Results of the Enterprise Project

Jussi Stader

AIAI-TR-209
September 1996

This paper appears in the proceedings of Expert Systems '96, the 16th Annual Conference
of the British Computer Society Specialist Group on Expert Systems,
held in Cambridge, UK on 16-18 December 1996

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

Abstract

In this paper we present the results of the Enterprise project which was concerned with developing a tool set for enterprise modelling. Our approach concentrates on integration, communication, flexibility, and support. We describe the Enterprise Tool Set which uses executable process models to help users to perform their tasks. The Tool Set is implemented using an agent-based architecture to integrate off-the-shelf tools in a plug-and-play style. To ensure effective interchange of information and knowledge between different users, tasks and systems, we developed the Enterprise Ontology which defines terms used in organisations. The project has been successful and valuable insights have been gained and made available. At AIAI, at the University of Edinburgh, we are interested in pursuing intelligent task management, ontologies, and enterprise modelling beyond the end of the project.

Keywords task management, enterprise modelling, agent-based architecture, process modelling

1 Introduction

Modelling the business environment to help management make sound strategic, tactical and operational decisions is the aim of the Enterprise Project.

Organisations are becoming increasingly complex. Competitive pressures require adaptation to rapidly changing markets. This in turn needs to be supported by methods and tools which help to model, analyse and improve various aspects of how a business works and how it is organised. Many such "enterprise modelling" methods rely on pencil and paper. The Enterprise project is aimed at providing an integrated set of computer-based tools to improve on this current lack of support.

An early deliverable from the Enterprise project was the State of the Art Survey [2]. This was a critical review of the state of the art in the application of (mainly) AI and traditional techniques to enterprise modelling. We wanted to understand the current status of technologies available for enterprise modelling and how these may be enhanced and integrated into a framework for improving business performance. The review helped us to set the overall goals for the Enterprise Tool Set.

In brief, the overall goal of enterprise modelling is to take an enterprise-wide view of an organisation which can then be used as a basis for taking decisions. In order to achieve, use, and maintain such an enterprise-wide view strong facilities for integration, communication, flexibility, and support are required. These can be detailed as follows.

Integration must be achieved for relating information to obtain different views of the enterprise, for relating tasks to be performed to the tools that support them, and to establish connections between the tools themselves.

Communication must be achieved between people, ensuring that the enterprise models are shared within the organisation, between tasks that are performed so that information can be used where it is relevant, and between the tools used to perform the tasks so that relevant data can be passed between them.

Flexibility is important to allow an organisation to adapt to changes in its environment, its processes, and the availability of tools. It is also important to allow flexibility in the enactment of processes to ensure that people's time is used as effectively as possible, giving people the choice of what to do and when to do it.

Support must be provided to take care of technical details and to ensure that the given flexibility does not result in confusion and that processes are carried out effectively.

This paper describes the Enterprise Tool Set. The Tool Set helps to capture aspects of a business and to analyse these aspects in order to identify and compare options for meeting the business requirements. The Tool Set provides task management support to users by helping them perform enterprise modelling activities and guiding them through the Tool Set facilities. We have concentrated not so much on the detailed modelling of different aspects of organisations, but on the integration of different views and the support for enactment. The facilities of the Enterprise Tool Set enable:

- building a process model of any enterprise activity,

- enacting the process by directly linking into the appropriate tools supported by an ontology,
- visualising of the process as it is enacted.

The Enterprise project is the UK government's major initiative to promote the use of knowledge-based systems in enterprise modelling. AIAI at The University of Edinburgh is leading the project and their partners are IBM, Lloyd's Register, Logica and Unilever.

This paper describes the approaches taken in the Enterprise project and the workings of a system that was implemented. The paper concludes with the main technological advances achieved and insights gained during the course of the project. For more details on this project's goals and on enterprise modelling see [3].

2 Approach

We developed the Enterprise Tool Set to provide suitable support for obtaining an enterprise-wide view of an organisation. In this section, we first describe our overall approach and fundamental design decisions. We then provide details on the components of the Enterprise Tool Set before describing the overall architecture of the system to illustrate how the components complement each other.

