Designing Coordination Systems for Distributed Teamwork

Nikolay D. Mehandjiev, Anthony A. Karageorgos, Gee Hin Tsang
Department of Computation, UMIST
Manchester. M60 1QD, UK
{mehandjiev,karageorgos}@acm.org

Abstract

Contemporary working environments are increasingly complex, volatile and physically distributed. Coordinating work in such environments is therefore becoming a major challenge as the traditional focus on efficient schedules is shifting to include other factors such as rapid reaction to market changes.

This paper argues that effective coordination support should be based on a two-level coordination model supported by agent-based implementation infrastructure. Operational coordination processes take place at the inter-level, while the meta-level is concerned with managing the inter-level coordination processes by considering dynamically changing goals and factors.

This division is at the heart of our novel approach to work coordination presented here. A case study of distributed telecom repair teams is used to illustrate the approach.

1. Introduction

Information and communication technology enables accomplishment of work to become independent of time and place. People now work at various locations and participate in the work of one or more organizations. Organizational forms are increasingly based on distributed working practices and localized decision-making. This offers many advantages for both the workers and the organizations but induces considerable coordination costs. At the same time, traditional coordination mechanisms based on predefined organizational constructs such as formal structures, procedures, methods and plans, are no longer satisfactory in a business environment of globalization and rapid change [20]. New mechanisms are needed and a large number of organizations adopt various forms of flexible and decentralized work management (cf. [6] Chapter 16), leading to flatter, coordination-based organizational structures such as network organizations [15] and virtual organizations [17]. All coordination intensive organizations are characterized by local control over job allocation and day-to-day work planning and control. Examples of organizational forms of this type are autonomous teams [6], and autonomous work groups [3]. These organizational forms have been found to allow rapid response to change at both local and global levels [14], to reduce job turnover rate [7] and to increase worker productivity and effectiveness [4]

We focus on organizations where members are not co-located, and are thus dependent on technology to provide them with the information and interactions necessary for autonomous decision-making and work coordination activities. We call this type of organizations *distributed*.

We aim to provide effective coordination support for such organizations. We argue that an effective coordination mechanism should take into account a number of factors that can change rapidly and impede coordination processes, yet are not covered by established coordination modes such as the Coordination Theory [15]. This paper details a two-level coordination framework that takes these factors into account when analyzing and modeling coordination activities. We then proceed to describe how such a framework can be implemented using an agent-based infrastructure.

The rest of this paper is organized as follows. A case study of telecom field engineers is outlined below to help in further discussion. It is followed by a description of a number of factors and their impact on coordination mechanisms. This leads to a description of our two-level framework, consisting of a formal coordination model and an agent-based architecture. Both are explained by a detailed example based on the case study. Review of related work and discussion conclude the paper.

2. Case Study: Coordinating Field Engineers

All fieldwork at the customer service teams of a major UK telecommunications company is coordinated by their in-house operational support system (OSS). The OSS is based on a combination of large monolithic applications, including a personnel database. The system processes the requests for work for the next day to produce a work schedule for the field engineer repair teams.

A task is dispatched to a worker by sending a message to his or her hand-held terminal. The task allocation is done based on the location, tools and skills of the engineer as well as the job priority. Upon finishing a task, the worker informs the OSS which in turn updates its records. For every customer service team, a local manager is regularly informed about the course of the work and is ready to intervene when required.

3. Factors which Affect Coordination

A number of factors affecting effective coordination have been identified, grouped into the following three groups: *human, environment and work context factors*.

Human factors can in turn be classified into:

(a) Social: given a choice, workers will often make decisions that affect the group advantageously as a whole, rather than making decisions which benefits them the most in the short term. An example might be a worker that agrees to do overtime so that a fellow colleague can have time off, or perhaps not negotiating for a certain job if they know a colleague wants it.

(b) Cultural: modern workforces are often multi-national and multi-cultural. Taking into account cultural preferences such as taking the Chinese New Year off instead of the usual New Year when coordinating work can be very beneficial for workforce morale.

(c) Personal: personal likes and dislikes, for example, can be quite important for workforce motivation. Sending a worker who fears dogs to repair equipment in a kennel is an example where personal factors should be considered.

