Applied Artificial Intelligence, 18:815–831, 2004

ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839510490509036

MULTI-AGENT INFRASTRUCTURES FOR OBJECTIVE AND SUBJECTIVE COORDINATION

ANDREA OMICINI, ALESSANDRO RICCI, MIRKO VIROLI, and MARCO CIOFFI
DEIS, Università di Bologna, Cesena, Italy

GIOVANNI RIMASSA
Research and Consulting Department,
Whitestein Technologies AG, Zurich, Switzerland

Coordination in multi-agented systems (MAS) can be conceived as either an agent activity (the subjective viewpoint) or an activity over agents (the objective viewpoint). The two viewpoints have generated two diverging and often contrasting lines of research, as well as different and noncompatible technologies, however, their integration is mandatory for modeling and engineering complex MAS. In this paper, we explore the issue of integration at both the model and the technology levels.

First, by taking FIPA agents and coordination artifacts as reference notions for subjective and objective approaches, respectively, we sketch a framework where agent interactions with coordination artifacts are modeled as physical acts, deliberated and executed by agents analogously to communicative actions. Then, we show how the JADE infrastructure for FIPA-compliant agents, and the TuCSoN *infrastructure providing agents with coordination artifacts can be integrated at the technology level, allowing JADE agents to access* TuCSoN *tuple centers through JADE services.*

SUBJECTIVE AND OBJECTIVE COORDINATION

The precise definition of the notion of coordination is still an open problem in computational sciences. In the distributed artificial intelligence field, coordination (Jennings 1993) was mainly interpreted as an individual, psychological activity performed by a component, typically, an agent trying to achieve its own subjective goals within a multi-component system, typically, a multi-agent system (MAS henceforth). Instead, in the programming languages and software engineering areas, coordination (Ciancarini 1998)

Address correspondence to Andrea Omicini, DEIS, Università di Bologna, Via Venezia 52, 47023 Cesena, Italy. E-mail: aomicini@deis.unibo.it

was basically regarded as a normative activity performed by some part of a multi-component system on behalf of the system's designer, typically, by a coordination medium provided by an infrastructure.

Such different viewpoints (that we henceforth denote as DAI and SE, respectively) have led to the development of separate and often contrasting lines of research on coordination, with different views over components: the former (DAI), where components are the *coordinating* entities and the latter (SE), where components are the *coordinated* entities. On the one hand, the DAI approach typically studies systems whose components exhibit a high degree of autonomy (intelligent agents being the most obvious example), and tends to focus on intra-agent issues, so that inter-agent issues (like infrastructural ones, which are mandatory for applicability to real-world scenarios) are still far from a satisfactory solution (Cost et al. 2001; FDS 2000). On the other hand, the SE approach (see, for instance, Arbab et al. [1993]) fits well with application scenarios involving a finer component granularity (as typical in the case of mobile agents), and often disregards any capability of the components in terms of autonomy or deliberation, not to speak of component intelligence.

Seemingly, the two approaches provide two complementary views over coordination. Strangely enough, this apparently obvious statement took its time to be shared by the different communities working on coordination. The first successful attempt to put the two things altogether was made by Schumacher (2001), where the notions of *subjective* and *objective* coordination were first introduced, and then used to classify the research on coordination. In the context of MAS, subjective and objective coordination were defined as coordination *inside* and *outside* the agents, respectively, thus accounting for the psychological vs. normative acceptations of coordination recalled above.

A step beyond was then the recognition that any nontrivial multi-component system cannot but rely on the use of the two approaches altogether. Along this line, Ricci et al. (2003) proposed Activity Theory (AT henceforth) as a unitary and coherent conceptual framework for both coordination approaches, whereas Omicini and Ossowski (2003) advocated that both play a fundamental role in the engineering of MAS, and that any methodology for the design and development of MAS should necessarily exploit both objective and subjective coordination models and technologies. It does not come as a surprise that the frameworks that better reconcile the two lines are organizational ones like AT (Leontjev 1978). In fact, a main concern for organizations is typically how to make individual (psychological) and social (normative) aspects fruitfully coexist. A social norm can be either imposed or accepted, and also interiorized by agents of the organization that can then perform their activity (either intelligent or not), according to their nature and goals.

However, to reconcile the two approaches is not enough. The divergence in the research lines has led to a technology/infrastructure legacy that should be now somehow re-composed. For instance, the TuCSoN coordination infrastructure (Omicini and Zambonelli 1999) and the JADE FIPA-compliant framework (JADE Board 2000) basically deriving from the SE and the DAI lines, respectively, are in some sense effective and powerful solutions to complementary classes of problems. However, it is apparently not easy to make them coexist and work together in a coherent and effective way. Accordingly, any attempt to reconcile objective and subjective approaches to MAS coordination should not only aim at providing a uniform conceptual framework, but also at suitably integrating technologies and infrastructures. While some steps in the right direction have already been taken, as in the cases of the notions of *agent coordination context* in TuCSoN (Omicini and Ricci 2003) or conversations in FIPA (Cost et al. 2001), most work still has to be done. As a result, in this paper, we aim to address the issue of integration of the two approaches at both the model and the technology levels.

