Collaborative Manufacturing Management in Networked Supply Chains

Michel Pouly[1], Souleiman Naciri[1] and Sébastien Berthold[1],

[1] Ecole Polytechnique Fédérale de Lausanne (EPFL)
Laboratory for Production Management and Processes, Station 9, 1015 Lausanne, Switzerland
{michel.pouly, souleiman.naciri, sebastien.berthold} @epfl.ch

Abstract. ERP systems provide information management and analysis to industrial companies and support their planning activities. They are currently mostly based on theoretical values (averages) of parameters and not on the actual, real shop floor data, leading to disturbance of the planning algorithms. On the other hand, sharing data between manufacturers, suppliers and customers becomes very important to ensure reactivity towards markets variability. This paper proposes software solutions to address these requirements and methods to automatically capture the necessary corresponding shop floor information. In order to share data produced by different legacy systems along the collaborative networked supply chain, we propose to use the Generic Product Model developed by Hitachi to extract, translate and store heterogeneous ERP data.

Keywords: Collaborative networked supply chains, ERP, RFID, data sharing

1 Introduction

The European manufacturing sector has experienced considerable changes in the last 15 years due to the reduction of the manufacturing depth. In the past, most industrial companies produced almost everything in house. Continuous pressure on the prices and the global competition forced them to focus on their core competences like engineering and final assembly as OEM manufacturers thus outsourcing almost the whole manufacturing operations.

Information is one of the most valuable resources for manufacturing companies and supply chains. Due to the huge amount of produced and exchanged data needed for the production activities, it is essential to identify the most useful ones and to focus only on the "strategic data" leading to potential improvements at the supply chain level and the "production data" leading to potential improvements at the company level.

The actual production management software (ERP, MES etc.) [1] does not rely on accurate and up-to-date shop floor data, as the classical manual collection is time consuming, error prone and not real time. Major improvements are expected from an automatic capture of shop floor data.

Numerous research works have outlined the benefits of information sharing throughout the supply chain [2]. Sharing data such as machine loads, sales previsions, inventory positions etc. has proven to increase companies and supply chain key performance indicators like the fulfil rate and the product cycle time [3], and to decrease order fluctuations [4] that characterize the bullwhip effect. However, data sharing and integration remain a major concern as companies may use heterogeneous hardware and operating systems, data management software, data models, schemas and semantics [5], which hinder data sharing efficiency.

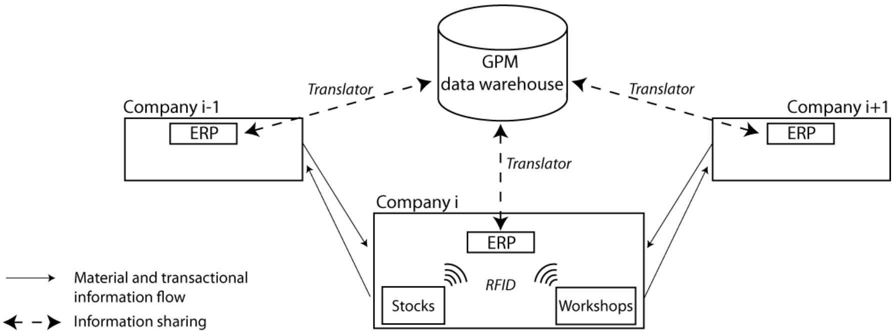


Fig. 1. Information acquisition and sharing within a collaborative networked supply chain

This paper addresses the data identification, collection and sharing processes by proposing corresponding methods and solutions based on the results of the IMS DiFAC and VIPNET research projects [6].

