Agents and Advanced Virtual Enterprises: Needs and an Approach

Report Workshop 2/Workgroup 1

H. Ted Goranson¹, (Ed.), Guillermina Tormo Carbó², Yoshiro Fukuda³,
Lee Eng Wah⁴, James G. Nell⁵, and Martin Zelm⁶

¹*Old Dominion University, USA;* ²*Universidad Politecnica De Valencia, Spain;* ³*Hosei University, Japan;* ⁴*Gintic, Singapore;* ⁵*National Institute of Standards and Technology, USA;* ⁶*CIMOSA Association, Germany, tedg@sirius-beta.com*

Abstract: see Quad Chart on page 2

1 INTRODUCTION

The following Quad-Chart (Table 1) summarizes the work of the group. It identifies the approach taken to address the issues of infrastructures for virtual enterprises exploiting agent technology and proposes future work on agent technologies and modeling languages.

1.1 Background

The working group decided to make an aggressive re-examination of infrastructure approaches for advanced virtual enterprises. The initial impetus for a significant reappraisal came from the report on new enterprise modeling challenges given in “ICEIMT: History and Challenges” in this volume. That report noted that the enterprise modeling problem set is significantly different now than it was for the first ICEIMT ten years ago. Several major constraints and enabling technologies have changed since then. Some early solutions now seem to present barriers. And in any case, the requirements of advanced virtual enterprises are significantly different than for centralized enterprises. Clearly a fresh look is required.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35621-1_43](https://doi.org/10.1007/978-0-387-35621-1_43)

Table 1: Working Group Quad-Chart

EI3-IC Workshop 2 Enterprise inter- and intra-organizational engineering and integration	Workgroup 1 Agents and advanced virtual enterprises: needs and an approach	2002-January 23/25 Gintic, Republic of Singapore
Abstract: Virtual enterprises, especially advanced types, have shown promise for some time but have not yet become common. Some techniques that should facilitate progress are knowledge management, agent systems and enterprise modeling. This workgroup examined how these techniques might be applied in concert for information infrastructure for such advanced virtual enterprises.		Major problems and issues: <ul style="list-style-type: none">— Need a theory of agents that is model centric— Need to migrate from enterprise models that merely represent the process to models that have actors that affect and control process work— Models should be formalized using ontologies— Determine if special modeling techniques are required to support enterprises driven by agents, actors and their models
Approach: <ul style="list-style-type: none">— Re-examine infrastructure approaches for virtual enterprises— Assume traditional processes will remain, same outcome metrics will apply (e.g. profitability), financial markets will regard virtual enterprises as normal enterprises, and it is apropos to integrate at the process-model level— Use the levels of agent capability defined in ICEIMT'97 workshop— The information aspect of enterprise components may comprise agents, sub-agents, and actor-model pairs. Agents consist of subagents, models, and actors, where each actor has a single purpose		Results and further work needed: <ul style="list-style-type: none">— Agents using enterprise models are the triggers that enable model-driven enterprises to work— Enterprise and process models are used for both reasoning about and controlling the processes— The group introduced an ordered way of bringing the notions of distributed-model integration to the virtual enterprise through the mechanisms of agents and exploiting the benefits of knowledge management— Traditional modeling techniques, if done properly, probably are sufficient to represent model-driven enterprises Future work: <ul style="list-style-type: none">— Extend Process Specification Language to be more agent friendly and to include state mechanics to allow models to drive the processes— Assure that the Unified enterprise modeling language under development includes requirements inherited from the PSL extension— Research is needed to develop index systems for existing self-organizing model frameworks

But in order to scope the effort, the workgroup decided to accept certain existing assumptions in order to better focus on more important and leverageable matter. The group assumed:

- That whatever forms advanced virtual enterprises take, they are likely to continue a component/responsibility breakdown along well established functional lines such as marketing, financial, human resources and so on.
- That the same outcome metrics will apply to the combined enterprises and major components that apply to old-fashioned enterprises: profitability and deferred profitability in the form of such things as goodwill, market share, knowledgeable workforce and so on. A corollary of this is that financial markets will evaluate virtual enterprises in much the same way in the future as the regular type they might replace.
- That integration at the process/resource model level is the most promising approach for improvement, for example as opposed to integrating applications, services or product data flows. This mirrors the implicit common denominator of the enterprise modeling community. Moreover, the group asserted that for practical advance in virtual enterprises, existing model and model integration paradigms must apply. This means models and methods covered by the unified enterprise modeling and process specification language efforts, and integrating frameworks along the lines of CIMOSA and GERAM.

