**Perfect Disruption: The Paradigm Shift from Mental Agents to ORGs**

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/MIC.2009.15">http://dx.doi.org/10.1109/MIC.2009.15</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Sun Feb 03 21:15:50 EST 2019</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/52392">http://hdl.handle.net/1721.1/52392</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>
The mental agent paradigm has had some success in modeling and simulating human-like behavior. However, computing has changed dramatically since its invention, and we’re now in the midst of a “perfect disruption” brought on by changes to hardware, software, and applications.

The hardware transformation is driven by the many-core architecture, which will soon support thousands of threads in processes for widely used software applications using semantic integration (described later). On the software front, look forward to changes from client cloud computing, in which information is permanently stored in servers on the Internet and cached temporarily on clients that range from single chip sensors, handhelds, notebooks, desktops, and entertainment centers to huge data centers. (Even data centers are clients that often cache their information to guard against geographical disaster.) Client cloud computing will provide much-needed new capabilities such as maintaining client privacy by storing information on servers encrypted so that it can be decrypted using only the client’s private key; providing greater integration of user information obtained from competing vendors’ servers without requiring the vendors to interact with each other; and providing better advertising relevance and targeting without exposing client privacy. In terms of applications, scalable semantic integration is poised to bring together information from calendars and to-do lists; email archives; physical, psychological, and social presence information; documents of all sorts; contacts (including social graphs); and search results.

This column examines how this perfect disruption is causing a paradigm shift from mental agents to organizations of restricted generality (ORGs) as the foundation for implementing large-scale Internet applications.

How ORGs Address the Perfect Disruption

ORGs is a name for the paradigm in which people are tightly integrated with information technology that enables them to function organizationally. ORGs formalize existing practices to provide a framework for addressing issues of authority, accountability, scalability, and robustness using methods that are analogous to human organizations. (Of course, as any reader of Dilbert comics knows, such organizational practices are imperfect.) Functional specialization within ORGs aids scalability by dividing up the workload and aids robustness by requiring specialized suborganizations to negotiate agreements according to their differing viewpoints and responsibilities.

ORGs are structured around organizational commitments, defined as information pledged. For example, ORGs can use contracts to formalize their mutual commitments to fulfill specified obligations to each other. Yet, manifestations of information pledged will often be inconsistent. For example, any given contract might be internally inconsistent, or two contracts in force at one time could contradict each other. Issues that arise from such inconsistencies can be negotiated among ORGs.

ORGs are very well suited to a many-core architecture. They can be implemented by actors – the universal primitives of concurrent computation. ORGs are message composable, meaning that systems are composed by sending messages. Contrast that with update compositibility, in which systems are composed by updating a transactional memory. Message compositibility has numerous advantages over update...
composability, including modularly, scalability, and the fact that message composability works across the entire computational spectrum — from within many-core chips to geographically distributed systems — whereas transactional memory works only within the address space of a single process on a single computer.

ORGs provide guidance on how to extend the X86 architecture (AMD, Intel, Via, and so on) to make it more suitable for many-core to provide

- concurrent nonstop automatic storage reclamation (garbage collection) and relocation to improve efficiency;
- prevention of memory corruption that otherwise results from programming languages such as C and C++ using thousands of threads in a process; and
- nonstop migration of ORGs (while they’re in operation) within a computer and between distributed computers.

ActorScript is a kind of actor programming language that’s well suited for implementing message-composable ORGs. Distributed ORG implementations can accommodate client cloud computing by maintaining some parts on the clients and some on the servers. Scalable semantic integration is also well suited for ORG-based implementation because ORGs supply a framework for implementing two critical components: a scalable semantic engine provides inference and action capabilities for semantic information, and an intuitive semantic user interface based on ontologically extended natural language provides visualization and interactive capabilities for semantic information.

ORGs have important advantages over mental agents, as I describe later.

**The Dream of Mental Agents**

A mental agent is defined behaviorally as cognitively operating in human-like fashion. The paradigm is deeply and pervasively psychological. The most popular kind of mental agent can be characterized as BDIA: beliefs, desires (goals), intentions (plans), and affect (emotions). It has moved beyond its original sequential conceptualization by introducing parallelism, which can be used for low-level I/O (vision, for example), (associative) memory operations, and other basic operations as performed by the parts of the brain. Yet, none of these changes the mental agent paradigm, which draws its fundamental strength from staying close to the mental operations of a single person.

