Supporting Human Interactions within Integrated Manufacturing Systems

Short Title: Supporting Human Interactions within Integrated Systems

William J. Tolone[1], Bei-tseng Chu[1], Junsheng Long[1],
Robert Wilhelm[2], Tim Finin[3], Yun Peng[3]

[1]Department of Computer Science
[2]Department of Mechanical Engineering
University of North Carolina at Charlotte
Charlotte, NC 28223
{wjtolone, billchu, jlong, rgwilhel}@uncc.edu

[3]Department of Computer Science
University of Maryland at Baltimore County
Baltimore, MD 21228
{finin, ypeng}@cs.umbc.edu

Abstract

Application integration is a focus of many major research initiatives. A driving force behind application integration is a desire for enterprise-wide integration of organizational business processes. On the surface, increased application integration appears advantageous – people are freed from mundane tasks and thus can focus on more serious issues. Yet, there are consequences that are not always recognized or appreciated. Specifically, the relevance of human participation within integrated systems often is underdeveloped, thus increasing the potential for integrated systems that dismiss rather than empower users. In this paper, we present the work of one research initiative (CIIMPLEX) that is examining the relevance of human collaborations as part of its manufacturing integration research efforts. The emphasis of the paper is on the principles of the proposed approach rather than details of the current implementation.

Keywords

Computer-Supported Cooperative Work, Software Agents, Agile Manufacturing, Human-Centered Systems, System Integration, Business Process Integration

Introduction

Application interoperation and plug-and-play are a focus of many major research initiatives, e.g., FAIME [1], NIIIP [2], OAG [3], and OMG [4]. Several factors contribute the recent emphasis on integrated solutions. One common objective of these initiatives is to provide enterprise solutions consisting of multiple COTS (commercial off the shelf) applications where the migration of tasks and the synchronization of data occur automatically across multiple applications. A driving force behind this recent emphasis on application integration is a desire for enterprise-wide integration of organizational business processes.

On the surface, increased application integration appears advantageous – people are freed from mundane tasks and thus can focus on more serious issues. Yet, there are consequences to application integration that are not always recognized or appreciated. First, people can lose 'context' when trying to manage exceptions within an integrated environment. Thus, when users perform operations within an integrated environment, the effects of those operations are not always well understood since they now can have immediate and direct impact outside the scope of a single application. Second, even when 'context' is maintained, its complexity has increased. Thus, although integration relieves people from performing certain tasks, it simultaneously introduces greater system complexity.

Why are these consequences of application integration commonly overlooked? Charles Billings discusses several myths of technology that help answer this question [5]. Two myths that highlight this problem are discussed below.

- **Myth**: *Technology can help supplant the unreliable human* ([5] p. 53). Often, application integration is viewed not only as a means of business process integration, but as a means of business process automation. That is, through application integration, the relatively slow, unreliable human can be supplanted by relatively fast, reliable technologies resulting in more reliable, streamlined execution of integrated business processes. However, there are at least two problems with such an assumption. First, and more fundamentally, technology does not remove human error, but merely reshapes its nature. Second, business process integration requires more than just a technical solution. This point is discussed in greater detail in section, "Integrated Systems and Human Collaborations."

- **Myth**: *Humans and machines are equivalent* ([5] p. 57). That is, if business processes are fragmented into small enough tasks, machines can replace humans in performing many of the component tasks. Here we see the tendency of application integrators to discretize processes into well-defined tasks on the assumption that such tasks require very limited situational knowledge and are well suited for automation. Once again, there are at least two problems with such an assumption. First, as Billings states, people and technology are complementary rather than comparable. Second, tasks, particularly those commonly performed by people, often require situational knowledge that extends beyond the immediate context of the task at hand. Yet, an application integration model to business process integration assumes an immediate knowledge requirement. That is, it assumes an inputs-to-outputs approach to task completion (or inbox-to-outbox). While such an approach may be appropriate for automated tasks within integrated business processes, it often is inappropriate for human tasks (as discussed in more detail in section, "Workflow and the Management of Responsibility").