At the model level, we devise a common framework for subjective and objective coordination models based on the idea of FIPA agents interacting with *coordination artifacts* through *physical acts*. To this end, the next section overviews the most common approaches to MAS coordination, typically adopted in the FIPA context, and rooted in the DAI/subjective research line. After that we discuss the notion of coordination artifact (Ricci et al. 2003) rooted in the SE/objective research line, and show how this works as a foundation for the integration of objective and subjective coordination. Then we introduce physical acts to model agent interaction with coordination artifacts in the FIPA context, so that coordination of FIPA agents is no longer limited to mere communicative actions.

At the technological level, we take TuCSoN as the reference infrastructure for coordination artifacts, namely, tuple centers (Omicini and Denti 2001) and JADE (JADE Board 2000), as a reference platform for FIPA-compliant agents. An integrated JADE/TuCSoN architecture is then presented and a simple example is discussed, where JADE agents coordinate through a TuCSoN tuple center, instrumented as a JADE service. Finally, we present conclusions and future works.

COMMUNICATION AND COORDINATION IN STANDARD MAS

The basic concept of agency, along with the core attributes that are generally attached to a single software agent, has its roots in artificial intelligence research. When modeling a software system as a software agent, one adopts the *intentional stance* that is taken by ascribing mentalistic qualities such as beliefs, goals, and desires to the system under study. In works such as Dennett (1987), this approach is claimed to be better suited to complex software systems because it raises the abstraction level. Moreover, in the same work it is argued that ascribing mentalistic properties to external systems is the natural strategy of humans as soon as the system complexity exceeds a certain threshold. From a software architecture point of view, modeling a system intentionally results in a coarse grained component model, with a small number of complex software agents gifted with reasoning and planning capabilities (the extreme case being the single, powerful intelligent agent that encompasses the whole system).

The subsequent research on MASs shifted the focus away from the single software agent to consider a system composed by many agents. With the proposal of the *social level* of abstraction (Jennings and Campos 1997), the suggestion was made to leverage theories and results from social and organizational sciences in structuring software systems. To effectively incorporate social abstractions within a system model made by intentional software components, proper connectors are necessary. An *agent communication language* (ACL in the following) is one such connector that proved itself quite well suited to MASs. Modern ACLs are component communication languages arising from a logic formalization of *speech act theory*, a linguistic theory considering language as a way used by humans and their societies to affect the environment in which they live (Austin 1962). The basic intuition driving the ACL approach brings forth the intentional stance idea: Just like the mentalistic agent models stem from a human designer taking an intentional stance toward a single software agent, ACL-based component communication considers two software agents (the speaker and the hearer of the speech act) applying the intentional stance to each other. As soon as the two become engaged in communication, each one starts modeling its peer intentionally, ascribing to it mentalistic features derived from the observed utterances. These derived features, in turn, drive the agent subsequent communication (as soon as it realizes something about its peer, it can act accordingly). Thus, due to the linguistic ACL connector, the intentional models of all the agents, each taken in isolation, come together to form the MAS model. So, the intentional model of a single, isolated agent can be considered its component specification, whereas the MAS model can be thought of as the overall system specification. With the intentional agents and ACL approach to MASs, the resulting system model is intentional in formulation but social in scope.

FIPA (Foundation for Intelligent Physical Agents) represents the standard approach in the context of MAS. It is an international nonprofit association of companies and organizations sharing the effort to produce specifications for generic agent technologies. Within the arena of distributed software infrastructures, FIPA promotes a landscape where *agent platforms* provide life support to communities of agents, which in turn cooperate to enable services and applications. FIPA tries to support both agent-level and platform-level interoperability through a comprehensive set of specifications.

The main assets described in FIPA agent-level specifications comprise all the items needed to support MASs (and their interoperability) using intentional specification, ACL, and subjective coordination.

In particular, subjective coordination is achieved by allowing an agent to decide to participate in those conversations that it believes useful to effectively bring about its goals. It is subjective because coordination among the participants of an interaction arises as a result from their various individual perspectives and from their mutual influencing. Even in purely subjective MASs, coordination is not just an emergent system property, but it is often explicitly planned and negotiated by the agents. As an example, when dealing with coalition formation problems, interacting agents can explicitly establish a common goal starting from their own individual goals, and they can use their linguistic ACL connector together with their mutual intentional modeling to make sure that both explicitly agree on the definition of the common goal.

OBJECTIVE COORDINATION IN STANDARD MAS

The problems of purely subjective approaches have been extensively discussed in the literature (Omicini and Ossowski 2003). Among the others, here we mention the impossibility of suitably modeling the environment and the agent-environment interaction (Odell et al. 2003); the difficulty to support complex coordination activities by relying exclusively on subjective capabilities of the individuals (Omicini and Ossowski 2003); the unfeasibility of engaging complex coordination only based on direct semantics-driven interaction (Cost et al. 2001); and the unworkability of prescriptive coordination, enforcing social norms and rules (Bergenti and Ricci 2002). It is worth remarking that the limits of the approaches using direct communication among individuals as the sole means for coordination also emerged in other contexts such as CSCW (Andersen et al. 2000; Schmidt and Simone 1996). The need to overcome such limitations is also recognizable in some recent approaches in the context of standard MAS: Notable examples are conversations/interaction protocols (Cost et al. 2001) and institutions (Noriega and Sierra 2002).

