Accepted for publication in *The International Journal of Agile Manufacturing*

.

Towards Intelligent Integrated Manufacturing Planning-Execution

B. Chu [1] W. J. Tolone[1] J. Long[1] W. J. Tolone[1] R. Wilhelm[2] Y. Peng[3] T. Finin[3] M. Matthews[4]

[1]Department of Computer Science
UNC Charlotte, Charlotte, NC 28223
704-547-456, billchu@uncc.edu
[2]Department of Mechanical Engineering
UNC Charlotte, Charlotte, NC 28223
[3] Department of Computer Science
UMBC, Baltimore, MD 21228
[4]IBM
8501 IBM Dr., Mail Group 70, Charlotte, NC 28262-8563

Acknowledgment

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.

ABSTRACT

In today's competitive global market place, agility has become a survival factor for many manufacturers. A key measurement of manufacturing agility is the time it takes an organization to respond to market changes. The production management system used by most manufacturers today has disconnected planning and execution processes. This situation often hinders the manufacturer to fully explore market opportunities because production plans and executions cannot be adjusted to changes in each other quickly. The Consortium for Integrated Intelligent Manufacturing Planning-Execution (CIIMPLEX) is formed to address this problem by developing an open architecture for manufacturing software applications and deliver integrated planning-execution solutions. With integrated planning-execution, a manufacturing plan can reflect real-time factory capacity. This enables a manufacturer to predict product delivery dates accurately. It also enables a manufacturer to change plans quickly (e.g., on a daily basis) to explore emerging market opportunities.

1. INTRODUCTION

The production management system used by most of today's manufacturers is based on Enterprise Resource Planning (ERP) applications. This model has evolved over the past 30 years [18]. In this model, a set of separate software applications are used for different parts of planning, scheduling, and execution processes.

Capacity Analysis (CA) software determines a Master Production Schedule (MPS) which sets long-term production targets. Enterprise Resource Planning (ERP) software generates material and resource plans. One of the main functions of ERP is to drive the purchase of raw material and subassemblies so that they will be delivered on time for production. Scheduling software determines the sequence in which shop floor resources (people, machines, material, etc.) are used in producing different products. Manufacturing Execution System (MES) tracks real-time status of work in progress, enforces routing integrity, and reports labor/material claims.

There is increasing evidence that such a system fails to support agile manufacturing, e.g. [1]. Key problems include: (1) capacity planning is too rough for adequate decision making, and (2) important data, such as lead times, are not updated dynamically. These problems result from disconnected planning and execution processes. Planners do not have accurate information about factory capacity, which often leads to bloated inventory and inaccurately promised delivery dates to customers.

With matching funds from the National Institute of Standards and Technology, the Consortium for Intelligent Integrated Manufacturing Planning-Execution (CIIMPLEX) is funded to address this problem. Participants of CIIMPLEX include IBM, Ingersoll-Rand, Lucent, QAD, Berclain, Intercim, University of North Carolina at Charlotte, University of Maryland at Baltimore County and University of Florida. The objectives of CIIMPLEX are:

- Develop a framework for intelligent integrated manufacturing planning-execution where manufacturing plans can be based on real-time capacity information, and

• Create an open application architecture to enable manufacturing software applications (ERP, production scheduling, capacity analysis, and manufacturing execution system) to deliver integrated planning-execution solutions.

These objectives are achieved with the help of new technologies that

- Provide real-time data and analysis, and
- Enable enterprise-wide business processes to quickly respond to events in the market place, supply chain, and shop floor.

CIIMPLEX explores the rapidly emerging technologies of software agents and Internet to achieve the above objectives. Unlike traditional software programs, software agents are programs that help people solve problems by collaborating with other software agents and resources on the network. Working with a software agent is like working with a travel agent who is aware of a customer's preferences and can collaborate with other agents and systems to fulfill customer needs. A set of software agents are designed to perform data collection and analysis of plans and schedules at different levels (e.g., master production plan, material/resource plan, and shop floor schedule) . They keep constant vigil against mismatches among these plans and schedules at different levels of abstraction and time horizon. People will be immediately notified if conflicts cannot be corrected automatically.

