Chapter XXIII
Modern Design Dimensions of Multiagent CSCW Systems

Tagelsir Mohamed Gasmelseid
King Faisal University, Saudi Arabia

ABSTRACT

This chapter introduces and investigates the applicability of the multiagent paradigm for engineering and developing CSCW systems with the aim of advocating modern design dimensions and software engineering implications. It argues that the use of multiagent systems can significantly improve and enhance the functionalities of computer supported work systems. To meet such an objective, the chapter raises the importance of "revisiting" the context and domain of CSCW in accordance with the growing organizational transformations, situational shifts, and technological developments. While such changes are motivating group collaboration, the information systems that support them must be powerful. The author believes that because of their specific limitations and the continuous changes in the collaboration environment, there is an urgent importance of using thorough system-oriented approaches to address the way they evolve. Furthermore, the chapter draws a framework for the use of the multiagent paradigm to understand and deploy CSCW systems by adopting an integrated context of analysis that improves our general understanding about their potentials.

INTRODUCTION

The proliferation and advancement of information technology is dictating new axioms for collaborative work especially in information intensive working environments. Within such environments information technology plays an increasingly significant role by extending the back office (core and support processes) to the front office and beyond the branch. The Internet and e-business, for example, have affected enterprise-wide information availability both in terms of type and quantity. Because the adoption of Internet-based business transaction models has outpaced the development of tools and technologies to deal with information explosion, many businesses are being motivated to share information and tasks through integrated computer supported collaborative work systems. Especially for global enterprises, the use of networks is enabling collaborative work through information sharing and task accomplishment. While such collaboration allows organizations to save resources, it also improves their learning curves within a wider environment of a computer supported collaborative work.

Despite their growing deployment, little has been done to investigate the development aspects of CSCWs. Emphasis continued to be placed on understanding the role of computer systems in group work by using different theories without focusing on the way such systems are being developed. The migration of organizations towards decentralization, micro-management, delegation, networking and alliances, and customer satisfaction coupled with the growing functionality of hardware, software, and communication systems are increasing the demand for augmenting the benefits of such developments by refining the process of CSCW systems design.

The review and analysis of related work provided evidence that the use of agents' technology can improve the functionality of computer supported collaborative work systems. Their use can improve task collaboration and refinement, communication, and coordination by coupling both task identification and implementation characteristics (domain) with the capabilities of agents (agent qualities) in accordance with organizational principles (i.e., unity of command, hierarchy, structure, and decision-making styles) and technological build ups (multiagent technology). However, it is only through this approach that it becomes possible to understand the context of collaboration and the way to support it.

CSCW: BACKGROUND

The growing deployment of computer network technologies (including the Internet) has drastically changed not only the way network-based systems are designed and used but also affected the styles, methods, and environments in different application domains and dictated new axioms for interorganizational collaboration. Within such a technology-intensive environment, it is becoming increasingly possible for groups located in remote trajectories to engage in both synchronous (where all members of the entire work group are working on the task on-line) and asynchronous (when at least some of them are off-line and working separately on the task) collaborative work processes.

The migration towards CSCW originates from the emerging pressures to reduce resources (e.g., lead time, costs, and defects), to increase client satisfaction, to improve communication with others, and to establish consistency in tools and procedures (Steve & Phebe, 2003). They are used to provide and maintain shared information resources and workspaces (David, Jenkins, & Joseph, 2006; Siriwan & Peter, 2006;).

Computer supported collaboration environments are often promoted as an open, safe, and trustable "learning" domains that allow equal opportunities—for collaborating members—to participate without the limitation of knowledge levels associated with work and individual characteristic, collaboration processes, and satisfaction with collaborative work (Silvia, Saskia, Wim, & Nick, 2007; Yan & Jacob, 2006). They have been also viewed as means for maintaining transparency for decision-making quality and trust for openness of communication (Henk, Paul, & van Doremalen, 2004). During such collaborative work, many activities, guidelines, operating procedures, and functions can be initiated, negotiated, "mainstreamed," revised, and implemented by the "collaborating members" of the entire group.

CSCW aims at understanding how collaborative activities, their coordination, productivity, and effectiveness can be supported by means of computer systems (Carstensen & Schmidt, 2002; Kevin, 2003). It is regarded as a fundamentally design-oriented concept that has two main dimensions: (a) technology-centric placing (emphasizing on devising ways to design computer technology to better support people to work together) and (b) work-centric placing (emphasizing on understanding work processes with an aim to better design computer systems so as to support group work). Such orientations reflect the role of computer systems in supporting work groups and the "technology-oriented" socially organized practices of their members (Bannon & Schmidt, 1989; Suchman, 1989; Wilson, 1999).

The operationalization of CSCW depends on the use of groupware (software and related computer networks) that facilitate interaction among collaborating members and sharing of "tasks" and "resources" through interfaces. CSCW's groupware includes, among others, software for tracking document changes, electronic mail software, application-sharing programs, videoconferencing software, instant and e-mail messaging, groupware, wikiwiki Web, computer assisted design (CAD), and software to support the collaborative viewing of Web pages. While different types of groupware can be used in different collaborative environments, they must facilitate cooperation, coordination, and communication among the collaborating members by splitting cooperative tasks into independent (yet integrated) subtasks and managing and supporting "dependencies" among tasks and activities. The functionality of the CSCW's groupware should not be limited to the provision of a sophisticated interface but it must also provide some degree of group awareness that enriches "mutual understanding" among the collaborating members. Group awareness plays an essential and integral role in group collaboration by simplifying communication, supporting coordination, and providing chances for process management and coupling in pursuit of group collaboration. In addition to groupware, perceiving and understanding the responsibilities, activities, and intentions of other members of a collaborating ensemble is a basic requirement for group interaction (Minh et al., 2006).

