Integrating Manufacturing Softwares for Intelligent Planning-Execution: A CIIMPLEX Perspective

B. Chu [1] W. J. Tolone[1] R. Wilhelm[2] M. Hegedus[3] J. Fesko[3] T. Finin[5] Y. Peng[5] C. Jones[3] J. Long[1] M. Matthews[3] J. Mayfield[5] J. Shimp[4] S. Su[6]

[1]Department of Computer Science
UNC Charlotte, Charlotte, NC 28223
[2]Department of Mechanical Engineering
UNC Charlotte, Charlotte, NC 28223
[3]IBM
8501 IBM Dr., Mail Group 70, Charlotte, NC 28262-8563
[4]QAD Inc.
10000 Midlantic Dr., Mt. Laurel, NJ 08054
[5] Department of Computer Science
UMBC, Baltimore, MD 21228
[6]Deparment of Computer and Information Sciences
University of FL, Gainesville, FL 32611-6125

ABSTRACT

Recent developments have made it possible to interoperate complex business applications at much lower costs. Application interoperation, along with business process reengineering can result in significant savings by eliminating work created by disconnected business processes due to isolated business applications. However, we believe much greater productivity benefits can be achieved by facilitating timely decision-making, utilizing information from multiple enterprise perspectives. The CIIMPLEX Enterprise Integration Architecture is designed to enable such productivity gains by helping people to carry out integrated *enterprise scenarios*. An enterprise scenario is triggered typically by some external event (e.g., delay of a shipment). The goal of an enterprise scenario is to make the right decisions considering the *full context* of the problem. Enterprise scenarios are difficult for people to carry out because of the interdependencies among various actions. One can easily be overwhelmed by the large amount of information. We propose the use of software agents to help gathering relevant information and present them in the appropriate context of an enterprise scenario. The CIIMPLEX enterprise integration architecture is based on the FAIME methodology for application interoperation and plug-and-play. It also explores the use of software agents in application plug-and-play.

1. Introduction

Application interoperation and plug-and-play are a focus of many major research initiatives, e.g., FAIME [2], NIIIP [10], OAG [15], OMG (particularly its vertical application SIGs) [16]. A common objective of these initiatives is to provide an enterprise solution consisting of multiple COTS (commercial off the shelf) applications. Such a solution has the following characteristics:

- Changes made in one application are reflected consistently within other applications in real-time (or near real-time).
- Tasks started in one application are continued or completed in other applications as dictated by business process semantics, and without human intervention.

The impending diffusion of interoperation technology also will pose many new challenges [24]. Critical among these challenges is our ability as system designers to extend *application interoperation* to what we term *enterprise integration*. Application interoperation aids business process reengineering by, for instance, eliminating many of the manual steps devoted to synchronizing isolated applications. Enterprise integration, however, must address the new business processes that emerge as people are required to view information from multiple perspectives as offered by different applications. Such business processes can dramatically increase a person's cognitive work load. We argue that intelligent assistants are needed to provide context sensitive help and to coordinate the execution of these business processes. Other important, though no less critical, challenges include:

- Supporting more complex collaborations among applications. When applications automatically feed information to one another, support for feedback, look-ahead simulation, and optimization becomes natural requirement. Such support introduces additional challenges in terms of overall system convergence.

- Managing an increasing influx of data. Application interoperation can produce massive amounts of real-time data. Effective filtering and mining of such data for decision support is critical to realize the full power of cooperating applications.

With the help of NIST/ATP's Technology for Integrated Manufacturing Application (TIMA) initiative, the Consortium for Intelligent Integrated Manufacturing Planning and Execution (CIIMPLEX) was established (1996) to address key aspects of the above areas in the domain of manufacturing enterprises. From a manufacturing enterprise perspective, CIIMPLEX has the following goals:

- To enable manufacturers to make plans based on real-time capacity information.

- To monitor manufacturing plan execution using real-time data and provide intelligent assistance in decision making.

- To develop an open architecture to enable the above functions to be delivered through the plug-and-play of COTS applications.

CIIMPLEX is realizing these goals through the *CIIMPLEX Enterprise Integration Architecture* that utilizes the latest technology of software agents.

Throughout this paper, a running example of updating a purchase order is used to illustrate a business process enabled by interoperation technology. This example is not intended to represent fully the types of enterprise business processes enabled by such technology; rather, it is to be a framework against which relevant discussion can occur.