Consider the scenario of updating a purchase order where the shipment date on a purchased part is changed. A software agent will evaluate the impact of such an event on 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. In an agile organization, such decision-making processes require people to take an enterprise-wide perspective by considering information from multiple perspectives. Such information is often offered by different software applications. Such business processes can dramatically increase a person's cognitive work load [11].The goal of the CIIMPLEX project, in part, is to help people make such decisions by providing appropriate, intelligent assistance.

2. THE CIIMPLEX MANUFACTUIRNG ENTERPRISE INTEGRATION ARCHITECTURE

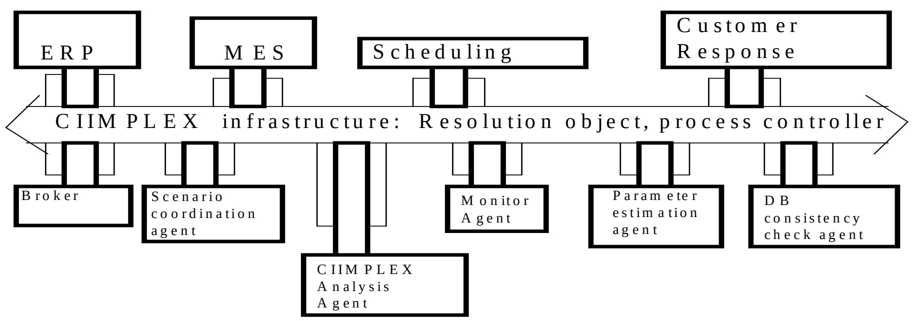

Figure 1. CIIMPLEX Enterprise Integration Architecture

Figure 1 illustrates the CIIMPLEX enterprise integration architecture [3]. One of the main objectives of the CIIMPLEX enterprise integration architecture is to support a new class of integrated businesses that demand people to solve problems from an enterprise-wide perspective. These business processes are represented as *Enterprise Scenarios*. An enterprise scenario, such as the change of purchase order scenario described earlier, commonly is triggered by some external event (e.g., the delay of a shipment). The goal of the enterprise scenario is to make timely decisions 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, tools, and actions in the context specific to particular enterprise scenarios; and
- communication links to the people and/or organizations participating in the scenario.

To use software agents to provide early detection of potential problems, harvest information based on raw real-time data. Software agents can be used to filter relevant information much like they are being used to find relevant information on the Internet.

The deliverables of CIIMPLEX will include:

- Agents that help organizations achieve the objective of integrated planing-execution.
- Tools that help end users to create and maintain enterprise scenarios.
- An open specification of how commonly used manufacturing applications can be plugged together to deliver integrated planning-execution. A reference implementation of that specification will be provided by CIIMPLEX participating vendors.

3. THE ROLE OF SOFTWARE AGENTS IN CIIMPLEX

The CIIMPLEX agents provide relevant information in the context of specific problems [19]. These agents fulfill three broad categories of functions:

- *Data mining and parameter estimation:* aggregate, interpolate and extrapolate from raw transaction data to generate information for higher level analyses, to 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. Agents can be created by the end user to monitor ad. hoc. situations. They also can 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 uses the services of the agents described above.

3.1 Mining Agents and 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:

- Track distributions of scheduling or planning parameters (e.g. process rate) to characterize process uncertainty and data reliability.
- Estimate 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.
- Estimate aggregate information such as lead times and average setup times used on higher levels plans. These parameters can vary widely depending on the product mix and production volume. One of the problems of using existing production management software in agile manufacturing environments is that these parameters have been set to some ill defined "steady state" of the factory. These parameters can be estimated using simulation techniques.

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 groups of users. Examples of such events include changing of purchase order due date, and machine break down. A set of events and their corresponding EAs that are of general interest to manufacturing enterprises will be defined as a base implementation of the CIIMPLEX methodology. Through the CIIMPLEX specification environment, end users can define additional EAs.

