Individual Product Data Management for FMS Control : a Step to Mobile DataBase Nodes Integration

F. Chaxel, E. Bajic, J. Richard
Centre de Recherche en Automatique de Nancy (CNRS URA 821)
Université de Nancy I - Faculté des Sciences - BP 239 - 54560
Vandoeuvre les Nancy Cedex - France
{Chaxel or Bajic or Richard}@cran.u-nancy.fr

1 INTRODUCTION

This paper presents a research work which considers programmable Escort Memories systems like parts of a global distributed database system providing manufacturing control, traceability, quality and maintenance assistance, ...
To reach these objectives an escort memory (also called electronic tag) has to play the role of a vector of the information system which is in charge of the overall application management, and information coherency, consistency and reachability. For this, a methodological approach, with associated methods and tools has to be defined with respect to genericity and compatibility with information exchange standards. The information system design cycle should deal with the overall product life-cycle, starting from product and/or production information specifications thus going up to data distribution on tags and functional access architecture definition. Such an approach could give a degree of freedom to the machines which can focus to production operations and not on information management. This will reduce the increasing cost of development, maintenance and modifiability of them.

The **MobIle DAtabase Nodes (MIDAN)** concept is proposed as a logistic support to heterarchical or multi-agents control architectures developed by Duffie (1986). These architectures propose to reduce the complexity of the automated manufacturing and handling machines by considering them as partners of products, acting like controllers of their own transformations : the machine only has to be able to perform manufacturing tasks, thus co-operating under the leadership of product moving around in the manufacturing environment. Such an approach is currently under development on the IMS-Esprit project "Holonic Manufacturing Systems" (Bongaerts 1995).

Individual information about each product must be described with a high precision level involving some management problems of a so high volume of data. Furthermore, the

L. M. Camarinha-Matos et al. (eds.), *Balanced Automation Systems II*

relationship between a product and its data must be stable in space and time. A solution could be to physically associate product data and the product by using the electronic tag technology.

A product or any other mobile object carries its own information on a programmable tag which can be accessed in read/write mode by each user involved in its development process. The novelty of this approach is that machine capacities are focused on operations (transport, measures, ...) and not on management nor transmission of product information, this particular point is conceptually assumed by each product.
Unfortunately, electronic tags are by now not easy to be integrated in an enterprise information system. It is impossible to read or write structured information on these systems because they use elementary data handling protocol instead of high level and semantic information manipulation dialogue. In this field, the MONOLIN project (**MO**bile **NO**de Logistics and **I**ndustrial **N**etwork, ESPRIT 6936) is working on the basic requirements for a standard identification systems network interconnection interface allowing integration : the Escort Memory System-Application Programme Interface so called EMS-API.

Our proposal is a methodology, complementary to MONOLIN, based on international standard work (STEP ISO 10303) and supported by classical tools (EXPRESS, SQL, ...) in order to consider tags as **Mob**I**le** **DA**tabase **N**odes parts of a global distributed database.

2 PRODUCT INFORMATION MODELLING

The CIM concept emphasises interconnection of technical and management functions from design to manufacture. These functions have to share information concerning market, process and products.

For a CIM process chain, it is widely recognised that the integration of tools and databases requires standard interfaces, from CAD to workshops, based on a common data model and using a common language (lexical, grammatical and semantical).

2.1 Information Data Sharing Techniques

Presently, two solutions are available to solve the data sharing problem :

- A strong connection through a multi-users network database. Each tool can store data on the database via a common API (Application Programme Interface). Such an interface could be for example the SQL language.
- An exchange through data files in a standard format (see IGES, SET, VDA, ...). The main problem of this technique is the impossibility of real time exchange of modifications and also an important data redundancy. It is a simple solution and does not obliged to break down software structures.

2.2 International Product Model Effort : STEP ISO TC184/SC4

One of the most significant approach today in product modelling is the development of ISO 10303 standards called STEP (STandard for Exchange of Product model data) which define models, database access and neutral data files format for representation and exchange of product data (Wilson 1994). The goal is to define complete models for product life-cycle in a CIM context, as well as the means for exchange of data between enterprise functions along a product life-cycle definition (from CAD to production, maintenance, ...).