RELATED WORK

The development of CSCW systems continued to be guided by a wide range of theories including activity theory, conversation analysis, coordination theory, distributed cognition theory, ethno-methodology, grounded theory, situated action, and social/symbolic interactionism (Ackerman & Halverson, 1999; Engestrom, Miettinen, & Punamaki, 1999; Fitzpatrick, Kaplan, & Mansfield, 1996; Shapiro, 1994; Schiff, Van House, & Butler, 1997; Strauss & Corbin, 1998).

The emphasis of such theories tends to be oriented towards describing native cooperative phenomena and computer support for (cooperative) work in fairly abstract and stable context of organizational interaction. The basic assumption is that collaborating members (actors) are interrelated as parts of communities (in which the principles of division of labor applies) to contribute different kinds of "interactive expertise." Based on this understanding, the design of CSCW systems is regarded as a joint activity crossing borders of different communities of practice engaged in interdependent activities. Corporate organizational memory has also been used to approach CSCW systems by using it as a way for capturing accumulated knowledge and making it available and accessible to collaborating members in pursuit of improving their efficiency and effectiveness in a knowledge-intensive environment. Even when using the concept of "active remembering" to improve the potential of "organizational memory" by incorporating organizational, technical, and process-specific constraints, emphasis continued to be made on "tasks" and "processes" rather than on "the way" to develop necessary collaborative work information systems.

Most of the theories are used to study and describe CSCW settings and systems but few of them have explicitly and thoroughly approached their design process and the appropriate support tools. While the development of CSCW systems has witnessed a shift from a system-centered view of information systems to a user-centered paradigm by focusing on contextual enquiry, participatory design, and end user development (Beyer & Holtzblatt 1998; Greenbaum & Kyng, 1991), little has been done to use "reliable" software engineering methodologies to investigate the nature and magnitude of collaboration, the "socio-behavioral" styles of collaborating members, the sophistication of the entire collaborative work, and the "envisioned" support to be provided by computer technologies in a "shared" environment.

Previous studies reveal that CSCW did not seem to be achieving the influence it promised. Because of the lack of integrated design dimensions that address the dynamics of individual and group interactions and the context of information exchanged, the majority of theories used to conceptualize CSCW did not display a sophisticated level of integration. There has been more emphasis on "multidisciplinary" working and less "interdisciplinary" working (Steve & Mann, 2003).

Because CSCW systems are characterized by a high degree of user-user and user-system interaction and hence generate and huge amount of information, it is important to adopt systematic measures to understand information acquisition and utilization domains, explain and predict patterns of group behavior, and detect collaboration breakdowns and support group activity with adequate feedback (Thanasis, Alejandra, & Fatos, 2006). This demands a paradigm shift that incorporates the growing organizational, institutional, technological, and structural transformations. This chapter suggests the use of multiagent systems and "agency" concepts for the development of CSCW systems and the articulation of relevant new design dimensions.

MULTIAGENT SUPPORTED COLLABORATIVE WORK

"Agents," as software entities, can carry out some of the tasks on behalf of their users, other agents or programs with some degree of autonomy using the appropriate information and communication platforms (Bradshaw, 1997; Hyacinth, 1996). Accordingly, they can play different roles including task delegation, users training, event monitoring, information search, matchmaking, and filtering. Despite the differences regarding "what constitutes agency," attributes such as autonomy, reactivity, collaboration, mobility, goal orientation rationality, and socialability are widely cited in agent publications. However, the topology of agents included a wide spectrum of agent types, such as collaborative, interface, information, task, mobile, and reaction agents. Based on the complexity of the agent representation style (individual vs. multiagent), additional agent qualities can be crystallized (Gasmelseid, in press).

A multiagent system is a system consisting of agents that communicate and cooperate to carry out work on the bases of intelligence, communication, cooperation, and massive parallel processing. In a multiagent system, agents jointly use knowledge and resources to solve problems in a context-dependent way. Multiagent systems are deployed to a wide range of applications such as electronic commerce, traffic control, healthcare provisioning, portfolio management, and telecommunications. They proved to be suitable for complex, distributed problems involving a multiplicity of interconnected processes whose solutions demand the allocation of fusion of information and expertise from demographically distributed sources (Sycara, 1998).

The potential and applicability of multiagent paradigm to support collaborative work is driven by the following considerations:

1. Within the context of the knowledge economy multiagent systems can facilitate collaboration by reducing the entry requirements of collaborating groups by shifting work burden to "agents" especially in sophisticated tasks using some integrated "delegation" parameters.
2. Its use enhances coordination and communication through the use of "specialized agents" like task, information, and interface agents linked within the same organizational chains of command by viewing them across a spectrum of superior-subordinate landscape of "agency."
3. The use of multiagent concepts facilitates negotiation, cooperative work, and information sharing in support of the growing institutional migrating to decentralized scales and the

use of cooperative distributed applications. Therefore, the nature and context of CSCW and its growing emphasis on communication, configuration, negotiation, coordination, usability, and information access improves the candidacy of using multiagent systems in CSCW.

The multiagent supported collaborative systems (MASCWS) is driven by the concepts derived from organizational theory, theory of decision making, theory of delegation, as well as agency and information system engineering and development theories. Because members are operating in a socio-organizational context, their "objective" functions and utility matrices can be represented by sets of "collaborating" specialized agents. Based on the principles of division of labor, agents have hierarchal task assignments that coincide with their hierarchy of command and goal congruence mechanisms. Therefore, as shown in Figure 1 below, task, superior, and high level agents are supported by "staff" ones acting in an advisory capacity. Privileges and coordination rights are preserved based on an integrated understanding of the collaboration environment.