Information sharing process	Relevant data identification	→ Data collection	→ Data sharing
Methodology	By sending a questionnaire to 150 companies	By using RFID tags	By using : -a modeling language (GPM) -a translation tool between ERPs and a GPM data warehouse
Description	Chapter 2	Chapter 3	Chapter 4

Fig. 2. Plan of the article

2 Industrial Requirements

A web based survey on 150 enterprises of the watch, medical appliances and mechanics sectors in the Western part of Switzerland showed the major requirements of companies working in collaborative supply chains [7]. Two thirds of the answering companies were small SME (less than 100 employees):

- 70% of them have already installed an ERP
- 84% of them have an Intranet solution
- 58% of them have their own IT department

Table 1. Industrial requirements

Requirements	Very important	Important
Orders follow-up along the supply chain	82%	9%
Stock information (WIP, raw material, components and finished goods inventory)	74%	17%
Availability to promise	70%	17%

2.1 Orders Follow-Up along the Supply Chain

The most important customers' expectation is to be accurately informed about the actualized real delivery schedules. In many cases, this information is important enough to motivate customers to select only suppliers able to provide this service. Actually, this information is not immediately available to the sales department clerks, who must investigate first and then call back. What is needed is an interface to seamlessly provide this information to the authorized supply chain collaborators for instance by linking a web portal to the legacy ERP systems. The actualized delivery schedules can be calculated by using the actual locations of the orders and the average actualized lead times of the remaining operations. The following shop floor data are required for this function:

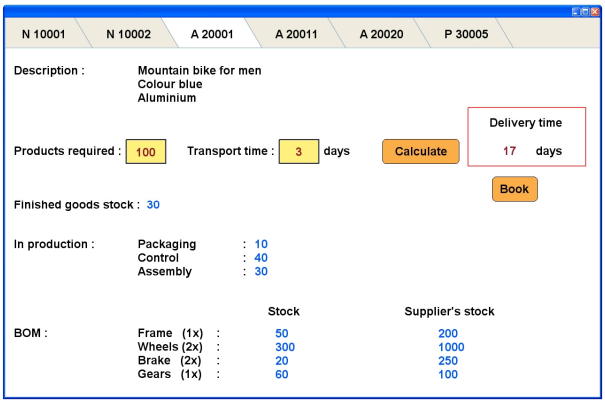
- actual locations and amounts of WIP (Work in Progress) along the supply chain
- lead time for each operation
- starting/ending time of each operation

2.2 Stock Information

The second most important industrial requirement is the updated information about the stocks that include the raw material and purchased components, the parts along the manufacturing lines (WIP) and the finished goods. Inventory management is also important because it is the base to calculate the available to promise schedules that are the third most important requirement of the above-mentioned survey.

2.3 Availability to Promise Function

The main objective of any Customer Relationship Management (CRM) ERP module is to better satisfy customer needs in order to retain them and to attract new ones. Sales people often promise their customers unrealistic delivery schedules that cannot be met. The customers become dissatisfied and resulting delay penalty fees can cut the supplier's margin. Moreover, this fact is a great source of tension between the sales and manufacturing departments. In this paper, we propose a tool connected to the company's ERP to help sales people providing more realistic delivery schedules (see figure 3).



Description : Mountain bike for men
Colour blue
Aluminium

Products required : 100 Transport time : 3 days

Delivery time
17 days

Finished goods stock : 30

In production :
Packaging : 10
Control : 40
Assembly : 30

BOM :		Stock	Supplier's stock
Frame (1x)	:	50	200
Wheels (2x)	:	300	1000
Brake (2x)	:	20	250
Gears (1x)	:	60	100

.

Fig. 3. Available to promise tool

This tool has multiple tabs corresponding to the different products sold by the company, each tab containing a product description, the number of available products in the finished goods stock (FGS), the number of products actually in production that are not yet reserved for a specific order, the number of each required component in the stock and in the supplier's stocks. The inputs are the number of required products and the transport time for the corresponding customer. The tool uses the following algorithm to compute the best possible actual available to promise schedule:

