Integration of Digital Structured Documents in Virtual Manufacturing Organizations

Su-Shing Chen
Department of Computer Engineering & Computer Science
University of Missouri
Columbia, MO 65211 USA
schen@risc1.ecn.missouri.edu

Pasi Tyrväinen
Department of Computer Science and Information Systems
University of Jyväskylä
SF-40351 Jyväskylä, Finland
Pasi.Tyrvainen@jyu.fi

Airi Salminen
Department of Computer Science and Information Systems
University of Jyväskylä
SF-40351 Jyväskylä, Finland
airi@jytko.jyu.fi

Abstract

The present manufacturing sectors are burdened with increasing speed of transitioning new global business developments into production processes. This forces the old approaches of information management to adopt better evolution capabilities. Although the concept of virtual manufacturing is to facilitate integration of production processes in global industrial organizations, its information management has not yet achieved the expectations and promises. The main issue is the chaos in digital documents and software platforms. Our research is to develop a framework for overcoming problems created by the lack of standardization of digital documents and software platforms in manufacturing. The paper focuses on the integration at the document level of virtual manufacturing organizations. Our contribution lies at connecting two interdependent domains, manufacturing and information management, rapidly converging through virtual manufacturing organizations.

1. Introduction

Due to the globalization of industrial organizations and the proliferation of legacy information systems, the present manufacturing sectors are burdened with information management with ever increasing cost. In particular, manufacturing organizations of less integrative information management have higher cost and lower efficiency than those of more integrative information management. Although the concept of virtual manufacturing is to facilitate integration of production processes in global industrial organizations, integrative information management has not yet achieved the expectations and promises [2]. As manufacturing organizations are more adaptively and flexibly integrated, it becomes increasingly important to reorganize and rethink information management in the manufacturing paradigm (e.g., [5,7]).

This paper reports some research aspects in progress at the University of Missouri in the USA and the University of Jyväskylä in Finland. First, we improve the present information management paradigm by injecting flexibility into structured documentation and related media (e.g., SGML, OpenDoc, CGM, ODA). Second, we develop the intelligent agent architecture for integrating information systems as well as structured document standard paradigms. In a subsequent paper, we will build information flow models of virtual manufacturing organizations based on interfacing digital documents in individual organizations.

The integration at the document level of two interdependent approaches, manufacturing and information management, is extremely complex due to many formats and standards of digital documents and software systems in manufacturing. Our approach is to develop a hierarchical framework of variable depth integration of information objects for overcoming this complexity. Virtual manufacturing organizations will require this critical capability to provide integrative information management. Integrative information management includes design, production, and downstream activities, delivery and maintenance. Since information in virtual organizations is represented by digital and legacy documents of all kinds, our object-oriented framework provides the necessary integrative information management of digital documents in a unified way. The standardization effort of digital documents has led to some structured document models (e.g., [6,9,10,13,14]). The hierarchical framework of variable depth integration of information objects enables the integration of digital structured documents, by bringing several standards together and developing interfacing agents supporting these standards (e.g., [4]).

This integrative information management methodology will enable the manufacturing industry to achieve business efficiency and global competitiveness through fast paperless data exchange. The interfacing agents supporting rapid integration will have significant impacts on many other industries, beyond our case studies - manufacturing organizations in United States and Finland.

The organization of this paper is as follows. The second section is concerned with virtual manufacturing organizations in general. The third section discusses the object orientation approach supporting adaptive life cycles of virtual manufacturing organizations and their variable depths of integration. The fourth section describes multiple representation schemes of information objects in digital document standards. The fifth section presents our hierarchical framework to variable depth integration of virtual manufacturing organizations. The sixth section provides a list of important information entities in manufacturing related standards. An example of interfacing digital documents in virtual manufacturing organizations is illustrated in section seven.

2. Virtual Manufacturing Organizations

In the information age, the manufacturing arena has become more dependent on information through the use of computers and computer-controlled machines. In the areas of global industrial organizations, a number of concepts, including "virtual manufacturing" and "virtual factory", have emerged, requiring manufacturing be highly information-intensive and knowledge-based [2,5,7].

