rules of behavior will be “intelligent” enough to avoid or cope with whatever deviant behavior or systemic dysfunctions they encounter. The contract net protocol, for example, one of the best-known mechanisms for structuring contractor and subcontractor marketplaces [21] owes its simplicity to many assumptions about agent behavior, some of which are listed in Figure 1. Although such assumptions are possible to guarantee in closed environments, where all agents are developed by the same team, they are becoming less realistic in the open world of the Internet.

Designing efficient and robust open electronic marketplaces, whose participants will be independently developed software agents, each attempting to satisfy the goals of its creator is a difficult problem. Some of the most important challenges include:

- *Heterogeneity*. Open marketplaces cannot expect that all of their members will have an equal level of sophistication. For example, in a contract net marketplace, some subcontractor agents may be able to respond to cancellation of their task by their contractor, while other agents may lack this capability. If they hope to attract a wide enough membership, open marketplaces should be able to provide a certain level of support, even to less sophisticated agents.
- *Limited trust*. Independently developed agents can not always be trusted to follow the rules properly due to bugs, bounded rationality, malice and so on. For example, subcontractor agents may crash or fail to deliver a promised service on time, contractor agents may refuse to pay, etc. Open marketplaces should be prepared to deal with potential fraud or other deviant behavior.
- *Possibility of systemic failures*. Almost any set of social rules of behavior, especially those simple enough to be reasonable for implementation and efficient in execution in a large set of agents, will have “holes” in terms of the potential for unintended emergent dysfunctional behaviors. This is especially true since agent societies operate in a realm where relative coordination, communication and computational costs and capabilities can be radically different from those in human society, leading to behaviors with which we have little previous experience. It has been argued, for example, that 1987’s stock crash was due in part to the action of computer-based “program traders” that were able to execute trade decisions at a

speed and volume that was unprecedented in human experience and thus led to unprecedented stock market volatility [22].

- *Need for rapid adaptation.* Just as their “real world” counterparts, open agent marketplaces should be viewed as dynamic, adaptable systems, sensitive and responsive to demands of their members or to other important changes of the competitive landscape.

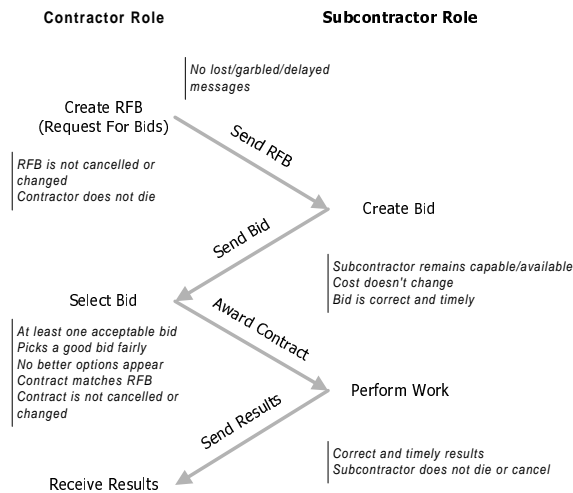


Fig. 1. Simplified description of the contract net protocol. Some of the protocol assumptions are listed in italics.

The typical response of multi-agent system researchers to the previous challenges has been to require all agents of a society to implement more complex, mutually compatible, versions of a mechanism, with hard-coded support for dealing with some of the above issues (see [4, 18, 19] for examples).

Agent societies that emerge in this way are similar to “survivalist societies” of early human history because their members are expected to completely fend for themselves. There is no control or safety net in case things go wrong. There is also no room for agents, which are less capable, or simply slightly “different”. This increases the barriers for participation in these societies. Often, these complex mechanisms impose a significant performance penalty and, in any case, they cover only a subset of the possible exception types. In fact, researchers have proven that for some classes of interaction (e.g. voting) there can be no mechanism, which completely avoids all possible systemic dysfunctions [1]. Finally, by “hard-coding” interaction mechanisms entirely within individual agents, such societies are not particularly easy to adapt.

Civil human societies have successfully coped with similar challenges by developing social institutions that set and enforce laws (e.g. courts, police), monitor for and respond to emergencies (e.g. ambulance system), prevent and recover from disasters (e.g. coast guard, firefighters), etc. In that way, civil societies allow citizens to utilize relatively simple, optimistic and efficient rules of behavior, offloading the

prevention and recovery of many problem types to social institutions that can handle them efficiently and effectively by virtue of their economies of scale and widely-accepted legitimacy. Successful civil societies have thus achieved a division of labor between individuals and institutions that decreases the “barriers to survival” for each citizen, while helping increase the welfare of the society as a whole. In an analogous manner, we believe that the design of the right electronic social institutions will be a crucial success factor in the new universe of open electronic marketplaces.

