OFFICE AUTOMATION

THE CHANGING MEANINGS OF THE TERM *OFFICE AUTOMATION*

The Office Automation Problem

The term *office automation* conjures up images of office buildings full of fax machines, telephones, photocopiers, and personal computers with printers; of busy people at desks shuffling paper, stacking binders, and making phone calls. It seems obvious that information and communication technology is highly applicable to office work. Office work consists almost entirely of information manipulation: information collection, processing, and information exchange.

Contrary to this widely held belief, office work has received only moderate attention regarding increasing efficiency through automation. Office communications are a good example. Written communications represent on average 20% of all activities in the office, yet these activities consume on average 80% of the time spent. In addition, media gaps are a major contributor to the great effort spent on office work.

Filing and retrieving of documents or files, for instance, remains time-consuming. The most effective way for filing is still the indexing system. While advances have been made toward searching and retrieving by content indicators, most offices still rely on the conventional indexing schema (alphabetical, date, numerical, etc.). This problem also applies to catalogs, lists, and inventories, to name just a few.

Another office problem concerns communications. Professionals and office workers are frequently hard to reach. People may be in meetings or on the road. Multiperson communications represents a particularly difficult problem. Time-consuming solutions are provided by teleconferencing, with its high overhead in preparations and setup.

Early Applications of Information Technology in Offices

Early applications of information technology (IT) in offices focused on the automation of tasks for individual users. The user was presented with a set of diverse functions, such as text processing, notebooks, calendars, telecommunications, filing, and archiving.

Specialized office machines document this focus on task automation. Typewriters turned into single-function text processors, complete with screens and limited memory capability. Fax machines were introduced to speed the delivery of letters. Photocopiers replaced carbon paper.

Early office automation systems evolved from text processing through increasing document management capabilities on the content side and improved mail capabilities on the communications side. At the convergence of these developments, a new kind of systems emerges: activity support systems. Currently such systems are called *workflow systems.* The important step was the change of view from task automation to process support.

A History of the Changing Focus in Office Automation

The change in focus that permeates the field of office automation is vividly illustrated by the various names for the special interest group (SIG) in the Association for Computing Machinery (ACM). The SIG was founded in the late 1970s, when factory automation was sweeping the industry and produced major success stories. It was expected that the same could be done for office work. The SIG therefore came about as SIGOA, the special interest group on office automation.

The original mission statement of SIGOA reads as follows:

> SIGOA is interested in all aspects of the application of computing techniques to office activities. SIGOA is of interest to technical and managerial personnel from user organizations, vendors, consultants and academia. Its bulletin covers methodological and technical articles, surveys and proposed standards for office automation. Also covered are specifications and evaluations from actual case studies.

After a few years, it became apparent that office work is qualitatively different from factory work. Office work as such could not be automated in the same fashion as factory work could, with numerically controlled machines or process control. The "paperless office," a buzzword of that period, has become the icon of that early misconception.

As a result, the focus shifted to information systems that supported office work. Automation was out, support was in. With this change in perspective, the field, and the SIG, were renamed *office information systems* (OIS). Individual office tasks, such as copying or faxing, still received commercial attention by the makers of specialized office equipment—and do so even today. Yet the core interest moved to the support of information through computer-based information systems. While this approach proved more fruitful, it did not make the original problem of the elusive office work any easier.

In the late 1980s and early 1990s, a new development changed the whole field again. Networks and distributed computing made the office worker more mobile. Suddenly offices were no longer stone-and-mortar entities. Offices today are everywhere. Office work has become virtual. Conventional offices with cubicles, desk, and so on still abound, yet the idea of what an office is has become much more flexible. Office work has been distributed to other workplaces, to hotel rooms, to airport lounges, even to the home. As such, office work has merged into the larger effort of *group work.* SIGOIS thus turned into the Special Interest Group on Group Work and Group Ware, SIGGROUP. The change in focus was reflected by SIGGROUP's mission statement:

> SIGGROUP is interested in topics related to computer-based systems that have a team or group impact in workplace settings. A strong emphasis of SIGGROUP is the integration of multiple computer-based tools and technologies and the impact on the human activities supported by those tools and technologies. Relevant issues include design, implementation, deployment, evaluation, methodologies, and impact that arise when researching computer-based systems in a development environment.

Software Systems in Office Automation: Office Information Systems

Many developments from both the research community and the commercial sector have come together over the last two decades to produce the current state of task support in information systems. These developments have not only improved the ability to accomplish office work but have broadened the definition of electronically supported office work.

In this subsection we examine the evolution of office information systems and trace significant improvements and knowledge gained. We explain why OIS evolved from rigidly specified procedures to flexible but knowledgeable systems. We show the influences of artificial intelligence and knowledge engineering. We also show how attention turned to more tool-based, open systems and away from research on organizational aware task management systems. We explain how two developments in the commercial sector, business process reengineering and workflow systems, are showing the need for organizationally cognizant systems, with human and machine agents as partners in communication. Finally, we show the influence of project management in broadening workflow functionality.

Office information systems had their genesis in the mid-1970s to early 1980s. They were originally based on the premise that the flow of paperwork through offices was analogous to the manufacture and assembly of parts through a factory. In the ensuing years, object-oriented technology, collaborative technology, and artificial intelligence made contributions to OIS.

Early OIS. Early work focused on the data, forms, or documents being manipulated through the office [e.g., *Office By Example* (1)]. This was an incomplete method because it did not consider the office workers' activities; for instance, it did not allow the user to specify when a task step was inappropriate (2).