In the context of these assumptions, the group focused on the leverage of enterprise modeling and knowledge management in the context of agent-supported virtual enterprises.

Luckily a workgroup of an ICEIMT workshop just a month earlier (sharing some members with this group) had devised a complementary strategy between approaches to knowledge management and enterprise integration. This group adopted all the results of that prior work. (See “A Merged Future for Knowledge Management and Enterprise Modeling” in this volume.) Some key issues of that examination were:

- Modeling of uncertainty, for instance the beginning of organizing a virtual enterprise, while the product or opportunity is still being defined
- Modeling of unknowns, for instance managing placeholders for implicit tacit knowledge
- Managing distributed knowledge in terms of “situated” models. The notion behind this is that agents and enterprise components share information and that information only becomes knowledge when registered in context. That registration, the previous workgroup concluded,

can be largely satisfied by normalizing the information in a model and integrating that model fragment system-wide.

- Accommodating non-deterministic outcomes

Concerning the requirements of virtual enterprises, the group adopted the capability model approach from ICEIMT '97. It defines certain levels of capability:

- The lowest is where agents are not discriminated in the enterprise
- Then add the modeling of the effect of the agent. (This is where most virtual enterprise infrastructure is today.)
- Then add the modeling of the agent and integration in a central “location”
- Then add the distribution and autonomy of the agents
- Then add the ability of an agent to change itself to enhance the system. A key behavior here is when the agent acts in a way that is apparently detrimental to itself. In all this, an agent is equivalent to a virtual enterprise component. So an example of this behavior might be a partner, which sacrifices work (and local profit) in such a way that the whole enterprise becomes more profitable.
- Then add the ability of an agent to negotiate and change others in concert with itself

The focus of the workgroup was on virtual enterprises at the last two levels. This is what is meant by “advanced” virtual enterprises. From that ICEIMT '97 workshop, this workgroup was able to begin with a partial, speculative list of functionally aligned agent types of virtual enterprise components.

These types are shown in Table 2 below. The first column denotes the function of the virtual enterprise component (drawn from ordinary enterprises). The second column has a few examples of agents of that type. For instance the first agent, “Opportunity Agent,” may be a new type of company that only identifies and defines opportunities as the kernel for virtual enterprise formation. The third column captures some of the modeling/knowledge management issues associated with those agents or components. The top row for example shows that the opportunity agent needs to model information that is “soft,” for example product features that are “cool,” customer needs that are dynamic and somewhat unpredictable, customer desires that have a certain measure of fickleness, customer values that reflect familiarity or safety in a brand. The rows associated with Distribution/Logistics/Service and Design/Manufacturing are not very populated because they are well known functions with less difficult and understood modeling needs.

Table 2: Types of agents and their relations to functionality

Marketing	Opportunity Agent	Soft Models (Cool, Safe, Dynamic, Fickle)
	Brand Management Agent	
Legal	Liability Agent	Intent Uncertainty
	Risk Mitigation Agent	
	Performance Trigger Agent	
Human Resources	Certification Agent	Knowledge Representation Knowledge Value Trust Metrics Learning Costs Knowledge Distance
	Knowledge Management Agent	
	Collaboration Catalyst	
	Trust Manager	
Financial	Capital Swarmer	Reverse Activity Costing Trust Aggregation Metrics
	Strategic Metric Harmonizer	
	Dispersal and Loan Manager	
Distribution/Logistics/Service	(Ordinary Agents Omitted)	
Design/Manufacturing	(Ordinary Agents Omitted)	Algorithm Fit Metrics State Maps
	Process Reuse Broker	
	Message Registration Agent	
	State Monitor	
General	Role Manager	Role Models Effect Controls
	Deconstructor	Reverse State Propagation
	Dating Service	Speculative Profiling Layered Virtual Exercising

