Experience with IMDAS in the Automated Manufacturing Research Facility

Edward J. Barkmeyer and Jane Lo
National Institute of Standards & Technology

1. The Automated Manufacturing Research Facility

The Automated Manufacturing Research Facility (AMRF) is a unique engineering laboratory at the National Institute of Standards and Technology (NIST†) Center for Manufacturing Engineering. The facility provides a basic array of discrete metal parts manufacturing equipment and control systems – a testbed – that researchers from NIST, industrial firms, universities and other government agencies can use to experiment with new standards and to study new methods of measurement and quality control for automated factories. Construction of the facility began in late 1981 and the facility has been operating as a fully functional laboratory since late 1986, although improvements and equipment changes are continually ongoing. The project is funded by NIST and the Navy Manufacturing Technology Program and is significantly supported by industry through donations or loans of major components and through cooperative research programs. The goal of the project is to identify and exercise potential standard interfaces between existing and future components of small-batch manufacturing systems and to provide a laboratory for the development of factory-floor metrology in an automated environment, delivering proven measurement techniques to American industry. In addition, the AMRF is being used as a testbed for research on the next generation of "knowledge-based" manufacturing systems.

To provide a realistic testbed for interface standards, the AMRF is intentionally composed of manufacturing and computing equipment from many vendors, thereby making its construction a major integration effort [Nanz84]. The configuration is structured around single self-contained "workstations", each capable of executing a well-defined set of manufacturing functions. Each workstation can operate either as an independent manufacturing unit under control of a local operator, or as an element of a multi-workstation manufacturing complex under control of a higher-level process. The intelligence complex of a typical workstation includes a robot control system, a machine tool control system, an automated storage controller, sophisticated sensory systems and a workstation controller to coordinate the activities. Coordination of multiple workstations in the complete production of a part batch is performed by a higher-level process termed the "cell" controller. All of these control and sensory processes are software/hardware systems which reside on a complex of interconnected computer systems, making the AMRF a distributed computing network. Elsewhere on the same network are the manufacturing engineering systems, including computer-aided-design (CAD) systems and process planning systems.

The 1988 AMRF (Figure 1) comprised the Horizontal Milling Workstation, the Vertical Milling

† The National Bureau of Standards became the National Institute of Standards and Technology on August 23, 1988, when the Omnibus Trade and Competitiveness Act became law. NIST retains all former NBS functions while expanding its mission to encourage improved use of technology by American industry.

NIST Internal Report 89-4132
National Institute of Standards & Technology
Gaithersburg, MD, October, 1989

Workstation, the Cleaning and Deburring Workstation, the Inspection Workstation, a Materials Handling Station and the Mare Island Flexible Manufacturing Station. The Materials Handling Station comprises an automated storage and retrieval system (ASRS), two automatically guided vehicles (AGVs) and several semi-automatic load/unload stations.

The control architecture of the AMRF is the hierarchical control model defined by Albus and Barbera [Albu81, Barb82]. In this model, each control or sensory process is associated with a logical identification, which conveys a clearly defined set of functions (or "work-elements"), rather than a particular piece of equipment. And on this collection of functional modules, a chain-of-command structure is imposed, so that any module has exactly one supervisor, which originates tasks for the module, and a collection of "dedicated" subordinates, to which it may distribute subtasks.

The network software architecture of the AMRF follows the International Standard Open Systems Interconnection Reference Model (OSI). The important aspect of this model is that it formalizes and separates the logical process-to-process link from the physical network considerations. By exposing only a common program-to-program communications capability to the production management programs, it insulates them from networking concerns. In particular, implementation of the OSI model permits a single physical medium to multiplex many separate process-to-process communications, and a given process-to-process connection to use several separate physical connections with relays between them. The 1988 AMRF network (Figure 2) [Rybc88] actually uses both of these capabilities, but they are invisible to the control software.