1. Test if the available FGS can cover the entered number of required products. If yes, the available to promise schedule is set to the corresponding transport time and the algorithm stops
2. If no, the number of available products is set to the value of FGS and the algorithm goes one step backwards in the manufacturing line starting from the end to check how much of the remaining needs can be covered by half-finished products in this processing step. Using the Bill of Materials (BOM) and the operation sequence, it calculates the corresponding lead times and adds them to the available to promise schedule
3. If the needs are still not fully covered, the algorithm continues another step backwards until the needs are covered. Finally, the transport time is added to the calculated available to promise schedule

The following shop floor data are needed for this function:

- WIP at each shop floor station
  - number of parts entering the station
  - number of parts leaving the station
- lead time for each operation
  - arriving time at the station
  - departure time from the station

The next chapter describes the possible methods to automatically capture the corresponding shop floor data.

3 Data Capturing Methods and Technologies

Different technologies can be used to automatically capture the shop floor data like image recognition, voice recognition and RFID (Radio Frequency Identification). After having compared these three technologies considering the implementation possibilities, the corresponding costs and the acceptance by workers, we selected the RFID as the most suitable one.

The four basic components of a typical RFID system are an antenna or coil, a reader, a transponder (RFID tag) and a middleware. Normally, antennas and readers are coupled and distributed around the working area. Each object to track has a tag attached to it. When a tag and a reader are in the same area, data can be exchanged in read only or read/write mode. The middleware is the software layer in charge of linking the RFID readers and the ERP.

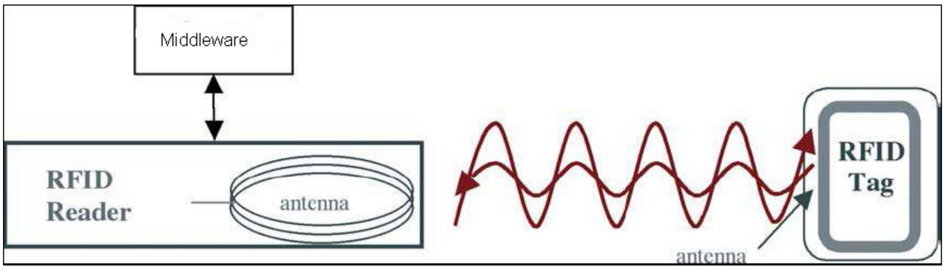


Fig. 4. Components of a RFID system

3.1 Example of RFID Implementation in the Shop Floor

Different implementations are possible using either fixed or mobile readers attached for instance to forklift trucks. We will present in the next section a possible implementation layout for a manufacturing station.

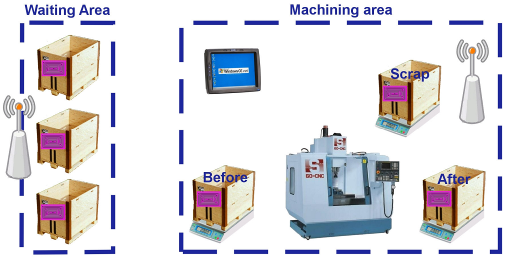


Fig. 5. Implementation for a manufacturing station

1. A pallet arriving at a station is stored in the corresponding waiting area where an RFID reader periodically reads all tags to identify the available pallets. This information is sent to the middleware.
2. The operator receives the next order on his screen with the data to identify the corresponding pallet(s) to bring to the machining area.

3. The operator puts the pallet on a scale before the machine. When the weight measured by the scale reaches a certain level, the scale sends a signal to the RFID reader of the machining area through the middleware to identify the pallet and make plausibility check against the order. The corresponding operation parameters are automatically downloaded from the ERP to the screen or directly into the CNC machine control. Furthermore, the code of the parts used after the manufacturing operation is written onto the tag attached to the pallet receiving the parts after the operation.
4. The operator takes the first part. At this time, the weight measured by the scale changes. The time between the first signal of the scale (weight reaches a certain level) and the second signal (weight changes) is recorded as setup time.
5. Then, with the weights of the three scales, it’s possible to get the number of parts in each pallet (parts before operation, parts after operation and scrap) and even to know the machining time of each part.
6. When the operation is finished, the operator can enter on the screen the data that are not automatically recorded: labor time, resources used, consumables, additional data, etc. Then he removes the pallet of machined parts that can be brought to the next station. At this time, the corresponding scale measures a weight equal to zero and sends a signal to the middleware. This signal allows the time spent in the machining area to be known and to inform the ERP that the operation is completed.
7. The operator takes the empty pallet before the machine, puts it on the scale after the machine and gets the next order on the screen.

