PSLX White Paper

Advanced Planning and Scheduling (APS) Conceptual Definition and Implementation

January 2005

PSLX Consortium

Table of contents

Introduction

Objectives of the white paper

PSLX Consortium published a technical specification in 2003. As a recommendation, the specification defines APS (Advanced Planning and Scheduling) and attempts to promote proper application of IT to current manufacturing industries. The concept of PSLX's APS (which is slightly different from those used in the late 1990s) and its competitiveness has been disseminated and is increasingly understood by certain industries. Practical applications of the specification reportedly have delivered rewarding results. However, in terms of system implementation, it has been argued that the specification has many ambiguous and incomplete descriptions, which need to be made more detailed and practical to users of this specification. In order to satisfy this requirement, and to increase the value of APS for industries, PSLX Consortium started developing a new version of the APS specification in 2004.

This white paper introduces the core concepts and an abstract of the new version of the PSLX specification, which is currently being developed by PSLX technical committees via the Web. The previous specification succeeded in engendering clear understanding of the scope of APS, of the problems in current industries, and of the need for a way of addressing such problems using Web-based technologies. This new specification is designed to contribute to actual industries that intend to implement APS by clearly defining their business processes and developing new information systems. The effect of the PSLX specification will be to make this process less onerous. In particular, interoperability between software applications, which is one of the main technical aspects of PSLX, will be dramatically increased.

From the point of view of manufacturing practitioners, the new specification will suggest how they can gain significant benefits by implementing APS. They need to lead the project and organize all IT vendors, presenting clear directions and requirements based on the APS concept. The involvement of manufacturing practitioners in the implementation and enforcement of APS is the key to achieving successful results, because only these practitioners have detailed knowledge of their own competitive situations.

The specifications introduced in this paper are incomplete and have not been approved as a standard. The reason for publishing this paper with only partial information on what will become the standard is that we hope to gather valuable input by inviting contributions from many of you who are stakeholders in this area, and who are interested in developing crucial specifications for future manufacturing industries. Since this white paper introduces the essence of the latest discussions in technical committees, and indicates the basic ideas embedded in the new version of the PSLX specification, we hope that it will contribute to your basic understanding of PSLX's APS and stimulate a variety of comments and suggestions for improving the future specification.

January 2005
Yasuyuki Nishioka, Ph.D.
Chair of PSLX technical committees

Chapter 1

Conceptual definition of APS

Industrial background

Industrial needs for which APS is applicable are addressed in this section. These needs are the starting point of our motivation for developing new standards. But first, we need to take into account three critical aspects of the market environment:

(1) Unexpected demand behavior and uncertainty of the manufacturing environment

Since demand forecasting doesn't necessarily correspond to reality, it is hard to establish a balance between demand and supply, even if planners use state-of-the-art software. This is caused by greater market segmentation and by the huge variety of product choice now available to customers. Manufacturers may also experience sudden tremendous increases or decreases in their sales due to technical or social factors.

(2) Shortened product life cycle and agile manufacturing

The market continuously seeks new products. Manufacturers need to synchronize their release of new products with market dynamics, otherwise risk losing sales. Synchronization with the market and keeping up-to-date with technological improvements become the most important factors in developing and maintaining competitive advantage for manufacturing industries. Consequently, product lifecycles become shorter and it is difficult to make predictions for the next period in advance.

(3) Changes in business processes for supply chain management and EDI requirements

In terms of supply chain management, old and sustainable alliances with suppliers have to change to take account of global optimization of procurement. Web-based purchase order transactions are replacing EDI (Electrical Data Exchange) transactions. The new style of supply chain requires flexibility and agility. Internal planning information should be provided to partners to enable collaborative manufacturing.

However, current manufacturing IT systems are not necessarily geared to provide solutions for the environment noted above. In fact, we believe that the current information systems have the following three weaknesses:

(1) Lack of functional capabilities and applicability of conventional ERP systems

Most manufacturers developed MRP or MRP II enterprise information systems in the late 1970s or 80s. Some of these were replaced with ERP over the last ten years. Yet, whether they have MRP, MRP II, or ERP, the basic algorithm employed in these systems is 1970-vintage and is hard to follow the current dynamic environment. It cannot provide accurate information for plant managers.

(2) Limitation of client-server architecture for information systems

Conventional databases in mainframe computers and in client-server systems need to be precisely designed with high-level attention given to ensuring that there is consistency between all related schemas. These databases refuse extensional information that does not match their schema. However, this extensional information is the source of knowledge on plant floors, and needs to be shared among different individuals in different divisions. So far, this can only be achieved via face-to-face communication.

(3) Less flexibility and robustness for continuous improvement

Both business processes and engineering processes need to be continuously improved in manufacturing industries. On the other hand, information systems implemented by manufacturers cannot be changed every time there is a process improvement. Sometimes, manufacturers may hold off modifying their business processes because of the huge investment and potential risks involved in revising IT systems. Furthermore, migration of data from previous systems to a new system is always a work-intensive and complicated exercise, and some critical information cannot be inherited.

The aim of PSLX Consortium is to provide new solutions based on the APS concept and emergent technologies, such as the Internet and object-oriented modeling, taking the current manufacturing environment and the functional limit of conventional information systems into consideration.

Significant features of APS

APS is defined as a system and methodology in which decision-making, such as planning and scheduling for industries, is federated and synchronized between different divisions, within or between enterprises, to achieve total and autonomous optimization (PSLX technical specification 05, PSLX common dictionary, 2003). The original concept of APS was launched in the US in the 1990s, defining advanced techniques of production planning or supply chain planning with detailed scheduling and optimization algorithms. The techniques developed since then have been partially employed as the engine of planning systems in some ERP and SCP (Supply Chain Planning) software packages.

The concept of APS proposed by PSLX Consortium has many new features, which improve on the conventional concept to meet the concerns of advanced manufacturing enterprises. PSLX's APS is not a part of the planning system in ERP, but rather the entire planning and scheduling system within an enterprise.

(1) Data-centric management with an abstract data model

Enterprise information systems have a data schema for representing information handled in each business process. The data schema and business processes are closely related, and changes in business processes need to be revised in the corresponding data model. Data stored in enterprises is a heritage and needs to be inherited by new systems. APS has an abstract data model that is completely independent of the system implementation view. Enterprise data is therefore protected from the lifecycle of IT systems.

(2) Extensible system for following real business changes

Manufacturers face huge market uncertainty and need to have a competitive business model along with well-designed products. The decision-making system in enterprises is the mainframe of business models. APS enables easy design and modification of an enterprise-wide decision-making system based on actual business changes because of the flexibility of its system architecture.

(3) Close relationship between design and manufacturing divisions

To cater for shortened product lifecycles and frequent releases of new products, or for changes in existing product designs, manufacturing management needs to prepare new technical documents, new equipment, and new instructions for workers, often without much advance warning. This means that planning and scheduling systems have to deal with the complications of creating or revising master data for the new products. APS manages technical data such as BOM (Bill of Materials) and production routing data by enabling collaboration between the manufacturing and engineering divisions. The information includes not only formal and approved data, but also informal data that may not be consistent with the formal data.

(4) Real-time performance analysis and KPI (Key Performance Indicator) support

All activities on plant floors affect the cost of products. Therefore, performance management becomes a very important issue. However, undertaking a performance analysis for each activity to determine which activities contribute to the final company benefit is very difficult. APS has a mechanism for collecting cost information from the performance data of each work center, and for immediately calculating and translating it into any of several parameters defined as KPIs (Key Performance Indicators), which are valuable for future planning decisions.

(5) Reliable and accurate Master schedule for enterprise-wide collaboration

In most ERP systems, the MPS (Master Production Schedule) does not correspond to the reality of plant floor operations because of a lack of two-way communication. Compared to this, the MPS managed by APS is defined as a pacesetter in enterprise-wide operations, and all enterprise activities can participate in the creation of the MPS. Detailed scheduling for each plant floor can be incorporated into the MPS in order to make it feasible for realistic applications.

(6) Synchronous manufacturing along the supply chain by detailed scheduling

Supply chain management deals with the long-term inter-enterprise relationship and with the short-term operational synchronization between the detailed manufacturing schedule and the supplier schedules. Since detailed scheduling collaboration with suppliers calls for high-level management, there are very few cases in which this integration is established. In fact, even within an enterprise, such scheduling integration requires considerable effort. Current SCP systems generally make plans based on aggregate level information. APS can provide a framework for collaboration with suppliers at the detailed scheduling level by sharing the schedule information, and by notifying those suppliers of schedule changes that may affect them.

(7) Autonomous automation support through system visualization and fault detection

One important feature of TPS (Toyota Manufacturing System), or lean manufacturing, is autonomous automation (autonomation), in which humans in a manufacturing line are always thinking about system performance and act voluntarily to make improvements. In manufacturing information systems, autonomation means human involvement with the information systems by means of visualization of information flow in order to detect and fix problems. Sometimes, the system itself needs to change. The decision-making process and logic embedded in APS are not permanent. They can be continuously changed, not only by business process designers, but also by personnel on the production line.

(8) Manufacturers' strong initiatives in system design and development

Information systems are key enablers of a business model whereby manufacturers can gain competitive advantage. Technologies for information system implementation are becoming more complex, so that only IT specialists, system integrators, or package vendors are able to understand the details. However, the leading roles in system development projects are taken by individuals within manufacturing enterprises. Projects that do not involve the users of the system always fail because of gaps between the final outcomes and the user's requirements. APS has a system architecture and framework that can be easily understood, even by non-IT specialists. Guidelines for implementing APS allow individuals in manufacturing enterprises to concentrate on their own business processes.

APS state-of-the-art technologies

Some advanced manufacturers have already implemented APS and its relative technologies. This section introduces some of the available technologies in current APS implementations. These are the starting point in achieving the final goal of APS.

(1) Operation-centric Bill of Materials (BOM)

The conventional BOM represents the product components, directly connecting a parent to a child. This is useful for calculating the quantity of each material needed to produce the required quantity of final product. On the other hand, routing information for producing the product is managed using independently represented data, while scheduling systems use the data to calculate the load of each resource. APS integrates the conventional BOM and routing data via a new data structure referred to as an operation-centric BOM, focusing on operations that can connect both items in the old BOM and resources in the routing table.

(2) Detailed data and constraint modeling on plant floors

Detailed schedules for final dispatch require high levels of accuracy. In order to achieve this, the scheduler has to be aware of the many different constraints existing on the plant floor and has to apply them to the schedule. Conventional schedulers can deal with very simplified constraints, such as resource constraints and precedence constraints. In addition to this, schedulers in APS can represent more detailed constraints, such as material constraints, changeover constraints, sub-resource (labor and tools) allocation constraints, multi-task constraints, and so forth.

(3) Finite capacity and inventory scheduling (FCIS) algorithms

Finite capacity scheduling (FCS) deals with resource constraints and calculates a schedule that does not exceed the resource capacity. One of the best advantages of scheduling logics in APS is the finite capacity and inventory scheduling (FCIS) capability, in which an operation is never scheduled on a Gantt chart if the materials necessary to produce a product do not exist. FCIS explicitly deals with inventory, which is consumed by downstream production and produced in advance by upstream production.

(4) Bottleneck optimization and synchronized scheduling

If the performance of a single bottleneck process significantly affects the performance of the entire system, APS can provide a schedule for the process and let the other processes synchronize to the bottleneck. In other words, APS first takes care of optimizing the bottleneck process, then, backward and forward scheduling algorithms are applied to the upstream and downstream processes, respectively. According to the theory of constraints (TOC), time buffers in the schedule protect against any disturbances to the bottleneck.

(5) “What-if?” simulation of the master production schedule (MPS) based

on plant reality

The master production schedule (MPS) contains important information for collaboration between the sales and manufacturing divisions. In APS, the date on which each product is shipped to customers is always elaborated upon, taking into account detailed production information from the plant floor. Feasibility of the schedule is evaluated by detailed scheduling, which functions as a “what-if?” simulator. Simulation results may be approved as part of the future MPS.

