International Symposium on Aviation Psychology - 2019
International Symposium on Aviation Psychology

5-7-2019

Towards a Meta-Model to Specify and Design Human-Agent Teams

Michael F. Schneider

Michael E. Miller

John M. McGuirl

TOWARDS A META-MODEL TO SPECIFY AND DESIGN HUMAN-AGENT TEAMS

Michael F. Schneider and Michael E. Miller
Air Force Institute of Technology
Dayton, OH, USA
John M. McGuirl
Integrity Applications Incorporated
Dayton, OH, USA

This paper describes work towards developing a meta-model useful in the design and specification of Human-Agent Teams. The meta-model adapts components from the cognitive systems, human factors, software and systems engineering literature to form a model and language which can be applied early in the system design process. The resulting model provides a description of desired system behavior. More importantly, the model produces artifacts useful in deriving requirements for both the human and the artificial agents, as well as for the software/hardware human interface. Insight is also provided for manpower, training, and personnel requirements; as well as, requirements for agent sensing, processing, and actuating. This method has been developed to support student projects in a graduate human-agent teaming course at the Air Force Institute of Technology and has been useful in describing systems employing both embodied and disaggregated agents.

With the advent of agent-based software and multi-agent systems, agent-based modeling languages have been developed within the software development community to specify the intended behavior during development (Deloach & García-Ojeda, 2010). Generally these languages focus on the design and interaction of software agents and consider humans as entities external to the system. However, the terminology and the concepts applied within this domain mimic the terminology that is applied to describe the structure of human-human teams. The intent of the current paper is to examine the expansion of agent-based modeling languages to the description, analysis, and communication of systems which include human-agent teams to aid design and the communication of design to development teams.

Our interest is to explore agents as they apply to automation. Automation is defined as the process of substituting an activity originally performed by a human with an activity performed by a man-made artifact or system (Parsons, 1985). In the systems of interest, we are particularly interested in adaptive automation performed by agents. Expanding on existing definitions of agents (Weiss, 2013), we define an agent as a persistent entity that can: 1) perceive the environment to obtain state information, 2) apply this information to engage in non-deterministic reasoning relative to a set of goals, and 3) apply this reasoning to drive actions in the environment. Therefore, in human-agent teaming we consider humans and artificial agents who collaborate to fulfill a common set of goals. Our approach recognizes that automation may only be possible under certain circumstances, such that successful performance across a broad range of environmental conditions requires collaboration between humans and artificial agents.

In the current context, agents may be either a single physical entity (i.e., a robot) or disembodied and distributed where sensors or actuators are located remotely from the reasoning engine. These distributed components logically perform as an agent. An example of such a logical agent might be a tsunami warning system with distributed sensors to continually gather geological activity and changes in water level, reason about this information, and issue a warning to a human teammate or a broader population. For clarity, we will refer to agents as either human or artificial when it is important to distinguish human from man-made agents.

A meta-model to describe agent-based software systems was proposed by DeLoach and colleagues to describe the Organizational Multi-Agent Software Engineering (O-MaSE) modelling framework (Deloach & García-Ojeda, 2010) . In this model, an organization of software agents is comprised of one or more agents, with each artificial agent playing a role which is designed to achieve a goal. Agents within this model interact with external actors, for example humans. In this model the artificial agents' goals are not shared with the human. Instead, the artificial agents work alone to achieve goals and the humans are exterior to the agent model. While this arrangement may facilitate the design of a software system, is limiting when attempting to design human-agent interaction.

While human and artificial agent goals are not shared in most agent-based software engineering models, it is not common to all. Sterling and Tavateer propose that humans and artificial agents should have common, shared goals within an interaction model (Sterling & Taveter, 2009). In the model they propose, goals within an environment can be decomposed into a hierarchy of goals, similar to the hierarchy of goals used to analyze human activity. These shared goals can provide a starting point from which to design the interaction between humans and artificial agents. Such a design model should permit the designer to consider the activities, personnel selection criteria, and training of the human; the required capabilities of the artificial agents; and requirements for the user interface between the humans, agents, and any machine that they collaborate to control. Thus, we seek to define and explore an agent model that is more suited to the description and analysis of human-agent teams.

Overview of Proposed Meta-Model

The proposed meta model is designed to be integrated with Digital Engineering (*Department of Defense Digital Engineering Strategy*, 2018) techniques to develop multidiciplinary design models, while directly supporting cognitive analysis methods. The result of applying this meta model is a hollistic view of the system design focused on the cognitive elements that produce system performance.

