INFORMATION SHARING IN COOPERATIVE PRODUCTION NETWORKS

Péter Egri[1], Dávid Karnok[1,2], József Váncza[1,2]

[1] Computer and Automation Research Institute, Hungarian Academy of Sciences
[2] Budapest Univ. of Technology and Economics

Abstract: The information opacity through supply chains seriously reduces overall efficiency, which today's production networks cannot afford in order to stay competitive. The relatively new paradigm of cooperative production networks requires high automation of supply and purchasing processes and information transparency along the supply chains. In this paper we present our research about a methodology for designing and implementing information sharing systems both from theoretical modelling and from software engineering aspects. *Copyright © 2007 IFAC*

Keywords: production networks, integration and communication, multi-agent approach, architectures of intelligent manufacturing systems, production planning and scheduling.

1. INTRODUCTION AND RELATED WORK

In the last decades the circumstances in manufacturing have dramatically changed. The increasing customer expectations require ever shorter delivery times, customised products and extremely high service levels. The widely accepted and utilised total quality management (TQM) principle states that all expressed and unexpressed wishes of the customer should be satisfied and the most significant manufacturers act upon this management philosophy. This taut situation boosts competition between manufacturing enterprises, which inspire them to work out new ways towards achieving more efficient production. The problem is characterised by a set of dilemmas: satisfying or shaping market requirements, exhibiting guaranteed or emergent behaviour, customising or standardising, optimising for local or global performance. The currently accepted direction for resolving these questions points towards extended coordination and even cooperation along the supply chains within production networks. Sustaining growth and competitiveness can be achieved only through transition from factory automation to network automation (Spur and Seliger, 2007).

As real-life case studies, we take two production networks in fundamentally different industrial sectors. However, as we will show, their fundamental goals and problems are surprisingly similar. One of them operates on a consumer goods market and manufactures low-tech electronic products applying mass customisation technology and mostly make-to-stock production. The other network is situated in the automotive industry and aims to shift toward customise-to-order approach.

Planning the behaviour of a production network poses plenty of problems. The participating enterprises are usually autonomous entities which means they intend to follow their own goals and objectives. In addition, the necessary information for decision making is also distributed, thus without appropriate information sharing the lack of information leads to poor quality of decisions. This decision structure is called **upstream coordination** (Pibernik and Sucky, 2007), since the effects of end-market uncertainties grow as information passes upward in the supply chains, which is often called bullwhip effect. The theoretical solution to this problem is to appoint a central decision maker, whom every participant has to share all relevant information. The resulted planning task is rather complex in itself, since the information about the future is still uncertain and in addition, different, conflicting objectives (e.g., service level and operation efficiency) should be considered. However, this **centralised coordination** approach is practically unrealisable. Several intermediate settings are also conceivable between these two extremes. Pibernik and Sucky (2007) call these approaches **partially centralised coordination** and they also introduce a measure for centralisation. This is a general model of describing stages of cooperation; their paper regards only the master planning task, though. These different stages can also be illustrated as a range of colours from cold blue to hot red.

Váncza et al. (2007) suggested a three staged roadmap for enhancing coordination in networks: (i) centralised coordination, (ii) decentralised coordination and (iii) information sharing infrastructure. Firstly, one should study the centralised setting and develop optimal or near optimal solutions for this problem, which—in the supply chain field—focuses mainly on inventory management policies. In the second step the real, decentralised network should be considered. The goal here is to design such **coordination mechanisms** (protocols and incentives for autonomous enterprises) wherewith the decentralised decision can achieve (or approach) the theoretical optimum derived in the previous step. This challenge is studied by supply chain contracting theory (Egri and Váncza, 2007). Thirdly, the information sharing infrastructure should be designed, which enables the practical implementation of the coordination mechanisms by increasing information transparency. This task contains three layers. The **conceptual layer** must define a common dictionary, so that the software systems at different enterprises can "understand each other" (operate together). This layer is studied by the enterprise application integration (EAI) and ontology engineering fields. The **protocol layer** should define the orchestration of the communication, i.e., what and when has to be communicated. The **infrastructure layer** describes the specific instruments of the communication, e.g., Web Service technology or ebXML Messaging Service.

