C. 4

SANDIA REPORT

SAND97–0048 • UC–310
Unlimited Release
Printed January 1997

Data Fusion for Adaptive Control in Manufacturing: Impact on Engineering Information Models

Olin H. Bray

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

SF2900Q(8-81)

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

Date: February 1, 1997

To: Distribution

From: Olin Bray, 4524

Subject: Final Report for FY93 LDRD
Information Integration for Data Fusion

The attached report is one of three reports that resulted from work done under the FY93 LDRD, *Information Integration for Data Fusion*. Copies are being sent to people who were involved in the project or who might be interested in its results. If you know of other people who would be interested in copies of these reports, please have them contact me or let me know and I will send them a copy.

Purpose of this LDRD:

Data fusion is the integration and analysis of data from multiple sensors to develop a more accurate understanding of a situation and determine how to respond to it. It can be applied in many application areas, several of which were explored in this LDRD project.

The *Information Integration for Data Fusion* LDRD project had two purposes: (1) to see if a natural language-based information modeling methodology could be used for data fusion problems, and if so, (2) to determine whether this methodology would help identify commonalities across areas and achieve greater synergy. Both of these hypotheses were confirmed. The project found five common objects that are the basis for all of the data fusion areas examined: targets, behaviors, environments, signatures, and sensors. Many of the specific facts related to these objects were common across several models and could easily be reused. In some cases, even the terminology remained the same. In other cases, different areas had their own terminology (e.g., a target in defense, a workpiece or machine tool in manufacturing, or an organ for health care), but the concepts were the same. This commonality is important with the growing use of multisensor data fusion. Data fusion is much more difficult if each type of sensor uses its own objects and models rather than building on a common set. Information model integration at the conceptual level is much easier than at the implementation level.

Report 1:

The first report, *Information Integration for Data Fusion* (SAND97-0195) provides a framework for considering data fusion from an information integration perspective, discusses how the synergy generated by this LDRD would have benefited an earlier successful project and contains a summary information model from that project, describes a preliminary truce management information model, and explains how information integration can facilitate cross-treaty synergy for various arms control treaties.

Report 2:

The second report, *Information Model for On-Site Inspection System* (SAND97-0049), describes the information model that was jointly developed as part of two LDRDs: (1) *Information Integration for Data Fusion,* and (2) *Interactive On-Site Inspection System: An Information System to Support Arms Control Inspections.* Section 1 describes the purpose and scope of the two LDRD projects and reviews the prototype development approach, including the use of a GIS. Section 2 describes the information modeling methodology. Section 3 provides a conceptual data dictionary for the OSIS (On-Site Inspection System) model, which can be used in conjunction with the detailed information model provided in the Appendix. Section 4 discussions the lessons learned from the modeling and the prototype. Section 5 identifies the next steps — two alternate paths for future development. The long-term purpose of the On-Site Inspection LDRD was to show the benefits of an information system to support a wide range of on-site inspection activities for both offensive and defensive inspections. The database structure and the information system would support inspection activities under nuclear, chemical, biological, and conventional arms control treaties. This would allow a common database to be shared for all types of inspections, providing much greater cross-treaty synergy. The details of the prototype are described in another Sandia report (SAND93-2300), *Interactive On-Site Inspection System: An Information System to Support Arms Control Inspections.*

Report 3:

The third report, *Data Fusion for Adaptive Control in Manufacturing: Impact on Engineering Information Models* (SAND97-0048), consists of four parts: Section 1 defines data fusion and explains its impact on manufacturing. Section 2 describes an information system architecture and explains the natural language-based information modeling methodology used by this research project. Section 3 identifies the major design and manufacturing functions, reviews the information models required to support them, and then shows how these models must be extended to support data fusion. Section 4 discusses the future directions of this work.

Outside Exposure:

This LDRD work also had exposure outside of Sandia. The first report provided the basis for a presentation, *Information Modeling Framework for Data Fusion Problems,* at the New Mexico DECUS Conference in Albuquerque, NM, in May of 1993. Part of the first report also provided the basis for a panel discussion at the DOE Expo 93 on Intelligence and Special Operations in Oak Ridge, TN. The third report was the basis for a paper, *Data Fusion for Adaptive Control in Manufacturing: Impact on Engineering Information Models,* for the ASME Engineering Information Management Symposium in San Diego in August 1993, which was reprinted in the ASME journal *Computers in Engineering.*

The work that resulted in the second report *(Information Model for On-Site Inspection System)* was done in conjunction with another LDRD that actually developed a prototype system based on the model, which was subsequently demonstrated to IAEA and other agencies. This system is now being shown at the Cooperative Monitoring Center.

SAND97-0048
Unlimited Release
Printed January 1997

Distribution
Category UC-310

Data Fusion for Adaptive Control in Manufacturing: Impact on Engineering Information Models

Olin H. Bray
Strategic Business Development
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1378

Abstract

Data fusion is the integration and analysis of data from multiple sensors to develop a more accurate understanding of a situation and determine how to respond to it. Although data fusion can be applied in many situations, this paper focuses on its application to manufacturing and how it changes some of the more traditional, less adaptive information models that support the design and manufacturing functions. The paper consists of four parts: Section 1 defines data fusion and explains its impact on manufacturing. Section 2 describes an information system architecture and explains the natural language-based information modeling methodology used by this research project. Section 3 identifies the major design and manufacturing functions, reviews the information models required to support them, and then shows how these models must be extended to support data fusion. Section 4 discusses the future directions of this work. This report is one of three produced by an FY93 LDRD project, Information Integration for Data Fusion. The project confirmed: (1) that the natural language-based information modeling methodology could be used effectively in data fusion areas, and (2) that commonalities could be found that would allow synergy across various data fusion areas, such as defense, manufacturing, and health

