# Models of Coordination

Robert Tolksdorf

Technische Universität Berlin, Fachbereich Informatik, FLP/KIT,
Sekr. FR 6–10, Franklinstr. 28/29, D-10587 Berlin, Germany,
*mailto:tolk@cs.tu-berlin.de, http://www.cs.tu-berlin.de/~tolk*

**Abstract.** While software is becoming decomposed in more and more finegrained entities, the interactions amongst those become of major importance. While methodologies for building such components are well established, the design and support of their interplay can not build on commonly understood and well defined models. In this paper, we review several coordination models from various disciplines, and describe how a coordination reference model could look like. We use a set of characteristics of coordination models to compare the reviewed ones.

## 1 Looking at Models of Coordination

Todays software is structured into modules, objects, components, agents etc. These entities try to capture rather small conceptual abstractions and supports it with functionality. While this is advantageous for the design and implementation of software, networked environments add additional benefits when running programs composed from those entities. Given a good encapsulation, they can be distributed or mobile, and different non-functional characteristics such as fault-tolerant or persistence can be attributed to them.

And in fact, the tremendous advances in network technology in terms of availability, cost and functions, has impact on how software is composed. Brereton et al., 1999 state: "Software will be fine-grained. Future software will be structured in small simple units that cooperate through rich communication structures and information gathering."

While the implementation of these small units is is commonly well understood and methodologies for their design become more and more adopted, the question on how the cooperation amongst them is designed and supported. Kahn et al., 1999 point out this change of focus: "The overall applications challenge, from a processing and communications viewpoint, is how to implement complex 'ensemble' behavior from a large number of individual, relatively small sensors."

Todays technologic state-of-the-art gives a first impression of the size of the challenge. With small devices being enhanced by processing power and its miniaturization one can expect, that within the next few year, the number of units coordinating will increase by several magnitudes. For example, a GPS receiver device was announced in SiRF Technology, Inc., 1999: "SiRFstarII will enhance location accuracy by an order of magnitude – from 100 meters to between 2 and 15 meters. SiRF has included a 32-bit, full-function, widely supported CPU onto the SiRFstarIIe chipset [...]. The CPU is a 32-bit, 50 MHz ARM processor, with fully 90 percent of the throughput available for non-GPS tasks." The chipset mentioned has the size of a US-quarter coin.

Taken to an extreme, miniaturization will lead to computing devices that are very small in size and that will be very cheap to produce in large numbers. They have certain characteristics, that lead to what is described in Abelson et al., 1999 as "[...] the challenge of amorphous computing: How does one engineer prespecified, coherent behavior from the cooperation of immense numbers of unreliable parts that are interconnected in unknown, irregular, and time-varying ways?"

It is the interaction of computing units that makes their composition useful. Enabling for a useful interaction is coordinated activity. And in order to provide technology that supports the interaction and its design, models of coordination are necessary. These models have to have certain qualities such as being complete wrt. interaction forms and open to new patterns of interactions. They have to be easy and safe to use to facilitate efficient software engineering. They must be scalable and efficient to implement to cope with the number of units to coordinate. And finally, the models have to be aware of the characteristics of future environments, eg. be robust to failures and dynamics.

Coordination models can be used to build middleware and coordination languages. Given a set of coordination primitives, they can be used to express coordination strategies that lead to the mentioned coherent behavior of interacting entities. Thus coordination models are enabling for the execution and design of coordinated applications. If it is possible to find commonalities amongst coordination models, then coordination patterns used in the various disciplines could be made transferable.

In this paper we take a first step towards that goal and look at a variety of existing coordination models from different sources. We evaluate them wrt. a set of qualities and ask how a coordination reference model – the meta-model of the coordination models reviewed – could look like.