A conceptual framework to bridge the gap between subjective and objective coordination was presented by Ricci et al. (2003). The framework is based on the studies on Activity Theory (AT) in the context of human collaborative/social activities (Nardi 1996) and its application to CSCW (Bardram 1998). Central to this framework is the notion of *coordination artifact*, which extends the AT notion of artifact for social context. In fact, research on AT and CSCW points out that any complex social activity is always *mediated* by *artifacts*, either physical or psychological, such as forms, blackboards, but also operating procedures, heuristics, scripts, individual/collective experiences, and languages.

In turn, a coordination artifact is a persistent entity (an embodied artifact) designed to provide a coordination service for the collectivity (society) of agents sharing it as a support for their cooperative/social work (Omicini et al. 2004). Basically, a coordination artifact works as a mediator for agent interactions, and also embodies and enacts coordination policies. Accordingly, two basic aims for coordination artifacts can be devised: (i) *constructive*, as an abstraction meant to enable/compose social activities and (ii) *normative*, as an abstraction meant to constrain/rule social activities.

A coordination artifact is not an agent: it needs not be autonomous or proactive, nor is it required to exhibit cognitive abilities. Instead, according to Ricci et al. (2003), a coordination artifact should first of all be specifically designed to mediate and govern agent interaction, and then feature a *predictable* behavior. Also, in order to let coordination designers (humans or agents) diagnose/monitor/debug the state and dynamics of MAS interaction, a coordination artifact should be dynamically *inspectable*. Finally, in order to reflect updates in coordination strategies and to adapt to changes of the social environment, coordination artifacts should be *malleable*, so that they can be created on-the-fly, and their behavior dynamically adapted or modified as needed.

Following the AT-based framework defined by Ricci et al. (2003), every collaborative activity in MAS deals with coordination artifacts, and can be modeled according to three hierarchical levels: *co-construction*, *co-operation*, and *co-ordination*. In short, at the co-construction level, agents establish the shared objective characterizing the social task. At the co-operation level, agents define cooperatively (typically, by means of negotiation) the structure and behavior of the coordination artifacts to be shared for the purpose of the achievement of the social task. At the co-ordination level, agents use the coordination artifacts selected or forged for the social task in the most automated and fluid manner.

In the analysis of collaborative activities, AT emphasizes that no activity can be said to exist at one level alone: co-ordination, co-operation, and co-construction are *analytical* distinctions of the same collaborative activity, and concur in different times and modes to its development. Consequently, the notion of dynamic transformation between the three hierarchical levels turns out to be crucial. Transformation from co-ordination to co-operation/co-construction occurs when the flow of work relying on coordination artifacts needs to be cooperatively re-established, and the behavior of the artifact inspected for possible changes; the reasons could be either coordination breakdown, or a deliberate re-conceptualization of the way the work is currently done. Transformation from co-operation to co-ordination works the other way around: Once re-established, the artifact behavior is changed accordingly and provided again to participants in order to be exploited for the co-ordination stage.

Along this line, it is now easy to recognize that objective and subjective approaches are to be exploited together in the same coordination context, but at different conceptual and operational levels: Subjective approaches are prominent at the co-construction and co-operation levels, objective ones at the co-ordination level. So, on the one hand, agents can exploit their (subjective) cognitive capabilities to reason about the coordination needed to achieve the objectives defined at the co-construction stage, and to design the required coordination artifacts at the co-operation stage. On the other hand, objective models and technologies provide the coordination artifacts that make the co-operation stage possible, but are then used in the co-ordination stage, where they embody and enact coordination laws and organizational rules in the most automated, fluid, and optimized manner.

COORDINATION ARTIFACTS IN FIPA

The conceptual framework developed above can now be exploited to integrate objective and subjective coordination in a standard ACL-based MAS. This means modeling and engineering the coordination services typically provided by objective approaches in a world of agents using a high level ACL and related ontologies to interact and coordinate subjectively. In the following, we use the FIPA model as representative of MAS and ACL-based approaches; however, the same considerations and results would also apply for other MAS infrastructure approaches based on high-level ACLs, such as RETSINA (Sycara et al. 2003), which adopts KQML as its ACL. Our target scenario consists then in FIPA agents that can dynamically discover and use coordination artifacts provided by the infrastructure as services to automate the coordination with other agents, which could also be non-FIPA agents involved in the same social activities, in the same organizational context.

The main difficulty that arises in this integration is that the FIPA standard currently does not model interaction between agents and other (non-agent) abstractions, unless the latter are "agentified" themselves by some agent wrappers. More generally, the current FIPA model does not properly account for the relationships and interactions between agents and the (physical and logical) environment: In order to overcome this limitation, investigations and proposals are under development (Odell et al. 2003).

So, the first point here is how to model coordination artifacts as first-class entities in the FIPA world. This raises a couple of relevant issues:

- *Discovery and Awareness*: How do agents discover and become aware of the coordination artifacts available in a specific organization context? How can they understand the coordination services provided by the artifacts?

- *Use & Management*: How can interaction between agents and coordination artifacts be modeled and engineered?