Isolated examples of useful “electronic social institutions” have been proposed and analyzed by software agent researchers (for example, social monitors in [10]; reputation mechanisms in [24]). However, up to this date, there has been no methodology or framework for systematically deciding what social institutions are needed in a given context and providing guidance on how to design, evaluate and adapt them.

Our work aims to fill this gap. The long-term goal of our research is to use the civil society metaphor in order to develop methodologies and tools for systematically designing open electronic marketplaces. Our work complements a lot of the current research in designing agent-mediated electronic marketplaces by focusing on the design of appropriate *social* (infrastructure) mechanisms that complement the mechanisms of (possibly independently developed) *individual* agents in order to improve the flexibility, robustness and efficiency of the resulting systems. Although this paper focuses on the design of open electronic marketplaces, we would like to emphasize that the results of our work can be applied to the design of any open multi-agent society.

The rest of the paper is organized as follows: Section 2 gives an overview of the Civil Agent Society architectural framework, which allows the rapid prototyping of a wide range of open agent marketplaces. Section 3 presents our methodology for constructing Civil Agent Societies and describes how it has been applied to develop an open marketplace of contract net agents. Section 4 discusses related work. Finally, Section 5 summarizes our conclusions and presents directions for future research.

2 A Civil Agent Society framework for constructing open agent marketplaces

Sociologists have observed that, despite their diversity, human societies can be described through a relatively small set of core elements and processes [13, 16]. These elements and processes thus form a design space that can be used to define a wide range of different societies (Figure 2). Our goal is to define an equivalent design space for software agent societies, supported by a methodology and architectural framework for designing, implementing and experimenting with societies in that space. It is our hope that these tools will enable our research community to better explore the space of possible agent marketplaces and, eventually, to develop guidelines for the design of “good” marketplaces within that space.

<i>Elements</i>	<i>Comprehensive or Master Processes</i>
1. Beliefs (knowledge)	1. Communication
2. Sentiments	2. Boundary maintenance
3. Goals or objectives	3. Systemic linkage
4. Norms	4. Institutionalization
5. Status-roles (positions)	5. Socialization
6. Rank	6. Social control
7. Power	
8. Sanctions	
9. Facilities	

Fig. 2. Elements and master processes of social systems (adapted from [13]).

The following sections present the results of our ongoing work on developing an extensible architectural framework for implementing “civil” open agent marketplaces.

2.1 Core elements of Civil Agent Societies

Civil societies provide an infrastructure for facilitating the conduct of social interactions. From a design perspective they represent a tradeoff between individual autonomy and social support. Societies constrain the behavior of their citizens by specifying a set of *norms*. Conforming to the norms is the cost that citizens have to pay in order to belong to a civil society. In return, civil societies provide *social institutions* that protect citizens from the actions of other citizens as well as from systemic dysfunctions. In order for citizens to have full access to the protection of the society, they typically need to formalize their interactions through *contracts*.

We can see, therefore, that the three core elements of a civil society are its norms, its institutions and mechanisms for formalizing social interactions as contracts. Below we describe how each of these elements is implemented in the Civil Agent Society framework (Figure 3).

Social norms

Marketplaces are a relatively simple type of society. Marketplace “citizens” usually interact with one another through short-lived, transactional relationships with a well-specified beginning and end. Furthermore, each marketplace supports a relatively small number of different transaction types. Based on these observations, Civil Agent Societies represent norms using a knowledge base, which enumerates the set of agent roles and the set of role interaction protocols that are permissible within a given marketplace. Agent roles and interaction protocols are organized in a specialization hierarchy. This way, new roles and protocols can be added relatively easily as special cases of existing ones. Figure 4 depicts a subset of the social norms knowledge base for a Civil Society of contract net agents.

