

Coordination Artifacts: Environment-based Coordination for Intelligent Agents *

Andrea Omicini, Alessandro Ricci, Mirko Viroli
DEIS, Università degli Studi di Bologna a Cesena
Via Venezia 52
47023, Cesena, Italy
{aomicini,aricci,mviroli}@deis.unibo.it

Cristiano Castelfranchi, Luca Tummolini
ISTC-CNR
Viale Marx 15
00137, Roma, Italy
{castel,tummoli}@ip.rm.cnr.it

Abstract

Direct interaction and explicit communication are not always the best approaches for achieving coherent systemic behaviour in the context of Multi-Agent Systems (MAS). This is evident when taking into account recent approaches dealing with environment-based coordination such as stigmergy and, more generally, mediated interaction. In this paper we propose a conceptual, formal and engineering framework based on the notion of coordination artifact, which aims at generally systematising implicit communication and environment-based coordination for heterogeneous, possibly intelligent agents. The features and benefits of our approach are exemplified in the Follow-me situation, where an agent's action/plan is considered as a model for the action/plan of other agents. We model this class of problems in terms of coordination artifacts, from simple to more challenging cases, stressing the advantages with respect to more "standard" MAS approaches.

1. Introduction

Direct interaction and explicit communication are not always the best approaches to achieve coherent systemic behaviour in the context of MAS and Agent societies. This is evident taking into account the main approaches dealing with environment-based coordination such as stigmergy and, more generally, mediated interaction frameworks and infrastructures based on forms of coordination/cooperation without direct communication [16, 8, 4, 1, 20, 9].

Mediated interaction and environment-based coordination are highly debated also in other research fields outside

MAS and CS, where collaborative and cooperative activities are studied in complex social contexts: notable examples are CSCW and HCI [18], recently focussing on cognitive and social theories which explicitly take into account the role of environment in coordination, such as Distributed Cognition [11] and Activity Theory [13]. In these contexts, a relevant issue is to understand what makes an environment a good place for actors to work together. How (if) the agent environment can be designed to suitably support the social (coordination/cooperation/competition) activities of a dynamic set of heterogeneous agents? This question can be considered of primary importance also in the context of MAS, and in our opinion it involves issues that are not fully considered by current approaches. In particular:

- "Not only ants" – approaches dealing with environment based coordination typically consider *reactive* agents, putting all the intelligence into the environment or as emergent phenomenon;
- "Not only special-purpose coordination" – typically, only solutions to specific coordination problems are provided, without the necessary abstraction to use and systematise coordination in the wide range of social activities;
- "Toward engineering" – frequently, investigations in literature only concern simulation and abstract models, and do not provide methodologies and infrastructures to make them effective in the engineering of systems. A notable example can be found in [9], where a model for situated MAS is provided for the engineering of systems.

In this paper we propose a conceptual and engineering framework based on the notion of *coordination artifact*, which aims at addressing these issues, systematising environment-based coordination for general coordination problems, extending the scope of applicability to heterogeneous, *cognitive/intelligent* agents. Coordination artifacts are runtime abstractions encapsulating and providing a coordination service, to be exploited by agents in

* Research under the auspices of: MIUR (the Italian Ministry of Education, University and Research), COFIN 2003 project "Fiducia e diritto nella società dell'informazione", paper no. 1; MIPAF (the Italian Ministry of Agricultural and Forestry Policies), project "SIPEAA", paper no. 24; and EC (the European Community), FP6 project "AgentLink III".

a certain social context. They can be exploited then as basic building blocks for designing and developing suitable working environments for heterogeneous multi-agent systems, supporting their coordination for collaboration or competition.

In particular, they can be used as suitable tools for modelling and engineering the *Behavioural Implicit Communication* (BIC) approach [3], which allows a wide spectrum of coordination problems for intelligent agents to be modelled without relying on direct communication. In order to exemplify our approach, in this paper we consider the application of coordination artifacts for modelling coordination in the *Follow-me* class of problems – which are a typical case where BIC applies. There, a follower agent delegates a goal to a guide agent, which then brings about that goal by executing a sequence of actions: such actions are meant to be observed and mimicked by the follower. This interesting interaction pattern resembles the idea of a guide leaving footprints in the ground, which are then followed by another agent — the role of footprints is played by the guide agent's trace of actions.

The remainder of the paper is organised as follows: Section 2 describes in detail the notion, the properties, and the formal model of coordination artifacts. Section 3 applies our framework to the conception of a coordination artifact realising the Follow-me situation, outlining basic design choices and providing a complete formal model. Section 4 shows the scalability of our approach as the complexity of the coordination rules increases, by discussing several extensions to the basic Follow-me situation. Finally, Section 5 concludes by providing final remarks.