Consider the scenario of updating a purchase order where the shipment date on a purchased part is changed. Manufacturing enterprises need assistance in evaluating the impact of such an event of the manufacturing plan in order to recommend, for example: (a) The manufacturing plan is still feasible, no action is required; (b) Order substitute parts; (c) Reschedule; or, (d) Reallocate available material. Unfortunately, the information and analysis functions required to make such recommendations are contained in multiple applications, e.g., Enterprise Resource Planning (ERP), Manufacturing Execution System (MES), Customer Response (CR), and Scheduling. The goal of the CIIMPLEX project, in part, is to help people make such decisions by providing appropriate, intelligent assistance.

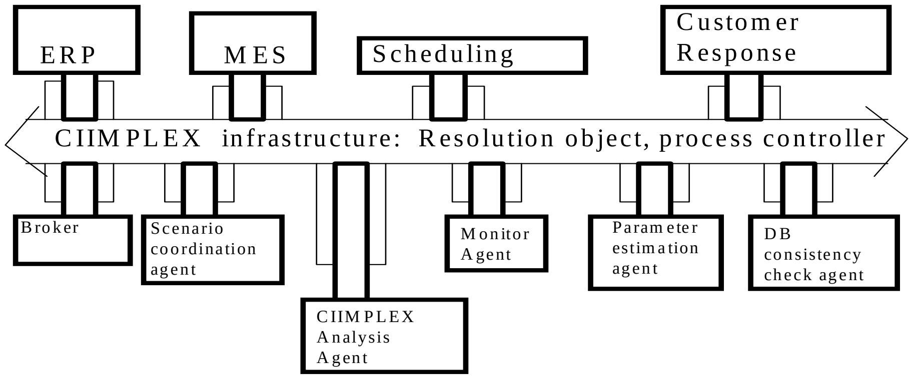

Figure 1. CIIMPLEX Enterprise Integration Architecture

.

2. The CIIMPLEX Enterprise Integration Architecture

Figure 1 illustrates the CIIMPLEX enterprise integration architecture. The main objectives of the CIIMPLEX enterprise integration architecture are:

- Support a new class of integrated business processes made possible by interoperating applications. These business processes are represented as *Enterprise Scenarios*. An enterprise scenario, such as the change of purchase order scenario described earlier, is typically triggered by some external event (e.g. delay of a shipment). The goal of the enterprise scenario is to make timely decision considering the *full context* of the problem. An enterprise scenario typically involves tasks spanning multiple applications and multiple organizations. *Scenario coordinators* are software agents that help people to carry out enterprise scenarios by offering:
    - inferencing for routine tasks;
    - information and actions in the context specific to particular enterprise scenarios;
- Use software agents to provide early detection of potential problems, harvest information based on raw real-time data, feed back controls, and look ahead simulation / optimizations. System convergence issues will also be investigated. Many thousands of transactions occur daily in a business enterprise. Software agents can be used to filter relevant information much like they are being used to find relevant information on the internet.
- Provide an open framework for plug-and-play application interoperation. This is primarily achieved through the system interoperation infrastructure developed through the FAIME project [2].

Software tools will be developed for power users to define / customized software agents. When an agent detects a situation where human decision is necessary, a scenario coordinator will be created to interact the appropriate users. Figure 2 illustrates the relationships between business transactions, background agents, scenario coordinators, and users.

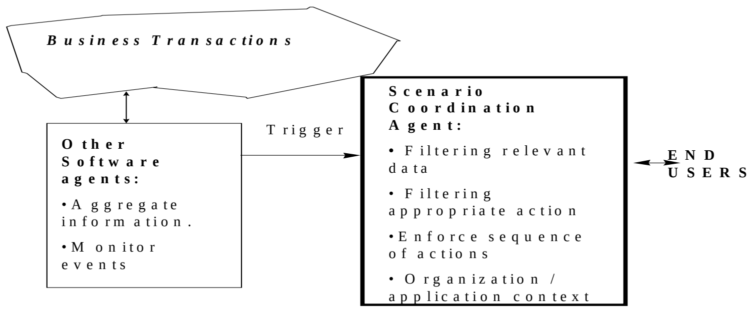

Figure 1. *Relationship between business transactions, background agents, scenario coordinators, and users.*

3. The role of software agents in CIIMPLEX

In CIIMPLEX agents fulfill two broad category of functions:

Agents serve as *information lenses* that provides relevant information in the context of specific problems. These agents perform the following types of functions:

- *Data mining / Parameter estimation:* aggregate, interpolate and extrapolate from raw transaction data to generate information for higher level analyses, characterize unreliable data, and to fill in missing data.
- *Problem detection:*
  - CIIMPLEX analysis agent: look ahead simulation and sensitivity analysis
  - Database consistency checking
  - Event monitoring agents. New agents can be created by the end user to monitor ad. hoc. situations. They can also be configured to take automatic actions.
- *Scenario coordination*: assist the execution of enterprise scenarios by providing relevant context, including filtered information, actions, as well as workflow. It will use the types of agents described above.