The entire MASCWS supports communication across two layers: participants' layer (a.k.a interactions and understanding domain) and agents' layer (a.k.a cooperation and collaboration domain) in the form of computer mediated communication. Computerized artifacts are activated in environments where information sharing is a focus for communication and cooperation. They also support coordination, cooperation, and negotiation through the automation of feedback, feed through, control mechanisms, and computerized artifacts vs. noncomputerized ones. Artifacts are regarded as tools for the achievement of objectives or ends. This demands that agents should understand the work itself as well as necessary linkages with other agents dictated by soft artifacts.

Interactions in the MASCWSs are based on the following main components:

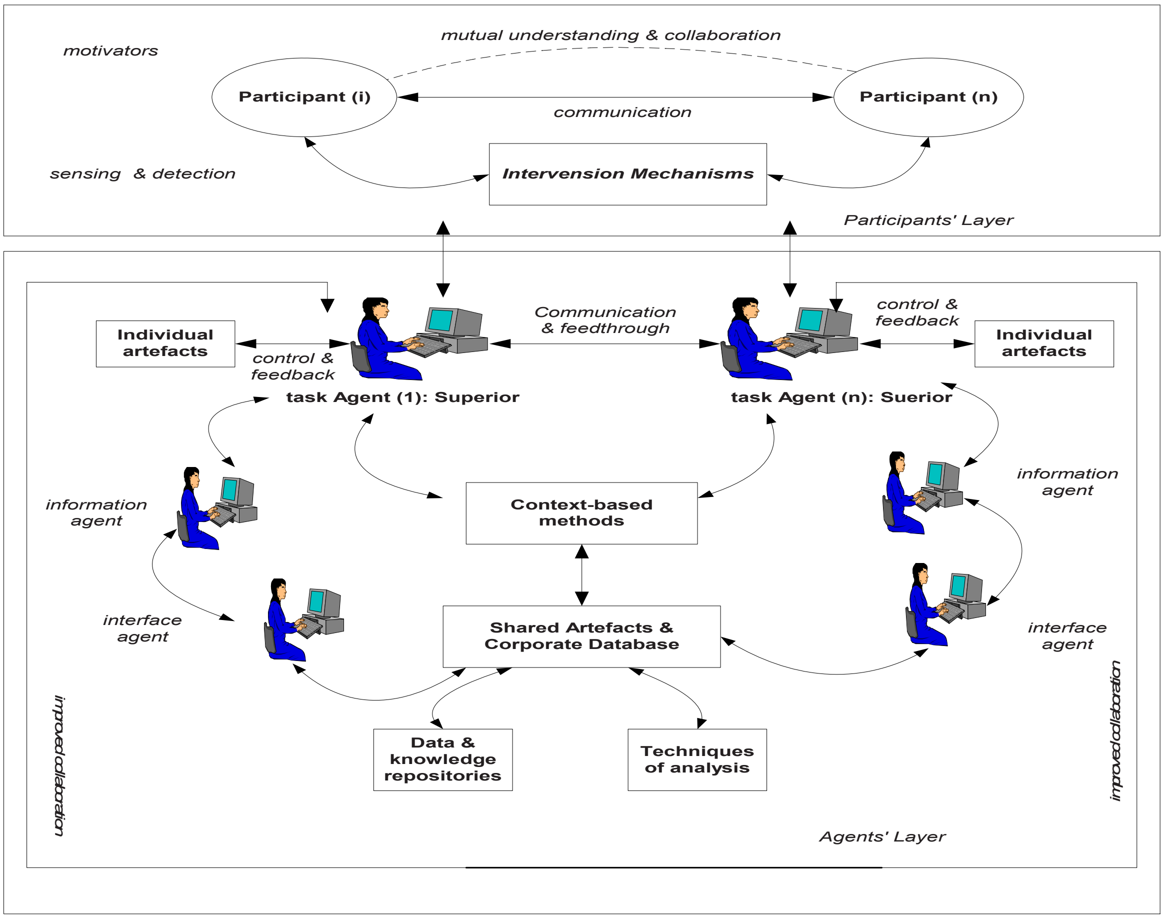
a. **Processing and functional artifacts**
The functionality of agents interacting in the context of the MASCWS environment depends on two levels of artifacts:

1. **Agent-specific artifacts:** These artifacts are agent-based, context-related, and are usually incorporated and represented into the agent's knowledge base and interface engine in order to guide its agent-user interaction and maintain conviviality across its specific and corporate domain of interaction.
2. **Shared artifacts:** These artifacts reflect the dynamics and functionality of not only interagent interaction and cooperation but also the nature and magnitude of tasks, changes of agents' behavior, expectations, and actions to be undertaken within a shared environment.

Artifacts (both individual and shared) are created and transformed incrementally across the entire multiagent CSCW environment until a detailed work is achieved. Shared and individual soft artifacts (databases, plans methods) are controlled and acted upon by all participating agents to improve collaborative practices and benefit from the outstanding processing, integration, communication, and negotiation capabilities of agent-oriented systems. Accordingly, the behavior of agents can be automatically adjusted to enhance mutual understanding among participants. For agents to control artifacts they must be able to perceive the state of the artifacts themselves because different agents and participants have different levels of control over them.

The use of artifacts and their orchestration in a collaborative processing environment brings two issues to the surface: (a) ontological and semantic considerations and (b) functionality orchestration. Ontology refers to the process of sharing and agreeing on a common definition of process-related concepts (e.g., structure of messages, instructions, requests to be supported, semantics, and the list of terms to be used in the content of messages). The explicit definition of (all) concepts to be represented

Figure 1. MASCWS's interaction framework



is not limited to the definition of the concepts themselves, but also expands to include attributes' meaning, constraints on attributes' values, and the relations between the attributes of different concepts. However, despite the use of general purpose ontologies and the provision of editors for creating domain specific ontologies and converters for translating between ontologies, the context and magnitude of the problem of ontology have not been yet appreciated (Gasmelseid, 2007a).