Procedural-based approaches to office work were developed to provide activity management automation. These systems allowed the specification of a series of actions and so addressed routine, repetitive tasks. Scoop (3) was an early method for specifying procedures with augmented Petri nets. Information Control Net (ICN) was a model for mathematically specifying office procedures, and it used roles, actors, and activities in defined relationships. Office Talk-D, for example, was built utilizing ICN at the front end and a database at the back end (4). BDL by Hammer (5) provided a formal language for describing office procedures.

POISE (procedure-oriented interface for supportive environments) (2) formalized defining typical user tasks and how the tasks could be carried out using the tools provided by the system, with the specification of procedures and subprocedures into a predefined procedure tree. POISE provided planning for routine task automation and for proposing actions and goals to recognize actions, propagating constraints to use in error correction, and allowing interrogation and explanation of the system.

A major criticism of OISs of that time was that their rigid structure did not support the reality of office work: that organizations' procedures are not static, that exceptions occur, and that, when they do, people find a way to resolve the problem. While the current rigid structures did provide control over routine, repetitive, linear procedures, they did not allow the flexibility of creating ad hoc workflows from existing procedures and of developing nonlinear procedures.

Another problem surfaced as organizations were collecting and describing, in a formal manner, office procedures with a wide range of granularity. With their monolithic approach of proceduralization, these contemporary OISs did not provide for specifying this necessary range of granularity. In addition, the only way the OISs could acquire a new procedure or a modification to an existing procedure was through end-user programming.

Hewitt (6) identified that offices are not closed systems; they are affected by external forces and information, and therefore the computer information systems that support them cannot be closed. Information coming into an organization is asynchronous, dynamic, and unplanned. Information within and coming into the organization is inconsistent and contradictory and must be rationalized into the organization's belief structure through a process of belief construction and negotiation.

Hewitt prescribes a method of logical deduction within a structure of due process to reach consistency in an organization's beliefs. Part of due process is creating and maintaining a record of the decision- and action-taking processes, which allows evaluation of the decision process and the decision.

Hewitt set a perspective on the situated social nature of organizations' human processes and computer systems and on the need for consistent organizational representation for both human and computer systems.

Responses to Early OIS Problems. Because of these problems with OISs, interest in office information systems subsided and many information systems researchers and industry turned their attention toward simpler, presumably more efficient, tools that would help a person to get the job done (7). This approach, commonly known as the tool perspective, rejected the idea of sophisticated, intelligent systems and replaced it with a focus on the way individuals manipulate artifacts. The fit between artifact and user and the way the artifact is constructed occupied researchers' attention, making office information systems nearly a forgotten field. There was, however, still some research on activity management problems. This research was also informed by developments in artificial intelligence. Three examples are Domino, Ubik, and OASIS.

Domino (8) is a system built in the mid-1980s that monitored and controlled well-structured cooperative tasks in the office. It also allowed free format document transfer via e-mail. It allowed some specific methods of handling exceptions, such as an actor suggesting forwarding information to another or objecting to errors in forms sent to him or her. Relevant messages were exchanged according to certain rules, with the procedure system always in the role of requester, so it was always in control; the user of the system was always the contractor.

From this base, DOMINO was modified in the late 1980s to integrate more flexibility in cooperation into its structure by making action conversation a central aspect of the system. It also enabled users to be requesters of the system and of each other. To handle this coordination of action conversations, conversation types were modeled as high-level Petri nets.

Ubik (9) is a system in which an organization's structure is represented using objects and links between them. Communication is represented as messages sent along the links. The links between objects describe and control the organization's workflow. The workflow itself is specified by sending messages along the links. These messages are also objects. Objects can be varied in the granularity that is represented.

Message-based "tapeworms," attached to objects, monitor and constrain organization structure and action and are capable of identifying mismatches between the representation model and the external (physical) organization. For example, number of deadlines missed appears as a mismatch to Ubik. In this way, Ubik's intelligence can provide aid to ensure that the representation is accurate and useful and aids in process improvement.

Ubik uses sponsor objects to give processing power to other objects, and by this method can control object execution priorities. Sponsors can be organized in four ways: centralized, decentralized, coordinated, and interacting.

Ubik is designed to support nonbiological agents. It is a system that adapted the parallel processing paradigm to organization modeling, extending parallel processing to support distributed processing and distributed intelligence; however, its distributed objects are not restricted to the client-server model.

OASIS (10) integrates communication agents, the use of doers (active objects that perform tasks), reasoning rules, blackboards, forms, and the architecture of MOAPs (micro-organization activity processors) to support a structure of distributed knowledge work. It is modeled to support user control over personal software applications; to integrate data processing, communication, and reasoning; to allow information exchange between an organization's independently developed applications; and to simplify some types of changes within the organization. It is a system that defines a high-level protocol for communication between distributed information systems and provides an interface to allow users to build distributed applications.

OASIS supports organizational activities by using a single, coarse-grained MOAP object to support each organizational unit, with all interactions occurring between MOAPs. This contrasts with Ubik, which uses the actor model of computation and does not tie objects to any organizational unit.

Developments in Other Fields That Influenced Office Automation

While specific office information systems research was going on, developments in the commercial sector (business process reengineering, workflow management systems), in other research disciplines (artificial intelligence and project management), and in technology (local area networking, graphical user interfaces, and the proliferation and availability of PCs/Workstations throughout the workplace) were affecting and influencing the way people work.

Business Process Reengineering. This tool view, which caused the de-emphasis of research in task management system developments, is currently being challenged by business process reengineering (BPR) (11). BPR views the processes in a business not from a single job's perspective, but as transcending the boundaries of the organizational units to follow the process from its triggers and logical start all the way to its final output. Thereby BPR integrates many functions and organizational units that were previously viewed as separate.