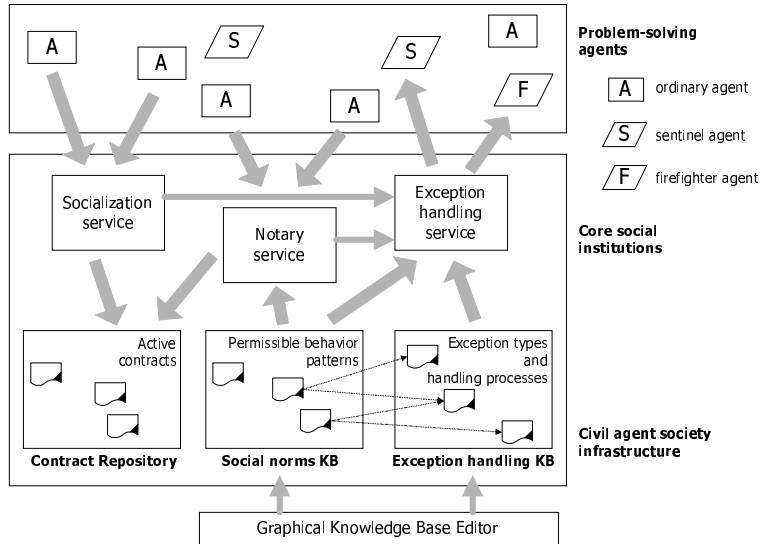


Fig. 3. Architectural overview of the Civil Agent Society framework.

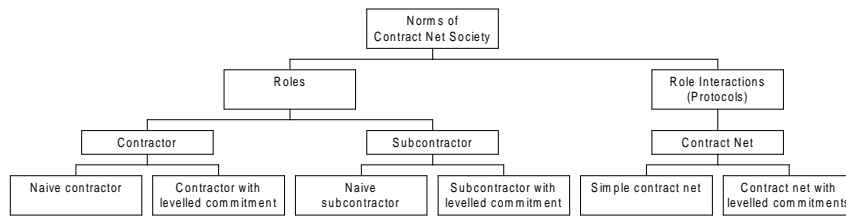


Fig. 4. The social norms knowledge base is organized as a specialization hierarchy of permissible roles and role interaction protocols for a given civil society.

Exception handling social institutions

Given the self-interested and transactional nature of most “social” interactions within a marketplace, an important role of social institutions in marketplaces is that of handling exceptions. The Civil Agent Society framework implements exception handling social institutions as the collection of processes, which anticipate, avoid, detect and resolve all known exception types of all interaction protocols contained in the social norms knowledge base of a civil society.

We define exceptions as any deviation from an ideal sequence of agent behavior, which may jeopardize the achievement of some individual or social goals [6, 11]. We further distinguish exceptions into local and systemic. *Local exceptions* are violations

of normal agent behavior in the context of a single multi-agent interaction (e.g. a single contract). Local exceptions may be caused by programming bugs, system crashes, malicious behavior or incompatible protocols among heterogeneous agents. In the context of the contract net protocol, an example of a local exception would be a situation where a contractor agent crashes after it has awarded a contract to a subcontractor, but before it has paid the subcontractor. Another example would be a situation in which the subcontractor delivers the contracted service late and with low quality. *Systemic exceptions* describe unintended emergent dysfunctional behaviors. Resource poaching [4], a situation where all subcontractors are tied up with low-priority tasks while high-priority contractors remain unsatisfied, is an example of a systemic exception that has been observed in the context of contract net marketplaces.

The exception handling knowledge base of the framework (Figure 3) contains representations of all exception types that are associated with at least one protocol stored in the social norms knowledge base. For each exception type, the knowledge base stores representations of processes for anticipating, avoiding, detecting and resolving exceptions of that type. This information is generated during the exception analysis phase of our methodology (see Section 3.2). Figure 5 shows a partial list of the exception type taxonomy for a civil society of contract net agents. Finally, Figure 6 shows how the social norms and exception handling knowledge bases relate to each other.

