LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

UN-11A 89-3244

LA-UR--89-3244

CONF-900137--1

DE90 000636

001 1989

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-38.

TITLE: EXPERIENCES WITH KNOWLEDGE-BASE AND LARGE DATABASE INTEGRATION

AUTHORS: Christie Michelsen
Donna Smith
Glenda McGrath

SUBMITTED TO: Hawaii International Conference on Systems Sciences

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

By submission of this article, the author recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the substance form of this contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

**EXPERIENCES WITH KNOWLEDGE-BASE
AND
LARGE DATABASE INTEGRATION**

**Christie Michelsen
Donna Smith
Glenda McGrath**

**Military Systems Analysis and Simulation Group
Los Alamos National Laboratory
Los Alamos, NM 87545**

Abstract

This paper will discuss the integration of several knowledge-bases (network data structures providing inheritance and rule-base definition/execution) and a large database in the context of a manufacturing simulation. The simulation represents the production activities at numerous facilities and is used to determine if a production schedule can be satisfied. The simulation is supplied with data from a large relational database which includes production facility characteristics as well as piece part information. The simulation is currently in use with new functionality being incorporated at this time. This paper will also illustrate the problems inherent in loosely-coupled knowledge-base and database data exchanges.

I. Introduction

This paper will present an overview of the architecture of a simulation which models a large production complex consisting of multiple distributed facilities and discuss the integration of the various data repositories. As in any manufacturing simulation, data reflecting resources, capacities, processing times, process and resource scheduling heuristics, and bill of materials (the final production part and all its associated children including the parent-child relationships) are essential for accurate interpretation of the capabilities of the complex or individual facility to meet production requirements. The necessity to perform cost analysis associated with the production of individual components and the aggregation to the entire assemble of parts presented many technical problems. The necessity to also investigate cause-effect relationships as well as inventigate modifications to the complex presents numerous challenges. These challenges and solutions will be presented in the paper.

The simulation itself is actually only one component in a series of modules, where each module represents different functionality and capabilities. This series of modules is refered to as the Workload Model. These various modules will be discussed in Section III entitled "The Workload Model."

There exists a menu hierarchy for simple retrievals against the database as well as numerous canned SQL queries and C programs. There also exists the capability to dynamically create new knowledge-bases, which represent new weapon systems, and then store the knowledge-base information to the database tables for future access. This is refered to as look-alike weapon systems. The requirement for bi-directional transfer of information between the knowledge-bases and databases presents many challenges in an

analysis tool. The menus and look-alike capability will also be discussed in detail.

This paper will first investigate related work in this areas of manufacturing simulations, object-oriented techniques, and knowledge-base/database integration. An overview of the application's functionality will then be presented followed by the implementation details associated with the database, knowledge-base, simulation, and associated modules. The problems encountered in the design and implementation will then be addressed as well as future requirements and challenges.

II. Background

Traditional manufacturing MRPII (Manufacturing Resource Planning) systems [1] are closed loop systems written in high-level procedural languages such as COBOL or FORTRAN and are generally very difficult to modify. The approach presented in this paper differs both in functionality and implementation from traditional methods. To obtain a realistic representation of the manufacturing complex which was modeled, it was important for the Workload Model to explicitly represent the decision-making capabilities that occur on the manufacturing floor in day-to-day operations. There have been other manufacturing applications which have incorporated these decisions in rule-based systems [2], but the extremely large database requirements as well as the flexibility and extensibility required to represent modifications to the complex present a different problem.

The object-oriented simulation approach for manufacturing schedulers has recently emerged and in fact several tools are currently being marketed [3]. These tools were investigated but were found to be too restrictive given the functionality required. The object-oriented paradigm is very amenable to the simulation of real-world, complex systems. In [4,5] the basic concepts of object-oriented design are discussed. The characteristics such as

modularity, inheritance, abstract type mechanisms, and message-passing are all conducive to simulation design, implementation, and maintenance. The real-world systems which are modelled are typically dynamic, evolving systems and the end-user quite often needs to investigate "what-if" situations (modifications to the complex, production demands, or weapon system characteristics) that may occur in the future.