For these reasons and others, the relevance of human participation within integrated systems often is underdeveloped, thus increasing the potential for integrated systems that dismiss rather than empower users.

In this paper, however, we discuss one recent research initiative that is examining the relevance of human collaborations as part of its manufacturing integration research efforts. With the help of NIST/ATP's Technology for Integrated Manufacturing Application (TIMA) initiative, the Consortium for Intelligent Integrated Manufacturing Planning-Execution (CIIMPLEX) [6] was established (1996) to address key aspects of application integration within manufacturing enterprises. At the midpoint of this three-year project, the relevance of human collaborations in relationship to the consortium's Intelligent Integrated Planning-Execution (IIPE) methodology has emerged as a central research thread for the project.

In the following, we examine the consortium's work addressing this requirement for collaboration support for the individuals interacting, managing, and resolving disturbances within an integrated manufacturing environment. The structure of the paper is as follows. First, we begin with an overview of the CIIMPLEX integration architecture and supporting technologies. Second, we discuss the appropriateness of software agents as an enabling technology within integrated enterprise environments. Third, we examine the nature of human interactions within integrated manufacturing systems and hilight the approach that CIIMPLEX is investigating. Fourth, we present a brief example that demonstrates how this approach is realized through the use of software agents within the CIIMPELX Integration Architecture. Unfortunately, confidentiality agreements currently prohibit a detailed discussion of the implementation of this approach. The paper concludes with a short discussion of related work.

CIIMPLEX

With the help of NIST/ATP's Technology for Integrated Manufacturing Application (TIMA) initiative, the Consortium for Intelligent Integrated Manufacturing Planning-Execution (CIIMPLEX) was established (1996) to address key aspects of application integration within manufacturing enterprises. Several important goals were established for the consortium [7].

- *To enable manufacturers to make plans based on real-time capacity information.* Data used as a basis for generated plant schedules commonly is out of date. That is, events within a Manufacturing Execution System (MES)[1] are not reflected within an Enterprise Resource Planning (ERP)[2] system and vice-versa, in a timely fashion. Consequently, production schedules often are short-lived, sometimes are dead on plant floor arrival, due to regular changes and/or exceptions captured within MESs.

- *To monitor manufacturing-plan execution using real-time data and provide intelligent assistance in decision making.* CIIMPLEX is investigating the use of software agents as a mechanism for 'filling in the white space' among enterprise applications. When agents are incapable of resolving 'the white space,' people are introduced into the situation. Examples of this are varied. A common situation might be assigning a software agent to monitor process rate for a specific operation and on a particular machine. When this rate deviates by some delta from the expected rate, people are informed and the enterprise's finite scheduler potentially is updated.

- *To develop an open architecture to enable the above functions to be delivered through the plug-and-play of COTS (commercial off the self) applications*. The rapid changes to computer hardware and applications coupled with the significant investment made by enterprises in such technology, necessitates a respect for legacy systems. Often, by the time technology arrives at a company site, it is 'out of date.' After making large investments, companies are reluctant to make wholesale changes – yet they still want to leverage new technologies. More accurately, market competition drives companies to leverage new technologies.

CIIMPEX is realizing these goals through the CIIMPLEX Integration Architecture that utilizes the latest technology of software agents (see Figure 1) and the CIIMPLEX methodology for Intelligent Integrated Planning-Execution (IIPE).

---

[1] A MES is responsible for managing plant floor execution based on a schedule. This includes such tasks as tracking machine operations as well as material location and consumption.
[2] An ERP based purchase orders and inventory produces long-range plant schedules.

Within the integration architecture, software agents are used as a mechanism for 'filling in the white space' among enterprise applications. Examples, of such agents include:

- Data mining agents – These are agents that monitor for specific application events or that gather raw data and produce more meaningful (usually aggregated) information for higher level analysis.

- Analysis agents – These are agents highly tailored for the manufacturing domain that automatically monitor integrated system performance and perform corrective actions.