Agents are used to facilitate application *plug-and-play* by providing brokering services.

One of the key technical objective of the CIIMPLEX project is to design an agent communication language which enables software agents to:
- Exchange state information.
- Instruct each other about new functions, using both a declarative language as well as mobile code segments (written in JAVA).

The agent communication language, based on KQML and KIF [5] will have primitives, some of them are specific to the domain of manufacturing enterprises. It will contain operators that allow new functions to be created using a declarative language.

3.1 Mining Agents: Parameter Estimation

Mining agents (MA) harvest raw transaction data and produce more meaningful (usually aggregated) information for higher level analysis. Examples of mining activities include:

- Estimating the distribution of scheduling or planning parameters to characterize process uncertainty and data reliability. In these cases, standard distributions may be fitted to production data or sampled data may be maintained as histograms.

- Estimating parameters that cannot be directly measured. For example, legacy applications used by a manufacturer may only collect work in progress (WIP) status information through material consumption events. MA can be used to estimate the status of a particular unit of WIP based on information in the legacy application and historical data for similar WIP transactions.

Mining agents will be based on a reusable framework that includes statistical knowledge such as distribution estimation, analysis of variance, and response surface modeling. Using the CIIMPLEX agent communication language, these MAs can be dynamically "instructed" to mine new types of information.

3.2 Problem Detection

This class of agents are used to detect problems in a timely fashion.

Event monitoring
Event monitoring agents (EA) are responsible for detecting events that might be of interest to different group of users. Examples of such events include: changing of purchase order due date, and machine break down. A set of events and their corresponding EA's that are of general interest to manufacturing enterprises will be defined as a base implementation of the CIIMPLEX methodology. Through the CIIMPLEX build time environment, end users can define additional EA's.

EA's can also be configured to take actions automatically. These actions include:
- Start anenterprise scenario by using a process controller [2].
- Send messages to other agents, e.g. send messages to start a scenario coordination agent.

The CIIMPLEX Analysis Agent
The CIIMPLEX analysis agent (CAA) is specially designed for the manufacturing application domain. It takes as input a manufacturing plan. Part of the plan, within the immediate time horizon will be in the form of a schedule[1]. It evaluates disturbances to the current schedule and determines what actions should be taken to address each disturbance. In cases where no action is suggested by CAA, the disturbance is effectively filtered from the attention of planners and decision-makers. The CAA works from the current manufacturing plan and reacts to MES data transactions collected by event monitoring agents. Typical CAA tasks include:

- Periodic calculation of projected completion dates for all orders in the manufacturing plan based on real time data from the MES. For late orders, an estimated range for delays is calculated (e.g. order xyz will be 2 to 4 days late) and alternatives are suggested for corrective actions with typical examples shown below.
    - Get X amount of material or substitute parts by date Y.
    - Work X amount of overtime, or delay scheduled maintenance.
    - Reallocate available resources.
    - Regenerate the plan to see if there is a better alternative.

- Response to what-if queries from planners, e.g. what if the due date of a certain order is moved up 2 weeks?

- Analysis and suggested actions according to conditions/rules defined by planners at specific manufacturing sites. For example, a particular site might always employ the following rules:
    - Overtime should not exceed Z hours.
    - When orders of type xyz are late by more than 3 days, regenerate the plan.

[1] A schedule is defined as a set of sequences of jobs to be carried out by various resources (e.g. machines, people) on the manufacturing floor.

A critical research issue for CIIMPLEX is to provide guidance to ensure system convergence as well as providing mechanisms to detect non-converging behaviors. CAA will play a key role towards these objectives.