2. ORGANISATION MODEL AND DYNAMIC SUPPLY LOOPS

Both artificial and natural complex systems usually build up in a hierarchical structure in order to deal with difficult problems efficiently. For an analogy, consider the human brain, where the temporal perception and cognitive control tasks compose five different levels: strategic, segmented tactical, maneuver, short-term integration and synchronisation levels. Each level differs in their temporal frames (granularity) and horizons. The planning levels are linked by a feedback-control cycle, which defines the relationship between them according to the so-called reafference principle (Tanida and Pöppel, 2006). Therefore the human brain can be regarded a pluralistic system with several different, often redundant or even inconsistent models of the very same environment, which also apply different "algorithms". The right choice of the appropriate model depends on the actual task to be solved.

In enterprises the hierarchy is separated into three proactive planning levels: strategic (long-term), tactical (mid-term) and operational (short-term) (Fleischmann and Meyr, 2003). In addition, a near-time execution control is responsible for executing the plans and reacting to unexpected events (see Fig. 1). Note that this is a hierarchy of tasks and independent from the organisation structure (e.g., pyramidal, network). To continue the analogy, the hierarchy of human cognitive tasks does not imply any hierarchy of the individual neurons.

Fig. 1. Planning matrix: the hierarchy of planning tasks in enterprises.

In a production network, every enterprise has similar inner structure regarding the above planning functions, although they differ in complexity, dimensions and realisation. When two enterprises are linked by a supply chain—and in absence of centralised coordination these are the dominant links—, they join the corresponding levels with each other. In this way, a tier-n supplier is linked only with tier-(n+1) and tier-(n-1) enterprises, and this is the basis of the **dynamic supply loops** (DSL) concept (Dangelmaier et al., 2007). In DSL, every inter-enterprise relationship is bilateral, hence they are easy to implement and control. In this case, also the cooperation can be only bilateral—which can be regarded as a special case of partially centralised coordination—and the operation of the whole network emerges from these cooperative agreements.

The horizon of the **strategic level** covers usually several years. The goal here is to design the network on the long term, which involves decision about core competencies, choosing from available suppliers (or sometimes even from customers) and adjust capacities to the planned yield. Since the problem in this level is too complex to be completely modelled and the decisions have consequences in the long run, it is generally supervised by human experts. During the planning process, several possible scenarios (frame plans) are generated and evaluated both with the help of decision support systems and with negotiations between enterprises. On the **tactical level** the objective is to plan cost efficient production on a medium term—approximately one year—with one week as the time unit. This level should create demand forecasts for the horizon and based on this, make plans for the yield, production, inventories, supply and distribution. On the **operational level** the main goal is to realise the medium-term plans. The horizon here is only a few weeks, the planning cycle is daily and the granularity of plans are often less than an hour, but maximum one day. This level should plan the demand fulfilment, transportation, schedule production and ensure necessary materials. Moreover, the delivery of components and products should be planned often in just-in-time (JIT) or just-in-sequence (JIS) manner. In order to adapt to changing circumstances, the planning tasks on these two latter levels should be performed on a rolling horizon, i.e., cyclically the previous plans have to be revised and modified according the to current situation. On the lowest level, the **execution control** should monitor the progress of the plans and some reflex actions have to handle exceptions, which can emerge from changes in plans (e.g., increased customer demand) or deviation from plans (e.g., due to machine breakdowns). In DSL the feedbacks along the supply chains are also communicated through this level.

As a demonstration for the role of the levels, we present a widespread form of cooperation, the vendor managed inventory (VMI) concept (Lee and Chu, 2005). The main advantages of this model are that the customer does not have to deal with ordering and inventory handling problems, while the supplier can plan its production and raw material purchase in a more flexible and more efficient way. On the strategic level, the partners should design the supply network and negotiate about specific conditions: prices, service levels, compensations, capacities and even penalties. On the tactical level, the customer gives component forecasts for the suppliers. These forecasts are not firm component orders, but the imprecision of them should be (partially) compensated in order to inspire the customer to increase the forecast quality and share the risk of uncertain market between the partners (Egri and Váncza, 2007). On the operational level, the suppliers have to satisfy the short-term demand of the customer, based on the detailed production (consumption) schedule. This can be done with JIT or JIS delivery, which does not necessary mean production. Any unforeseen increase of the demand or supply problems cause events, which can be handled with appropriate material buffers (safety stocks) or in more serious situations they can only be managed by re-planning and rescheduling production.