As a basic model of support for enterprise modelling, we decided to use process models which provide a process-oriented view which can be enacted in a running system. We developed a Procedure Builder which supports the capture of process models.

In most organisations there is a variety of tools in use. In general, people know how to use these tools and they are reasonably happy with them. We decided to support the integration of available tools with as little change to the tools as possible, rather than trying to replicate existing tools and their interfaces. The framework for this is an agent-based architecture together with a library to support the process of adding tools to the system, the Agent Toolkit.

As well as providing support for integrating tools, we provide support for enactment of processes. The Task Manager provides integration between the tools and the process models themselves. Furthermore, there is an agenda-style support for enactment.

In order to achieve this high level of integration and to allow all components to communicate effectively, there must be an agreement about the way in which terms are used. We have developed an Enterprise Ontology for this purpose.

In summary, the components of the Enterprise Tool Set are:

- the Procedure Builder for capturing process models (section 2.1.1),
- the Agent Toolkit for supporting agent development (section 2.1.2),
- the Task Manager for integration, visualisation, and support for enactment (section 2.1.3), and
- the Enterprise Ontology for communication (section 2.1.4).

2.1 The Components

As outlined above, the Enterprise Tool Set consists of a few custom-built, special-purpose tools and a flexible set of integrated, off-the-shelf tools.

2.1.1 Procedure Builder

The Procedure Builder is a graphical tool for describing and recording business process models. It uses part of the IDEF3 Process Description Capture Method [6]. Specifically it uses the notation for the Process Flow Networks (PFNs) that are one of the two different views on processes provided by the IDEF3 method. The Procedure Builder allows the user to build process diagrams that are true IDEF3 PFNs, but also to attach additional information to each process. The Procedure Builder uses a process modelling language which was developed during the course of the project. The changes to IDEF3 and the special-purpose modelling language were necessary because of the requirement to capture information specific to the needs of task management. In developing the language we took account of emerging standards, such as PIF (Process Interchange Format, [5]) and WAPI (the Workflow Application Programmer's Interface developed by the Workflow Management Coalition).

The process models captured with the Procedure Builder contain information about relationships between processes:

- decomposition of processes into sub-processes,
- required sequences of processes,
- alternative paths;

and about a process itself, like:

- its name,
- capabilities required to perform it,
- input required by it,
- output produced by it,
- its purpose.

The output from the Procedure Builder can be exported for use directly by the Task Manager. In addition the Procedure Builder can produce reports containing the process diagrams and associated process information.

2.1.2 Agent Toolkit

We investigated a variety of externally available agent-based architecture solutions and concluded that none of them was mature enough to use as-is or met all our requirements. This resulted in us developing our own agent-based solution. This solution is supported by the Agent Toolkit. One of the essential principles guiding its design, is to make the creation of new agents as easy as possible. We want to be able to support as-yet unspecified tools as agents without any redesign of the Agent Toolkit or any other component of the Enterprise Tool Set in order to accommodate these new tools.

When tools are turned into agents, most modifications are required for communication, i.e. creating, sending, receiving, and parsing messages. Therefore the Agent Toolkit's main area of support is in setting up the communication between agents. The agent communication language we support is KQML (Knowledge Query Manipulation Language, [1]); the content of the messages is in KIF (Knowledge Interchange Format, [4]).

There is also an Agent Registration Tool which can be used to register agents with the Enterprise Tool Set, specifying details about the agent, such as its name and its capabilities. Agent registration allows human agents to be registered as well as software agents. The agent-based architecture allows agents to be added and removed from the overall Tool Set in a plug-and-play style.

2.1.3 Task Manager

The Task Manager is the interface between the user and the Enterprise Tool Set. It directly supports the user in performing their current tasks by helping them to follow the tasks' corresponding processes captured with the help of the Procedure Builder.

The Task Manager plans user tasks and the use of agents. Appropriate agents are identified by matching the task's required capabilities (from the process models) against the agents' registered capabilities. This is done at run-time, so that the most suitable agent can be identified, taking into account which agents are available at that time. Thus the use of agents is coordinated at the level of the user's tasks.