The integration of knowledge-bases and databases is a topic currently under examination. Stonebraker [6] presented an overview of loosely-coupled and tightly-coupled databases and knowledge-bases. The advantages and disadvantages to the approaches were discussed and indicated a preference of tightly-coupled databases in order to remove duplication. Tightly-coupled databases imply a rule system residing in the database; loosely-coupled imply two separate systems which exchange data.

There has also been work performed in integrating an object-oriented programming language with a database system [7]. This is similar but somewhat different from the integration of knowledge-based systems and relational database systems; especially when as in the Workload Model, the structure of the data is vastly different in the two systems. Work has also been performed in the area of integrating knowledge systems and object-oriented databases [8] as well as integrating logic programming with databases [9].

III. The Workload Model

Prior to the development of the Workload Model, no consistent or integrated data existed on the production processes at the various facilities which comprise the DOE Weapons Production Complex. A central integrated source for the cost information and default production schedules did not exist. Inconsistent and hardcopy only data were common problems. Information on production times (time duration from actual start of production to end of production), inter-shipment times (time duration from end of production to arriving at

the next facility), preparation times (time duration from receipt of all subcomponents to start of production), resource requirements including equipment, personnel, and capacities, lot size requirements, in-house attrition, starting inventories, procurement processes and times, etc. had to be defined and collected. Each facility is operated differently and thus they maintain different information. The number of interconnections represented in the model and the data storage requirements are immense. The massive data requirements and the need to define and represent scheduling heuristics were very ill-defined at each of the production facilities. This greatly complicated the problem from that of a traditional manufacturing scheduler.

The Workload Model is a tool to allow the end-user to perform cost analysis studies for the various weapon systems. The model also provides the capability to analyze changes in the production complex and to perform planning activities associated with the DOE weapons complex. The user can investigate new weapon systems, their anticipated costs, and time frames for new production. The model provides the analyst the capability to study budget cuts and tradeoffs between make and buy decisions.

The Workload Model consists of numerous integrated modules which provide an analyst the capability to perform cost estimation for the production of ten different weapon systems. The analyst also has the capability to investigate if the production complex can meet a user-specified production schedule or build schedule. If the build schedule can not be met, the analyst has the capability (through post-processing queries which are non-DBMS access procedures which manipulate the output stored in the knowledge-bases) to determine the reason the production schedule could not be satisfied. The analyst also has the capability to dynamically create new systems and determine the effect of additional production requirements on the production complex. The database which drives the generation of the knowledge-bases and thus the execution of the simulation contains enormous quantities of data on cost, production information per piece part, resource information, DOE published build

schedules, starting inventory levels, and weapon system information such as how it is used, which service owns it, etc.

The Workload Model used an object-oriented design and implementation for the simulation components. It is written in LISP and uses KEE [10] (Knowledge-Engineering Environment, marketed by Intellicorp). The relational database is implemented and accessed using ORACLE and C language utilities. This will be discussed in more detail later.

The design of the Workload Model was centered around the concept of actors that comprised the real world system. These actors can also be viewed as objects in the object-oriented paradigm and represent entities in the production complex such as battery production or a particular building at a facility. The actors have physical and cognitive capabilities. The physical capabilities, for example, include adding and removing items from inventory, production of a part, assembly of a weapon system, adding personnel, shutting down a facility, or procuring a part. The cognitive capabilities include hiring personnel, make/buy decisions, and determining the right combination of weapon systems that can exist in a given physical location. These cognitive capabilities are the decision-making activities of a particular actor. Actors also send messages to other actors to initiate certain actions. This modular approach to design and implementation provides a very flexible environment, amenable to adding new actors to the simulation as well as temporarily or permanently removing actors (facilities). Actors are designed to be stand-alone entities that exist at the facilities and only rely on other actors to supply component parts for which they are responsible.

In the Workload Model, seven production facilities are modeled. These are broken down into 39 actors. The Workload Model schedules production for ten different types of weapon systems each composed on the average of 200 piece parts (individual items, subcomponents, subassemblies and final assemblies. The

typically production cycle extends for fifteen years. Numerous systems are produced each year. The simulation of a typical build schedule will require the scheduling of 360,000 demands. The analyst has the capability to modify the scenario to produce any number of systems over any time frame and determine if the production schedule can be satisfied.