The tight relationship between BPR and OIS can most prominently be seen in the early work of Michael Hammer (5). For BPR, describing a business process requires a formal language that has been developed in OIS. BPR and OIS actually emerge from the search for this process description language. While in BPR it is then used to analyze and describe new processes, in OIS it is used to implement these processes in an information system.

With the surging interest in BPR, a renewed interest in embodying the streamlined processes in the organization has resurfaced. Workflow systems are taking the place of office information systems. With BPR's philosophy of not automating rote processes but rather changing processes to take advantage of technology, process reengineering works hand in hand with workflow automation as a way of increasing businesses' efficiencies in today's global competitive business climate. Workflow management systems, however, are often repeating OIS's conceptual shortcomings while hiding them behind a more user-friendly exterior.

Task Support and Artificial Intelligence. Artificial intelligence research has also had its influence on automating office tasks with the direction of intelligent task support. For instance, domain knowledge can be acquired through knowledge engineering (12) and machine learning.

The late 1980s saw the introduction of distributed artificial intelligence (DAI), which provided for the allocation of tasks among distributed agents. The focus was on intelligent cooperative information systems. AI research has provided models for multiagent negotiation that have applicability for workflow management. An example of this is TEAM (13), a model developed in the early 1990s that provides a method for intelligent, heterogeneous agents to negotiate to resolve problems detected when developing interacting subproblem solutions. It was designed for agents to address the problem at hand by sharing information on their problem-solving preferences and abilities.

One research domain for AI was plans and planning. For example, in the 1980s, Callisto (14) provided a structure of formalized knowledge of in the application domain of project management. Pietras and Coury (15) developed a cognitive model project management processes that has applicability over various domains, such as software engineering and construction management.

Planning and Project Management. We have been elaborating on OIS issues, but there are parallel issues in another domain of information-based work, project management. Project activity in the 1990s is becoming increasingly collaborative, facing similar human communication problems and relying on the same productivity tools as OIS.

OIS and workflow are typically associated with repetitive routine processes; however, as mentioned previously, exceptions occur in routine processing and are unique events, becoming like small projects. Conversely, repetitive tasks in a project are becoming repetitive and workflow-like. For instance, the process steps for developing each program in a software development project are similar, if not identical. Just as the domain of project management involves plans, the Regatta model specifies workflows as plans. Like OIS, project management has been influenced by artificial intelligence search.

Regatta (16) is a model for developing software that not only supports work groups but also aids in business process reengineering. The developers of the Regatta model recognized that prior OIS and workflow systems separated the workflow planning process from the actual participants in the workflow. This resulted in inaccurate workflows. The Regatta approach is to capture domain knowledge by enabling the group members to have control over their workflow specifications through collaborative planning. In addition, it allows the users to experiment to find the best process and to incrementally improve the process. It allows individual views of the process. As users grow more sophisticated in their use of Regatta, they can create plans and plan templates to automate their own tasks.

The Regatta model is task oriented. While maintaining a formal flow of responsibility, Regatta is designed to provide flexible access to information, flexibility in exception processing, and an informal flow of communication. It supports negotiation with natural language communication rather than with formal models of negotiation.

POLYMER (12) offered assistance in the creation and execution of business processes to achieve stated goals in real-world domains. This included supporting the specification and refinement of tasks or steps in the workflow to the level of granularity needed, the performance of the actions comprising the workflow, and the verification that these actions are producing the expected results.

One of POLYMER's main aims was to overcome the control versus flexibility dilemma that plagued most office information systems and continues to bother many workflow systems. POLYMER acknowledged that the real world cannot be modeled completely and that it is not at all predictable. Rather than generate accurate and detailed (manual) workflows a priori, people initially generate high-level workflow outlines or abstract workplans based on goals, and refine and revise these abstract workflows as the execution of the plan progresses. POLYMER enabled task workflows to be retrieved for instantiation in new workflows.

The general functionality of POLYMER was to assist all agents participating in a business process, concerning the transactional aspects of the process. POLYMER was based on functionality of the more traditional project management, but went beyond this to cover intelligent task support in general. Like its contemporary systems Domino and Ubik, POLYMER was influenced by the advances that formalisms for describing tasks were making in the field of artificial intelligence. It was also influenced by the opportunities that arose from introducing the power of plan-based expert systems to task support. For example,

1. Tate's NONLIN provided nonlinear (partially ordered) planning in the domain of project management. It was used to model electricity turbine overhaul (17).
2. Wilkins's SIPE introduced human/machine planning. It was used for aircraft carrier mission planning (18).
3. Vere's DEVISOR provided real-time sequencing with protection intervals. It was used for used for *Voyager* space craft mission sequencing (19).
4. Callisto used a series of constraint-directed negotiations to model the specification process and revision process of project management. The series also provided a knowledge-based representation method for project management and other planning tasks (14).

POLYMER enabled agents' actions to be recorded and compared to those anticipated by the system and the effects of these steps on the remainder of the workflow to be computed. When there was a deviation from the initial workflow, these deviations could be explained and recovery steps could be performed to repair the remainder of the workflow.

A system cannot supply this support automatically and autonomously. The burden of workflow management must be shared between a software support system and its users throughout the process of plan formulation, execution, and maintenance. POLYMER allowed the style of interaction between person and workflow environment to vary from a passive approach, in which the system performs recordkeeping and post hoc analysis, to a more active approach, which monitors interactions and alerts the user to anomalous conditions, pending tasks, deadlines, and so on, to partially or fully automated performance of tasks, where feasible.

To go beyond the limited syntactic support provided by existing tools, the developers of POLYMER recognized the importance of cognitively valid and computationally adequate domain models, developed and maintained with a robust language. For this model to be elicited from and presented to task managers and domain agents, POLYMER had to be accessible and understandable to computer-naive users of the support system.