While significant degree of ontological sophistication of agents is necessary for knowledge sharing, cooperation, and interoperability, the interaction between "domain" and "task" ontologies is cumbersome. When designing a MASCWS, knowledge of the task for which the ontology will be used is a prerequisite for the definition of the domain ontology.

The importance of maintaining an appropriate level of (both agent-based and institutional) functionality orchestration stems from the imperativeness of using coherent sets of shared rules, behaviors, expectations, and authority relationships. Particularly in open dynamic environments, negotiation, criteria assignment, partial global planning, assumption surfacing, argumentation, and evidential reasoning techniques, among others, can be developed and deployed to build coherence, orchestrate functionality, and mainstream coordination in the multiagent CSCW environments (Cammatara, McArthur, & Steeb, 1983; Durfee, 1987; Huhns &

Bridgeland, 1991; Steeb, Cammarata, Hayes-Roth, Thorndyke, & Wesson, 1988; Sycara, 1998).

Associated with the use of individual and shared artifacts is the development and activation of appropriate feedback mechanisms. It is paramountly important that all agents must be capable of perceiving and controlling artifacts and states of objects directly not through other agents or participants and be capable also of gaining feedback. Feedback is associated with the capability to control artifacts. Within this context, all agents must have equal (or layered) access to shared and individual artifacts and should be in a position to control the interactions associated with these artifacts. This is because shared artifacts are usually used as the subject and medium of communication between agents with regards to actions to be implemented based on such artifacts in a way that significantly affects the actions of other agents, the work itself, and the overall collaborative system.

The ability of other agents to observe the reactions of an entire agent or participant is called a "feed through." While feedback denotes the information an agent receives about its entire performance and actions, feed through provides information about the actions of their fellow agents, other programs, or users (collaborative members). Because of this feed through agents are able to communicate through the artifact where they, collaboratively, feel the effects of each other depending on the nature and magnitude of the tasks implemented and shared through shared artifacts. Accordingly, agents became able to indicate different objects or reactions by moving, sending information, or doing any other action which may be accompanied with phrases. Such a process becomes more complicated when multiple means are used because that will result into some semantic and ontological complications.

b. **Context-based methods and models**

The question of models and methods assumes high importance in MASCWSs. The development of the model base of such collaborative system is guided by two basic considerations:

1. **Model congruence and methods' consistency:** The model base of a MASCWS incorporates two modeling layers: agent-specific and system-oriented. Agent specific models reflect the nature and magnitude of objectives of its owner, the degree of task stability, and the set of expectations. However, the efficiency of such models is contingent upon its conviviality with the operating environment shaped by the interactions between agents and their owners and interfacing with different knowledge and data repositories or context. System-oriented models reflect the dynamics of the entire collaborative work across the processing landscape managed by different agents representing and serving collaborating members. The modeling of the entire MASCWS depends exclusively upon
    i. The ability of system developers to, thoroughly, understand the context of collaborative work and, accordingly, model individual and collaborating members' practices, processes, and expectations. Such a consideration reflects the degree of model congruence and coupling needed.
    ii. The nature of the work to be collaborated, that is, whether members are collaborating on work that can be achieved in both synchronous and asynchronous work patterns or does the existence of all collaborating members' constitutes a prerequisite for task implementation. Another dimension is that the work to be achieved may have some decision-making dimensions which significantly affect not only the entire model base of each collaborating member and the entire CSCW system but also calls for developing the appropriate mechanisms necessary for the management of privileges, decision-prioritization, negotiation, and model-method-modification matrix techniques.

2. **Context orientation of models and methods:** The term "context" reflects different dimensions. It is usually used to denote "a domain for data collection, processing, information generation, and use." It has also been associated with the ability of different CSCW systems to "be aware about the objects in their surrounding environments." The orientation of the models incorporated into the entire CSCW system is known to be context-oriented when they reflect both the functionality and objectives of the collaborating members while they interact with their respective agents as well as when they interact in the open space of the multiagent SCW and its environment. However, based on the nature of the task to be achieved as well as the nature and type of agents (and accordingly models) to be involved it is possible to decide upon the importance of incorporating different levels of "context awareness" to account for the mobility of members, machines, and database and understand their affects on the process of agent and system modeling.

c. **Data and knowledge repositories**
The MASCWS includes a corporate database that represents different types of data based on the functionality of agents and collaborating members. The degree of sophistication to be exhibited in such database reflects the degree of work complications and the nature of agent and system interface matrix. Therefore, the design of such database as shown in Figure 1 above brings the following issues to the frontline agenda of system developers:

   i. The importance of focusing on data refinement, integration, and management to ensure that all data elements can be incorporated as "usable" components into the agent's database and the corporate database of the SCW system. While such consideration may affect the overall functionality of the system and collaboration practices, it also increases the need for developing and using appropriate mechanisms for the discovery and management of database dependencies.

   ii. The need to place emphasis on knowledge mainstreaming, management of privileges, and accessibility control. Such consideration is driven by the fact that the process of database design will be complicated with the variety, unpredictability, and instability of the "qualities" of collaborating agents and the growing importance of incorporating "mobility" and "context-awareness" dimensions.

d. **Analytical techniques**
The set of analytical methods includes all possible data analysis tools that are needed for the achievement of the entire task. They include data analysis tools such as data mining, OLAP (multidimensional data analysis, etc.), task scheduling and prioritizing, information acquisition and sending, dialogue enablers, and user-agent interventions, among others. For a comprehensive functioning of the MASCWS the applications that run such tools must be orchestrated to enable agents to switch from synchronous to asynchronous modes of work and migrate across applications and networks. While such techniques can be a part of the shared artifacts of the system, they must be incorporated into the knowledge and interface engine of the collaborating agents.However, it should be noted that the development and the implementation of such tools for cooperative work may result in unpredictable changes not only in the work to be shared but also in the communication, resource use, and coordination matrix of the overall system.