In this paper, we focus on the latter issue. Speech act theory and, more generally, the theory about communication among agents is of no help in modeling interaction between agent and coordination artifacts, since the latter are not meant to be perceived as cognitive entities/agents. The research on theory of action—rooted in works such as Moore (1985) and including speech act theory as a special case—is indeed useful, but at the same time too general. In fact, our aim here is not to model agent actions in the environment in general, but specifically actions over coordination artifacts interpreted as entities featuring specific ontological (formal) properties. In this article, we refer to these actions as *physical acts*, in order to remark the difference with respect to speech acts used for inter-agent interaction.

In accordance to the FIPA model, a coordination artifact is not conceived as a medium enabling and ruling agent communication acts, as it usually happens for coordination media in objective coordination approaches, but as a medium enabling and ruling physical acts executed by agents. Following the AT-based framework, it is then possible to identify the two types of relationships that link agents and artifacts:

- *Agents as Users of the Artifacts*: This includes the physical acts that can be used by agents to access and exploit the coordination service, according to protocols established for their role inside the society and the organizational rules.
- *Agents as Creators/Administrators of the Artifacts*: This includes the physical acts that can be used by agents to inspect/change/adapt dynamically, at runtime, the coordination laws and social norms which define the behavior of the coordination artifacts. Typically, coordination laws and norms are expressed in some specific language, depending on the coordination model adopted, which should be suitably modeled in the FIPA context.

The first relationship typically concerns agents involved in the co-ordination stage of the collaborative activity, while the last typically concerns agents involved in the co-operation stage.

So, in the FIPA context, a coordination artifact can be characterized as an embodied entity providing:

- An interface composed by a set of *operations* which agents can execute to use the coordination artifact. The operations are meant to have a well-defined formal semantics, possibly based on some mentalistic framework. Agent actions cause the execution of artifact operations. An action is generally characterized by two stages: *execution* and *completion*. Action execution is realized with the agent invocation of the action. Action completion instead is realized with the completion of the operation caused by the agent's action execution. Analogously to the FIPA speech act model, we can identify *pre-conditions* and *rational effects* for agent actions. Preconditions include both the conditions that should be satisfied in the mental state of an agent in order to execute the operation on the coordination artifact, and the satisfiability of the constraints about action execution with respect to agent position (roles) inside the organization. Rational effects include instead the new knowledge that the agent can assume by the completion of the operation on the coordination artifact, expressed in terms of the specific semantics and ontology provided by the artifact.

- *Operating instructions*, as a set of instructions for using the coordination artifact, expressed by means of some FIPA ontology. In particular, operating instructions describe the available patterns of operations that allow agents to participate in the coordination activity supported by the artifact. This information is typically useful for agents participating in the coordination stage.
- A coordinating behavior, which makes it possible to achieve the objective of the coordination activity. This information is meant to be inspected and changed by agents participating in the co-operation stage.

The *service* abstraction can work as the bridge to embed coordination artifacts inside FIPA, following the *coordination as a service* approach (Viroli and Omicini 2003). The integration of a new service model with FIPA is actually the goal of one of the FIPA services technical committees (FST 2003). It should be noted that the concepts of service and ontology already permeate the FIPA agent management specification (FAM 2002). However, neither an explicit service model nor a meta-level model for service description are defined there. The final objective is to capture the concepts of service elements and composition at the level of the FIPA abstract architecture, thus enabling the integration with service-oriented technologies such as Web services and Web service orchestration.

In such a conceptual model, coordination artifacts are then to be exploited by FIPA agents by accessing specific services. The notion of physical act can then be used to model in FIPA the interaction between agents and services, which differs indeed from inter-agent communication based on FIPA ACL. Accordingly, embedding coordination artifacts as coordination services requires: (a) the definition of specific service ontologies, providing the semantic description of services required to support agent communication and reasoning about services and (b) the definition of a meta-service (FIPA meta-service) as a special service for management (query information, activation, deactivation), including a discovery service support—similar to FIPA-Yellow Pages service—for runtime discovery and location of existing coordination artifacts.

TuCSoN AS A JADE COORDINATION SERVICE

In this section, we apply the integration framework devised in the previous sections at the technology level, by taking TuCSoN as the reference infrastructure for coordination artifacts and JADE as the reference platform for FIPA-compliant agents.

On the one side, TuCSoN is an agent coordination infrastructure providing coordination services through artifacts called *tuple centers*, which are collected in nodes of the network and belong to specific organizations (Omicini and Zambonelli 1999). Each tuple center is a *programmable tuple space*, that is, a sort of reactive blackboard (Omicini and Denti 2001). Agents interact by writing, reading, and consuming *tuples*—ordered collections of heterogeneous information chunks—to/from tuple centers via simple communication operations (*out*, *rd*, *in*) that access tuples associatively. While the behavior of a standard tuple space in response to communication events is fixed and pre-defined by the model, the behavior of a tuple center can be tailored to the application needs by defining a suitable set of *specification tuples*. In TuCSoN, such specification tuples are expressed in the ReSpecT logic-based language, and define the behavior of a tuple center in terms of *reactions* to communication events which change the tuple center state.

As a result, tuple centers can be seen as general-purpose, customizable coordination artifacts, whose behavior can be dynamically specified, inspected, and adapted so as to automate the coordination stage (Ricci et al. 2003). Also, as TuCSoN makes no assumptions on the model of individual agents, the TuCSoN infrastructure can be, in principle, exploited along with different agent platforms.