A manufacturing organization comprises hundreds of operations in several basic activities and related information elements - e.g., design, prototype, process plan, production, delivery, and maintenance. Some operations are performed serially and others concurrently, at time scales ranging from a fraction of a second up to several months. A traditional manufacturing environment relies on good process and product design, and effective management of tools and materials. Requiring on-demand, volume-independent, scaleable, and high yield production, a virtual manufacturing organization depends on smooth information flow and efficient information management. The information technology nature impacts the complete sequence of manufacturing activities.

There is the need to develop a manufacturing software environment for information integration in virtual manufacturing [3]. An innovative suite of software tools will coordinate smooth information flow and support efficient information management of key components in virtual manufacturing organizations. At present, these components are not seamlessly integrated. Even within one component, such as engineering analysis and design, various tools are not fully integrated, due to the incompatibility among various existing standards and different vendors. There have been attempts to integrate some limited domains. However, the complexity of the domain and the rapid pace of change in technologies, requirements and standards prohibits a complete integration of various tools, whether they are newly developed or legacy software systems.

Smooth information flow and efficient information management of key components in virtual manufacturing organizations imply solutions of the following three fundamental issues. First, the interoperability in key components encompasses design and analysis tools, databases and information bases, computing and communications hardware, and manufacturing processes. Second, the coupling of human workers and key components in manufacturing induces efficient development and robust production. Finally, a supply chain for necessary parts in the projects must be reliable to guarantee fast and stable turnaround. These key components when integrally developed will enable the on-demand, volume-independent, scaleable, and high yield manufacturing and maintenance in virtual manufacturing organizations.

An integrative manufacturing software environment consists of communications networks, computers, control systems, production equipment, databases of various information about the virtual manufacturing organization, and human workers. Integration will improve manufacturing functions including:

(1) Design that maximizes the productivity of artifacts
(2) Process planning and control that assures productivity of the entire manufacturing operation
(3) Logistics of materials, tools, and information that maximizes utilization of capital plant and equipment and minimizes accumulation of unused inventory
(4) Information archives and analysis tools that enable continuous product and process refinement critical to competitiveness

Effective integration of information management into a virtual manufacturing organization assures maximal social, organizational, and economic benefits to the organization. Also, it enables and realizes vertical integration between key components leading to a new organizational form known as virtual organizations. In this paper, we propose the object-oriented approach and the hierarchical framework of variable depth integration for realizing this new form.

3. Object Orientation in Manufacturing

A virtual manufacturing organization established for the purpose of creating new products by using the resources and knowledge of several organizations is a good test place for integrative information management. It will span the information flow across the borders of several organizations and will create extensive interaction among organizations. Manufacturing organizations are used to operate according to their private practices and with their own systems and methods. The organizational structure, workflow as well as information systems will have to adapt to the ones of the other interacting organizations in various geographical locations. Yet they are not to be changed permanently. Integration of digital documents is the adaptation between organizations through interfacing software. Interfacing software is modifiable depending on the occasions. When one interaction is complete, another may begin.

Object-orientation provides a systematic approach to integration in virtual manufacturing organizations. A more detailed description of object-orientation of large information infrastructures (including manufacturing information) has been provided in the forthcoming book on digital libraries (see [3] and related papers quoted there). Similarly to digital libraries, object-oriented abstractions can be used to represent information entities in virtual organizations. Basically manufacturing documents are considered as Information Objects and classes with information contents and basic functions. Basic functions are programs encapsulated in information objects. Furthermore complex manufacturing operations are defined in terms of objects and their classes.

Object-orientation supports adaptive life cycles and variable depths of integration in virtual manufacturing organizations. The life cycles of a virtual manufacturing organization can be several years, but also as short as several months or weeks. This means that there is not much time for recreating the information systems through tightly integrated approaches such as federated information management systems. The levels of integration must be flexible (i.e., variable depths), and objects and classes' inheritance and polymorphism may be applied to the adaptive life cycle and flexible integration.

In case that participating organizations have adopted the same set of standards or even have used the same production processes, the integration will be somewhat simpler while the different organization and culture aspects are still present. Organizations actively promoting the use of open technologies and integrative approaches, such as the one proposed here, have both the advantage of being technically able to reach a high level of integration as well as able to adopt to different organizations for adaptive life cycles. However, it is clear that the integration of a virtual manufacturing organization with the life cycle of a couple of months can not reach as deep levels of integration as the ones that have longer life cycles of organizations.