III.A Functional Capabilities

The screen image presented to the user by the Workload Model is depicted in Figure 1.0 entitled "The Workload Model Desktop." In the upper right hand corner is a menu which guides the user through the system. For example, since the "Resource Scheduler" relies on the "Demand Schedule" to produce the appropriate input, the model will not allow the analyst to run the "Resource Scheduler" prior to producing the "Demand Schedule" and similarly the user cannot run the "Simulation" until the "Resource Scheduler" has been run. Figure 2.0, entitled "Workload Model Module Dependencies", illustrates the interconnections and functional relationships between the various modules and databases. These various phases of the simulation process will be discussed shortly. The hierarchy of menus, where the main menu can be located in the upper left-hand corner of the Workload Model Desktop as illustrated in Figure 1.0, entitled "Workload Model Desktop" also contains the post-processing queries required by the analyst to examine pre-processor and simulation runs. Certain queries are designed to be executed at different times within the overall simulation process.

The Workload Model Desktop
Figure 1.0

Workload Model Module Dependencies
Figure 2.0

Initially the user can specify a new build schedule, modify an existing schedule, or run with the default schedule. The user can specify the number and completion dates (either annually or monthly) by type of system and the priority associated with that build. The analyst then initiates the "load knowledge-bases" command which accesses the database for only the necessary weapon systems and creates the requisite knowledge-bases. A portion of the knowledge-bases (KBs) represents the bill of material for each weapon system (one KB per system). This is in the form of a network which represents the parent-child relationships. Figure 4.0, entitled "Bill of Material Knowledge-Base" represents one such network which represents a bill of material which depicts the parent-child relationships of all piece parts in a particular weapon system.

SAMPLE WEAPON SYSTEM

A Bill of Material Knowledge-Base

Los Alamos

Figure 4.0

After the analyst has completed the specification of the build or production schedule, the "Demand Scheduler" should be run. This is a piece of backward scheduling software which uses the final system completion date to determine, in isolation, the start dates for the various subcomponents, i.e. the software runs back in time recursively determining the completion dates for child parts. In this phase, resources, capacities, and lot sizes are not considered; however percentage of in-house attrition is calculated at this time. If too many parts are scheduled to complete prior to the overall start date of the simulation (the critical path time frame is longer than the simulation time frame), the analyst should modify the build schedule; the simulation execution will never be able to satisfy the demands in the user-specified overall time frame. Incorporated in the determination of the start dates are inter-ship times, preparation time, and processing time. Calendars are created dynamically via the Demand Scheduler to track completion dates. The final demand schedule is stored in the Demand Schedule knowledge-base.

The next phase of the Workload Model Simulation Process is the generation of the resource schedule. This is also computed via a backward scheduler which utilizes the current demand schedule and attempts to aggregate demands into lots, and determine if sufficient resources (given other competing requests) are available to produce the newly derived demand. Prior to running the Resource Scheduler, the user also can modify or remove the availability of a particular resource. The Resource Scheduler will attempt to move backward (earlier) in time, within a user-specifiable window, to meet resource requirements or lot size aggregation. In moving production start dates backward in time, all children that this part is composed of must also be moved back as well. This process is complicated if the children have been aggregated with other demands. The backtracking and rescheduling can be extremely computationally expensive given the number of demands typically scheduled which is typically 360,000. In addition, one part may be required for several different types of weapon systems and therefore be a component of multiple demands. The output of this phase is a schedule of

production which contains production start dates for all piece parts. The user now has the capability to examine anticipated resource problems and to run additional queries against the resource scheduler output contained in the "Resource Scheduler" knowledge-base. Figure 4.0, entitled "Resource Scheduler Post-Processing Queries," illustrates the types of post-processing queries that can be executed to facilitate the interpretation of both the resource scheduler and simulation results by an analyst.

```
QUERIES MENU
MINIMUM RESOURCE UTILIZATION
MAXIMUM RESOURCE UTILIZATION
RESOURCE UTILIZATION
RESOURCES SCHEDULED ABOVE A THRESHOLD
COMPONENTS REQUIRING A SPECIFIED RESOURCE
EARLIEST PRODUCTION TIME FOR A PART
EXIT QUERIES MENU
```

Resource Scheduler Post-Processing Queries

Figure 4.0

The analyst can also determine when the earliest a single new production request can be completed. This is a forward scheduler comparing resource availability and the original resource schedule. The Resource Scheduler maintains in knowledge-bases the resource availability and maximum capacity on a monthly basis for the projected production period. These knowledge-bases can then be used for subsequent post-processing. The Resource Scheduler also places start events for the piece parts in the controller queue to be executed in the course of the simulation. In the future, both the Demand Scheduler and the Resource Scheduler will be referred to as the Pre-Processor.