Interdependence as a guiding principle of agent and team design

It is well understood that environmental impacts affect the ability of any system or human to perform any task. For example, environmental effects can deprive human or artificial agents from energy sources, degrade information sources, overwhelm computational capabilities, or change the interaction requirements. Teams are then structured to promote backup behaviors to improve system resilience by exploiting interdependence between agents. Therefore, as we explore meta-models for team design, it is critical that these meta-models include methods to understand and to design in interdendence (Johnson et al., 2014).

Proposed Meta-Model

The proposed structural meta-model is shown in Figure 1. This figure has been arranged so that items on the left side of the figure are heavily influenced by the user of the system prior to the introduction of automation and items on the right side are heavily influenced by the the designer during the design process. Items towards the center require both knowledge from the user and the designer. Elements are joined by associated relationships, indicated by arrows with descriptions; composition, indicated by filled diamonds, genearlizations, indicated by hollow arrows, and aggregation, indicated by hollow diamonds. It is important that the human in Figure 1 represents the system user(s) after the introduction of automation.

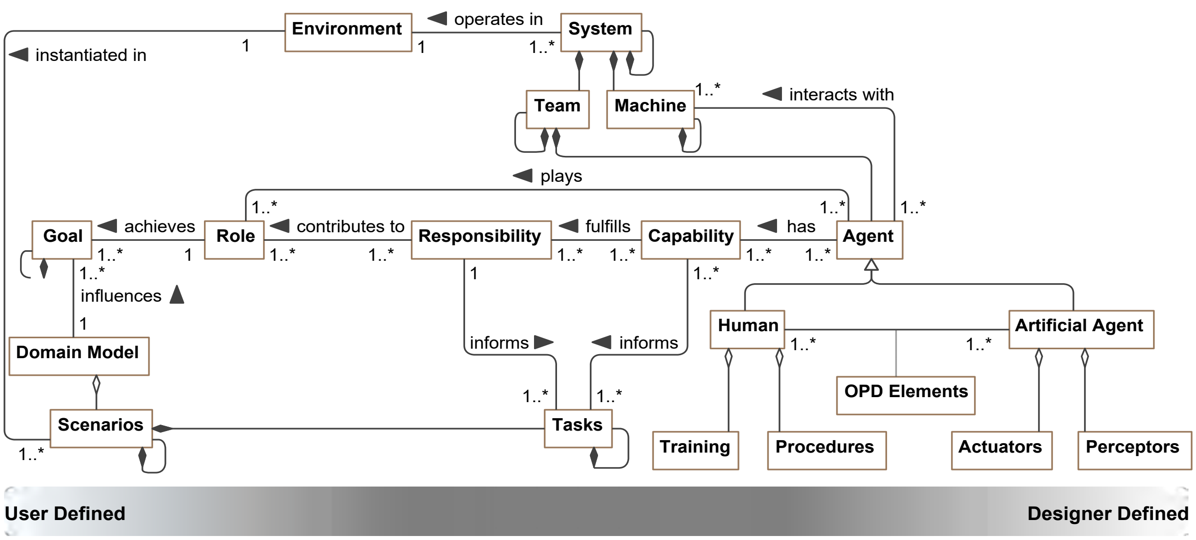

Figure 1. Proposed Meta-model of Human-Artificial Agent Team design in SysML notation

As shown, a *System* may be comprised of other systems, machines, and teams. Systems are operated in an *Environment* comprised of physical (e.g., weather), social (e.g., external collaborators), and informational (e.g., available knowledge) elements. We use the term *Machine* to denote the noncognative, supporting elements of a system (e.g., an airframe). In complex systems it is useful to loosly couple the hardware and supporting software from the agents. Any *Team* may be comprised of other teams and agents (e.g., air crew). An *Agent* is an abstract element which may be specified as either a human (e.g., pilot) or an artificial agent (e.g., autopilot). These agents interact with the one or more machines in the system in which the team is situated. These agents possess capabilities that can fulfill system functions. Of course the difficulty in system design is to understand the capabilities that are required and the interaction of team members to enable the most appropriate capabilities to be applied at the correct time. Understanding and defining these capabilities is central to the design of successful systems.