Environmental factors refer to factors in the natural environment over which the business has little or no control, *e.g.* the accessibility of transport routes and weather conditions. For example, if the center of York were flooded, travel and hence work would become impossible for a large majority of commuters.

Work context factors affect the business in how it operates, which includes general company policies and strategic decisions. One example might be a company's hiring policies, which may change when new legislation is enacted such as ensuring the company hires workers fairly, regardless of ethnic or racial origin.

These factors are significant when considered either separately or in combination. Consider a market-based coordination process where a worker may bid for a specific task based on their personal preferences and working practices, or apply for overtime to enable a colleague to have time off. Previous experience has shown that workers may not bid for problematic tasks that they have done before. The coordination process is affected in all of these cases, and the effect of social factors can be further changed by the introduction of bidding rules and policies.

In addition, an unreliable network link could result in allocating tasks manually, since no bidding will be able to take place. If workers submit no bids for particular type of tasks then these tasks may need to be allocated using a different coordination process like direct task allocation.

4. Effect of Factor Changes

Traditional approaches to coordination are focused on effective work scheduling and optimized for a particular type of situation. When the situation changes, they are not optimal any longer. If teams work together, shortcuts can be devised and coordination recommendations overridden to adapt to changes in situational factors. However, coordination in distributed teams often relies on computer-mediated support in the absence of direct communication. Dynamic adaptation of work coordination processes to suit the current situation would thus be of significant benefit for these teams.

It is perhaps feasible to construct a general coordination approach that considers the factors that determine a situation at run-time, treating them as just another type of input in addition to work activities and their dependencies. This would result in one super-algorithm that would handle all possible contexts. However, this approach has the following two problems:

- Every combination of possibilities should be catered for. In an ideal world, this may be possible. In the real world, business environments change rapidly and any single algorithm would rapidly become outdated.
- The size and complexity of the algorithm would make it extremely difficult to update or modify.

We propose an approach for coordinating work, which recognizes two levels of coordination processes - inter-level and meta-level. At the inter-level, regular coordination processes manage interdependencies between activities to achieve some coordination goals. At the meta-level, dynamic changes in the environmental factors are re-examined in order to optimize the coordination processes and goals from the inter-level.

The advantages from the two-level approach include:

- *Openness*: There is no restriction to the number or type of coordination processes that can be applied. The complexity and the dynamism of contemporary open environments can be tackled by selecting appropriate coordination processes at the meta-level.
- *Explicitness*: Operational coordination functionality is separated from the factors that can affect it strategically. This reduces the complexity of each level, and allows separate explicit descriptions of how coordination processes should be modified under dynamic changes in the environment.
- *Objectivity*: Separating the selection of a coordination processes from the actual work coordination formally excludes from consideration the extrinsic factors that are not relevant to the goals or activities that need to be coordinated.

- *Responsiveness*: When conditions change, an appropriate coordination process can be selected.
- *Effectiveness and efficiency*: A number of modularly extensible coordination processes can be developed and maintained, which are optimized for specific situations and combinations of factors.

5. Description of the Two-Level Framework

We formalize our two-level approach to work coordination into a framework, consisting of a formal model and an agent-based architecture for designing coordination support systems**.** The model builds upon the theoretical foundations of Malone's coordination theory [15] and Mowshowitz's theory of virtual organization [16].

5.1. Formal model of two-level coordination

Malone and Crowston [15] stress that characterizing situations in a way that helps generate and choose appropriate coordination mechanisms is an open issue. We tackle this issue by proposing the following formal model for analyzing coordination processes using dynamically updated situational factors:

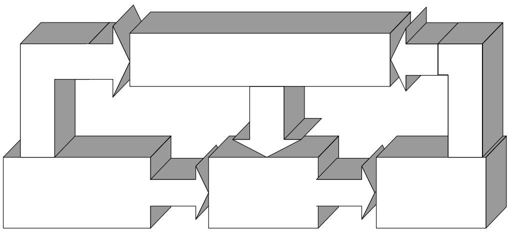

Figure 1: Our Two-level Coordination Framework

A work coordination process *C* can be determined as operating within the tuple (*A, Ac, D, F, G*), where:

A is a set of activities;

Ac is a set of actors carrying out the activities;

D is a set of dependencies between activities, including shared resources and ordering;

F is a set of factors affecting the coordination process;

G is a set of goals that need to be achieved.