In STEP data modelling is realised with the help of an object-oriented modelling language EXPRESS and data exchange through neutral files (STEP part 21) or via a database interface SDAI (Simple Database Access Interface). STEP do not considers implementation infrastructure. In this field, projects like Esprit IMPPACT and Esprit NEUTRABAS (Brun et al. 1991) have already obtained significant results re-used in project Esprit AIT.
STEP and NEUTRABAS results are one of the basis framework of our research work.

3 PRODUCT-DRIVEN MANUFACTURING

Some researches in the field of control architectures or in the field of distributed artificial intelligence systems consider that products can be seen as intelligent entities as well as machines. In such a case, product circulation must involves product data circulation in the overall plant : each product is a vector of its data model and the server of its own information for the process.

3.1 Product/Process Interrelationship

We consider every product as a real actor within its process, i.e. it is able to :

- Co-operate with other entities that have to participate in its evolution (e.g. machines, other products);
- Adapt itself to its environment;
- Participate to the acquisition of process knowledge it is exploring.

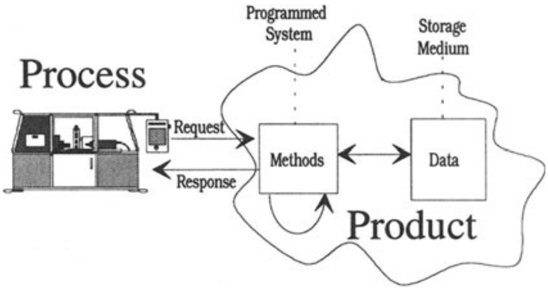

Figure 1 Process/product relationship based on the client/server model.

The schema Figure 1 points out the concept adapted from the client/server model we chose in order to respond to these objectives. In such a case each product is the manager of its own information and gives them to the process after it received a service request (*"What is the next operation to be proceed on yourself"* for example).
On the basis of a client/server relationship between product and process, user requests (i.e. process requests) are sent to an interface supporting product access methods in an object-oriented interrogation form compatible with STEP-SDAI query language.

The data storage medium, in our case, consist of identification tags carried by products or by pallets and also an optional network database. Explanation about the choice of two possible storage medium is made in paragraph 3.3.

3.2 Product Model Translation for Manufacturing Control Services

The main problem is the integration of tags in the information system i.e. :

- Reference conceptual models to put the information about the products themselves on tags;
- Tools to implement these models and to access the information.

For the purpose of our work, we assume that a conceptual model of product for manufacturing operations exists and is defined in the EXPRESS language (this model could be a STEP application protocol or a specific model)

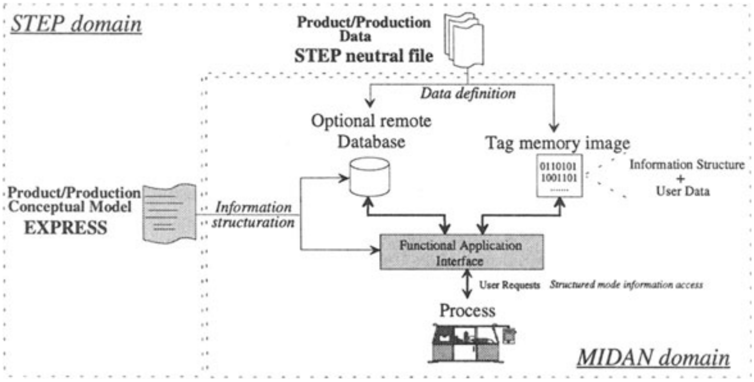

Figure 2 From EXPRESS schema to users requests.

Figure 2 (Chaxel 1994) shows the STEP-based methodology allowing translation of product model, described in the EXPRESS formalism (entities, attributes, functions, rules) which is used to generate a product-information access library supporting a direct requesting of target application on the EXPRESS schema data and to create the database structure.
In the same way, the product definition within the STEP neutral file is used to create the tag memory image, for the first state of a product associated to the current phase of its life cycle

in the manufacturing environment and if needed, to populate the previously created database schema.