Database consistency checking

A logically consistent database is the foundation for the CIIMPLEX enterprise integration methodology. The CIIMPLEX infrastructure contains mechanisms to synchronize database changes under normal usage. However, the CIIMPLEX infrastructure can by no means guarantee consistency. The following is a partial list of possible ways inconsistencies can occur over time:

- Failure to recover from database transactions that synchronize databases. Distributed ACID (atomic, consistent, isolated, and durable) transactions are still very difficult to implement, especially for legacy applications. The CIIMPLEX infrastructure relies on manual recovery for certain failure modes.
- It is sometimes undesirable to propagate certain local updates that are transient in nature (e.g. modifying an MES routing to accommodate a very special circumstance for an order). Propagating such changes may cause the whole system to "churn" with undesirable effects. However, these transient updates may become "permanent" leading to inconsistency.
- Missing requirements (use cases for interoperation): failure to identify cases where databases should be synchronized.

Software agents can be employed to audit "consistency indicators" which are selected data elements in different databases. Some indicators can be dynamically created and transmitted. For example, if a transient update occurs, the DB consistency checking agent can be notified of such an update and be instructed to check, after a specified period of time, whether the transient update have been reversed.

3.3 Scenario Coordination

Scenario coordinators assist people to carry out (integrated) enterprise scenarios spanning multiple organizations and requiring information / actions from different applications. This assistance comes in several forms, specifically through awareness, knowledge, communication media, resource management, and activity coordination.

Awareness

As disturbances occur within an enterprise environment (detected by EAs), scenario coordinators will bring disturbances to the attention of those individuals who, by their position within the organization, share responsibility for resolving the disturbances. *Awareness* is critical if not essential to this activity. Unfortunately, the issue of appropriate mechanisms for awareness has demonstrated itself to be a very difficult and somewhat controversial issue (e.g., [4,19,9]). Nevertheless, the scenario coordinator would be remiss if it did not attempt to address this issue, if only in a simplistic manner.

Initially, we will investigate using mechanisms which are becoming increasingly ubiquitous within organizational settings, such as electronic mail and pagers, to inform individuals that a disturbance has occurred. However, the issue of appropriate and effective awareness does not stop here. Not only must individuals become aware of disturbances, but they also must be provided with the appropriate knowledge and tools which assist them in understanding and assessing both the disturbance itself and the requirements for successful resolution. In addition, as the resolution process progresses, the 'state' of the disturbance should be accessible to participants. As development on the scenario coordinator progresses, investigations into additional awareness facilities will continue.

Knowledge

The scenario coordinator also will attempt to provide *knowledge* to the individuals responsible for resolving disturbances within an enterprise environment. By knowledge, we mean *information (or data) in context*. The source of such information will be from the enterprise systems (in this case, manufacturing systems such as MES and ERP), monitoring agents, process controllers, reference objects, and repositories of organizational data. The important point is that the scenario coordinator not overwhelm individuals with data but provide individuals with access to the 'right' data in context.

Communication Media

Communication is essential for collaboration [7]. Often, disturbances require the attention of several individuals. In order to effectively assist collectives of individuals as they (a)synchronously resolve disturbances, communication mechanisms must be available. The scenario coordinator initially will support asynchronous communication via shared data (e.g., electronic mail). Synchronous communication, however, will not be supported directly in the initial versions of the scenario coordinator. Thus, users either will have to be co-located or they will have to rely on an external communication medium such as the telephone. We will investigate audio/video desktop conferencing applications if direct support for synchronous communication by the scenario coordinator becomes necessary and if funds to support such investigations are available.

Resource Management

A key function of the scenario coordinator is *resource management*, i.e., managing the applications, data, tools, and monitoring and mining agents which form the context that assists individuals as they resolve disturbances. By providing such a service, members of an organization are freed to focus on the disturbance(s) at hand.

Activity Coordination

Often, the resolution of disturbances ultimately requires a *coordination of activities* among several individuals. The ability for a scenario coordinator to enforce a protocol, or ordering, on such activities is fundamental to the design of the agent. Initially, the scenario coordinator will support a partial order of critical decisions, specified at build time and depicted by a simple, appropriate visual notation. In addition, possibilities for action will be associated with each decision. For example, if an avail to promise disturbance is deemed feasible and desired, operations which un-backflush one purchase order, backflush another purchase order, and modify the shipping dates for both orders must be performed (either automatically or as a result of user directives).