The feedback mechanism in the brain is functioning well—even though we do not fully understand it—, but in enterprises the relationships between these levels are often ill-defined and sometimes completely disregarded. Inappropriate communication between planning tasks can lead to suboptimal efficiency in an enterprise, let alone in the whole network. A common example is when the sourcing department on the strategic level regards only the piece price and chooses suppliers with the lowest bids. On tactical and operational levels however, this can cause enormous inventories and logistic costs. Another frequent case is the inconsequential models of production planning and scheduling, which lead to either infeasible problems or idle capacities on the operational level (Váncza et al., 2004). In order to manage these issues, it is not enough to focus on independent planning functions, but they should be regarded as coherent processes.

We modelled the planning functions and the interactions in the production networks using Gaia, a multi-agent organisational modelling framework (Egri and Váncza, 2005). The agent-oriented methodology seems to be a proper choice for this task, since it offers (i) a design metaphor for complex systems, (ii) technology for handling interactions and (iii) simulation tools alike. We used the analytic models of Gaia, which provides tools for describing processes with **roles** and **interactions**. We extended the model with the description of **shared information resources**, which is the main communication interface between different roles within the same enterprise, and in addition, the inter-enterprise communication is also based on these data. This part of the model is an abstraction of the practically used enterprise resource planning (ERP) systems. Although agent technology is an appropriate tool for modelling purposes, in industrial applications agent-oriented frameworks are surprisingly seldom used (Monostori et. al, 2006). We also act upon this approach, cf. Section 4.

3. CONCEPTIONAL DATA MODEL OF SUPPLY CHAINS

Sharing of relevant knowledge—both within an organization and between enterprises—is a widely studied problem for decades, which was raised by the need to consolidate data for decision support and e-business purposes. Since different data repositories usually use different formats, electronic data interchange is a hard problem; therefore data integration methods are required to transform information into a common representation format. Two possible approaches are **data warehousing**, when data from the different sources is extracted and stored in a common, uniform repository, and the other is applying **mediator systems**, i.e., every data source has to be surrounded by wrappers—modules that access data and present it in a standardized format, whenever required. In a decentralised system, like a production network, the former centralised warehouses are less practical.

The description of integrated data belongs to the field of **ontological engineering**. Gómez-Pérez et al. (2004) distinguish between two levels of ontologies. **Lightweight ontologies** contain concepts, concept taxonomies, properties of concepts and relations between concepts. The authors show that these ontologies are closely related to techniques used in software engineering (UML class diagrams) and database technology (Entity/Relationship diagrams), thus they can be used also for such ontology descriptions. In contrast, **heavyweight ontologies** restrict lightweight ones by adding axioms and constraints in forms of formal logic in order to support automatic reasoning. Accordingly, most domain ontologies—especially in the e-commerce domain—are lightweight.

One can take basically two main approaches for developing ontologies for inter-enterprise communication: top-down and bottom-up design. In the former one, one follows a general standardised guideline, like the Supply-Chain Operations Reference-model (SCOR), as in (Ureten and Ilter, 2006) or (Fayez et al., 2005), and specialises it according to the actual circumstances. The more common bottom-up approach analyses the existing local database schemas in order to build a common ontology upon them (Schnurr and Angele, 2005).

We surveyed the ontology engineering literature in the area of supply chain information sharing. There exist lots of specialized ontologies that cover only specific domains thus are hard to reuse. There are also several upper-level ontologies which are too abstract to be applied in particular situations. And there are many initiatives to align the different ontologies by providing mapping rules between them. Since supply chains and their requirements are fundamentally different, there is no "one-size-fits all" standard ontology, which would be detailed enough to be applicable in every practical case. The conclusion was that it is much easier to design a specialized ontology from the scratch—using the experiments of the previous initiatives—than to adopt existing ones. Further moral of ontology engineering—in accordance with other related research projects—is that one cannot build real working ontology efficiently, if the goal is only to collect ontology. Ontology must be built during a process (such as design), otherwise the collected ontology doesn't work. Furthermore, one should not try to build a general purpose ontology, because the scope becomes too wide so it quickly becomes difficult to handle.