On the other side, JADE is a well-known Java-based FIPA compliant platform (JADE Board 2000). Recently, the architecture of JADE has been extended with an articulated service layer, which also supports—other than standard services provided by agents—true infrastructure services which are exploited by agents directly via Java API, rather than ACL-based interaction. Figure 1 depicts the elements of the JADE service architecture that are most relevant here.

Platform, *container*, and *agent* are concepts of the JADE architecture: A platform represents a set of at least one container, and containers are places where agents live. *KernelService* represents a generic service provided by the infrastructure and differs from an agent-level service, which is provided by agents. It is worth noting that kernel services typically provide the so-called *orthogonal services* (messaging, transaction, security), whereas agent-level services typically implement those domain-specific aspects that are amenable

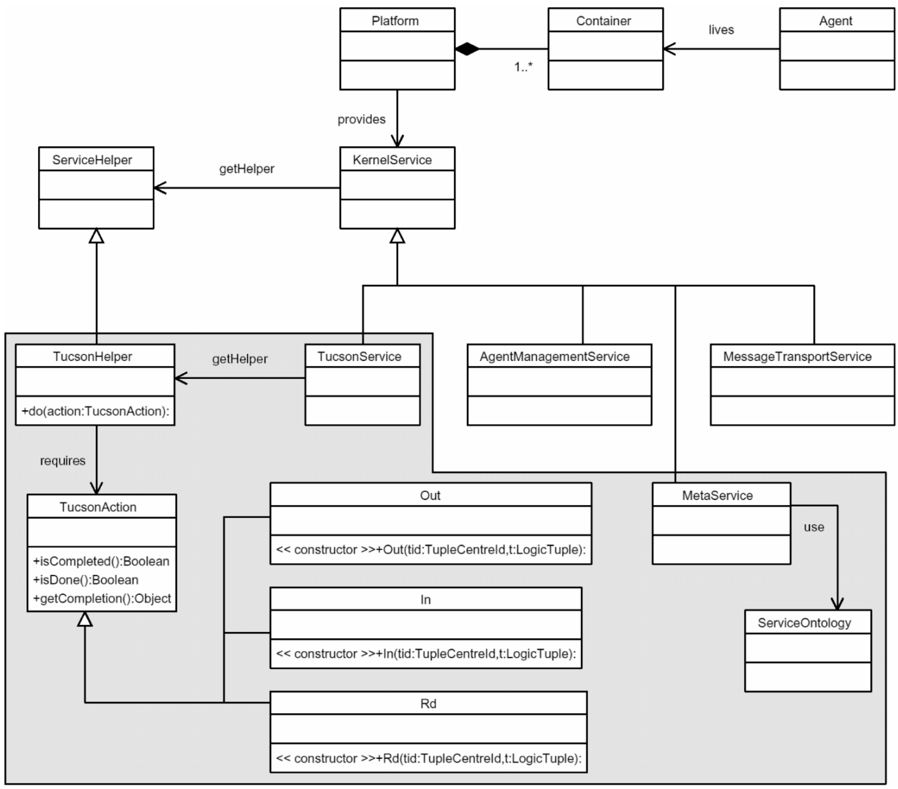

FIGURE 1. Overview of the main components of the JADE platform integrated with TuCSoN. The gray background identifies the parts specifically introduced for TuCSoN, which do not belong to the basic JADE architecture.

to be effectively represented as services (e.g., an agent offering a weather-forecast service). Finally, *MetaService* was introduced for the management (activation, deactivation, and query information) of others kernel services, such as *AgentManagementService* and *MessageTransportService*. In JADE, each kernel service can define a *ServiceHelper*, which is a handle that an agent can use to interact with a service. From a pragmatic viewpoint, a service helper typically manages the marshaling/unmarshaling and the transmission of parameters to the service.

Adding the TuCSoN Service

JADE support for services makes it possible to easily experiment with the integration between TuCSoN and JADE—enabling scenarios where JADE agents seamlessly use TuCSoN tuple centers—by simply conceiving TuCSoN as a JADE service.

Actually, in order to map tuple centers as JADE (coordination) services, there are basically two possibilities: (i) to completely re-design TuCSoN tuple centers to fit the JADE infrastructure (including integration of the topology aspects), so as to embed them directly inside JADE containers and (ii) to build a bridge between TuCSoN tuple centers and JADE service management, allowing tuple centers of a TuCSoN node to be directly accessed from the JADE infrastructure. We opted for the latter approach, which requires no re-design of the two infrastructures (only a suitable service specification in JADE is needed), and also promotes heterogeneity, since it allows JADE/FIPA agents to transparently interact and coordinate with any other agents accessing TuCSoN (including non-FIPA agents).

Figure 1 shows the main components of the integration. *TucsonService* is the TuCSoN service in JADE, derived from a basic kernel service, working as the bridge to access TuCSoN infrastructure; *TucsonHelper* provides agents with the operations, conceived as physical acts, to access and use the tuple centers situated in a TuCSoN node. In particular, the *do* method makes it possible to execute basic operations on a tuple center, namely *in*, *out*, and *rd* (in the figure, *set_spec* and *get_spec* are skipped for readability reasons). TuCSoN actions are explicitly represented by the class *TucsonAction*, modeling the concept of the physical act. Accordingly, *TucsonAction* provides methods to check if the action has been correctly executed (*isDone*), and if it has reached its completion (*isCompleted*). In the latter case, the result of the action can be retrieved by the *getCompletion* method.