(As an aside, the workshop was held in Singapore, whose economy is centered in shipping and functions related to expediting and scheduling. Advanced virtual enterprises may have several of the new agents noted above, but it is more likely that the expert operation of any one of them can be the focus for such advanced enterprises. It was speculated that Singapore concentrate on the legal function since many types of virtual enterprises are highly distributed. Shared risk implies fairly complex issues associated with who is responsible for events when material and subassemblies are in transit. A successful component focused on this — and concurrent maritime law with arbitration — could form a reusable component for advanced global virtual enterprises “hosted” in Singapore.)

The group decided to focus on the “coordination catalyst” as an example to focus discussion. That agent represents operations that cover all the interesting challenges. In general, the agent monitors the goals of the entire enterprise together with capabilities (including knowledge) of the various com-

ponents and likely future and alternative components. It manages the optimization of component knowledge and resources in collaboration to optimize the enterprise system. Specific challenges are:

- The product and market of the enterprise system are likely to be partially vague with a set of unknowns and uncertainties. The environment will be dynamic and some dynamism will be unexpected.
- Some of the strategic goals concerning this market will involve indirect benefit such as customer goodwill, brand awareness and improved market share. These “soft” benefits need to accrue in some way to the virtual enterprise components.
- The components must collaborate in a way that individual processes (which contain some tacit knowledge) need to be continuously mapped to global benefit (which is partially soft). Individual and collaborative adjustments in processes must be made to optimize against the global need and or to adjust to changing global need.
- Part of the above includes “learning,” and the learning is of several types as outlined by the previous workshop.
- A risk/reward strategy must be maintained, trust incubated, and controversies arbitrated, all in a context which will be non-linear and likely non-deterministic.
- The potential opportunities of the virtual enterprise are drawn partly by external conditions and partly from enterprise capabilities. But since the composition of the virtual enterprise is effectively unlimited, the potential opportunities are not bounded by the relatively simple constraints of old-fashioned enterprises.

2 THE APPROACH

Following general convention, the group assumed an architecture consisting of virtual enterprise components that may be companies or relatively independent operations of a company. Each component is represented as an agent. Each component is likely to have subcomponents, usually defined by process groups, which can likewise be represented as an agent. Each of these agents communicates by information flows. Some of the agents are automated. Some of the information exchanges are explicit, well-formed messages, perhaps from machine to machine. Models capture the mechanics of the process within each agent. And because models also capture the “process” of agent-to-agent interaction, the structure of the information exchanged is defined by those models. An advantage of agents is that some of them can act autonomously and negotiate solutions from the “bottom up.”

Here is where the real work of the group began. The problem is that at present there is not a well-founded theory of agents that is model-centric in a way that can exploit the relatively mature mechanisms of model integration by frameworks. This is combined with the problem noted in the previous workshop that there is no well-founded collection of practices that relates knowledge in the enterprise in a leverageable way to models of what goes on in the enterprise. The group set out to make the first step toward such foundations, with the goal of defining a new research agenda. In the near term, the group intended to “pass the torch” to the next ICEIMT workshop for further exploration of foundations and research issues.

The participants of the group were talented thinkers, familiar with the nature of representation and abstraction. In general, when faced with a situation where current abstractions are inadequate, the logical solution is to introduce a new layer of abstraction. That way, the resulting demands on the abstraction primitives are reduced, but at the cost of greater complexity in the mechanics of the representation system. The way around that problem is to if at all possible design the new layer so that it uses as much of the existing representation mechanism as possible.

After some experimentation and debate, the group introduced the notion of actor-model pairs as the key components of an agent. Any agent has as its “active” components this pair. An agent can have many such pairs: where the level of granularity of agents varies, and agents can have subagents. The same is not true of actor-model pairs. Actors are extremely simple, capable of one action only. The associated model would be considered a model fragment in the standard lexicon of process modeling. Many such fragments (and associated actors) would be needed to do some meaningful work. For the remainder of this paper, “model” is used for such model fragments.

