MAS Meta-model Adopted in ADELFE

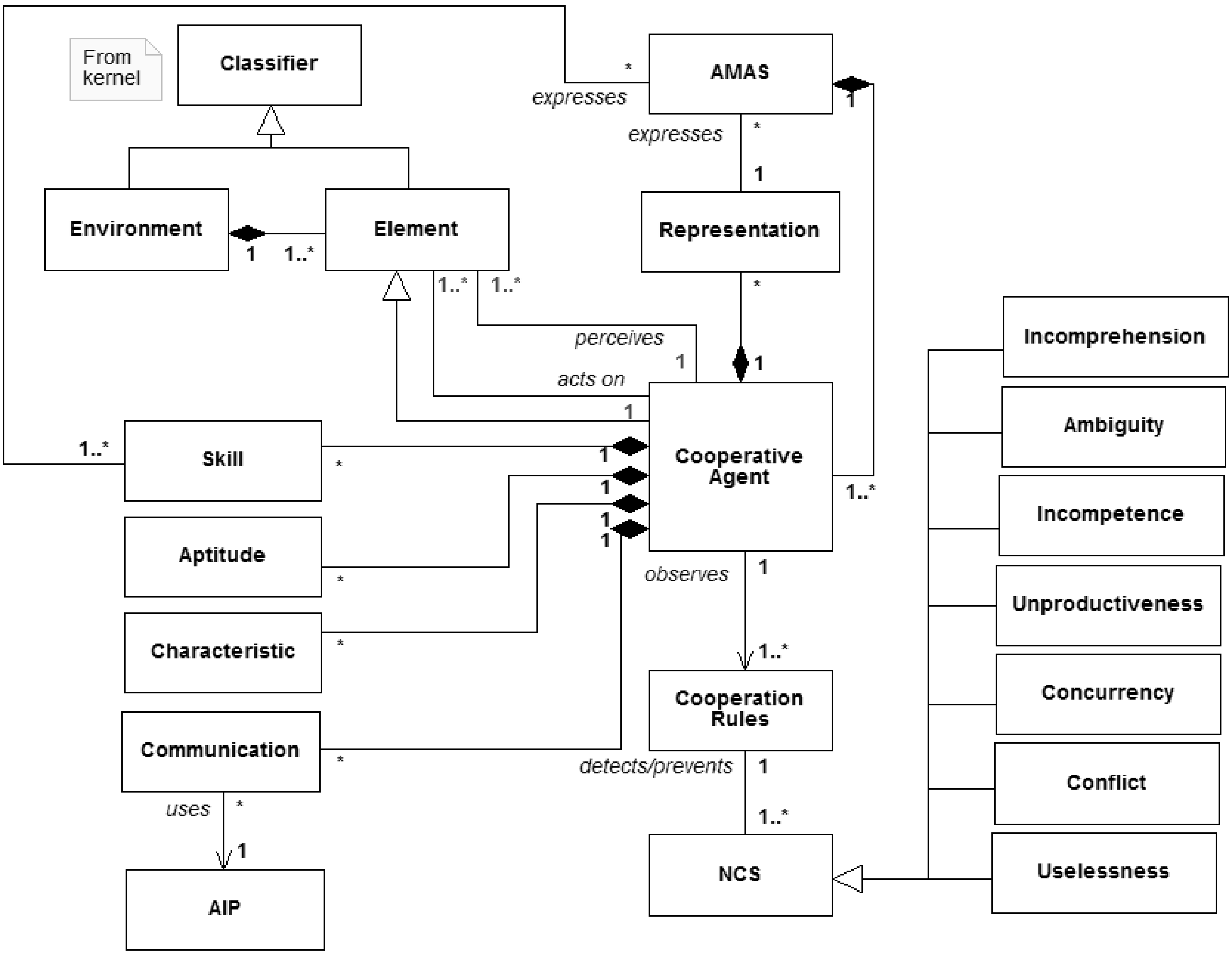
1. Some Words about ADELFE

ADELFE[1] is a methodology that aims at helping a designer to build multi-agent systems (MAS) that are able to adapt themselves when their environment is changing. The design is made by following the AMAS (Adaptive MAS) theory[2] according to which building a system which realises the right desired global function (which is functionally adequate) is achieved by designing agents with a cooperation-driven social attitude.