2. The Coordination Artifact Abstraction

Coordination artifacts can be conceived as embodied¹ entities specialised to provide a coordination service in a MAS. The term coordination should be here understood in its most general acceptance, as the management of dependencies among separate activities [12], shaping and constraining the (agent) interaction space [4]. Coordination artifacts are *infrastructure* abstractions meant to improve coordination activities automation; they can be considered then as basic building blocks for creating effective shared collaborative working environments, alleviating the coordination burden for the involved agents.

Human society is full of entities like coordination artifacts, engineered by humans in order to support and automate coordination activities: well-known examples are street semaphores, blackboards, queuing tools at the supermarkets, maps, synchronisers and so on.

¹ The term embodied is used here to remark their independent existence from the agents using them.

Basically, a coordination artifact (*i*) entails a form of mediation among the agents using it, and (*ii*) embeds and enact effectively some coordination policy. Accordingly, two basic aims can be identified: (*i*) *constructive*, as an abstraction essential for creating/composing social activities, (*ii*) *normative*, as an abstraction essential for ruling social activities.

Taking inspiration also from our society, we can then devise a basic abstract model; a coordination artifact is characterised by:

- a *usage interface*, defined in terms of a set of *operations*. Agents *use* coordination artifacts by two kinds of interaction: executing actions on the artifact, by specifying the artifact operation involved, and by eventually perceiving information about the *completion* (i.e. outcomes) of such actions. Notice that due to the nature of coordination artifacts and their interaction schema, these actions are more similar to *physical acts* rather than *communicative acts*, which make our approach sensibly different from direct, ACL-based interaction;
- a set of *operating instructions*. This information describe (formally) how to use the artifact in order to exploit its coordination service. For instance, operating instructions might specify the protocol of interactions to be used, and the mentalistic semantics of actions and perceptions [22];
- a *coordination behaviour specification*. This information describe (formally) the coordinating behaviour of the artifact, in terms of coordination rules required for enacting the coordination service.

In particular, taking the agent viewpoint, to exploit a coordination artifact simply means to follow its operating instructions, on a step-by-step basis. It's worth noting that, since a considerable coordination burden can be charged upon the artifact and be masked to the agents, operating instructions are generally quite simple compared to the interactive behaviour required in the case of direct communication (protocols). Hence, our approach to interaction can be fruitfully leveraged by intelligent agents, which can exploit an artifact through its operating instructions so as to take part to complex coordination scenarios.

2.1. Main Properties

The basic properties of the agent abstraction have been extensively described in literature, in terms of autonomy, pro-activeness, reactivity, social ability and so on [23]. Analogously, here we focus on the main features that characterise coordination artifacts, which are indeed different.

Specialisation. Coordination artifacts are specialised in automating coordination activities. For this purpose, they

typically adopt a computational model suitable for effective and efficient interaction management, whose semantics can be easily expressed with concurrency frameworks such as process algebras, Petri nets, or Event-Condition-Reaction rules.

Encapsulation: Abstraction and Reuse. Coordination artifacts encapsulate a coordination service, allowing user agents to abstract from how the service is implemented. As such, a coordination artifact is perceived as an individual entity, but actually it can be distributed on several nodes of the MAS infrastructure, depending on its specific model and implementation. Encapsulation is the key to achieve reuse of coordination. Agent society engineers can create and exploit handbooks or catalogs of coordination artifacts, embodying the solutions to general coordination problems in organisations. Finally, a coordination artifact provides a certain *quality of coordination*, in particular in terms of the scalability with respect to the dimensions identified by Durfee in [7], which are related to performance, robustness, reliability, and so on; the description of these qualities is important to identify the range of applicability of the artifact in the engineering of agent societies.

Malleability. Coordination artifacts are meant to support coordination in open agent systems, characterised by unpredictable events and dynamism. For this purpose, their coordination behaviour can be adapted and changed dynamically, either (i) by engineers (humans) willing to sustain the MAS behaviour, or (ii) by agents responsible of managing the coordination artifact, with the goal of flexibly facing possible coordination breakdowns or evolving/improving the coordination service provided.