3.4 Plug and play

A natural way to implement the CIIMPLEX enterprise integration architecture is to use a registration architecture. A set of monitoring / notification services (e.g. process rates, yields) will be defined by CIIMPLEX. These services, along with protocols of how they should be used is referred to as the *CIIMPLEX reference model* which will be the basis for open application plug-and-play. Agents exist to provide monitoring / notification for these services. These services will be managed by broker agents.

At integration time, the focus is on which of these services will be utilized and how the limits will be set. Utilizing a service means registering the recipient of notifications with the broker.

4. Scenario coordination vs. workflow management

The concept of workflow management has received a lot of attention as a way to model work in business enterprises. Over the past several years an abundance of new applications have emerged which claim to support workflow. Also, the Workflow Management Coalition (WfMC) was established [23]. Membership in WfMC now numbers in the hundreds.

Unfortunately, the success of workflow management is somewhat unclear at this time due, in part, to a rapidly developing and evolving understanding of the appropriate role of workflow management within enterprise settings. There is a large body of sociological studies [18,20,21] that point to the multidimensional nature of work. For example, work tend to be situation / context based. Current workflow methodologies model (e.g. [8,14]) work in a simplistic way that does not take into account the full complex nature of work. For example a popular class of workflow products (e.g. FlowMarks and KI-Shell) focuses on the coordination of user actions. It tends to de-contextualize work, i.e., viewing work as process instead of action in context, and therefore not respecting the situated nature of work [6,7,12,22].

We propose the use of the concept of *enterprise scenario* to model work in business enterprises. A enterprise scenario models work by first specifying the context for carrying the enterprise scenario. Such a context include the information, tools, people and actions necessary to complete the scenario. If orderings among actions are necessary, they also may be specified; however, *there are not required*. In other words, actions are defined in context and only incorporated into process when appropriate. In contrast, the workflow models work exclusively based on the ordering of activities.

There are several advantages to the context-based approach. First, workflows are simplified because users are not forced to formalize informal activities as well as activities which are unordered. For example, consider an activity supported by five actions a, b, c, d and e where a must occur before b and b must occur before c; while d and e can occur at any time. Many traditional workflows would depict this activity as shown in Figure 3. A context-based workflow, however, has less complexity. All five actions are support but only actions a, b and c must be depicted in the workflow (see Figure 4). The result is flexible, light-weight coordination support as part of context (i.e., orderings among actions augmenting context to help facilitate work).

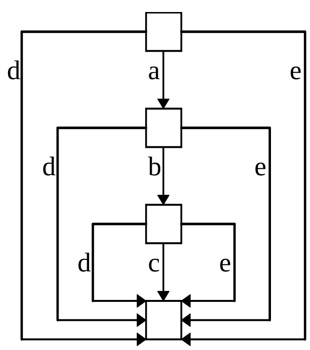

Figure 3: Traditional Workflow Complexity

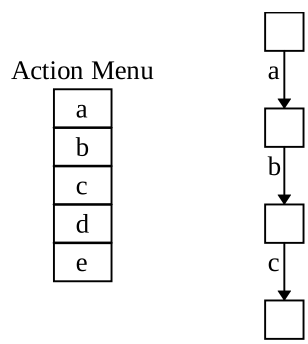

Figure4: Context-based Workflow Complexity

Second, although information is critical, it is often useless out of context. A context-based approach, however, facilitates what we call *knowledge*-based workflows by supporting *information in context*. Third, this approach respects the essential nature of communication to the effectiveness of workflow applications by incorporating communication mechanisms as part of context rather than requiring they be formalized within process. Finally, a context-based workflows can promote better awareness and understanding of the activities at hand as opposed to obtrusive structure and control. That is not to say that structure and control are not important to this approach of workflow management but that their presence is less obtrusive to system users because they appear more natural, and thus appropriate, when viewed in context.

A final comment about workflows management relates to an aspect that the workflow community well acknowledges. That aspect is the importance of flexibility in workflow support. Users must be able to dynamically change workflows as contingencies arise and work practices evolve. A context-based approach to workflow management must respect this aspect as well.