The next phase of the Workload Simulation is the actual simulation which executes the events previously placed in the controller queue by the Resource Scheduler. Prior to actually running the simulation, the analyst can modify resources, e.g.,

shutdown a facility for April and May during the simulated production period, add a new machine press to a production facility, or reduce starting inventories of a particular piece part. It is during the simulation that the scheduling heuristics are applied. To rectify a resource problem, resources may be juggled, e.g., another shift may be added, overtime may be used, hiring personnel may be planned for a future time frame, or alternative suppliers may be used. Rules concerning the duration of change or restrictions are also included, e.g., overtime can only occur for two weeks because of personnel burn-out or ensure that the budget allocation for overtime has not been exhausted. Typical events in the simulation include start production, end production, add to inventory, remove from inventory, wait on inventory, and resolve a resource problem. The analyst can run the simulation allowing resources to fall below zero (essentially, unbounded resources but tractable) or force resources never to fall below zero. There are several post-processing queries which allow the examination of exception conditions. The occurrence of a negative resource available value indicates bottlenecks in the production system. Figure 5.0, entitled "Simulation Execution," is a window which depicts the execution of the simulation. Images are attached to the inventories for two different piece parts.

Simulation Execution

Figure 5.0

Los Alamos

The user can dynamically create a new system from existing systems. The Workload Model allows the user to graphically prune a weapon system tangle (knowledge-base representing the network of parent-child relationships), attach a part and all of its associated children to another part (node) in the network, and move nodes from one tangle to another. The result is the creation of a new weapon system which is stored in a knowledge-base as a collection of parent-child relationships. The user can save the new system to the database for processing in the future and incorporate the new weapon system in a build schedule for subsequent simulation.

Of a somewhat different nature is the cost analysis and weapon system queries which operate on the database itself. A subset of these queries include: determination of costs associated with an entire system; critical path calculations; bill of material structure for any part, cost reports by component, assembly, weapon or producer plant; cost expenditures and savings for weapon system cancellation; production schedules for a weapon system; and operational and engineering characteristics.

III.B Design and Implementation Methodology

As mentioned previously, the Workload Model is written in several languages and utilizes several tools. The pre-processor, simulation, and post-processor portion of the model is written in LISP and KEE. KEE is an expert system tool which is frame-based and provides the capability for inferencing using either backward or forward chainers. The manufacturing scheduling heuristics are represented as IF - THEN - ELSE structures (rules) which the inference engine can utilize. The attributes, which are characteristics, and assets, which are possessions, of the various simulation actors as well as their physical and cognitive capabilities are stored in the knowledge-bases. Some of the attributes and assets are scenario dependent and can be specified by the user or loaded from the database. The physical and

cognitive capabilities are not loaded from the database but rather are realized procedurally as methods which are initiated via a message. The DataBase Management System used was ORACLE. Several of the predefined queries are nested SQL queries as well as C language utilities with embedded SQL queries. Recursive traversal of the relational tables which represent the parent-child relationships of the network was often required. For efficiency reasons, access to the database was not permitted from the simulation side of the application. This requirement for certain queries to access the database directly was intentional to take advantage of the flexibility of the data manipulation facilities and execution efficiency.

The software which supports the integration of the knowledge-bases and the database is written in C with embedded SQL statements and LISP. The interconnection is hardwired and the versatility and generality that could exist from a tightly-coupled interface is missing. This will be discussed in more detail later. Originally KEE Connection [11] was acquired to provide this capability. However, resulting from a delayed production version of the system and concerns over query generation and execution efficiency, the Workload Model opted to develop a loosely-coupled interconnection.

KEE also provides the capability for the user to attach active images to various attributes or assets of an actor and view the changes to that data item over time. This can be useful for viewing the impact of changes to particular inventories or to a resource, which is perceived to be a problem, through the execution of the simulation.