DESIGN-ORIENTED CHALLENGES

The growing organizational and operational shifts are dictating some transformations that significantly affect the process of designing CSCW systems as shown below:

Information Integration and Model Coupling

CSCW systems usually run on a WAN or LAN or separated independent application-oriented and functional processing systems. Therefore, the central concern is how to allow for interconnection and interoperation of multiple existing legacy systems to facilitate information sharing. Such a requirement calls for the adoption of an integrated approach for conceptualizing such legacy systems by maintaining an integrated view that accommodates a wide range of system specific and task-oriented issues. The basic challenge is how to acquire and integrate relevant information to support distributed problem solving and processing. The acquisition, communication, sharing, and use of relevant integrated information aims at improving the capacity of collaborating parties to model relevant phenomenon based on the majority (if not all) interdependencies, using multidimensional decision data. The failure to adopt an integrated approach that enables collaborating parties (and their applications) to interact with databases, application servers, content management systems, data warehouses, workflow systems, search engines, message queues, Web crawlers, mining and analysis packages, and other enterprise integration applications, challenges the efficiency of the entire MASCWS.

The development and appreciation of enterprise and system models is a complex task that deserves the mobilization of different resources particularly in legacy systems encompassing a wide range of complex infrastructures. While enterprise and business models are concerned with promoting understanding of business deliverables, functionalities, and utilization of resources, system models address design issues that fulfill both technical and functional specifications. Coupling such models is constrained by a wide range of organizational and policy limitations that shape information requirements to be gathered and the extent of flexibility of data retrieval and response times. On the other hand, the process of model integration is also constrained by the thoroughness of understanding the technological infrastructure in terms of database, programming languages, operating systems, CASE tools and compliers, and so forth, and the way they can be used to produce the relevant system.

System and Network Optimization

Because CSCW systems operate as industrial networks they need to be optimized at two layers: the system layer and the network layer. Unless, under careful monitoring, the functional aspects of the legacy systems are optimized at the expense of the data and the network for implementation expediency, that is, the data and hardware/systems software were tailored to the application and therefore disintegrated with regard to the enterprise.

Related to the optimization of system and network processes is the management of flexibility issues. The banking system, for example, operates as a "legacy system," and is simply organized around existing applications which were built under the assumption that nothing would ever change. If anything changes, particularly in models, they have to be "reverse engineered" from available information in a very costly domain with significantly questionable confidence in its accuracy and "do-ability."

MODERN DESIGN DIMENSIONS AND SOFTWARE ENGINEERING IMPLICATIONS

The above mentioned complexities and challenges motivate the inclusion of some system design dimensions such as the following.

Dynamic Knowledge Streamlining

Determining relationships such as functional and inclusion dependencies within and across databases is important for information integration in MAS-CWSs. When such information is not available as explicit metadata, it is impossible to discover potential dependencies from "distributed" source databases, to manage redundancy of space and time and to minimize complexity. The discovery of inclusion dependencies will be beneficial in any effort to integrate or compare unknown databases particularly with applications in which data about similar real world objects is collected independently. The inclusion dependency discovery problem is loosely related to the problem of association rule mining (Agrawal & Ramakrishnan, 1995).

Generally speaking, an inclusion dependency (IND) over a database schema **R** is a statement of the form $R_1[X]$ □$R_2[Y]$, where R_1, R_2□ **R** and X, Y are sequences of attributes such that X□ schema (R_1), Y □ schema (R_2) and |X| = |Y|. Let d be a database over a database schema **R**, where r_1, r_2 □d are relations over relation schemas R_1, R_2□**R**. An IND $R_1[X]$ □$R_2[Y]$ is satisfied in a database d over **R**, denoted by d |= $R_1[X]$ □$R_2[Y]$, if for all t1 □r_1, there exists t_2□r_2, such that $t_1[X] = t_2[X]$.

As shown in Figure 2 below, it is possible to articulate and manage at least six interconnected schemas together with their relations and dependencies in a collaborative work system. Such a system is designed and basically dedicated for collaborative assessment and approval of credit requests placed by farmers to different bank branches. These schemas include a banker information schema, a branch schema, a loan schema, a customer's schema, a credit card schema, and an accounts' schema as a base for articulating relevant database inclusion dependencies.

The problems associated with the discovery and management of database (especially) inclusion dependencies affect the semantics of databases, relational database design and maintenance, database reverse engineering, semantics query optimization, and efficient view maintenance in data warehouses. The discovery of suitable inclusion dependencies is a complex process because it is impractical to discover all nontrivial inclusion dependencies satisfied by a particular instance especially when testing multiple relational schemas. Inclusion dependency should not be regarded as a process of "duplicating attributes that are used to link together the relational schemas in a database schema" but instead, it should reflect processes at a single functional system (such as investment) as well as cross-system linkages in a way that promotes information sharing and the development of learning-oriented value-adding networks (Gasmelseid, 2007a, 2007b).

One of the main design dimensions of multiagent CSCW systems is to streamline the process of database dependencies through dynamic streamlining of agents' knowledge and interface engine. The basic aim of such design dimension is to account for the nonlinearity associated with the shift in the context of data processing and interface capabilities on the one hand and the paramount importance of maintaining sustained database orchestration domains on the other hand.

While the discovery of such dependencies may be simple in a computer supported collaborative loan approval process as shown in Figure 2 above because of the limited number of schemas, the change of the context of data processing may affect the whole process significantly. The simplicity of articulating these six schemas stems from the possibility of developing and managing "associations." However, under the situation of mobility (i.e., user, application, or service), creating that association is not a simple task.