Inspectability and controllability. A coordination artifact typically supports different level of inspectability: (i) inspectability of its operating instructions and coordination behaviour specification, in order to let user agents to be aware of how to use it or what coordination service it provides; (ii) inspectability of its dynamic state and coordination behaviour, in order to support testing and diagnosing (debugging) stages for the engineers and agents responsible of its management. So, from an operational point of view, a coordination artifact can be understood as a sort of *virtual machine of coordination*, executing some form of coordination specification, fully inspectable and controllable by coordination artifact administrators.

Predictability and formalisability. The coordinating behaviour of an artifact strictly follows the specification/service for which it has been forged: given that specification and the agent interaction history, the dynamic behaviour of the artifact can be fully predicted. So, along with operating instructions, we have a global, operational model of the group of agents using the coordination artifact, enabling forms of verification of coordination activity properties.

Summing up, coordination artifacts are thought to be engineering abstractions used for designing, building and supporting *at runtime* coordination in agent societies. But also, they can be useful to support forms of scientific investigation of collective behaviours. As mediating entities, coordination artifacts typically reify and manage agent communication events; accordingly, they can be used to trace and log the overall interaction behaviour of the agent societies exploiting them. Thus, they can act as kinds of *social memories*, which can then be inspected for possible scientific analysis about global behaviours.

2.2. Artifacts as First Class Citizens of MAS

How to model the coordination artifact abstraction in MAS? Given the characteristics and features described previously, it is evident that the intentional model and – more generally – the agent notion do not fit: properties such as inspectability, controllability, malleability, predictability are foreign (and in some extent contrary) to agents, and viceversa, autonomy, proactiveness, and rationality are foreign to coordination artifacts. Moreover, the inter-agent and agent-artifacts interaction models are profoundly different: typically agents have not interfaces, and they are not used by other agents through operations. Coordination artifacts have much more affinity with environment resources, which agents use, instead of communicating with according to an high level ACL. So, the agent abstraction is probably not the most proper one to understand, model, and engineer a coordination artifact behaviour.

For this reasons, it can be reasonable to introduce in a MAS model/infrastructure coordination artifacts as first class entities. An example of this approach can be found in the TuCSon coordination infrastructure [14], where *tuple centres* runtime coordination abstractions are used to play the role of coordination artifacts [17].

2.3. Formal Model

In this section we provide a formal model of the behaviour of coordination artifacts, taking into account the concepts of actions and perceptions, the usage instructions associated to each agent, and the coordination policy that the artifact realises. Since we argued that coordination artifacts are not suitably modelled and engineered as cognitive entities like agents, we do not describe their behaviour in terms of modal logics of mental properties. Rather, as far as interaction and coordination are concerned, we find useful to leverage the formal framework of concurrency theory and process algebraic approaches, whose application to the MAS field is not completely new but it is still under development [10, 21]. The main strength of our approach is that it easily leads to an operational model of coordination arti-

facts, which — by definition — can be directly exploited to devise a correct implementation. This contrasts modal logics approaches, which are more suitable to describe properties of a system, but are more hardly implementable [2].

A coordination artifact is a tuple $\langle \alpha, \beta, \rho, \delta, \longrightarrow_\sigma \rangle$. α is a meta-variable ranging over the operations allowed by the coordination artifact, namely, the actions the agent can execute on it. β is the meta-variable ranging over perceptions of action completion, which may possibly contain some information about the outcome of the action. Correspondingly, the set L of interactions between agents and the coordination artifact, ranged over by l , is defined by syntax

$$l ::= id!\alpha \mid id?\alpha\beta$$

where $id!\alpha$ represents agent id executing action α , and $id?\alpha\beta$ represents agent id perceiving the completion β to action α .

ρ is a function associating to each agent identifier id the usage instruction I he has to follow, here expressed as the protocol of admissible actions and perceptions for that agent. Following the approach described in [22], instructions can be defined by exploiting typical process algebraic operators, e.g. by the syntax:

$$I ::= 0 \mid !\alpha \mid ?\beta \mid I + I \mid I; I \mid I \parallel I \mid \mathcal{D}$$

Here, 0 is the void instruction, $!\alpha$ is execution of an action, $?\beta$ is perception of a completion, operator “+” is used for choice between instructions, “;” for sequential composition of instructions, “ \parallel ” for parallel (concurrent) composition of instructions, and \mathcal{D} is invocation of a recursive definition. As an example, the definition $I_X := !a; ((?b; I_X) + ?c)$ means that the agent is initially allowed to execute action a , and later perceive either completion b or c : while c involves termination of the instructions, b causes the whole instructions to be allowed again, through the recursive call to I_X . This is a typical schema for an agent asking information through the artifact until the protocol is shut-down.