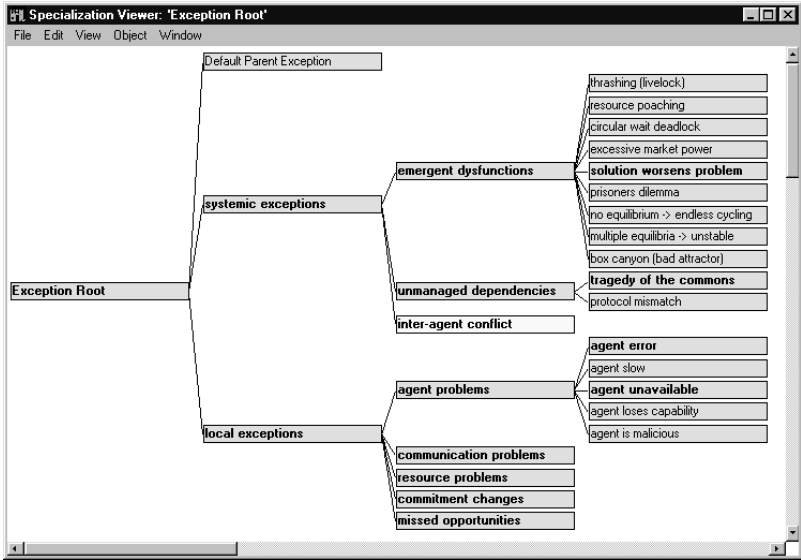


Fig. 5. A subset of the exception types taxonomy for a civil society of contract net agents.

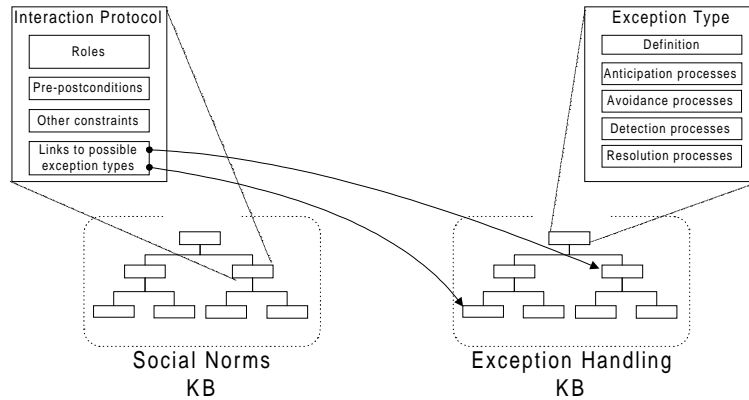


Fig. 6. Overview of the social norms and exception handling knowledge bases.

Contracts

Agents join societies in order to interact with other agents. Contracts make such interactions “visible” to the social institutions. A contract defines a joint commitment of a number of citizens to engage in a “legally acceptable” social interaction in order to achieve a mutually desirable outcome. A social interaction is “legally acceptable” if it conforms to the norms of the given society. The value of contracts in civil societies is that their existence implies the commitment of the society to enforce them, i.e. mobilize its institutions in order to protect the parties involved from contract breaches and other exceptions.

The Civil Agent Society framework supports two classes of contracts:

- *Private contracts*, that is, commitments of two or more society members to engage in a “legal” transaction. The society then commits to protect the agents from local exceptions for the duration of the contract.
- *Social contracts*, that is, commitments of an agent to participate in a society and obey its norms. In return, the society commits to enforce the agent’s private contracts and to protect the agent from systemic exceptions for the duration of the agent’s membership in the society.

2.2 Core services of Civil Agent Societies

During run-time, the Civil Agent Society framework relies on a small set of core services to provide the benefits of a civil society to all “citizen” agents. Core services are responsible for generating new social contracts (i.e. admitting new agents to the society), generating new private contracts and mobilizing the exception handling institutions in order to protect citizens from local and systemic exceptions. Although our approach deliberately leaves the detailed architecture of citizen agents open, in

order to participate in a civil society, citizen agents must, at the minimum, be capable of interfacing to the following three core services:

Socialization service

The process of socialization is an enhanced version of the registration process of other agent environments. During this process, the agent and socialization service engage in an explicit negotiation concerning the agent's capabilities and the society's norms, resulting in a *social contract* between the agent and the society. The social contract indicates membership of the agent in the society.

The following is an example scenario of how agents interact with the socialization service. Suppose that we have developed a buyer agent, who is capable of playing the role of a contractor in a variety of marketplaces, using either the simple contract net protocol described in [21] or the leveled commitment contract net protocol described in [20]. The agent wishes to become a citizen of civil marketplace XYZ and trade with other subcontractors who are citizens of the same marketplace. It contacts the socialization service of marketplace XYZ and declares itself a contractor agent who is capable of interacting through either of the above two variants of the contract net protocol. The socialization agent responds that in marketplace XYZ, only the leveled commitment contract net protocol is acceptable. In addition, the agent needs to pay a membership fee of \$5. Our agent pays the fee and commits to only use the allowed protocol. The marketplace then creates a social contract, which identifies the agent as a member. It also mobilizes the exception handling mechanism of the society in order to "protect" the agent from local or systemic exceptions.