The Task Manager keeps track of the progress that has been made during the enactment of a task: it keeps track of which tasks are currently active, which have been done, etc. This progress can be visualised in different ways, the visualisation being supported by the process diagrams captured with the help of the Procedure Builder.

The Task Manager also handles outputs generated by tasks. It stores that information and it can pass it on as input to other tasks. The amount of support that the Task Manager gives to the user is flexible, depending on what the user wants. Advice can be given on what to do next, determining which tasks are ready to be executed and taking into account what has recently been completed. The Task Manager can also help the user to recover from failures, determining alternative routes of action with the help of the process models.

The Task Manager effectively puts an extra layer of control on top of the agent services. This lets the user participate in the coordination of agents, according to the tasks in which the user is engaged.

2.1.4 Ontology

We require an effective communication mechanism to achieve integration of a wide variety of tools, both new and old. For example, the Task Manager must know whether the output (say $T1_o$) from one tool might be appropriate as an input (say $T2_i$) to another tool. Due to terminology differences between the two independently developed tools, $T1_o$ and $T2_i$ might have the same name, but be different things. Conversely, they might be the same thing but with different names. To resolve such issues, a standard terminology is required to be used when the tools are integrated. An ontology provides such a standard terminology. For an introduction to the field of ontologies and their development and use see [8].

The Enterprise Ontology was developed to provide a standard language for the Enterprise Tool Set as a basis for communication. It is a set of terms frequently used in enterprises, each carefully defined to conform as best as possible to common usage. We have concentrated on the following areas:

- organisation,
- strategy,
- activities and processes, and
- marketing.

The Enterprise Ontology is available on the world wide web [7].

Committing to this ontology has the advantage that terms are used consistently and unambiguously throughout the enterprise (at least when the Tool Set is involved). The Task Manager understands this standard language, and all agents must conform to it to ensure effective integration. The ontology thus provides the basis for communication between agents, whether they are human or software agents.

We expect every organisation to have their own set of terms that they use, so we did not try to define all terms that are likely to be required. The terms we defined are ones that we expect to be generally useful. The ontology can be extended to suit the specific needs of the organisation.

In order to support task management, we have had to add another section to the ontology, concerning capabilities. Full support for coordinating the use of agents can only be given if capability terms are used consistently during the capture of process models and during agent registration. The terms of this ontology are organised into a hierarchy to provide more suitable support for agent registration and process modelling. For example, specifying "calculate" as a capability for a calculator is easier than specifying all the individual calculations it can do.

2.2 The Architecture

The general architecture of the Enterprise Tool Set is shown in figure 1. In the figure, Tool Set components are shown as rectangles, information like models and data is shown in diamonds and users (different roles) are shown as ovals. Arrows show information flow.

Tools are turned into agents by agent programmers, in most cases by adding a communication layer to the tool. This can be done with the help of the Agent Toolkit. All software agents must be able to communicate using KIF and KQML and make their capabilities available to the Tool Set. Tools that have been turned into agents can be registered with the Enterprise