An operational semantics can be defined for this language, based on a transition relation \longrightarrow_I , where notation $I \xrightarrow{! \alpha}_I I'$ means that instructions (state) I moves to I' by the execution of action α , and $I \xrightarrow{? \alpha \beta}_I I'$ that I moves to I' as action α completes with perception β . The details of that semantics are not particularly relevant, for the meaning of the above algebraic operators is quite standard and plays no significant role in the following. Therefore, operational rules are avoided for the sake of brevity: the interested reader can refer to [22] for their presentation.

Meta-variable δ ranges over the data reified into the coordination artifact to keep track of the state of the coordination task. Correspondingly, we let meta-variable σ range over the set Σ of states of the coordination artifact, which is defined as:

$$\sigma ::= 0 \mid \delta \mid l \mid (\sigma \parallel \sigma)$$

Operator \parallel is characterised by the following congruence rules:

$$\sigma \parallel 0 \equiv \sigma \quad \sigma \parallel \sigma' \equiv \sigma' \parallel \sigma \quad \sigma \parallel (\sigma' \parallel \sigma'') \equiv (\sigma \parallel \sigma') \parallel \sigma''$$

Thus, elements σ are easily understood as parallel compositions of elements δ and interactions l — the latter used to represent pending actions waiting to be executed and pending completions waiting to be perceived.

This state changes as interactions occur: this dynamics is modelled by the transition relation $\longrightarrow_\sigma \subseteq \Sigma \times \Sigma$, representing the fact that a state σ may eventually move to another σ' , which typically happens when a new pending action has to be computed.

So, while α and β define the shape of interactions allowed by the coordination artifact, ρ defines the protocols allowed to the agents, while δ and \longrightarrow_σ define the actual coordination task. Given the tuple $\langle \alpha, \beta, \rho, \delta, \longrightarrow_\sigma \rangle$, the (interactive) behaviour of a coordination artifact is described by a transition system $\langle C, \longrightarrow, L \cup \{\tau\} \rangle$. C is the set of configurations of the coordination artifact, which are of the kind $\rho \otimes \sigma$ namely, the composition of a function ρ associating to each agent the instructions it currently has to follow, and the current state of the artifact σ . The transition relation $\longrightarrow_\sigma \subseteq C \times (L \cup \{\tau\}) \times C$ is then defined by the rules:

$$\begin{array}{c} \frac{\rho(id) \xrightarrow{! \alpha}_I I}{\rho \otimes \sigma \xrightarrow{id! \alpha} \rho[id \mapsto I] \otimes \sigma \parallel id! \alpha} \quad \text{[ACT]} \\[10pt] \frac{\rho(id) \xrightarrow{? \alpha \beta}_I I}{\rho \otimes \sigma \parallel id? \alpha \beta \xrightarrow{id? \alpha \beta} \rho[id \mapsto I] \otimes \sigma} \quad \text{[COMP]} \\[10pt] \frac{\sigma \rightarrow_\sigma \sigma'}{\rho \otimes \sigma \xrightarrow{\tau} \rho \otimes \sigma'} \quad \text{[COORD]} \end{array}$$

The first rule handles a new action α executed by agent id to the coordination artifact. This is allowed only if the associated instructions $\rho(id)$ admit the transition towards some instructions I , in which case such instructions become the new instructions associated to id — ρ moves to $\rho[id \mapsto I]$ — and interaction $id! \alpha$ is reified in the state σ .

In a similar way, the second rule deals with a general completion β to action α : when this is reified in the state σ and the instructions admit its perception the completion is actually executed, and the ρ function is updated.

Finally the third rule deals with the actual coordination task realised inside the artifact: simply, when transition relation \longrightarrow_σ enables a transition for the state σ this can be applied to the current configuration and becomes a silent transition for the whole coordination artifact.

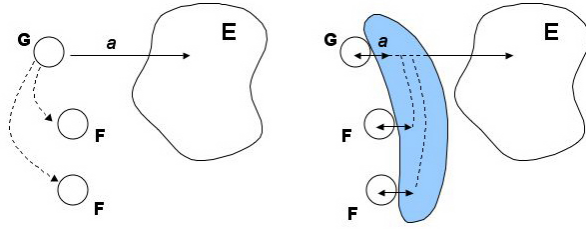


Figure 1. The Follow-me scenario using direct communication (left) and a coordination artifact (right). G is the guide, F are the followers and E is the environment

3. The Follow-Me Case Study

As an example to clarify the framework of coordination artifacts, we consider its application to model and engineer the conceptual approach provided by Behavioural Implicit Communication [3]. The BIC framework proposes a general conceptual scenario where, to achieve coordination, autonomous agents should be able to interact also through their *practical actions* or their *traces* (as in stigmergy), without necessarily adopting direct communication (i.e. speech acts as in FIPA ACL).