Consequently, we decided to apply the bottom-up approach for designing a conceptual data model for information sharing, but we also respected the shared information resources model of our multi-agent organisational model. The central concept in our model is called **channel**, which identifies the flow of a component between a supplier and a customer. This channel-oriented design enables that the same component can be purchased from different suppliers (through different channels) and vice versa, the supplier can offer the same components to different customers. According to the degree of cooperation we distinguished two types of channels: the traditional **purchase order channel** and the **coordinated channel**. In the latter the supply is based on forecasts instead of purchase orders, and only short-term automatically generated delivery orders exist for accounting purposes. A simplified version of the model—without the attributes—can be seen in Fig. 2.

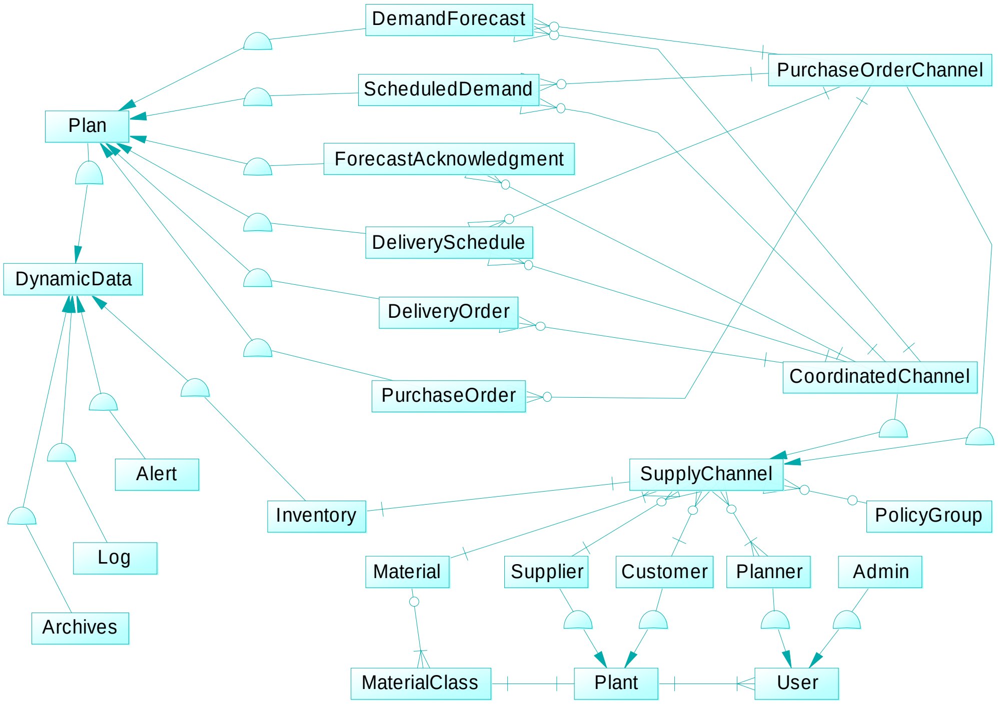

Fig. 2. Simplified conceptual data model of inter-enterprise information sharing.

4. APPLICATION CASE STUDY

Based on the above introduced models, we implemented an information sharing system which handles data on the tactical and operational levels of the planning matrix and also predicts shortages in advance. The so-called **Logistics Platform** (LP) serves a Hungarian production network with a focal manufacturer of mass-customized products and several (internal and external) suppliers. The system bridges different aspects along three dimensions. Firstly, it supports automatic information exchange between enterprises in order to automate component supply. Secondly, it covers different levels of the planning hierarchy. The third dimension is along time: it supports bridging forecasted demand and planned supply of the future, and also evaluates the precision of the past plans compared to their executions. Fig. 3. presents the architecture of the implemented system.