In our meta-model, understanding the necessary capabilities begins with defining goals. *Goals* are desirable states that must be achieved by the human-agent team (e.g., arrive on schedule)(Deloach & García-Ojeda, 2010). In the goal model, high level goals are decomposed into lower and lower level goals to provide a goal hierarchy. Importantly goals describe *what* needs to be achieved, but do not attempt to describe *how* these goals are achieved. This distinction is critical as high level organizational goals are unlikely to be changed significantly by technology. However, automation often changes how a goal is achieved or who accomplishes the steps to achieve the goals (Endsley & Jones, 2012).

Within this model, agents play *Roles*, which are defined as a distinct set of responsibilities necessary to fulfill one or more goals (e.g., flyer). Roles provide a mechanism to compartmentalize the responsibilities to reduce human training and the communication necessary between team members. These roles are designed to partition the high level activities necessary to fulfill the goals within the goal hierarchy for the human-agent team. At least one agent must play each role, however, multiple agents can contribute to responsibilities to fulfill a role. By abstractly mapping agents to goals, through roles, we avoid definitive allocations and provide the trade space of possible team configurations to achieve a goal.

To complete a role, one must fulfill certain responsibilities. A *Responsibility* is an abstract item which must be accomplished to fulfill a role (e.g., achieve and maintain heading). This requires the ability to perceive certain information, make appropriate decisions, and take particular actions. These responsibilities are intended to be functionally oriented, without any presumed temporal sequence, or particular implementation in mind. Responsibilities are initially derived from the goal hiearchy which typically contains limited temporal information and should be design agnostic. Temporal assessment is handled by scenarios and tasks discussed later.

Once we have defined the responsibilities, we can identify agent capabilities or new capabilities which must be designed to fulfill the responsibilities. *Capabilities* are the ability to complete some action (e.g., control throttles to maintain desired airspeed). These capabilities describe how a specific agent can fulfill a responsibility and must be supported by the agents to permit goal completion. Responsibilities and capabilities describe the possible methods for the team to achieve a goal, but during execution they are tied together by tasks. *Tasks* are temporal sequences of actions that, when executed, fulfill a responsibility. They trace a specific path through the trade space of possible actions that agents can take to accomplish a goal.

The *Domain Model* is a model that captures the policy, resource, value, and technique considerations that influence how the human agent team consider their operations (e.g., air track routes). The domain model influences goal definition and holds a set of scenarios that demonstrate and justify the elements of the domain model. *Scenarios* are specific examples of execution of temporally-arranged tasks by the team to accomplish some set of goals. Synonymous with use cases or user stories, they describe an instantiated form of the environment and are composed of individual tasks. Not all possible scenarios can be modeled, but common, critical, and special interest scenarios provide an oppportunity to identify capability or responsibility gaps when their tasks lack a relationship to the properties of the team.

Analysis Approach

While it is important to understand the capabilities of each of the agents, it is also important to understand that each agent will have certain constraints which limit the conditions under which these capabilities can be applied. Another key insight is that the agents' capabilities are context sensitive. That is, they are shaped, in part, by the environment and thus subject to change as the various constraints (both internal or external) impact the system. It is this context sensitivity that drives the need for an what Johnson termed an "interdependence analysis" as part of the Coactive Design process (Johnson et al., 2014). This process highlights the tasks and constraints that will impact a agent's capabilities and the potential need for multi-agent support or retasking. This analysis also supports the design process by identifying requirements to address observability, predictability, and directability (OPD) issues between agents. Johnson has applied interdependence analysis primarily to relatively simple robotic interactions usually between individual humans and agents rather than the design of multi-agent collections in complex systems as intended in our method. Therefore, the incorporation of this process into our model requires modification. In our model, this analysis is facilitated through scenarios. Scenarios also provide a method to assess new or novel situations and determine how the team could react based on the capabilities and responsibilities required by the tasks. Finding gaps and exploring potential automation surprises contributes to refining the training and procedures the human agents use to establish their capabilities. Complimentarily, the artificial agent's requirements for perceptors and actuators are refined during this same analysis.

Example Application

We applied this meta model to a hypothetical airborne recconaissance mission in a GPS denied/degraded environment. The system, environment, team, and machine elements were modeled using Systems Modeling Language (SysML) external and internal system context diagrams. A goal directed task analysis was applied to develope the goal hierarchy for the geo-locating targets portion of a mission. Six roles were developed by functionally grouping common goals. A total of 23 responsibilities were assigned to these roles with five contributing to multiple roles. These overlaping responsibilities related to commonality in the situation awareness necessary for tightly coupled roles (e.g., a flyer and a navigator). As stated earlier, this portion of the analysis indicates "what" the team needs to be able to accomplish.