4.1 Scenario composer

CIIMPLEX will provide a tool, referred to as the *scenario composer*, for people to specify enterprise scenario coordinators to assist enterprise scenarios. We envision the user of such a tool to be a “power” end user with domain knowledge. The role of scenario coordinators within CIIMPLEX is to assist an organization as it resolves disturbances within an enterprise environment. Such assistance includes facilitating better awareness while providing knowledge (i.e., information in context), communication media, resource management and activity coordination. Several components of scenario coordinators must be specified at build time to provide such support. These components include (but are not limited to):

- A collection of monitoring agents which, in addition to detecting problems, provide feedback information which further characterize the problems.
- A collection of data mining and parameter estimation agents which provide composite data (i.e., missing data, filtered data, higher-level representations of raw data).
- Organizational data such as the people who should participate in resolving the scenario. These data may be specified in terms of organizational roles (e.g., process control rep. or production control mgr.) or specific individuals (e.g., Mary Jones or Joe Smith). In either case, more than just a name will be specified. Instead, the 'business card' (i.e., name, phone number, fax number, e-mail address, cell-phone number, etc.) for the individual or role will be specified from an organizational database.
- Information / data about the scenario or which are required to resolve the scenario.
- A collection of issues or questions to be addressed which help resolve the scenario. The ability to place constraints among these issues also will be supported.
- A collection of actions, as well as their organizational context (e.g. ERP vs. MES), that initiate business integration processes using FAIME process controllers. Examples of such actions include 'generate bill of materials' and 'un-backflush purchase order.' Such actions may be invoked directly by users, indirectly as a consequence of an answer to a question, or automatically due to the status of the scenario or events within the enterprise environment.
- Actions mentioned above could be subject to sequence constraints as well as organizational constraints.

The scenario composer will be supported by a hybrid of graphical and textual interfaces. Graphical interfaces will support activities such as specifying constraints among scenario issues. In addition, manufacturing systems and agents which have iconic representations may be manipulated via drag-n-drop operations. The scenario composer also will support a collection of browsers which allow users to browse organizational, system and agent repositories. Thus, several aspects of scenario coordinators will be specified through point-n-click operations. Finally, form or text-based interfaces will be used to support activities such as specifying the collection of scenario issues, providing documentation to the scenario and labeling scenario components (e.g., actions).

Scenario coordinators initially will be supported by a World Wide Web-based interface. Thus, the build time environment for a scenario coordinator will produce an HTML document which is supported by a collection of JAVA applets and applications.

Still, there are many open issues regarding the scenario composer. A large number of these issues exist because the answer to the question of what is to be specified, is still evolving. For example, what is the relationship, if any, between the build time environment and the run time environment? Work is highly contingent [22]. As such, systems designed to support the resolution of disturbances within an enterprise environment must respect this inherent nature of work by allowing the support which they provide to change as both expected and unexpected contingencies arise and as work practices evolve.

5. An example

We have constructed a prototype to demonstrate the mining and plug-and-play aspects of the CIIMPLEX enterprise integration architecture. Components of this prototype are illustrated in Figure 3.

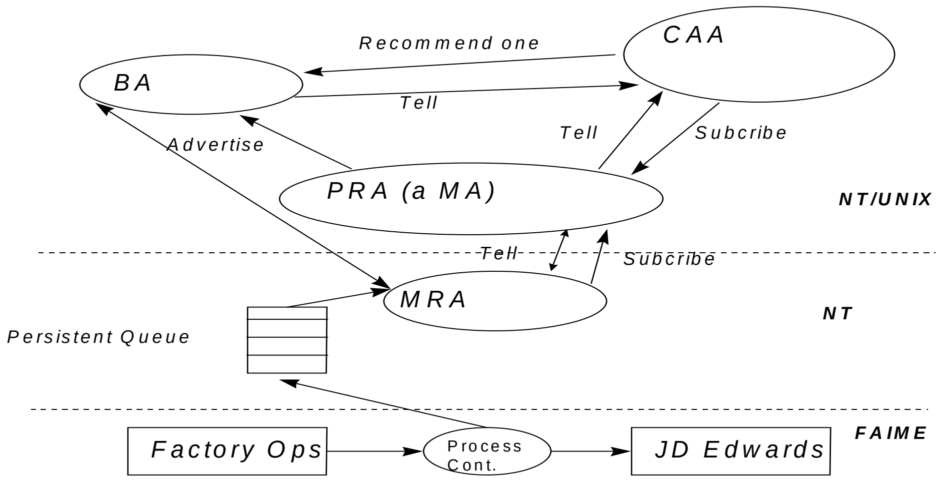

Figure 2. CIIMPLEX agent prototype.