Object-orientation permits us to design and implement our information objects and their related entities according to the life cycles and levels of integration in virtual manufacturing organizations. This leads to an integrative information management paradigm sufficient enough for satisfactory integration, but still under the restrictions of various existing standards of digital structured documents. A more detailed comparison of information entities in various standards is given in the sixth section for object orientation. Interfacing is then the integrating technique to satisfy various standards. It is the mediating force to integrate various objects and classes under standardization constraints.

4. Standards of Digital Documents: Multiple Representations

Thanks to many ideas, concepts and technological solutions, the notion of "a document" carries a rich collection of definitions and connotations in research and practice-oriented sources. However, the connotations of "a document" still remain implicit adding confusion among researchers and disrupting also the integration of digital documents in manufacturing organizations. [8]

Standardization is one obvious way to accomplish integration. For example, the CALS (Computer-aided Acquisition and Logistics Support) Initiative started by the US DoD was an effort to bring together standards: IGES (Initial Graphics Exchange Specification), CCITT (Consultative Comm. for International Telephony and Telegraphy), CGM (Computer Graphics Metafile), and SGML (Standard Generalized Markup Language). This initiative has originally had mixed results for the lack of integrative methodologies to adaptively and flexibly integrate them with many legacy systems. More recently, PDES (Product Data Exchange) using STEP (Standard for the Exchange of Product Model Data) was initiated by NIST with more success. But much more work is still needed. We refer to [1,6,11,12] for more references on standards.

Today, there are so many standards established throughout the years that they actually are not necessarily beneficial to integrative information management. Structured documents created in different standards are not easily integrated for more constraints are required in them. There are possibly many representations of the same information objects. In fact, the integration of all of them may be the biggest bottleneck ever encountered in this information revolution, not only in virtual manufacturing. Our approach is a hierarchical framework of variable depth integration across heterogeneous standards, thus avoiding the bottleneck. It does not require full integration of all structured documents, rather only interfacing at a necessary level through interfacing agent's mediation. This agent approach has been discussed in an earlier paper [4]. It works for all existing standards - SGML, STEP/PDES, CGM, and others. It functions also well in the modern integrative software approach of Java and ActiveX. In fact it takes advantage of the integrative Java and ActiveX technology.