As a simple example, the following pseudo-code describes a fragment of JADE agent code interacting with a TuCSoN tuple center:

```
tc_helper = (TuCSoNHelper) getHelper(TuCSoNHelper.NAME);
Out tc_out = new Out(new TupleCenterId("jade@'127.0.0.1'"),
  LogicTuple.parse("test('JADE')")));
tc_helper.do(tc_out);
```

There, the JADE agent inserts the logic tuple `test('JADE')` in the tuple center `jade` located at `localhost` (`127. 0 . 0. 1`) by executing an out operation. Instead, the following code executes a simple in operation of the logic tuple `test (X)` on the same tuple center:

```
tc_helper = (TuCSoNHelper) getHelper(TuCSoNHelper.NAME);
In tc_in = new In(new TupleCenterId("jade@'127.0.0.1'"),
  LogicTuple.parse("test(X)")));
tc_helper.do(tc_in);
//the agent chooses to wait the completion of the action
//get the result
Logic Tuple result = tc_in.getCompletion()
```

An Example

In order to exemplify the approach, we consider here a classic coordination problem, that is, the workflow among a dynamic set of producers and consumers of information: Producers provide information needed by consumers to work, and at the same time, they must synchronize with consumers' activity in order to proceed.

Two main challenging points are raised by this apparently trivial problem. First, the set of producers and consumers is not fixed *a-priori*, and can change dynamically. Second, also the policy defining producers and consumers coordination is subject to adaptation, due to possible changes in the environment. For instance, a consumer may require to atomically receive all information produced by *n* producers.

Modeling and engineering the solution to such a problem only in terms of subjective coordination is indeed very difficult: In fact, it would require complex interaction protocols supporting agent negotiation basically in every stage of agent activity, due to the openness of the problem. Instead, an approach that suitably integrates subjective and objective coordination easily proves to be the most effective choice.

In fact, two basic stages can be identified in the example, which define the system dynamics:

1. A co-operation stage, in which producers and consumers exchange information with a designer agent about their roles and needs, so that the designer agent can properly instrument a coordination artifact they can use for their interaction. This stage is naturally modeled using a subjective interaction protocol, such as the Contract Net, which is initiated by the designer agent where participants are prospective producers and consumers.
2. A co-ordination stage, in which producers and consumers execute their tasks, exploiting the designed coordination artifact to interact.

An integrated JADE/TuCSoN solution is now easy. For instance, the FIPA `call-for-proposals` (CFP) performative could be used for the cooperative stage, and a suitably programmed tuple center (called in the example `sync`, located at the node `deis.unibo.it`) for the coordination stage. In particular, in the co-operation stage, the designer agent issues the CFP to prospective producers, informing them about the details of the coordination stage—the identifier of the tuple center and the syntax/semantics of the basic protocol to be used. A fragment of the corresponding JADE (pseudo-)code is reported in Figure 2.

The protocol used by producers in the coordination stage can be specified using a process-algebraic style, as in Viroli et al. (2004). The variable `producer_protocol` represents the usage protocol for producers. In particular,

```
// Send the CFP to producers
ACLMessage cfp = new ACLMessage(ACLMessage.CFP);
// Defines syntax and semantics of the protocol
cfp.setContent(producer_protocol);
// Set the producers as receivers of the CFP
cfp.addReceivers(producers); ...
// Receive the reply from each producers: for each producers..
  //mt represent a filter on received messages, the agent
  //is interested only at messages from producers agent
  ACLMessage reply = myAgent.receive(mt);
  if (reply != null) {
    // Reply received
    if (reply.getPerformative() == ACLMessage.PROPOSE) {
      ACLMessage accept =
        new ACLMessage(ACLMessage.ACCEPT_PROPOSAL);
      accept.addReceiver(producer); ...
      accept.setContent(producers_protocol);
      myAgent.send(accept);
      // Receive only the reply from current producer
      reply = myAgent.receive(mt);
      if (reply != null) {
        if (reply.getPerformative() = = ACLMessage.INFORM) {
          // Operation successfully completed,
          // the producer will be inserted in producers list
          ...
```

FIGURE 2. Fragment of pseudo-code for the designer agent.

when a producer completes its work: (i) it must insert the tuple product(ID, V) with an out operation in the tuple center sync situated on node deis.unibo.it (ID denotes a value for the identification of producer, and V is a value that is domain dependent) and (ii) it must wait for a consumer that consumes its product with an in operation on tuple consumed(ID). This usage protocol can be expressed as:

$$
\begin{aligned}
\mathtt{PRODUCER} = {} & \mathtt{sync@'deis.unibo.it'?out(product(ID,V))};\\
& \mathtt{sync@'deis.unibo.it'?in(product\_consumed(ID,V))};\\
& \mathtt{PRODUCER}
\end{aligned}
$$

An analogous CFP is sent to the consumers, with a different usage protocol, in particular the variable producer_protocol will be replaced by consumer_protocol with the following content:

$$
\begin{aligned}
\mathtt{CONSUMER} = {} & \mathtt{sync@'deis.unibo.it'?in(products(V1,V2))};\\
& \mathtt{CONSUMER}
\end{aligned}
$$

This allows the consumer to atomically receive the requested information on products. To simplify our treatment without loss of generality, we suppose, in the example, that the consumers require precisely two products.