## 2 Sources of Coordination Models

Models of coordination are widespread and come in a variety. A large set of domains and disciplines has developed their own models on how entities interact, and do in part also provide technologies that take advantage of these models. Examples are:

- Daily life naturally suggests that independent entities cooperate. Thus, *everyone* has some – diffuse – understanding of what coordination is and should at least be able to tell uncoordinated behaviors from coordinated ones.
- In *computer science*, parallel programming has been one of the first fields in which coordination models were developed. The fields of distributed computing has added further models (Chin and Chanson, 1991).
- *Distributed AI* is concerned with the design of coordination in groups of agents and uses a variety of models.
- *Organization theorists* try to predict the future behavior and performance of organizations. As in these the interaction of actors is a key factor, coordination models are used.
- *Economics* looks at how interactions in markets takes place and models them.
- *Sociologists* and *Psychologists* try to explain the behavior of groups composed by individuals (Gillette and McCollom, 1995).

– *Biologists* study natural phenomena in natural agent systems such as ant colonies or swarms and try to discover the embodied coordination mechanisms (Bonabeau et al., 1999).

This incomplete list already indicates that there is a huge variety of coordination models. But there is currently no consensus on the relations between coordination, communication and cooperation. Although these individual models each are able to explain important characteristics of collaboration, coordination and communication, the definitions used are often in conflict.

In order to compare and integrate such models, it is necessary to work towards a standardized terminology which contains terminological definitions and clarifications of basic notions including the very term "coordination". Based on that, a conceptual model has to relate those terms in the most general and flexible manner. With such building blocks, a uniform representation of collections of coordination patterns would be enabled and the patterns could be implemented coordination services. In Klusch et al., 1999 these four dimensions are considered a road map towards a reference model for communication, coordination, and cooperation.

## 3 Models of Coordination

In this section we examine several coordination models. We begin with the naive view on coordination and then examine two models coming from an organizational perspective, namely the model on organizational structure by Mintzberg and the coordination theory model from Malone and Crowston. We briefly look at formal models and distributed artificial intelligence. Finally, we examine two views on coordination from computer science, the Linda model by Gelernter and Carriero and the workflow reference model by the Workflow Management Coalition.

### 3.1 Naive Model

A common understanding of coordination is that active entities coordinate to achieve a common goal, as depicted in figure 1.

This understanding assumes a common goal shared by all entities. Also, the entities are assumed to be willing to cooperate, that is to follow the goal. And, all entities have to be aware that they cooperate with others and they have to know how they to that.

From an architectural perspective this means that coordination is not encapsulated external to the agents involved and is therefore not interchangeable.

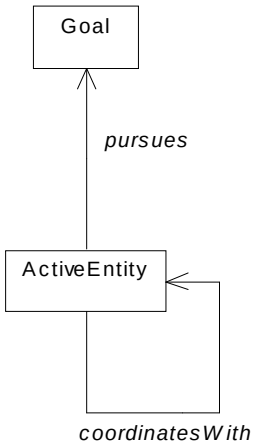



**Fig. 1.** The naive model of coordination

### 3.2 Mintzberg Model

Mintzberg, 1979 is a seminal work on structures of organizations, coordination mechanisms used therein and dominant classes of configurations of organizations.

Mintzberg develops a theory on the structure of organizations by postulating five basics parts which can be found in any organization. As depicted in figure 2, at the basis of the organization is the *operating core* where the actual work is performed. At the (hierarchical) top of organizations is the *strategic apex* – managers who have the overall responsibility for the organization and that take strategic decisions and guide the direction of the organization. The *middle line* is a chain of managers that implement the decisions by supervising subordinates and reporting to their supervisors. The *technostructure* serves to analyse and organize the work done. *Support staff* includes all the indirect support of work, eg. by running a plant cafeteria.

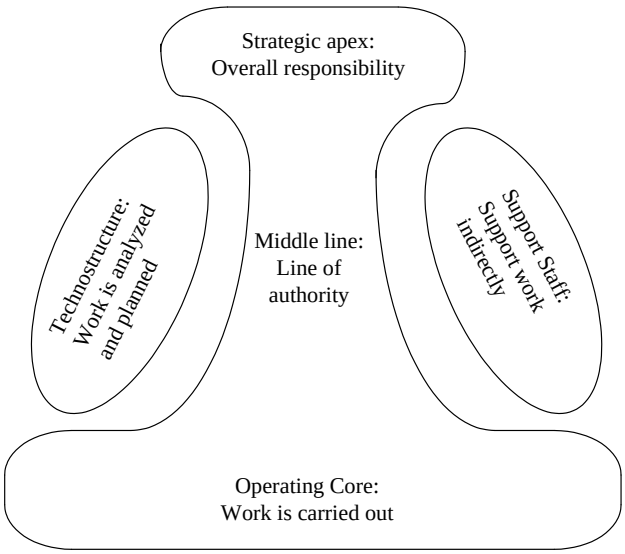



**Fig. 2.** The structure or organizations

Coordination in organizations is explained by five mechanisms as shown in table 1:

1. *Mutual adjustment* builds on informal communication amongst peers. There is no outside control on decisions and peer coordinate their work themselves.
2. With *Direct supervision*, a supervisor coordinates the work of its subordinates by giving instructions.

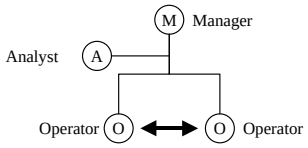
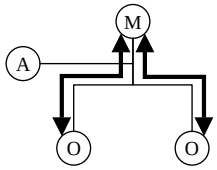
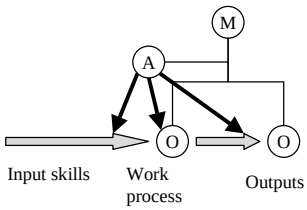
| | |
|---|---|
| Mutual adjustment:<br>Informal communication |  |
| Direct supervision:<br>One is responsible for others |  |
| Standardization of work<br>Standardization of outputs<br>Standardization of skills |  |

**Table 1.** The five coordination mechanisms in organizations (Mintzberg, 1979)

3. *Standardization of work* ensures coordination by specifying the work to be done so that no decisions have to be taken later.
4. *Standardization of outputs* refers to specifying the result of work, thus these can be used by others without additional coordination.
5. *Standardization of skills* specifies the training necessary for a specific work. Additional coordination then is not needed, as peers know what to expect from each other.

From the resulting design space for organizations, Mintzberg selects five configurations of coordination mechanism and preeminent part in the organization as in table 2.

The Mintzberg model assumes a role model for actors in an organization. The choice of coordination mechanisms is induced by the choice of the organizations structure and thus not easily exchangeable. Actors are very aware of the coordination mechanism they have to use. Mintzberg discusses patterns of transitions amongst the different configurations, but these tend to be rather slow.

### 3.3 The Coordination Theory Model

Malone and Crowston, 1994 introduces the term *coordination theory* to "refer to theories about how coordination can occur in diverse kinds of systems." It draws on differ-

| Name | Coordination mechanism | Key part |
|---|---|---|
| Simple structure | Direct supervision | Strategic apex |
| Machine Bureaucracy | Standardization of work processes | Technostructure |
| Professional bureaucracy | Standardization of skills | Operating core |
| Divisionalized form | Standardization of outputs | Middle line |
| Adhocracy | Mutual adjustment | Support staff |

**Table 2.** The five structural configurations of organizations (Mintzberg, 1979)

ent disciplines such as computer science, organization theory, management science and others.

As shown in figure 3, the model defines coordination as *the management of dependencies amongst activities*. In order to make an interdisciplinary use of coordination mechanisms found in various kinds of systems, the processes involved in the management of dependencies have to be studied.

In order to do so, the kinds of dependencies have to be analyzed and the respective processes be studied (Crowston, 1991, Dellarocas, 1996). Figure 4 shows such a hierarchy of kinds of dependencies and processes that manage them.

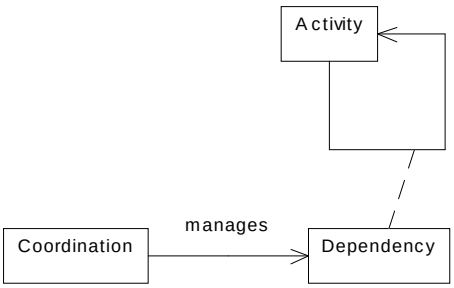



**Fig. 3.** The coordination theory model

The model assumes that the dependencies are external to activities studied. It is not the activities that are managed, but relations amongst them. The model abstracts from the entities that perform activities and their goals.