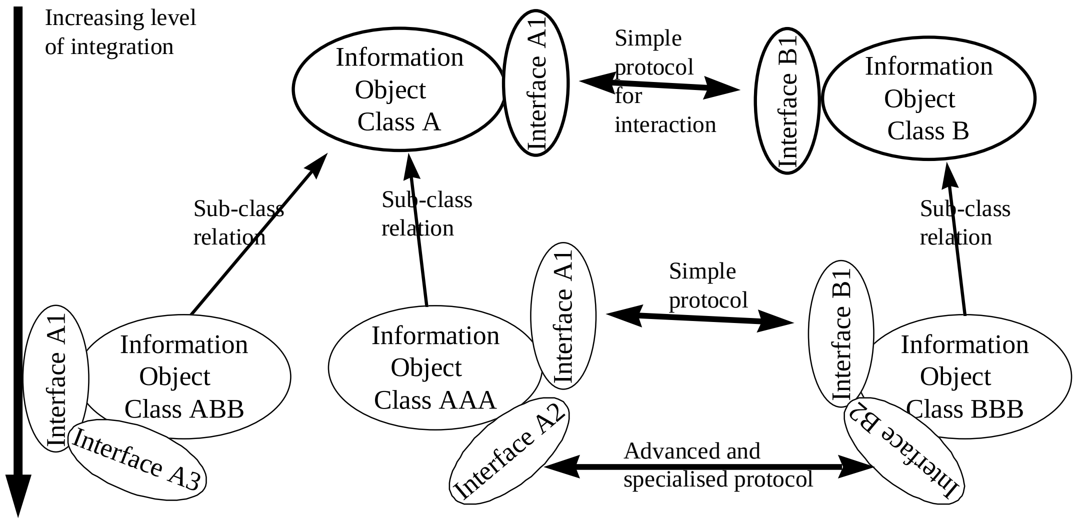

Figure 1. A Hierarchical Framework of Variable Depth Integration

5. A Hierarchical Framework of Variable Depth Integration

The variation in the life cycles for the flexible information integration of virtual manufacturing organizations needs a hierarchical framework of variable depth integration (see Figure 1). Basically we propose integration to be carried out at a necessary depth only. This is also useful for short-term cooperation in virtual manufacturing organizations not requiring very deep integration. The general framework exploits the flexibility of object-orientation.

The framework utilizes the object-orientation and information encapsulation capabilities. It can further be enhanced by the inheritance of object classes. The basic integration functions are clustered to object classes with just the basic functions to provide the interfaces needed for short-term integration purposes. In case of long life cycles of integrated organizations, the basic functions can first be made operational to reach the first level of integration with minimal time. Later on, the integration proceeds to add more functions from level to level. It implements the basic functions needed through protocols of the next levels of specialized information objects in an inheritance hierarchy.

The hierarchy is a means to organize the protocols between the classes of interfacing information objects, such as structured documents. This approach is somewhat similar to the use of application profiles, e.g., in the STEP standard, but somewhat more oriented towards application frameworks. The application profiles and other means for clustering domain-oriented information within various standard frameworks are useful. But we need the means to generalize the idea of more generic and specialized interfaces to a level applicable across all the standards commonly used in the information management of virtual manufacturing organizations.

In the lightweight integration, exemplified by the information object classes A and B in Figure 1, only the essential functions needed for cooperation are implemented. This means that the objects need a common naming scheme for the other objects to enable referencing them. This does not require extensive effort from the viewpoint of technical implementation whenever the systems have the capability to carry external references without understanding of the contents of these objects. The communications infrastructure creates the broker functions needed for locating objects with their references. It launches software tools to manipulate them and even to provide means to associate attribute information for the objects and associations between the objects for high-level protocols, such as status information describing the state of life cycle of an information object, routing information for workflow systems, and etc. Without much more efforts, the objects can make use of the standardized semantics between the objects, with or without the awareness of the internal structure of other objects. The encapsulation principle allows the internal structure to be sheltered and permits the interfacing at the necessary parts of the information objects. Those necessary parts are defined as elements of information entities in the next section.

6. Characteristic Elements of Information Entities in Manufacturing Related Standards

The diversity of the manufacturing information has made it difficult to achieve high levels of integration, not only between the various forms of digital information, but also between various standards. However, there are generic trends that are visible also in the standards to be used as the basis for integration of manufacturing information. Object-oriented and object-based aspects are the one of interest for us, because our integrative information management of virtual manufacturing organizations makes use of these object-oriented characteristics of the standards at the appropriate levels. Some common characteristics can be described as follows, starting from the most common features also needed by the simplest protocols, proceeding to features specific to certain specialized protocols.

Information Objects

Many of the standards structure the information into a form of information objects. The granularity of the objects varies, but is a sufficient base for our purposes, for examples, the "elements" and "entities" in SGML and the "entities" in STEP/PDES. This serves as the basic structure for our purposes.

Identifiers

Several of the standards provide a means for defining a unique identity of an information object. For example, the ID attribute of elements in SGML and the instance numbers and attributes in STEP. In some cases, the identifier is just based on the use of a file system. We present an example of the role of identifiers in integration. Suppose that several virtual manufacturing organizations wish to manage product documentation based on the product structure. In many cases, they have chosen to use SGML for the documents and STEP for the product structures. In order to avoid inconsistencies between two separate information stores managed under two systems and two standards, some unification is needed. Those simple approaches include embedding SGML documents into the product structure or, as usual, to interconnect the SGML elements with the product structures in STEP by using the identifier of one system in the other or through maintaining mappings between the identifiers in a database. However, even providing the basic functionality, i.e. having unique information objects with unique identifiers for each of them, is not easily reachable in many organizations. Further, the different approaches in version and configuration management and product variants make this basic problem even more complicated.