(Abstract continued to page ii)

(Abstract continued from title page)

care. The project found five common objects that are the basis for all of the data fusion areas examined: targets, behaviors, environments, signatures, and sensors. Many of these objects and the specific facts related to them were common across several models and could easily be reused. In some cases, even the terminology remained the same. In other cases, different areas had their own terminology (e.g., a target for defense, a workpiece or machine tool for manufacturing, or an organ for health care), but the concepts were the same. This commonality is important with the growing use of multisensor data fusion. Data fusion is much more difficult if each type of sensor uses its own objects and models rather than building on a common set. Information model integration at the conceptual level is much easier than at the implementation level. Another Sandia report (SAND97-0195), *Information Integration for Data Fusion,* provides a more detailed data fusion framework and addresses this commonality more specifically.

Contents

Figures

Intentionally Left Blank

Data Fusion for Adaptive Control in Manufacturing: Impact on Engineering Information Models

Introduction

This paper addresses the impact of data fusion on engineering and manufacturing information systems. Data fusion is the integration and analysis of data from multiple sensors to develop a more accurate understanding of a situation and determine how to respond to it. Although data fusion can be applied in many situations, this paper focuses on its application to manufacturing and how it changes some of the more traditional, less adaptive information models that support the design and manufacturing functions.

Data fusion requires changes, primarily extensions, to these traditional information models. Engineering models normally address geometry, features, and performance characteristics of a part, while manufacturing models address machine tool characteristics and how they interact with the workpiece and operations scheduling. For data fusion and adaptive control, these models must include the dynamic behavior and interactions between the workpiece and the machine tool. In addition, these models must specify how these dynamic behaviors can be seen and interpreted by various types of sensors (e.g., temperature, pressure, and vibration). On a broader level, the models must also consider how the machine tools and robots interact within a work cell or production line.

The paper consists of four parts:

- The first section defines data fusion and its impact on manufacturing.
- The second section describes an information system architecture and explains the natural language based information modeling methodology used by this research project.
- The third section identifies the major design and manufacturing functions, reviews the information models required to support them, and then shows how these models must be extended to support data fusion.
- The fourth section discusses the future directions of this work.

Intentionally Left Blank

1. Data Fusion

There are three major reasons for manufacturing interest in sensors and data fusion:

- First, sensors are critical to monitor the manufacturing process to allow for better and more consistent product quality, especially for flexible facilities with short production runs.

- Second, by detecting and correcting problems before tolerance limits are exceeded, scrap, rework, wasted effort, and production delays can be reduced or avoided.

- Third, early detection and avoidance of serious problems prevents damage to workpieces and machine tools and prevents serious personal injury. Examples of problems that could be detected early include misaligned equipment, worn or broken tooling, a wrong or mispositioned or misoriented workpiece, and poor or inconsistent workpiece material quality.

What is Data Fusion?

Data fusion involves the integration or fusion of data from multiple sensors. These sensors may be of the same or different types, such as pressure, temperature, infrared, laser, radar, or acoustic. They may be at the same or different locations. The inputs may occur at the same or different times. The key point is that one is trying to analyze and identify a specific situation and each sensor is providing additional data to be used in the analysis. These sensors may be treated as a collection of individual sensors or as an integrated sensor array.

The reason for using multiple sensors is to improve the analysis of the situation. This improvement may involve a higher-quality (i.e., more accurate and reliable) understanding of the situation or it may simply involve a more efficient or faster analysis.

Data fusion is a specific type of approach that can be applied in many distinct domains. The defense domain, where much of the data fusion work started, includes many applications such as battlefield monitoring and strategic reconnaissance and surveillance. The Defense Department considers data fusion one of its critical technologies. In manufacturing, data fusion is essential to determine and control what is happening on the factory floor. Data fusion can also be a critical component in environmental monitoring and health care. Therefore, while this paper focuses on applying data fusion to manufacturing, the same approach is applicable to all of these areas. This means that there are many opportunities for technology transfer among these areas. In fact, the specific purpose of this research project is to test and validate a common information modeling approach that can support all of these areas and to show there is some synergy among the areas.

In all of these areas, data fusion consists of the same four steps:

- First, there is the input and analysis of raw data from a sensor to identify something and its characteristics — the position, temperature, or pressure of a cutting tool at a workpiece.

- Second, there is the analysis and integration of data from multiple sensors (e.g., infrared, laser, pressure, or temperature sensors) to determine the situation.

- Third, there is an analysis of the situation to identify any problems or threats. For example, high-pressure and vibration could create a threat that the cutting tool is about to break.

- Fourth, the system must respond appropriately to the developing situation to eliminate or repair a problem.

Some researchers adopt a more limited definition of data fusion and include only steps one and two. However, to understand and model the data fusion process and information requirements, all four steps are essential. This four-step data fusion process is essential to flexible manufacturing and adaptive control. The signal conditioning part of the first step is almost always done at the sensor. The higher-level steps are usually done by a more centralized processor, although there is a trend toward distributing more processing downward to more intelligent sensors, which may do their own calibration, performance monitoring, and diagnostics.

Types of Data Fusion Problems

Some researchers have classified data fusion into three broad categories or worlds: the designed world, the real world, and the hostile world.

The designed world applications include machine tool, production line, and factory control. The real world applications usually involve monitoring broad environmental areas such as air and/or water from metropolitan or regional areas. The hostile world applications involve battlefield monitoring and strategic surveillance and reconnaissance, where an opponent is actively trying to deceive you about his actions and intentions.