Notary service

Once admitted into a civil society, "citizen" agents are free to contact one another and engage in informal "friendly" interactions. The society does not get involved in those interactions. However, whenever a set of agents intends to engage in a transaction, which requires the protection of the society, they contact the notary service. The notary service verifies that the intended interaction is legal (by comparing it against the set of legal interactions enumerated in the social norms knowledge base), verifies that all agents jointly commit to that interaction and its outcome and generates an appropriate private contract structure. A *private contract*¹ is a data structure, which specifies:

1. a pointer to a legal pattern of interaction, which must be an instance of one of the protocols contained in the social norms knowledge base
2. a set of attributes that define the specifics of this particular instance of the interaction (for example, in the case of a contract, the promised delivery date, payment amount, cancellation penalties, etc.)

¹ Private (agent-to-agent) contracts are distinguished from social (agent-to-society) contracts. Social contracts are created by the socialization service and indicate membership of an agent to a society.

3. a set of agents who commit to play the roles defined in (1) in order to meet the outcomes specified in (2)

Exception handling service

The exception handling service is triggered whenever new contracts are created, or change status (e.g. become canceled, discharged, released, etc.). Such contracts include both the ‘agent to agent’ private contracts recorded by the notary service, as well as the ‘agent to society’ social contracts created by the socialization service. In the latter case, the exception handling service initiates the mechanisms, which look for symptoms of potential systemic dysfunctions.

Upon contract creation, the exception handling service first anticipates all exception types that are associated with the type of interaction defined in the contract. This is achieved by locating the corresponding interaction protocol template in the social norms knowledge base and following the links between that protocol template and its characteristic exception types (Figure 6). Based on the information contained in the exception handling knowledge base, the exception handling service starts a number of “sentinel” agents, whose role is either to try to avoid a given type of exception, or to detect some of its symptoms.

Sentinels work by monitoring some of the communication between agents or by proactively querying agents about their status. In addition, citizen agents may explicitly call the exception handling service, for example, when they believe that a contract they have signed has been breached. Whenever an exception symptom has been detected, the diagnostic component of the exception handling service is triggered. After the exception cause has been determined, the resolution component selects one of the resolution strategies present in the knowledge base and starts “firefighter” agents in order to enact it and bring the society back to an acceptable state. The exception handling service is described in more detail in [12].

3 A methodology for developing Civil Agent Societies

The framework presented in the previous section is meant to be the basis for a systematic, exploratory methodology for designing open marketplaces. This section describes our ongoing experience with using the framework in order to construct and evaluate “civil society” versions of contract net marketplaces.

3.1 Design social norms

The first step of our methodology defines the norms of the target electronic marketplace. As explained in Section 2.1, this involves an enumeration of all roles and interaction protocol variants that are permissible in an open version of the marketplace. Each role and interaction protocol is subsequently modeled as a finite automaton and added to the social norms knowledge base.

In the case of a contract net marketplace, there are two basic roles (contractor, subcontractor) and one basic protocol (contract net). To accommodate “diversity”,

variations of the basic roles and protocols can be easily added as specializations of the basic ones (Figure 4).

3.2 Design exception handling social institutions

For each protocol variant identified in the previous step, a systematic identification of possible exceptions that may arise in an open environment is performed. In our previous work we have developed methodologies and tools that can be used to facilitate the systematic discovery of exception types [6, 11]. We have applied these methodologies in the context of the contract net protocol family and developed a list of possible exceptions, some of which are listed in Figure 5.

For each such identified exception type, a set of processes for avoiding, detecting, diagnosing and resolving it is designed. This collection of processes is added to the exception handling knowledge base of the framework. Finally, links are established between the protocols stored in the social norms knowledge base and their characteristic exception types, as shown in Figure 6.

Figure 7 presents a partial summary of the exception handling analysis performed for two sample exception types. A comprehensive description of our analysis appears in [2].

3.3 Prototype and evaluate

The final step in our methodology consists of developing prototype implementations of the citizen agents (i.e. the contractor and subcontractor agents in the case of contract net) and performing deductive or simulation-based analysis of our prototype agent society. The purpose of the analysis is to evaluate the effectiveness of the norms and institutions designed during the previous steps. Effectiveness is usually measured against our basic design objectives of open societies, such as the ability to cope with heterogeneity, limited trust, unreliable infrastructure and systemic dysfunctions. However, different societies may have different design objectives.

