

Using Scenarios for Contextual Design in Agent-Oriented Information Systems

Kibum Kim, John M. Carroll, and Mary Beth Rosson

Department of Computer Science and Center for Human-Computer Interaction,
Virginia Tech, Blacksburg, 24061, USA
{kikim, carroll, rosson}@vt.edu

Abstract. In this position paper, we argue that current agent-oriented development methodologies are limited in their ability to model social aspects of the agents and human-software agent interactions. We identify how these limitations can be rectified using scenarios for contextual design in agent-oriented information systems (AOIS).

1 Limitations in Traditional Approach for Agent Oriented Development Methodologies

Currently, the primary approaches to agent-oriented development methodologies involve either adopting conventional software engineering methodologies—for example, Agent UML (AUML) and Methodology for Engineering Systems of Software Agents (MESSAGE)—or extending knowledge-engineering methodologies with agent-related concepts such as Conceptual Modeling of Multi-Agent Systems (CoMoMas), Gaia, and MAS-CommonKads [1]. However, because these approaches possess system-centric rather than user-centric natures or designs, they remain inappropriate for dealing with the important developmental process of human-software agent interactions, as well as the human factors for developing interactive agents.

Most problems associated with the adoption of conventional software engineering methodologies derive from the essential differences between distributed objects and agents. In particular, although objects are not social, agents are characterized by their social aspects, and existing software development techniques usually do not adapt to this purpose. In addition, while extending knowledge-engineering methodologies to agent development can provide techniques for modeling agent knowledge, they do not effectively deal with the distributed or social aspects of the agents, or modeling such social interactions. Therefore, theoretical frameworks must be presented that will analyze how people communicate and interact with the variety of agents that constitute their work environments.

2 Scenario-Based Contextual Design in Agent Oriented Information Systems

Increasingly, information technology companies are moving from being technology-driven to being customer-driven, focused on ensuring system functions and structures that will work for the customer. A broad range of analysis involving human-computer interaction has already recognized that system design can profit from explicitly studying the context in which users work [2]. To achieve systems that are more “customer-centered,” one must depend upon Contextual Design as the backbone methodology for front-end design [3]. AOIS can achieve its goal most effectively when its design methodology takes into account what customers need, as well as how human-software agent interactions and social interactions between agents are structured within a usage context. Scenarios—descriptions of meaningful usage episodes—have become key to making abstract models understandable. They help designers examine and test their ideas in practice, with each narrative created to evoke an image of people doing things, pursuing goals, and using technology to support these goals. In *Scenario-Based Design* (SBD), designers also evaluate scenarios through *claims analysis*, wherein the positive and negative implications of design features are considered through “what if” discussions and the scenarios serve as a usage context for considering interaction options. Scenarios and claims analysis are useful in describing initiatives or actions taken by a software agent and considering their usability implications and emphasize the context of work in the real world.

The *Point-of-View Scenarios* (POVs) describe each agent’s responsibilities in the scenario, including the extent of its collaboration with other agents [4]. Creating POVs encourages software designers to anthropomorphize agents and their behaviors, as a heuristic for reasoning about what the agent could or should do to support user needs (See Table 1). They help designers construct an initial analysis of the attributes and responsibilities of individual agents, which might lead them to consider how different agents might influence what users will be able to do with the system. In light of the POV analysis described in Table 1, Table 2 depicts usability tradeoffs that must be considered.

Table 1. Points of view scenarios (POVs) created from the perspective of a software agent

Scenario Agent	Point of View Scenarios
Social Network Visualizer	I was created based on Mrs. Parry’s constant email correspondence with her colleagues. When she first opened me, I asked a database manager for information about her social networks and displayed her personal connections and groups. When she sent email to a new person, I worked with it to set up a new relationship in her social network. Whenever I was asked display myself, I made sure all my nodes and links were shown correctly within the frame.

Table 2. Examples of usability tradeoffs to consider in light of the POV analysis

Scenario Feature	Possible Positive (+) and Negative (-) Attributes of Feature
Automatic creation based on email correspondee	+ assists users by maintaining networks of colleagues, acquaintances, and friends based on their personal histories or behaviors. + can quickly and conveniently explore who knows whom through social networks. - user intentions might be incorrectly determined by the agent. - there is potential for losing user's control, predictability, and comprehensibility.

3 Conclusion

Developing a sound solution for well-defined agent-oriented development methodologies is not an easy task. We have briefly explored the possible contributions that a scenario-based contextual design approach for AOIS might provide to designers interested in pursuing a methodology that can facilitate modeling social aspects of the agents and human-software agent interactions. For AOIS development, we recommend using Point-of-View Scenarios created with the aid of anthropomorphism, because they can envision a user's task in terms of a usage context relevant to the problem domain. Several usability issues are also raised by these POVs. Such a methodology will provide a well-organized framework within which software engineers can more effectively develop agent-oriented information systems.

References

1. Iglesias, C.A., Garijo, M., Gonzalez, J.C.: A Survey of Agent-Oriented Methodologies. In *Intelligent Agents V: Proceedings of the ATAL'98, LNAI*, vol. 1555. Springer, Berlin Heidelberg NY. (1999)
2. Nardi, B.A.: *Context and Consciousness: Activity Theory and Human Computer Interaction*, 1st edn. MIT Press, Cambridge, MA. (1996)
3. Beyer, H., Holtzblatt, K.: *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann Publishers, Inc., San Francisco, CA. (1998)
4. Rosson, M. B., Carroll, J. M.: Scenarios, objects, and points-of-view in user interface design, In M. van Harmelen (ed.): *Object modeling and user interface design*. Addison-Wesley, London. (2000)