Figure 3 reports the ReSpecT specification of the sync tuple center that supports such a coordination task, along with a short explanation. The main-point here is that the coordination artifact required for the producer/consumer

```
% consumers
reaction(in(products(V1,V2)),(
  pre,
  out_r(request(V1,V2)))).

reaction(out_r(request(V1,V2)),(
  in_r(request(V1,V2)),
  in_r(product(ID1,V1)),
  in_r(product(ID2,V2)),
  out_r(serve(ID1,V1,ID2,V2)))).

reaction(out_r(serve(ID1,V1,ID2,V2)),(
  in_r(serve(ID1,V1,ID2,V2)),
  out_r(products(V1,V2)),
  out_r(product_consumed(ID1,V1)),
  out_r(product_consumed(ID2,V2)))).
```

```
% producers
reaction(out(product(ID1,V1)),(
  in_r(request(V1,V2)),
  in_r(product(ID2,V2)),
  out_r(serve(ID1,V1,ID2,V2)))).

reaction(out(product(ID2,V2)),(
  in_r(request(V1,V2)),
  in_r(product(ID1,V1)),
  out_r(serve(ID1,V1,ID2,V2)))).
```

FIGURE 3. Specification of the `sync` tuple center. When a producer inserts a new product, a check for pending requests is triggered. If a matching request is found, meaning that at least a consumer request is waiting to be served, the other producer is notified and the involved consumer served. When a consumer tries to retrieve a couple of products (tuple `products(V1,V2)`), a check is made on available products: If both are found, the request is served, and the producers notified.

nsumer workflow can be forged by the designer agent (i) by synthesizing the ReSpecT specification in Figure 3 and (ii) by making it become the `sync` behavior specification through a single `set_spec` operation.

This approach ensures some interesting properties:

- The system is controllable at any time. Since tuple centers are inspectable coordination artifacts, the designer agent can evaluate the behavior of the system over time. For instance, the designer agent could add new specification tuples to the `sync` tuple center to evaluate the global system performance, by recording and monitoring producers and consumers accesses.
- The system naturally allows for change in the productive process. Changing a rule for consumers (e.g., changing the number of products required) can be handled as an ordinary operation because tuple centers are malleable coordination artifacts.
- The system is ready for self-healing. The designer agent (say, *D*) might be intelligent enough to understand that the quality of coordination is low, for example, due to (i) an under-sized artifact, so *D* can proactively choose to add a new coordination artifact to the system and (ii) a super-sized artifact, so *D* can proactively choose to contact new producers and consumers to achieve the optimal results.
- The system is truly uncoupled. Through the initial subjective phase, the system is configured by a designer agent that defines the semantic of interaction. However, different scenarios are possible: For instance, a smarter agent or a team of agents can be leveraged that use more advanced optimization techniques. Co-construction and co-operation stages ensure complete system reconfigurability without affecting the agents involved in the coordination.

CONCLUSIONS AND FUTURE WORKS

In this work, we recalled the basic motivations for reconciling subjective and objective approaches to coordination. By taking the FIPA model and JADE platform as representatives for the subjective approach, and the TuCSoN model and infrastructure for the objective one, we defined the basic conceptual framework for such integration in the context of standard MAS, based on the notion of physical acts of agents over coordination artifacts. The framework has then been applied to make TuCSoN coordination artifacts (the tuple centers) available to FIPA agents as JADE services.

Several related issues are the subject of ongoing research and will be further explored in the future. In particular, a formal model defining coordination artifacts and their perception through agent coordination contexts in the context of FIPA is under development. The model will formally define the operations provided by the artifacts, along with their semantics, in terms of preconditions and rational effects, and artifact instructions, in terms of patterns of executable operations. Some research results using an algebraic framework have been already presented in Viroli et al. (2004).

Future work will be devoted to exploring the modeling and embedding of the notion of agent coordination context (Omicini 2002) inside standard MAS, as a fundamental runtime brick to enable and rule the interaction between FIPA agents and the coordination artifacts.

Also, we plan to investigate the issues of the dynamic discovery of coordination artifacts, and of the understanding of their coordination services by cognitive agents. This investigation will focus in particular on the infrastructure support that standard MAS should provide for the discovery and the awareness of coordination services.

Finally, we aim to test the proposed integration in many specific application domains, by using the available technology, that is, the JADE FIPA-compliant platform and TuCSoN open source technology (TuCSoN 2002).

REFERENCES

Andersen, P. B., P. H. Carstensen, and M. Nielsen. 2000. Dimensions of coordination. In *Proceedings of the 5th International Workshop on the Language-Action Perspective on Communication Modelling (LAP 2000)* 4–16 September, Achen, Germany, pp. 41–60.

Arbab, F., I. Herman, and P. Spilling. 1993. An overview of Manifold and its implementation. *Concurrency: Practice and Experience* 5(1):23–70.

Austin, J. L. 1962. *How To Do Things With Words*. Oxford, UK: Oxford University Press.