As a first test of our approach, we have implemented a prototype version of a “civil” marketplace of contract net agents and are in the process of evaluating how its various “institutions” affect the efficiency and robustness of the overall system. Figure 8 summarizes a typical simulation experiment. The goal of this experiment is to measure the usefulness of a “social monitor” institution in a failure-prone contract net agent environment. This institution is designed to alleviate the negative performance effects caused by subcontractor agents that may crash unexpectedly after they have been awarded a task but before they have completed the work. It works by periodically monitoring the “health” of subcontractors and assisting in the immediate reassignment of tasks performed by failed subcontractors. In essence, this institution implements the processes labeled **Detection-Process-2** and **Resolution-Process-2** in the analysis of exception type “Delay from Subcontractor Death” (see Figure 7).

Type	Delay from Subcontractor Death
For Protocol	Contract Net
Class	Local

Definition	A subcontractor dies after it has been awarded a contract but before it has completed its task
Criticality	This exception can have a potentially high performance impact when subtasks take a long time.
Anticipation Processes	Anticipation-Process-1 (Maintain a “reliability history” which tracks past crashes of all subcontractors; raise a flag if subcontractor has been unreliable in the past)
Detection Processes	Detection-Process-1 (Contractor times out if results are not received on schedule) Detection-Process-2 (Sentinel periodically polls subcontractor)
Avoidance Processes	Avoidance-Process-1 (Advise contractors to choose different subcontractor or to frequently poll unreliable subcontractor)
Resolution Processes	Resolution-Process-1 (Locate substitute subcontractor; reassign task; update contract) Resolution-Process-2 (Notify contractor when subcontractor dies)

Type	Resource poaching
For Protocol	Contract Net
Class	Systemic
Definition	One or more high-priority tasks are unable to access needed subcontractors because they already have been ‘grabbed’ by lower priority tasks
Criticality	This exception can have a high fairness impact when there is a significant variation in task priority and the available subcontractors population can be oversubscribed.
Anticipation Processes	Anticipation-Process-1 (The potential subcontractor population is currently busy with low priority tasks, and a set of high-priority tasks is expected)
Detection Processes	Detection-Process-1 (The priority of the tasks that have the resources they need is less than the priority of those that do not) Detection-Process-2 (A high-priority contractor does not get any bids for an offered task within time-out period)
Avoidance Processes	Avoidance-Process-1 (Require that subcontractors collect several request-for-bids before bidding, and respond preferentially to higher-priority bids)
Resolution Processes	Resolution-Process-1 (Allow subcontractors to suspend lower-priority tasks and bid on later higher-priority tasks)

Fig. 7. Results of exception handling analysis for two sample exception types.

In the absence of “social monitoring”, the particular variant of the contract net protocol used in this experiment only checks for subcontractor death after a task result fails to arrive by the specified deadline. Figure 8 shows how the existence of such a “social monitor” significantly reduces the completion delay of supply chains where at least one of the subcontractors unexpectedly fails.

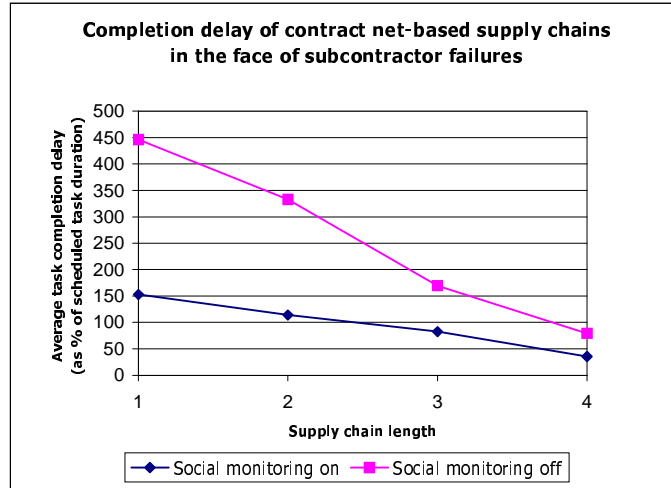


Fig. 8. Effect of a “social monitoring” institution on the completion delay of supply chains where at least one subcontractor agent unexpectedly fails.

4 Related work

By focusing on the integrative goals of developing architectures and methodologies for building multi-agent marketplaces, our work relates to several aspects of multi-agent system research. Due to space limitations this section is, by necessity, partial and discusses only the most important relationships.