### 3.4 Formal Models

In this subsection, we briefly look at formal models of coordination. We follow the overview in Ossowski, 1999 and refer to this source for a closer description.

In centralized formal models, there is a known global set of entities to be coordinated. The state of each wrt. coordination activities is modeled by a decision variable. Thus, the system to be coordinated is represented by a set of decision variables $V = \{v_1, \ldots, v_n\}$ (with values from a set of domains $D = \{D_1, \ldots, D_n\}$). Any coordination process leads to an instantiation $x$ of decision variables from decision space $X$.

In quantitative formal models, a global utility function $U : X \rightarrow \mathbb{R}$ is associated with each of these instantiations that models how good the system is coordinated. The model can be analyzed to find an optimum $y \in X$ such that $\forall x \in X : U(x) \leq U(y)$, meaning that there is no instantiation that provides better coordination.

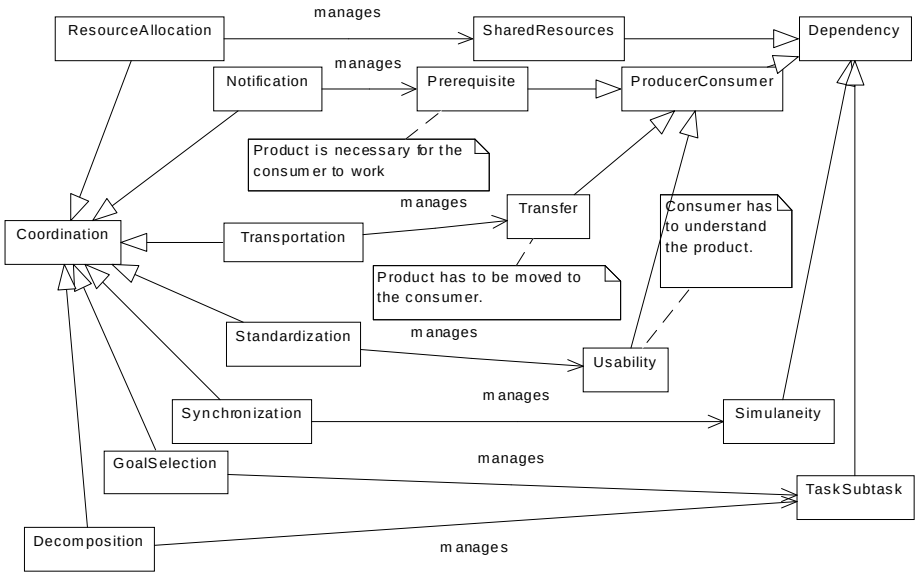



**Fig. 4.** A hierarchy of dependencies and management processes

For a qualitative mode, a set $D$ of constraints $\{C_1, \ldots, C_m\}$ is used. The notion of a consistent instantiation $x$ of decision variables from decision space is defined by $X$ as $y \models C_1 \wedge \ldots \wedge C_m$.

Game theory provides a model of decentralized cooperation. The set of entities to be coordinated is modeled as a game which consists of a set $I$ of $n$ players. There is a space $S$ of joint strategies $S = S_1 \times \ldots \times S_n$. It collects the individual strategy $S_i = \sigma_{i_1}, \ldots, \sigma_{i_k}$ that each player has.

In contrast to the global utility function of the quantitative model above, a set $P$ of payoff functions is defined for each player individually by $P_i : S \to \mathbb{R}$.

Two kinds of games are distinguished. In zero-sum games the payoff of one player is "financed" by lower payoffs of the others: $\forall \sigma \in S : \sum_{i=1}^{n} P_i(\sigma) = 0$. In non-constant sum games, this restriction does not exist: $\exists \sigma, \sigma' \in S : \sum_{i=1}^{n} P_i(\sigma) \neq \sum_{i=1}^{n} P_i(\sigma')$.

The situation can be analyzed in two ways. In non-cooperative analysis, the players try to get the best payoff they can individually. The set of strategies is said to be in a Nash equilibrium, if deviation from it by one player will not increase that players payoff: $\forall i \in I : \forall \sigma_i \in S : P_i(\sigma_1^*, \ldots, \sigma_i, \ldots, \sigma_n^*) \leq P_i(\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_n^*)$. In a cooperative analysis, the players coordinate strategies and join payoffs. The situation is said to be Pareto-optimal if no one can achieve a higher payoff without lowering that of some other player.