The prototype of this kind of coordination problem is exemplified by the Follow-me situation, where an agent's action/plan is considered a model for the action/plan of other agents. Two agents can coordinate their actions agreeing on an initial norm and then communicating only implicitly. Suppose a natural situation such that: an agent (called here follower, F) wants to know how to get to a place and ask for this information to another one (the guide, G). Then, the guide agent G says "Follow me!", by a speech act of the kind *order*. From there on, each practical action of the guide is to be observed by F , and can be understood as an implicit message with a sort of deontic content ('turn left', 'turn right', etc.).

This scenario can be naturally modelled/engineered by instrumenting a suitable coordination artifact, which mediates G actions on the environment. Such traces are reified in the coordination artifact as part of its dynamic state, and can be observed in an uncoupled way by F through its interaction with the coordination artifact. Figure 1 shows from a conceptual point of view the shift from an approach based on direct communication to an implicit, artifact-based approach. Basically, the operating instructions for the artifact would account for G operations for doing the actions, and for F operations for observing them.

The design and complexity of the coordination artifact depend on the coordination which we aim to capture and

provide as-a-service, removing the involved burden from the individual agents. In this first example, the interaction uncoupling naturally supported by the artifact is yet sufficient to have some significant benefits. For instance, it allows agents to focus on their on task, without worrying about carrying on a successful coordination activity: G is not asked to communicate action execution to any one, he is only focused on acting until reaching the desired goal; F is not worried about loosing any G 's actions or to observe them in a wrong order. Also, agents can have different timings and speed in executing their individual tasks; for instance, imagine G as explorers, very quickly moving inside an hypothetical labyrinth, and F as analysers, slower in analysing the places discovered by G .

3.1. The Follow-me Coordination Artifact

According to the above discussion, we here describe the Follow-me coordination artifact, which the guide agent acts upon, and which let a follower agent perceive actions in the right order through successive request actions.

Let meta-variable a range over practical actions subjected to the the Follow-me coordination pattern, and among them, e a meta-variable over the final actions — after which the coordination task can be considered successfully concluded — and o a meta-variable over ongoing, non final actions. Syntax, including actions and perceptions allowed by this coordination artifact, is defined as:

$$a ::= o \mid e, \quad \alpha ::= do(a) \mid get, \quad \beta ::= act(a) \mid ok$$

The guide executes action a by operation $do(a)$ over the artifact, eventually perceiving the completion ok^2 ; the follower asks for the next practical action by operation get , perceiving the completion $act(a)$ which means that this is a . These two protocols of interaction are actually stored within the role function ρ , respectively associating to the guide agent id_G and the follower agent id_F the two operating instructions:

$$\begin{aligned} I_G &::= (!do(o); ?ok; I_G) + !do(e) \\ I_F &::= !get; ((?act(o); I_F) + ?act(e)) \end{aligned}$$

The meta-variable δ over data to be reified is defined by the syntax

$$\delta ::= 0 \mid a \mid \delta.\delta$$

with the congruence rules

$$0.\delta \equiv \delta \quad \delta.0 \equiv \delta \quad \delta.(\delta'.\delta'') \equiv (\delta.\delta').\delta''$$

2 Such a trivial completion is supposed for the sake of simplicity, for it plays no significant role in the Follow-me situation

Namely, elements δ represent trace of actions, which are sequences $a_1.a_2.a_3.\dots.a_n$. In particular, in this basic Follow-me δ will contain the trace of practical actions of the guide which have not yet been perceived by the follower. Finally, the transition relation \longrightarrow_σ is defined by the two simple rules:

$$\begin{aligned} \delta \parallel id!do(a) &\longrightarrow_\sigma a.\delta \parallel id?ok & [\text{DO}] \\ \delta.a \parallel id!get &\longrightarrow_\sigma \delta \parallel id!act(a) & [\text{GET}] \end{aligned}$$

The [DO] rule says that as an operation $do(a)$ is executed, a is queued in the trace within the artifact, and the completion $id?ok$ is reified waiting to be served — by rule [COMP] shown in previous section. Dually, rule [GET] handles a pending operation $id!get$ of a follower asking for the next action: this is served only if the trace has at least one action a , in which case this is consumed and the completion $id!act(a)$ is reified.