(6) Dynamic full-pegging technique for dispatch orders and manufacturing lots

The MRP system generally cannot detect a direct effect on final customer orders when delays or problems occur for a particular dispatching order or manufacturing lot. This is due to single-level pegging information. On the other hand, production on a direct pegging system allows plant operators to determine the final customers for each operation. This is flexible and allows requests to be easily changed, but it is inefficient because of uneconomical lot size. Dynamic full-pegging in APS is a technique that shows the relationship between final customer orders and actual work orders or lots in the plant, even for an economical lot production. At the same time, the relationship can be revised when urgent, high priority orders come in.

(7) Optimization methods using meta-heuristic algorithms

In order to create an optimum planning solution for manufacturers, APS has several optimization algorithms, such as the GA (Genetic Algorithm) and Tabu-search techniques. Planning and scheduling problems have many different constraints and many decision variables, which can result in an explosion of combinations. However, these optimization algorithms, called meta-heuristics, allow planners or schedulers to find sub-optimum feasible solutions within a practical period of calculation time.

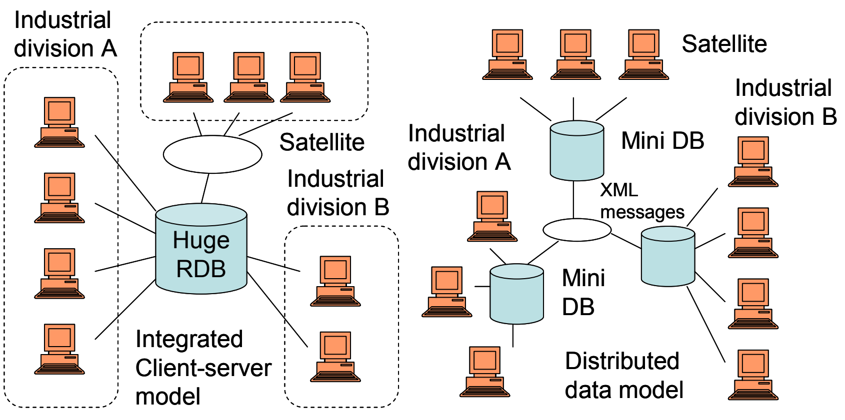
Chapter 2

Information System Architecture for Manufacturers

Information Systems for APS

Current popular architecture for information systems is designed on the basis of client-server systems. However, there are disadvantages to running huge systems on this technology, in that all the detailed data that are not referred to in the schema remain outside of the system, and revision of the schema requires extraordinary effort because it can greatly affect many existing software applications. In order to adjust information systems to particular cases of plant floors, data representing each production process are very important. Furthermore, a schema representing a plant needs to be flexible so that it can be adapted in response to actual changes.

The information system architecture that PSLX Consortium is proposing features both the advantages of current client-server technology and further advantages that flow from distributed autonomous systems, which allow local systems to change in detail levels without any agreement with others. Fig. 2-1 shows the difference between the two information system architectures. Compared to the current architecture, which has all the data stored in a central database, the future architecture will have several small databases. Local and domain specific information is created and stored independently in local databases. These distributed databases are roughly integrated by occasional limited communications.



Current architecture

Future architecture

Fig. 2-1 Information system architecture

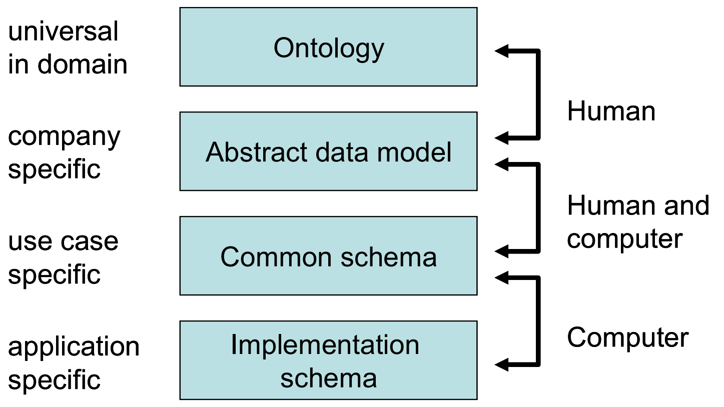
In the future architecture, consideration must be given to interoperability between autonomous databases, each of which has its related subsystems. Collaboration across different databases and data consistencies will be the issues for the designers. In order to establish communication among different data schemas, there needs to be proper discussion on the semantics behind the data schemas. An abstract data model is put in place for this purpose. Abstract data models can represent data elements of business processes in a particular enterprise independently of individual implementations of computer systems. It is useful to define components of business processes, input and output of components, resources and controls, etc.

An example of an abstract data model is the APS domain model, which is defined by PSLX Consortium as a reference for various types of industrial data models. This is a kind of common data model that can be modified to adjust to each enterprise characteristic. Particular procedures and algorithms in business components are not included in this model. An abstract data model is all about business and engineering from the viewpoint of enterprise planning and scheduling. The number of abstract data models is equal to the number of enterprises. The model will be changed if the corresponding processes are changed.

Abstract data models need to have PSLX ontology behind their definitions. Ontology is the basis of all definitions of semantics, representing primitive elements and their relations. Each abstract data model of an enterprise can be represented by attempting modifications based on the APS domain model. Alternatively, a model can be developed from scratch without reference to the APS domain model. In both cases, PSLX ontology provides a basic semantic structure for interpreting data elements in different contexts in different industries. Data become information to be effectively used in a system based on the existence of PSLX ontology.

While an abstract data model is independent from system implementation, an information system designer needs a certain data schema for implementation. He or she designs an optimum schema with respect to each system environment. RDB schema and internal data structure in an application program are examples. However, as shown in Fig. 2-1, in the information system architecture proposed by PSLX, the data schema for implementation should never be created directly from an abstract data model. In terms of interoperability, there exists the serious problem of translation between the abstract data model and the implementation schema requiring human support, because the abstract data model contains subjective definitions using ontology. This means that precise data communication between applications cannot be obtained successfully.

Fig. 2-2 Layers for application interoperability



In order to establish application integration between different business activities, there needs to be a common data schema that is defined in advance, together with agreed usage of semantics for the data elements in the schema. Fig. 2-2 shows the hierarchy of data models, all of which are necessary for the APS information system architecture. In Fig. 2-2, the common schema is defined as a data model, which acts to integrate different implementation schemas on different applications. A common schema should be defined for each business activity before application systems are implemented. Then, application system programmers can choose the most appropriate schema and interpret the semantics, using PSLX ontology, in order to apply this to their own implementation schema.

Decision-making in manufacturing

As a general principle, ”decision–making” is regarded as a top-level human mental process. This paper uses the term more restrictively, to focus on decision-making as the process of determining information related to planning and scheduling for activities within an enterprise. It includes not only planning and scheduling, but also the associated processes that create and manage information for those purposes. Before discussing the features of decision-making in APS, we need to be aware of certain disciplines. For APS, decision-making should:

(1) Be performed on the basis of correct and timely information;
(2) Involve plant floor personnel;
(3) Monitor actual activity to provide feedback;
(4) Share information on causes and results in order to collaborate with others; and
(5) Be simple and visible in terms of mechanisms.

With respect to APS, PSLX defines planning and scheduling as the primary aspect of decision-making in enterprises. Target actions of decision-making in APS may or may not be explicitly clear in advance. Sometimes, planners need to establish parameters within which they can make a decision at a particular stage. In order to design APS as the primary decision-making system in an enterprise, and to make it perform better, designers of enterprise information systems and business models should understand some features of decision-making. The basic features of decision-making could be described as follows.

(1) Feedback and feedforward

A decision on a particular action can be made using feedback mechanisms. Although relatively low-intelligence in nature, this is the most popular type of decision-making. In this type, the difference between the state of actual results and the projected state at the time of decision-making is considered, and used to assist future decision-making. This type of decision-making is called “feedback”, because the output of a decision is re-used as an input for future deliberations. Most control systems in the physical area utilize the feedback process.

On the other hand, “feedforward” systems can also be effective in situations where future events can be predicted at the time of decision-making. In the feedforward mechanism, planning and scheduling are used to avoid potential negative effects of the future event, and to optimize performance. This enables calculation of projected states, based on current conditions and actions decided. The feedback mechanism reacts after changes occur, so that errors are increasing during the reaction “gap”. In the feedforward mechanism, performance down is relatively short, as a result of prediction and more elaborate decision-making.

(2) Initial planning and re-planning

Decision-making for actions to take place in a certain future period can be divided into two types: Initial planning and re-planning. Initial planning makes decisions for the target period from scratch. The plan developed in initial planning will become obsolete if a different situation arises before the planned time for the action approaches. In such a case, re-planning is implemented to partially revise the plan in order to adjust to new circumstances that could not be foreseen at the time of initial planning.

In actual industrial situations, various kinds of actions are planned along the timeline from decision to eventual completion, and those actions are generally revised along the way. In other words, most decision-making activities occur in re-planning or re-scheduling, rather than during initial planning. Of course, some re-planning can be defined as initial planning when certain additional constraints are deliberately introduced.

In the target period of re-planning, there will be actions decided on the basis of previous planning. Some of these cannot be revised, some do not need to be revised, and some are completely open to change. Timing of particular actions is an important factor in their degree of change difficulty. Generally, the farther the time of action from the time of decision, the easier any revision of the action.

(3) Planning horizon and planning cycle

Two important parameters that characterize decision-making in a business model are the planning horizon and the planning cycle. The planning horizon is the target period of decision-making. Only the actions and events expected in that period are considered in the decision-making. The planning period is relatively defined by start time and end time (or duration). For example, “from next Monday and over the following seven days”.

On the other hand, the planning cycle is the interval between a given decision-making process and the one that preceded it. Those two decision-making points address the same issue in an actual ongoing situation. The duration of a planning cycle is either fixed, variable, or a blend of the two. In fixed cases, the cycle would be defined as in, “every Wednesday”, “the 25th of each month”, etc. The variable type includes such cases as unexpected arrival of an order quantity, or inventory dropping below a certain level.

Planning and scheduling

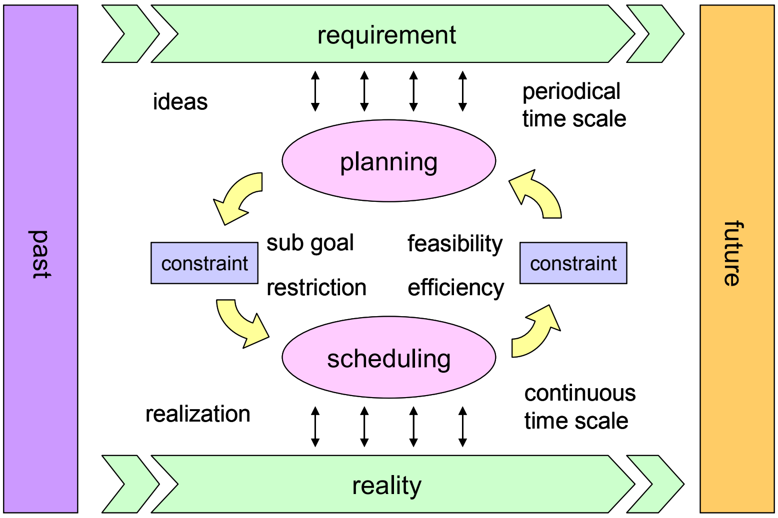
In a general sense, planning is more general decision-making than scheduling, however, distinctions between the two are usually fuzzy. In this white paper, we provide clear definitions of and distinctions between planning and scheduling, in order to support an efficient APS design. First, planning is generally defined as an activity for clarifying actions or operations to achieve a given goal and reserve enough resource capacity to hit minimum targets.

By comparison, scheduling is generally defined as an activity for allocating actions and operations to particular resources at particular times, taking into account various actual constraints and optimization of several evaluation parameters. Scheduling decides many parameters for resources that deal with constraints on the continuous time scale that parallels real time.