This prototype agent system consists of five agents, they are *ANS* (not shown in the figure), the agent name server; *BA*, the broker agent; *CAA*, the CIIMPLEX analysis agent; *PRA*, the process rate mediation agent; and *MRA*, the message routing agent. KQML, the Knowledge Query Manipulation Language, is chosen as the agent communication language and protocol[5]. KQML, based on the theory of speech act, provides a set of performatives such as *ask*, *tell*, *recommend*, *subscribe*, to convey the intentions of agents when they communicate with each other. A set of conversation policies are designed to ensure that proper operational semantics is followed. KIF, the knowledge Interface Format, is chosen as the content language for KQML messages[5]. As a first-order predicate logic based language, KIF has sufficient expressive power to accurately represent queries and responses for inter-agent communications.

The purpose of this agent system is to aggregate information about operation completion's, one of the low level shop floor activities important for high level integration tasks. The agent system is built on top of the FAIME technology which provides a stream of messages of operation completion's. A persistent queue, realized by MQ Series, is used to store all such messages.

When MES triggers the completion of a WIP, it kicks off a FAIME-style process controller. In addition to performing the FAIME-STYLE transaction (updating ERP), the process controller will send a message to agent PRA. The responsibility of PRA is to aggregate process rates for different part numbers at different operations based on FAIME messages routed to it by MRA, and to make the aggregated rates available to other agents. The PRA is also responsible for accepting monitor instructions from other agents. For example, CAA may instruct PRA that if the process rate for part number *xyz* at operation *abc* changes by 20% from the current value, then CAA need to be notified. PRA is thus an example of both data mining and event monitoring agent mentioned in the previous section. CAA and other agents can dynamically change the monitoring specification of PRA.

The Broker agent (BA) is responsible for tracking what services are available and for linking available services to outstanding requests. For example, PRA may advertise on BA that it has available the aggregated process rates for certain operations. Then CAA or other agents who need this information (e.g., a PRM agent) can ask BA to recommend or recruit agents that possess such rates for them.

The collection of agents like PRA and CAA works differently from traditional Statistical Process Control (SPC) in several important ways:

- SPC requires manual set up, i.e., defining the monitoring parameters and setting the limits, whereas PRA is designed to be set up by CAA and other agents dynamically through run time messages.
- As an intelligent agent, PRA can perform tasks involving process rate that are not foreseen at the build time but may arise at the run time. For example, a scheduling algorithm may want to monitor ratios of process rates.

When fully developed, agents like PRA will have more sophisticated statistical knowledge, such as the ability to support simulation, and to provide certain statistical forecasting and estimation

6. Implementation strategy

We plan to build our agents in JAVA for maximal portability. JAVA also makes it easy for agents exchange executable code segments.

We plan to use JAVA enabled browsers as the desktop for end users. Scenario coordination agents will be implemented as JAVA applets.

The ability to specify rules declaratively is very useful for system understandability and maintainability, particularly in the area of specifying constraints for scenario coordination. We plan to use adapt the rule-based interoperation technology developed by the NIIIP consortium [10] for use in the CIIMPLEX implementation.

7. Comparison with related work

A number of major research initiatives have resulted several large-scale agent-based architectures for advanced information systems. They include: AARIA [17], KAoS [1], INFORSPHERES [3], WORLDS [6,7,12,22], and KNOWLEDGE SHARING [5].

The CIIMPLEX enterprise integration architecture shares with most of they in offering a vision of distributed agents. As is in CIIMPLEX, most of these architectures employ agents to perform information filtering, monitoring, and brokering.

In several respect AARIA has similar objectives as CIIMPLEX: integration between ERP and MES, better manage customer commitment dates, fast and flexible response to disturbances. However, AARIA develops a pure agent-based approach. Most the functions are provided by agents, i.e., new applications.

Both INFOSPHERES and WORLDS address the need for computer assistance to human problem solving in a complex, information-rich environments. An infosphere/locale (in WORLDS) is characterized by:

- the *primary work activity/activities* for which the setting is constructed or for which it is being used;
- the *particulars* of the setting (i.e., the artifacts, data, tools, actions, etc. that tailor a locale to its use or purpose);
- the *people* who participate in, and interact with, the setting;

and, the *processes* which exist or arise within and among settings. WORLDS is also based on the research in the field of Computer-Assisted Collaborative Work (CSCW).