2. The Data

There are five principal classes of information in the AMRF, each of which has its own collection of repositories: design data, process planning data, resource planning data, work-in-process data and tooling data.

Design data originates in ComputerVision† and Autotrol Computer-Aided-Design systems and is currently uploaded into a common form in a VAX-resident private database of the AMRF "Geometry Modeling System". This system permits different views of the geometry data to be extracted for such differing applications as process planning, NC cutter path generation, robot grip point identification and inspection feature identification [Tu87].

Process planning data originates in two different systems, both of which use feature information from the geometry data and generate process plans in the AMRF Process Plan Interchange Format[Brow86], which are then stored in local files at the generating sites. Such information units include:

1) Resource lists: data and material resources required to be present at a workstation for production of a particular part type.

2) Process plans: a hierarchy of operations sequences for fabricating specific parts, organized as scripts for different logical elements of the hierarchical control system.

† Certain commercial hardware and software products are identified in this paper in order to specify the IMDAS operating environment adequately. Such identification does not imply endorsement by NIST, nor does it imply that the products identified are deemed the best available for the purpose.

3) Control codes: robot programs, NC programs, inspection programs for the control of specific automata in the manufacture of a specific part.

4) Kits: instructions for removing raw materials from inventory and packaging them for delivery to workstations.

Once generated, this information is never modified, unless a new "version" is created. This class of data is retrieved by virtually every controller on the floor at some point in the production process.

The AMRF intentionally lacks a plant administration system - such a system was viewed to be beyond the scope of the "manufacturing laboratory". Consequently, the resource planning and administrative information is limited to:

1) Resource allocations: allocations of trays, blanks and machine time for production orders being scheduled; and

2) Shop schedules: schedules for the use of machining and transportation resources in the production of "customer" orders.

Order data for parts for which engineering data has previously been developed is fed directly into the high-level "cell" process, which then stores it in its local database and extracts it as necessary to feed the scheduling process.

Work-in-process information is kept primarily in Ingres† databases on one of two central data servers and secondarily in various local repositories in controllers on the factory floor. This information includes:

1) Orders and work-orders: customer orders for parts fabrication and their status, internal work-orders generated from the process-plans for such parts and their status;

2) Parts inventory: count, location, composition, and possibly measured geometries, of part blanks, in-process workpieces and finished parts;

3) Tray status: all trays in the AMRF, current location, configurations, contents and relative locations of workpieces or tools on them.

One of the benefits of a totally automated facility is that such information can be automatically acquired. But as a consequence, it is also frequently accessed and updated, by the cell controller, the material handling station and the material management functions within the workstations themselves.

The information on tools and machines is also stored in the central data servers and in the local repositories of the controllers. This includes:

1) Tools: count, type, location, status and remaining lifetime of all portable tools, fixtures and end-effectors in the facility.

† Ingres, a trademark of Relational Technology, is a relational database management system which may be run on several different computer systems.

2) Machines: identification, type and location of machine tools and material handling equipment, status and time-in-process of the current machining task, total hours in service, coolant levels, contents of the tool changer, etc.

Most of the tooling information is accessed and maintained by the cell and material handling controllers, and by the tool management functions within the workstations, but the machine status is almost exclusively maintained by the machine controllers themselves.

3. The Integrated Manufacturing Data Administration System

Conceptually, all information exchanged among AMRF processes passes through a single integrated database, managed by a single global database manager (Figure 3). The conceptual AMRF database is in reality a collection of separate disk-resident databases and memory areas distributed over the subsystems of the facility, and the global data manager is a distributed set of processes sharing the global data management function. This set of data management processes is called the Integrated Manufacturing Data Administration System (IMDAS) [Bark86, Su86a, Kris87]. It provides access to existing databases distributed over many computer and database systems, allowing both updates and retrievals via a common interface to application programs.