4.1 Computational market mechanisms

A significant amount of recent work has focused on the analysis and development of computational market mechanisms. This work has typically made use of normative theories, such as game theory and general equilibrium theory, in order to design a variety of useful mechanisms for areas such as auctions [14], contracting [20, 21], negotiation [17] and task allocation [23]. Mechanisms have typically been analyzed for Pareto efficiency, stability and computational or communication efficiency.

The results of the above work can be considered as the starting point of our approach. Our work complements the above mechanisms with a social infrastructure that aims to improve their performance and robustness in the face of heterogeneity, limited trust, unreliable computation environment and systemic failures. Our contribution in this area is that we are providing a methodology and implementation framework that helps *society* developers (as opposed to *agent* developers) systematically consider the issues that arise when a particular class of mechanisms are

used in an open environment, as well as experiment with various design tradeoffs between individual autonomy and social support, in order to build practical, efficient and robust open systems.

4.2 Social concepts in multi-agent systems

Several researchers have studied the concepts of norms, commitments and social relations in the context of multi-agent systems (see [5] for a representative collection of papers). Such work has typically produced ontologies for describing these concepts, as well as their various states and operations. Furthermore, a number of researchers have proposed architectures for developing agents with social awareness. Jennings and Campos [8] propose the concept of socially responsible agents, which retain their local autonomy but draw from, and provide resources to the larger community. Castelfranchi, et. al. [3] discuss normative agents, that is, agents capable of recognizing, adopting and following norms.

We believe, again, that our work is complementary to these efforts. Instead of proposing a specific architecture for building citizen agents, we take the perspective of the society designer. Our work focuses on how norms and contracts can be represented and used by the society infrastructure in order to build stable, robust systems in the face of heterogeneous agents whose internal architecture may not be reliably known.

5 Conclusions and future research

In the emerging model of 21st century electronic commerce, a variety of open agent marketplaces will be competing with one another for participants. The most successful marketplaces will, in all likelihood, be those that provide the best “quality of service” guarantees (in terms of security, fairness, efficiency, etc.), while meeting such challenges as agent heterogeneity, limited trust, and potential for systemic dysfunctions.

We believe that civil human societies provide a useful model for designing the infrastructure needed to achieve these guarantees. Civil societies have successfully dealt with many of the issues that confront open electronic marketplaces. Through the development of a set of core social elements and processes (Figure 2), successful civil societies have managed to leverage the capabilities of their members, reducing the “barriers to survival” while increasing the total social welfare.

We presented a framework that captures some of the fundamental elements and processes of Civil Agent Societies and helps agent marketplace developers design, prototype and evaluate “civil society” versions of open agent marketplaces. Our aim is to help marketplace developers systematically consider the issues that arise when a particular class of market mechanisms are used in an open environment, as well as experiment with various design tradeoffs between individual autonomy and social support, in order to build practical, efficient and robust open systems.

This is a long-term, ambitious project. The results presented in this paper describe only our first phase of exploration. The following paragraphs describe some of the directions of our ongoing work:

- *Extend the Civil Agent Society framework.* Compared to the elements and processes of human social systems listed in Figure 2 the framework described in Section 2 currently only supports a subset of the elements (roles, norms, sanctions) and processes (socialization, social control). We are working to capture the remaining core elements of civil agent societies as design dimensions that can be easily prototyped and varied within the framework. For example, we are interested in exploring the meaning of *power* in an agent society, including the various alternative ways that power can be exercised and related to the processes of socialization, social control and communication; the *institutionalization* process, that is, the process of dynamically setting and changing the norms and institutions of an agent society; the *systemic linkages* that an agent society should maintain with other agent societies, especially in the context of competing agent marketplaces. Understanding these dimensions can have far-reaching implications, not only for designing agent marketplaces, but also in the more general issues of Internet legislation and governance.
- *Develop guidelines for building “citizen agents”.* One of the motivations behind civil agent societies is the need to accommodate independently developed agents with possibly different internal architectures. Nevertheless, all citizens of civil societies should exhibit a minimum set of capabilities, such as the ability to articulate and reason about norms and contracts. We are working towards formalizing these requirements into minimal interfaces and languages that agents should support in order to participate in civil societies.
- *Develop a “design handbook” for open electronic marketplaces.* The ultimate goal of constructing frameworks is increased understanding and guidelines for action. As we are refining our methodology and framework, we are applying them in order to construct “civil society” versions of the best-known electronic market mechanisms. Our ultimate goal is to organize our findings in a “handbook” for designing open electronic marketplaces.

