Applying ADELFE Methodology to a Mechanism Design Problem

Davy Capera, Gauthier Picard, Marie-Pierre Gleizes
IRIT – Université Paul Sabatier
118, route de Narbonne
31062 Toulouse Cedex - France
{capera,picard,gleizes}@irit.fr

1. Introduction

The ADELFE toolkit [4] enables the development of software with emergent functionality and consists in a notation based on UML/AUML, a design methodology, a platform made up of a graphical design tool and a library of components that can be used to make the application development easier. Likewise, the process is supported by an interactive tool with the aim of easing the following of the process by monitoring the advancement, the produced models and documents, and by providing a guideline example (available at www.irit.fr/ADELFE). Mechanism design consists in assembling mechanical components such as links (rigid bodies, bars) and joints (hinges, cams, etc.), in order to build a mechanical system which performs an specific function such as following a precise trajectory. Our objective is to develop an automated tool – based on the adaptive multi-agent systems paradigm [1]– which is able to synthesize a mechanism from a given set of goals and constraints through a self-assembling process. In this paper, we illustrate the use of the ADELFE methodology to this problem of mechanism design.

2. Analysis

As we are working collaboratively with partners coming from aeronautical industry (European project SYNAMEC), our software – named Mechanical Synthesis Solver or MSS – focuses on solving design problems for "X-bars" mechanisms (see figure 1). This kind of mechanisms contains only three types of mechanical components: rigid bodies, joints (rotoidal or prismatic) and attachment points. Usually, the goal of the mechanism consists in following a given trajectory, in avoiding obstacles and in staying contained in a given envelop. At the highest level, MSS learns a relevant mechanism by using an interaction loop between MECANO and a MAS [2].

The domain description is represented in a preliminary class diagram shown in figure 2.

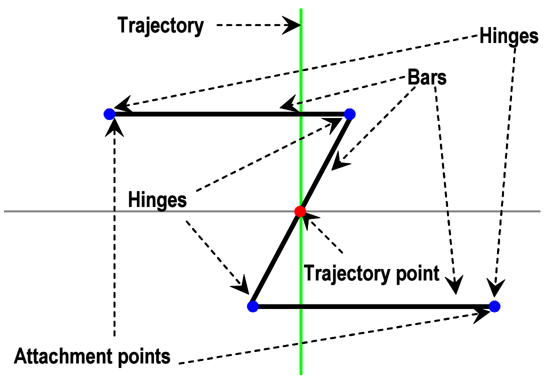

Figure 1. An example of a 4-bars mechanism

The agent definition in ADELFE defines the features that will be ascribed to the entities that the developer will choose to consider as agents. Moreover, ADELFE proposes to analyze preliminary interaction models to identify high-risk entities in terms of cooperation or cognition. ADELFE advocates to identify agents by focusing on active objects thanks to the analysis of information flow from MECANO to each part of the mechanism through class diagrams and sequence diagrams. By using AMAS terms, TrajectoryPoint can be in a NCS –called incompetence– when it does not fit with the trajectory. In the same manner, a RigidBody or AttachmentPoint are useless if not attached to joints, and vice versa. In addition, these MechanicalComponent classes (and its subclasses) and the TrajectoryPoint have to manage their acquaintances not to flood the system by information exchange, and have to be able to add or remove agents from their acquaintance data base. These criteria are sufficient to identify these classes as being cooperative agent classes.

3. Agent Design

ADELFE proposes a generic cooperative agent model [4]. This model decomposes agent's cognitive functions into six modules. The Characteristic module manages the intrin-

AAMAS'04, July 19-23, 2004, New York, New York, USA.