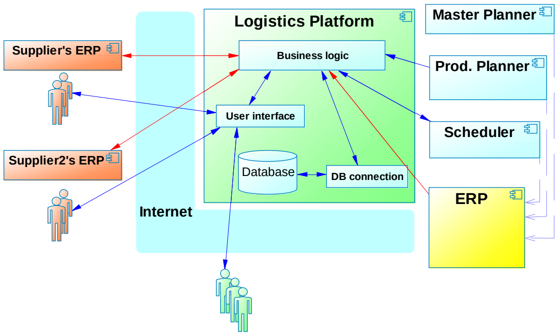

Fig. 3. Architecture of the LP.

As the figure shows, the LP provides interface for information sharing between various systems as well as a user interface for near-time monitoring of component supply. The communication with other systems is now done via direct database access due to more rapid software development—in this way other systems did not have to be changed. The short-term refinement of LP includes the development of **message-based communication** for which the XML message schemas are already being developed. This modification points toward the service oriented architecture (SOA) approach.

The developed and deployed version of the LP follows the focal structure the supply network where it will be applied. It is a Java Enterprise Edition (EE) web application built on the customer's proprietary web application framework. This framework manages the database connection pool, the request dispatching and corporate Single Sign On (SSO) authentication. The application can be accessed from the customer's intranet as well as from the external suppliers through the Virtual Private Network (VPN) of the customer.

Each user has an associated list of channels which he/she can see and modify. This allows the privacy to be retained between different suppliers. On a channel, every assigned user can read the same data, but users at the sides of the customer and supplier have different modification rights. For example, the supplier's user can modify the delivery schedule for component while the customer's user can not. On the other way around, the inventory checking rules of a channel can be set solely at the customer's side.

The web application collects data from legacy systems either via

- direct Java Database Connectivity (JDBC) data access to the customer's scheduling and planning systems, or
- XML-based (eXtensible Markup Language) data exchange with the suppliers' and customer's ERP systems.

The XML-based data exchange with the suppliers can be automatic by using Secure-SOAP (Simple Object Access Protocol) services built into the web application or direct XML file upload in which case the logged-in user's account is used for the data validation context.

The data and information acquisition process works in three different ways: periodically scheduled, event-based and ad hoc. Currently, a job is running every morning right after the customer's scheduling system has generated its new production schedule (and, subsequently, its scheduled component demand). This job collects also the actual inventory data as well as the material planning and forecast data (from the customer's planning system). During the day, the customer's operators can change the initial schedule by hand and this change is propagated automatically to the LP. In the general case, the web application's administrator can trigger any time a complete resynchronisation of the LP with the related systems. The LP can calculate a delivery schedule for the supplier automatically or by hand. These changes are then propagated back into the customer's scheduling system and are used as firm delivery promises. Because the delivery promises can cause the schedule to change, and possibly cause a ping-pong effect between the LP system and the customer's scheduling system, only the job at morning is allowed to call the rescheduling function. After the schedule changes are loaded back into the LP and the delivery schedules are recalculated and again written back into the scheduling system, no further rescheduling is performed.

The web application's report and input screens are designed for maximum data and access security by utilising:

- user roles, page level access check and object access checks;
- client- (JavaScript) and server-side form validation and data integrity checks; and
- anti-SQL injection and Cross Site Scripting (XSS) techniques by using only JDBC's PreparedStatement and HTML-encoding of all user entered text before presentation.

Beyond guaranteeing security, another primary design goal was to make the access of the vast amount of data behind the LP fast and filterable. The speed requirement was achieved by using in-memory object caching technology for critical master data objects such as channel and material properties. Filtering is a key feature in the application, because each user can have hundreds of assigned channels, but space and time restrictions allow them to operate only on a small subset at a time. Therefore each user can define his/her own set of filters which he/she can use later on in any situation. The filters which are logical constructs of (property – value set) pairs belong to the personal profile of the users. Note that filters are used also for collecting basic and generating aggregate values for a set of channels, like for evaluating the overall performance of a supplier who is responsible for a number of channels.