The following raw data are captured automatically in such a manufacturing station:

Table 2. Captured raw data and corresponding capturing method

Raw data	Capturing method
Arriving time in the waiting area	RFID reader, periodically
Departure time from the waiting area	RFID reader, periodically
Arriving time in the machining area	Scale (sends a signal to the middleware)
Departure time from the machining area	Scale (sends a signal to the middleware)
Number of parts entering the machining area	Scale (divides total weight by weight of one part)
Number of parts leaving the machining area	Scale (divides total weight by weight of one part)
Setup time	Scale (time between two signals, see point 4 of process description)
Starting time of operation	Scale (signal when weight changes)
Ending time of operation	Scale (signal when first pallet is empty)
Scrap	Scale (divides total weight by weight of one part)

4 Data Sharing along the Supply Chain

Once we have defined the technical means to collect shop floor data, we introduce a methodology and a tool to make collaborative networks partners able to share this data. To reach this goal, two major conditions must be met:

- a method to extract data from the different legacy ERP systems
- the possibility to store the data in a common data warehouse using a common language

4.1 Generic Product Model (GPM)

GPM was initially developed by the Hitachi group to store and keep nuclear power plants data (technical data, BOM and drawings) over a very long period of time even if the software that generated these data had disappeared meanwhile. GPM is a core model [8] that describes one to many relationships. It defines that objects are simply connected to objects via associations. These associations can be seen as verbs that link together a subject and one to several objects:

A valve “is classified as” piping components, “is classified as” stop valve, on/off valve etc. and “has property of” size, diameter, throughput etc.

4.2 GPM-XML

The GPM model is used to represent the relations between objects. However, computers cannot process it. To do so, GPM-XML has been developed. This language based on the W3C XML uses strings that make GPM-XML very close to natural language and therefore easily understandable by humans. XML also ensures data persistent storage through its adoption as a standard and its ease of understanding [9]. This point is very important, as it is one of the main objectives of GPM, to allow people to retrieve and visualize data that may have been produced decades ago.

The ***association library*** contains the whole set of GPM available associations that are used to connect objects together. It contains information about an association name, its meaning and its roles (which define the viewpoint from which the association is seen (subject or predicate)). The 19 existing GPM associations are based on EPISTLE, POSC/Caesar and USPI-NL. Among these associations, we can cite ”is an instance of”, ”is assembled from”, and ”has property of”. The ***class library*** is an XML file that gathers all the information about GPM associations and classes. A class is defined as the regrouping of several objects that have the same classification, semantics, names and attributes and that can be recognized as common by everybody [10]. GPM classes are object-oriented and inherit all the nature of their super class(es) [9]. Consequently, classes dealing with specific products’ fields or views can be gathered in groups called ”genres” as mechanical components, plant and system, production management and linked together using associations. ***Instances*** represent the allocations of classes and attributes for the purpose of the unique recognition and the designation of the attributes and relationships between other instances [9]. While classes describe concepts and groups of objects, instances represent real world objects. Instances are built from the association and class libraries using a translator as detailed later in this paper. Initially designed to model nuclear power plant data

[11], new GPM classes have been created to handle management data as illustrated in following section.