2. MAS Meta-model for ADELFE

An agent belonging to an **AMAS** ignores the global function of the system, only pursues a local goal and tries to always keep cooperative relations with other agents. Such an agent is called a "**Cooperative Agent**" because its social attitude is based on cooperation but its lifecycle is still a classical one. It consists in having perceptions, taking decisions and then doing actions (perceive-decide-act).

The MAS meta-model adopted for ADELFE is explained by the features such cooperative agents possess.



[1] ADELFE is a RNTL project (2000-2003) which partners were ARTAL Technologies, TNI-Valiosys, L3I (University of La Rochelle) and IRIT (University of Toulouse).

[2] See http://www.irit.fr/SMAC

The **environment** is an important notion for an agent because it may perceive some **elements** (other agents that constitute its social environment or in a more general way, other entities that constitute its physical environment) from this environment. An agent can also act on its environment to behave in a given way. Data (e.g., a move length or the maximum size for a message) or something the agent can perform (e.g., to move or to send a message, FIPA's ACL) may represent an action. These perceptions and actions can be viewed as two kinds of interactions.

An agent uses local **cooperation rules** both to detect and solve Non Cooperative Situations (**NCS**) but also to prevent them to appear. These NCS are cooperation failures (e.g., cooperation protocol not obeyed, unpredictable situation…) that are, from the agent's point of view, inconsistent with its cooperative social attitude. Different kinds of such failures can be detected according to the context of the concerned application, such as **Incomprehension** (an agent does not understand a perceived signal), **Ambiguity** (it has several contradictory interpretations for a perceived signal), **Incompetence** (it cannot satisfy the request of another one), **Unproductiveness** (it receives an already known piece of information or some information that leads to no reasoning for it), **Concurrency** (several agents want to realise the same activity), **Conflict** (an agent wants to realise an activity that prevents another one to achieve its own activity) or **Uselessness** (an agent may make an action that is not beneficial, according to its beliefs, to other agents). When detecting a NCS, an agent does all it is able to do to solve it to stay cooperative for others. For example, faced up with an incomprehension situation, it does not ignore the message but will transmit it to agents that seem (from its point of view) relevant to deal with it.

An agent possesses world **representations** that are beliefs concerning other agents, the physical environment or the agent itself. These representations are used by the agent to determine its behaviour. If an agent has representations that may evolve (e.g., a semantic network), these representations can be expressed using an adaptive multi-agent system (**AMAS**). A representation can be shared by different agents.

An agent is able to communicate with other agents or its environment. This **communication** can be done in a direct manner (by exchanging messages) or an indirect one (through the environment). Tools that enable an agent to communicate are interaction languages. When an agent uses a direct communication through messages exchanges, **AIP**s may also be used to express the communication pattern between agents.

Aptitudes show the ability of an agent to reason both about knowledge and beliefs it owns. For instance, an aptitude of a software agent can be expressed by an inference engine on a base of rules or any other processing on perceptions and world representations. Aptitudes can also be expressed using data, e.g. an integer value which represents the exploration depth of a planning tree.

An agent owns some **skills** that are specific knowledge that enable it to realise its own partial function. For instance, a skill may be a simple datum which is useful to act on the world (e.g., an integer distance which represents the minimal distance a robot has to respect to avoid obstacles) or may be more complex when expressing a reasoning that the agent makes during its decision phase (e.g., a reasoning to avoid obstacles). If they are complex and able to evolve, skills may also be implemented by adaptive MAS (**AMAS**).

An agent may possess some **characteristics** which are its intrinsic or physical properties. It may be, for instance, the size of an agent or the number of legs of a robot-like or ant-like agent. A characteristic may also be something the agent can perform to modify or update one of its properties; for example, if the agent is an ant, enabling it to modify its number of legs.

Carole BERNON, Marie-Pierre GLEIZES and the SMAC team
IRIT, Université Paul Sabatier, Toulouse, France