This example and the others presented in the remainder of this article show that a coordination artifact is specified in terms of structural properties of the collaboration and in terms of coordination rules. The former include aspects related to the interaction with agents, that is, the usage interface and the operating instructions — structures α , β and ρ . The latter only concerns the way these interactions are coordinated, specified by the reified data δ and the internal transition relation \longrightarrow_σ .

It's worth noting that this formal model can be implemented concretely using TuCSon infrastructure. In particular, the Follow-me rules can be translated in reactions expressed in the ReSpecT language, defining the behaviour of a tuple centre used as Follow-me coordination artifact.

4. Scaling with Coordination Complexity

It is easy to conceive more complex versions of the basic Follow-me situation, moving to those more complex scenarios that typically characterise working environments in MAS. Accordingly, we can design and engineer suitable coordination artifacts, with specialised coordination behaviour to handle the new dependencies and provide some extended coordination behaviour. Moreover, we can think to extend/adapt existing ones, even dynamically, following an online engineering approach.

The important point here is that changes in pure coordination activities are localised in changes on the coordination artifacts, without the need to intervene globally on the systems, that is also on agents participating to the coordination: in other words, the agents viewpoint remains mostly unchanged. Of course, if necessary, agents can be made aware of the new functionalities or features — if they are not themselves among the agents/engineers adapting the coordination artifacts — and then invited to inspect and learn possible new operating instructions.

4.1. Extended Applications of the Follow-me Pattern

The coordination artifact for the Follow-me interaction pattern presented in previous section already features a number of interesting properties that direct interaction would more hardly cope with. First, it features *time uncoupling*, that is, the possibility to handle the case where guides and followers are not synchronised, but either the guide produces actions more rapidly than the follower can perceive, and viceversa. This property is guaranteed by the fact that traces are made persistent in the coordination artifact, and are thus independent of the actual dynamics. Second, our approach also features *space uncoupling* — also called *identity uncoupling* —, namely the guide is not aware of the identity of the follower: it could even be unaware of its actual presence! In this section we show that other than these basic properties, the coordination rules of the artifact can be adapted so as to include other desirable behaviours, extending the applicability of the coordination artifact. Indeed, it is crucial to notice that such extensions are realised without any modification to the basic architecture: suffices it to change the internal structures δ and \longrightarrow_σ , the operating instructions for the agents remaining unchanged.

As a first example we consider the case where the artifact is not used to coordinate one follower and one guide, but in general to deal with a multiplicity of agents willing to follow the same guide. Thanks to space uncoupling, this is a smooth extension to the basic model, obtained by (i) reifying the association of a follower's identifier to a guide's identifier, (ii) keeping the guide's trace persistent instead of consuming it as it is perceived, and (iii) keeping track of the traces of actions perceived by each follower. Sticking to the scenario that suggested the Follow-me situation, it would mean to support the idea that traces are not delete as followers see them, so that other followers on the same path can still perceive them.

As a second extension we consider the scenario where among all the practical actions perceived by followers, ranged over by meta-variable a , there are actions — ranged over by f — which are considered failures: if the guide cannot reach the desired goal for any reason, it then executes the failure action f . Correspondingly to this extension, it is reasonable to let followers intercept failed traces as soon as failure occurs, instead of waiting until reaching the failure themselves. This is simply achieved by turning a guide's trace into an empty, failed trace as an action f is executed by the guide. Conceptually, this is as if a trace would disappear as soon as something bad occurs to the guide that was leaving it.

A final situation we here analyse concerns the idea of *consolidated traces*. We may suppose that a coordination

artifact keeps track of the traces that successfully completed, and from then on, other than the usual Follow-me behaviour, implements also a notion of *map*: instead of following a guide, an agent might also be allowed to perceive a previously left trace – namely a consolidated trace. This is simply obtained in the coordination artifact by allowing to bind a follower to a trace based on information about the initial state and final expected state, instead of just binding a follower to a guide. Conceptually, this is as if a trace on the ground becomes persistent, i.e. it becomes a true path, as soon as the guide successfully reaches the intended destination.

4.2. A Formal Model

We model here a coordination artifact supporting the Follow-me pattern with the extensions described above. We coordinate more followers and more guides, with every single follower being bound to either one guide or a consolidated trace: more followers are also allowed to follow the same guide or the same consolidated trace. We also support the notion of failure actions, and the corresponding early detection to followers.