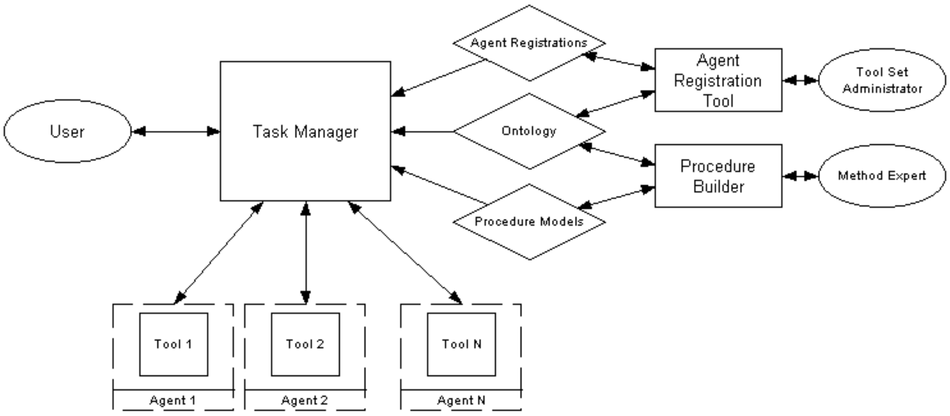

Figure 1: Enterprise Tool Set Architecture

Tool Set by the tool set administrator. The agents are registered stating their types (software or person) and their capabilities. The Ontology helps to ensure that terms are used in a consistent way. Once an agent has been registered its capabilities can be called upon by the Task Manager for the enactment of tasks.

Processes are captured by the method expert using the Procedure Builder. Again this is done using the ontology. The processes can be loaded into the Task Manager and are then used as a basis for supporting users and for coordinating the use of agents.

The Task Manager is the interface between the Tool Set and the user. It is the heart of the system and provides the main mechanism for integration and support. It makes use of all other development and modelling efforts, like process capture, agent programming, and ontology development. On the task level, all communication goes via the Task Manager. However, during task execution, a tool may "take over" and use its own user interface to communicate with the user (these communication links are not shown in the diagram).

3 How it Works

As indicated above, there are several phases of working with the Enterprise Tool Set. Apart from initial specifications these phases are relatively independent of each other and we expect that these phases will involve different users. The phases are:

- extend the ontology to include additional concepts specific to the desired application,
- transform tools into agents,
- register agents with the Tool Set,
- generate process models,
- make process models available to the Tool Set,
- enact process models.

The first five phases can be seen as support phases for the last one, but they will have their own benefit, independently of their use for the tool set. For example, modelling work (ontology and processes) usually is a useful exercise in itself to gain insights into structures and processes, and it often shows scope for improvements.

In the rest of this section we consider the last phase (enacting process models) and we concentrate on the Task Manager, as it is the driving force of the Enterprise Tool Set. For this we will assume that the Tool Set is ready to be used, i.e. the system includes results from the first five phases above.

3.1 Modes of Operation and Settings

During nearly all interactions with the system the user is in direct control. The level of support can be set such that the Task Manager takes more or less initiative, but in general the user can choose freely what to do. Choices include working on tasks, browsing models or other information, leaving a task in mid-flow and starting another task, going back to a previous task, etc.

There is a considerable amount of flexibility in the way the Task Manager can look and behave. For example, the system can be run in simulation mode without its agent network (e.g. for instruction or to check process models), and the user can ask for different amounts of progress information to be provided. There are also different levels of support at which the Task Manager can operate, ranging from full task management to no support at all. The levels of support are:

1. Full task management: the Task Manager always tries to find a suitable task to perform next (using the criteria described below) and it starts the task on behalf of the user. This mode is particularly useful for routine tasks and for quickly running tasks in simulation mode.
2. Run the next task: In this mode, the Task Manager again works out what can be done next. If there is only one task that can be done next, it starts that task on behalf of the user. If there is more than one, it will offer these doable tasks to the user for selection.
3. Suggest the next task: this mode is like the previous one, but it does not start tasks on behalf of the user, even if there is only one possible next step. It always offers the possible next tasks for selection, making sure that tasks are only started when the user is ready for them. This is the default level of support and probably the most useful one.
4. Background task management: In this mode the Task Manager does not volunteer any information. It does not offer tasks for selection or select them for the user. Basically, it lets the user get on with things. However, it keeps track of what is going on and the user can ask it for advice.
5. No support: the user can choose to run tasks in isolation or to just run agents without specifying which task they are used for. In these cases, the Task Manager does not try to keep track of what is going on. It goes away and when the user asks it to come back, it will not be aware of what the user has done in the meantime.

In the next section, the level of support is assumed to be the default level. However, many of the points apply to all levels of support, apart from the last one (no support).

3.2 Operation

Initially, the user identifies tasks that are to be performed during the session (today's work). To identify tasks for which the Task Manager can provide support, the user can browse the process models which are currently available in the Task Manager (from the Procedure Builder). There are various visualisations that allow the user to inspect details of these models.