All these formal models share some assumption. First, they make a fundamental assumption about the environment, namely that payoff and utility can be defined exact and static. Second, they assume that agents behave exactly rational to maximize utility or payoff.

### 3.5 Coordination mechanisms in DAI

Jennings, 1996 asserts that the key to understanding coordination processes to to look at the internal structures of agents. There, commitments – pledges of agents about actions and beliefs in the future or the past – and conventions – general policies on reconsiderations of commitments – are determent for coordination mechanisms. In addition, social conventions give policies on interactions in a community of agents and local reasoning is necessary to use that knowledge.

In Computational Organization Theory, roles are defined that constrain behavior of agents. In Multi-agent Planning, commitments are based on plans that agents develop. In a multi agent society setting, negotiation is the coordination mechanism by which agents take joint decisions after following some negotiation protocol.

The models assume that the conventions governing the coordination processes are external to the agents. They assume commitments a priori to events that take place, and thus assume knowledge about future events. Also, agents act rational.

### 3.6 Uncoupled Coordination

In parallel computing, questions on how to organize the execution of multiple concurrent threads in a computation has led to several coordination models. The model embodied in the language Linda (Gelernter and Carriero, 1992) takes the view that coordination has to be performed explicitly by the parallel processes and that it is worthwhile to use a separate language for that. Such a coordination language focuses merely on the expression of coordination and defines a respective coordination medium (Ciancarini, 1996).

The language Linda embodies a model of uncoupled coordination as depicted in figure 5. Here, a set of agents together form an ensemble in which they coordinate their interaction indirectly by using a shared dataspace, called the *tuplespace*.

The coordination language is defined in terms of operations that access and modify the tuplespace. The operation *void out(Tuple t)* emits a piece of data – a *tuple* which is a list of values of some primitive data types – into the tuplespace. To retrieve it, one uses the operation *Tuple in(Template t)*, which takes a *template* of a tuple to describe what kind of tuple is sought. A template can contain actual values or placeholders denoting only the type of a value expected in some field of the tuple. The operation blocks until a tuple that matches the template is emitted, removes that tuple from

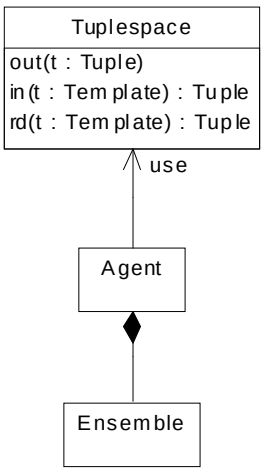



**Fig. 5.** The Linda model

the tuplespace and returns it to the agent that issued the *in. Tuple rd(Template t)* basically does the same, but leaves a copy of the matching tuple the space.

The model assumes explicit coordination amongst agents that have to use the coordination language in an appropriate way. The pattern of coordination is scattered over the use of the coordination operations with the agents.

The model provides an abstraction from the location of agents in space and time. Agents remain anonymous to one other and do not necessarily have to exist at the same time. Also, it abstracts from the computational model and programming language used by the agents.

### 3.7 Workflow

Workflow Management Systems (WfMS) coordinate human work and its support by applications. In recent years, there has been an urge to define a common denominator of such workflow modeling languages to enhance interoperability amongst systems of different vendors. The Workflow Management Coalition (WfMC) is the industry consortium of the leading WfMS vendors and has published a reference model. Part of it is a process definition language (Workflow Management Coalition, 1998) that represents a minimal language to express workflow models.

Here, a workflow is modeled as a graph of activities as nodes and transitions between them. The transitions represent dependencies amongst activities and can be augmented with additional constrains, such as start- and end-times. The topology of the graph includes coordination constraints on activities.

So called AND-JOIN- and AND-SPLIT-nodes synchronize activities or span new parallel activities. An XOR-JOIN makes the execution of an activity dependent on the termination of one out of several other ones. XOR-SPLIT selects one new thread of activities to start. The flow of activities can be further specified by introducing loops and sub-workflows.