We extend the syntax of the basic Follow-me as follows:

$a ::= f \mid c$	failures and correct actions
$c ::= e \mid o$	termination and ongoing actions
$t ::= 0 \mid a \mid t.t$	trace
$\delta ::= id_F/id_G \mid$	follower-guide association
$id_F/\theta \mid$	follower-trace association
$id_F : t \mid$	Follower's trace
$\theta id_G : t$	Guide's trace

Meta-variable f ranges over actions representing failures, e over actions representing successful termination, and t over traces of actions, namely, sequences of elements a of the kind $a_1.a_2.\dots a_n$ – subjected to the same congruence rules described in previous section. Reified data can be of the kind (i) id_F/id_G , meaning that follower id_F is bound to guide id_G , (ii) id_F/θ , meaning that follower id_F is bound to follow a consolidated trace from position θ' to position θ'' (representing initial state and goal to be achieved), (iii) $id_F : t$, representing the fact that follower id_F has already perceived the trace t , and (iv) $\theta id_G : t$ meaning that guide id_G has left trace t while moving from θ' to θ'' .

The coordination rules corresponding to the new coordination task are shown in Figure 2. Rule [E-DONE] says that any new correct — i.e., non-failure — action c by a guide id_G is simply attached to its trace, which moves from t to $c.t$, and that the completion $id?ok$ is generated. Viceversa, rule [E-FAIL] handles the case of a failing action, by which the guide's trace moves from its state t to the new trace made by the only failure action f , signifying that the

whole trace is failed. Rule [E-BLOCK] deals with the case where the follower perceives the failure of the trace: the association id_F/id_G is dropped from the artifact state, and the follower is allowed to perceive the guide's execution of action f . The situation where a follower perceives an ongoing action o is instead modelled by rule [E-ACT]: given that the follower has already perceived trace t and the guide has left trace $t'.o.t$, then o is next action, which is then added to id_F 's perceived trace. A similar situation occurs when the guide performs a final action e (rule [E-LAST]), which differently from previous rule causes all data concerning the follower (trace and association) to be dropped from the artifact state. Finally, the two rules [E-CONS] and [E-CLAST] correspond to rules [E-ACT] and [E-LAST], but deal with the case where the follower is bound to a consolidated trace instead of a guide.

Again, it's worth noting that our engineering methodology supports the adaptation of a coordination task to more and more complex situations without involving significant redesign, but just changing the operational specification of the coordination artifact behaviour.

5. Conclusions and Related Work

In this paper we introduced the notion of coordination artifacts, discussing its effectiveness for modelling and engineering social activities inside open/heterogeneous agent societies, and, in particular, for systematising implicit communication and environment-based coordination approaches.

Coordination artifacts and their supporting infrastructure can be thought as a conceptualisation and generalisation of specific coordination mechanisms found in agent literature (an example is Singh's coordination service [19]).

More generally, the coordination artifact abstraction brings in MAS ideas and concepts that have played a central role in other (un)related fields. From concurrent and distributed systems, they can be considered the generalisation of traditional coordination abstractions, from low level ones such as semaphores, monitors, to high level ones, such as tuple spaces and, more generally, coordination media as found in coordination models and languages [15]. Blackboards as defined in Distributed Artificial Intelligence context [5] can be framed and modelled in MAS as coordination artifacts, toward the integration of the two different points of view (traditional multi-agent and blackboard systems) in designing collaborating-software engineering space [6]. Actually, the main inspiration for the coordination artifact concept comes from fields involved with the analysis and research on human activities, in particular social & collaborative ones; in particular, coordination artifacts embody in the context of MAS the notion of coordinative arti-

$id_G!done(c) \parallel \theta'' id_G : t$	$\longrightarrow_\sigma id_G?ok \parallel \theta'' id_G : c.t$	[E-DONE]
$id_G!done(f) \parallel \theta'' id_G : t$	$\longrightarrow_\sigma id_G?ok \parallel \theta'' id_G : f$	[E-FAIL]
$id_F!get \parallel id_F/id_G \parallel \theta'' id_G : f$	$\longrightarrow_\sigma id_F?act(f) \parallel \theta'' id_G : f$	[E-BLOCK]
$id_F!get \parallel id_F/id_G \parallel id_F : t \parallel \theta'' id_G : t'.o.t$	$\longrightarrow_\sigma id_F?act(o) \parallel id_F/id_G \parallel id_F : o.t \parallel \theta'' id_G : t'.o.t$	[E-ACT]
$id_F!get \parallel id_F/id_G \parallel id_F : t \parallel \theta'' id_G : e.t$	$\longrightarrow_\sigma id_F?act(e) \parallel \theta'' id_G : e.t$	[E-LAST]
$id_F!get \parallel id_F/\theta'' \parallel id_F : t \parallel \theta'' id_G : e.t'.c.t$	$\longrightarrow_\sigma id_F?act(c) \parallel id_F/\theta'' \parallel id_F : c.t \parallel \theta'' id_G : t'.c.t$	[E-CONS]
$id_F!get \parallel id_F/\theta'' \parallel id_F : t \parallel \theta'' id_G : e.t$	$\longrightarrow_\sigma id_F?act(e) \parallel id_F : c.t \parallel \theta'' id_G : e.t$	[E-CLAST]