A short example of the Operator interface configuration (i.e. programming) is shown in Figure 3. The C code is obtained by an automatic translation from the EXPRESS schema. In the case of a full SCHEMA we have:

- One reading function per entity *<TAGSREAD_Entity>* implementing the two SDAI functions *<GetAttr>* and *<IndexedGet>*.
- One writing function per entity *<TAGSWRITE_Entity>* implementing the two SDAI functions *<PutAttr>* and *<IndexedPut>*.

```
EXPRESS :
ENTITY Machine;
      Id_Machine : INTEGER;
      Name : STRING (20);
END_ENTITY;

IDTAGS Interface:
void TAGSREAD_Machine   (
id_Inst, Attribut)
int id_Inst, Attribut;
{/* Physical position of
attributes in data block */
 int        P_Id_Machine=0,
            P_Name=2;
 /* Attributes Value */
 int        Id_MachineVal;
 char       NameVal[20];
```

```
 if (LockInst ("Machine", id_Inst)==TRUE)
 switch (Attribut)
 {
  case 1 :
   TAGS_Int (P_Id_Machine, &Id_MachineVal);
   Send_Value (2, &Id_MachineVal);
   break;
  case 2 :
   TAGS_Str (P_Name, NameVal);
   Send_Value(strlen(NameVal), NameVal);
   break;
  default :
   Send_Error (ATT_NOT_DEFINED);
 };
else
 Send_Error (ENTITY_INST_NOT_FOUND);
}
```

Figure 3 EXPRESS to the C Interface translation.

The resulting code has to be linked with a library providing the called functions like *LockInst(), TAGS_Int(), TAGS_Str(),* ... and some communication functions like *Send_Value(), Send_Error()*, ... in order to generate the Functional Application Interface.

3.3 Remote Database Reachability

Tags are of course limited in capacity, and storing complex structures (ENTITY) needs more memory than only storing the contents (Attributes value). Today, tags tend to have higher storage capacity (up to 128 KBytes) but in some cases (like automotive industries) this may not be enough, so if it is relevant for the application a part of the data could be stored on other medium.

A network database in client/server mode seems to be the natural architecture to manage the overflow of data :

- We need to have a persistent computer storage in order to manage EXPRESS schema population and translation;
- It stores and manages high volume of data;
- Time response is efficient for most of the applications;
- And enterprises are already trained with such an architecture.

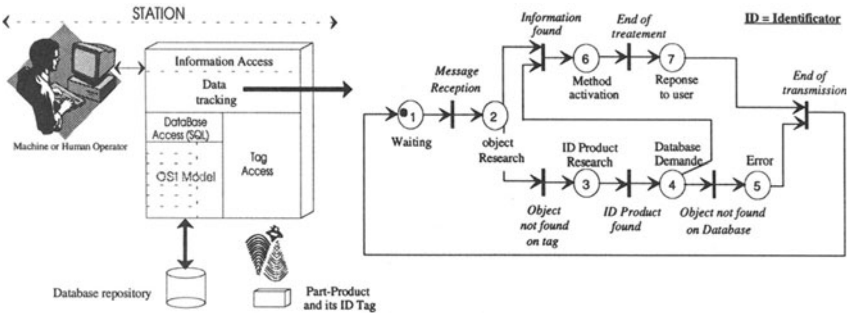

Figure 4 State graph of product data remote access from a process point.

Our approach has the same philosophy as distributed database that is, "users do not have to know where data are stored to access to them". The system performs data locating, formatting and transferring through the structure (see Figure 4). In fact, when we populate the schema, we specify where EXPRESS entities must be (on tags or remote database). Afterwards, the user can send a request to the system, without specifying the location of the objects he wants to access.

In a prototype application, we restrained the features to static objects storage (data just read by users) on the remote database and static or dynamic objects (data read or written by users) on tags.