Attributes

Attributes of objects are defined in several standards, e.g. attributes of elements in SGML, attributes in STEP/PDES and MHEG (Multimedia Hypermedia Expert Group), etc. If the intended level of integration requires use of attributes, but some standard used for the class of the information elements does not, it is still possible to implement the attribute function with the infrastructure. For example, the infrastructure can provide attribute storage based on identifiers by storing <identifier, attribute-name, attribute-value> triplets in a database. When the infrastructure is assumed to pass the service requests from the user objects to the object assumed to provide the service, it will catch the call and execute a database fetch for the attribute value instead of requesting it from the information object itself. By this means the lack of the functionality is hidden from other interacting objects assuming the object to support the full functionality of the protocol at the level of integration used.

Relations/References

Some standards provide means for objects to refer to other objects. This can simply be an identifier of another object (as a link in HTML or hyperlink in CGM version 4) or contain behavioral information by launching operations for the associated object (as a link in MHEG). As in the case of attributes, this functionality can be implemented by the use of identifiers and infrastructure support in cases where the standards do not support it. E.g. by storing triplets of the form <relation, identifier, identifier> with the help of the infrastructure.

Aggregations

Objects can typically contain other similar objects that are either referred by using the identity mechanism (as in STEP) or contained in the object, as document elements containing other document elements in a SGML document instance. Sometimes the contained objects are structured differently, as picture elements in the CGM picture body or data types in the IPI-IIF. The ODA/ODIF supports both logical and layout structures so that the contents of a document can be a part of two separate aggregation hierarchies, while typically only either of the two is supported in other standards. The protocols manipulating product structures typically require support for part-of hierarchies / aggregations.

Clusters

Several objects can also be clustered or associated together, e.g. webs in HyTime, schemata in STEP, and composite classes and container classes in MHEG. This function is related with aggregation and information contents, but is not necessary for the simplest protocols.

Information Contents

In addition to containing other objects, the objects can often contain internal structures. This internal structure can be strictly defined by related standards, such as EXPRESS in STEP/PDES or MPEG, CGM, JPEG in MHEG. Sometimes this standardization of the contents is less emphasized and left open to be extended by applications, such as the data formats within tags in SGML document instances.

The standards formats oriented towards the construction of manufacturing objects (e.g. STEP, PHIGS) typically have more object orientation in the description of the aggregation relationships and contents. However, the standards formats oriented towards presentation contain large bodies of contents with less semantics embedded in the internal structures. MPEG and JPEG are good examples of these formats that can typically be classified as formats of the realization environment of the Computer Graphics Reference Model (CGRM).

The formats of contents are mainly utilized when the information object needs to be displayed, an associated editor program is launched to establish the Interface Behavior needed. In these cases, the minimal requirement is to have the associated programs (e.g. component programs in compound document architectures) available for the kind of content in hand. In some cases, the content includes also generic information needed for other purposes, such as Aggregation and Attribute information.

Interface Behavior

The older media standards focused on defining the internal structure of the information objects by defining it strictly in advance. Later on, the standards provided additional standard libraries or interfaces for manipulating the objects (e.g. IPI-PIKS) or defined interfaces for the objects defined (e.g. actions in MHEG or procedural semantics for architectural forms in HyTime) and generic interfaces to the standards, e.g. SDAI (Standard Data Access Interface) in STEP.

Some of the most recent standards provide less visibility to the internals and focus on the interfaces or the means to define the interfaces. These ones include the OLE/ActiveX and the OpenDoc document architectures. This approach enables changing or extending the internal structures, e.g. for adding new semantics or media while preserving the old interfaces and enabling use of new extended interfaces.

The behavior manipulating the objects through standard interfaces can sometimes be manipulated through some forms of scripts, as with MHEG and the scripting included in document architectures.

Specialization of the Standards

Many old standards provide profiles to define a subset of the full contents of the standards. In ODA/ODIF and STEP, application profiles provide means to specialize the data models for a domain. SGML makes the mechanism explicit in the form of DTD. HyTime expands this to include processing semantics for the architectural form containing SGML elements and attributes. These common features are utilized gradually for the purposes of integration, extending the required functionality level by level in the specialization hierarchy of the information objects.