Based on Figure 2, the entire database processing can be done online where finance seekers can interact with an investment information agent and place a query. The information agent actually is a task agent that uses other information and interface agents to retrieve information from a database or knowledge base that includes not only string and numeric data but also a repository of maps, sounds, multimedia and so forth, a feature that no human can handle over time. Information included in maps is difficult to update and needs much more expertise and a complex interface.

The dynamics of database dependencies become more complicated (especially in global enterprises) because of "task re-engineering," the migration from centralized decision making towards "micro management" and "delegation," and the engagement in international coalition systems or multinational mergers and acquisitions. While such shifts are complicating the management of database dependencies and generating modified management styles they are also being accompanied with a considerable migration towards distributed database and file management systems, mobility and context awareness considerations and incorporation of advanced ubiquitous pervasive features to maintain "reliable" management information ensure operational and outstanding reach and integrate processing among remote trajectories of operating units. Because the movement from the "mainframe era" through the PC era to the "ubiquitous computing era" is shaping business environment and dictating new axioms for doing business, the dynamic knowledge streamlining appears as an emergent design consideration through which not only task orchestration can be handled but also the interface, model base, and knowledge engine of the entire collaborative multiagent organization can be reengineered.

"Good Practices" Oriented Accumulation and use of Integration-Design Knowledge

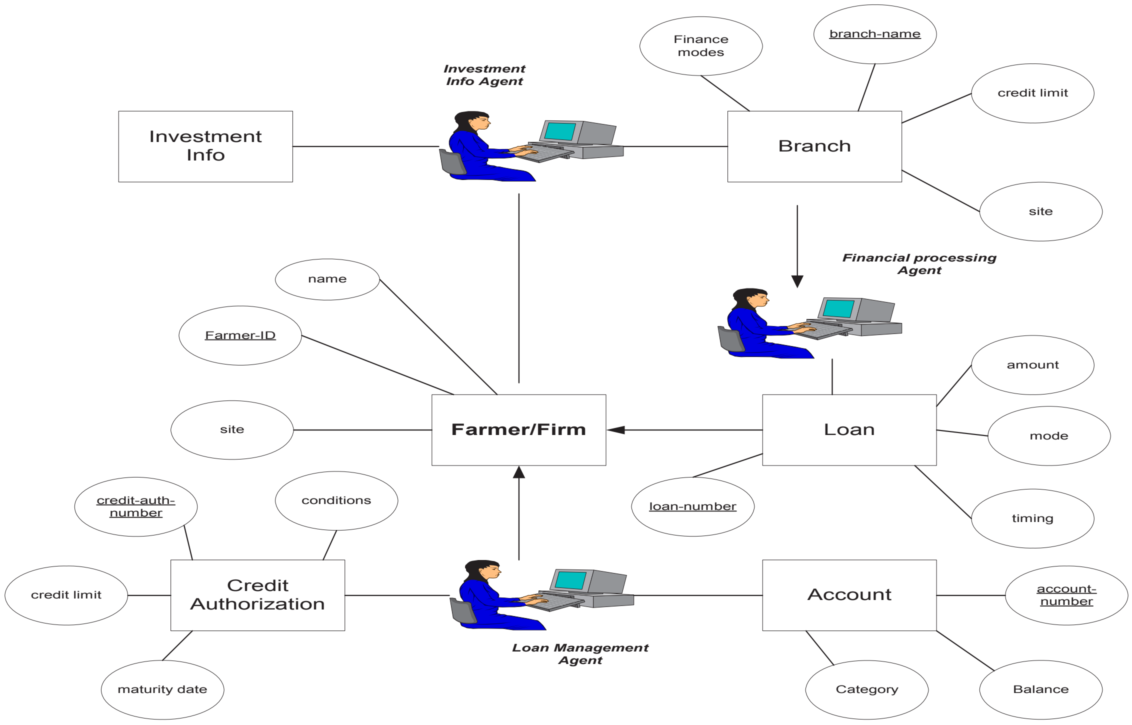
Because of the uniqueness of applications and hence system design mechanisms, the development of multiagent collaborative systems should be based on the use of multiple methodologies. In addition to generally accepted agent-oriented software engineering theories, there is a growing importance to use the principles of "good practices" and "lessons learned" as complementary design techniques. While previous research has advocated some "agent qualities" and characteristics based on the existing business and industrial functionalities, additional features may be acknowledged as a result of future shifts. Therefore, the use of the principles of good practices and lessons learned can provide more process-oriented emphasis on clear specification and communication of problems and constitute a base for a new deign dimension with regards to the development of knowledge-based models (domain-specific and domain-independent), and enhanced capturing of the (formal) semantics of different application domains. When we adopt the principles of good practices and lessons learned for the development of multiagent CSCW systems the functionality of the domain-specific approach to self-integrating systems may be enhanced with regards to the management of operational complexities. Because good practices and lessons learned may result into new agent qualities, the design of the entire multiagent CSCW system may significantly be affected due to the possible change of designer's "perception" with regards to agent (and collaborating members') intentions, goals, plans, beliefs, and "prioritization" of experience models on top of knowledge-based models to share domain-specific concepts. Integrating good practices and lessons learned as design dimensions also affects "machine learning" strategies and automated adaptation methods that provide for an efficient means of self-integration.

The adoption of good practices and lessons learned from different application domains enhances knowledge acquisition and use and facilitates the development and refinement of a set of "implementation" metrics supported by theoretical foundations necessary for gauging semantic equivalence. While coordination, collaboration, and negotiation theories and technologies may allow for flexible integration of systems at the level of task and process definitions, incorporating good practices and lessons learned allows for the development of smart human-computer interfaces and enables the development of usable semantic-based ontological descriptions.