The description of the meta-level of coordination decision-making processes (DMP) from Figure 1 is then given by *Ci=f(F,G,D)*, that is the selection of the coordination process *Ci* out of a set of available coordination processes *{C}* is a function of the factors *F*, goals *G* and dependencies *D* in the given situation. A coordination process *Ci* can be composed out of two or more coordination processes as evident in the example further down. The application of the mapping function *f* will be triggered by a significant change in the observed tuple *(F, G, D).*

Once selected at the meta-level, the coordination process *Ci* operates at the inter-level *to* achieve a set of goals *G*. The goals may be interdependent (for example conflicting) with each other, and therefore the degree at which they need to be achieved must be able to be measured either quantitatively or qualitatively. The coordination process *Ci* operates by structuring the interactions of the set of actors *Ac* and manipulating the set of interdependent activities *A* and their dependencies *D* as exemplified further down.

Note that we have extended the framework of Malone and Crowston in three main dimensions:

- Workers are no longer passive resources to be allocated to activities, but separate actors capable of negotiation and decision-making. They interact within the coordination processes *C* to determine sequencing of work activities.
- Environmental and human factors are as important as nature of dependencies and goals in determining which coordination processes is to be used.
- A meta-level of coordination is introduced where coordination processes are selected.

In the next sub-section we will proceed to describe the decision-making and coordination processes within the proposed two-level framework.

5.2. Processes in the two-level framework

In determining the decision-making and coordination processes in our two-level coordination framework, we can learn from research in goal-driven organizations [13]. The theory of *virtually organizing* [16] is of a particular interest since it formalizes management under constant change as a dynamic process of matching abstract requirements to concrete satisfiers.

To apply this theory to our work coordination model, we will have to consider the presence of autonomous actors *Ac* and the activities *A* in which they play a part. *An inter-level coordination process C* can then be described as matching the abstract requirements represented by the set of coordination goals *{G}* to concrete satisfiers represented by the work activities *A* and actors *Ac* whilst taking into account dependencies *D*. Because of the pro-active behavior of actors *Ac*, however, some coordination processes such as allowing workers to bid for jobs, for example, are structuring interactions between actors *Ac* rather than directly allocating work *A* to actors *Ac*. Notably, such a coordination process will be conducted in a distributed fashion as a set of interactions between autonomous entities. This motivates the use of software agents as an implementation technology as described in the next section.

Looking at the *meta-level of our model,* we see that the main decision-making process (*DMP*) there is tasked with selecting a suitable coordination process for the inter-level when a significant change occurs in environment and human factors F or in the organizational goals G.

meta-level DMP

F

Dynamically changing environments are indeed considered by the theory of *virtually organizing* [16], by introducing the activity of *metamanagement* based on the *principle of switching*. According to the principle of switching, before a task is executed, the procedure that needs to be followed is reconsidered and possibly changed as required. In our case, before coordinating work activities, we need to re-examine inter-level coordination processes in the light of changing factors. Dynamic switching offers many advantages including facilitating efficient use of resources, enhancing organizational responsiveness and promoting organizational reflection.

The DMP at the *meta-level*, therefore, can be seen as performing switching of inter-level coordination processes *C*. It senses dynamic changes in the tuple *(F, G, D)*, and selects/changes the operational coordination processes *C* at the inter-level using the following steps:

Step 1: Set trigger levels for changes in F, G and D.

Step 2: When triggered by a significant change, identify a set of candidate coordination processes.

Step 3: Rank the available coordination processes based on the factors and the goals.

Step 4: Select and enact a coordination process.

Since factors and goals may change dynamically, Steps 2 to 4 are repeated every time a significant change is sensed. For example, coordination processes need to be re-considered each time the workforce is updated.

5.3. Implementation Architecture using Agents

In this section we propose a top-level architecture to be used in developing coordination support systems within our two-level framework.