IMDAS internally models the entire AMRF information collection in a unified "enterprise" model. The information modelling method IMDAS uses is the Semantic Association Model – SAM* – developed by the University of Florida [Su83, Su86b]. This method represents information as a semantic network, and is therefore capable of representing the complex structures and relationships and many integrity constraints found in the manufacturing enterprise. The IMDAS data manipulation language (DML) resembles the American National Standard relational data manipulation language SQL [ANSI86], but its semantics and some of its constructions are adapted to the SAM* information model.

An application program phrases a transaction to the IMDAS in a character string form of the DML, specifying source and destination data areas, if any. The referenced data areas may be local files or shared memory areas in a fixed format (or "report form") chosen by, and presumably convenient to, the application program itself (Figure 4). The reason for the string representation of transactions and the user-specified and formatted data areas is that this permits database access by arbitrary controllers on the floor, without the need for precompilers or elaborate user-interfaces.

Internally, the IMDAS, like the AMRF manufacturing complex, is a hierarchical control system, and within it, as in the AMRF, control is separated from data. Control, in the form of commands and status, flows through the hierarchy, while data flows directly between data repositories as directed by the commands. User data areas (files and shared memory) mentioned in user commands are simply additional data repositories to and from which data can flow (Figure 5). The objective of this feature is to move data directly from producer to consumer, with as little overhead as possible.

The upper level of the IMDAS is the Distributed Data Server, or DDAS (Figure 6). The DDAS provides the data service interface to all application programs, performs the query processing and transaction management functions, and logically integrates the collection of data repositories into the global database. The DDAS query processor accepts transactions from the user programs and converts the strings into internal descriptions of the operations to be performed on the conceptual global database. Then the DDAS transaction manager determines the distribution of the conceptual operations over the actual databases, using its *fragmentation schema*, directs the execution of the transaction by the Basic Data Servers, and reports completion to the user program. The DDAS *manages* the data manipulations.

The Basic Data Servers, or BDAS (Figure 7) are the lower level of the IMDAS. The IMDAS operates on *existing* data repositories - databases, files, controller memories - managed by commercial DBMS, file systems, home-grown application-specific servers, etc. These repositories are "front-ended" by BDAS modules supporting an "interchange query form", which is a standardized representation of the operations to be performed, and an "interchange data form", which is a standardized representation of the modelled information units. The front-end modules, called "Command and Data Translators", or "CTs", translate both operations and data between the interchange forms and the local forms used by the particular database management system. In addition, a BDAS contains a Basic Service Executive, which provides the network interface to the DDAS and the network links to other BDASs. Each computer system in the IMDAS has its own Basic Data Server. The BDAS *executes* the data manipulations.

4. The AMRF IMDAS Configuration

IMDAS has been running as the integrating data management system in the AMRF since mid-1987. In the 1987 configuration (Figure 8), the IMDAS was centralized on a VAX-11/785 running VMS, and comprised exactly the VAX DDAS and a VAX BDAS with Command Translators for RTI Ingres, the AMRF Process Plan databases, the AMRF Geometry databases, the VMS file system, and the global common-memory [Mitc84].

The current (1989) configuration (Figure 9) contains a single DDAS, which may be run on either the VAX or on an AMRF Sun 3/260 workstation dedicated to IMDAS. In addition to the VAX BDAS, which is essentially unchanged, there may now be one more Sun BDASs, comprising CTs for Unix-based Ingres, the new AMRF Process Plan databases and the Unix file system(s).

Twelve AMRF control systems, including the Cell, and elements of Material Handling, the Horizontal Workstation, the Vertical Workstation, the Cleaning and Deburring Workstation and the Inspection Workstation, are potential clients of the IMDAS, via the AMRF network. (The actual running configuration changes frequently.)

5. Usage and Performance

From the point-of-view of operational characteristics, IMDAS transactions issued by AMRF controllers can be broken down into seven categories:

1) Insertion of new process plans, NC programs, etc. – creation of new entries in the form of large bodies of simply structured information. (AMRF process plans and control programs are currently stored as large bodies of text with descriptive headers.)