7. An Example

As an example, let us now consider a virtual manufacturing organization with a remotely located design center and a production facility. The integration of the production design and production planning functions follows roughly the list of entities described in the previous chapter. We start by identifying the two Information Object classes, namely a Product Design class and a Production Plan class and continue by defining the interactions needed for various levels of integration. In a short-term integration, it may be sufficient to apply only use of the identifiers of the information objects. By this means a Production Plan is able to refer to the Product Designs of the plan, to launch a program to view the designs etc. These operations are rather generic to any Information Object class and can be defined in an interface for a Generic Information Object, a super class of both the Production Plan and the Product Design classes.

The additional interfaces for the Product Design and the Production Plan classes support interactions specific to these classes. Figure 2 presents two interactions not supported by the interfaces of a Generic Information Object, in the form of an object interaction diagram. The interacting objects are an implementation of the Information Object class Product Design and an object of the Information Object class Production Plan.

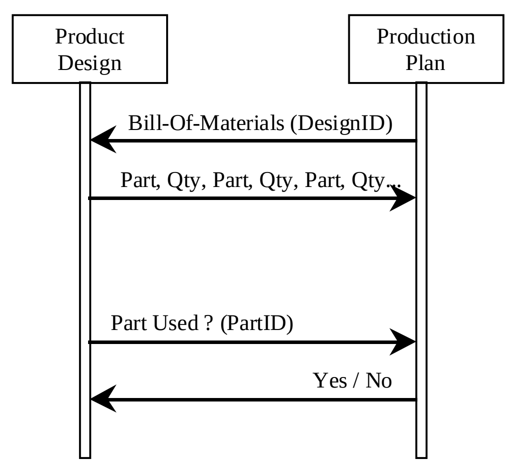

Figure 2. Some interactions between instances of Information Object classes Product Design and Production Plan.

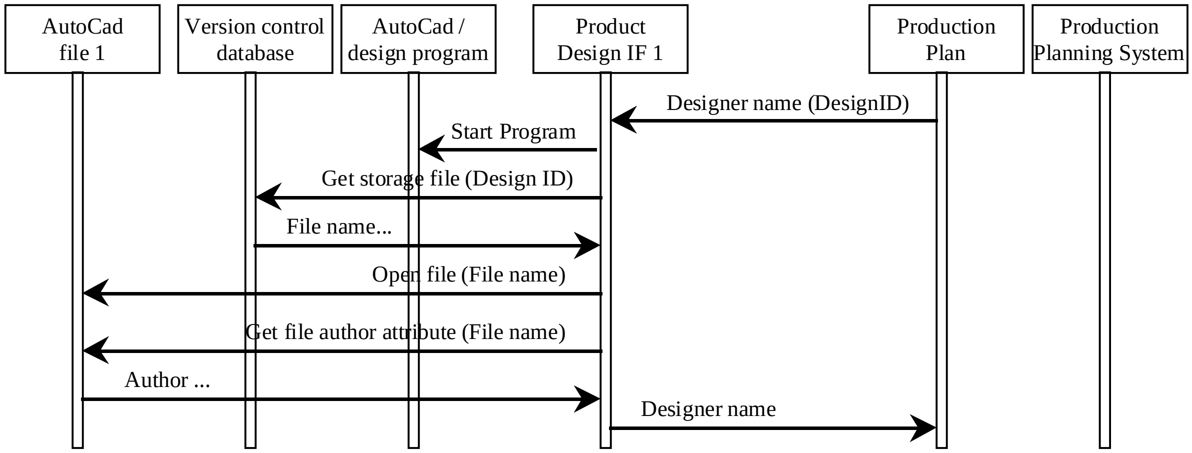

Figure 3. Implementation of the service "Designer name (Design ID)" in an implementation of a Product Design, that is based on the use of AutoCad and a version control database.