8. Acknowledgments

This work is supported in part by the Advanced Technology Program administered by the National Institute of Standard and Technology under the agreement number: 70NANB6H2000

9. Reference

1. Bradshaw, J., Dutfield, S., Benoit, P. & Woolley, J. "KAoS: Toward An Industrial-Strength Open Agent Architecture" to appear in *Softwware Agents* Bradshaw, J.M. (Ed), MIT Press
2. Chu, B. Matthews, M. Long, J. Barnes, J. Sims, J. Hamilton, M., and Lambert, R. "The Essence and Accidents of Application Interoperability", submitted for publication.
3. Compositional Research Group "Caltech Infosheres Project" available at http://www.infospheres.caltech.edu.
4. Dourish, P. Bellotti, V.. "Awareness and Coordination in Shared Workspaces". *In Proceedings ACM 1992 Conference on Computer-Supported Cooperative Work: Sharing Perspectives (CSCW '92*), pp107-114, Toronto, November 1992.
5. Finin, T., Labrou, Y., & Mayfield, J. "KQML as an agent communication language". In *Software Agents.* Bradshaw, J.M. (Ed.) MIT Press, to appear.
6. Fitzpatrick, G., Mansfield, T., & Kaplan, S. "Locales Framework: Exploring Foundations for Collaboration Support". In *IEEE Proceedings of OzCHI '96*. To appear.
7. Fitzpatrick, G., Tolone, W.J., & Kaplan, S. "Work, Locales and Distributed Social Worlds". In *Proceedings of the 1995 European Conference on Computer-Supported Cooperative Work (ECSCW '95*), pp 1-16, Stockholm, September 1995.
8. Flores, F., Graves,M., Hartfield, B., & Winograd, T. "Computer Systems and the design of organizational interaction". In Ronald M. Baecker, editor, *Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*, pp. 504-513. Morgan Kaufmann Publisher, Inc., 1993.
9. Fuchs, L., Pankoke-Babatz, U., & Prinz, W. "Supporting Cooperative Awareness with Local Event Mechanisms: The GroupDesk System". *In Proceedings of the 1995 European Conference on Computer-Supported Cooperative Work (ECSCW '95*), pp. 247-263, Stockholm, September 1995.
10. Goldschmidt, A. *Report on NIIIP* in *CACM* 39-3, pp. 100-103, 1996
11. Jacobson, I. *Object-Oriented Software Engineering,* Addison-Wesley, Reading, MA. 1992.
12. Kaplan, S., Fitzpatrick, G. Mansfield, T. & Tolone, W.J.. "Shooting into Orbit". In *Proceedings Inaugural Australian National Symposium on Computer-Supported Cooperative Work, DSTCPty. Ltd*., pp. 38-48, Brisbane, August 1996.
13. Mark, L., Roussopoulos, N. & Chu, B. "Update Dependendencies" *Database Semantics,* Steel, T. B. and Meersman, R (Eds). North-Holland, The Netherlands, 1986, pp. 303-319.
14. Medina-Mora,R., Winograd, T., Flores, R., Flores, F. "The action workflow approach to workflow management technology". In *ACM 1992 Conference on Computer-Supported Cooperative Work: Sharing Perspectives*, pp. 281-288, Toronto, November 1992.
15. OAG: Open Application Group. Available through http://www.oag.org.
16. OMG: Object Management Group. Available through http://www.omg.org.
17. Parunak, V., Baker, A., & Clark, S. "AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design" available at http://www.aaria.uc.edu
18. Robinson, M. "Double-level languages and co-operative working". *AI and Society*, 5(1):34-60, 1991.
19. Sohlenkamp, M., "Integrating Communication, Cooperation and Awareness: The DIVA Virtual Office Environment." In *Proceedings ACM 1994 Conference on computer-Supported Cooperative Work: Transcending Boundaries (CSCW '94),* pp 331-343, Chapel Hill, NC, October 1994.
20. Strauss, A. *Continual Permutations of Action*. Aldine de Gruyter.New York, 1993.
21. Suchman, L. *Plans and Situated Actions*. Cambridge University Press,1987
22. Tolone, W.J. *INTROSPECT: a Meta-Level Specification Framework for Dynamic, Evolvable Collaboration Support* . Ph.D. Thesis. University of Illinois at Urbana-Champaign, 1996.
23. WfMC http://www.aiai.ed.ac.uk/WfMC/ Membership now several hundred
24. Wiederhold, G. "Forward: Intelligent Integration of Information" *Journal of Intelligent Information Systems* 5. pp.1-5, 1996.
25. Windom, J. & Ceri, S. *Active Database Systems Morgan Kaufmann,* San Francisco, CA. 1996..