3.2.1 Planning

When a high-level task is identified for execution, the Task Manager plans this task: it first breaks the task down into its full hierarchy of decomposition (sub-tasks). Then it goes through each of these tasks to identify all agents that can perform it, checking the agents' advertised capabilities against the task's required capabilities. Agents are then ranked according to how well they can perform the task and the best agent is selected. If there are tasks for which no suitable agent can be identified, this is noted but it does not stop the Task Manager from progressing.

The next planning step is to determine at which level to execute tasks in the hierarchy. If a task is broken down into sub-tasks, the Task Manager can either support the execution of the task itself, or it can help the user to step through the sub-tasks. The Task Manager bases its decision on the availability of agents to perform the tasks. The level that is best supported by agents will be chosen, and high levels are preferred to low levels. For example, a high-level task could be "Analyse Bid". If there is an agent "Bid Manager" which has just the right capabilities to perform that task, then the Task Manager assumes that the agent is a special-purpose tool which provides specific support for this task and is thus likely to be better suited for supporting the task's enactment than the Task Manager itself.

When the Task Manager has selected agents and levels of execution, it is ready to start running. The user can inspect the results of this planning phase using different visualisations.

3.2.2 Running

The Task Manager goes through all tasks that are to be done and identifies the ones that are ready to execute. A task is ready if its required inputs are available and all tasks that are to be done before it have been completed. The Task Manager then offers these tasks to the user for selection. When the user has selected a task, the Task Manager checks whether it has an agent selected for it and if that agent is available. If there is a problem it again tries to find a suitable agent. If it cannot, it will ask the user which of the currently available agents should be used.

The Task Manager then gathers the information required by the task from its store of information generated previously by other tasks. It asks the agent to start the task, passing it the inputs along with the request. The task is then assumed to be active and marked as such in the progress visualisation window.

The user may be required to interact with the agent in order for the task to be performed. However, it is up to the agent to organise this interaction through its own user interface.

The Task Manager does not have any information about how long the task is going to take (seconds, days, months...) so it does not wait for a reply from the agent. It can ask the agent for a progress report if the user requests it, or it can wait for the agent to tell it that the task has been finished. If the task has been performed successfully, the Task Manager obtains the task's output information from the agent and adds it to its store of information. The task is then noted as "done" and the Task Manager re-evaluates all relevant tasks as to whether they are ready to be executed. When these new doable tasks are offered to the user for selection, any successors of the task just finished will be shown before other doable tasks, i.e. it assumes that the best advice is to follow the temporal links in the process model.

When a task fails, the Task Manager gives the user the option to re-start the task. Failing that, it looks at the process model to find alternative courses of action (e.g. loading data from a file *vs* entering it by hand). If there are such alternatives, they are taken as options to recover from the failure and offered to the user.

This process of identifying executable tasks and managing their execution is repeated until no more executable tasks can be found or until the user stops the process. At that point the user receives a report on the overall progress of the high-level task, which can be a note that the task is done, or information about which tasks failed and which are still to be done.

4 Conclusion

The approach taken during the Enterprise project is proving to be appropriate. The Enterprise Tool Set has been implemented and a demonstrator has been built to illustrate the advances made.

In this section we evaluate the components of the Tool Set separately, before describing the main technological advances and indicating areas which we intend to take further in future.

4.1 Evaluation of Components

The Procedure Builder is an advance on earlier software for capturing processes, in that it is able to capture and export models in a form that can be enacted.

Like most agent infrastructures, the Enterprise one provides services for communication between software applications. Enterprise goes further than most, in that it provides an Agent Toolkit which makes it easier to add agents.

The Task Manager provides the overall integration between process models and tools, passing information between tasks. It is unlike other agent systems in that it puts an extra layer of control on top of the agent services. This lets the user participate in the coordination of agents, according to the tasks in which the user is engaged. It thus provides the user with a better chance to make effective use of the flexibility provided by agent-based technology.