Software agents are often characterized as encapsulated computer systems situated in a certain environment that are able to act autonomously and flexibly in that environment to meet their design objectives [23]. Considering additional agent properties this core definition can be specialized. Such properties include mobility (static or mobile), deliberative thinking (proactive or reactive), learning, co-operation and role that agents play in a larger application context [18].

We have chosen software agents as an underlying implementation technology for the following reasons:

Decentralized nature of inter-level coordination processes C and the prominent role of actors *Ac* which have their own personal beliefs, act autonomously aiming to achieve their goals and negotiate to resolve conflicts. In these circumstances, where distributed coordination arises from interactions between software modules, the software model needs to be based on synthesis and construction [9] rather than decomposition or reduction.

The meta-level DMP reasons with abstract conceptual models of the working environment. To minimize the semantic distance [9] between these conceptual models and the design of the software that would implement the coordination processes, we need software constructs with pro-active behavior close to the notion of actor.

Finally, to meet the *dynamism of the open environment*, the software needs to support actors in both proactive and reactive manners not necessarily specified in advance. This involves constructing plans for future behavior and reacting to triggers of the environment as appropriate.

Using software agents as a basic building block, we have constructed an architecture for building agent-based support of two-level coordination, described on Figure 2.

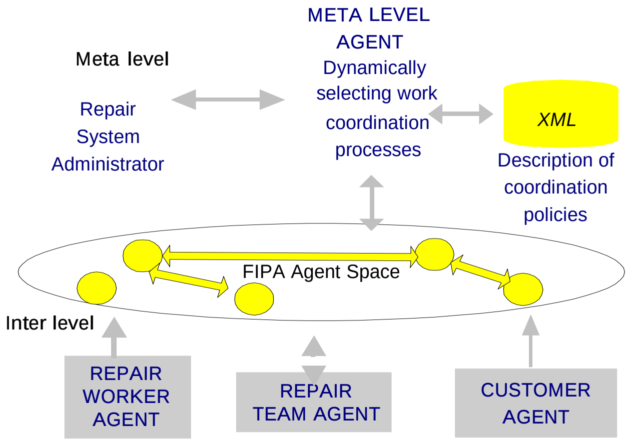
META LEVEL AGENT
Dynamically selecting work coordination processes
Meta level
Repair System Administrator
XML
Description of coordination policies
FIPA Agent Space
Inter level
REPAIR WORKER AGENT
REPAIR TEAM AGENT
CUSTOMER AGENT

Figure 2: Top-level agent architecture

In the core of our approach is the facility for software agents representing actors such as repair workers and customers to coordinate their behavior using explicit representation of coordination processes and policies.

Representing coordination policies can be done by proprietary XML files, using a standard specification such as WS-Coordination, or using a generic service description formalism such as DAML-S. We are currently exploring the suitability of WS-Coordination and DAML-S for specifying coordination mechanisms specified as agent interaction protocols. These specifications will be selected by the meta-level agent according to a number of factors. Once selected, the chosen specification will be parsed by the inter-level agents and will be used to guide their behavior. Our current mechanism for loading behavior in agents uses the Behavior facility incorporated in the Java-based agent development environment JADE.

Our future plans include implementing support for the scenario described earlier, and deploying it on our node of the Agentcities distributed agent-based platform (`http://agentcities.co.umist.ac.uk`).

6. An Example

To show how our proposed approach and analysis framework can be applied in a complex work situation we consider an example scenario drawn from the telecom case study, characterized with the tuple (*A*, *Ac*, *D*, *F*, *G*):

Activities *A* include telephone repair, exchange repair, telephone line installation, telephone line conversion, etc.

Actors *Ac* that carry out the activities A include repair workers, customers, team managers, call-center staff.

Dependencies *D* between A, *e.g.* a local exchange needs to be upgraded before high-speed internet access can be installed for a customer.

Factors *F* affecting the coordination process, for example physical conditions and geography as explained further down in this example and,

The set of goals *G* includes the provision of a good quality of service while maintaining high work efficiency.