In terms of decision hierarchy, planning ranks higher than scheduling because a scheduling decision is made using the results of a planning decision. Some constraints and target levels for objective functions in scheduling problems are determined by planning problems in advance.

Conversely, the results of scheduling show whether or not the result of planning is feasible and efficient. If it is not feasible, planning needs to generate another result for scheduling. Feasibility and efficiency of scheduling are types of constraints of planning. Fig. 2-3 illustrates the relationship.

Fig. 2-3
Planning and
scheduling



Technically, the differences between planning and scheduling result from differences in the target data models. Decision parameters of planning and scheduling correspond to different types of data models, each of which are related to different aspects of time concepts.

In planning, the main parameters will have values for certain periods of time. For example, “amount of production this month”, “divisional sales next month”, “summary of overtime hours for next week”, etc. These decision parameters are not used for decisions on timing, but rather for any variables associated with a period of time.

On the other hand, decision parameters in scheduling are defined as representing the particular timing of actions, e.g. start time and completion time of operation, inventory issue time, shipping time, etc. Parameters of sequence information for operations are also included as relative time representation.

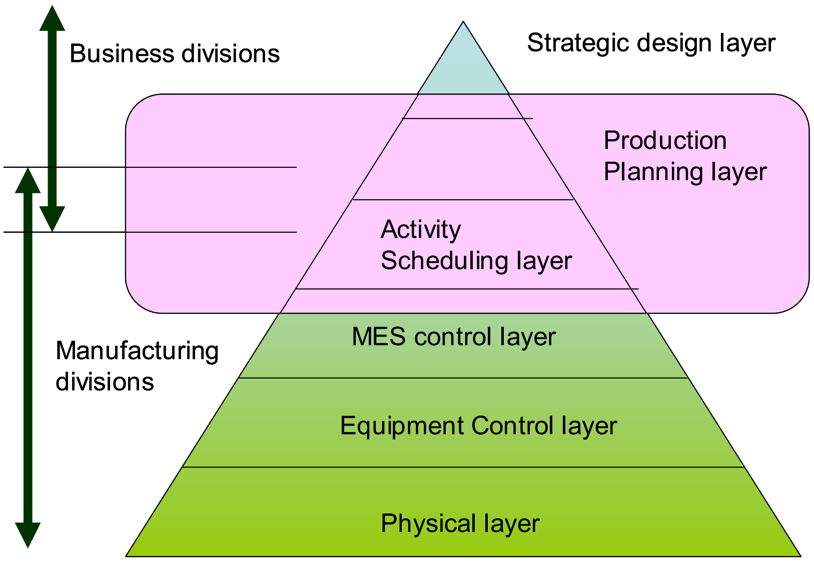
The relationship between planning and scheduling is asymmetrical, because a plan can have several possible schedules whereas a particular schedule corresponds to just one plan. There can be a very large number of applicable combinations for a solution to a scheduling problem. In such cases, planning has to reduce the number of combinations by addressing restrictions as a result of the planning process.

The length of period in the decision parameters for planning problems is significant. The shorter the time becomes, the more precise the planning required for resolving schedule problems. On the other hand, if the period is long, planning can take into account a great deal of information from a holistic viewpoint. Therefore, the length of period in planning parameters is important in order to achieve optimal integration of planning and scheduling.

Decision hierarchy of APS

This section explains APS as a decision-making system for enterprises in the manufacturing industries. Typical modules and structure of APS are described. From the viewpoint of companies overall, decision-making can be divided into six layers. Fig. 2-4 identifies these as the strategic design layer, production planning layer, activity scheduling layer, MES control layer, equipment control layer, and physical layer.

Fig. 2-4
Information
processing
hierarchy



APS controls the production planning layer, activity scheduling layer and the overlapping areas of the layers immediately above and below those two layers. As shown in Fig. 2-4, business divisions cover the top two-and-a-half layers, while manufacturing divisions are responsible for the bottom four-and-a-half. The planning layer and the activity scheduling layer are handled by both business and manufacturing divisions. In addition, the lower three-and-a-half layers are sometimes distributed geographically.

The area covered by APS in the decision-making hierarchy can be further classified by some other considerations. The three levels shown at left in Fig. 2-5 represent different granularity of the target of decision-making parameters. The top level deals with decision-making for total volume of production, where information on different product items is summarized in the same group or category. On the second level, each product item is distinguished and parameters associated with the products are decided. The third, or detailed level, is where not only information about final products but also information on their components, such as sub-assemblies, parts and materials, is discussed.

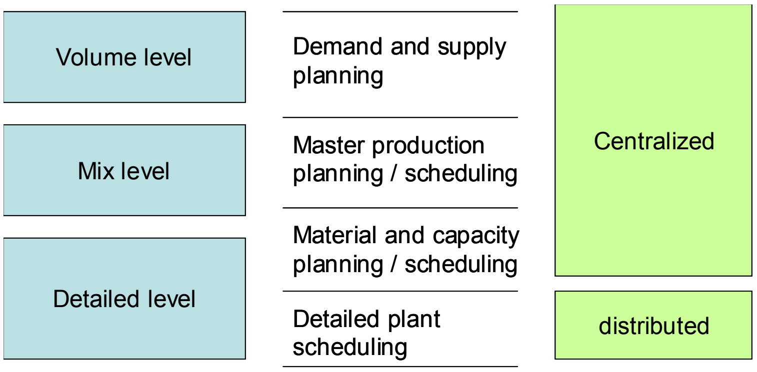


Fig. 2-5 Level of decision-making for production

The right-hand category in Fig. 2-5 indicates whether decision-making is centralized or distributed. In general, most decision-makers in manufacturing business divisions prefer the centralized approach. On the other hand, decisions for detailed manufacturing management are better made by the distributed method. As shown in Fig. 2-5, the border between the two is at the detailed level, because all items from product to materials need to be considered enterprise-wide at least once in order to achieve synchronization across all processes.

According to the two views described above, decision-making in APS can have four detailed layers, each of which corresponds to a functional module of decision-making described below. In the hierarchy, one layer is usually managed by one business activity; however, some adjacent layers can be merged and managed by advanced integrated software.

(1) Demand and supply planning

In demand and supply planning, production is considered at an aggregate level, such as “product family”. In terms of resources, demand and supply planning deals with capacity aggregated either at enterprise level or at particular area levels within factories. This decision-making cycle has a relatively long- or medium-term planning horizon. The maximum capacity of resources for production can be changed, if necessary. Financial aspects are involved in this decision-making so that enterprise-wide benefits can be optimized.

(2) Master production planning and scheduling

Master production planning and scheduling decides production volumes and timing for particular final products, according to customer demand. This is a short- to medium-term decision–making horizon. The quantity of each product is determined relative to a combination of received customer orders and projected orders calculated by demand forecasting. Target resources in this level are similar to those for demand and supply planning; however, the capacity limitation for a whole factory or particular area is based on constraint parameters rather than decision parameters. A schedule generated at this level is used to forge a kind of "contract" between the sales and manufacturing divisions. At the same time, all business activities are synchronized to this by confirming feasibility of the schedule according to their local capacity information.

(3) Material and capacity planning and scheduling

In material and capacity planning and scheduling, the quantity and production date of final products in the master production schedule provided by the upper level are extended to operations necessary for producing the products. Then, those operations are allocated to particular resources at certain times on the planning horizon. This decision-making process deals primarily with operations; subsequently, resource capacities and inventory of intermediate parts and components are discussed in relation to the associated operation. The concepts of MRP (Material Requirements Planning) and CRP (Capacity Requirements Planning) are included in this level.

(4) Detailed plant scheduling

Finally, detailed plant scheduling is addressed for actual plant floor operations. As with material and capacity planning and scheduling, this also focuses on operations. A feature of detailed plant scheduling is dealing with detailed constraints and requirements on each distributed plant floor. Furthermore, the granularity of scheduling outputs arrived at by this decision-making process is more precise than that of material and capacity planning and scheduling. Generally, the granularity of the elements of detailed plant scheduling corresponds to an appropriate unit of activities ordered by plant floor managers as part of daily procedures. Output of the detailed plant scheduling is used as a source for dispatching information when the time for an activity in a schedule is approaching and entering the action period. Work orders are forwarded to the corresponding plant operators.

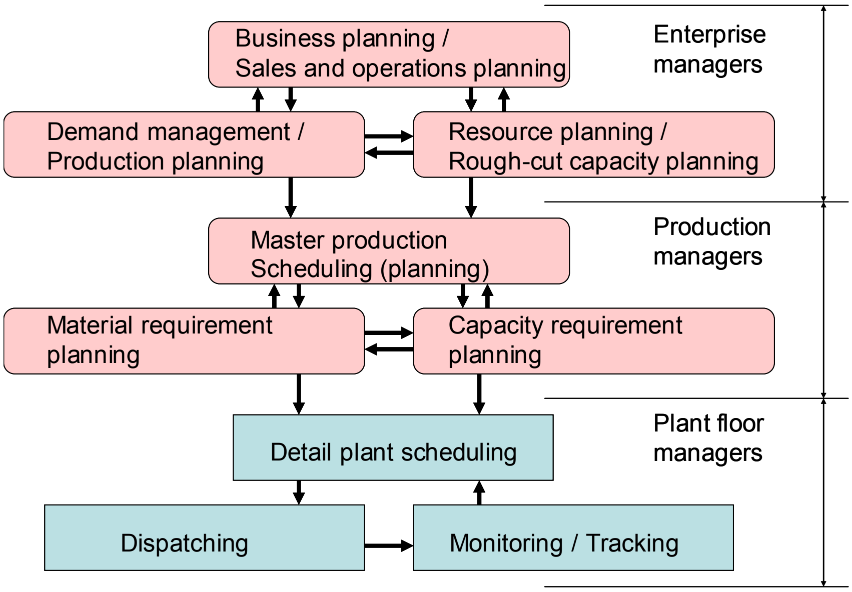
Mapping to legacy ERP systems

Recently, most medium to large manufacturers have used MRP II or ERP as their enterprise information systems. This section shows schematic relations and differences between these legacy systems and APS. First, a conventional enterprise information system framework is illustrated in Fig. 2-6.

In Fig. 2-6, there are three levels in the target decision-making for business activities. Each level is managed by enterprise managers, production managers, and plant floor managers, respectively. Different levels have different aspects and objectives developed by the different types of managers. Enterprise managers make decisions with respect to the profitability of the enterprise, while production managers deal with total throughput of the factory regarding plant-wide parameters such as average production lead-time, inventory levels and so forth. Ultimately, plant floor managers modify the final execution schedule in order to maintain safety and efficiency in the plant. They will know if the schedule developed by production managers is not feasible and requires more detailed information to be added at plant level. MES (Manufacturing Execution System) corresponds to this level.

Decision-making modules addressed in Fig. 2-6 are briefly described as follows.

Fig. 2-6 Legacy information system for manufacturers



(1) Business planning/sales and operations planning

In business planning, or sales and operations planning, management executives who are responsible for final decisions relating to future profitability and strategies participate. New product development, investment in production lines, hiring or laying off workers, and alliances with suppliers are discussed in this module. Results of this decision-making will act as constraints in the lower modules.

(2) Demand management/production planning

To achieve a balance between demand and supply, this module makes decisions from the demand side. Demand forecasting and sales volumes for each period are decided, together with particular actions for future demand creation. Complementary problems, such as product inventory planning and transportation of inventory in order to meet any future supply requests, are included.

(3) Resource planning/rough-cut capacity planning

This is the highest level of decision-making from the viewpoint of balancing supply with demand. From medium- or long-term aspects, the overall capacity of the factory is considered and relative actions, such as additional requests for equipment and disposition of plant capacity are planned. Personnel management for overtime work should also be decided in this module.

(4) Master production scheduling (planning)

Master production scheduling determines a master schedule in terms of quantity and item of final products of the enterprise. From a resource viewpoint, rough-cut capacity planning is implemented to confirm feasibility. Information developed in this module is shared between the sales and manufacturing divisions. Capable to Promise is developed using this function.