The original assumption was that these three broad categories differ with respect to a number of dimensions (e.g., degree of control of the environment, information complexity, and reliability). However, on closer examination, there is considerable overlap on most dimensions. Obviously, in a hostile world, a failure may mean the destruction of a vehicle and its crew. However, some failures in a chemical plant can result in major destruction and personal injury or death.

Similarly, with information complexity, one can select applications from each category that have comparable levels of complexity. Therefore, the main difference among the three categories is in the level of control of the environment:

- In a **designed world** such as a factory, the goal is the complete control of all of the major processes.
- In the **real world** you are monitoring a very complex set of variables over which you have relatively little or no control. You may have control over the level of pollution discharge, but you have no control over the weather system into which it is released and how that system spreads the discharge.
- Finally, in the **hostile world** you are essentially playing a game with and responding to actions and reactions by an opponent. You control some of the variables, your opponent controls other variables, and neither of you can control other variables such as the weather.

Although there seems to be considerable technical overlap among these categories of data fusion, in practice there is little cross-fertilization because the experts in each area have very little contact with each other.

How is data fusion different in these three cases? The same four steps described above apply in all three cases. The basic information modeling approach and methodology are still the same. One difference is in the content of the information models. Another difference is that in a hostile world, the number of situations can be much larger because of deception, and the analysis and discrimination processes must be more rigorous.

Data Fusion Dimensions

There are several dimensions along which a data fusion problem can be classified. These dimensions (except for the "control of the environment" dimension) are independent of the three categories (i.e., designed, real, and hostile) since applications in each category can appear anywhere along any of these dimensions. However, certain approaches or solutions are determined by the emphasis the applications place on one or more dimensions.

These dimensions include:

- Control of the Environment
- Information Complexity

- Information Volume
- Timeliness/Response Time
- Reliability and Maintainability

The rest of this section provides a brief explanation of each of these dimensions and their significance for data fusion.

Control of the Environment

This dimension is probably the key determinant for the data fusion category as designed, real, or hostile world. A designed world (e.g., manufacturing) is one in which you understand and have almost complete control over the processes of interest, for example, a machine tool, work cell, production line, or factory. Also in a designed world you can identify, and if necessary monitor, all or most of the variables of interest.

On the other hand, in the real world you have much less control. You have direct control over very few factors and some limited influence over a few more, but you can only monitor, not control or influence, most factors.

In the hostile world, you have direct control over some factors and no control over others. However, with this latter set of factors some are controlled by your opponent while neither of you can control other factors, i.e., true real world factors.

In summary, the degree of control of the environment is perhaps the key determining factor for placing a data fusion application in one of the three categories.

Information Complexity

An application's placement along this dimension is determined by the number of entity types and the number of the relationships among them — or in a natural language context (described below) by the number of fact types in the information model. For example, the extended information model for data fusion in Section 3 is more complex than the non-data fusion model because it has more entity types (i.e., measurement type and instance and sensor type and instance) and some additional facts (designated by asterisks) for the existing entity types. (Note that each of these entity types may generate several tables in a normalized relational database design.)

Information complexity is a useful metric for estimating the resources and schedule to develop, document, and maintain a system. However, it is not a reliable indicator for the category of data fusion. The information complexity of a fully automated factory may be comparable to some air defense data fusion applications. Realize that information complexity implies nothing about the data volume, functional complexity, or processing load on the system.

Information Volume

This dimension involves the volume of data generated per unit of time. A more limited, related factor is how much of this data must be archived and stored for future use or as a permanent record. Since many data fusion applications involve real time control (for a factory or a weapon system), much of the data is used immediately and can be discarded when the next cycle of updated data is obtained. However, there are two exceptions:

- First, some processes need data from the previous N cycles, but still only a narrow window of data needs to be saved.

- Second, for some mission-critical parts and products, all of this data may be archived for future quality audits or analysis if a problem occurs, such as a part failure in the field. This means that the volume of archived data is normally much lower than the volume of raw data passing through the system. (Environmental monitoring is an exception in that data are usually collected and stored for later processing.)

The volume of data passing through the system is related to the number of measurements being taken and their frequency, although all of the variables may not be sampled at the same rate. This data volume affects the processing load on the system and is a factor in determining the response time of the system. There is a significant overlap in the information volume for data fusion application in the three categories.

Timeliness/Response Time

The timeliness and response time requirement depends on the specific application. In general, the closer the process is to an actual sensor, the faster the response time requirement. However, this is usually ultimately driven by the application requirement. If the process being monitored and/or controlled is very fast, then a fast sensor is needed.

The application response time includes both the sensor (or sensors) response time and any additional communications and processing time needed to move the data and do any necessary calculations on the data sent directly by the sensor (or sensors). Many applications in each category are dealing with many types of events, each with different response time requirements. For example, in a factory, millisecond response time may be needed to control a machining operation, second response time for robots loading and unloading a machine tool, and minute response time for scheduling the material handling system to move material through the factory — although the actual control of an autonomous vehicle moving materials through a factory would need a much faster response time.

Reliability and Maintainability

Reliability and maintainability are two related dimensions along which data fusion applications can be placed. Reliability relates to how robust the system and its sensors must be to continue operating. The greater the reliability requirement, the more fault tolerance and redundancy needs to be built into the system. A key metric for reliability is the mean time between failure (MTBF).

Maintainability partly refers to what happens once the system has failed — i.e., the mean time to repair (MTTR). If the application requires high reliability and rapid response, then the MTTR should be very low. Also the system may need extensive diagnostics to detect degrading components before they actually fail and bring the system down. In general, the reliability and maintainability issues for a data fusion application are similar to those for any real time control system with comparable requirements.