The development of mental agents has continued steadily since the earliest days of artificial intelligence, and researchers have realized impressive achievements (see the Summer 2006 issue of *AI Magazine* for some examples). That said, progress in using mental agents as a foundation for software applications has been frustratingly slow. Impressive demonstrations of mental agents’ capabilities in some application areas have repeatedly failed to garner widespread commercial adoption. Nevertheless, researchers continue the quest to develop mental agent frameworks for software systems. Expressions of confidence and hopes for the future have long been regular features of conferences. Researchers have been heartened because no convincing principled arguments have proven it impossible; indeed, human behavior presents a kind of existence proof that something like the mental agent paradigm can be made to work. Moreover, the community has evolved and gained insights into multiagent systems.

In contrast to the situation with ORGs, the perfect disruption is causing mental agents to lose ground. First, many-core architectures pose a challenge because the computer’s information processing is no longer anything like a person’s information processing:

- Using human-like mental operations becomes an increasing bottleneck as the number of cores grows because the cores perform independent tasks.
- Using a human-like I/O system becomes a bottleneck as the number of interconnections increases because the wires carry independent messages.

Scalable semantic integration requires capabilities beyond those of a mental agent. The current family of standards (such as OWL 2) represent significant progress. However, the semantics of OWL 2 need improvement:

- OWL 2 semantics require that ontologies (theories) be absolutely consistent. A single inconsistency can allow anything and everything to be inferred (for example, “the moon is made of green cheese”), whereas adopting paraconsistency would make it possible to reason more safely in the presence of inconsistent information.
- OWL 2 semantics are for single anonymous ontologies (theories). In contrast, ORGs require the simultaneous use of multiple interrelated ontologies (theories).
- OWL 2 semantics don’t support reification reflection — the ability to abstract propositions from text and to reify propositions to text — which is needed to relate natural language I/O to reasoning.
- Furthermore, OWL 2 semantics are rigid and inexpensive because they don’t support ontologically extended natural language.
Simulated and avatars will be implemented using ORGs!

The mental agent paradigm might increasingly be used in avatars (both human- and animal-like) and cognitive models of individual humans, but operational implementations will require ORGs, just as all large software systems will. The original conception of the development of mental agents is thus turned upside down: instead of ORGs being implemented using mental agents, humans will be and scalability. As Petrie predicted, the old agent technology has essentially disappeared from large-scale software systems. Consequently, a conundrum is emerging, such that researchers must choose whether to

- stay the current mental agents course despite the paradigm switch from mental agents to ORGs, or
- change course and adopt the ORGs paradigm as fundamental, thus begging the question, “Where are the agents?”

Many artificial intelligence researchers have long presupposed that

No software agent architectures can compete with ORGs in understandability, manageability, and scalability.

simulated and avatars will be implemented using ORGs!

The Software Agent Conundrum

According to a published consensus of researchers, a software agent is basically a mental agent adapted for software engineering. More general conceptions have been attempted without success. As Charles Petrie noted, for example, “some have tried to offer the general definition of agents as someone or something that acts on one’s behalf, but that seems to cover all of computers and software.”

Despite many years of trying, none of the software agent development systems for large-scale Internet applications have had any significant commercial success. ORGs (although not currently called such by practitioners) are trumping mental agents for implementing large-scale Internet systems. No software agent architectures can compete with ORGs in understandability, manageability, agents are a principal subject of their field. This is especially poignant for the Autonomous Agents and Multiagent Systems (AAMAS) community, which includes the term in its name twice. AAMAS is a vibrant community whose members are performing exciting and important research, but its conceptual foundations are badly in need of reformulation because of the paradigm shift from mental agents to ORGs.

As Microsoft’s Steve Ballmer suggested in a recent appearance at the Churchill Club, this perfect disruption represents a “perfect opportunity” that’s enabling fundamental new capabilities in the computer industry. I believe his assessment is correct. As a result, IT will be

- easier to use (leading to civilized discourse in interactions with people, unlike current systems, which are often unresponsive and provide no explanations of what they’ve done and why),
- more useful (semantic integration of information from diverse sources), and
- more privacy-friendly because of client-based information integration.

Similarly, IT will become more valuable as revenues increase and new IT employment opportunities arise for ontologists, ORG technologists, and discourse experts. In the midst of the current downturn, we can look forward to a more prosperous future.

No software agent architectures can compete with ORGs in understandability, manageability, and scalability.

Acknowledgments

Charles Petrie suggested the phrase “perfect disruption” and made other valuable suggestions for improvement. Jean-Pierre Briot, Jeremy Forth, Michael Genesereth, Fanya S. Montalvo, and members of the Stanford Logic Group provided helpful discussion and comments.

References