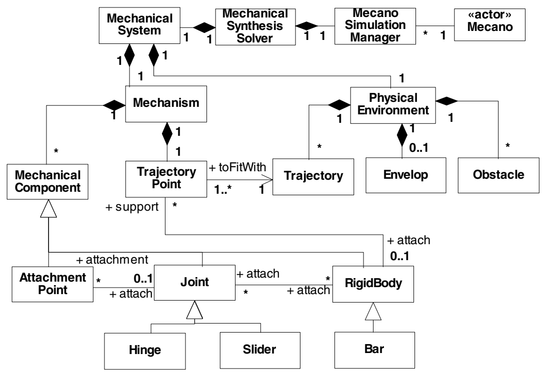

Figure 2. Preliminary class diagram of the MSS domain

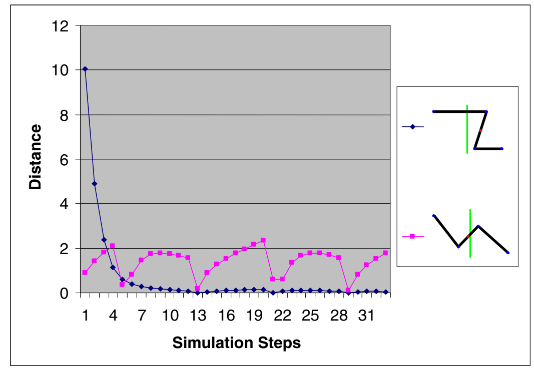

Figure 4. Results for two different 4-bars mechanisms with a reactive resolution

sic properties of agents. The Interaction module is the interface between the agent and its environment. The Representation module manages information about the agent's environment (social or physical) or itself. The Aptitude module manages messages processing according to the message type and action selection. The Skill module is not used for MechanicalAgents because they don't perform their "function" themselves (it is simulated by MECANO) and then they can't have any knowledge about their real activity. The Cooperation module contains the local cooperation rules that lead agents' self-organization. Ideally, every agent knows when it is no more cooperative and then knows what to do to come back to a cooperative state. This is the main design challenge of AMAS: find all the NCS. In the MSS application, two main NCS have been identified: *uselessness* (when an agent does not have all its partners) and *incompetence* (when an agent does not reach its goal).

ADELFE proposes to model agent interactions by specifying AUML protocols [3]. These protocols enable to describe agents' interaction languages (the methods they need

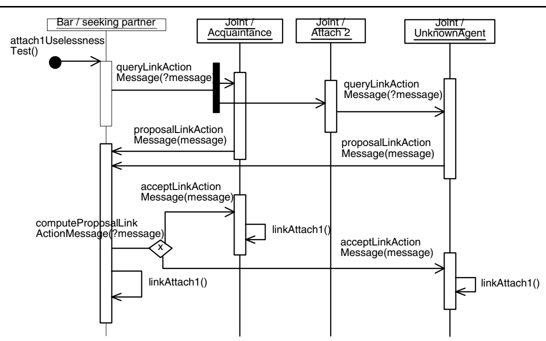

Figure 3. A sample protocol diagram for MSS

to communicate) and resolution negotiation algorithm. For example, the figure 3 shows the resolution protocol for a Bar agent to solve a uselessness NCS.

4. Conclusion and Results

The ADELFE methodology has been applied to a mechanism design problem. The two main phases, analysis and design, led to a self-organizing multi-agent model where each mechanical component is "agentified". Finally, a prototype called Mechanical Synthesis Solver (MSS) has been developed and shows promising results. Firstly, a very reactive algorithm has been tested (see figure 4). The Bar's NCS resolution action is directly induced by the stiffness value it perceives – the Bar applies a translation vector to each of its ends – that leads to a modification of its position and/or length. In the same manner, the TrajectoryPoint computes a translation for its position.

References

[1] Capera, D. and Georgé, JP. and Gleizes, M-P. and Glize, P. The AMAS theory for complex problem solving based on self-organizing cooperative agents. In *1st TAPOCS Workshop at IEEE 12th WETICE'03*. IEEE CS, 2003.

[2] Capera, D. and Gleizes, M-P. and Glize, P. Self-Organizing Agents for Mechanical Synthesis. In *1st ESOA Workshop at AAMAS'03*, 2003.

[3] Odell, J. and Parunak, H.V. and Bauer, B. Extending UML for Agents. In *AOIS Workshop at the 17th National Conference on Artificial Intelligence (AAAI)*, 2000.

[4] Picard, G. and Gleizes, M.-P. The ADELFE Methodology – Designing Adaptive Cooperative Multi-Agent Systems. In Bergenti, F. and Gleizes, M-P. and Zambonelli, F., editor, *Methodologies and Software Engineering for Agent Systems*. Kluwer Publishing, 2004.