- Collaboration agents – These agents facilitate disturbance resolution within the integrated environment by support end-user collaborations.

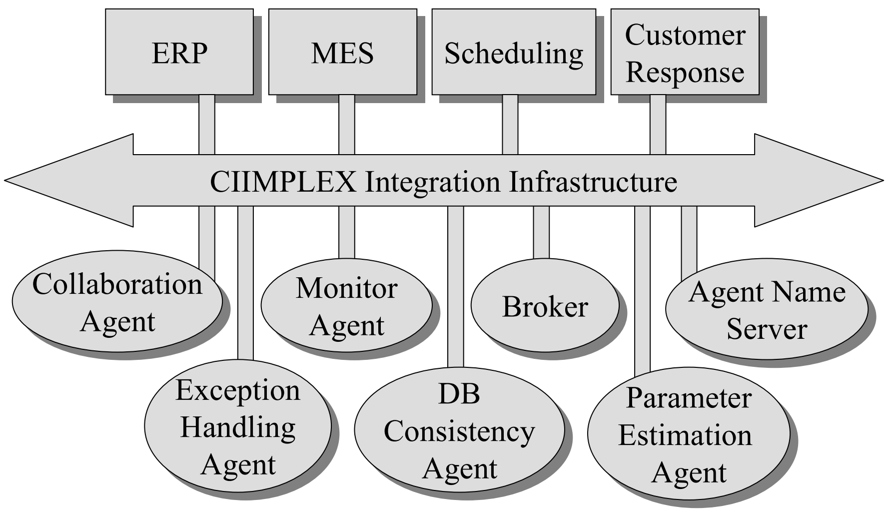

Figure 1: CIIMPLEX Integration Architecture

Thus far, the consortium has successfully demonstrated manufacturing application integration and the use of agents to 'fill in the white space' among applications to detect and resolve disturbances for specific customer scenarios. A consequence of this work has been increased automation within the manufacturing environment. Redundant activities are eliminated and replaced by a single activity that is distributed automatically across the integrated enterprise. For example, instead of independently adding new purchase orders to multiple manufacturing applications (this is today's common practice), new purchase orders are added to a single application and reflected by the integration architecture to all other interested applications. When situations arise where agents are incapable of resolving 'the white space,' people are enlisted to bring such situations to appropriate resolutions. This later aspect, the places where people and integrated systems meet, is the primary focus of this paper.

Why Software Agents?

The term *software agent*, though commonplace in today's technical literature, is difficult to define universally. To some, simple scripting programs or the practice of tool enveloping constitute software agents. To others, software agents are much more autonomous, complex, comprehensive, and knowledge-enriched.

> *An ideal agent knows what its goal is and will strive to achieve it. An agent should be robust and adaptive, capable of learning from experience and responding to unforeseen situations with a repertoire of different methods. Finally, it should be autonomous so that it can sense the current state of its environment and act independently to make progress towards its goals* [8].

As a programming methodology, the software agent approach is a next step in the evolution of programming methodologies. One can trace an evolution from machine and assembly languages, to higher-level structured languages, to groups of subroutines contained in modules or libraries, to object-oriented languages, to distributed-object frameworks, to the current trend of distributed-agent frameworks [9]. Most distributed-agent frameworks consist of communities of agents that cooperatively offer and fulfill service/resource requests to accomplish desired tasks or attain desired goals. Furthermore, inherent to such frameworks is support for both evolution and agility that respects rapidly changing application requirements so prevalent among current Information Technology (IT) support.

In CIIMPLEX, software agents offer an attractive, enabling technology that can help facilitate the integration of heterogeneous software systems by 'filling in the white-space' among applications to support flexible, agile manufacturing enterprises.