Manufacturing Applications of Data Fusion

In manufacturing, sensors can be used in several ways. At one extreme a sensor may simply take a measurement and pass it to an operator for action. At the other extreme a sensor may be part of an automated system that takes a set of measurements, determines what is happening, and takes the appropriate corrective action. However, at this extreme the sensor is only one part of the fully automated system. Obviously, there is a range of intermediate possibilities.

It is important to clearly distinguish between what a sensor measures and the characteristics that are important for decision making and control. Sometimes they are the same, but often the characteristic or decision variable, which is the actual user or application requirement, must be inferred or derived from one or more sensor measurements. For example, a sensor may measure force, strain, resistance, sound, or voltage. However, the actual characteristic of interest may be cutting force, position, movement, temperature, surface finish, material uniformity, or probability of tool breakage.

A range of processing is necessary between the point where the sensor actually gets its raw data and where processed information is used by the application to make a control decision. This processing can be done at one or more of several places. The sensor may do some signal processing and pass a more reliable reading to the next component in the automated system. A sensor may include the capability for self-diagnostics and/or calibration and/or correction to improve its reliability. As more of this capability is loaded onto the sensor, it is considered more intelligent.

There is a clear trend toward the increased use of intelligent sensors because their improved measurement quality and reliability outweigh their added cost. In fact, this is simply the sensor equivalent of a general information systems trend toward distributed processing. Initially, dumb terminals were connected to mainframe computers that did all of the processing. Now with distributed processing, more and more processing functions are distributed to increasingly powerful personal computers and workstations, with the mainframes or larger processors being reserved as special servers for data management, integration, and/or number crunching.

The following list indicates the types of processing that must be done:

1. Make measurement and pass raw data on to an operator or another subsystem.

2. Make measurement and do initial signal processing on raw data.

3. Do complete processing and pass on application-specific data.

4. Do self-calibration.

5. Do error detection.

6. Do error diagnosis.

7. Do error correction.

8. If a sensor array is involved, do sensor fusion (integrate signal data from multiple sensors and pass on more reliable signal data).

9. If a sensor array is involved, do data fusion (integrate application data from multiple sensors and pass on more reliable application-oriented data, i. e., analysis of situation).

Intentionally Left Blank

2. Information System Architecture and Methodology

This section describes an approach to information systems architecture and how data fusion fits within that architecture. It then describes a natural language based information modeling approach that allows a user or group of users to precisely define their information requirements. In Section 3, this modeling approach is used to define the information model for several manufacturing functions and how they must be extended for data fusion.

Information Systems Architecture

Zachman [9] has proposed a framework for information system architectures. In this paper, a simplified framework will be used to explain the concepts and the place of information modeling in the overall framework.

In this simplified architecture (Figure 1) there are three architectural views — user, designer, and implementer. There are two "components" — function and data. Different architectural views involve different concepts and different representations. Zachman uses the example of a house. What the buyer wants to see is different than what the contractor wants and needs, which in turn is different from what the plumber or electrician need. Everyone is talking about the same house, but their needs — and therefore their architectural views — are different.

View	Component	
	Function	Data
User		
Designer		
Implementer		

Figure 1. Simplified information architecture

Some people confuse "level of detail" with "architectural views." However, different levels of detail exist within, not across, architectural views. A high-level electrical view may show only the major wiring conduits, while a more detailed wiring diagram may show the specific placement for all of the switches and lights. However, none of these electrical views are more or less detailed versions of the floor plan or of the buyer's model.

How do these concepts apply to an information system architecture? The three architectural views — user, designer, and implementer — involve three distinct sets of concepts and levels of detail, which apply within, not across, these views.

- The **user** is interested in a set of business or manufacturing functions, which may be defined in greater or lesser detail.

- The **designer** uses a set of information system concepts to create a design that will support the user in performing the user functions that have been specified.

- Finally, the **implementer** uses programming language constructs to create one or more programs that implement the design. Neither the designer nor the user are interested in the implementer's programming language concepts such as GOTO, Do While, or Do Until.

A similar case can be made for the data component. The user wants to specify his or her information requirements in familiar terms. The designer takes these requirements and creates a logical design, i.e. a data model. Finally, the implementer takes the data model and generates the appropriate schema in terms of a specific database management system's (DBMS) Data Definition Language. Unfortunately, the data process often breaks down in the first step. The user is asked what objects he is interested in and what are their attributes, and finally what are the appropriate constraints. In other words, the user is supposed to provide a high-level data model, which the designer will refine with more detail. (Even Zachman's framework falls into this trap by asking the user for the business entities and attributes, which are then mapped in more detail into the system or data entities and attributes.) The data model, at whatever level of detail, belongs in the designer's view, not the user's view.

The obvious question is what is the user's view of the data. How do users communicate with each other when they discuss a problem or exchange information? They use sentences in a natural language. If the problem is too complicated, they may use figures and equations, but they still talk about them in sentences. Therefore, the user's view of the data should be based on natural language. This natural language based information model should be the starting point of the data component, i.e. the user's view of the data. The next section, "Information Modeling Methodology," introduces the concepts and the way to build and validate an information model working with one or more users. Questions and ambiguities are easily resolved with examples. Although this approach has existed for more than a decade, it is now becoming a major focus for both U.S. and international standards work. This methodology is described in more detail in the documents referred to in References 1, 4, 5, 7, and 8.