EAs can also be configured to take actions automatically. These actions include:

- Start an enterprise scenario by performing a transaction in one of the applications (e.g. ERP).
- Send messages to other agents, e.g., sending messages to start a scenario coordination agent.

The CIIMPLEX Analysis Agents

A set of CIIMPLEX analysis agents (CAAs) are designed. They take as input manufacturing plans. Part of the plan, within the immediate time horizon, will be in the form of a schedule[1]. The CAAs continuously evaluate the current

[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.

plan/schedule with the most recent information obtained from mining agents. They determine what actions should be taken to address disturbances in a timely manner. A typical CAA will perform tasks that 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.

A critical research issue for CIIMPLEX is to provide guidance to reduce system nervousness . The CAA will play a key role towards these objectives.

Database consistency checking

A set of logically consistent databases 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 database 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 has been reversed.

3.3 Scenario Coordination

Scenario coordinators assist people as they carry out (integrated) enterprise scenarios spanning multiple organizations and requiring information and 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 (as 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., [5,7,14,10]). Nevertheless, the scenario coordinator would be remiss if it did not attempt to address this issue, if only in a simplistic manner.

Initially, we are 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 moves forward, 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 [9,15]. Often, disturbances require the attention of several individuals. In order to effectively assist groups of individuals as they (a) synchronously resolve disturbances, communication mechanisms must be available. The scenario coordinator initially supports 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 [8,14,16,17]. 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 update purchase order 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).

3.1 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" 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 and data about the scenario 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 multi-application transactions. 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 coordinator agents use 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. 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.

4. AN EXAMPLE

In this section, we use an example to illustrate the use of software agents and the CIIMPLEX manufacturing enterprise integration architecture. Components of this prototype are illustrated in Figure 2. In this example, a monitoring agent detects that there is a significant decrease of production rate on a work center. A CIIMPLEX analysis agent will 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 will initiate an enterprise scenario to react to this situation through a coordination agent.

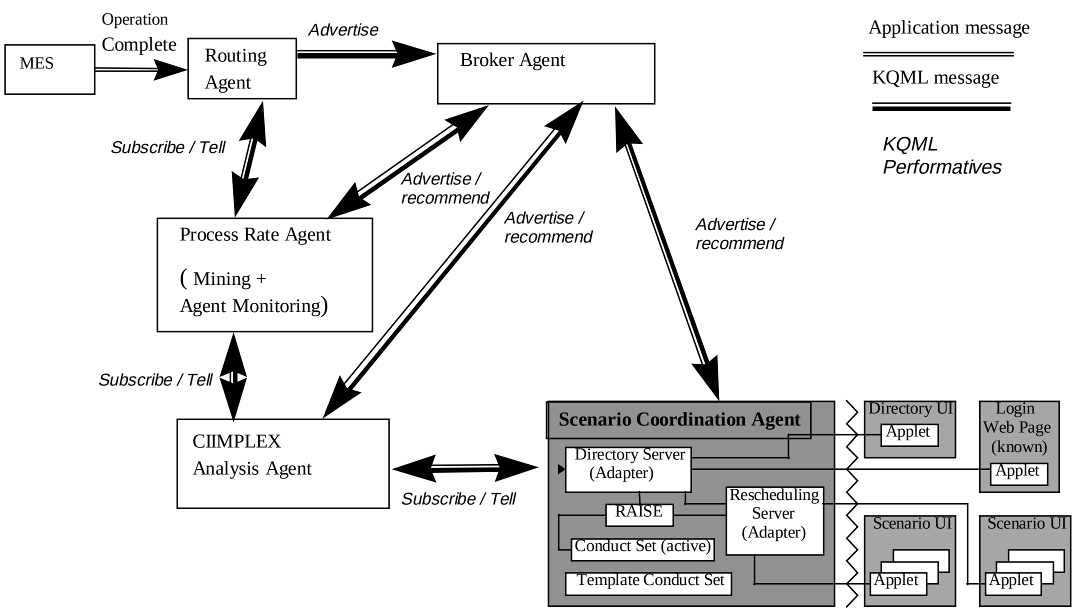

Figure 2. An example

The scenario coordination agent will inform the necessary people about this situation. It will generate a new schedule and access the global impact of the new schedule. It will also support people to make decisions e.g., support what-if analyses by varying priority of orders, ensure proper approval and notifications of delivery date changes. It may also use personal communication agents (e.g. active message on Java phone). It will also carry out final decisions by performing the appropriate transactions in respective applications.