Several of the knowledge-bases had to be re-designed to facilitate efficient searching and sorting. Access to knowledge-base objects and their associated data elements (slots) is substantially less efficient than the database accesses. Thus, there was a large effort to optimize the simulation software and the knowledge-base definitions and structures. Software was redesigned to remove extraneous

instruction execution, to remove recursive functionality when stack size limits were encountered on the SUN, and general inefficiencies were identified, e.g., the setting of certain KEE flags which resulted in improved execution. The knowledge-bases were redesigned to reduce the number of KEE access invocations which were found to exhibit additional overhead and to reduce execution time associated with knowledge-base and object creation.

III.C Analysis Studies

The types of studies currently underway or anticipated in the future include the ramifications to the production complex of a new weapon system with characteristics similar to another existing system. The analysts also investigate production trade-offs between different weapon systems, i.e., can ten additional type A systems be produced per month instead of five type B systems. The analysts are finding the database extremely useful for answering cost-related study questions which were not originally envisioned for this model. The database provides an integrated, validated repository for cost data and weapon system characteristics, providing substantially increased turn-around. for studies.

III.D Database and Knowledge-base Integration

Two situations compounded the problem associated with integrating the database and the knowledge-bases. The first situation was the massive amounts of data stored in the database and required by the simulation. Currently the database occupies four gigabytes and consists of 17 tables. The software implemented selectively loaded only data which was required by the study scenario as presented by the analyst. Systems which were not specified were not be stored in knowledge-bases. As stated previously, the kbs are dynamically created and intialized; the execution time would suffer if all the data was uploaded.

To illustrate the activities associated with the bi-directional transfer of data, a brief overview of the process will be discussed. When the software determines a particular weapon system and its production information is required for the pre-processor and simulation to execute, the C code recursively extracts the final assembly and all of its children, maintaining all relationships of the generated network. An intermediate form is produced which contains data labels and transformed data. Often the transformed data format was substantially different than the database format. For example, the use of lists by LISP resulted in numerous transformations. This intermediate form was utilized by LISP code to create and initialize the knowledge-base. Each part in the weapon system, a node or object in the knowledge-base was created and the appropriate parent pointers created. The object was then initialized with the associated part information, e.g., part specific production information. The inverse to this process utilized LISP code to translate the knowledge-base to the intermediate form and then C code to create the appropriate tuples in the various database tables.

The functionality and execution efficiencies necessary to implement the various queries and access the extremely large database required the use of a DataBase Management System (DBMS). The power represented by the DBMS in the data manipulation facilities, the data dictionary which provided data verification and view manipulation, the forms capability for data entry purposes, and the report generation capabilities were too important to the implementation. Searching the database through the DBMS, both thru SQL and C, was much more efficient than searching the various Knowledge-bases. Thus, the cost analysis queries are performed through the database and the pre-processor, simulation, and post-processor requests are performed through the knowledge-base.

The second situation was the need for multiple relational tables

to be automatically mapped to a network representation and vice versa. All tables in the database are normalized and parent-child relationships are explicitly stored as two attributes in one table. This facilitates the generation of the network representation. Because the tangles (knowledge-bases) can be created dynamically, software must exist to map these new networks back to the database and create the necessary associated data items which are stored in other tables.

IV. Problems Encountered

One of the problems faced by the designers of the Workload Model was the activities associated with manipulating the simulation controller (event) queue. Because of the large number of events initially stored in the queue (360,000), the simulation controller queue was implemented as multiple queues. The Workload Model was implemented as a time step model where the time step was one month. Each time step was partitioned into types of events: initialization events, normal events executed on a daily basis for that time step, and wrap-up events. Therefore there exists three queues for each month and its associated pre- and post- events. Immediately prior to the month, wait on inventory situations are resolved. Processing events (start and end) are resolved in the monthly queue and items are added to inventory as a wrap-out to ensure synchronization.

More efficient sorting algorithms were written to handle the large number of items maintained by the simulation in the various list structures (e.g., scheduling queues and calendars). Model execution and database execution required constant examination. Several components of the model are currently being redesigned to enhance execution efficiency. The database was also optimized, but network packet traffic is still a major concern. The model runs on diskless workstation connected to a large fileserver.

Model and Data Validation also presented numerous problems. Cost data which was received was often incorrect. The production information which was supplied by the various production facilities was inconsistent with one another and there was often missing data. The process of correcting and entering the data took approximately one year. To verify the model and its associated data, numerous representatives from the sponsoring organizations were invited to analyze results.