A major benefit of this approach is its formal, algorithmic nature. Once an information model has been developed and validated with the user, there is an algorithm (implemented in several CASE tools) that will automatically generate a fifth normal form neutral data model, which can be implemented in any DBMS. This neutral data model specifies the tables, the keys and attributes for each table, and which attributes provide foreign keys to which other tables. Given this neutral data model, some of the CASE tools go a step further and provide the actual schema for several specific DBMSs.

There is a clear analogy between this mapping process and the application process that takes a high-level language input and generates the necessary machine language code to process the request. Extending the analogy, when you want to change the application, you go back to the high-level language version, make the change there, and recompile the lower-level machine code. You do not patch the actual machine code. Using the analogy, the schema corresponds to the machine code, while the information model is the high-level specification. Therefore, changes should be made to the information model with the system generating the appropriate revised data model and schema.

How does this information architecture apply to data fusion? First, consider the function column. The user view is the application requirements, i.e., the process that needs to be controlled and its timing and reliability characteristics. The designer's view involves sensor fusion and intelligent sensors. The implementer's functional view involves sensors and sensor arrays. On the data side, the user's view is the complete application information model, which includes the data fusion information model that is needed to support the user's functional requirements. The designer's data view is the more traditional data model derived from this information model. Finally, the implementer's data view is the database definition, just like with the general information architecture.

Information Modeling Methodology

This section provides an introduction to the concepts used in natural language based information modeling. The information model can be represented in either of two ways — verbally or graphically.

- The **verbal representation** can be read, critiqued, and corrected by anyone who knows the subject matter, with virtually no explanation of the methodology.

- The **graphical representation** shows the relationships among the entity types more clearly and concisely, but it does require a few minutes of explanation to be understood. After reading this section, a person should be able to read and understand, although not construct, most of a graphical representation of an information model.

Concepts Covered in this Section:

- sentence
- elementary sentence
- fact
  - fact type
  - fact instance
- entity
  - entity type
  - entity instance
- label type
- role/verb
- constraints
  - total
  - uniqueness

Verbal Representation of Information Model

A **sentence** is simply a natural language statement by a user describing some aspect of the problem area. It may be simple, describing a specific example or instance — "Part X weighs 10 pounds," — or complex — "Part X, which was designed by John Smith last year, now sells for $100 and comes in red, green, and blue."

Any complex sentence can be decomposed into **elementary sentences**: "Part X was designed by John Smith." "Part X was designed in 1994." "Part X in 1995 sells for $100." Note that in this case, an elementary sentence is not binary: "Part X in 1995" does not provide the price. "Part X sells for $100" does not specify when, but the price may change over time. The other binary alternative — "1995 sells for $100" — has even more problems. The above sentences were all examples of specific instances, but they could equally well have been done in terms of types: "A part is designed by an engineer." "A part sells for a price in a year." "A part has a color."

The initial problem statement from the user is often a narrative consisting of simple and complex sentences referring to both types and instances. The information modeling methodology provides a way to decompose the problem statement into elementary sentences and formally model them to unambiguously identify all of the relationships and constraints in a way that the user can review them to verify or correct them.

The user describes the problem in natural language sentences: some are already elementary sentences while others are complex sentences. The complex sentences are decomposed into the corresponding elementary sentences. Many elementary sentences are binary, but they do not have to be. The key criterion is that an elementary sentence cannot be decomposed into more basic sentences without losing information, as shown in the previous part, date, and price example.

A structured sentence, sometimes called a **fact**, has a very specific form. It consists of two **entity types** (such as person, part, or department) that are related by a **role**, usually a **verb phrase** (such as designs, works in, or is responsible for). Examples of fact types include the following:

- A person designs a part.
- A person works in a department.
- A department employs a person (the inverse of the previous fact).

For each fact type there can be many **fact instances**, such as "Bill designed part 1234," or "Sam works in Engineering."

To completely capture all of the required information, a deep structure sentence or fact has a specific form. It specifies the first entity type, its identifier or **label type**, several examples or **instances** of that label type, a verb phrase, and another entity's set of information (i.e., entity type, label type, and label instance). Although label instances are sometimes called examples (in the sense that they are examples of entity types), the information modeling methodology really requires examples of facts or fact instances.

Entity type:

Person

Label type:

SSN

Label instance:

123-45-6789

Verb:

works in

Entity type:

Department

Label type:

Department name

Label instance:

Engineering

Fact instances or examples are critical because they explicitly define the data constraints, which the DBMS must enforce. Let's explain the **constraint types** using specific examples for the fact pair: "A person designs a part" and "A part is designed by a person."

The **total constraint** tells whether or not every entity instance of a specific type must participate in the fact type. Must every person design a part? No, so the first fact/role is not total. However, must every part be designed by a person? If we assume the answer is yes, then this fact/role is total. If we know anything about a part instance, we must know the person who designed it. (Database experts will recognize that this is a mandatory attribute for an entity; but the user has remained insulated from that designer view.)

In an actual modeling session, someone may raise the issue that we buy some parts from suppliers and for those parts the designer is unknown and probably irrelevant. In other words, for some parts, one set of facts apply, while for other parts, a different set of facts may apply, although all parts share a common set of facts. This distinction defines the subtype-supertype relationship. The supertype (part) has a set of facts that are common to all of the subtypes (designed part and purchased part). The subtypes are distinguished from each other by the unique set of facts that apply to each subtype. All of the common facts (except the identifier) are removed from the subtypes and are associated with the supertype.

To determine another important constraint — the **uniqueness constraint** — requires an additional example. Consider the following examples for the fact "a person works in a department."