INSTITUTIONAL IMPLICATIONS

There are some institutional and organizational implications associated with CSCW systems and

Figure 2. Schema representation



their development. This is because "information systems development" is usually viewed as an "organizational development" process through which foundations can be set for the resolution of different potential "conflicts" across organizational landscapes. Accordingly, CSCW systems can be used as means for managing (primary, secondary, tertiary, and quaternary) contradictions and mainstreaming processes as described by the activity theory.

The deployment of multiagent systems to support task handling, information sharing, and decision making in a multiagent SCW environment differs from their use to enable a robot to move (as it is the case of engineering applications), a customer buys a commodity through on line platforms (transaction processing as it is the case of electronic commerce), or researcher acquires and filters information. In contrast to other areas, the use of multiagent systems to support collaborative work is significantly affected by the domain and environment of such collaboration which significantly dictates their architectures, models, and frameworks. The style of task integration and coordination adopted determines the type of "agents" to be included in an agent organization, the function-capability matrix, and the framework to be adopted to design the appropriate architecture. On the other hand, because the dynamics of collective work as well as their environments are complex, the deployment of multiagent systems should encompass a wide range of hierarchical, organizational, and institutional considerations.

Within this context, significant efforts are required to streamline technological as well as organizational concepts of division of labor (which guides the articulation of agents and their capabilities), hierarchy (which establishes linkage among agents and guides the distribution of control and supervision), goal congruence, and unity of command (which shape the extent of coordination needed). The orientations to be followed to conceptualize and develop "decision-oriented" multiagent organizations to support collaborative work must emphasize these concepts to maintain a considerable degree of decision-technology feasibility.

The complexities associated with collaborative work have been related to the various organizational transformation exhibited in organizations which originate from competitiveness, resource scarcity, and technological development. Such transformation has been accompanied with the need for maintaining competitive advantage, curtailing conflicts, and emphasizing the involvement of stakeholders. Within this context, the incorporation of the concepts derived from the context of multiagents systems can improve the capacity of decision support systems to provide the appropriate aid in complex decision-making environments. However, particularly under conditions of scarcity where the potentials of conflicts are high, multiagent decision support systems can play a significant role in the process of involvement and negotiation.

However, the feasibility and effectiveness of using multiagent systems to enhance collaborative work in a complex decision making environments has some institutional considerations depending on:

a. The level of information accessibility available for the collaborating parties and whether any sort of information asymmetry exists.
b. The degree of flexibility and resource modifications affecting the context of collaboration through improved capabilities on the part of the provider of resources.

These two basic variables indicate the degree of independence an agent exhibits during the collaborative process and the level of "collaborative" involvement. The level of such involvement is governed by the nature of problems faced by the collaborating members, the basic contributions of agents and the degree of agent-user delegation. Especially in collaborative works that take place in uncertain situations, agents cannot be provided full autonomy to carry on all processes and functions in a fully-delegated fashion. The intervention of the collaborating members is paramountly essential to manage exceptional situations arising from societal, political, and economic considerations that cannot be a part of the corporate model.

CONCLUSION

The technological advancements experienced in the areas related to computer-based information systems such as telecommunication, databases, and software together with the expanding mobility, parallel processing, and enabled human-computer interaction functionalities have set the foundation for the growing deployment of software agents' technology in different areas. The approaches used to conceptualize "agents" ranges from the attempts oriented towards stating definitions to the focus on "the qualities that constitute agency." On the other hand, the variety of the "solutions" envisioned from the use of agents in different applications reflects the diversity and multiplicity of such technology. The deployment and use of multiagent CSCW should rather help not only "mediate" between collaborating groups but also reduce differences in a wider context of organizational learning in which collaboration takes place.

While the use multiagent technology to support collaborative work practices may provide outstanding advantages that improve interorganizational operations and cooperation, their success warrants considerable degree of system monitoring, network administration, infrastructure reconfiguration, and, most importantly, wider involvement of the collaborating groups.

REFERENCES

Ackerman, M., & Halverson, C. (1999). *Organizational memory: Processes, boundary objects, and trajectories.* Paper presented at the 32nd Hawaiian International Conference on Systems Science, Maui, HI. IEEE. Retrieved December 26, 2005, from www.eecs.umich.edu/~ackerm/pub/99b26/hicss99.pdf

Agrawal, R., & Ramakrishnan, S. (1995). *Mining sequential patterns.* Retrieved November 2006, from http://citeseer.ist.psu.edu/agrawal95mining.html

Bannon, L., & Schmidt, L. (1989). CSCW: Four characters in search of a context. In *Proceedings of the First European Conference on Computer Supported Cooperative Work*, Amsterdam. Retrieved November 2006 from http://www.it-c.dk/~schmidt/papers/cscw4chart.pdf

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems.* San Francisco: Morgan Kaufmann.

Bradshaw, J. (1997). An introduction to software agents. In J. Bradshaw (Ed.), *Software agents* (pp. 3-46). Menlo Park, CA: AAAI Press.

Cammarata, S., McArthur, D., & Steeb, R. (1983). Strategies of cooperation in distributed problem solving. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)* (pp. 67-770).

Carstensen, P., & Schmidt, K. (2002). Computer supported cooperative work: New challenges to systems design. In K. Itoh (Ed.), *Handbook of human factors*. Tokyo. Retrieved July 2006, from http://www.it-c.dk/people/schmidt/papers/cscw_intro.pdf

David, G., Jenkins, J., & Joseph, F. (2006). Collaborative bibliography. *Information Processing & Management, 42*(3), 805-825.

Durfee, E. (1987). *A unified approach to dynamic coordination: Planning actions and interactions in a distributed problem solving network.* Unpublished doctoral dissertation, University of Massachusetts, Department of Computer and Information Science.