Note that the model is not just a notational record divorced from the process. Early modeling was indeed that: a way of describing the work without directly affecting or controlling the work. The model in this scheme is used for both reasoning about the process and controlling the process. So the model actually “stands for” lots of stuff within the agent: resources, data stores, sensors, people, perhaps material and so on. The model is both the representation of that part of the world of such stuff that matters, and the surrogate for that stuff.

Consider the example case of two simple agents, each of which consists of one actor-model pair. Typically, each agent would have many such pairs. The primary information flow is between actors: one a “sender” and the other a receiver” actor. Each model is serviced by resources, data stores and so on. The group simplified the exchanges by making each actor capable of only one action. This was at the cost of greater numbers of components, but they all are of the same basic type. The group presumes that model integra-

tion via frameworks can be accomplished feasibly, so further decomposition of models is not unmanageable.

The granularity of the actors is defined by the granularity of the messages between agents. Fortunately, there is a well-founded theory of such messages in both the agent and virtual enterprise worlds. It is based on the same simplification used for the actors: interactions between agents are of a very few simple types. In the agent world, these are called "speech acts," for which there is a robust formal understanding. In the virtual enterprise domain, these are sometimes called "transactions" to emphasize the collaborative nature and the sometimes explicit contractual nature of these messages.

The U. S. National Institute of Standards and Technology has a program to codify virtual enterprise transactions in the context of speech act theory. Typical types are: query, affirm, solicit, request, refuse and ship. There is some debate on what are the optimum few types. A standard would be desirable, though not necessary. The workgroup assumed that this NIST work, or a similar standard that results from it would be the driver in determining the size, type and number of actors and accompanying models.

The introduction of the actor is one of two new layers of abstraction introduced by the group. The actor is suggested, even required by the formalism of transactions, so it goes a long way toward harmonizing the virtual enterprise and agents. But it introduces one new problem on the modeling side. Modeling is sufficiently mature now that there are projects underway to thoroughly formalize the approach. This centers on the notion of an "ontology," which is a formal specification of the laws used by a particular modeling method.

Ontologies are useful in two ways. The first is that they bring a mathematical formalism to modeling; this makes possible such things as automated correctness and systems for reasoning. In fact, the formalism of ontologies is what allowed the previous workgroup to assert that a simple bridge could be made between models and knowledge representation.

Knowledge is "situated" information. The message passed between two agents is simple information. The receiving actor "situates" that information, turning it into knowledge. The actual process is that the actor simply registers the information in the model. But behind the scenes, that model is linked to all other models in the enterprise, so that the act of placing in the model fragment actually places the information, or situates it, in a global context. The formalism of ontologies makes this possible.

Process ontologies have another benefit. Suppose that the originating agent used a different modeling method or lexicon than the receiving one. In this case, the message would consist of the message itself, and the ontological information about that message so that the actor on the right could perform the necessary translations to register it in its differently conceived

model. A simple example message might be: “it is 2:00,” which might be all you need if both models are part of precisely the same world. If not, the ontological wrapper might say: “2:00 is a measure of time; for us, time is an irreversible sequence of half-seconds; for us, 0-12 o’clock always means AM; for us, the number refers to GMT; for us, GMT is...” and on and on until everything is “explained” in simple standard terms that both sides understand.

The standard ontology for process models is the PSL, process specification language. Process ontologies are the trickiest of ontologies; where most ontologies focus on noun-like things, process ontologies are concerned with verb-like things. That means the ontology has to capture the notion of state, because it is a different thing that a process can happen, compared to it has happened, or even that it is happening. The mechanics that PSL uses is the formal mechanism of situation calculus, which can simply define, relate and reason about those three states.

But the group introduced some new states. The process between the model and actor represents a new state: in the originating agent it is a “pre-state” state. A process can now be in the state of having its actor get it ready for happening, or beginning to happen. In sequence, this new state goes in between “can happen” and “is happening.” This is a new complication. The current use of situation calculus can be extended to account for these new state situations, but it is likely not a simple extension.

The introduction of the actor is at some cost, as noted above. But it solves a key problem: the explicit mechanism for “situating” information to become knowledge in a global context. However, it alone does not solve the other problems of concern having to do with soft, uncertain and dynamic knowledge. For that, the group introduced a second new layer of abstraction.