As shown in figure 6, activities are performed by participants in the workflow and might involve data and applications. The participants are constituent for the organization in which the workflow takes place.

The approach taken by most WfMS assumes that all activities and dependencies amongst them are known in advance. Also, reliable execution of activities is assumed – at least the reference model lacks the notion of exceptions.

## 4 Towards a Coordination Reference Model

In this section, we outline how a coordination reference model could look like, define attributes for coordination models and review the models from the preceding section wrt. those.

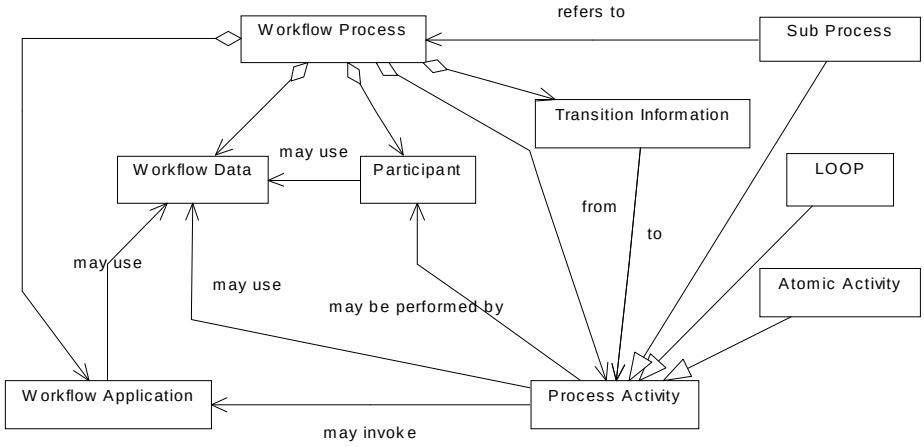



**Fig. 6.** The reference model of the WfMC

### 4.1 Structure of a Coordination Reference Model

We propose the following constituents for a coordination reference model. As shown in figure 7, four model layers (see Kobryn, 1999) seem of interest to us.

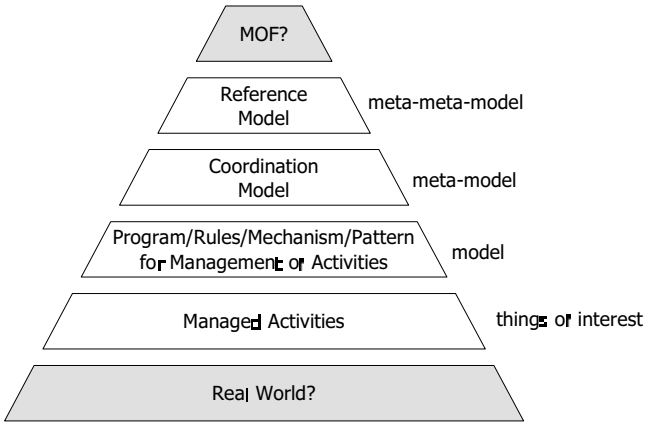



**Fig. 7.** A structure for a coordination reference model

While the real world knows situations in which coordination is neither present nor necessary, we are interested only in that part of the world in which activities are managed in order to be coordinated. These form the object-level in our model-hierarchy.

For a specific set of such objects, the concrete management of activities is described by a set of rules, specific mechanisms, programs or a selection of coordination patterns. Thus, they are models of specific managed activities – the blueprints for actual interactions.

The coordination models described in the preceding section each provide a specific framework to express such models. Thus, they are meta-models above the specific models that describe specific managements of sets of activities.

The coordination reference model that we are interested is a meta-model to the coordination models. It contains terminologies and concepts to describe coordination models.

The reference model itself also has a meta-model that describes for example, what a concept or a term is. For our purposes, we are not interested in that model layers, and would make use of some existing meta-model, such as the OMG MOF.

