

# Synchronous versus Asynchronous Collaboration in Situated Multi-agent Systems

Danny Weyns  
AgentWise, Dept. of Computer Science  
K.U.Leuven, Belgium  
danny.weyns@cs.kuleuven.ac.be

Tom Holvoet  
AgentWise, Dept. of Computer Science  
K.U.Leuven, Belgium  
tom.holvoet@cs.kuleuven.ac.be

## ABSTRACT

According to the taxonomy for agent activity, proposed by V. Parunak, a collaboration is an interaction between agents of a multi-agent system (MAS) whereby the agents explicitly coordinate their actions before they cooperate. We discuss two sub-types of collaboration in the context of situated MASs, namely asynchronous and synchronous collaboration. After setting up collaboration, the interaction between the agents in an asynchronous collaboration happens indirectly through the environment. Agents direct their actions via the perceived state change of their environment. On the other hand, during a synchronous collaboration agents have to act simultaneously and this requires an additional agreement about which actions should be executed. Although they both fit the characteristics of collaboration, the requirements for their implementation is quite different. Whereas agents in an asynchronous collaboration can be implemented as separate processes that act directly into the environment, the implementation of synchronous collaboration is more complex since it requires support for simultaneous actions. In the paper we give examples of both kinds of collaborations and outline the necessary support for their implementation.

## Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Multi-Agent Systems

## General Terms

Theory, Design

## Keywords

interaction, collaboration, synchronization

## 1. INTRODUCTION

The taxonomy for multi-agent activity, proposed by V. Parunak [3], uses the term *collaboration* for an interaction that

involves communication between the agents, i.e. the coordinating part, and whereby agents select their actions on the basis of a joint intention, i.e. the cooperative part.

From our experiences with the implementation of different forms of collaboration, we conclude that it is important to distinguish between *synchronous* and *asynchronous collaboration*. Although they both fit the characteristics of collaboration they require quite different support for implementation. In this paper we illustrate both kinds of collaboration with examples in the Packet-World and discuss support for their implementation.

## 2. ASYNCHRONOUS COLLABORATION

We start with introducing the Packet-World [2][4], i.e. a simple MAS application we use as a case in our research.

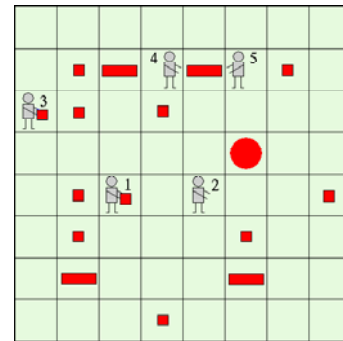


Figure 1: Collaboration in the Packet-World.

In the Packet-World, see Fig. 1, agents have to bring packets (squares and rectangles) to the destination (circle). Agents are allowed to make one step at a time to a free neighboring field or pick up a small packet from a neighboring field. An agent is able to carry one small packet that he can deliver at any free field or at the destination. Large packets have to be lifted up or put down by two agents together. To make a step with a large packet, the two agents have to make a simultaneous step in the same direction. Besides acting into the environment, agents can also send messages to each other to request information or to setup collaborations. The job of the agents is to deliver all packets efficiently, i.e. with a minimum number of steps, packet manipulations and communication. Notice that agents can observe the environment only to a limited extent.

In the depicted situation, agent 2, who is positioned near the destination, is able to form a chain with agents 1 and 3.

Along such a chain, agents can pass packets more efficiently to the destination then when they would deliver the packets individually. To set up such an asynchronous collaboration, the involved agents first have to coordinate, i.e. they have to agree about their role in the collaboration. For the depicted situation, agent 2 becomes head of the chain, agent 1 link and agent 3 tail. Once the collaboration is established, the three agents start passing all reachable packets along the chain towards the destination. During this cooperation, the agents no longer have to communicate. Each agent works in his role, at his own pace, driven by his own perception. As soon as agent 3 has passed the last packet, he signals agent 1 in front of him that he withdraws from the collaboration. If there is still work for agent 1 to do, he continues collaboration (as tail) with agent 2. The collaboration ends when agent 1 informs agent 2 that he has finished his work.

**Implementation of Asynchronous Collaboration.** As soon as coordination is established, each agent in an asynchronous collaboration acts at his own pace, thus the agents of the MAS can be implemented as separate processes that act directly into the environment. With regard to the consistency of the environment, this latter should be synchronized, so that concurrent manipulation of the environment by the agents (e.g. two agents that try to pick up the same packet together) is properly handled.