3 A META PROCESS: MODEL AND ACTOR

The new notion is a “meta-actor and meta-model.” The basic role of this pair is to monitor the functioning of the agent and modify it under certain conditions. In this way, the agent can “learn” and adapt its performance. Such mechanisms are common in the agent community, but there is a special, novel constraint proposed by the group.

The clarity of process modeling is the core concept being leveraged. There is no reason that the processes that govern learning be considered — or modeled — any differently than the processes, which actually do the work of the enterprise. The novelty suggested by the group is to have all these actors/meta-actors and models/meta-models use the same conceptual infrastructure — the same speech act performatives as well. The state mechanics

are apparently simpler, not more complex. (This presumption is revisited below.)

In fact, the group believed that except in two cases, there need be no special accommodation. Treat them all the same.

The first special case addresses the integrating strategy for the meta-models apart from the enterprise models. A non-automated example will clarify this. Consider that a function, perhaps a manufacturing cell, is part of an enterprise. It has processes that contribute to the work of the function, and thereby the work of the enterprise. Now suppose that the enterprise hired a management consultant to do process re-engineering. The consultant would introduce actors into the manufacturing cell to observe and change. But the algorithms used to do the process re-engineering would be part of the corporate knowledge of the consultant.

In this scenario, there are two integrated enterprise models: that of the manufacturing enterprise, and that of the consultant. They touch at the level of individual processes, but are integrated separately. If the manufacturing enterprise chooses to become distributed and virtual, with a distributed and virtual model integration strategy, there is no coincident requirement for the consultant to do so as well. In fact process re-engineering processes are likely to be better served by central, CIMOSA-like integration. Those algorithms are probably more static, and the decisions to "trigger" them likely to require more centralized oversight.

More generally, meta-models can refer to, or be shadowed from a central meta-model repository. This is seen as a simplifying constraint, designed to manage the complexity of introducing a new level of abstraction. It also reflects a reality.

But there is another special case that is trickier. The group wanted to explore the ability to have processes that were explicitly adaptive. This would at least require that the meta-model is wholly within the processes of the enterprise. This also reflects a reality: for instance manufacturing managers that are smart enough to improve their processes without the wisdom of an external consultant.

But the additional mechanics of having unconstrained meta-meta-models may not be so friendly. The problem is not additional burdens on modeling, control, or integration because those are all handled as before, in the simple case. It is instead the burden of preventing circular linkages and the state control problem of initiating a change at a high level of a process that "is in the middle" of something at a lower level. Apparently, the circular problem is manageable by restricting what actors can do, and by "hardwiring" them. In other words, the composition of an agent's actor-model composition cannot change from its birth, only the model. So if sufficient checks are done at

birth, the risk of circular processes is eliminated. (Actors can only act on one model.)

The other problem is the problem of state mechanics. Earlier it was mentioned that the state mechanics for the internal actors is simpler than those who act outside the agent. But now it seems that the complexity is the same. In extending the state ontology, the system at all levels needs the state of “something is getting ready to happen” which can alert downstream processes. As noted before, this seems to be a manageable problem.

All in all, the group introduced an ordered way of bringing the notions of distributed model integration to the virtual enterprise through the mechanism of agents and exploiting the benefits of knowledge management. The group believes this preserves and leverages the considerable tools and advantages of CIMOSA-like modeling and integration. While it introduces new levels of abstraction, these appear to be manageable within the current tool framework and by extending existing standards rather than proposing new ones.

4 SOFT: TACIT, UNCERTAIN, DYNAMIC AND NONDETERMINISTIC KNOWLEDGE

The question remains. How many of the “grand challenge” problems identified in the previous workgroup does this resolve, and what additional work needs to be done?

Some problems are solved in a straightforward way: namely those associated with “tacit.” knowledge. Tacit knowledge is all that stuff that you know that you never think about. In a normal interaction there is lots of tacit knowledge. So when one looks at the content of a message, the message has to be considered in context, or “situated,” to use the term from above. For example, if someone says “hand me the small spanner,” the speaker assumes certain tacit knowledge. Such knowledge is the target of much in knowledge management. It is also the bane of virtual enterprise components since there is a high level of tacit knowledge in normal process transactions, and the components don’t have the shared past of built implicit understandings.