(5) Material requirements planning

Material requirements planning (MRP) calculates the date of production and quantities of all parts and materials associated with the schedules for final products. Bill of Materials (BOM) information is used to expand the product structure. This is a short- to medium-term, periodic decision–making cycle; however, partial re-planning may occasionally be implemented.

(6) Capacity requirements planning

Capacity requirements planning (CRP) calculates the capacity requirements for each work center, so that their individual loads do not exceed capacity limits. This is done by shifting the timing of operations that are planned for the peak of the load. Routing information for each product is required for this module.

(7) Detailed plant scheduling

Detailed plant scheduling deals with various plant floor operations. The rough schedule developed though upper level decision-making is referred to over the next couple of days. Actual status of plant equipment and particular constraints are also considered, in order to decide detailed sequences and routing for manufacture of a particular product.

(8) Dispatching

Ultimately, work orders based on a detailed schedule are dispatched to each work center or relevant personnel on the day scheduled for execution of the work. Dispatch may be carried out simultaneously for work orders applicable within a certain period, such as a shift. Otherwise, an order will be released whenever the previous operation is completed. Plant floor personnel prepare tools and setups for the corresponding production.

(9) Monitoring/tracking

All actions resulting from [work order] dispatching are monitored and stored as tracking performance data, which is used for route tracing and quality analysis of each product. Furthermore, these performance data become real-time input fed back to the detailed plant scheduling for comparison of the schedule against results. Then, work orders already executed are eliminated in the next detailed scheduling.

In Fig. 2-6, the boxes with rounded corners represent planning, which means a decision-making process with parameters on a certain time bucket or time phase. The other boxes in the figure represent scheduling, which is decision-making on a continuous time scale. The left-side modules of demand management/production planning and master production scheduling (planning) have a product-and-material-centric view, while the right-side modules of resource planning/rough-cut capacity planning and capacity requirements planning have a resource-capacity-centric view.

The conventional framework of decision-making modules described above suffers from three serious problems: (1) First, in the development of a production schedule the information flow between plant floor managers and production managers is one-way. A production manager is not fully aware of details about the actual plant floor situation. (2) Second, in both the enterprise managers' and the production managers' decisions, a module of product-and-material-centric view and a module of resource-capacity-centric view are separated. This can lead to inefficiency and inflexibility in the system. (3) Finally, it should be noted that not all of the modules have explicit collaborative relations with plant engineering, production design, and financial or cost accounting functions.

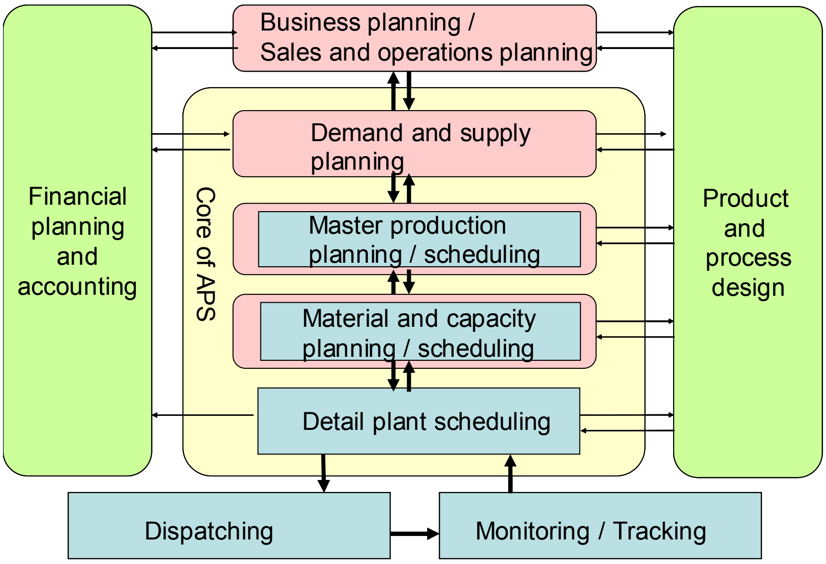
To replace the conventional system framework, PSLX proposes a new framework based on the APS concept. Fig. 2-7 shows an overview of the proposed framework, which includes demand and supply planning, master production planning and scheduling, material and capacity planning and scheduling, and detailed plant scheduling. These four modules are the nucleus of APS and correspond to the basic decision-making layers shown and explained in Fig. 2-5.

Comparing Fig. 2-6 and Fig. 2-7 demonstrates several schematic features of APS. First, master production planning and scheduling, and material and capacity requirement planning and scheduling (Fig. 2-7) have both planning and scheduling aspects. In other words, both decision-making processes can deal with continuous time scales as well as conventional time-bucket-based decisions. This is a very important feature that PSLX provides by integrating the schema of planning and scheduling.

Second, two pairs of separated modules (Fig. 2-6) are merged into single decision-making processes. Demand management/production planning and resource planning/rough-cut capacity planning become demand and supply planning. Material requirements planning and capacity requirements planning are merged into material and capacity planning and scheduling. This can be achieved with the PSLX ontology and APS domain model, which represents explicit relations between a concept of products or materials and a concept of resources. such as equipment and work centers.

Finally, Fig. 2-7 shows that APS provides mechanisms for collaboration with decision-making in financial planning and accounting activities, as well as product and process design activities.

Fig. 2-7
APS system components and structure



Target categories of business models

Design of system implementation on information system architecture requires knowledge of the classification of each manufacturer in terms of characteristics of production processes, and business strategies for sales and manufacturing integration. The overall APS decision-making system contains every kind of business process, each of which has a different context in its particular business circumstances.

This section shows management classification from the viewpoint of APS. Before going into design detail of APS, the category of the target enterprise should be clarified. First, there are three groups in terms of physical production systems.

(1) Flow manufacturing

Flow manufacturing includes continuous manufacturing, such as chemical and petroleum plants, and repetitive discrete manufacturing, such as final processes in the automotive industry. In this type, production velocity is stable and resource load is always flat. Most of the inventory in this type of production is part of pipeline inventory, and work-in-progress inventory awaiting the next production step is minimized. Scheduling of work orders in production processes is not an important issue, but line balancing and release control take place.

(2) Intermittent manufacturing

This type represents production that is repetitive but usually involves a greater variety of items and broader range of quantities. Most manufacturing enterprises come under this category. It includes discrete manufacturing that turns out several varieties in small lot sizes, and batch manufacturing runs in process industries. Management to eliminate loss of efficiency during changeover operations, and detection/control of bottlenecks are important considerations in the scheduling process. Planning issues include creating a balance between demand and supply to meet various customer requirements with minimal investment in plant resources.

(3) Project manufacturing

Project manufacturing is not repetitive, and all materials, components, production equipment and labor are procured once during the period of production, and then released. Most examples of this type of manufacturing include design processes, e.g. production of large size products such as ships and aircraft, construction of factories or production lines. Project scheduling methods such as PERT and CPM are available.

The classifications described above are defined automatically by the respective products and services of the enterprise. On the other hand, the following classifications arise out of the business strategies of the particular enterprise, with the objective of developing a competitive business model for the relevant market. Each manufacturer is able to choose an appropriate strategy.

(1) Make-to-Stock strategy

The Make-to-Stock strategy allows manufacturers to make their products according to forecasting data on projected customer orders. Once final products are produced and stored as inventory, customers draw their purchases from the finished inventory. Inventory holding cost will be much higher. However, customer order lead-times can be minimized, and production costs are also lower as a result of larger, more economical lot sizes. This strategy works better when future demand is predictable, and the products have long shelf lives, or will not become obsolete in the near future.

(2) Make-to-Order strategy

The Make-to-Order strategy, on the other hand, recognizes customer orders as a “trigger” to activate production. Manufacturers using this approach try to ship the final product as soon as the production processes are finished. However, they never initiate production before receiving an order from a customer. Therefore, order lead-times will be much longer, and inventory can be reduced to zero. Purchasing processes for materials are sometimes included in the lead-time, sometimes not. This strategy works better when the production process involves less repetition and greater variety.

(3) Design-to-Order strategy

The Design-to-Order strategy can be applied to one-of-a-kind production, in which the product does not have a stock design. Manufacturers carry out particular types of design and manufacturing according to customers' preferences, chosen from a number of options. The customizing processes may refer to types of templates or standard components of products in order to minimize production input. Customer lead-times are very long and unreliable. Standardization of components is an important factor in reducing lead-times.

(4) Finish-to-Order strategy

The Finish-to-Order strategy is an intermediate one, falling between Make-to-Order and Make-to-Stock. This features a certain stocking point between the processes from raw materials to final products, corresponding to the BOM. Processes upstream of that point are managed by the Make-to-Stock strategy, while downstream processes come under Make-to-Order operations. There is no inventory of final products in order to reduce stockholding costs. At the same time, this strategy allows manufacturers to reduce customer lead-times, as compared to the Make-to-Order strategy. This type has many advantages if the products have BOM, in which product variation can be implemented in the final steps of the production processes.

Chapter 3
Business Activity model

What are business activities?

A business activity as defined in PSLX is a unit of business that is necessary from a manufacturing perspective. Many other system architectures have similar definitions, but those classify activities in a top-down manner, because they have modules of information systems according to a descending hierarchy of activities. Instead, PSLX Consortium focuses on bottom-up approaches, in which various kinds of exceptional information are involved.

PSLX business activities have to deal with all decision-making information related to planning and scheduling. In other words, such business activities are all about business processes based on plant floor management. In particular, management of detailed information, which has life cycles including creation, revision and disposition on plant floors, should be carried out in PSLX business activities.

Granularity of business activities corresponds to the unit sizes of business process elements. This is different from the activities in UML activity models. Business activities cannot have hierarchy, and are defined without any overlap between them. Mapping from an activity of real-world problems to a model of business activities may feature multiple paths.

Each business activity has at least one “use case”, which is a sequence of actions for achieving certain objectives within the business activity. Use cases define interactions between circumstances and the business unit that is in charge of the business activity. The interactions incorporate physical processing and information processing.

When there is a need for certain use cases to interact with other use cases in different business activities, those interactions across business activities are defined as “collaboration”. There can be a huge variety of collaborations in real-world business. Therefore, PSLX defines only the typical and useful collaborations in the specifications.

Business activity model

In order to define business activities as a PSLX specification, two comprehensive views are introduced. First, an administrative view shows particular objectives depending on the different business aspects. Groups divided according to the administrative view have their own information depending on the business context, making it easy to create a business organization with respect to this view. Outsourcing management, production management, inventory management, transportation management, personnel management, plant maintenance, quality management, cost management, and plant engineering are addressed in the administrative view.

The second view is a functional view, which is complementarily defined to classify more detailed functions in manufacturers' business processes. This shows functional aspects of decision-making, such as how each activity relates to planning and scheduling. The functional view of PSLX includes plan management functions, order management functions, schedule management functions, execution management functions, performance management functions, and technical information management functions.

Relationships between the administrative view and the functional view can be established for every combination of each of those two. In other words, each node in the 9 X 6 matrix, shown in Fig. 3-1, is the intersection between the administrative view and the functional view, respectively. Business activities defined by PSLX are a conceptual classification of business activities, corresponding to these 54 areas of intersection.