A set of coordination processes *{C}* at the inter-level of coordination will include allocating tasks to worker by:

(C1) customer,
(C2) functional specialization
(C3) product
(C4) by geographical region
(C5) workers bidding for certain jobs they like.

Changes to *Ci* due to dynamically changing factors can then take place within the following five example states.

State 0: Scheduling by functional specialization
Originally repair teams are allocated repair tasks by some work scheduling software without their participation. Repair tasks are allocated according to workers functional specialization (C2).

State 1: Workers bidding for jobs
Torrential autumn rain (a environmental factor F) causes localized flooding in the area. Repair engineers have to work in flooded areas. Unions point out that this is not in the contract regarding conditions of employment, and insist that workers are given a choice not to take up the jobs in the flooded area. Consequently, workers are allowed to bid for the latter type of jobs (C5 coordination process), and can choose their reward for doing so: extra payment as overtime, or less jobs in their daily workload.

State 2: Scheduling by geographical region
The rain continues with some storms, and renders all bridges in the area unusable. The river is now swollen and dividing the geographical area in two parts, the connections between them requires a 70 miles de-tour. As a result, the coordination mechanisms need to be changed once again and are now based on the area where the repair worker lives (C4). C4 and C5 are now active simultaneously, composed into a new coordination process C45.

State 3: Bidding for jobs (C5) is withdrawn.
The waters recede, and the bidding for jobs is withdrawn, but the coordination by area (C4) is left in place until the bridges are repaired.

Stage 4: Switching in coordination and dependencies because of company policies.
A few months later the problems caused by the flood no longer exist and therefore work coordination returns to normal (based on C2 as in stage 1). However, at this point the company introduces a policy related to a recent government anti-competitive ruling. This requires allowing access of competitors to the company's core equipment. The policy recognizes the dangers of this for the company, and stipulates that a local manager has to be presented with a plan of every repair task involving such access, and he or she should visit the site at the completion of the repair task, and approve the work. This results to changing the work coordination mechanism again by introducing a new dependency which requires the presence of the manager for a particular class of upgrade activities..

7. Relevant Work

Work can be coordinated in many ways and therefore a large diversity of systems to support work coordination has evolved. The approaches can be classified according to many variables including the degree of human participation in the coordination process. The range of available systems is very wide extending from fully automating coordination decisions [11] to simply facilitating human communication so that coordination takes place via direct human interaction [8]

Traditionally, work coordination systems have been focused on the most efficient scheduling of the actions of the workers, often based on dynamic scheduling techniques applied within the context of workflow management systems [5]. The workflow management approach involves fully automated and computer-driven coordination. Such techniques however, although optimizing scheduling efficiency result to decreased worker motivation [19].

Facilitating human interaction is another means of supporting work coordination. This category includes systems that provide synchronous and/or asynchronous communication capabilities including e-mail, whiteboards, teleconference and videoconference. By using these technologies, workers interact and work is coordinated based on well-known management techniques [12]. Human-driven coordination has the advantage of enabling human participation to work coordination and hence increasing motivation. However, it is very time consuming since it requires a lot of direct interaction among humans. This approach can be met in cooperative work support systems where the emphasis of the system is providing a means for carrying out the work instead of coordinating it.

Relaxing the dynamic scheduling constraints and then reconsidering the unique, “correct” workflow process stance is an approach that has been followed in an increasing number of systems. The aim of this approach is to increase worker motivation by enabling user involvement in the coordination and by improving the quality of working life of workers. There are many ways to enable user participation including gathering and providing information to the users [2],[3], supporting sharing of experiences by maintaining an organizational memory [1],[10] and providing users with the capability to customize workflow processes according to their needs [21][22]. However, these systems still fall short of providing full support for effective coordination, for example they do not formally support team building and informal collaboration among workers. Furthermore, to our knowledge, no system supports two levels of dynamically changing coordination processes.

8. Conclusions

The novel two-level coordination framework proposed here provides a powerful generic approach to coordinating work in contemporary environments, where new factors and coordination processes can be identified and dynamically incorporated into work coordination activities. Agents are considered suitable for implementing support for such framework, and corresponding architecture is described in this paper.