5. EXPERIENCES

The Logistics Platform is now being used by the planners of the manufacturer and of the suppliers for months now. To start with, channels for thousands of components (including packaging materials) have been set up. The warm-up phase of using the application has shown that it really meets the design goals and supports the detection, prediction and resolution of actual and anticipated conflict situations. By using the LP, both supply and demand planners can see, compare and analyze dynamic information coming from different and heterogeneous sources. They can analyze the same situation both from the aspects of demand and supply. This work of the planners is supported by a sophisticated filtering function and several task-oriented reports.

6. CONCLUSIONS

The automated information sharing between enterprises along supply chains are sorely needed for coordinating supply with demand and even for enhancing efficiency by cooperation. We developed high-level conceptual models for designing proper data exchange and we also implemented a software system based on these ideas. However, as it turned out, using the LP also helped human experts detecting serious glitches and inconsistencies in the existing planning processes and data administration. The practical benefit of our work is not only a developed system, but also a methodology and know-how of designing and implementing inter-enterprise information sharing systems. Our ultimate goal is to make efficient local planning, as well as reliable plan execution each partner's primary interest. This is the key for inspiring cooperation in a production network whose global behaviour emerges from the decisions of its autonomous members.

ACKNOWLEDGEMENT

This work was supported by the NKFP Grant No. 2/010/2004 (VITAL) and the FP6 AC/DC project No. 031520.

REFERENCES

Dangelmaier, W., A. Döring, B. Hellingrath, J. Lentes and T. Timm (2007). Integration of Customize-To-Order and Build-To-Order Concepts For The 5-Day-Car. *Proc. of IFAC MIM 2007* (this volume).

Egri P. and Váncza J. (2005). Cooperative Planning in the Supply Network - A Multiagent Organization Model. *Proc. of CEEMAS 2005*, pp. 346-356.

Egri P. and Váncza J. (2007). Cooperative Production Networks – Multiagent Modeling and Planning. *Acta Cybernetica*, in print.

Fayez, M., L. Rabelo, M. and Mollaghasemi (2005). Ontologies for Supply Chain Simulation Modeling. *Proc. of the 2005 Winter Simulation Conference*, pp. 2364-2370.

Fleischmann, B. and H. Meyr (2003). Planning Hierarchy, Modelling and Advanced Planning Systems. In: *Supply Chain Management: Design, Coordination, Cooperation. Handbooks in OR and MS, Vol. 11* (de Kok AG and Graves SC (eds.)), pp. 457-523, Elsevier.

Gómez-Pérez, A., M. Fernández-López and O. Corcho (2004). *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer.

Lee, C.C. and W.H.J. Chu (2005). Who Should Control Inventory in a Supply Chain? *Eur. J. Operational Research*, **164**, pp. 158–172.

Monostori L, S. Kumara and Váncza J. (2006). Agent-Based Systems for Manufacturing. *CIRP Annals*, **55**(2), pp. 697-720.

Pibernik, R. and E. Sucky (2007). An Approach to Inter-domain Master Planning in Supply Chains. *Int. Journal of Production Economics*, **108**, pp. 200-212.

Schnurr, H-P. and J. Angele (2005). Do Not Use This Gear with a Switching Lever! Automotive Industry Experience with Semantic Guides. *Proc. of ISWC 2005, LNCS 3729*, pp. 1029–1040.

Spur, G. and G. Seliger (2007). Flexible Manufacturing Systems – History, Reality and Visions. *Proc. of 40th CIRP International Manufacturing Systems Seminar*, in print.

Tanida K. and E. Pöppel (2006). A Hierarchical Model of Operational Anticipation Windows in Driving an Automobile. *Cognitive Processing*, **7**(4), pp. 275-287.

Ureten, S. and H. K. Ilter (2006). Supply Chain Management Ontology: Towards an Ontology-Based SCM Model. *Proc. of the Fourth International Logistics and Supply Chain Management Congress*, pp. 741-749.

Váncza J., Egri P. and Karnok D. (2007). Planning in Concert: A Logistics Platform for Production Networks. *Proc. of DET 2007*, pp. 461-470.

Váncza J., Kis T. and Kovács A. (2004). Aggregation – The Key to Integrating Production Planning and Scheduling. *CIRP Annals*, **53**(1), pp. 377-380.