The first interaction takes place when a Production Plan (i.e. an instance of the Production Plan class) wants to know the bill of materials (BOM) of a Product Design, i.e. the parts and their quantities needed for the production of a single product. To enable this interaction, the implementation of the interface of an Information Object class Product Design has to support the service called "Bill-Of-Materials (DesignID)" that will return a list of parts and their quantities. Through this interface, a production planning system is able to get the information needed for the creation of a purchase schedule, based on the master Production Plan, the part lists and the stocks available. There are also interfaces in the other direction. For example, the Product Design may be changed due to the change of any part to an almost equivalent part. To check the feasibility of this change, the Product Design (or the designer using the design tool) is able to request, if the old part is included in the current Production Plan and if it should be replaced also in the next plan or not. (See the bottom of Figure 2.)

Implementation

From the implementation viewpoint, each instance of the Product Design class is a digital Information Object with an associated program used for manipulating it. The associated program (e.g., AutoCad in Figure 3) will store the contents of the object in an internal storage format, but can probably support some level of STEP/PDES integration. It will also be used as the means to implement most part of the Interface Behavior assumed by the protocols operating on the instances of the Product Design. The heterogeneity of the contents formats and the variation of the associated programs are hidden from the other objects by the interfaces and protocols described earlier.

Figure 3 demonstrates various implementation aspects of the Product Design class. A service called "Designer name" in the interface of a Product Design is used for requesting the name of the designer of a product based on the design identifier or a product code of the Product Design. Let us now assume, that AutoCad is used as the design tool and a version control database is used for storing the relation in between file names, product codes and product versions. Further, let us assume, that the author of the design is stored in the design file, but cannot be accessed by other means. This time the implementation of the interface (IF 1) is an agent, that uses the support of infrastructure to store attribute information. I.e. it will have access to the file name based on the DesignID by using the services of a separate database (a version control database) that happens to be available. With this information, the interface is able to launch the associated program (AutoCad) with the right file. Then the implementation uses a macro facility of the design program to fetch the information needed, based on the attribute storage provided by the storage format. In the end, the "Designer name" is returned by the interface.

One of the key enablers for the integration lies in the infrastructure support needed for implementation of the interfaces. This is achieved with the contemporary tools by mapping the interfaces into the languages used for defining application level interfaces, such as IDL in ORB-based implementations and in DCOM-based implementations. Both of these provide the broker functionality needed for locating the implementation for the service request by locating the appropriate tool implementing it. The recent fast development in WWW technology on similar functionality proposes that in the future the most promising infrastructure for our purposes could be based on Java, JavaBeans and ActiveX implementation environments. These provide the facilities for defining hierarchies of interfaces providing the more detailed interfaces of the more specialized information object classes with more elaborate functionality. In case of a rather loose integration, the effort needed for implementing the interfaces will be kept rather small, such as the simple interface example mentioned above. However, more specialized interfaces will depend much more on the support provided by the associated programs and their means of manipulating contents against which the functions in the standards protocols need to be mapped.

Interoperation Across Standards

To continue the example, let us assume that the organization designing the products will change. For some reason, the new organization uses SGML also for storing the Product Designs in the form of SGML documents. What happens now? Probably the very same interfaces will be implemented for the interactions in between the Product Design and the Production Plan. Instead of launching AutoCad, a request for "Designer name" will be implemented as a query to a SGML database. In this case, the designer name will be stored as an value of a DTD element or as a value of an attribute. From the perspective of the Production Plan, there is no change in the interactions.

All together, the design of the Information Object hierarchy is critical to the success of flexible integration. Arranging the functionality into interfaces of hierarchical Information Object classes has to take into account several aspects. First, from the viewpoint of domain semantics:

- The functionality needed for integrating the systems in according to the inherent functionality of the domain is essential. For example, it is natural to arrange the support needed for production planning as a separate interface to the Information Object class Product Design.
- The interfaces common to several classes are natural to be assigned to the super-classes, implying that a large set of Information Object classes will implement similar interfaces.

But from the viewpoint of standards and pragmatics:

- The functionality commonly supported by standards should be available in the super-classes of the hierarchy for fast integration while functionality less frequently supported should be in more specialized sub-classes for long-term integration.