In the late 1980s, D-POLYMER (20) extended POLYMER's single-agent support to multiple, distributed agents. D-POLYMER used a protocol similar to contract nets to manage task sharing among the agents. It used these nets as a representation of the control structure rather than for modeling the domain, such as Ubik (21). The nets are used by the controller to determine the next activity to be performed. Under this protocol a bidding process is used to negotiate and establish preferences for an agent.

D-POLYMER also considered activities throughout cooperative work in any domain that could be classified as generic activities; however, its basis was project management. D-POLYMER addressed distributed planning and sharing of goals; however, the implementation of sharing domain objects and resources, monitoring distributed activities, and supporting conflict resolution was still an open issue.

GROUPWARE AS AN OFFICE AUTOMATION APPROACH

In the late 1980s to early 1990s, computer-supported cooperative work (CSCW) was brought to bear on the problems of OIS. CSCW incorporated research in human interaction in the accomplishment of tasks. It emphasized the social aspects of work. This research addressed conversation transactions through speech acts. It studied group/team activities and the interaction of person, information, and artifacts through enactments.

The initial Winograd and Flores systems (22), such as Coordinator, provided tools to create and manage records of conversations (synonymous with workflow loops). The model built on the speech act theory of interpersonal communication and viewed work as language action. Language action is a higher-level view of the organization.

This model represented a shift from task structure to coordination structure; coordination is the structure, rather than a task. Under this model, elementary loops of action are identified and constructed in which a performer completes an action to the satisfaction of the internal or external customer. Users are constrained to use one of a small number of types of utterances in their computer-mediated communications.

Action workflow was designed with the intent to work across diverse computer system platforms and to be interoperable with various applications. It can initiate, participate, or manage workflow applications.

ConversationBuilder (CB) (23) is a system developed to balance flexibility with structured collaborative support. It is directed to the problem that the greater the flexibility of a system, the less knowledge the system has about users' activities, which therefore reduces the active support the system can provide; and conversely, the greater the potential to provide active support, the less flexibility the users of the system have in performing their activities.

With its concept of protocols, CB implements Winograd and Flores's theory of the equivalency of language and action. Protocols are activities structured as users engaged in conversations. Users' utterances are user actions. Examples of protocols are complex documents, such as collaborative papers, and software source code. Unlike Winograd and Flores's Coordinator, however, CB allows specification of multiple protocols.

The mechanism of obligation connects individual activities within CB. It uses a flow structure of obligations to represent steps of an activity. Under the premise that it is difficult to program procedures in advance because agents and roles change all the time and steps of an activity and the required data are not known in advance, these obligation structures can be edited ad hoc to address the dynamic nature of organizations. CB provides its users with a context switching capability so that they can work on as many simultaneous activities of different types they would like.

Information Lens (24) focuses specifically on the issue of automated support for the systematic, intelligent transference of information. It addresses the problem associated with information sharing: getting information to those who need it without bothering people who do not. It adds a support feature to office work by providing a system that uses rules to distribute messages to interested people and to sort the messages people receive into special folders (in so doing helping people prioritize their activities).

The system is message based but does not have the formalized specification of message objects such as in Ubik (21). For example, relevant *New York Times* articles can be filtered in to a user's folder.

The use of semistructured messages arose from the use of Information Lens. The semistructured messages are templates providing a partial structure, with defaults that can help formulate information in messages. With added intelligence, the system can offer default values to the user, prioritize what the user has received, suggest to the user actions to take upon receipt of a message, and automatically respond to incoming messages.

Because semistructured messages are not rigid, like form-based systems, exceptions to the structure do not bring the system to a halt. In addition, semistructured messages facilitate organizing message types in a frame inheritance network.

With Information Lens, users are given control over the system in that it can be used for support ranging from automated to human controlled.

An Office Automation Perspective on Groupware

CSCW represents a change of perspective in the way we view computers, networks, and the opportunities they represent for people and their work. While the dominating technology-centered view has slowly been challenged by the view of *human–machine interaction,* CSCW introduces *human–human interaction* as the primary focus. In this view the computer is a tool or a medium that facilitates communication and collaboration between humans, rather than acting as a technological computation device.

A number of factors contributed to this change of focus. A first prerequisite factor is the ready availability of computing devices and the extension of these computers as a personal tool to the group. A second prerequisite is the growth of pervasive networking that enables widespread computer-based communication between people and their information appliances. Another factor that spurred the emergence of CSCW is the merging of telecommunications and computing, as network providers seek new applications to exploit bandwidth and widen their service spectrum. Telecommuting, telepresence, mobile offices, and working at a distance further motivated the emergence of CSCW and concepts connected to it.

Before defining the term *computer-supported cooperative work,* we need to clarify a number of terms and issues in the name that may be misleading. CSCW often includes technologies other than the computers. Video, cameras, displays, telecommunication devices, and other devices form the hardware components of CSCW systems in addition to computers. Further, CSCW need not be supportive. Often technology and applications can be disruptive of the individual's work or the organizational context. CSCW technologies are not limited to the concept of work in the strict economic sense; they can assist casual and social interactions for learning, discovery, or pleasure.

Cooperation in this context is a particularly sensitive keyword. CSCW systems may at times foster competition rather than cooperation. Systems that support negotiation between groups are examples of CSCW systems that are employed in such competitive situations.

Another source of confusion is the meaning of the word *cooperation* or *cooperative work* as contrasted with the terms *collaboration, collective work,* and *coordination.* All these "co-words" apply to group situations and the work context. The distinction between them is not very well established in the CSCW community. A specific set of criteria (nonhierarchical, autonomous, etc.) must apply for work to be called cooperative. At the core of this view of work is the notion of interdependence between individuals in their work. For many researchers there is a connotation to the term *cooperative* that assumes compliance, shared sentiments, and so on. Collective work, on the other hand, is seen by many as a more general term for a group's working in a social context that is not as narrow as the preceding definition of cooperation calls for.