CIIMPLEX software agents are *theoretically* characterized by the following properties:

1. ***Persistent****: An agent maintains its own internal state.*
2. ***Autonomous****: An agent has full control over its internal state and behavior.*
3. ***Reactive****: An agent detects and reacts to changes in its environment.*
4. ***Proactive****: An agent is goal-directed; that is, it has its own goals and actively works on accomplishing those goals.*
5. ***Sociable****: An agent is able to communicate with other agents or entities and exchange information.*
6. ***Mobile****: An agent is capable of migrating to different location while preserving its own internal state.*
7. ***Capable of reasoning****: An agent is able to infer and deduce new knowledge, given its current knowledge and experiences.*
8. ***Able to plan****: An agent is able to produce alternative courses of action and choose among actions to achieve its goals.*
9. ***Able to learn/adapt****: An agent processes experiences to accumulate knowledge that may alter its behavior* [9].

In current practice, however, CIIMPLEX software agents are better characterized as autonomous, goal-directed processes that are situated, aware of, and reacting within integrated manufacturing environments while cooperating with other agents (both software and human) to accomplish manufacturing IT practices.

Integrated Systems and Human Collaborations

In addressing the relevance of human collaborations within integrated manufacturing systems, CIIMPLEX has given particular attention to an emerging group of applications termed *workflow* applications.[3] Within the consortium's Intelligent Integrated Planning-Execution (IIPE) methodology several design problems have emerged as candidates for a workflow solution. The foremost candidate for workflow support is the automation of application collaborations within the CIIMPLEX integrated manufacturing environment. Such support facilitates the automatic routing of information among applications while respecting the data and business models of each application. CIIMPLEX is investigating the use of the FAIME methodology and tools [1] in combination with commercial workflow applications to provide this support. Such workflow technology is referred to as *production* workflow support.

Another situation that has presented itself as a candidate for a workflow solution is supporting human interaction within the manufacturing disturbance resolution process. In other words, as the CIIMPLEX architecture responds to disturbances within the integrated environment, how is human cooperation, either routine or as a consequence of exception, supported? Initially, this problem appeared to be a strong candidate for a workflow solution, i.e., model human cooperation as a simple workflows and support their execution. However, further examination indicates such a solution is problematic.

Workflow and the Management of Responsibility

The CIIMPLEX architecture has proposed the use of cooperating agents as a form of exception management within integrated manufacturing environments. Whenever human intervention, either individual or collaborative, is required, a human agent could model such intervention. To a certain extent, this is an appropriate characterization and under this characterization, a workflow solution seems adequate. That is, human agents are another type of application that have input and output data requirements. Yet, the problem is not so simple. Disturbance resolution requires human intervention often because coded agents are inadequate. Thus, we believe representing the complexities of human intervention as just another agent is equally inadequate. We offer two significant points as evidence.

- First, a close examination of the workflow industry demonstrates successful products within the *production* workflow domain, but very few successful products within the *knowledge* workflow domain. By knowledge workflow we mean applications that model the complexities of knowledge workers as they resolve disturbances within an organization - disturbances that are highly contingent, not unlike those being addressed by CIIMPLEX.

- Second, the Computer-Supported Cooperative Work (CSCW) research community has offered increasing insight into how people work through and with technology. One very important observation is that work is highly contingent and highly situated [10, 11]. While using workflow to support the *procedures* of application interoperation may be appropriate, using workflow to support human *practices* that surround such procedures, with their inherent complexities and dynamism, is most often inappropriate.

Fundamentally, we believe such evidence indicates inadequate support within the workflow model for *responsibility management over integrated business processes*. The workflow model for application integration tends to discretize activities into well-defined tasks and support only immediate situational knowledge for such tasks. That is, the workflow model assumes an inputs-to-outputs approach to task completion. Within production workflow applications where the emphasis is on automated transactions across multiple applications, limited situational knowledge often is sufficient to complete integrated business processes – assuming exceptional situations do

[3] See the Workflow Management Coalition at www.aiai.ed.ac.uk/WfMC/

not occur! However, when exceptional situations occur, responsibility for managing such situations is not directly supported by the workflow model.

The inadequacies are magnified when considering integrated business processes that require human intervention and possible collaborations from the outset. In these situations, while the workflow model may support responsibility management at the level of business *task*, limited situational knowledge may be insufficient to complete such tasks. Thus, an inputs-to-outputs approach (or inbox-to-outbox approach) is an insufficient model for task completion not only when exceptional situations arise, but many times when they do not.