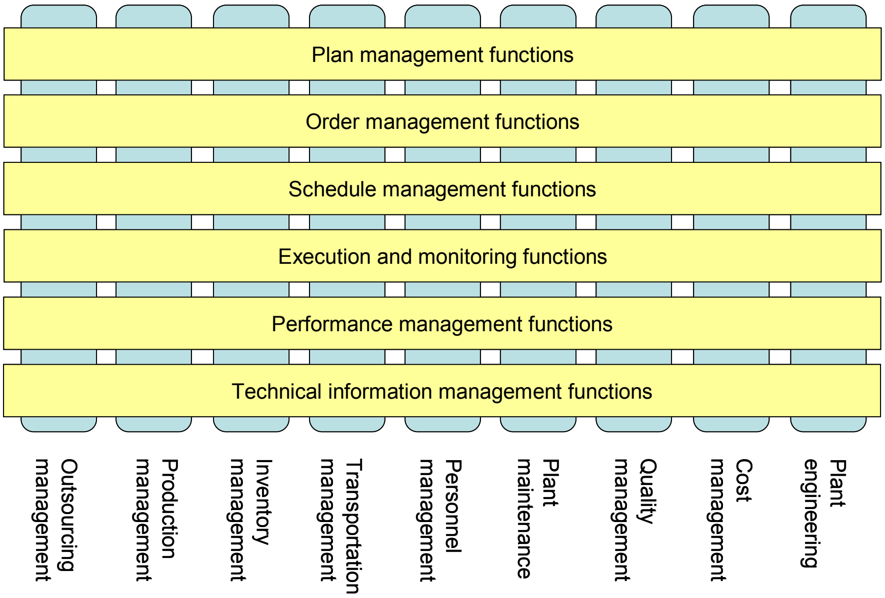


Fig. 3-1 Granularity of business activities

Depending on industrial segments, PSLX tries to define particular use cases for each business activity. At the same time, data models necessary for each business activity and corresponding interfaces for collaboration among business activities are defined in PSLX. Collaborations are either vertical, which is within an administrative view, or horizontal, which is within a functional view. Use cases, data models, interfaces and collaborations are not mandatory but are very useful in the design and review processes of system development to realize interoperability between software applications across enterprise business processes.

Administrative view of activities

This section briefly introduces the administrative view of business activities. The following nine administrative groups are defined:

(1) Outsourcing management

Outsourcing management deals with activities conducted outside the enterprise, such as parts suppliers and manufacturing service providers. Activities in which control of detailed operations resides with other companies are managed, even if they are geographically located within the factory. This management is concerned with not only the purchasing of supply parts and materials, but also procurement of production resources such as equipment, personnel and tools.

(2) Production management

In production management, the target activities are various operations that transform parts or materials into other valuable items by changing their shapes and qualities. For these operations, efficient use of manufacturing resources such as equipment and personnel is organized to meet production orders. This includes dispatching of production work orders, Takt time monitoring and throughput control, bottleneck process management, work-in-progress inventory or Kanban control, etc.

(3) Inventory management

The quantity and status of all items produced or consumed by production processes are managed by inventory management. Those processes include all related operations from the point of receiving materials made by suppliers to the point of delivering products to customers. Materials inventory, finished product inventory, and components or intermediate parts inventory in storage zones are effectively controlled by managing their issue orders.

(4) Transportation management

Consistent with production, transportation management makes decisions about the movement of resources and materials between different work centers or different areas of plants. Methods and routes of transportation, resources for transportation, and lot size of transportation are discussed. Inter-enterprise transfers of products or materials are also included if the delivery processes are under the control of the management. The objectives are to provide the right quantity of right items to the right place, at the right time.

(5) Personnel management

Personnel management deals with every person who is responsible for a production process, either directly or indirectly. It covers the working hours and performance of each worker, qualifications and skills for particular operations, shift and group assignments, standard time analysis, training and instruction, etc. Maximizing the output of every worker in the enterprise is the goal of this management process.

(6) Plant maintenance

Plant maintenance activities focus on maintaining the performance of plant floor resources as designed. In order to do this, plant maintenance deals with monitoring and tracking of resource status, historical data on resources, preventive action for malfunctions, analysis of performance data, utility and supply parts for equipment, inspection schedules, etc. In the event of accidents or malfunctions, plant maintenance activities operate to remedy problems as soon as possible.

(7) Quality management

In order to improve the quality of products and processes, quality management operates to eliminate errors that could reduce quality below the designed level. The targets of quality inspections are not only products and materials, but also include production processes and their resources and methods. For example, Total Quality Control (TQC) activities detect the sources of problems, using statistical analysis, and implement remedies to improve quality.

(8) Cost management

Cost management considers appropriate resource investment for equipment and personnel, in terms of cost and throughput. The performance of each resource is monitored relative to the designed specification and accounting information. Cost management can also deal with strategic decision-making for dynamic pricing in accordance with market reaction to the product. This requires knowledge of the real production cost of the product.

(9) Plant engineering

Plant engineering constructs actual plant and production processes that produce the products required by customers. Based on substantial engineering knowledge and experience, plant engineering manages and executes the development of a trial line for preliminary production. It also deals with improving the performance of the final production line for customer products. Information necessary for production and maintenance, such as instruction manuals, equipment specification documents, and engineering drafts are prepared.

Functional view of activities

The functional view of business activities defines horizontal aspects across every element of the administrative view. Functions addressed in this view are:

(1) Plan management functions

Long- or medium-term activities are managed by plan management functions. Target parameters to be achieved by activities are defined for each time phase (days, weeks, or months). Plan management functions operate to develop a plan, revise a plan, compare the plan against the result, store the history of the plan, etc. They also facilitate collaboration and coordination between enterprise divisions, to achieve enterprise-wide optimization.

(2) Order management functions

For each administrative element, orders to request particular actions are managed by order management functions. Orders include incoming and outgoing external orders, as well as internal orders created to provide greater detail. Order management functions create and revise orders and their relations, dealing with the reliability and probability of orders, which can change according to circumstances.

(3) Schedule management functions

Actions related to administrative activities require both resources and removal of constraints on execution. In particular, potential conflict between different actions should be resolved in advance. Schedule management functions are defined as scheduling activities on plant floors, taking into account various constraints and objectives. They also manage the results of such scheduling to collaborate with other administrative activities.

(4) Execution and monitoring functions

Taking into account real-world states, actions and operations on plant floors are finally executed in accordance with a schedule. Execution and monitoring functions manage these processes, and also monitor their performance. When exceptions are detected, they implement predefined procedures to remedy any problems and, where necessary, request revision of the schedule. Subsequently, a report on the event is forwarded to relevant business divisions.

(5) Performance management functions

Actions and operations carried out on the plant floor are recorded and their histories are filed for possible future reference. Data gathered from plant floors comprise information appropriate to requirements, which is duly aggregated and analyzed from various standpoints. Data obtained are also valuable in tracing materials flows in the plant in order to identify the causes of particular problems. Evidence of quality assurance also requires these data.

(6) Technical information management functions

These functions manage technical information for each administrative element. Technical information includes parameters or indicators for evaluation, specifications of products and processes, documents and engineering drafts, operation manuals, etc. These data are managed in conjunction with their revision history. Technical information that needs to be shared between different business divisions is managed with respect to consistency, negotiation with stakeholders, and data security.

Business transactions between activities

As mentioned in previous sections, PSLX defines nine categories under the administrative view and six categories under the functional view, which are aggregated to fifty-four elements of business activities. There are at least 54 X 53 potential combinations of types of data transactions between these business activities. In order to reduce complexity, the number of combinations is reduced by applying a rule that permits data exchange only if the two parties are in the same category. Thus, data transactions within the same category in the administrative view, or within the same category in the functional view, can be defined. In addition, some transaction combinations are ignored, depending on the context of the actual business situation.

Transactions between business activities are defined as special cases of collaboration, since business activities can collaborate with others in the same category under the administrative view or the functional view. Collaboration within the same category under the administrative view is referred to as vertical collaboration, while collaboration under the same functional view is referred to as horizontal collaboration.

As a general rule, a description using UML collaboration models contains several transactions between two or more actors. However, PSLX allows definition of transactions as common specifications to compose various collaborations in practical business processes, such as:

(1) Invoking a receiver’s activities

This is a case where one activity invokes a use case of another activity. This transaction can initiate a particular use case or resume from a breakpoint in a use case that is executing in different activities. The initiator needs to send detailed information on options and preferences to invoke the functions, while the reactor responds with the results of execution of the request.

(2) Pulling information for sender’s activities

This is a case where one business activity requests certain information from other activities. An information query is a typical example. In this transaction, information is either stored in a certain place managed by the other activity, created for each request using stored data, or obtained through further collaboration with another activity.

(3) Pushing information for receiver’s activities

In this case, one business activity sends information to the other. Recipient business activities need to have a mechanism for reacting to any requests from initiators of such a transaction, including a rejection mechanism. An initiator sends information to the reactor with additional information to identify the context of data for interpretation in a particular business situation.

The three transaction patterns described above are types of business protocols of collaboration between different business components. This should be distinguished from data communication protocols such as TCP/IP and SOAP, which are beyond the scope of this specification. It is also necessary to define security-related protocols; however, PSLX does not have a specification for security.

In PSLX specifications, business collaborations are defined at the business transaction level, rather than being complete definitions of business collaboration. However, it will soon be possible to access information on this subject as a result of publication of PSLX business collaboration best practices.

Chapter 4

Specifications for system implementation

Business transaction framework

In real-world business situations, data transfer between two business activities can be carried out by people without the aid of computers; for example, through paper documents containing typed, printed or handwritten information. Today, however, most information exchanges are made via computers and networks. Some are automatic, some require human involvement. In PSLX specifications it is assumed that all information exchanges between business activities require explicit representation schema to enable transfer to be conducted partially or totally by computers. There are two types of implementation methodology for data exchange in a business transaction:

(1) Storage type communication

The first method is storage type communication to exchange data. In this type, data are stored once in a database, and then sent or received via the database. Relational databases are popular tools for all kinds of business information system, and are also useful in conducting this type of data communication between different business activities. The PSLX-RDB common schema is a standard specification for system implementation to facilitate this data exchange.

(2) Messaging type communication

The second method of data exchange is messaging type communication. This is where two application programs operating on different business activities exchange data directly. In this case, the data in each message are generated solely for the purpose of data exchange, and then scrapped. Basically, one message is generated for one purpose, on one business transaction. PSLX defines a standard specification for this type of data exchange in the PSLX-XML common schema.

As a general rule, different business activities need to share common business protocols in order to make a data exchange. “Business protocol” defines rules of communication at business level, including exception processes that apply when communication fails. More importantly, it needs to deal with the issue of the data being able to be properly interpreted, as intended by the initiator of the data exchange.

Both the PSLX-RDB common schema and the PSLX-XML common schema feature a data schema that represents a particular data structure, in which the correct meaning of all elements should be understandable in each business context. To support clear understanding of schema semantics, PSLX defines PSLX Ontology.

PSLX Ontology

An ontology is a fundamental element in representing the semantics of information related to problems in a certain domain. Since the target domain is decision-making processes in planning and scheduling for manufacturing industries, PSLX specifications have defined PSLX Ontology, which provides a basis for common understanding of information about planning and scheduling.

PSLX Ontology is defined by a structure which corresponds to the unique and fundamental structure underlying problems of planning and scheduling. In other words, the semantics of an ontology is represented implicitly by the structure, with respect to its similarity to reality. A term named for an ontology is simply a label used to identify the ontology in the semantic structure, and is not precisely concerned with the correct meaning. An additional intuitive explanation for an ontology is also helpful, but not essential.

In PSLX specifications, an ontology is represented using UML class diagrams. However, this doesn’t mean that the ontology is a kind of class defined in object-oriented modeling technologies. The ontology is used to define the semantics underlying classes. In class diagrams, the ontology is classified into several types, which can be observed provisionally in the form of stereotypes. The types of ontology are introduced in order to properly establish the meaning. An ontology does not have any attributes or methods.

Information that represents any elements in planning and scheduling for manufacturing should be defined using PSLX Ontology, to ensure common understanding of problems between different parties. In other words, all information objects dealt with in business activities should have specific relations to the ontology structure. A PSLX specification shows details of the process to describe the relationships, using UML diagrams.

Aspects and structure of ontologies

PSLX Ontology can be classified according to the following aspects, each of which has common features. The classifications are used to decide a stereotype for a class that is directly defined using one of these ontologies.