Bardram, J. 1998. Designing for the dynamics of cooperative work activities. In *Proceedings of the 1998 ACA4 Conference on Computer Supported Cooperative Work*, 14–18 November, Seattle, WA, USA, pp. 89–98.

Bergenti, F. and A. Ricci. 2002. Three approaches in coordination of MAS. In *Proceedings of the 17th ACM Symposium on Applied Computing (SAC 2002)*, pages 367–372, Madrid, Spain.

Ciancarini, P. 1998. Coordination models and languages as software integrators. *ACM Computing Surveys* 28(2):300–302.

Cost, S. R., Y. Labrou, and T. Finin. 2001. Coordinating agents using agent communication languages conversations. In *Coordination of Internet Agents: Models, Technologies, and Applications*, eds. A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, 183–196. Heidelberg: Springer-Verlag.

Dennett, D. 1987. *The Intentional Stance*. Cambridge, MA: Bradford Books/The MIT Press.

FAM. 2002. *FIPA Agent Management Specification*. http://www.fipa.org/specs/fipa00023/

FDS. 2000. *FIPA Domains and Policies Specification*. http://www.fipa.org/specs/fipa00089/

FST. 2003. *FIPA Services Technical Committee*. http://www.fipa.org/activities/services.html

JADE Board. 2000. *Java Agent DEvelopment Framework*. http://sharon.cselt.it/projects/jade/

Jennings, N. R. 1993. Commitments and conventions: The foundation of coordination in MAS. *The Knowledge Engineering Review* 8(3):223–250.

Jennings, N. R. and J. R. Campos. 1997. Towards a social level characterization of socially responsible agents. *IEE Proceedings on Software Engineering* 144(1):11–25.

Leontjev, A. 1978. *Activity, Consciousness, and Personality*. Englewood Cliffs, NJ: Prentice Hall.

Moore, R. 1985. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*, eds. J. Hobb, and R. Moore, Norwood, NJ: Ablex Publishing.

Nardi, B. 1996. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge MA: The MIT Press.

Noriega, P. and C. Sierra. 2002. Electronic institutions: Future trends and challenges. In *Cooperative Information Agents VI*, eds. M. Klusch, S. Ossowski, and O. Shehory, Vol. 2446 of Lecture Notes in Computer Science, 14–17. Heidelberg: Springer Verlag.

Odell, J., V. H. D. Parunak, M. P. Huget, and R. Levy. 2003. *FIPA Modelling Area: Environment*, Technical Report 030412, Foundation for Intelligent Physical Agents.

Omicini, A. 2002. Towards a notion of agent coordination context. In *Process Coordination and Ubiquitous Computing*, eds. D. C. Marinescu and C. Lee, 187–200. Boca Raton, FL: CRC Press.

Omicini, A. and E. Denti. 2001. From tuple spaces to tuple centers. *Science of Computer Programming* 41(3):277–294.

Omicini, A. and S. Ossowski. 2003. Objective versus subjective coordination in the engineering of agent systems. In *Intelligent Information Agents: An AgentLink Perspective*, eds. M. Klusch, S. Bergamaschi, P. Edwards, and P. Petta, Vol. 2586 of LNAI: State-of-the-Art Survey, 179–202. Heidelberg: Springer Verlag.

Omicini, A. and A. Ricci. 2003. Reasoning about organisation: Shaping the infrastructure. *AI*IA Notizie* XVI(2):7–16.

Omicini, A. and F. Zambonelli. 1999. Coordination for Internet application development. *Autonomous Agents and Multi-Agent Systems* 2(3):251–269.

Omicini, A., A. Ricci, and M. Viroli. 2003. Formal specification and enactment of security policies through agent coordination contexts. *Electronic Notes in Theoretical Computer Science* 85(3).

Omicini, A., A. Ricci, M. Viroli, C. Castelfranchi, and L. Tummolini. 2004. Coordination artifacts: Environment based coordination for intelligent agents. Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '04) 2: 286–293. New York, NY, 19–23 July. ACM.

Ricci, A., A. Omicini, and E. Denti. 2003. Activity theory as a framework for MAS coordination. In *Engineering Societies in the Agents World III*, eds. P. Petta, R. Tolksdorf, and F. Zambonelli, Vol. 2577 of LNCS, 96–110. Heidelberg: Springer Verlag.

Schmidt, K. and C. Simone. 1996. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work: The Journal of Collaborative Computing* 5(2–3):155–200.

Schumacher, M. 2001. *Objective Coordination in Multi-Agent System Engineering – Design and Implementation*, Vol. 2039 of LNAI. Heidelberg: Springer-Verlag.

Sycara, K., M. Paolucci, M. van Velsen, and J. Giampapa. 2003. The RETSINA MAS infrastructure. *Autonomous Agents and Multi-Agent Systems* 7(1-2):29–48.

TuCSoN. 2002. TuCSoN at SourceForge. http://tucson.sourceforge.net/

Viroli, M. and A. Omicini. 2003. Coordination as a service: Ontological and formal foundation. *Electronic Notes in Theoretical Computer Science* 68(3).

Viroli, M. and A. Ricci 2004. Instruction–based semantics for agent mediated interaction. *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi–Agent Systems (AAMAS '04)* 1:102–109. New York, NY, 19–23 July. ACM.