

**WER'04 Panel:****Requirements Engineering – Challenges from the agent-oriented approach**

Luiz Marcio Cysneiros  
Dept. of Math and Stat  
Information Technology Program  
York University  
cysneiro@yorku.ca

Many methodologies have been proposed to systematize the software development process. Many of them have been widely adopted. However, the majority has focused on analysis and design. Requirements have frequently been forgotten or only superficially dealt with. In, fact in the past we have seen methodologies evolving from programming. That happened when structured analysis evolved from structured programming and more recently with Object-oriented analysis evolving from object-oriented programming.

Although that may have worked up to some extent, nowadays we are facing much more complex problems to be supported by software systems. The agent-oriented community have introduced new concepts to cope with the needs arisen from these complex problems. Among other concepts, autonomy is something one may strongly expect to find in nowadays software. Software should be able to mimic the ability people have to opt for doing anything they want. Thus a software system must be able to be autonomous to decide whether it should provide a service or not. Nowadays world is also heavily interconnected both by software integration as well as human contact both personally and through any telecommunication's artifact. Thus we have to be able to model and reason about all the social aspects involved in the problem being addressed by the software we are developing. Complex relationships and dependencies must be modeled and understood. Different alternatives to achieve individual or common goals must be addressed. Non-functional requirements must also play an important role during the software development process. Aspects such as privacy, security, reliability should be first class requirements. Methodologies should provide a way to reason and model them since the early stages of software development process.

Introducing these concepts raise many questions that have still to be answered in detail. For example, are the current elicitation techniques adequate to handle such complex problems using such innovative concepts? How should the requirements engineer elicit needs for autonomy? How should he evaluate the social aspects emerging from adopting a new software system? Are the existing agent-oriented methodologies heading to the correct place? Should we use agent-orientation to all new software systems? Or there is a class of problems that would call for using agent-orientation while more well known problems could continue to use current methodologies? These and many other aspects should be investigated by the requirement engineering community to understand how agent orientation will impact requirements engineering itself