This example consists of the following agents: they are

- A broker agent which matches agents needs with available services,
- A CIIMPLEX analysis agent
- A process rate mining agent which estimates process rates and monitors rate changes

- A message routing agent which translate data formats between different systems
- A scenario coordination agent which helps people execute the enterprise scenario.

KQML, the Knowledge Query Manipulation Language, is chosen as the agent communication language and protocol [6]. 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.

When 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 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 percent from the current value, then CAA needs to be notified. PRA is thus an example of both a 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. CAA or other agents who need this information can ask BA to recommend or recruit agents that possess such rates for them. The subscription / notification architecture allows agent analysis functions to be added incrementally.

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 sensitivity analysis, and to provide certain statistical forecasting and estimation.

End users work with scenario coordination agents through Java-enabled browsers such as the Netscape Navigator.

5. 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 [13], KAoS [2], INFOSPHERES [4], and WORLDS [12,8,7].

The CIIMPLEX manufacturing enterprise integration architecture shares with most of them 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 respects 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, 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[2]. A locale 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-Supported Cooperative Work (CSCW).

CIIMPLEX furthers develops these ideas by modeling the type of new business processes advocated by Hammer [11] as enterprise scenarios. It also enables business users to maintain the life cycles of such processes (creation, evolution, and re-engineering).

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

Many commercial vendors (e.g. I2, Red Pepper) are also working to provide solutions for agile manufacturers to develop better plans [1]. However, they are primarily focused on improving the planning process. CIIMPLEX's approach further extends to the integration of planning and execution processes.

REFERENCES

1. Bermudez, J. "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.
2. 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.
3. Chu, B., Tolone, W. J., Wilhelm, R., Hegedus, M., Fesko, J., Finin, T., Peng, Y., Jones, C. Long, J., Matthews, M. Mayfied, J., Shimp, J., & Su, S. "Integrating Manufacturing Softwares for Intelligent Planning-Execution: A CIIMPLEX Perspective" in Plug and Play Software for Agile Manufacturing, Boston, MA Proceedings of SPIE Vol. 2913. pp. 96-108, 1996.
4. Compositional Research Group "Caltech Infosheres Project" available at http://www.infospheres.caltech.edu.
5. 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.
6. Finin, T., Labrou, Y., & Mayfield, J. "KQML as an agent communication language". In *Software Agents.* Bradshaw, J.M. (Ed.) MIT Press, to appear.
7. Fitzpatrick, G., Mansfield, T., & Kaplan, S. "Locales Framework: Exploring Foundations for Collaboration Support". In *IEEE Proceedings of OzCHI '96*. To appear.
8. 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.
9. 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.
10. 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.
11. Hammer, M. *Beyond Reengineering : How the Process-Centered Organization Is Changing Our Work and Our Lives* Harpercollins, 1996.
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. 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
14. Robinson, M. "Double-level languages and co-operative working". *AI and Society*, 5(1):34-60, 1991.
15. 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.
16. Strauss, A. *Continual Permutations of Action*. Aldine de Gruyter.New York, 1993.
17. Suchman, L. *Plans and Situated Actions*. Cambridge University Press,1987.
18. Vollmann, T., Berry, W. & Whybark, D. *Manufacturing Planning and Control Systems* Irwin: New York, NY. 1992.
19. Gio Wiederhold: Value-added Mediation; Proc. of the IFIP DS-6 Conference, to appear in Meersman(ed): Database Application Semantics, Chapman and Hall, 1995.