The group’s notion of using model integration as a situating knowledge framework provides a formal basis for the identification, explication and management of tacit knowledge. There don’t seem to be any significant technical barriers to exploiting this approach; they all are probably cultural barriers between industrial engineers and social scientists, which is a tacit knowledge problem in itself.

“Dynamic knowledge” is the obvious problem of using models (which are relatively static) to characterize and drive an intrinsically organic, evolving enterprise. The group feels that this problem has been deftly solved by

introducing the meta-actors and meta-models. The reason for this confidence is that the processes of learning are relatively static. Separating the models of work from those of learning about work is a particularly elegant approach to leveraging the two communities without trying to synthesize them.

“Non-deterministic results” are those, which cannot be precisely seen by examination of the populated models involved and their inputs. Typically, non-deterministic results can come from many agents interacting with each other to optimize a result. Often that optimization is unintuitive and possibly would never have been found by predictive methods. The underlying power of virtual enterprises is some measure of non-determinism: such enterprises are expected to improve themselves in ways and at speeds unattainable from centrally managed enterprises.

The architecture described above is a conventional agent architecture that supports any measure of emergent behavior. That is not the problem. The problem is that many external metrics require the enterprise to appear deterministic in important ways. For instance, financiers want to know where their money will be going and how it will map to the working of the enterprise. Some balance of distributed model independence (for non-deterministic behavior) and model integration (for whole system analyses) must be supported. This is left as an open issue for the next workshop.

The final issue is the problem of modeling uncertainty. Uncertainty in this context goes beyond the non-determinism described above. The major need is modeling uncertainty in the external environment, for example to understand completely unexpected, singular threats or opportunities. Often, these can be explained after the fact, but rarely predicted. But there are all sorts of uncertainties with internal processes as well.

What is needed is a model entity for a suspected but unknown fact, or a collection (or “situation”) of them. All of these four issues may be helped by expanding the current use of the situation calculus in process ontologies to a more full-blown situation theory. In particular, situation theory shows significant promise for modeling uncertain and tact situations.

These outstanding issues are passed to the next ICEIMT workshops. The next workshop (number 3) was on “Enterprise Inter- and Intra-organizational Engineering and Integration,” an appropriate topic for these matters.

5 CONCLUSIONS

The workgroup had two sets of results: suggestions for extending existing standards and proposals for new research.

The mapping of agents to virtual enterprise components depends on a standard set of speech act related transaction primitives. A standard set of

transaction primitives should be adopted. This should be relatively easy because small differences among the definitions in common use convey no clear special advantage. The work, which has been started at the U. S. NIST, is a promising start toward such a standard.

The process specification language is mature, workable and well on its way through the standards process. But it was designed to support the translation between model methods, which is the second use described above. It is not agent-friendly, and does not have the additional state mechanics described. The standard should be so extended.

By definition, the workgroup assumed that exploitation of existing modeling methods was necessary. So it is likely that the work on a unified enterprise modeling language does not inherit any new requirements from agent-based virtual enterprises. But this should be checked, with particular attention to the implications of the PSL extensions, and the introduction of the “meta-actor” which can change models.

There is a standard model integration framework, coming from a CIMOSA legacy. This framework presumes a top down, relatively static organization, where all relevant processes are accessible. Again, it was the assumption of the group that the principles in this standard be exploited. There probably is a significant research project required to determine how best to distribute an index that allows distributed, diverse model fragments to self-organize against such an integrated registration framework.

The workgroup believes this to be a highly leverageable approach to evolving an infrastructure for virtual enterprises from existing foundations.

Research projects are needed.

Work needs to be done on the theory of situations to allow PSL’s state mechanics to be extended in a formal manner as noted above:

- Research also needs to be focused on developing index systems for self-organizing model frameworks of the familiar type; and.
- Research must explore the issue of enterprise features that accommodate non-determinism but support apparent determinism.