(1) Engineering aspect

This aspect represents engineering knowledge or know-how that is obtained through long experience in the domain. Information in this aspect is stored in advance of its usage. Then, it is repetitively retrieved and applied to problems. This aspect has four ontologies: Function, Capability, Item, and Event.

(2) Scheduling aspect

The scheduling aspect deals with the relationship of actions with time and space, which are necessary to realizing actions and operations in the physical world. In other words, an object in this aspect is an instance of an object in the engineering aspect with respect to particular time and space. Time includes the past, present and future. This aspect has four ontologies: Operation, Task, Lot, and Action.

(3) Temporal aspect

This aspect represents physical features from the viewpoint of changes in the time horizon. Physical phenomena of an object can be described temporally by objects in this aspect. Physical phenomena are the results of interaction between a physical object and operations defined in the scheduling aspect. In this aspect, three ontologies exist: Capacity, Inventory, and Change.

(4) Physical aspect

The physical aspect is defined for physical objects that exist in reality. This may contain a conceptual object if it is not duplicatable. Usually, objects in this aspect occupy certain places during certain periods of time in the real world. The objects in this aspect do not have a tense, because they are qualified by objects in the temporal aspect. Resource is an ontology that exists in this aspect.

(5) Order aspect

Every action and operation on plant floors is created and executed with respect to orders. This aspect is defined for orders in all kinds of situations. One order may create other orders. All intentional changes in enterprises are provided by orders. Information flows in business processes can be defined as order propagation flows, which execute actions along the flows. This aspect has three ontologies: Work order, Lot order, and Task order.

(6) Plan aspect

Business activities activated by order propagation in a business process are evaluated as to whether or not they are balanced and consistent with others from a medium- or long-term viewpoint. The plan aspect deals with conceptual parameters, objectives and goals associated with a certain period of time. Constraints, preferences and functional relations can be applied to these objects to obtain an optimum solution. Ontologies included in this aspect are Production plan, Capacity plan, and Inventory plan.

(7) Party aspect

The party aspect is defined for objects that can perform autonomously with their own decisions and preferences. The granularity of an object depends on the fact that final goals should be shared within the object. This aspect can also represent a type of ownership of resources in business processes between enterprises. This aspect has three ontologies: Maker, Supplier and Customer.

(8) Time aspect

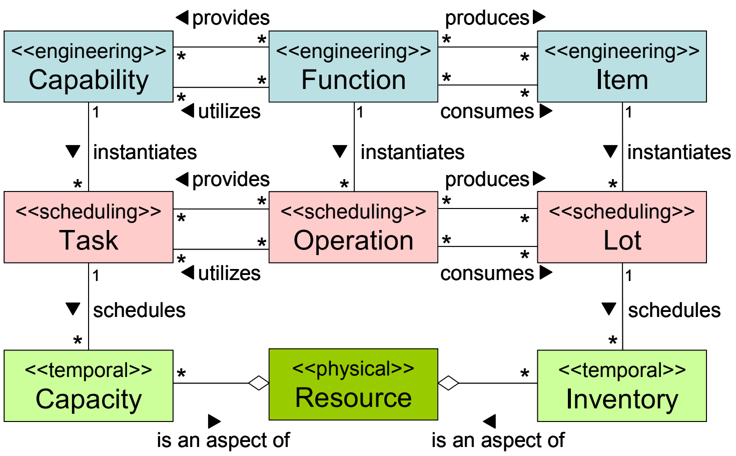
Although it cannot literally define "time", the time aspect defined by PSLX is simply used to provide useful classifications for concepts related to time, where definition of time is assumed. This aspect has three ontologies: Time point, Time period, and Time span.

(9) Space aspect

Similar to the definition of time, a definition of space is also difficult to establish, but is a highly important aspect of planning and scheduling problems. The space aspect is defined to classify basic elements of the concept related to space. This aspect is useful in determining the granularity of objects from a spatial or geographical view. There are three ontologies in this aspect: Position, Region, and Distance.

PSLX defines 27 ontologies in the nine aspects described above. Ontologies have relations with each other, representing their semantics depending on their positions. The semantics are shown in the following figures, with explanations of the special meaning of each position.

Fig. 4-1
PSLX
ontology (1)



First, Fig. 4-1 shows several basic aspects of ontology and their relations. Engineering, scheduling, temporal, and physical aspects are connected by relations with particular roles and cardinalities. For example, it can be seen that ontologies in the scheduling aspect are instantiated by ontologies in the engineering. Ontologies in the temporal aspect exist between the ontologies in the scheduling and physical aspects.

The right column in Fig. 4-1 represents something that is produced or consumed by production. This is an aspect of an ontology that can fix a particular space in the real world. On the other hand, the left side of Fig. 4-1 represents something that is utilized by production. This is an aspect of an ontology that can fix a particular time, so that the ontologies cannot move across different periods. For example, capacity of resources is utilized within a particular period of time.

Fig. 4-2
PSLX
ontology (2)

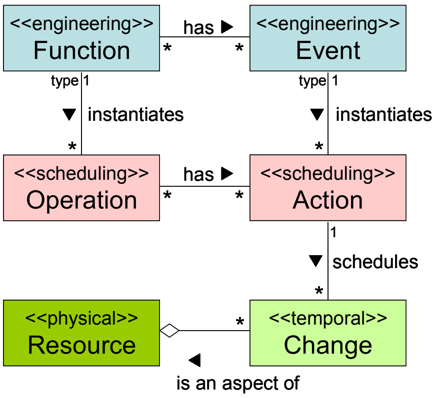


Fig. 4-2 is a complementary chart to Fig. 4-1, in which the right column is added. The additional column represents the most primitive view of those three aspects. While the left-hand column in Fig. 4-2 (center column of Fig. 4-1) associates with the concept of time span, this column associates with time point. For example, a concept of Operation can have a concept of Action between its start and end if it has several different phases.

Fig. 4-3
PSLX
ontology (3)

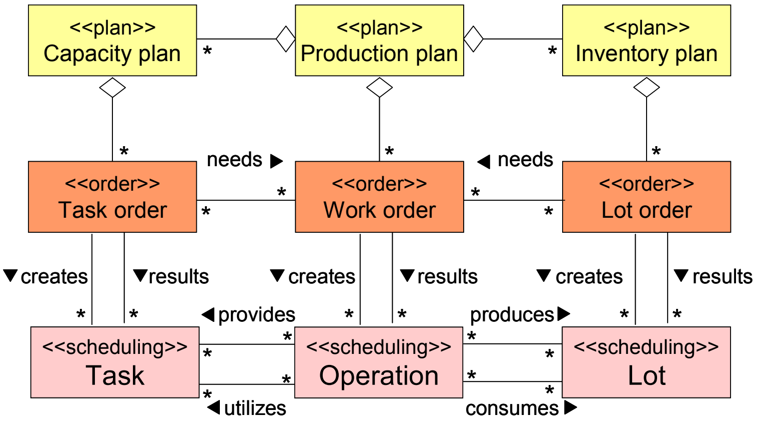


Fig. 4-3 represents relations of the other two aspects: plan and order. As shown in Fig. 4-3, ontologies in the order aspect create an ontology in the scheduling aspect, while ontologies in the plan aspect are aggregated from the ontologies of orders. Similar to Fig. 4-1, this chart also has special meanings for the right column and left column. In addition, it can be noted that there is no ontology in the plan and order aspects that corresponds to the right column of Fig. 4-2.

APS domain model

An ontology is a universal definition that can be shared with different standards in this area. However, it is so abstractive that users who represent something using the ontology take time to understand correctly. Therefore, it is necessary to have more detailed vocabularies for general use in production planning and scheduling that allow an object to be chosen easily.

An abstract data model has a comprehensive set of objects and relations for this purpose. This is independent from the form of information system implementation. This section introduces the APS domain model, which is a common reference model for abstract data models. Each manufacturer can easily define a particular abstract data model according to its unique business model by using the APS domain model.

The APS domain model is represented in UML class charts. All classes in the model should have some connections with PSLX Ontology, which can be drawn as a path in the chart. In conclusion, the definition of class in the APS domain model should be a:

(1) Class corresponding to an ontology;
(2) Sub-class of (1);
(3) Aggregate class of (1); or
(4) Class that has direct relations with a class of (1), (2), or (3).

In addition to the ontological relations, two classes as defined above (1) - (4) can have particular relations, each of which must have a role and cardinality. Classes in the APS domain model, as well as abstract data models, do not have attributes or methods.

The APS domain model is partially shown in Fig. 4-4 to Fig. 4-7. First, Fig. 4-4 has an operation class that corresponds to the ontology, and its sub-classes: Manufacturing, Setup, Inspection, Transport, Administration, Product design, Plant engineering, Maintenance, Ship goods, Receive goods, Issue inventory, and Store inventory. An asterisk beside a stereotype denotes that the class directory connects to the ontology.

Fig. 4-4
Sub-classes of operation

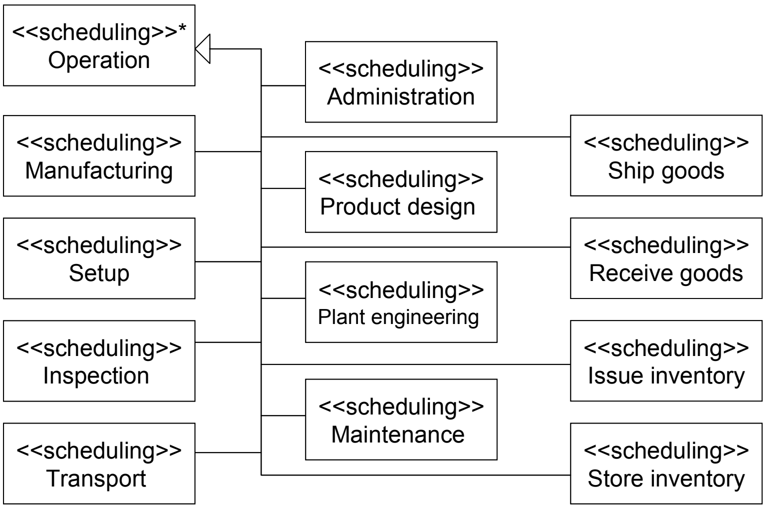
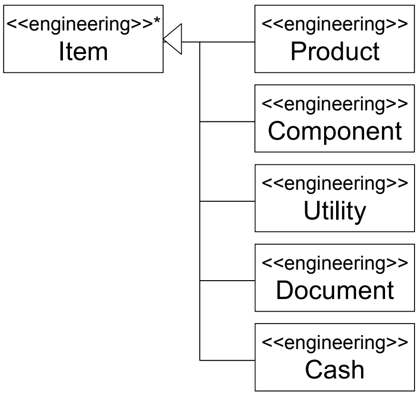


Fig. 4-5 shows an item class and its sub-classes, where the item class has a direct relation to the ontology. All the classes in the chart have the engineering aspect, and represent some features of identical substances or concepts at an abstract level. Sub-classes of item include Product, Component, Utility, Document, and Cash.

Fig. 4-5 Sub-classes of item



A class corresponding to the resource ontology and its sub-classes are illustrated in Fig. 4-6. Sub-classes of resource are defined as Personnel, Equipment, Tool, and Material. Equipment has further sub-classes of Machine, Tester, Route, Storage, and Supporter,

Sub-classes of resource have two different functional characteristics. One is consumable resources such as materials. The other is kinds of non-consumable resources that can remain during production. Most resources can be classified according to one of those characteristics, although a few classes have both. Fig. 4-6 represents the characteristics by connecting to either Capacity or Inventory, while classes connecting to Capacity are non-consumable, and classes connecting to Inventory are consumable.