2) Retrieval of process plans, geometry data, NC programs, etc. – retrieval of large bodies of simply structured information by a few key values.

3) Retrieval of small simple relations of work-in-process data - tray definition, lot description, etc.

4) Update of simple relations of work-in-process data.

5) Update of lots and kitting orders by insertion of rows – updates with multiple information units obtained from user data areas.

6) Retrieval of small reports which require complex manipulation of the underlying databases – production of views very different from the storage organizations.

7) Update of complex relations requiring spawning of secondary update transactions. (The most common one is moving a tray out of an ASRS, which affects both the tray and the ASRS "shelf" status.)

Categories (2), (3) and (4) represent over 80% of the actual transactions issued by controllers during "production" runs of the AMRF. Categories (6) and (7) represent the remainder of the production load. The insertions in category (1) are uploads of engineering data developed on various systems and occur sporadically during production.

The 1987 performance of IMDAS was totally inadequate for supporting production manufacturing. Response times of 30 seconds or more for transactions in the first five categories were typical, and response times of several minutes for transactions in the latter two categories were not uncommon. This was determined to be partly a consequence of the quality of the prototype IMDAS implementation and partly a consequence of the total load on the VAX.

By 1989, IMDAS response times had improved considerably, using the (dedicated) Sun DDAS and distributing the workload. The 1989 response times are, by category, as follows:

1) Insertion of new process-plans, etc.: 5-6 seconds.

2) Retrieval of process-plans, etc.: 5 seconds.

3) Retrieval of simple relations : 5-10 seconds.

4) Update in simple relations: 5-10 seconds.

5) Update by insertion: 5-10 seconds, depending very little on number of new entries.

6) Retrieval of views requiring complex manipulations: 10-45 seconds, depending on how complex the actual operation is. These vary considerably.

7) Update which spawns secondary updates: 30-45 seconds.

The first five categories represent essentially simple database manipulations, even though some are retrievals and some are updates. The variation in performance depends largely on which actual database contains the information. By comparison, the latter two categories represent relatively complex data manipulations which may require several separate operations on a relational database and can therefore be expected to take longer. But the current IMDAS mapping from operations on the semantic model to operations on the relational tables, which involves the construction and maintenance of additional relations, greatly increases the number of relational operations which must be performed and thus the time required.

While these response times are marginally acceptable in the AMRF, it is clear that without considerable improvement in performance, IMDAS could not usefully be used in a production facility. Since IMDAS is a prototype, production quality performance has not thus far been a goal of the project. The question then is: To what extent can IMDAS performance be improved by improvements in the implementation rather than changes in the design?

We can break down the time consumed by IMDAS for a given transaction into four elements:

- propagation time — the time required to get the transaction from the application process (controller) to the IMDAS and through the IMDAS to the affected Command Translator modules (CTs), plus the time required to get the completion status from the CTs through the IMDAS hierarchy and back to the application.
- transaction analysis time — the time required by the DDAS to interpret the transaction string into the complete operation tree in "query interchange form" directed to the proper Command Translators. (This time is excluded from propagation time .)
- execution time — the time required by the CT and the underlying database management system to execute the transaction and convert the data to/from interchange form.
- editing time — the time required to convert the input or output data between the interchange form and the report form in the user data areas.

Table 1 gives a rough breakdown of transaction execution times into the four elements, according to the internal characteristics of the transaction.

From Table 1, it becomes apparent that half of the time expended on simple transactions is propagation time. While some propagation time is unavoidable, most of this time is a consequence of the mechanisms of communication used: the AMRF common-memory mechanisms and the "early OSI" AMRF network [Libe85, Rbyc88]. Improvement of the common-memory mechanism, or selection of an alternate internal communication mechanism, plus inevitable improvements in the performance of the networking software, should reduce this time considerably. On the other hand, propagation will always be a noticeable overhead in distributed systems. With the present hardware, reduction of this time below 200 msec is not to be expected.

