An extract from **Jürgen Lind: Issues in Agent-oriented Software Engineering**, LNCS 1957/2001:

In the *strong notion* of agency, an agent is modeled in terms of mentalistic notions such as beliefs, desires and intentions. Furthermore, the strong notion requires that these mental concepts have an explicit representation within the implementation of the agent. Thus, this notion forces a *white-box* on the agent. The *weak notion* of agency, on the other hand, requires only a *black-box* view on the agent in that it defines an agent only in terms of its observable properties. According to this definition, an agent is anything that exhibits autonomy, reactivity, pro-activity, social ability [WJ95].

In my opinion, these two notions of agency are both too strict. I would argue for a more pragmatic definition of agency that allows the designer to decide what should be an agent regardless of a particular implementation or a minimal degree of external properties. I call this the *very weak notion* of agency. To explain why this absence of formal aspects still makes sense, I have to fall back upon a famous article from the early days of Artificial Intelligence.

In [M79], the author argues that it is useful to ascribe mental qualities such as beliefs, goals, desires, wishes etc. to machines (or computer programs) whenever it helps us to understand the structure of a machine or a program or to explain or predict the behavior of the machine or the program. McCarthy does not impose any constraints such as a minimal required complexity onto the entities that we want to ascribe mental categories or onto the mental categories that we would like to use. In his view, ascribing mental qualities is a means of understanding and of communication between humans, ie. it is a purely conceptual tool that serves the purpose of expressing existing knowledge about a particular program or its current state.

> "All the [...] reasons for ascribing beliefs are epistemological; i.e. ascribing beliefs is needed to adapt to limitations on our ability to acquire knowledge, use it for prediction, and establish generalizations in terms of the elementary structure of the program. Perhaps this is the general reason for ascribing higher levels of organization to systems."

To illustrate why this point of view is reasonable, McCarthy uses the example of a program that is given in source code form. It is possible to completely determine the programs behavior by simulating the given code, ie. no mental categories are necessary to describe this behavior. Why would we still want to use mental categories to talk and reason about the program? In the original paper, McCarthy discusses several reasons for this. In the following list, I have selected those reasons that seem to be most relevant to me:

1. The programs state at a particular point in time is usually not directly observable. Therefore, the observable information is better expressed in mental categories.
2. A complete simulation may be too slow, but a prediction about the behavior on the basis of the ascribed mental qualities may be feasible.
3. Ascribing mental qualities can lead to more general hypothesis about the programs behavior then a finite number of simulations.
4. The mental categories (eg. goals) that are ascribed are likely to correspond to the programmers intentions when designing the program. Thus, the program can be understood and changed more easily.
5. The structure of the program is more easily accessible then in the source code form.

Especially the fourth point in the above enumeration is extremely important for AOSE because the task of understanding existing software becomes increasingly important in the software industry and is likely to outrange the development of new software. Thus, if it becomes easier to access the original developers idea (that is eventually manifested in the design) it becomes easier to understand the design and this leads to higher cost efficiency in software maintenance.

A more general conclusion from McCarthy's approach is the idea that *anything can be an agent.* . This view has been discussed from controversial points of view [WJ95] and it has been argued that it does not buy us anything whenever the system is so simple that it can be perfectly understood. I do not agree with this. In my view, the conceptual integrity that is achieved by viewing every intentional entity - be it a simple as it may - in the system as an agent leads to a much clearer system design and it circumvents the problem to decide whether a particular entity is an agent or not. In my personal experience, this problem can be quite annoying during the design phase whenever two software designers have different views.

References

[WJ95] Wooldridge and Jennings, Intelligent Agents: Theory and Practice, 1995

[M79] John McCarthy, Ascribing mental qualities to machines, Martin Ringle (ed.) *Philosophical Aspects in AI*, Harvester Press, 1979.