Coordination often refers to the arrangement of individual tasks according to a number of constraints, such as time or resources, while collaboration often connotes the close interaction of two individuals. Over the last few years the term *collaborative work* in the context of CSCW has come to mean any type of work that involves more than one individual.

Definitions and Taxonomies

The term *computer-supported cooperative work* was first used by Irene Greiff and Paul Cashman at CHI (Computer Human Interaction Conference of the ACM) in 1984. It was the title of an interdisciplinary workshop on how to support groups of people in their work arrangements with computers. Since then CSCW has come to mean the large field that is concerned with computer-assisted activity that is carried out by a number of collaborating individuals. While the term *CSCW* was felt by many to be unwieldy and ill defined, it has not disappeared but stuck to a field whose boundaries are still unclear. CSCW has become an umbrella term under which people from many disciplines could discuss aspects of system design and use by more than one user.

Liam Bannon and Kjeld Schmidt (25) define CSCW as "an endeavor to understand the nature and characteristics of cooperative work with the objective of designing adequate computer-based technologies," while Lucy Suchman (26) defines it as "the design of computer-based technologies with explicit concern for the socially organized practices of their intended users." From this discussion it should have become clear that CSCW is still an evolving field that covers a wide spectrum, ranging from social and organizational research to the development and deployment of computer-based technologies.

CSCW is often used synonymously with the term *groupware.* That term was originally coined by Peter and Trudy Johnson-Lenz (27) to mean "intentional group processes and procedures to achieve specific purposes plus software tools designed to support and facilitate the group's work." Generally, the CSCW community has adopted a slightly different view. CSCW is viewed as the overall scope of research, development, implementation, deployment, evaluation, and so on, while groupware is viewed as the technology that supports group work. Esther Dyson (28) notes that "groupware is about as useful a term as singleware." She goes on to state, "More than a way of coding or building applications, groupware is a way to define, structure, and link applications, data and the people who use them." Ellis, Gibbs, and Rein (29) emphasize the task and goal aspect of groupware: "Computer-based systems that support groups of people engaged in a common task (goal) and that provide an interface to a shared environment." This goal-focused view of task interaction is augmented by Lynch, Snyder, and Vogel (30): "Groupware is distinguished from normal software by the basic assumption it makes: groupware makes the user aware that he is part of a group, while most other software seeks to hide and protect users from each other. . . . Groupware . . . is software that accentuates the multiple user environment, coordinating and orchestrating things so that users can 'see' each other, yet do not conflict with each other." Finally, Thomas Malone (24) defines groupware as "information technology used to help people work together more effectively." Groupware is thus the combination of multiuser software and hardware that supports the activities of more than one person to accomplish a task.

In addition to the terms *CSCW* and *groupware,* a number of related concepts are often used. We frequently find the terms *work group computing* or *team computing.* These terms arose to denote the extension of personal computers to small groups. The focus here is the size of the group. Work group computing is directed at a closely knit set of individuals working together. This small group (four to twelve people) is frequently referred to as a *team* (thus the term *team computing*). This distinction makes sense when it is contrasted with applications that are written to be used enterprisewide (many hundred users) or even for the Internet (thousands of users).

Extending the size of the group turns work group computing into *organizational computing.* Organizational computing spans a large variety of settings (e.g., domains as diverse as business, hospital administration, professional organizations, or the military). Organizational computing is often used by researchers who are concerned with the impacts of computer and communication technology on organizational design, operations, and performance.

Collaborative computing is a term used to cover research in computer-supported cooperative work and computer-mediated communications. People in this field particularly emphasize the development of computer and communication technologies to support group work. In this sense it is rather close to groupware, since the design of cooperative systems and group support technology is a central theme in collaborative computing. Topics in collaborative computing cover groupware architectures and environments, the theoretical basis of collaborative systems, models of collaboration, evaluations of and experiences with collaborative systems, the relationship between collaborative systems, user interface tools and techniques for collaboration, and the use of multimedia technologies for collaboration.

CSCW as a multidisciplinary field had input from many different sources. Among the major origins are computer and information science, sociology, organizational design, and human-computer interaction. Each one of these fields maintains a special interest or flavor in CSCW. Organizational design is often associated with organizational computing, while computer and information science is close to collaborative computing. Arising from the tradition of human-computer interaction and user-interface design is a niche in CSCW that is

Time \ Space	Co-located	Distributed
Synchronous	Electronic classroom Interactive liveboards Whiteboarding Workstations	Video Desktop Large screen Whiteboarding Content sharing Telepresence
Asynchronous		Discussion boards Case annotation E-mail Structure discussion

Figure 1. Time/space matrix for CSCW.

concerned with multiuser interfaces. This work approaches CSCW from the system's component that connects the user to the application. Multiuser interface explicitly acknowledges that the interface and the application(s) behind it are shared by a number of users. This requirement imposes a number of design constraints and demands on the interface. The accommodation of multiple pointing devices, the questions of what to show to all the users and how often to update their displays are issues that arise from this perspective.

