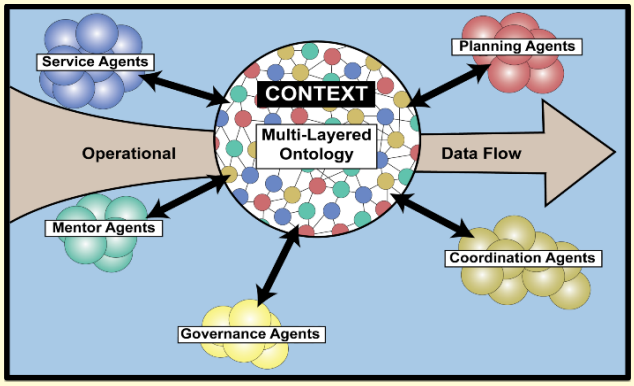
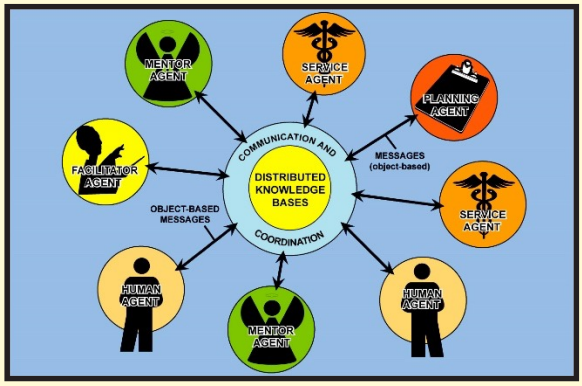
Are Computer Software Agents Really Intelligent?

Surely, computer intelligence is a misnomer? From a commonsense point of view it would appear that humans have intelligence and computers are just very fast but unintelligent machines. Looking at this question from an entirely human point of view we may well come to such a conclusion. However, are there different kinds of intelligence? In other words, is intelligence something that is entirely reserved for living beings or can a machine display behavior that is akin to intelligence?

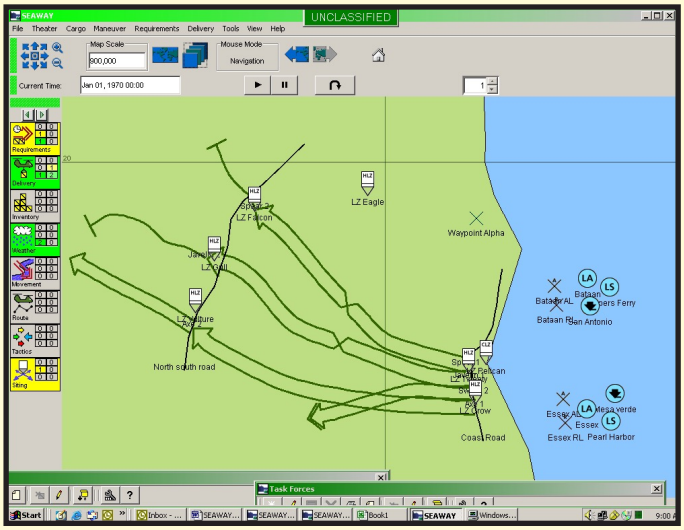
Before attempting to answer this question we should perhaps first address another seemingly less difficult question: Are there different levels of intelligence? If there are levels of intelligence then remembering is probably the lowest level of intelligence. Certainly, computers can store vast amounts of data and can retrieve this data quickly and accurately. The immediate response might be that remembering is more than just retrieving data. Remembering also involves relationships and context. It is this context that makes data meaningful and relevant.