The search for physical storage of objects is made after each user request reception, as shown in the model exposed in Figure 4 :

1. If object attributes are stored on the tag (2), the message treatment is realised (6);
2. If not, an identification of the product is made by reading its distinctive characteristics (product identification number, product type identification number) (3);
3. After which, we get data from the database (4) and we realise the treatment (6).

3.4 Application Tool

Our prototype **Tag-STEP** (see Figure 5), running under MS-Windows, uses a standard computer to generate the MobIle DAtabase Nodes structure. It is connected to a relational network database management system (ORACLE V7) through TCP-IP protocol.

The EXPRESS schema is converted in a relational schema which has to be put on the ORACLE database. In this case each EXPRESS entity is translated to a relational entity (a table) and to a view in order to manipulate subtyping. Such a conversion has already been made by the National Institute of Standards and Technology (Eggers 1988).
The schema is also converted as a C programme to generate the Functional Application Interface as explained before. This program is able to read or write data independently on tags or on the database.

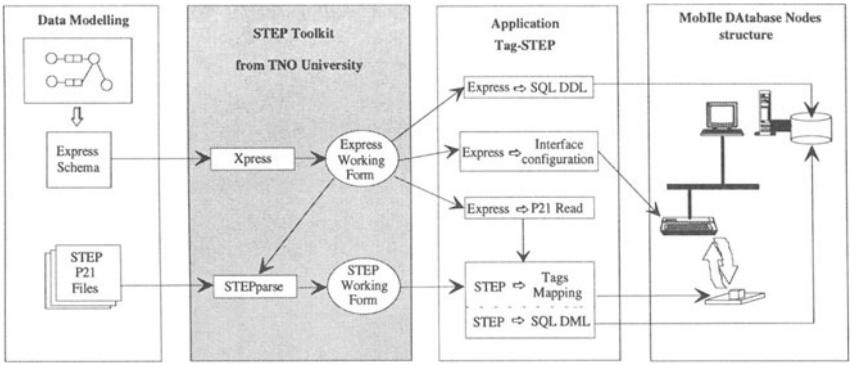

Figure 5 Structure configuration with EXPRESS schema and STEP P21 neutral files.

To populate the schema we use a STEP neutral file which has to be divided in two parts. The first is containing data stored on tags and the second data stored on the database. Each file is the characterisation of a product type circulating in the workcell.

4 APPLICATION ON A FLEXIBLE MANUFACTURING RING

The application site, is a flexible assembly ring with four independent stations. Each station is controlled by a PLC connected to the cell network. Information about products can be read at the station entry point and on the station itself by sending request to the Functional Application Interface. Because identification system controllers are not sufficiently open, each Interface is constituted by a computer and by an identification system controller.

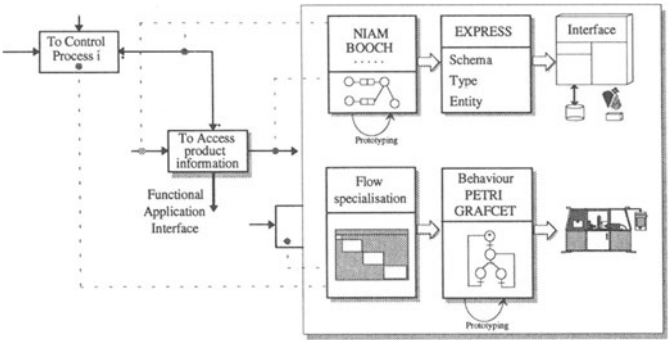

Figure 6 Modelling process.

The cell modelling process is done by using different methods to cover the functional, behavioural and informational points of view described on the Figure 6.

4.1 Functional Model

The first step was to models the cell functions by using the SADT method on an computer tool. The designer has to lets appear a special function "To Access Production Information" each time a sub-function on the model need product related information (like the parameters for the next operation). This function is fully supported by the Functional Interface described previously.

To validate the functional model we have chosen the associative method which consist in the description of all the terminal functions by using a behavioural model in Petri Nets, Ladder, pseudo-code or programming language.