Michael Hammer offers an excellent example that demonstrates the importance of responsibility management across integrated business processes. In his book *Beyond Reengineering* [12], Hammer discusses the reengineering of GTE's process for supporting customer reports of outages. Initially, GTE supported this process by having a different 'specialist' complete each significant task that comprised the process. However, after 'reengineering' this process, GTE made one individual responsible for the entire process. With multiple people responsible, each at the task level, the process was inefficient. Yet, when one person assumed responsibility at the process level, even when other people completed the individual tasks, the effectiveness of this process improved significantly. The point is that application integration as a means of business process integration requires more than just a technical solution. Furthermore, any technology that attempts to contribute to an overall solution must respect the need for situational knowledge that extends beyond the immediate context of any one task. As Hammer put it, "Even when one person cannot perform an entire process, it is still possible to have every person who is involved in the process to understand it in its entirety and focus on its outcome." ([12] pp. 35-36). Unfortunately, the workflow model focuses on responsibility and knowledge at the tack level instead of the process level.

Where does that leave support for human interaction and collaboration among integrated applications? Hammer offers potential insight into this question when he goes on to say, "When people appreciate the larger context of their work they do not work at cross purposes with others engaged in the same process. When everyone has a common measure there is no need for reconciling inconsistent activities." ([12] p. 36) In other words, *support for human intervention and collaboration support must be context-based not workflow-based.*

There is a large body of social research addressing the very nature of 'work' (e.g., [11, 13]) that supports the context-based approach CIIMPLEX is investigating; however, a detailed discussion of these is outside the scope of this paper. Instead, we focus on principles of a *context-based* approach to collaboration support.

Context-based v. Workflow-based Collaboration Support

CIIMPLEX has proposed the development of a Collaboration Agent to provide context-based collaboration support to integrated manufacturing systems. Although context-based collaboration support and workflow-based collaboration support share similar goals, namely to facilitate the execution of business processes by multiple people, fundamental differences exist between each approach. The following table describes the principles of a context-based approach to collaboration support and contrasts these principles with those of a workflow-based approach.

<table>
<tr><th></th><th>CONTEXT-BASED COLLABORATION</th><th>WORKFLOW-BASED COLLABORATION</th></tr>
<tr><td>Supports business process execution</td><td>Yes</td><td>Yes</td></tr>
<tr><td>Enforces sequences of activities</td><td>Yes</td><td>Yes</td></tr>
<tr><td>View of business process</td><td>Holistic view of business process supporting rich situational knowledge in addition to sequencing of steps.

Rich situational knowledge includes:
• Goals and objectives
• Communication mechanisms for people: e.g. white-board session, video conferencing, and active messages.
• Enterprise impact analysis for courses of action, i.e., what if analyses.
• Timeliness of actions
• A multidimensional view of business process:
- Synchronous v. asynchronous.
- Data-centered v. application-centered v. communication-centered v. action-centered.
- Business process granularity.
• Both process and people's roles evolve in a dynamic environment (i.e., business processes move along the above dimensions over their lifetimes)
• Situational knowledge is dynamic & evolutionary
• Sequences of steps are rule-based and thus less prescriptive and more flexible</td><td>Reductionist view of business process, i.e., sequences of steps having well-defined inputs and outputs.

Each player is asked to do his/her well-defined tasks supported by limited situational knowledge

Both process and people's roles are stable.

Prescriptive, though potentially modifiable.</td></tr>
<tr><td>Representation mechanisms</td><td>State machines, business rules, natural language.

Specification of what, where, & when.</td><td>Flow charts, entity-relationship models.

Specification of how.</td></tr>
<tr><td>Targets</td><td>Knowledge workers, semi-structured work.

Business users.</td><td>Backroom production.