To organize and systematize the various views and systems that sprung up in CSCW, Malone (24) developed a widely used 2-by-2 matrix (Fig. 1). This matrix uses *time* and *space* as the two major dimensions to differentiate between systems and applications. Each dimension has two values, which are *same* and *different.* According to this simple classification, activities can happen in the same space or in different spaces or activities could happen at the same time or at different times. Group members can collaborate at the same time or work whenever they please. The team can meet in a conference room; the team could be located in offices on different floors or even in different buildings spread over the country or the face of the earth. Usually the terms *synchronous* and *asynchronous* are used to denote same time versus different time activities. On the space dimension the terms *co-located* and *distributed* are used to indicate same space versus different space. The cell representing co-located and asynchronous settings is frequently left blank since it is easily subsumed by the more powerful asynchronous and distributed cell. Once the time constraint is relaxed, it is no longer necessary to return to the same space. Even though examples for this type of work exist (shift work, war rooms), none of them feature prominently in CSCW.

This basic taxonomy is easily extended. First the space dimension can be seen as *group proximity.* The values on this dimension then become *multiple individual sites, one group site,* and *multiple group sites.* This classification was developed in particular for electronic meeting rooms by Nunamaker et al. (31).

A further extension can be created by adding group size as a third dimension. Usually small groups or project teams of three to seven members are distinguished from larger task forces above seven. In the business world it makes sense to distinguish between groups or teams, departments, divisions, units, and the whole enterprise or organization. Outside the business context different ways have been developed to categorize groups based on the number of members and the basis for the groups foundation. Research by McGrath (32) has led to the following enumeration of social aggregations:

- Artificial aggregations: statistical group, or social category
- Unorganized aggregates: audience, crowd, public
- Units with patterned relationships: culture, subculture, kinship group
- Structured social units: society, community, family
- Deliberately designed social unit: organization, suborganization, crew, project group, team
- Less deliberately designed social unit: association, peer group, circle of friends

Another important dimension along which CSCW and groupware can be categorized are the varying application domains. A meeting system used in education will differ from a meeting system used in a business context. Engineering tasks usually require different tools than administrative tasks. The following list of task and application domains can only serve as a first attempt and needs to be augmented as new systems are built and applied to an ever-growing set of domains:

- Engineering
- Administration
- Manufacturing
- Research and development
- Banking and insurance
- Logistics and warehousing
- Decision making
- Teaching and learning
- Authoring
- Drafting
- Entertainment
- Software engineering
- Project management
- Military planning and operations

WORKFLOW AS AN OFFICE AUTOMATION APPROACH

Definition of Workflow Systems as Office Automation

Workflows can be defined as the structure that is applied to the movement of information and objects in order to execute business processes. Workflow systems are groupware applications designed to support activities, which can range from routine, repetitive procedures in organizations to very complex and highly unique projects. Workflows are created by workflow planners (process engineers, business planners, management consultants, etc.); are implemented and deployed by industrial engineers, system integrators, and other such professionals; and are executed by human and machine agents, usually the professional of an organization, who are involved in producing the core products or services. Workflow software actively manages the coordination of activities among people in general business processes.

Workflow systems originate from three system sources:

1. Office automation efforts in the 1970s
2. Imaging-based systems in the 1980s
3. Recent developments in CSCW and groupware

Imaging systems and office automation aimed at automating office paper processing. Many workflow systems today, therefore, are highly integrated with one of the two types of systems, which makes a general definition fairly abstract. Only recently have workflow systems emerged that were built directly on a workflow paradigm.

The relationship of OIS, CSCW, and current imaging products to workflow systems needs to be understood to avoid the hidden traps and mistakes that have been made in the past (e.g., lack of exception handling, no end-user acquisition of activity descriptions, little networking across local area networks [LAN], and absence of ad hoc workflows). The field of task mapping and task support, tackled by OIS, CSCW, and workflow systems, is extremely difficult, as problems have had a tendency to reoccur in new systems in a different form or shape yet remain essentially unsolved. Goal-based workflow systems offer an avenue to overcome the persistent problems in task support that have resurfaced.

The major components for a workflow context can be found in Ref. 33:

1. organizational structure and context
2. individual workers and aggregations of workers (departments)
3. tools (in this case workflow software and other technology)
4. processes (the maps according to which workflows happen)
5. business goals (the organizational goals to be accomplished by processes)
6. product (the final service or thing produced as a result of the processes done by the individuals)

Form-Based Workflow. The original commercial workflow systems were influenced by the first generation of OIS and were conceived as form publishing and document imaging management. First-generation workflow emphasized routing and indexing capabilities of digitized of document images. The forms basis is still apparent in many of contemporary commercial office task support systems. Form-based workflow uses rule-based logic to systematize and speed up business processes, moving and coordinating data within the work group.

FileNet's WorkFlow made its appearance in the mid-1980s. It was one of the first generation of workflow products and enabled electronic flow of digitized documents from one processing step to the next. It improved the speed of document movement through the office (34).

In Delrina Corporation's FormFlo (version 1.0) electronic forms package, it is easy to set up and to produce forms. FormFlo provides good security levels. To improve speed, it does not use open database connectivity (ODB) but rather uses native support to improve speed. While it has features that easily re-create exact copies of existing paper forms, the programming capabilities are not for the average end user.

Word Perfect Corporation's InForms (version 1.0) is an electronic forms package in which the forms are stored in proprietary format. Its strength is in its word processing features and security. It is not easy for nonprogrammers to use.

Process-Based Workflow. Business process reengineering has heavily influenced the direction of workflow systems, many of which are based on the process of the workflow rather than on its content (the document).

Action Technologies' Action Workflow Manager (version 1.01) uses a highly structured analysis and design methodology and is also highly structured in how it is specified. It meets process-oriented needs, but a workflow cannot be modified once it is implemented; the user must build new workflow.

While Action Workflow can generate and manage forms, it is based on the business process structure and its improvement. Before the workflow is specified to the system, the work is analyzed to determine what is actually being done. The workflow is analyzed for improvements, new functionality, new ways to satisfy customer, and for problems or breakdowns. Action Workflow considers workflow as the interaction of customer and performer during the four phases of preparation, negotiation, performance, and acceptance (22).