Within the reference model, the following set of concepts seems necessary:

- *Interactors* are those entities that are related to other interactors.
- *Relations* associate two or more interactors in some way. Coordination mechanisms then apply to relations amongst interactors.
- *Non-Interactors* are those entities that are related to interactors or to none.
- *Operations* can be performed by interactors on non-interactors.
- *Attributes* can be assigned to Interactors and non-interactors do describe them or their current state.
- *Meta-Attributes* describe the models built from those concepts wrt. their characteristics.

As an example of how these concepts describe a coordination model, we can look at a part of the Mintzberg model from organization theory.

- *Manager*, *Line-Manager*, *Analyst*, *Operator*, *Support staff* are instances of *Interactor*.
- *controls* is a *Relation* put forth by the model and relates a manager with an operator. There is no further coordination mechanism described than the enactment of that relation.
- *input*, *output*, *skills* are *Non-Interactors*.
- An operator can increase his/her skill by an *operation learn*.
- *Span of control* is an *attribute* of a line-manager.

### 4.2 Comparing Coordination Models

We believe that all the models reviewed can be described by the above reference model, which is enabling for their comparison. In order to compare them, we now introduce a set of meta-attributes of the models. Highlighted are those characteristics that we consider dominant for a model.

- To what degree the model provides a *clear distinction* amongst interactors, non-interactors and management of relations.

- How *orthogonal* the coordination model is with computational models used by interactors.
- The degree of *coupling* between interactors. This also includes coupling to an external goal (modeled as a non-interactor).
- The degree of *autonomy* of interactors. It is inverse to the degree of centralization assumed or introduced by the model.
- Whether the management of relation is *external* to interactors or not.
- How much *awareness* to management of relation is required from interactors.
- The degree of *stability of interactors* assumed by the model on the interactors, eg. whether there is a static set of them or not.
- The *stability of relations* assumed by the model, ie. long- vs. short-term relations. Longterm means the lifetime of the situation in which coordination is necessary. Midterm means the lifetime of one interaction and shortterm refers to single coordination activities within an interaction.
- How much *reliability* is assumed about the interactors
- The *scalability* provided by the model wrt. the number of interactors and relations amongst them.
- How *usable for programming* a coordination model is.
- Whether there are qualitative or quantitative *measures* on the management of relations provided.

Wrt. those dimensions, table 3 shows a comparison of the models reviewed.

For the naive model, the existence of a shared goal is the dominant characteristic. It induced a high coupling and determined the stability of interactors and relations.

The Mintzberg model is characterized by the assumed stability of the relations, which matches steps in the life-cycle of an organization. The model shows a high flexibility as it offers multiple coordination mechanisms The coordination theory model knows about an extensive set of possible coordination mechanisms and models their relation towards dependencies to be managed. This is the dominant characteristic of this model. Stability of dependencies is assumed to be midterm, as one interaction dissolves a dependency.

Formal models provide at their core measures of how optimal a situation is coordinated. They tend to be centralized and thus are not well scalable.

DAI models assume reasoning of agents about their own coordination activities, thus they are highly aware of coordination. However, this also leads to scalability problems. The Linda model is characterized by low coupling of interactors in space and time which imposes low requirements of stability of interactors and relations. Dominant for workflow models is the assumption that interactors and relations – and thus coordination procedures – are stable of a long time.

The overview thus shows that each model can described by dominant characteristics along our dimensions. Also, it shows that the models reviewed here cover very good the scale of those dimensions.

| Model | Distinction | Orthogonal | Coupling | Autonomy | External | Awareness | Interactors stability | Relations stability | Reliability | Scalability | Programming | Measures |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Naive | No | Yes | High | **Low**: Shared goal | No: Entities coordinate | Yes | Midterm: Until goal reached | Midterm: Until goal reached | Yes | No | No | No |
| Mintzberg | Yes: Inputs and outputs in addition to actors | Yes | Dep. on mecha-nism | Low | No | Yes | Midterm | **Longterm** | No | Yes | Unclear | No |
| Coord. Theory | Yes | Yes | **Dep. on mecha-nism** | Mid | Yes | Yes | Midterm | Midterm: Until dissolved | Yes | No | Unclear | No |
| Formal | No | No: Evaluation of functions in model | High | Low | No | Yes | Longterm | Longterm | Yes | No | Yes | **Yes** |
| DAI | Yes | Yes | Mid | High | Yes | **Yes** | Midterm | Midterm: Horizon of reasoning | Yes | **No** | Yes: Agent systems | No |
| Linda | Yes | Yes | **Low** | Mid | Yes | Mid | **Shortterm** | **Shortterm** | Yes (see Tolksdorf and Rowstron, 2000) | Unclear (see Menezes et al., 2000) | Yes | No |
| WfMC | Yes | Yes | High | Low | Yes | Yes | **Longterm** | **Longterm** | Yes | No | Yes | Yes |