The second largest time sink in simple transactions is transaction analysis time in the DDAS. This is to some extent a design consequence, in that runtime transaction analysis can be dramatically reduced in a system which preprocesses the transactions. On the other hand, this is the part of IMDAS which is still, for the most part, the original "student code". Several of the algorithms and

Timing Element	Simple Insertion or Retrieval	Complex transactions on one database	Transactions with multiple subparts
Propagation	2.5 sec	2.5 sec	up to 4 sec
Analysis	1.2 sec	1.5 sec	up to 2 sec
Execution	0.5-4.0 sec	10-40 sec	25-40 sec
Editing	< 0.5 sec	< 0.5 sec	< 0.5 sec

Table 1: Distribution of IMDAS Execution Times by Transaction Types

representations in this part of the code are known to be particularly inefficient, and are being replaced. But even then, the time required for an efficient execution of this critical task in the IMDAS could be a substantial fraction of a second.

Editing time is not a significant factor in the current transaction response times, and has been measured only very grossly. Unfortunately, this time probably cannot be improved much and consequently will become a noticeable factor in the performance of an efficient IMDAS.

This leaves us with "execution time". Simple transactions not involving Ingres, primarily geometry and process-plan retrievals, have transaction execution times substantially under 1 second, and are not currently instrumented to provide more exact information. Since both the geometry and process-planning databases in the AMRF are currently being revised in terms of both models and underlying data systems as a part of the Product Data Exchange Standardization (PDES) effort [PDES88], further analysis of those systems as currently built has little relevance to the future performance of IMDAS.

Simple transactions involving Ingres have transaction execution times in the 1-second range, caused by converting the nodes of the transaction tree to QUEL (the Ingres data manipulation language) statements one-at-a-time and transmitting them one-at-a-time to Ingres. With this interface, except possibly for some local optimization, this time may be difficult to improve on in a pure front-end. IMDAS response time suffers in this regard from the use of the now-outdated QUEL interface. By comparison, "decompiling" large segments of the transaction tree into a single SQL statement, for data systems which provide that interface (which will soon be all major relational systems), may produce significant performance improvement. Such a technique could reduce the actual CT-to-DBMS interfacing to a very few transactions and allow the DBMS itself to optimize the execution of the resulting "large" transactions as a whole.

Complex transactions have very long execution times in IMDAS. This is, for the most part, attributable to a design choice in the current IMDAS. The IMDAS internal view of the semantic network is neither relational nor object-oriented. Rather the DDAS threads the network to produce a *hierarchical* view relative to the transaction at hand, and this hierarchical view is imposed on the Command Translator. The Ingres CT faithfully maintains these hierarchies as it threads the nodes of the transaction tree. The resulting data mapping to the underlying relational databases is fairly simple, but the operation mapping is quite complex. Since the operations are performed one-at-a-time, this gives rise to elaborate unoptimized manipulations of the Ingres databases. NIST and the University of Florida are currently moving IMDAS toward the newer object-oriented OSAM* model [Su88] and a corresponding object-oriented view of transactions. This makes necessary somewhat more complex data mapping, which produces more complex transaction trees initially, but these trees can be optimized and the operation mapping for relational databases is much simpler. We expect this to result in significant reductions in overall processing time.

6. Summary

The first version of IMDAS has been run as the production data system in the AMRF for 18 months. What we have demonstrated is that the IMDAS design works and can support production manufacturing, but current IMDAS performance cannot. Much of the IMDAS performance weakness can be attributed to implementation choices rather than design choices, and can therefore be considerably improved by re-implementing certain elements with performance in mind. But there is one design characteristic of the current prototype which will prevent it from being a production-quality distributed data system. This characteristic – the "hierarchical view of data" – must be changed, as envisioned, to the "object-oriented view", for satisfactory performance to be achieved. Until these changes are made, IMDAS may be a successful feasibility demonstration, but it cannot be considered a production data system for the support of automated manufacturing.