IT analysts.</td></tr>
</table>

Table 1: Principles of Context-based and Workflow-based Collaboration Support

Thus, CIIMPLEX has studied various approaches to collaboration support, but has settled on context-based approach to support human interaction and collaboration within integrated manufacturing systems. To realize this approach, a Collaboration Agent is under development that allows users to create a richer notion of context than that afforded by workflow applications. To construct such context, users specify the data, tools, people and actions required to complete integrated business processes. If orderings among actions are necessary, rule-based support for defining such orderings is provided (i.e., actions are defined in context and ordered only when necessary). Furthermore, integrated with this notion of context are various mechanisms for communication.[4] In contrast, workflow applications require that communication patterns be formalized within workflow specifications, less they go unsupported. Unfortunately, a detailed architectural discussion of the Collaboration Agent currently is prohibited due to confidentiality agreements.

Finally, the benefits of *production* workflow systems discussed above is leveraged by this approach through a proper integration of the Collaboration Agent with CIIMPLEX's production workflow support. Ultimately, we believe that context-based collaboration support can promote better awareness, better management of responsibility, improved understanding, and subsequently, more efficient execution of integrated business processes.

Example

In this section, we use an example to illustrate the use of software agents and specifically the Collaboration Agent within the CIIMPLEX manufacturing integration architecture. Components of this example are illustrated in Figure 2. The purpose of this example is to demonstrate a situation resulting from application integration (and subsequent business process integration) that can benefit from the application of software agents, and in particular, context-based collaboration support.

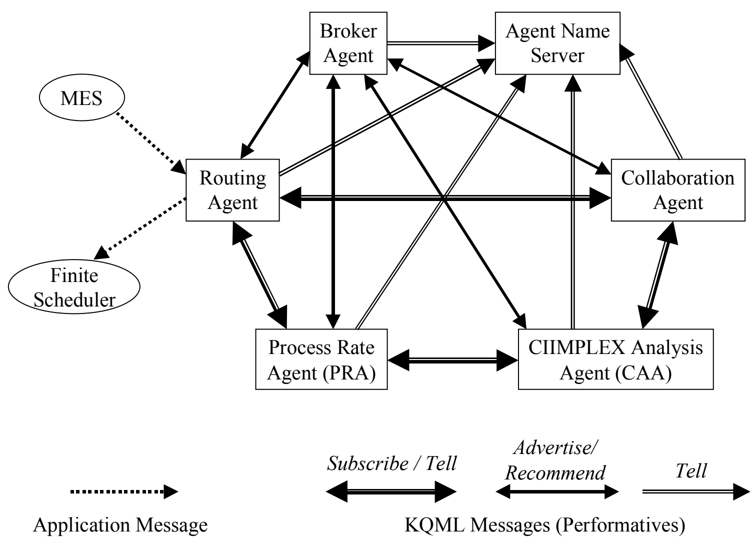

Figure 2: Example Agent Scenario

In this example, a monitoring agent can detect if there is a significant change of production rate at a particular work center. When a rate change is detected, a CIIMPLEX analysis agent can analyze the new rate and determine if this rate change will affect promised customer delivery dates. If customer delivery dates will be delayed, the analysis agent notifies the Collaboration Agent, which initiates a collaboration scenario to facilitate the resolution of this situation.

The functions of the Collaboration Agent are to:

[4] This definition of context is not original to CIIMPLEX but is based on past work of the lead author and his colleagues [14, 15, 16].

- inform the necessary people about the rate change situation;
- generate new schedules;
- assist people in the global impact assessment of the new schedules;
- facilitate decision making by supporting what-if analyses while varying the priority of orders;
- ensure proper approvals for different courses of action;
- notify relevant parties of delivery date changes that are a consequence of the selected courses of action;
- provide appropriate mechanisms for communication among collaboration participants (e.g., e-mail, pagers, cellular phones, active messages on Java-enabled cellular phones, textual 'chat' rooms, bulletin boards, audio/video web-conferencing, etc.);
- function as a launching point the selected courses of action (e.g., initiating appropriate application transactions);
- and, maintain a history log of collaboration activities.