Xerox's InConcert (34) combines workflow management with document management. It is developed on the philosophy of easily obtaining necessary information to perform a task and therefore uses document and data residing in standard file systems rather than in its own database storage. It takes advantage of current technologies, such as graphical user interface and object-oriented programming, and an open architecture. It integrates third-party productivity tools such as spreadsheets and word processors. InConcert uses a job-based model for business transaction and collaborative processing. It allows reuse of workflows and dynamic workflow modification.

IBM Corporation's FlowMark also takes advantage of the latest technologies, with its object-oriented database and ability to handle any file format, including images and multimedia presentation. Both process logic and workflow are defined using a graphical diagramming module that consists of icons for building diagrams and notebooks or data stores that contain specific process details, such as the definition of how an action should be performed and the assignment of activities. It also provides simulation capabilities (33).

A business process engineering feature of Wang Laboratories' Open/Workflow is that it allows the user to refine and optimize workflow processes continuously by monitoring process efficiency and providing graphical tools for modifying the workflow. To aid process improvement, Open/Workflow also comes with a toolkit of precoded business measurements, such as time and cost benchmark tests. Like FlowMark, it can handle a variety of data types, such as images, voice, and video.

Lotus Development Corporation's Lotus Notes (releases 3.0 and 4.0) is a multiuser, collaborative product that combines a flexible, document-oriented database structure with e-mail, routing, and processing. This basic database architecture can handle both free-formatted and highly structured data, with control of access to both databases and documents. Its flexibility allows a form in one document to retrieve data from a form within the same database, from a different Notes database, or from an external database (using Lotus's Datalens drivers). Notes also allows object linking and embedding. Most recently, Domino has allowed Notes to act as a database back end to Internet servers and brings limited Notes capability to the World Wide Web.

While Notes was not developed specifically for workflow management, it can be used for workflow with the help of expert, dedicated developers. Its version control capabilities make it suited for collaborative workflows. Neither Notes graphical design tools to develop workflows nor its modest macro language are easy to work with; the latter has inconsistent syntax and lacks control structures such as loops. The Notes form-design interface is old fashioned and awkward to use, and Notes lacks sophisticated reporting and analysis features for workflow.

The usage of LotusNotes add-on FlowMaker (version 1.0) by WORKflow will help overcome some of the difficulties of workflow specification for an experienced Notes programmer. It is used once the desired workflow has been designed. The programmer then specifies this workflow to FlowMaker in outline form and also creates a Notes database of application-relevant forms. FlowMaker will then generate Notes macro instructions that can then be pasted into a Notes application. For this benefit of less tedious Notes programming, FlowMaker requires that the flow be described in terms of roles, moves, and states (as in state of the document), along with the conditions that need to be satisfied before the work can move on to the next state. FlowMaker also aids in debugging because events are all visible.

Evaluation of Current Workflow Systems

Contemporary workflow systems are an improvement over previous generations of OIS; however, many of the conceptual and social problems of OIS have resurfaced in these current workflow systems.

Control Versus Flexibility. Most prominent among those problems is the question concerning the balance between giving agents control and providing agents with flexibility. While a system that controls many aspects of the workflow can provide a wide range of immediate support, this advantage is bought at the price of limited flexibility. The system can only provide a wide range of functional support if it makes assumptions about the structure and type of tasks that occur in a workflow.

Exception Handling. Changes, exceptions, and even slight deviations do not fit into the framework of the system. One of the best-known examples for this rigidity is the widely reported case of a phone number appearing in the customer number slot in an order form. While this slight deviation from the routine can be easily detected and corrected by a person (35), a rigid system is completely thrown off.

Current workflow systems have an appeal on the surface, both visually and by virtue of the fact that they are better than previous generations of task management systems. The issues are as follows: Do these workflow systems have the ability to address conceptual problems which research spanning more than a decade has revealed? Will they be tools that can truly assist agents in their work?

Goal-Based Activity Management. The research cited in this article indicates that, in order for a workflow to be flexible, a goal must be specified in terms of the desired state of the task domain for each step in the workflow. This goal must then be refined until it is clear what actions must be performed to achieve the specified goal. Current PC-based workflow systems are not goal based and therefore will not be able to provide the flexibility needed by dynamic organizations.

Domain Knowledge and Intelligent Planning Support. Current workflow systems are lacking in domain knowledge specified in a way that it is useful for intelligent support in planning, execution, monitoring, and modifying workflow. Having these capabilities allows the fullest automated support of workflow management. Agents may choose not to take advantage of this option of full automation support. Current workflow systems, without intelligent workflow management support, do not give agents the opportunity of this time-saving functionality.

CONCLUSIONS

The previous sections have shown that office automation has considerably changed its promise and focus over the last two decades. Starting with the notion of automation, it has moved, via task support, to a tool view of group support in distributed environments.

From the dedicated support of single office tasks, such as faxing or printing, work support environments have moved to a more integrated office view. Today we can fax from a PC that is connected to a network using fax-server software. (Actually, the need to fax has decreased since more and more people have e-mail and can receive electronic documents as attachments, which in turn enables them to view and manipulate those documents). A completely different solution for a different set of cooperative work needs are document management systems and discussion databases, whereby documents can be exchanged and stored.

With continued advances in telecommunications (in particular, wireless services and increased bandwidth), with increased availability of high-volume, low-cost storage, and with accompanying social contributions, such as telecommuting and virtual teaming, office automation, will continue to play a major role in our technological surroundings.