	Person	Department
1.	Sam	Eng
2.	Mary	Mfg
3.	Bill	Eng
4.	Sam	Finance
5.	Joe	—
6.	—	Accounting

When shown the previous six examples, the user can quickly determine which ones are good:

- Examples 1 and 2 are good because there is no overlap; they are two independent fact instances.
- Example 3 is good because a department (Eng) can have more than one person in it.
- Example 4 is incorrect, however, because a person (Sam) can only be in one department. This defines a uniqueness constraint, an specific instance of a person can only appear once in this fact type.
- Examples 5 and 6 simply document the totality constraints described above. Example 5 is incorrect because every person must be in a department, i.e., the total constraint. Example 6 simply verifies that departments may be created and other data collected about them before people are actually assigned to them. However, this is only a business rule constraint, not a physical constraint, so another company could decide that they wanted to consider example 6 as incorrect.

After analyzing these examples, a more precise statement of the facts is possible. These were the initial facts:

- A person works in a department.
- A department employs a person.

Considering the examples, the more precise facts are as follows:

- Every person must work in one department.
- A department may employ one or more people. (Note: The zero people case is implied by the "may" in this example.)

The possible uniqueness constraints are that the object on the left may be unique, the object on the right may be unique, each object may be unique, or the combination may be unique. An example of fact with the combination is "a person is assigned to a project." A person can be assigned to many projects and a project can have many people, but you would not assign Sam to project X twice.

Graphical Representation of Information Model

For completeness, the rest of this section briefly describes the graphical representation of the information model. In Figure 2, the solid circle represents an object or concept in the real world, such as a person, a department, a part, or a release status. Dashed or dotted circles represent data objects that identify or further describe real objects, such as employee name, social security number, or release code. Boxes or rectangles represent the roles played by one object type with respect to another. The two boxes together indicate that two roles are complementary — a person works in a department and a department employs a person. With appropriate naming, facts in graphic model can be read as sentences.

Figure 2 shows several basic facts (in both directions) and their constraints. The facts shown include: "a person is identified by a SSN," "a person works in a department," "a person designs a part," "a person is assigned to a project," and "a part has a current release status." (Note: The model must specify "current" release status because a part will have many release statuses over time.) The constraints are also shown. The V indicates a total constraint and the line over a role indicates uniqueness. Obviously there are additional constraint types and symbols, but this should provide the reader with a general understanding of the graphic model representation. The neutral data model that can be generated from the information model (in either its verbal or its graphical representation) can be represented in any of the traditional data modeling notations.

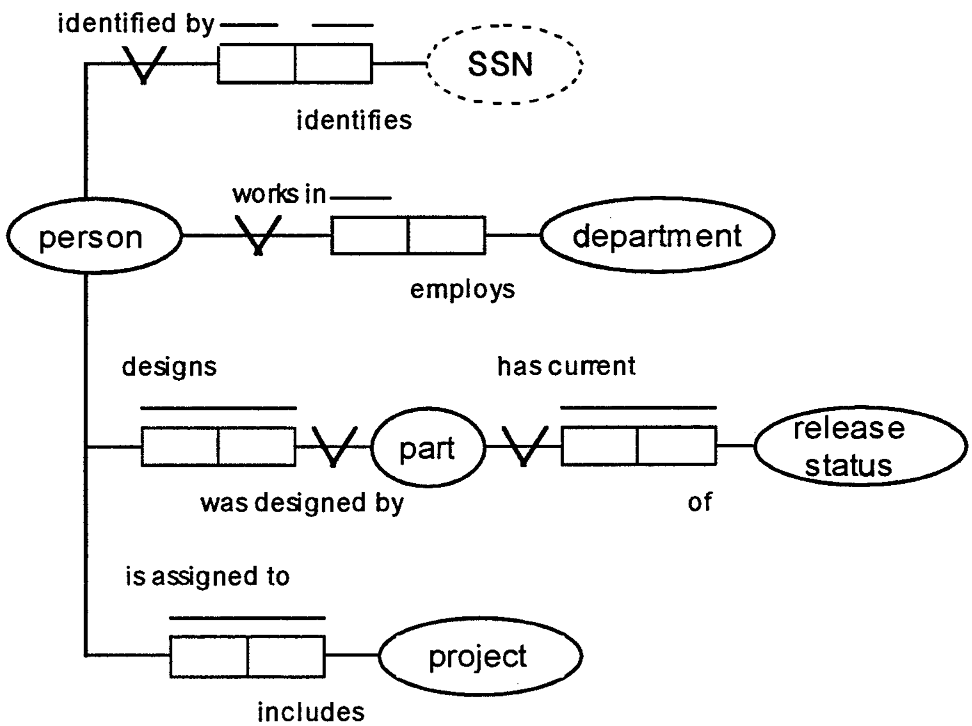

Figure 2. Example of graphic representation of information model

Intentionally Left Blank

3. Manufacturing Functions

The Manufacturing Process

At the highest level, manufacturing consists of four functions:

- Design the product.
- Design the manufacturing process.
- Schedule the production.
- Manufacture the product.

Since this paper focuses on the role of sensors and data fusion in manufacturing, we need to focus on two of the four functions — the second and fourth ones.

In **designing the manufacturing process**, the manufacturing engineer must determine the raw material or starting point, the operations that need to be performed, the sequence of operations, and the types of machine tools and tooling. For each operation the target feeds, the speeds of the machining operation and other machine tool control parameters need to be specified.

During the actual **manufacturing process**, the operator or the NC program tries to control the machine tool using these parameters. However, there is little or no direct feedback from real time sensors to control the actual process. This approach is not uncommon or ineffective. Today, many work cells and even flexible manufacturing systems are still run this way. The difficulty is that if a problem or an abnormality occurs, human intervention is often required. However, with better real time feedback from one or more sensors, many of these problems can be automatically detected, diagnosed, and corrected, or in some cases, anticipated and avoided.