This last criterion suggests, that it would be useful to follow roughly the order of features listed in Chapter 6. To start with the utilization of identifiers, attributes and relations in the super-classes of the hierarchy of Information Object classes. These features are most often supported by standards and can also be replaced with the use of infrastructure support, if not available. The level of integration reached with these means is sufficient for aiding humans in their work, e.g. by supporting launching viewers and editors for Information Objects referred in other objects and by providing attribute values and simple navigation in between Information Objects. Further integration is needed to support interoperability of systems and for automation in long-term. For these purposes, the use of aggregation, clusters and common features of information contents in the standards can be used. For example, the exchange of contents by using common standards for contents formats is more likely in the interfaces of more specialized Information Object classes.

8. Conclusions and Future Research

We have described the main idea of our research - the interfacing of digital documents - as an adaptive and flexible approach to integrate virtual manufacturing organizations and their production processes. A hierarchical framework is proposed to integrate classes of information objects at variable depths of organizations and processes. The framework is based on the common object-oriented and object-based features available in most of the standards. It is realized as a set of application level protocols between the information object classes. A simple example of integration of the design department and the production facility as separate organizations in a single virtual manufacturing organization is provided. As the CORBA and DCOM software interoperability advances, the timing is ripe for us to employ the CORBA and DCOM software facilities and the Internet WWW technology (e.g., Java and ActiveX) for digital structured documentation in the important arena of virtual manufacturing organizations.

Our work in progress at the University of Missouri in the USA and the University of Jyväskylä in Finland targets the areas of integrative information management and document management in industrial organizations. The METODI project [15] develops methods for analyzing and planning document management in industrial organizations, from the perspective of total document management. The focus of the research is in the role of documents in achieving the business goals of the organization, especially

- in the information interchange in between the organizations in the form of documents, for example, in large delivery projects of industrial systems spanning several organizations,
- in the continuous development of document management and especially in the role of specialized design methods and notations in it and
- in the technical achievements in the domains of digital structured documents and document management technologies.

REFERENCES

1. Bloor, M. S., and Owen, J. Product Data Exchange, UCL Press, 1995.
2. Byrne J. A. The virtual corporation, Business Week, February 8, 1993, pp. 98-103.
3. Chen, S. An Object-Oriented Approach to Digital Libraries, to appear, MIT Press.
4. Chen, S., Role of the information infrastructure and intelligent agents in manufacturing enterprises, Journal of Organizational Computing, 5, pp. 53-67, 1995.
5. Goldman, S. and Preiss, K. Twenty First Century Manufacturing Enterprise Strategy: An Industry-Led View, Vol. 1, Iacocca Institute, Lehigh University, 1991.
6. Maler, E. and Andaloussi, J. E. Developing SGML DTDs, Prentice Hall PTR, 1996.
7. Malone, T. W., Crowston, K., Lee, J., and Pentland, B. Tools for inventing organizations: Toward a handbook of organizational processes, MIT CCS WP 141, MIT, May 1993.
8. Päivärinta, T. and Tyrväinen, P. Documents in Information Management: Diverging Connotations of "a Document" in Digital Era. Technical report J4 of the METODI project, submitted to IRMA-98, May 1997.
9. Salminen, A., Tague-Sutcliffe, J., and McClellan, C. From text to hypertext by indexing, ACM Transactions on Information Systems, 13, 1 (Jan.1995), pp. 69-99.
10. Salminen, A. and Watters, C. A two-level structure for textual databases to support hypertext access, Journal of the American Society for Information Science 43, 6 (July 1992), pp. 432-447.
11. Smith, J. M., An Introduction to CALS: The Strategy and the Standards, Technology Appraisals, Middlesex UK, 1990.
12. Tucker, H.A. (ed.), The Open Information Interchange Technology Handbook, Technology Appraisals Ltd, Great Britain, 1996.
13. Tyrväinen, P. DMG-model based hypertext generation for practical production of diagnostic advisory systems, Proceedings of EUROPAL'90, Cambridge, UK, March 27-29, 1990, pp. 329-339.
14. Tyrväinen, P. Saarinen, P. Hätönen, K. DTM-domain modeling for technical documentation retrieval, in Yesha, Y. (ed.), Information and Knowledge Management, CIKM-92, Publication of ISMM, 1992, pp. 509-516.
15. The METODI project, Methods for Development of Total Document Management for the Industries, WWW home page, http://www.titu.jyu.fi/Metodi.