In definining the team, a total of five agents, a human pilot, a human sensor operator and three artificial agents were identified. To fulfill the responsibilities, 27 functional capabilities were identified and associated with the responsibilities and agents. Figure 3 depicts the entire definition of the flyer role. Independently, narrative scenarios were developed to test the model. The tasks and domain model were derived from the scenarios with the tasks and capabilities forming the basis of an extended form of interdependency analysis. The analysis and modeling identified missing interfaces between an artificial agent and a human. We also identified sub capabilities that should have been explicitly modeled to provide a clearer understanding of the requirements. These are in addition to the OPD, training, and procedure requirements derived from interdependence analysis, sample in Figure 2.

Mission Phase	Role	Responsibility	Task	Required Capabilities	Primary Agent	Available Capabilities Primary Agent	Secondary Agent	Available Capabilities Secondary Agent	Observability, Predictability, Directability Requirements
Search AOI	Flyer	Fly aircraft	Fly the plane along a search pattern to cover the AOI	Control Aircraft Follow Route Understand Airspace Understand Current	AP	Control Aircraft Follow Route Understand Airspace Understand Current Situation Understand Terrain	PIC	Control Aircraft Follow Route Understand Airspace Understand Current Situation Understand Terrain Understand Threats	The PIC should have a method of placing constraints on the autopilot's execution such that it does not conflict with threats and or weather. There should be a method for the PIC and the AP to

Figure 2. Example of Extended Interdependence Analysis

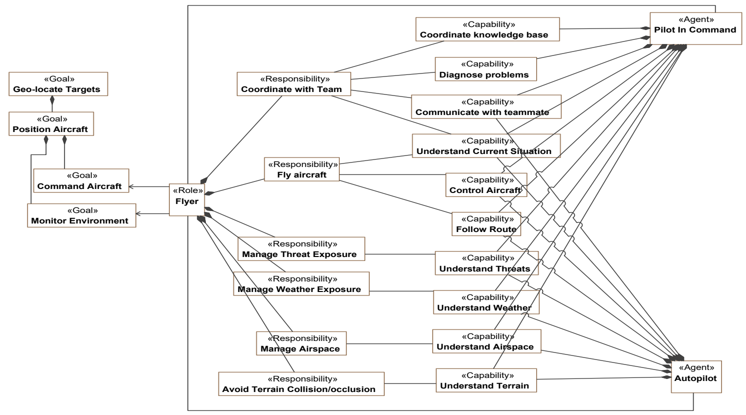

Figure 3. Flyer Role Definition Example

Conclusion

A meta-model for modeling human-machine development was discussed and illustrated through a short example application. It is believed that this approach provides a method for documenting a design for a human-machine team and provides a method for implementing interdependence analysis to the design of complex systems.

Disclaimer and Acknowledgement

The views expressed in this article are those of the authors and do not necessarily reflect the official policy or position of the Department of the Air Force, Department of Defense, nor the U.S. Government. The authors gratefully acknowledge the financial support of the Air Force Office of Scientific Research, Computational Cognition and Machine Intelligence Program.

References

Deloach, S. A. (2004). The MaSE Methodology. In F. Bergenti, M. P. Gleizes, & F. Zambonelli (Eds.), *Methodologies and Software Engineering for Agent Systems* (pp. 107–125). Springer.

Deloach, S. A., & García-Ojeda, J. C. (2010). O-MaSE: a customisable approach to designing and building complex, adaptive multi-agent systems. *Int. J. Agent-Oriented Software Engineering*, *4*(3), 244–280.

Department of Defense Digital Engineering Strategy. (2018). Washington, D.C. Retrieved from www.acq.osd.mil/se

Endsley, M. R., & Jones, D. G. (2012). *Designing for Situation Awareness: An Approach to User-Centered Design*. Boca Raton, FL: CRC Press, Inc.

Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., Van Riemsdijk, M. B., & Sierhuis, M. (2014). Coactive Design: Designing Support for Interdependence in Joint Activity. *Journal of Human-Robot Interaction*, *3*(1), 43. https://doi.org/10.5898/JHRI.3.1.Johnson

Parsons, M. H. (1985). Automation and the individual: Comprehensive and comparative views. *Human Factors*, *27*(1), 99–112. https://doi.org/10.1177/001872088502700109

Sterling, L. S., & Taveter, K. (2009). *The Art of Agent-Oriented Modeling*. Boston, MA: MIT Press.