In other words, the Collaboration Agent is designed to provide 'contextual' support to people by facilitating access to relevant data, tools, applications and people and enabling the management of responsibility across integrated business processes.

Supporting Architecture

This example demonstrates several other software agents and supporting technologies that assist the Collaboration Agent in its function. The supporting agents include: a *broker agent* that matches agents needs with available services; a *CIIMPLEX analysis agent* (CAA) that performs various forms of sensitivity analysis; a process rate mining agent (PRA) that estimates process rates and monitors rate changes; an *agent name server* that provides a naming service enabling software agent to explicitly name one another in their collaborations; and, a *message routing agent* that translates data formats between different systems.

The language or protocol for agent communication is KQML, the Knowledge Query Manipulation Language [17]. KQML, based on Speech Act Theory [18], provides a set of performatives such as *ask, tell, recommend,* and *subscribe*, to convey agent intentions during agent conversations. A set of conversation policies is designed to ensure that proper operational semantics are followed.

Agent Collaborations

When Manufacturing Execution System (MES) triggers the completion of units of work, the process rate agent (PRA) updates process rates for different part numbers at different operations. The PRA also is responsible for accepting monitor instructions or requests from other agents. For example, the CAA may instruct the PRA that if the process rate for part number xyz at operation abc changes by 20 percent from the current value, then the CAA should be notified. Thus, the PRA is an example of both data mining and event monitoring agent. Furthermore, the CAA and other agents can dynamically change the monitoring specification of the PRA.

The Broker agent (BA) is responsible for tracking what services are available and for linking available services to outstanding requests. For example, the PRA may advertise to the BA that it has available the aggregated process rates for certain operations. The CAA or other agents who need this information can ask the BA to recommend or recruit agents that possess such rates for them. The subscription / notification architecture allows agent analysis functions to be added incrementally.

Agents similar to the PRA and CAA work 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 the PRA is designed to be set up by the CAA and other agents dynamically through run time messages.

- As an intelligent agent, the 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 the PRA will have more sophisticated statistical knowledge such as the ability to support simulation and sensitivity analysis, and to provide certain statistical forecasting and estimation.

Users interact with the Collaboration Agent through Java-enabled browsers such as the Netscape Navigator.

Related Work

A number of major research initiatives have resulted several large-scale agent-based architectures for advanced information systems. Such initiatives include AARIA [19], KAoS [20], INFOSPHERES [21], and WORLDS [14, 15, 22].

The CIIMPLEX manufacturing integration architecture shares a similar vision of distributed, cooperating agents with most of these initiatives. As is in CIIMPLEX, most of these architectures employ agents to perform information filtering, monitoring, and brokering.

In several respects AARIA has similar objectives as CIIMPLEX; that is, integration between ERPs and MESs, better management of customer commitment dates, and, fast and flexible response to disturbances. However, AARIA develops a pure agent-based approach. Agents provide most manufacturing functions, whereas CIIMPLEX achieves its objectives leveraging functions supplied by commercial applications.

INFOSPHERES, KAoS, and WORLDS address the need for computer assistance to human problem solving in a complex, information-rich environment. In WORLDS, such assistance is provided via locales.[5] A locale is characterized by:

- the *primary work activity/activities* for which the locale is constructed or for which it is being used;

- the *particulars* of the locale (i.e., the artifacts, data, tools, actions, etc. that tailor a locale to its use or purpose);

[5] The notion of infosphere appears to be somewhat similar to a locale.

- the *people* who participate in, and interact with, the locale;
- and, the *processes* which exist or arise within and among locales [22].

CIIMPLEX furthers develops these ideas by modeling a new type of integrated business processes advocated by Hammer [12] as enterprise scenarios. It also enables business users to maintain the life cycles of such processes (creation, evolution, and re-engineering).

Many commercial vendors (e.g., I2, Red Pepper) also are working to provide solutions for agile manufacturers to develop better plans [23]. However, these vendors are focused primarily on improving the planning process. The CIIMPLEX approach, however, extends further to include the integration of planning and execution processes.