4.3 Data Sharing Methodology

The management data we want to share are the data stored in ERP that are used to run companies and supply chain processes as planning, purchasing, production etc and the automatically captured shop floor data. To handle this type of data in the GPM data warehouse, the first step consists in mapping these data with GPM objects and related attributes. However, this task is not straightforward as we deal with alphanumerical data stored in tables that are eventually the result of other queries. As a consequence, apart from the case where we know the query and the underlying tables, it is often difficult to understand the nature and the meaning of the data that are displayed on ERP panels. Therefore, the only information users have to identify data is the name of the table and the label of the data. Knowing that these labels can have several meanings or even differ between companies and ERP, it appears very difficult to propose an automatic way to translate management data as it could be done for drawings and other technical data. Consequently, the mapping must be done in a semi-automatic way depending on the data we deal with. It means that it is the label of the data, and the knowledge of the person in charge of the mapping that will allow the creation of the mapping schemes.

Collaborative networked supply chains of SME are characterized by a large number of legacy ERP systems, and consequently a large variability of data formats and databases schemas. A common way of sharing information is through the Excel files database extractions that are proposed in the basic package of almost all ERP applications. Excel extractions make information sharing quite complex as the initially structured data contained in the ERP database is flattened in the Excel file, which can hinder extracted data understanding. The methodology detailed below explains how to give back a coherent GPM structure to flattened Excel files.

We focus in this paper on widespread products like watches that can be easily described using a Generic Bill of Materials (GBOM) [12], defined as "a Generic Bill of Material (GBOM) is designed to describe the components structure for a family of products in one data model. The specific Bill of Material for any particular product variant can then be generated on demand" [13]. Alternative methodology for customized products ERP data translation into GPM can be found in [14].

When users deal with data belonging to domains that haven't been explored yet, they may be confronted to some articles or features that are not represented by GPM classes and may have to build new classes. In the case of ERP management data, few relevant GPM classes were found in the initial class library. Consequently, a few classes covering the most common production management features (lead time, delivery time, inventory level, production cost, etc.) have been created. However, to produce a more consistent set of GPM management classes, GPM developers could use all the existing work related to business process modelling and ontologies .

4.4 Translator Development

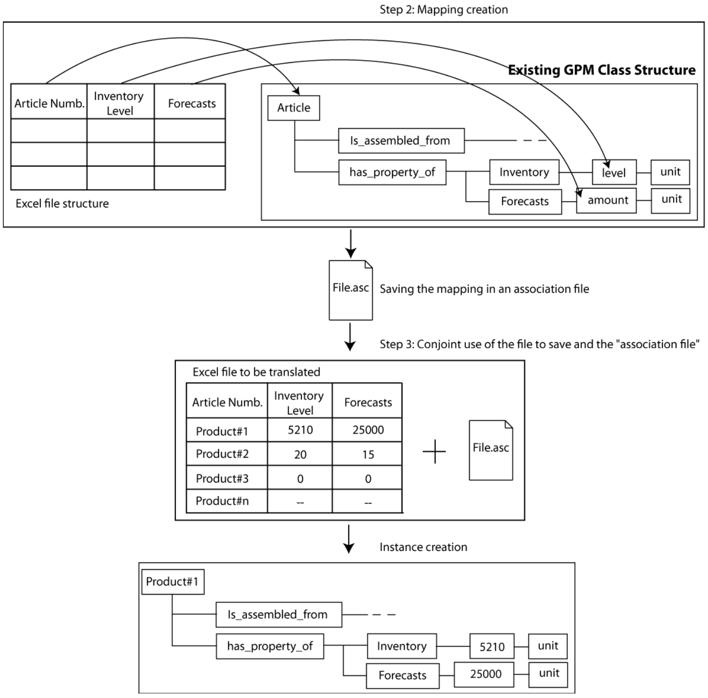
Even if a reverse translator has also been developed to extract and convert GPM data into Excel files, we will focus on Excel to GPM translator development in the following section. The translation process for widespread products is described below:

1) Identification of relevant GPM classes and attributes for representing products GBOM (*),

2) Mapping between product structure (previous step) and Excel file. The mapping is made between GPM classes and attributes on one hand, and Excel file head of columns on the other hand as illustrated in figure 6. The mapping file is saved as an ”association file” (*),

3) once a user wants to import an Excel file content into a GPM data warehouse, he or she must browse two files: the ”association file” and the Excel file to upload. Then translation is made automatically and data is stored in GPM XML in the data warehouse.