**Table 3.** Comparing the coordination models

## 5 Conclusion

In conclusion, we have seen that the variety of coordination models is large and draws on various disciplines. After reviewing them, we have tried to outline a reference model of coordination which could be capable of serving as a meta-model to those reviewed.

Wrt. to a set of characteristics of models, we found that the models are well-distinguishable along those dimensions. We found that each model has a dominant characteristic. We also found that the set of models covers substantial parts on the scales of the dimensions considered.

## References


Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T., Nagpal, R., Rauch, E., Sussman, G., and Weiss, R. (1999). Amorphous computing. Technical Report AI Memo 1665, MIT AI Lab.

Bonabeau, E., Dorigo, M., and Guy, T. (1999). *Swarm Intelligence*. Oxford University Press.

Brereton, P., Budgen, D., Bennnett, K., Munro, M., Layzell, P., MaCaulay, L., Griffiths, D., and Stannett, C. (1999). The future of software. *Communications of the ACM*, 42(12):78–84.

Chin, R. S. and Chanson, S. T. (1991). Distributed object-based programming systems. *ACM Computing Surveys*, 23(1):91–124.

Ciancarini, P. (1996). Coordination Models and Languages as Software Integrators. *ACM Computing Surveys*, 28(2):300–302.

Crowston, K. G. (1991). *Towards a Coordination Cookbook: Recipes for Multi-Agent Action*. PhD thesis, Sloan School of Management, MIT. CCS TR# 128.

Dellarocas, C. N. (1996). *A Coordination Perspective on Software Architecture: Towards a Design Handbook for Integrating Software Components*. PhD thesis, Massachusetts Institute of Technology.

Gelernter, D. and Carriero, N. (1992). Coordination languages and their significance. *Communications of the ACM*, 35(2):97–107.

Gillette, J. and McCollom, M., editors (1995). *Groups in Context, A New Perspective on Group Dynamics*. University Press of America.

Jennings, N. (1996). Coordination techniques for distributed artificial intelligence. In O'Hare, G. M. P. and Jennings, N. R., editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons.

Kahn, Katz, and Pister (1999). Next century challenges: Mobile networking for "smart dust". In *ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom 99)*.

Klusch, M., Petta, P., Fensel, D., and Pitt, J. (1999). Premises and challenges of research and development in information agent technology in europe. Technological Roadmap of the Special Interest Group on Intelligent Information Agents as part of the ESPRIT Network of Excellence for Agent-Based Computing.

Kobryn, C. (1999). UML 2001: a standardization odyssey. *Communications of the ACM*, 42(10):29–37.

Malone, T. W. and Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119.

Menezes, R., Tolksdorf, R., and Wood, A. (2000). Scalability in LINDA-like coordination systems. In Omicini, A., Zambonelli, F., Klusch, M., and Tolksdorf, R., editors, *Coordination of Internet Agents: Models, Technologies, and Applications*. Springer.

Mintzberg, H. (1979). *The Structuring of Organizations: A Synthesis of the Research*. Prentice Hall, Englewood Cliffs, N.J.

Ossowski, S. (1999). *Co-ordination in artificial agent societies: social structures and its implications for autonomous problem-solving agents*, volume 1535 of *LNCS*. Springer Verlag.

SiRF Technology, Inc. (1999). Gps goes mainstream. Press release. http://www.sirf.com.

Tolksdorf, R. and Rowstron, A. (2000). Evaluating fault tolerance methods for large-scale linda-like systems. In *Proceedings of the 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2000)*.

Workflow Management Coalition (1998). Interface 1: Process definition interchange process model. http://www.wfmc.org.