Acknowledgement

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

References

[1] B. Chu, J. Long, M. Matthews, J. Barnes, J. Sims, M. Hamilton, and R. Lambert. "FAIME: an Object-Oriented Methodology for Application Plug-and-Play", Journal of Object-Oriented Programming", to appear.

[2] A. Goldschmidt. "Report on NIIIP." Communications of the ACM. 39(3):100-103, 1996.

[3] OAG: Open Application Group. Available at www.oag.org.

[4] OMG: Object Management Group. Available at www.omg.org.

[5] C. Billings. Aviation Automation: The Search for a Human-Centered Approach. Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey, 1997.

[6] B. Chu, J. Long, W.J. Tolone, R. Wilhelm, Y. Peng, T. Finin, and M. Matthews. "Towards Intelligent Integrated Planning-Execution." International Journal of Advanced Manufacturing Systems 1(1) pp.77-83, 1997.

[7] B. Chu, W.J. Tolone, R. Wilhelm, M. Hegedus, J. Fesko, T. Finin, Y. Peng, C. Jones, J. Long, M. Matthews, J. Mayfied, J. Shimp, and S. Su. "Integrating Manufacturing Softwares for Intelligent Planning-Execution: A CIIMPLEX Perspective" in Proceedings of Plug and Play Software for Agile Manufacturing Boston, MA, SPIE Vol. 2913, pp. 96-108, 1996.

[8] P. Maes.

[9] A. Bou-Ghannam. "Agent Framework Architecture." CIIMPLEX Internal Design Document. 1997.

[10] M. Robinson. "Double-level languages and Co-operate Working." In AI and Society, 5(1):34-60, 1991.

[11] L. Suchman. "Plans and Situated Actions." Cambridge University Press, 1987.

[12] M. Hammer. “Beyond Reengineering.” Harper Busniess, 1996.

[13] A. Strauss. “Continual Permutations of Action.” Aldine De Gruyter, New York, 1993.

[14] G. Fitzpatrick, W.J. Tolone, and S.M. Kaplan. “Work, Locales and Distributed Social WORLDS.” In Proceedings of the Fourth European Conference on Computer Supported Cooperative Work (ECSCW ’95), pages 1-16, Stockholm, Sweden, September 10-14, 1995.

[15] W.J. Tolone. INTROSPECT: a Meta-level Specification Framework for Dynamic, Evolvable Collaboration Support. Ph.D. Thesis. University of Illinois at Urbana-Champaign, 1996.

[16] G. Fitzpatrick, T. Mansfield, and S.M. Kaplan. “Locales Framework: Exploring Foundations for Collaboration Support.” In IEEE Proceedings of OzCHI '96, to appear.

[17] T. Finin, Y. Labrou, and J. Mayfield. “KQML as an Agent Communication Language.” In Software Agents. J.M. Bradshaw (Ed.), MIT Press, to appear.

[18] J.R. Searle. “Expression and Meaning: Studies in the Theory of Speech Acts.” Cambridge University Press, 1979.

[19] V. Parunak, A. Baker, and S. Clark. “AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design.” Available at www.aaria.uc.edu.

[20] J. Bradshaw, S. Dutfield, P. Benoit, and J. Woolley. “KAoS: Toward An Industrial-Strength Open Agent Architecture.” In Software Agents. J.M. Bradshaw (Ed), MIT Press, to appear.

[21] Compositional Research Group “Caltech Infospheres Project.” Available at www.infospheres.caltech.edu.

[22] W.J. Tolone, S.M. Kaplan, and G. Fitzpatrick. “Specifying Dynamic Support for Collaborative Work within WORLDS.” In Proceedings of the ACM 1995 Conference on Organizational Computer Systems (COOCS ’95), pages 55-65, August 12-16, 1995.

[23] J. Bermudez. “Advanced Planning and Scheduling Systems: Just a Fad or a Breakthrough in Manufacturing and Supply Chain Management?” Report on Manufacturing, Advanced Manufacturing Research, Boston, MA. Dec. 1996.