To do this automatically, the information models to support the process planning and the manufacturing operations must be extended. The next section shows the main facts for the traditional information model for manufacturing and the extensions, flagged with asterisks (*). In general, there are several basic types of extensions. The process plan must specify, in observable terms, the desired behavior expected during each operation, the tolerance limits of that behavior, and the actions to be taken when those tolerance limits are exceeded or approached. The observed behavior may or may not be what is directly controlled.

For example, during the process planning, the manufacturing engineer may specify a target feed and speed for a machining operation, given the workpiece and cutting tool material, the power of the machine tool, and the amount of material to remove. However, he may also specify certain temperature and strain levels that should not be exceeded, perhaps because of their effect on the workpiece or feature quality or the risk of tool breakage. Furthermore, he needs to specify the appropriate control actions: For example, if the temperature is approaching its limit, but the strain is acceptable, then increase the coolant flow. However, if the strain is approaching its limit, reduce the feed rate regardless of the temperature.

In the above case, the symptoms and control actions are based directly on observable behavior, but data from two types of sensors must be used. In other cases, the problem may not be directly measurable. For example, if the cutting tool is about to break, then reduce the feed rate. Unfortunately, the probability of tool breakage is not directly observable. However, the probability may be related to a combination of strain and vibration, where vibration may be measured by rapid changes in position of the workpiece or by rapid changes in the strain. This is a clear case of the user specifying requirements using one set of concepts and the designer choosing to satisfy them with one of several alternatives, but again by fusing or integrating data from multiple sensors, or possibly multiple data over time from the same sensor.

The above extensions must be reflected in the information model to support process planning. Similar extensions are needed to actually control the manufacturing operations. The actual measurements must be made by the appropriate sensors — related to each other and to the manufacturing operation, and if necessary, related to specific workpiece and part instances — and stored for a quality audit.

As an area becomes better understood, the information model frequently evolves in the direction of greater specificity. For example, the following information model includes measurement type and measurement instance, where measurement type has a name and measurement instance has a measurement type, value, and unit of measure. This provides a common, initial structure for dealing with any type of measurement at any point within the model. Over time, some information models evolve toward greater specificity. The generic measurement type would be replaced by specific measurements, such as temperature, pressure, or length. Both versions of an information model are equally correct. The more specific version requires a greater understanding of the application area, but it often provides better performance. On the other hand, the more general version provides greater flexibility and fewer maintenance problems. The decision is simply a trade-off.

Information Model for Manufacturing

This section lists the facts for a basic information model, with the data fusion extensions flagged with asterisks (*). The types of entities are listed alphabetically, except where it makes more sense to describe a type of entity before considering an instance of the entity. (Note: Although this information model is still being refined, this latest version is a good example of the methodology and direction of the research.)

Facts for the Information Model

Characteristic Type

A characteristic type must be identified by a characteristic id.

A characteristic type must be called by a characteristic name.

A characteristic type must be described by a characteristic description.

A characteristic type must have a standard units of measure.

*A characteristic type may be determined by a measurement type.

*A characteristic type may be determined by one or more measurement types.

*A characteristic may be determined by a calculation process.

Characteristic Instance

A characteristic instance must be identified by a characteristic instance id. (characteristic instance identifier = characteristic type + object instance + date/time stamp)
(an object instance may be a workpiece, part, or process)

A characteristic instance must be of one characteristic type.

A characteristic instance must be derived from one or more measurement instances.

Cutting Tool Type

A cutting tool type must be identified by a cutting tool type id.

A cutting tool type must be made of a material.

A cutting tool type may be useable on one or more machine tool types.

A cutting tool type may be called out by one or more manufacturing operation types.

A cutting tool type must have a geometry.

A cutting tool type must have tolerance limits.

Cutting Tool Instance

A cutting tool instance must be identified by a cutting tool serial number.

A cutting tool instance may be used in one or more manufacturing operation instances.

A cutting tool instance must have an actual geometry.

A cutting tool instance may have an actual usage period in hours.

Machine Tool Type

A machine tool type must be identified by a machine tool type id.

A machine tool type must have a maximum workpiece weight.

A machine tool type must have a maximum workpiece size.

A machine tool type must have a maximum feed rate.

A machine tool type must have tolerance limits.

A machine tool type must be covered by a maintenance schedule.

A machine tool type may be specified by one or more manufacturing operation types.

A machine tool type may use one or more NC programs.

Machine Tool Instance

A machine tool instance must be identified by a machine tool serial number.

A machine tool instance must be of a machine tool type.

A machine tool instance must be located at a position.

A machine tool instance may be in a work cell.

A machine tool instance may be used in one or more manufacturing operation instances.

*A machine tool instance may have attached one or more sensor instances.

A machine tool instance must have a cumulative hours.

A machine tool instance must have an hours since last maintenance.

A machine tool instance must have an actual maintenance schedule.

A machine tool instance may have one or more maintenance actions.

Manufacturing Operation Type

A manufacturing operation type must be identified by a manufacturing operation type id. (manufacturing operation type id = process plan + sequence number)

A manufacturing operation type must be included in one or more process plans.

A manufacturing operation type may involve a machine tool type.

A manufacturing operation type may involve a cutting tool type.

A manufacturing operation type may specify a maximum measurement instance value.

A manufacturing operation type may specify a minimum measurement instance value.

Manufacturing Operation Instance

A manufacturing operation instance must be identified by a manufacturing operation instance id (manufacturing operation instance id = routing + sequence number).

A manufacturing operation instance must be specified in a routing/traveler.