References

1. Arrow, K. J. “Social Choice and Individual Values”, 2nd ed., Yale University Press, New Haven, CT, 1963.
2. Aryananda, L. “An Exception Handling Service for the Contract Net Protocol Family”, Master’s thesis, MIT Department of Electrical Engineering and Computer Science, May 1999.
3. Castelfranchi, C., Dignum, F., Jonker, C.M. and Treur, J. “Deliberate Normative Agents: Principles and Architecture”, Proceedings of The Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99), Orlando, FL, July 1999.
4. Chia, M.H., Neiman, D.E., and Lesser, V.R. “Poaching and distraction in asynchronous agent activities”, in Proceedings of the Third International Conference on Multi-Agent Systems, Paris, France, July 1998, pp.88-95.
5. Conte R., Falcone R. and Sartor G. editors, “Special Issue on Agents and Norms”, Artificial Intelligence and Law, 7 (1), March 1999.
6. Dellarocas, C. and Klein, M. “A Knowledge-Based Approach for Handling Exceptions in Business Processes”, Information Technology and Management, forthcoming.
7. Jennings, N.R., Faratin, P., Johnson, M.J., Norman, T.J., O’Brien, P. and Wiegand, M.E. “Agent-based business process management”, International Journal of Cooperative Information Systems, 5 (2&3), 1996, 105-130.

8. Jennings, N.R. and Campos, J.R. "Towards a Social Level Characterisation of Socially Responsible Agents", IEE Proceedings on Software Engineering, 144 (1), pp. 11-25.
9. Jennings N.R., Sycara K. and Wooldridge M. "A Roadmap of Agent Research and Development", Autonomous Agents and Multi-Agent Systems 1 (1), 1998, pp. 7-38.
10. Kaminka, G and Tambe, M. "I'm OK, You're OK, We're OK: Experiments in Distributed and Centralized Socially Attentive Monitoring", Proceedings of the Third International Conference on Autonomous Agents, Seattle, Washington, May 1999, pp. 213-20.
11. Klein, M. and Dellarocas, C. "A Knowledge-Based Approach to Handling Exceptions in Workflow Systems", Computer Supported Cooperative Work (CSCW), forthcoming.
12. Klein, M., and Dellarocas, C. "Exception Handling in Agent Systems" in Proceedings of the Third International Conference on Autonomous Agents, Seattle, Washington, May 1999, pp. 62-68.
13. Loomis, C.P. "Social Systems: Essays on their Persistence and Change", D. Van Nostrand Company, Inc., 1960.
14. McAfee, R.P. and McMillan, J. "Auctions and Bidding", Journal of Economic Literature 25, 1987, pp. 699-738.
15. Maes, P., Guttman, R.H. and Moukas A. "Agents that Buy and Sell", Communications of the ACM 42 (3), March 1999, pp. 81-91.
16. Parsons, T. "The Social System", The Free Press, 1964.
17. Rosenschein, J.S. and Zlotkin, G. "Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers", MIT Press, Cambridge, MA, 1994.
18. Saad, A., Kawamura, K., Biswas, G., Johnson, M.E. and Salama, A. "Evaluating a contract net-based hierarchical scheduling approach for flexible manufacturing", Proceedings of the IEEE International Symposium on Assembly and Task Planning, Pittsburgh, PA, August 1995, pp.147-152.
19. Sandholm, T.W. and Lesser, V.R. "Issues in automated negotiation and electronic commerce: Extending the contract net framework", Proceedings of the First International Conference on Multi-Agent Systems, San Francisco, California, June 1995, pp. 328-35.
20. Sandholm, T.W. and Lesser, V.R. "Advantages of a leveled commitment contracting protocol", Proceedings of the National Conference on Artificial Intelligence (AAAI), Portland, OR, August 1996, pp. 126-133.
21. Smith, R.G. "The contract net protocol: high level communication and control in a distributed problem solver", IEEE Transactions on Computers 29 (12), December 1980, pp.1104-1113.
22. Waldrop, M. "Computers amplify Black Monday", Science (238), Oct. 30 1987, pp. 602-604.
23. Wellman, M.P. and Wurman P.R. "Market-Aware Agents for a Multiagent World", Robotics and Autonomous Systems (24), 1998, pp. 115-125.
24. Zacharia, G., Moukas, A. and Maes, P. "Collaborative Reputation Mechanisms in Online Marketplaces", Proceedings of 32th Hawaii International Conference on System Sciences (HICSS-32), Maui, Hawaii, January 1999.