Engeström, Y., Miettinen, R., & Punamaki, R. (1999). *Perspectives on activity theory.* Cambridge University Press.

Fitzpatrick, G., Kaplan, S., & Mansfield, T. (1996). Physical spaces, virtual places and social worlds: A study of work in the virtual. In *Proceedings of the Conference on Computer Supported Cooperative Work, ACM* (pp. 334-343).

Gamelseid, T. (in press). *Engineering multiagent systems: An information security perspective. Encyclopedia of information security and ethics.* Hershey, PA: IGI Global.

Gamelseid, T. (2007a). Multiagent Web-based DSS for global enterprises: An architectural blueprint. *Engineering Letters, 13*(2), 173-184.

Gamelseid, T. (2007b, July-September). From operational dashboards to effective e-business: Multiagent formulation and negotiation of electronic contracts. *International Journal of E-Business Research, 3*(3), 77-97.

Greenbaum, J., & Kyng, M. (1991). *Design at work: Cooperative design of computer systems.* Hillsdale, NJ: Lawrence Erlbaum.

Henk, A., Paul, B., & van Doremalen, J. (2004). Travail, transparency and trust: A case study of computer-supported collaborative supply chain planning in high-tech electronics. *European Journal of Operational Research, 153*(2), 445-456.

Huhns, M., & Bridgeland, D. (1991). Multiagent truth maintenance. *IEEE Transactions on Systems, Man, and Cybernetics, 216*(6), 1437-1445.

Hyacinth, S. (1996). Software agents: An over view. *Knowledge Engineering Review, 11*(3), 1-40.

Kevin, L. M. (2003). Computer-supported cooperative work. *Encyclopedia of Library and Information Science*, 666-677.

Minh, H., Tran, G., Raikundalia, K., & Yun Yang, S. (2006). An experimental study to develop group awareness support for real-time distributed collaborative writing. *Information and Software Technology, 48*, 1006-1024.

Schiff, L., Van House, N., & Butler, M. (1997). Understanding complex information environments: A social analysis of watershed planning. In *Proceedings of the Conference on Digital Libraries* (pp. 161-168).

Shapiro, D. (1994). The limits of ethnography: Combining social sciences for CSCW. *Computer supported cooperative work*. Chapel Hill, NC: ACM.

Silvia, D., Saskia, B., Wim J., & Nick, J. (2007). Students' experiences with collaborative learning in asynchronous computer-supported collaborative learning environments. *Computers in Human Behavior, 23*(1), 496-514.

Siriwan, S., & Peter, H. (2006). A Bayesian approach to generating tutorial hints in a collaborative medical problem-based learning system. *Artificial Intelligence in Medicine, 38*(1), 5-24.

Steeb, R., Cammarata, S., Hayes-Roth, F., Thorndyke, P., & Wesson, R. (1988). Distributed intelligence for air fleet control. In A. H. Bond & L. Gasser (Eds.), *Readings in distributed artificial intelligence* (pp. 90-101). San Francisco: Morgan Kaufmann.

Steve, G., & Phebe, M. (2003). Interdisciplinary: Perceptions of the value of **c**omputer-supported collaborative work in design for the built environment. *Automation in Construction, 12*(5), 495-499.

Strauss, A. L., & Corbin, J. M. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications.

Suchman, L. A. (1989). *Notes on computer support for cooperative work*. Jyvaskyla, Finland: University of Jyvaskyla, Department of Computer Science.

Sycara, K. P. (1998). Multiagent systems. *AI Magazine, 19*(2), 79-92.

Thanasis, D., Alejandra, M., & Fatos, X. (2006). A layered framework for evaluating on-line collaborative learning interactions. *International Journal of Human-Computer Studies, 64*(7), 622-635.

Wilson, P. (1999). *Computer supported cooperative work: An introduction*. Kluwer Academic Publishers.

Yan, X., & Jacob, F. S. (2007). Emergent CSCW systems: The resolution and bandwidth of workplaces. *International Journal of Medical Informatics*. Corrected Proof, doi:10.1016/j.ijmedinf.2006.05.037, Available online 5 July 2006.

KEY TERMS

Collaborative Systems Engineering: It describes the integrated process of using (collaborative) software engineering methodologies to develop collaborative systems by addressing their contexts, functionalities, interface qualities, and implementation parameters.

Computer Supported Collaborative Work Systems: The group of computer-based systems that facilitate collaboration among people through enhanced information availability and sharing, task accomplishment, and consensus building. Their functionality is based on the use of a mix of groupware components.

Context Orientation: It denotes the ability of the collaborative work system to benefit from the qualities of computer systems to sense and understand the characteristics of its work-related and function-based attributes to use them for group collaboration.

Data Mining: Describes the process used to collect and analyze large amounts of data in order to understand patterns of behavior. Data analysis techniques such as OLAP, OLTP, and multidimensional analysis, among others, are commonly used for data mining to produce reports and generate functional data marts.

Groupware: Groupware includes all components (i.e., hardware, software, and data communication processes) that facilitate group collaboration through task accomplishment and information sharing. Based on the nature of the work to be done collaboratively, the appropriate mix of groupware technologies (e.g., e-mail, GDSS, digital voice mail systems, text conferencing, and video teleconferencing) can be selected.

Ontological Representation: Refers to the systematic representation of a shared (and agreed upon) common definition of the concepts that derive the functionality of the entire MASCW such as the structure of messages to be exchanged among collaborative members, semantics, and the list of terms to be used in the content of messages, instructions, individual, and joint requests.

Software Agents: They are the software components that use resources to perform activities on behalf of their owners and/or other agents or programs. They possess qualities of reactivity, autonomy, collaboration, mobility, and conviviality, among others, that increase their deployment in the form of multiagent organizations to support distributed problem solving and collaborative work.