(*) Refers to steps that are done just once. In concrete terms, Step 2 is made by dragging and dropping Excel head of columns on corresponding GPM objects or attributes.



Article Numb.	Inventory Level	Forecasts
Product#1	5210	25000
Product#2	20	15
Product#3	0	0
Product#n	--	--

Fig. 6. Translator for widespread products

5 Conclusions

Collaboration within networked supply chains only makes sense if up to date information is shared between the different actors to be better informed about the corresponding real delivery schedules and lead times, the stocks all along the supply chain and the demand for products. Actually the large majority of the industrial SME don't have a correct view of their own shop floor activities. The automatic capture of shop floor data using RFID is an important step towards this goal as the actual manual capturing methods are time consuming, error prone and not real time.

In this paper, the focus is also placed on the translation and the storage of initially flat ERP management data that are characterized by a huge variability between companies. A methodology is proposed to make a semi-automatic mapping in order to translate in a friendly way (using drag and drop) unstructured and proprietary management data into a structured modelling language.

Acknowledgments. The authors would like to thank our industrial partners SAGE ProConcept SA and the Nuclear Power Plant Division of the Hitachi Group for their support and commitment and the Commission for Technology and Innovation (CTI) of the Swiss federal government, which funded this research..

References

1. Jacobs, G.P., Weston, F.C.: Enterprise resource planning (erp) – a brief history. In Journal of Operations Management, vol. 25, no. 2 , pp 357-363, (2007)
2. Cachon, G.P., Fisher, M.: Supply chain inventory management and the value of shared information. In Management Science, vol. 46, no. 8, pp 1032-1048, (2000)
3. Chen, M.C., Yang, T., Yen, C.T.: Investing the value of information sharing In multi-echelon supply chains. In Quality and Quantity, vol. 41, no. 3 ,pp 497-511, (2007)
4. Croson, R., Donohue, K.: Impact of pos data sharing on supply chain management: an experimental study. Production and Operations Management, vol. 12, no. 1 , pp 1-11, 2003
5. Ziegler, P., Dittrich, K.R.: Three decades of data integration all problems solved ? In Building the Information Society, vol. 156, pp. 3-12, (2004)
6. IMS DiFAC, http://www.difac.ch, IMS VIPNET, http://openknow.com/vipnet/index.htm
7. Berthold, S.: Integration of real time shop floor data into ERP software. EPFL master thesis (2008)
8. Hitachi Group: "A GPM basics," Technical report., (2004)
9. Hitachi Group: B - markup languages (gpm-xml), Technical report , (2004)
10. Koizumi, Y., Seki, H., Yoon, T.: Data integration framework based on a generic product model. In Tools and Methods of Competitive Engineering, *Vols 1 and 2*, pp. 891–902, (2004)
11. D. Mun, J. Hwang, S. Han, H. Seki, and J. Yang, "Sharing product data of nuclear power plants across their lifecycles by utilizing a neutral model," *Annals of Nuclear Energy*, vol. 35, no. 2, pp. 175–186, 2008.
12. van Veen, E.A.: Modelling Product Structure by Generic Bill-of-Materials, Elsevier Science Inc., pp 139-188, ISBN:00444896767, (1992)
13. Wu, P., Olsen, K.A., Saetre, P.: Visualizing the construction of generic Bills of Material. In Lecture notes in computer science, ISSN:0302-9743
14. Naciri S., Pouly, M., Binggeli, J.-C., Glardon, R.: Using the Generic Product Model for storing and sharing ERP data. CSCWD 09. 13th International Conference on Computer Supported Cooperative Work in Design. 22-24 April 2009. Page(s):618 – 623.