We believe that our Enterprise ontology is distinct from other efforts at developing ontologies for enterprise information in that:

- it attempts broad coverage to include most terms important to enterprises. Most others address limited areas;
- it exists in the form of a comprehensive, carefully prepared natural language glossary *and* in a formal language. Others must be gleaned from various scattered papers, or exist mainly in formal languages and are thus inaccessible to non-technical readers.

4.2 Main Technological Advances

The major strength of the Enterprise Tool Set is that a process model can be built of any business activity which can then be used without modification to enact the process. The steps in the process are linked directly into the tools required to perform them. This enables the most suitable tools currently available to support the business process. In addition, the tool set provides a visualisation of the process illustrating progress as the process is enacted.

4.3 Lessons Learnt and Future Extensions

Developing the Enterprise Ontology was not easy. However, we are pleased with the final result and during the project we developed a method for building ontologies (see [8, 9]). We intend to build on this experience and extend the existing Enterprise Ontology and develop new ontologies for other domains.

Discussions during the first phases of the project and the designs of the Tool Set addressed many important and interesting issues which we had to disregard during the implementation as they were outside the scope of the Enterprise Tool Set. We intend to go back to these ideas and investigate them further.

An important example is the use of a central repository in which enterprise information is kept. With such a repository, it would be possible to generate multiple views (visualisation) of information, illustrating different aspects of the information.

Another aspect that has been discussed in some detail is translation, both between different ontologies and between different agent communication languages. The Enterprise Tool Set would benefit from translators, because they would provide more flexibility for agent communication and would reduce the demands made of the agent programmers. (Some of the task management aspects of this issue have already been addressed, but we decided to focus our efforts on more central issues.)

Enterprise has many aspects that are suitable for adding coordination technologies to support groups of people who contribute to a common process. For this, the current approach to task support can be extended to support multiple users. It should be possible to take advantage of results of other projects which concentrate on aspects of coordination.

5 Acknowledgements

The work of the Enterprise project is supported under the DTI's Intelligent Systems Integration Programme, Project Number 8032. I acknowledge the contributions to this project from the partners on this project: IBM UK, Lloyd's Register, Logica, and Unilever. I would also like to thank other staff at AIAI for their contributions.

References

[1] T. Finin, Don McKay, and Rich Fritzson (Editors). An overview of KQML: A Knowledge Query and Manipulation Language. Technical report, The KQML Advisory Group, March 1992.

[2] J. Fraser and A. Macintosh, editors. *Enterprise State of the Art Survey.* The Enterprise Consortium, September 1994. Available from AIAI, The University of Edinburgh.

[3] John Fraser. Managing Change through Enterprise Models. In R. Milne and A. Montgomery, editors, *Applications and Innovations in Expert Systems II.* SGES Publications, December 1994.

[4] M.R. Genesereth and R.E. Fikes. KIF knowledge interchange format version 3.0 reference manual (logic-92-1). Technical report, Logic Group, Computer Science Department, Stanford University, California, 1992.

[5] Jintae Lee, Michael Gruninger, Yan Jin, Thomas Malone, Austin Tate, Gregg Yost, and other members of the PIF Working Group. The PIF Process Interchange Format and Framework Version 1.1. Technical report, MIT Center for Coordination Science Working Paper 194, Cambridge, MA, 1996. Also available from AIAI as AIAI-TR-198.

[6] R. Mayer, Cullinane T., P. deWitte, W. Knappenberger, B. Perakath, and S. Wells. IICE IDEF3 process description capture method report (al/tr-1992-0057). Technical report, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 1992.

[7] The Enterprise Partners. The Enterprise Ontology. Available on World Wide Web at URL
http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html.

[8] M. Uschold and Gruninger M. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2), 1996. Also available as AIAI-TR-191 from AIAI, The University of Edinburgh.

[9] Mike Uschold and Martin King. Towards a Methodology for Building Ontologies. Technical Report AIAI-TR-183, AI Applications Institute, 1995. Presented at the Workshop on Basic Ontological Issues in Knowledge Sharing Held in conjunction with IJCAI-95 in Montreal.