Figure 2. Coordination Rules for the Extended Follow-me Coordination Artifact

fact as found in CSCW [18] and of mediating tool as found in Activity Theory [13].

References

- [1] E. Bonabeau, M. Dorigo, , and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
- [2] M. Broy and E.-R. Olderog. Trace-oriented models of concurrency. In *Handbook of Process Algebra*, chapter 2, pages 101–195. North-Holland, 2001.
- [3] C. Castelfranchi. Silent agents: From observation to tacit communication. Submitted to AAMAS'04 Workshop "Agent Tracking: Modelling Other Agents from Observations", Available at: <http://www.istc.cnr.it/createhtml.php?nbr=62>, 2004.
- [4] P. Ciancarini, A. Omicini, and F. Zambonelli. Multiagent system engineering: The coordination viewpoint. In N. R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, volume 1757 of *LNAI*, pages 250–259. Springer-Verlag, 2000.
- [5] D. D. Corkill. Blackboard systems. *Journal of AI Expert*, 9(6):40–47, 1991.
- [6] D. D. Corkill. Collaborating software: Blackboard and multi-agent systems & the future. In *Proceedings of the International Lisp Conference*, 2003.
- [7] E. H. Durfee. Scaling up agent coordination strategies. *IEEE Computer*, 34(7), July 2001.
- [8] M. Fenster, S. Kraus, and J. S. Rosenschein. Coordination without communication: Experimental validation of focal point techniques. In *Proceedings of International Conference on Multi-Agent Systems (ICMAS'95)*, pages 102–108. AAAI, 1995.
- [9] J. Ferber and J.-P. Müller. Influences and reaction: a model of situated multiagent systems. In *Proceedings of International Conference on Multi-Agent Systems (ICMAS'96)*, 1996.
- [10] D. Kinny, editor. *The Psi Calculus: An Algebraic Agent Language*, volume 2333 of *Lecture Notes in Computer Science*. Springer, 2002.
- [11] D. Kirsh. Distributed cognition, coordination and environment design. In *Proceedings of the European conference on Cognitive Science*, pages 1–11, 1999.
- [12] T. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.
- [13] B. A. Nardi. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, 1996.
- [14] A. Omicini and F. Zambonelli. Coordination for Internet application development. *Autonomous Agents and Multi-Agent Systems*, 2(3):251–269, Sept. 1999.
- [15] G. A. Papadopoulos. Models and technologies for the coordination of Internet agents: A survey. In A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, editors, *Coordination of Internet Agents: Models, Technologies, and Applications*, chapter 2, pages 25–56. Springer-Verlag, Mar. 2001.
- [16] H. V. D. Parunak, S. Brueckner, M. Fleischer, and J. Odell. A preliminary taxonomy of multi-agent interactions. In *Proceedings of the 2nd International Joint conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, pages 1090–1091. ACM Press, 2003.
- [17] A. Ricci, A. Omicini, and E. Denti. Activity Theory as a framework for MAS coordination. In P. Petta, R. Tolksdorf, and F. Zambonelli, editors, *Engineering Societies in the Agents World III*, volume 2577 of *LNCS*, pages 96–110. Springer-Verlag, Apr. 2003.
- [18] K. Schmidt and C. Simone. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *International Journal of Computer Supported Cooperative Work (CSCW)*, 5(2–3):155–200, 1996.
- [19] M. P. Singh. A customizable coordination service for autonomous agents. In *Agent Theories, Architectures, and Languages*, pages 93–106, 1997.
- [20] L. Steels. The artificial life roots of artificial intelligence. *Artificial Life Journal*, 1(1):89–125, 1994.
- [21] R. M. van Eijk, F. S. de Boer, W. van der Hoek, and J.-J. C. Meyer. A verification framework for agent communication. *Autonomous Agents and Multi-Agent Systems*, 2(6):185–219, 2003.
- [22] M. Viroli and A. Ricci. Instructions-based semantics for agent mediated interaction. In this volume, 2004.
- [23] M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.