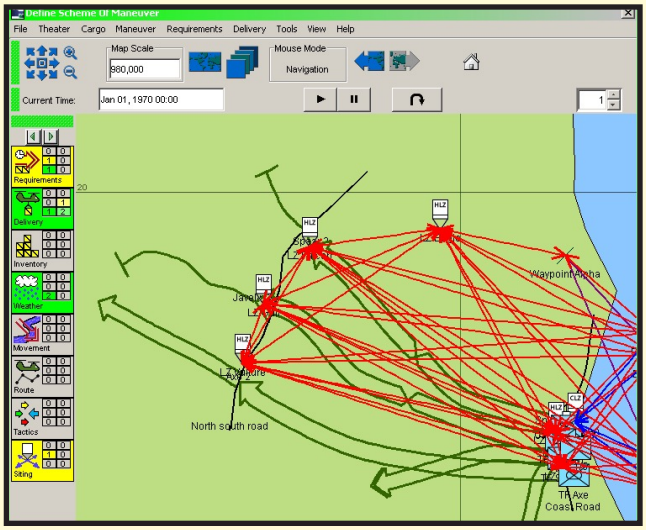




Interestingly enough that is what the current paradigm shift in computer software design and development is all about. We are moving from a data-centric to an information-centric software environment. What this really means is that we are representing information rather than data in the computer (i.e., information is data with relationships to provide some degree of context). This allows us to include modules in the software that are able to automatically reason (commonly referred to as agents) and communicate the results of their reasoning activities to other agents (including human users). One could argue that in some respects we are able to create in this way a virtual copy of a problem situation, or even a limited form of human society, in the computer environment. The players (i.e., the agents) in this virtual society can assume many different roles and can contribute and collaborate at many levels; - from the most primitive to the more sophisticated levels.

So, it seems that if we are careful to store not only data in the computer but also the relationships that convert such numbers and words into information then we can also embed in the software rule sequences that are capable of reasoning about this information. Such sequences may be as simple as condition-action statements. For example, if we are delayed in heavy traffic then our navigation system will automatically suggest an alternative route to our destination. In this way we can implement, through software, automatic reasoning capabilities in computers.





The SEAWAY decision-support application for expeditionary warfare that CHOBU team members developed at CDM Technologies (prior to CDM's acquisition by Boeing) will serve as an example. With a focus on sea-basing operations, SEAWAY includes agents that perform very elementary tasks, such as calculating the fuel consumed by a helicopter in transporting supplies from the sea base to an inland supply point. However, SEAWAY also incorporates agents that perform more sophisticated tasks. For example, selecting the best mix of lift assets (e.g., helicopters, hovercraft, vertical take-off aircraft, etc.) to transport a wide range of supplies to multiple landing zones within requested time windows, and within constraints such as weather conditions, enemy actions, and so on. The latter agents consider results received from other agents, and utilize a wide range of heuristic and algorithmic methods to arrive at a possible solution. In some respects, this is similar to human society where problems are often solved through a team effort. In such teams some people contribute very simple capabilities and others contribute more sophisticated capabilities.

However, in respect to a multi-agent software system like SEAWAY one might ask: Are the combined actions of these agents totally predictable? The answer is, no. While the results produced by the simple agents are certainly predictable, the impact that these results may have on the collective actions of the system is not necessarily predictable. In other words, the intelligence of the software system derives from the interactions (or more appropriately the collaboration) of the communicating elements of the system (i.e., the agents). With some exceptions one would generally not attribute intelligence to any single agent in current information-centric software systems like SEAWAY. However, such systems do display a collective intelligence that is not necessarily predictable and that can be quite powerful.

It therefore seems that we human beings must be willing to accept the proposition that there are different types of intelligence. In other words, intelligence cannot be measured only in human terms. There is no a priori reason to assume that computer intelligence is like human intelligence. In a corollary sense, it is unlikely to be productive to attempt to create a single software agent with human-like intelligence. A better approach is to look upon software as a virtual environment in which many software agents (utilizing their automatic reasoning capabilities, both reactively and proactively) navigate themselves into a solution area, through their countless interactions (i.e., collaborations). When we add the human user (i.e., human intelligence) to this environment we increase the potential capabilities of the system manifold.