At present, the model is being used in a single user mode. If and when multiple users concurrently run the model and update the database, the issues of database consistency and reliability will have to be examined. There is currently no method for the knowledge-base to detect when the database has been altered. We do not anticipate this to be a significant problem due to the fact that production and cost data is updated only annually; the creation of look-alike weapon systems is where the database is updated periodically.

X. Future Work

There are two primary areas which require further work. The first area is automated analysis. As experience is gained in using the Workload Model as an analysis tool, the methodology used in the analysis will be automated to provide simple interpretation of simulation results. This goes beyond the post-processing queries currently implemented and will attempt to answer questions such as: Why did a resource problem occur in month 13; Why wasn't overtime permitted; or What is the result of a three percent budget reduction for the entire production complex.

The second area is the use of a more tightly-coupled database and knowledge-base interface. KEE Connection will also be examined with respect to efficiency and ease of integration. Along these lines, the use of object-oriented databases are also being explored. Static

[12] and PostGres [13] are currently being investigated for integration into the model. In initial examinations of production versions of object-oriented databases, the products are still premature with respect to the long-term potential. Both the use of a tool to provide a tightly-coupled interface and object-oriented databases could provide a more efficient approach to many of the database-related problems. There is still some question regarding the performance of these tools as demonstrated in [14] and therefore, they must continue to be monitored in the future.

XI. Conclusions

This paper has provided an overview of the design and development of the Workload Model, a simulation of a manufacturing production complex. The simulation actually relies upon knowledge-bases which are created from an extremely large database. The problems associated with interpreting large amounts of simulation data and integrating knowledge-bases with a database were addressed. Mapping massive amounts of data from a relational format to a network format and vice versa still requires additional investigation. The current approach hardwired this capability into the model and a more general purpose approach which is efficient would be desirable. This paper has demonstrated that the integration of large databases, knowledge-bases and object-oriented simulations is feasible and provides very attractive benefits which include modularity, flexibility, and execution efficiency.

XII. References

- [1] Riggs, J. L., Production Systems: Planning, Analysis and Control, Third Edition, 1981, John Wiley & Sons.
- [2] Bruno, G., Elia, A., and Laface, P., "A Rule-Based System to Schedule Production." In *Computer*, Vol. 19, No. 7, pp. 32-40, July 1986.

- [3] SimKit User Manual, Intellicorp, 1989.
- [4] Stefik, M., and Bobrow, D.G., "Object-Oriented Programming: Themes and Variations." In *The AI Magazine*, pp. 40-62, January 1986.
- [5] Meyer, B. Object-Oriented Software Construction, Prentice Hall, 1989.
- [6] Stonebraker, M. and Hearst, M., "Future Trends in Expert Data Base Systems." In Expert Database Systems. Proceedings from the Second International Conference, pp. 3-20, 1989.
- [7] Kim, W., Ballou, N. Chou, H., Garza J. G., and Woelk, D., "Integrating an Object-Oriented Programming System With a Database System." In *OOPSLA '88*, pp. 142-152, Sept. 1988.
- [8] Brodie, M. L. and Jarke, M., "On Integrating Logic Programming and Databases," In Expert Database Systems. Proceedings from the First International Workshop, pp. 191-207, 1988.
- [9] Zaniolo, C., Ait-Kaci, H., Beech, D., Cammarata, S. Kerschberg, L., and Maier, D., "Object Oriented Database Systems and Knowledge Systems," In Expert Database Systems. Proceedings from the First International Workshop, pp. 49-66, 1988.
- [10] KEE User Manual, Intellicorp, 1989.
- [11] KEE Connection User Manual, Intellicorp, 1989.
- [12] Weinreb, D., Feinberg, N., Gerson, D., and Lamb, C., "An Object-Oriented Database System to Support an Integrated Programming Environment." Preliminary Draft of 8 April 1988, Symbolics.
- [13] Stonebraker, M., Hanson, E., and Potamianos, S., "A Rule Manager for Relational Database Systems." Memorandum No. UCB,ERL M97/38, Electronics Research Laboratory, University of California, Berkeley, 1987.
- [14] Duhl, J. and Damon, C., "A Performance Comparison of Object and Relational Databases Using the Sun Benchmark." In *OOPSLA '88*, pp. 153-163, Sept. 1988.