### 3. SYNCHRONOUS COLLABORATION

Let us now turn our attention to synchronous collaboration in situated MASs. In the example situation of the Packet-World, agents 4 and 5 are involved in a synchronous collaboration. This collaboration started at the time when both the agents picked up the packet they actually carry. In the depicted situation the agents are maneuvering with the packet towards the destination. Before each simultaneous step they make, the agents have to coordinate, i.e. they have to agree about the direction of the next simultaneous step to make. In the depicted situation for example, agents 4 and 5 can agree to make the next step in the direction South-East toward the destination. As soon as the agents arrive at the destination and deliver the packet simultaneously, the collaboration comes to an end.

**Implementation of Synchronous Collaboration.** Since on a single/sequential processor system, actions are executed separated in time, extra support for *simultaneous actions* is needed. Such support involves two aspects: (1) agents must be able to act together and (2) the outcome of simultaneous actions must be in accordance with the laws of the MAS that represents the modelled domain. The implementation of simultaneous actions in a correct and efficient manner is a non-trivial problem. Fig. 2 depicts a generic model for simultaneous actions towards implementation. Before action decision making, an agent triggers the synchronization layer to compose a set of simultaneously acting agents for him. Synchronization can be organized in a centralized [1] or distributed [5] way. Distributed synchronization is more complex to implement, but results in better scalability and autonomy for the agents. After synchronization is established, the agent decides about his next action and invokes it into the environment. This action is reified as an influence and collected by the influence collector. When the collector has received a complete set of influences from a set of sim-

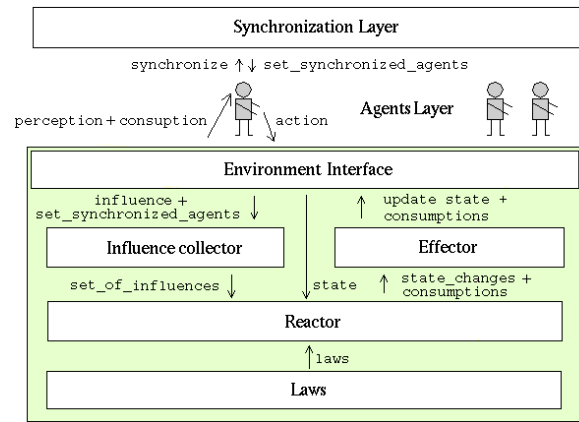


Figure 2: A generic model for simultaneous actions.

taneously acting agents, he passes the set to the reactor. The reactor combines the influences with the valid laws of the domain. The resulting effects are passed to the effector who updates the state of the environment and makes the consumptions available for the agents. A consumption is an element from the environment meant for a particular agent that results as an effect of the previously handled set of influences. Next, the agent can consume his consumption and perceive the updated environment to start a new action.

### 4. CONCLUSION

In this paper we discussed two types of collaborations between agents, namely asynchronous and synchronous collaboration. Both types of collaborations require coordination before the cooperation can start. In addition, for synchronous collaboration the agents also need to coordinate before the simultaneous actions they execute. Whereas the implementation of asynchronous collaboration is fairly straightforward, implementing synchronous collaboration is more complex since it requires support for simultaneous actions. We presented a generic model for simultaneous actions that can serve as a guidance for its implementation. In the model, acting together is established through synchronization, while the reification of actions as influences and subsequently the composition of influences per set of simultaneously acting agents according to the valid laws ensures the correct outcome of the simultaneous actions. For concrete implementations of the model we refer to [4].

### 5. REFERENCES

- [1] J. FERBER AND J. P. MULLER, *Influences and Reactions: A model for situated multi-agent systems*, ICMAS, Nara, Japan, 1996.
- [2] M. N. HUHS AND L. M. STEPHENS, in *Multi-Agent Systems*, G. Weiss ed., MIT press, 1999.
- [3] H. VAN DYCKE PARUNAK, ET AL., *A Preliminary Taxonomy of Multi-Agent Activity*, AAMAS, Melbourne, Australia, 2003.
- [4] D. WEYNS AND T. HOLVOET, *The Packet-World as a Case to Investigate Sociality in Multi-agent Systems*, Demo presented at AAMAS, Bologna, Italy, 2002. [www.cs.kuleuven.ac.be/~danny/aamas02demo.html](http://www.cs.kuleuven.ac.be/~danny/aamas02demo.html).
- [5] D. WEYNS AND T. HOLVOET, *Model for Situated Multi-Agent Systems with Regional Synchronization*, CE '03, AMAS track, Madeira, Portugal, 2003.