4.2 Behavioural Model

The SADT model is automatically transferred on a tool call SPEX. Firstly it permit to specify the type (in the computer sense) of each arrow between functions. Secondly, the designer can describe the dynamic behaviour of each function. Thirdly it is possible to simulate the model by adding control panels. Moreover, behaviour descriptions are made in PLC languages like Grafcet, Ladder, C, and they can be exported into a tool CADEPA to program the cell PLCs. The functional aspect can be cover by using the SADT analysis tool, SPEX and CADEPA. For a full description of the methodology see the paper by G. Morel and P. Lhoste (1993).

4.3 Informational Model

Each product information requirement is specify by the input and the output flow on all the identified functions "To Access Product Information". These flows are the basis of the product data model analyse process. In our application case we are using a tool supporting EXPRESS-G able to generate the corresponding EXPRESS model. It is used as the data entry point for Tag-STEP.

This EXPRESS-G model describes the assembly operation sequence for the FMR, the quality control result of each operation, and information about product history like operation time and duration.

4.4 Models integration and simulation

Tag-STEP generates two types of product data access interface (a programme) one for a real use on a computer connected to a escort memories controller and the other for a simulation use on SPEX.

So we can play with the interface in order to test the data model and also to play with SPEX to test interactions between the information model and the process behaviour model. This coupled information-behaviour simulation gives interesting results about electronic tags (in words of baud rate and capacity).

By now the real implementation on the FMR is not yet achieved and the simulation is our actual ending point.

5 CONCLUSION

Electronic data carriers for distributed manufacturing information management is taking more and more importance in industrial applications.

This new approach for information management necessitates methodological framework for Information System implementation as well as interfacing tools based on international standard definitions. Our proposal dealing with STEP formal models has demonstrated the feasibility of high level information structuration, based on product data definition, with advanced management technics on attached escort memories.

6 REFERENCES

Brun P. et al., "A Neutral Product Database for Large Multifunctional Systems", *Seventh CIM Europe Annual Conference proceeding - CEG DG XIII Telecommunication, Information Industries and Innovation,* Turin, Italy, 29-31 May 1991, pp 87-97

Chaxel F., Bajic E., Richard J., "From STEP Product Modelling to Product Manufacturing : an Approach using Identification Tags", *European workshop on Integrated Manufacturing Systems Engineering, IMSE,* Grenoble, France, 12-14 Decembrer 1994, pp 529-535

Duffie N. A. , Piper R. S., "Nonhierarchical Control of Manufacturing Systems", *Journal of Manufacturing Systems*, Vol 5, N° 2, 1986, pp 137-139.

Eggers J.A., "Implementing EXPRESS in SQL" - ISO TC184/SC4/WG1 Document N292, National Institute of Standards and Technology, Gaithersburg, Maryland, 1988.

Esprit MONOLIN 6936, "Deliverable 06 - MONOLIN Implementation Guide : Implementation Context and General Overview ", 15 March 1994

ISO 10303-11 Product Data Representation and Exchange, ISO TC184/SC4 - 1994.

L. Bongaerts et al., "Schedule Execution for a Holonic Shop Floor Control System", *Preprints Advanced Summer Institute '95, ASI'95,* Lisbon, Portugal, 25-28 June 1995

Morel G., Lhoste P., "Outline for Discrete Part Manufacturing Engineering", *COMPEURO'93, Invited paper, 7th Annual IEEE European Computer Conference,* Paris-Evry, France, 24-27 May 1993, pp 146-155.

Wilson P.R. , "A View of STEP", *IFIP Transactions B* [Applications in Technology], Vol B-8, 1994, pp 267-296

7 BIOGRAPHY

F. Chaxel is doctor of the University Henri Poincaré - Nancy I (France). He undertakes his researches within the "Quality engineering" team of CRAN.

E. Bajic is Associate Professor at the University Henri Poincaré - Nancy I (France). He co-manages the "Quality engineering" team of CRAN.

J. Richard is Professor at the University Henri Poincaré - Nancy I (France). He manages the "Quality engineering" team of CRAN.