Acknowledgements

The authors acknowledge a debt to Mary J. Mitchell, the principal information analyst for the AMRF, who provided the AMRF information models and the list of the floor transactions, and who, together with Ms. Lo, constructed the IMDAS performance test suites.

References

[Albu81] Albus, J.S., Barbera, A.J., Nagel, R.N., "Theory and Practice of Hierarchical Control", Proceedings of the 23rd International Conference of the IEEE Computer Society, September, 1981.

[ANSI86] American National Standards Institute, X3.135: "Database Language - SQL", New York, December, 1986.

[Barb82] Barbera, A.J., Fitzgerald, M.L., Albus, J.S., "Concepts for a Real-Time Sensory-Interactive Control System Architecture", Proceedings of the 14th Southeastern Symposium on System Theory, April, 1982.

[Bark86] Barkmeyer, E., Mitchell, M., Mikkilineni, K., Su, S.Y.W. and Lam, H., "An Architecture for an Integrated Manufacturing Data Administration System", NBSIR 863312, National Bureau of Standards, Gaithersburg, MD, January 1986.

[Brow86] Brown, P.F., and McLean, C.R., "Interactive Process Planning in the AMRF", Proceedings of the ASME Winter Annual Meeting, pp. 245-262, December, 1986.

[Kris87] Krishnamurthy, V., Su, S.Y.W., Lam, H., Mitchell, M., Barkmeyer, E., "A Distributed Database Architecture for an Integrated Manufacturing Facility", Proceedings of the International Conference on Data and Knowledge Systems for Manufacturing and Engineering, pp. 4-13, October 1987.

[Libe85] Libes, Don E., "User-Level Shared Variables", Proceedings of the Summer 1985 USENIX Conference, Portland, Oregon, June, 1985.

[Mitc84] Mitchell, M. and Barkmeyer, E., "Data Distribution in the NBS AMRF", Proceedings of the IPAD II Conference, Denver, CO, April, 1984.

[Nanz84] Nanzetta, P., "Update: NBS Research Facility Addresses Problems In Set-ups for Small

Batch Manufacturing," Industrial Engineering, pp 68-73, June 1984.

[Rybc88] Rybczynski, S.F., Barkmeyer, E., Wallace, E., Strawbridge, M., Libes, D. and Young, C. "AMRF Network Communications", NBSIR 88-3816, National Bureau of Standards, Gaithersburg, MD, June, 1988.

[PDES88] "Product Data Exchange Specification, First Working Draft", NISTIR 88-4004, National Institute of Standards and Technology, Gaithersburg, MD, December, 1988.

[Su83] Su, S.Y.W., "SAM*: A Semantic Association Model for Corporate and Scientific/Statistical Databases", Journal of Information Sciences #29, 1983, pp. 151-199.

[Su86a] Su, S.Y.W., Lam, H., Khatib, M., Krishnamurthy, V., Kumar, A., Malik, S., Mitchell, M., Barkmeyer, E., "The Architecture and Prototype Implementation of an Integrated Manufacturing Database Administration System", Spring COMPCON 1986.

[Su86b] Su, S.Y.W., "Modeling Integrated Manufacturing Data Using SAM*", Proceeding of GI-Conference on Database Systems for Office, Engineering and Science, Karlsruhe, Federal Republic of Germany, March 1985, reprinted in IEEE Computer, Vol. 19, No.1, January, 1986.

[Su88] Su, S.Y.W., Krishnamurthy, V., Lam, H., "An Object Oriented Semantic Association Model (OSAM*) ", in *AI in Industrial Engineering and Manufacturing: Theoretical Issues and Applications*, edited by Kashyap, R.L., Kumara, S., and Soyster, A.L., American Institute of Industrial Engineers, 1988.

[Tu87] Tu, J.S., and Hopp, T.H., "Part Geometry Data in the AMRF", NBSIR 87-3551, National Bureau of Standards, Gaithersburg, MD, April, 1987.