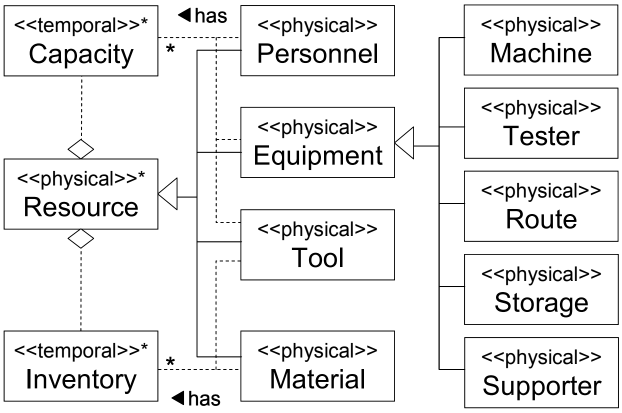


Fig. 4-6 Sub-classes of resource

Many classes that directly connect to an ontology have an aggregate class, as shown in Fig. 4-7. These aggregate classes are useful when a decision-making process has hierarchy, as in planning and scheduling. In fact, most of the basic classes in Fig. 4-7 are typically used in scheduling problems, and the aggregate classes are useful representation for planning problems.

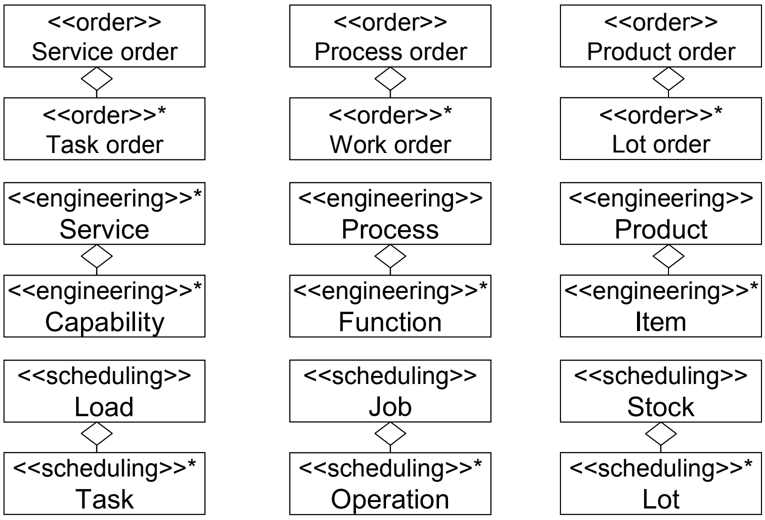


Fig. 4-7 Aggregation classes for planning

In the order aspect, we find Task order, Work order, and Lot order at the level that directly corresponds to the ontology. These have aggregate classes of Service order, Process order, and Product order, respectively. Similarly, in the engineering aspect, Service, Process, and Product are aggregate classes of Capacity, Function, and Item, respectively. Finally, the scheduling aspect has Load, Job, and Stock as the basic ontological classes of Task, Operation, and Lot, respectively.

In PSLX specifications, all the classes in the APS domain model are explained in detail using UML class charts.

Standard schema for implementation

While the APS domain model deals with the correctness of knowledge representation in various kinds of practical business activities, the data schema referred to in a computer system needs to have a very application-specific view. In general, each information system has its own implementation schema to represent business data in the form of each application system. In order to realize interoperability of the data, there needs to be another kind of data model that shares the common format of the implementation-specific schemas. This common schema represents information in the APS domain model. PSLX defines two common schemas for implementation: the PSLX-RDB common schema, and the PSLX-XML common schema.

Similar to the APS domain model, the two common schemas defined in PSLX are based on PSLX Ontology. The common schemas should be created using the steps described below. In other words, all the classes in the common schema can be explained by referring to the APS domain model. One difference to the APS domain model is that the common schemas can have attributes.

(1) Change the name of a class
(2) Merge two classes
(3) Create a class instead of a relation
(4) Create an attribute instead of a relation
(5) Delete a class
(6) Add a new class
(7) Add a relation between two classes

(8) Add an attribute to a class

The steps above have the same sequence as those in the list. Furthermore, steps never have to be redundant. In other words, it should not to have other shorter sequences of steps that get the same schema from the APS domain model. The detailed rules of the schema generation are defined in the PSLX specifications.

PSLX-RDB common schema

A PSLX-RDB common schema is a [shared][common] schema for a business transaction in storage type communication. In this case, the PSLX-RDB schemas correspond to views of a particular RDB, while the views are created by SQLs. A unique feature of a PSLX-RDB common schema is that every class in the schema does not have an explicit relation with others. The relations behind the classes in the schema are defined by referring to the APS domain model. Every class and attribute can be identified by knowing the schema generation steps from the APS domain model.

As mentioned earlier, RDBs implemented in information systems need to provide a view corresponding to a class in the PSLX-RDB schema as a common interface between different applications. A systems engineer who is responsible for the implementation needs to know the relations between a table in the RDB schema and a view that corresponds to a class in the PSLX-RDB common schema. He or she also needs to know about more detailed relations between each field of the table and a field in a view that corresponds to an attribute of the class in PSLX-RDB schema.

There are a huge variety of classes in a PSLX-RDB common schema, because of the number of different business activities. Therefore, some useful categories have been introduced, as follows. The details of the classes are described in the PSLX specifications.

(1) Master classes

Master classes directly relate to classes in the APS domain model. Data in a class act as master data, which is referred to by business activities. This group includes customer master, destination master, supplier master, capacity master, personnel master, equipment master, tool master, storage master, operation master, transport master, process master, product master, material master, item master, area master, work center master, and route master.

(2) Relation classes

Relation classes represent relations between two classes in the APS domain model. Because of the technical constraints of RDB, relations need to be another class if the cardinality is more than one. Relation classes include bill of operation, bill of resource, bill of process, bill of area, bill of material, resource table, process table, precedence table, and loading table.

(3) Order classes

Order classes cover various kinds of orders in business activities, corresponding to the order class and its sub-classes in the APS domain model. The following classes belong to this type: customer order, purchase order, production order, work order, ship order, receive order, issue order, and store order.

(4) Planning classes

Planning classes are used to represent data in terms of plans, such as goal parameters in business activities and aggregated results of those activities. These are defined with respect to a certain unit of time phases. The planning classes include: production plan, master production schedule, material requirement plan, rough-cut capacity plan, capacity requirement plan, demand plan, inventory plan, and logistics plan.

(5) Scheduling classes

Scheduling classes deal with certain actions on the time horizon, defining information associated with any kind of operation on plant floors. All the classes in this group contain information about a particular time. The scheduling classes include: factory calendar, equipment calendar, personnel calendar, resource capacity, lot data, task data, product inventory, material inventory, resource load, area load, and tracing data.

(6) Performance classes

Finally, performance classes are defined to classify classes that represent results of any business activities. Some of these are obtained by monitoring activities on plant floors, whole others are results of commitments, as in a business situation. This group includes: order pegging, lot pegging, work result, production result, resource result, receive result, ship result, store result, issue result, send invoice, and receive invoice.

PSLX-XML common schema

A PSLX-XML common schema is defined for data exchange between different applications in different business activities by means of messaging type communication. This type of communication is becoming popular even between different computer environments. Examples include Web-services technologies for Web application servers and Web clients on the Internet, and XML/EDI for inter-enterprise information exchange using a flexible data format rather than the fixed length format of conventional EDI.

Classes defined in the PSLX-XML common schema have hierarchical structures, because XML basically is a data specification language for documents. In the schema, network type relations are implicitly defined by the identifiers of each data item and their references. However, those relations should be managed in the APS domain model, outside of the formal definition of the XML schema. The identifier is described as an attribute of a class, while the reference data are also an attribute in the other class. These relations are usually one-way.

The top-level classes in the PSLX-XML common schema correspond to the level of transactions between business activities. A feature of this schema is that a class that corresponds to the same class in the APS domain model can have additional constraints, depending on the context of the business activities. The number of classes in the top level is large, but the number of classes in the second and lower levels is relatively small.

The following are all the classes and their categories of the PSLX-XML common schema, excluding classes in the top level.

(1) Primitive elements

Primitive elements correspond to the PSLX ontology. These are described in the second level of hierarchy (top level of a message). There are nine elements: customer, supplier, item, resource, function, operation, order, lot, and task.

(2) Administrative elements

Administrative elements are also described in the top level of a message. These are used as common subsidiary data among the primitive elements. In this group, scale and stone (milestone) are defined.

(3) Administrative properties

Administrative properties, on the other hand, define some subsidiary data as an attribute of the primitive elements. These data are required by decision-making processes. This group includes description, priority, and display.

(4) Relational elements

Relational elements represent relations between the primitive elements. Ten classes are included: produce, produced, consume, consumed, assign, assigned, predecessor, successor, pegging, and partof.

(5) Property elements

This category is for any classes that represent attributes of the primitive elements. All of the attributes can have a temporal aspect. There are seven classes: spec, location, progress, load, stock, available, and calendar.

(6) Temporal elements

Temporal elements define a class or sub-class of event[s] in the APS domain model. There are five classes in this category. Classes named start, end and event are related to an operation, while release and duetime are temporal elements related to an order.

(7) Basic data elements

The basic data elements are defined for any basic data that are necessary to describe a value of parameters. The basic data elements have five classes: qty, price, char, duration, and time.

(8) Data auxiliary elements

The data auxiliary elements are additional information for the basic data elements. Classes in this category define a range of data. Seven classes exist: min, max, earliest, latest, shortest, longest, and enumerate.

The classes in the top level of the PSLX-XML common schema are defined in PSLX specifications in accordance with the business collaborations and transactions. One example is a class named “pslx”, which can incorporate every class described above. The detailed information of classes in the second and lower levels, which are the main concern of XML tag representation, is also defined in the PSLX specifications.

Chapter 5

How to use PSLX specifications

Stakeholders in the standard

The PSLX specifications are used as a tool for implementation of management systems in accordance with the APS of manufacturing industries. Furthermore, the specifications act as a form of common rules to provide greater benefits for all stakeholders. The expected benefits of applying the PSLX specifications can be listed as follows:

(1) Support for realizing successful grand design for APS implementation
(2) Support of system life-cycle management and evolutional development
(3) Establishment of interoperability between different applications
(4) Realization of system extensibility and environment for add-in software
(5) Facilitation of user-oriented system development and continuous improvement
(6) Sustainable support for business data management and re-design

processes

The impact of these outcomes of the specifications differs according to the stakeholder. Advantages to respective stakeholders are:

(1) CEOs of manufacturing businesses

For CEOs or executive personnel in manufacturing enterprises, the specifications provide detailed information on the competitive advantages and functionalities of an APS. The information depends on technical background, so that the recommendation is not some kind of sales pitch, but rather a reliable guide to decision-making. Investment in an enterprise information system is significant and can have a huge impact on the future of the enterprise. The specifications guide the directors in developing a clear vision and grand design for the enterprise.

(2) Business process designers

For business process designers who understand and improve the detailed business processes in a manufacturing enterprise, the specifications point the way to a clearer definition of their processes, facilitating discussion of such knowledge with a broader range of people in the enterprise. This converts implicit knowledge of the processes into explicit information to facilitate collaborative design of a business process between different business divisions. Moreover, it becomes possible to efficiently manage data used across several business activities by clarifying the primary division of business activity that will be responsible for managing the life cycle of the data, from creation through revision, storage and disposal.

(3) Information system designers

First, information system designers can make decisions as to the priority of any system developments in the enterprise with respect to the total balance to be achieved in the business model. Some systems may need to be designed from scratch, while others can be developed by modifying legacy systems. Using the specifications, the designer can analyze and select best-of-breed software or IT service providers, comparing in detail the requirements of the enterprise and available external solutions.

(4) IT consultants/business consultants

For business consultants/IT consultants specializing in manufacturing industries, the specifications provide a fundamental knowledge environment in which to initiate discussions with their manufacturing clients. Common terminologies and shared understanding of business activities on plant floors facilitate progression to more detailed and company-specific discussions. Furthermore, when they seek to implement a formal analysis of enterprises, or benchmarking, the specifications enable them to evaluate a number of quantitative factors by focusing on the detailed structure of the business model, instead of basic and largely conceptual qualitative factors.