Many conceptual issues require further examination in parallel to the technical plans for further work outlined earlier. An issue of major concern is of course the design of a generic work coordination framework where new factors and coordination processes could be identified and incorporated in the coordination dynamically. As the next step towards this direction we plan to investigate the optimal distribution of coordination functionality within a coordination support system based on software agents and will formulate a set of design principles. We will do this by conducting a set of simulation experiments to identify the strengths and weaknesses of various agent organizations, and evaluate our design recommendations**.**

9. REFERENCES

[1] Ackerman, M.S. and C.A. Halverson, Re-Examining Organizational Memory. IEEE Computer, 2000. 43(1).

[2] Busbach, U., Activity Coordination in Decentralized Working Environments, in Remote Cooperation: CSCW Issues for Mobile and Teleworkers, A. Dix and R. Beale, Editors. 1996, Springer-Verlag: Berlin. p. 95-112.

[3] Carstensen, P., K. Schmidt, and U.K. Will. Supporting Shop Floor Intelligence. in Proceedings of GROUP 99: International Conference on Supporting Group Work. 1999. Phoenix, Arizona: ACP Press.

[4] Cotton, J.L., Employee involvement: Methods for improving performance and work attitudes. 1993, CA: SAGE.

[5] Dangelmaier, W., S. Kress, and R. Wenski. Telework under the Co-Ordination of a Distributed Workflow Management System. in Proceedings of {GROUP} 97: International Conference on Supporting Group Work. 1997. ACM Press.

[6] Gates, B., Business at the Speed of Thought. 1999, Harmondsworth, Middlesex, England: Penguin Books.

[7] Greenberg, J. and R.A. Baron, Behavior in Organizations: Understanding and Managing the Human Side of Work. 1995, Englewood Cliffs, New Jersey: Prentice-Hall.

[8] Haywood, M., Managing Virtual Teams: Practical Techniques for High-Technology Project Managers, Sixth edition. 1998, London: Artech House.

[9] Jennings, N.R., On Agent-based Software Engineering. Artificial Intelligence, 2000. 117: p. 277-296.

[10] Kristoffersen, S. and F. Ljungberg, {MobiCom}: Networking Dispersed Groups. Interacting with Computers, 1998. 10: p. 55-65.

[11] Lesaint, D., et al., Engineering Dynamic Scheduler for Work Manager. BT Technology Journal, 1998. 16(3).

[12] Linpack, J. and J. Stamps, Virtual Teams: Reaching Across Space, Time and Organizations with Technology. 1997, New York: John Wiley.

[13] Mahling, D.e. and R.C. King, A Goal-based Workflow System for Multiagent task coordination. Journal of Organizational Computing and Electronic Commerce, 1999. 9(1): p. 57-82.

[14] Malone, T.W., Is 'empowerement' Just a Fad? Control, Decision-Making and Information Technology. BT Technology Journal, 1999. 17(4): p. 141-144.

[15] Malone, T.W. and K. Crowston, The Interdisciplinary Study of Coordination. {ACM} Computing Surveys, 1994. 26(1): p. 87-119.

[16] Mowshowitz, A., Virtual Organization: A Vision of Management in the Information Age. The Information society, 1994. 10(4): p. 267-288.

[17] Mowshowitz, A., Virtual Organization. Communications of the ACM, 1997. 40(9): p. 30-37.

[18] Nwana, H.S., Software Agents: An Overview. Knowledge Engineering Review, 1996. 11(3): p. 205-244.

[19] Sachs, P., Transforming Work: Collaboration, Learning and Design. Communications of the ACM, 1995. 38(9): p. 36-44.

[20] Schmidt, K. and C. Simone, Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. Computer Supported Cooperative Work: The Journal of Collaborative Computing, 1996. 5: p. 155-200.

[21] Siebert, R. An Open Architecture for Adaptive Workflow Management Systems. in Must write full citation here. 1999.

[22] Swenson, K.D., et al., A Business Process Environment Supporting Collaborative Planning. Journal of Collaborative Computing, 1994. 1(1): p. 15-34.

[23] Wooldridge, M., Agent Based Software Engineering. IEE Proceedings in Software Engineering, 1997. 144(1): p. 26-37.