A manufacturing operation instance may be done on a machine tool instance.

A manufacturing operation instance may be done with a cutting tool instance.

A manufacturing operation instance may be monitored by one or more sensor instances.

A manufacturing operation instance may be the target of one or more measurement instances.

Material

A material must be identified by a material id.

A material must be described by a material description.

A material must have one or more characteristic types.

A material may be used in a workpiece.

A material may be used in a part type.

A material may be used in a cutting tool.

*Measurement Type

A measurement type must be identified by a measurement type id.

A measurement type must have a measurement type name.

A measurement type may be done using one or more sensor types.

A measurement type may be used to derive one or more characteristic types.

*Measurement Instance

A measurement instance must be identified by a measurement instance id (measurement instance id = sensor instance + date/time stamp).

A measurement instance is of a measurement type.

A measurement instance must have a value.

A measurement instance must have a unit of measure.

A measurement instance must be taken using a sensor instance.

A measurement instance must be taken at a date/time.

A measurement instance may be made under one or more environmental conditions (such as temperature, pressure, atmosphere).

A measurement instance must be take of an object instance (an object may be a workpiece, a part instance, or a manufacturing operation instance).

Part Type

A part type must be identified by a part id.

A part type must be made according to one or more process plans.

A part type must be composed of one or more materials.

A part type may include one or more features.

A part type must include one or more geometries.

A part type may be used in one or more part types.

A part type may include one or more part types.

Part Instance

A part instance must identified by a part id (part id = part type + part serial number).

A part instance is of type part type.

A part instance was made under a manufacturing work order.

*A part instance may be the target of one or more measurement instances.

*A part instance is documented in a part process record. (A part process record is the measurement instances for the part.)

Process Plan

A process plan must be identified by a process plan id.

A process plan must be used to make a part type.

A process plan must include one or more manufacturing operation types.

*Sensor Type

A sensor type must be identified by a sensor type code.

A sensor type can make one or more measurement types.

*Sensor Instance

A sensor instance must be identified by a sensor id.

A sensor instance must be of a sensor type.

A sensor instance may currently be attached to a machine tool instance.

A sensor instance may have been attached to one or more machine tool instances.

A sensor instance may be used to make one or more measurement instances.

A sensor instance must be made by a manufacturer.

A sensor instance has a maintenance record.

Work Cell Instance

A work cell instance must be identified by a work cell id.

A work cell instance must be located at a location.

A work cell instance must include one or more machine tool instances.

Workpiece

A workpiece must be identified by a workpiece id.

A workpiece must be made from one or more materials.

A workpiece may be used to make one or more part types.

*A workpiece may be the target for one or more measurement instances.

4. Conclusions and Future Directions

This research project, some of the results of which were discussed in this paper, has identified and tested a common natural language based information modeling approach that is applicable for all three categories of data fusion problems. The generic manufacturing information model was described in this paper. Comparable models have been developed for other categories of the data fusion problem. The methodology is applicable for all three categories. Many of the specific entity types and facts can be reused by several areas.

The common entity types are target, behavior, environment, signature, and sensor. Most of the explicit reuse involves characteristics, sensors, and measurements. Conceptually, there is even greater commonality because many of the concepts are the same, although different disciplines use different terms. For example, in the defense area the object being sensed is called a target, while in manufacturing it may be any number of items, such as a part, a workpiece, a cutting tool, or a robot gripper.

In the manufacturing area, the next step is to apply this generic manufacturing information model to several specific manufacturing processes.

Intentionally Left Blank

5. References

1. Olin H. Bray, *Computer Integrated Manufacturing: The Data Management Strategy.* Digital Press, Bedford, MA, 1988.

2. Olin H. Bray, Integration of Configuration Management, MRP II, and Factory Floor Systems, in AutoFact, held in Detroit, MI. AutoFact Proceedings, SME, 1988.

3. David A. Dornfeld, "Sensor Fusion," in Nello Zuech, ed., *Handbook of Intelligent Sensors for Industrial Automation.* Addison-Wesley Publishing Co., New York, 1992.

4. International Standards Organization, *Concepts and Terminology for the Conceptual Schema and the Information Base,* J. J. Van Griethuysen, ed., ISO Report TR9007 from TC97/SC5/WG3, 1982.

5. G. M. Nijssen and T. A. Halpin, *Conceptual Schema and Relational Database Design: A Fact Oriented Approach.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

6. R. Andrew Russell, *Robot Tactile Sensing.* Prentice Hall, Englewood Cliffs, NJ, 1990.

7. Sharad H. Gadre, Building an Enterprise and Information Model, *Database Programming and Design.* Premier Issue, 1987, pp. 48-58.

8. J. J. V. R. Wintraecken, *The NIAM Information Analysis Method: Theory and Practice.* Kluwer Academic Publishers, Norwell, MA, 1990.

9. John Zachman, "A Framework for Information Systems Architecture," in *IBM Systems Journal,* vol. 26, No. 3, September 1987, pp. 276-292.

10. Nello Zuech, ed., *Handbook of Intelligent Sensors for Industrial Automation.* Addison-Wesley Publishing Co., New York, 1992.

DISTRIBUTION:

1	MS 0160	Virgil Dugan, 4500
1	0188	Chuck Meyers, 4523
1	0661	Gary Rivord, 4012
1	0863	Paul Flores, 14307
1	1138	Larry Claussen, 6533
11	1378	Olin Bray, 4524
1	1378	Shanna Narath, 4524
1	1407	Al Beradino, 1809
1	9018	Central Technical Files, 8940-2
5	0899	Technical Library, 4414
2	0619	Review & Approval Desk, 12690 For DOE/OSTI