BIBLIOGRAPHY

1. M. Zloof, Office by example: A business language that unifies data and word processing and electronic mail, *IBM Syst. J.,* **21**: 1982.
2. W. B. Croft and L. Lefkowitz, Task support in an office system, *Trans. Office Inf. Syst.,* **2** (3): 197–212, 1984.
3. M. Zisman, *Representation, specification and automation of office procedures,* Ph.D. thesis, Wharton School, Univ. Pennsylvania, 1977.
4. C. Ellis and M. Bernal, OFFICETALK-D: An experimental office information system, *Proc. ACM SIGOA Conf. Office Syst.,* 1984.
5. M. Hammer et al., A very high level programming language for data processing applications, *Commun. ACM,* **20** (11): 832–840, 1977.
6. C. Hewitt, Offices are open systems, *ACM Trans. Office Inf. Syst.,* **4**: 271–287, 1986.

7. P. Ehn and M. Kyng, A tool perspective on design of interactive computer support for skilled workers, *7th Scandinavian Res. Seminar Syst. Eng.,* Univ. Aarhus, 1984, pp. 211–242.
8. G. Woetzel and Th. Kreifelts, Die Systemarchitektur des Vorgangssystems DOMINO-W, Gesellschaft fuer Mathematik und Datenverarbeitung, TR FM-GMD-85-23, 1985.
9. P. De Jong, The ubik Configurator, *Proc. ACM COIS-88,* 1988.
10. C. C. Woo and F. H. Lochowsky, Supporting distributed office problem solving in organizations, *ACM Trans. Office Inf. Syst.,* **4**: 185–204, 1986.
11. M. Hammer and J. Champy, *Reengineering the Corporation,* New York: Harper Business, 1993.
12. L. S. Lefkowitz and W. B. Croft, Planning and execution of tasks in cooperative work environments, *Proc. 5th IEEE Conf. Artif. Intell. Appl.,* 1989.
13. B. Laasri et al., A generic model for intelligent negotiating agents, *Int. J. Intell. Cooperative Inf. Syst.,* **1**: 1, 1992.
14. A. Sathi, T. E. Morton, and S. F. Roth, Callisto: An intelligent project management system, *AI Mag.,* Winter 1986.
15. C. Pietras and B. Coury, The development of cognitive models of planning for use in the design of project management systems, *Int. J. Man-Mach. Stud.,* 1993.
16. K. D. Swenson et al., A business process environment supporting collaborative planning, *Collaborative Comput.,* **1**: 1, 1994.
17. A. Tate, *Project planning using a hierarchic non-linear planner,* Ph.D. dissertation, Department Artif. Intell., Univ. Edinburgh, 1976.
18. D. E. Wilkins, Domain independent planning: Representation and plan generation, *Artif. Intell.,* 1984.
19. S. Vere, Planning in time: Windows and durations for activities and goals, *IEEE Trans. Pattern Anal. Mach. Intell.,* **PAMI-5**: 246–267, 1983.
20. N. Bhandaru and W. B. Croft, An architecture for supporting goal-based cooperative work, *Proc. 1990 IFIP 8.4,* 1990.
21. P. de Jong, Methods for organizational development, *Proc. ACM Conf. Organizational Comput. Syst.,* 1993.
22. R. Medina-Mora et al., The action workflow approach to workflow management technology, *CSCW'92 Sharing Perspectives,* 1992.
23. S. Kaplan et al., Flexible, active support for collaborative work with conversation builder, *CSCW'92 Sharing Perspectives,* 1992.
24. T. W. Malone et al., Semi-structured messages are surprisingly useful for computer-supported coordination, *Trans. Office Inf. Syst.,* **5**: 2, 1986.
25. L. Bannon and K. Schmidt, CSCW: Four Letters in Search of a Context, in J. Bowers and D. Bendford (eds.), *Studies in Computer Supported Cooperative Work: Theory, Practice, and Design, Proc. 1st Eur. Conf. Comput. Supported Cooperative Work,* Amsterdam: North Holland, 1991, pp. 3–16.
26. L. Suchman (ed.), *CSCW '88: Proc. Comput. Supported Collaborative Work,* Portland, OR, 1988, pp. 26–29.
27. Johnson-Lenz, P. and T., Groupware: The Process and Impacts of Design Choices, in Kerr and Hiltz (eds.), *Computer Mediated Communication System: Status and Evaluation,* New York: Academic Press, 1982.
28. E. Dyson, *A Framework for Groupware,* 1992, pp. 148–154.
29. C. Ellis, S. Gibbs, and G. Rein, Groupware: Some issues and experiences, *Commun. ACM,* **34** (1): 38–58, 1991.
30. K. J. Lynch, J. Snyder, and D. Vogel, *The Arizona Analyst Information System: Supporting collaboration research on international technology trends,* 1990, pp. 159–174.
31. J. Nunamaker et al., Electronic meeting systems to support group work, *Commun. ACM,* **34** (7): 40–61, 1991.
32. J. McGrath, *Groups and Human Behavior,* Englewood Cliffs, NJ: Prentice-Hall, 1984.
33. D. Mahling and R. King, A goal-based workflow system for multiagent task coordination, *J. Organizational Comput. Electron. Commerce,* in press.
34. M. A. Bragen, Go with the flow, *PC Mag.,* July 14: 253–302, 1994.
35. K. Abbot and S. Sarin, Experiences with workflow management: Issues for the next generation, *Proc. ACM CSCW 1994 Conf.,* 1994.
36. L. A. Suchmann, Office procedures as practical action, *ACM Trans. Office Inf. Syst.,* **4** (4): 320–328, 1984.

DIRK E. MAHLING
University of Pittsburgh

OFF-LINE PROCESSING. See BATCH PROCESSING IN COMPUTERS.