(5) Systems integrators

Systems integrators, who deal with numerous different software applications to arrive at solutions for particular problems in different industries, gain an advantage from the specifications by being able to make significant reductions in the labor needed to achieve interoperability between systems. Such interoperability is one of the most important issues in PSLX; accordingly, a number of technologies for interoperability are proposed in the specifications. The business approach of systems integration companies will become more knowledge-intensive and value-added.

(6) IT package vendors

For IT package vendors who provide software packages as best-of-breed IT solutions in their specialized application fields, the PSLX specifications can expand business opportunities and market size. If they develop an interface of PSLX common schema, the software can connect to third party products. As the number of the applicable third party software increases, business also increases. Small and medium-size manufacturers emerge as a new market for their services.

The remainder of this chapter describes some typical cases in which the above stakeholders are trialing PSLX specifications in their specific contexts in real-world business situations.

Engineering for current business processes

This section introduces the case for considering the grand design of a manufacturing enterprise in terms of the fundamental structure of decision-making systems specifically related to planning and scheduling in accordance with the PSLX specifications. It looks at redesign of the business model with suitable business processes and reallocation of decision-making modules, appropriate to the current categories of business models in particular business environments. Appreciation of best practice in this field will aid understanding of this topic.

In designing a business model for enterprise decision-making, the planning horizon and the planning cycle are important factors in each decision layer. This is the first step in determining an overall decision-making structure. The collaboration patterns of planning and scheduling should also be designed in advance, while the granularity of planning decisions is decreasing. All decision-making systems for all business processes in the enterprise are based not only on well-formed computer systems, but also on people-centered business decision systems, wherein both human and computer interactions can capture real and precise business information.

The business processes so designed need to be described in terms of use cases and collaborations between particular business activities. In design processes, business activities are connected to create a flow of information, which results in a business process. First, both the “to-be” model, which is the final goal of the enterprise, and the “as-is” model, which is the current status of the enterprise, are described and compared. Then the designer should identify an intermediate stage of the business model that is possible to achieve early in the system development process. A step-by-step future development plan is also necessary in order to achieve the final stage defined as the to-be model for the enterprise.

One of main features of APS defined by PSLX is an operational level collaboration between business activities using scheduling software applications. This includes not only horizontal collaboration between different area schedulers on plant floors, but also vertical collaboration between the plant floors and business planning divisions, which deal with more aggregated but wide-ranging information. Scheduling in business divisions is centralized decision-making, whereas scheduling on plant floors represents decentralized and distributed systems. Vertical collaboration between these different systems involves such a variety of patterns so that it requires the support of specialized PSLX knowledge.

Conventionally, the systems knowledge of a business process and that of an engineering or manufacturing process are divided into two individual systems. However, PSLX discusses those two separate worlds on a single APS system platform. The APS domain model defined by PSLX allows the designer to represent both business process knowledge and engineering knowledge at the same time, because there is no distinction between the two. This means that PSLX can integrate the interdisciplinary area between engineering and business as a practical and value-adding application field of information technologies.

Profiling the various software systems

Information systems in current manufacturing enterprises are very complex and difficult to implement for information systems support personnel. Even an IT solution vendor cannot keep abreast of all the technologies that need to be applied in an enterprise-wide system development project. Therefore, the best-of-breed approach, in which many IT solutions are combined in a complete system, performs very well in such a diversified environment. However, there is a problem with this approach, in that many kinds of software feature a huge variety of functions, which are difficult to understand in the context of each manufacturing enterprise unless the user invests considerable time and money in actually buying and evaluating the system. As a general rule, finding appropriate software is a difficult and potentially risky process.

The PSLX specification for defined business activities is used as a software profiling tool that shows software functions and application fields in detail. This is valuable for manufacturers when soliciting proposals from IT vendors or systems integrators. It also benefits IT package vendors when promoting their products, and systems integrators when proposing systems to their clients. The procedure of system profiling is as follows.

First, all the business activities defined in the PSLX specification are checked to determine whether or not the target software covers the relevant functions. The corresponding areas of business activities are usually adjacent to each other within the same view of administrative or functional classifications. If the software contains several independent areas on the PSLX business activity model, then it needs to be divided according to those areas, so as to achieve clear definition in the software.

The second step is to elaborate use cases registered in the business activities in the target area of the software. If there is no use case that represents the particular requirement, a new use case should be described in the same representation format. The new case can be submitted for registration as a PSLX common use case. The third step involves describing business collaboration by referring to the PSLX specification. This has some similarity to Step 2, in that the corresponding specifications are selected and elaborated for the system.

Finally, the system interfaces to other business activities that are not a part of the system but are within a same collaboration defined by PSLX should be described as part of its functional specifications. Specification of the system interfaces is also required if a collaboration is within a business activity but there are several distributed software modules for the same business activity across the target manufacturer's operations.

Data management and continuous improvement

In typical enterprise information system development projects, the process that consumes the greatest time and effort is practical data preparation for regular execution of all business processes in accordance with the information system. This process is more complex than a specific software module customization for each requirement because it requires the involvement of end users in creating various new data. PSLX specifications show the divisions responsible for each set of data, while business activities in the PSLX specification are mapped to particular software functions, and data items necessary for the activities are clarified. Therefore, it is possible to start preparing data for the new information system before development has been completed.

When implementation of a new business process is planned for competitive advantage purposes, the cost of gathering data to maintain the process is an important factor, as is the cost of software module development. Even if a completely new software system suite is chosen, most data managed in the system suite are of a conventional type, collected and modified from the current business activity. Data migration support thus becomes a very important process in system development. This is a kind of business data centric approach to software system development for particular systems that involve a significant number of human decision activities.

An example of reverse engineering of a legacy system is used to explain a case of application of the PSLX specification. First, similar to the process described earlier in the section, “Profiling the various software systems”, the functions and behaviors of the legacy system are described in detail, together with the business activities and collaborations related to the system. Then, every transaction performed in the use cases and collaborations is described in terms of data classifications by means of the APS domain model. This step does not require data contents, but does need a list of classes.

Reverse engineering processes, in which RDB schemas in the legacy system are analyzed in detail, usually entail onerous work involving considerable time and effort. In many cases, schemas in the legacy systems are not well formed and easy to understand, due to histories of multiple modifications and use of intuitive terminologies. The process of analyzing the detail of implementation schema is time-consuming and non-productive. In its place, PSLX offers a more elegant and sensible approach that uses only the external specifications of the legacy system in order to obtain an abstract data model.

After arriving at an abstract data model corresponding to real-world business activities, the next step is to develop a particular implementation schema, and then input various data captured from actual situations. In the first few weeks, it may be necessary to have duplicate data input to both the legacy system and the new system. In some cases, package vendors provide data migration tools; however, it is preferable to develop an independent system for data preparation. Even though many IT system development projects focus on the ability of processes to introduce new software functions, frequently the issue of transforming business processes from current real-world operations to the new design is much more important and critical in terms of the ultimate success of the enterprise.

Another example of application of the PSLX specification in terms of application data consistency management can be shown in a case of bill of materials (BOM) management. Depending on the particular manufacturing business context, there may be several types of bills, such as engineering BOM, manufacturing BOM, sales BOM, and service BOM. This different BOM information is usually stored in different RDBs and managed separately, even though the data separately stored represents the same information. Therefore, data consistency management for the distributed databases is required.

Data consistency management is difficult if the system architecture is not on a centralized basis. Many distributed and autonomous systems have disadvantages in terms of data consistency. To solve this problem, the PSLX specifications provide a guide to maintaining consistency between the different data implementations in distributed RDBs by addressing an APS domain model or PSLX ontology as common identifiers.

In order to maintain data consistency, some rules need to be prescribed. For example, if the same information is allocated on two different databases and both are rewritable, then a negotiation rule that defines a process for revising the data according to the other database must be prescribed. This rule can achieve synchronization between these distributed databases. This is possible because the PSLX specification can show semantic identity between different data fields in different data tables, in different RDB implementations.

Support for RDB schema design

PSLX information system architecture is distributed and autonomous architecture rather than centralized. Centralized architecture that manages data in the same place has a long history in data processing - from the mainframe computer era to the client server information system era. On the other hand, distributed and autonomous architecture that consists of small databases is a new concept, and not so popular compared to the previous type. However, the advantage of this architecture is its ability to process extensional and precise data in a formal data format, and flexibility to adapt to changes in the data schema that closely correspond to continuous improvement. The effect of revision on all business processes can be minimized.

In order to get this advantage, many people will have to learn the techniques of schema design so that adjustments to their own localized business environment can be made. Schema design hitherto has tended to be a special knowledge preserve of information system engineers. System implementation techniques for proprietary systems are outside this scope; however, a particular business environment has to be covered by a data schema that has sufficient data and can easily be applied by efficient business algorithms. This is demanding work for a typical IT engineer.

The PSLX-RDB common schema is a template for actual database schemas that meet each requirement of a real-world business environment. Using the PSLX specifications, an average information system designer can create a high-quality RDB schema without any special knowledge of schema design. The procedure to define schemas is outlined below.

First, suitable classes in the PSLX-RDB common schema are selected for each business activity, taking into account the particular use cases involved. If special use cases not included in the PSLX specification are required, then the minimum set of classes containing all information in the use case is added. All the classes selected are defined as views in the target RDB schema.

The classes in a PSLX-RDB common schema each have a minimum set of attributes, where an attribute corresponds to a data field in a view of the RDB. Data in the enterprise business activities are partially represented by the attributes; however, most data require definition of an additional attribute in a class of PSLX-RDB common schema. When a new class or attribute of a class is added to the implementation schema, the position of the additional object in the APS domain model needs to be clarified. Then, the semantics of the additional information can be distinguished, so as to allocate the same data to several different classes in the PSLX-RDB common schema.

The next step is divided into two cases: designing a new RDB schema from scratch, and designing additional classes and modifying some classes in a legacy RDB schema. For new design cases: First, all the classes that are in the APS domain model and may have an attribute of the common RDB schema are listed. The classes listed there are defined as tables in the implementation schema. Furthermore, additional tables for relations between the classes are defined with respect to cardinality. If the relations correspond to relation classes in the PSLX-RDB common schema, they are also added to the table list.

Consequently, a new implementation schema is defined based on the APS domain model, by a schema-transforming procedure in the PSLX specification. This procedure allows the schema to move or duplicate a field between different tables in order to increase calculation speed and reduce the complexity of links between tables. Finally, the data types of each field, i.e. an attribute of a class, are determined. The implementation schema completed to this point should be able to generate a subset of the PSLX-RDB common schema as a view of the tables by SQL or stored procedures. It is possible to create an implementation schema based directly on the PSLX ontology, rather than on the APS domain model.

On the other hand, if there is a legacy RDB system to be modified so as to maintain the data, the schema of the current RDB needs to be associated with the PSLX-RDB common schema by providing corresponding views. In the first step, the schema of the legacy RDB is mapped on the PSLX-RDB common schema. Then, similar to the case of new schema design, classes in the APS domain model are selected, taking into account necessary relations between the classes in the PSLX-RDB common schema and the classes in the APS domain model.

Consequently, the additional data object that exists in the APS domain model, but not in the implemented RDB schema, is identified. If the data object is an attribute of an existing class, then it is described as an attribute of a class in the implementation schema that initially represents the legacy schema. If the data object is a new class, then it needs to be added to the implementation schema.

The procedure to produce a final implementation schema contains further, subsequent detailed steps. These involve precise operations peculiar to individual system environments, so they have been omitted from this paper. In conclusion, the common thread in the two cases of implementation schema design is that the PSLX-RDB common schema is used as a common interface for access to internal data from outside the application software. The data schema inside the implementation schema can be revised frequently without any great concern about inconsistency with other data in different RDBs, while the semantic relations are maintained in the common schema for all the data in the implemented RDB, so that the detailed data schema in each application does not affect those of the others.

Application integration in manufacturing

Systems integration is very demanding work in an information system development project, and is usually where the major technical difficulties arise. Systems integration requires suitable communication capability between two programs. It also requires methods to access a common database. But, that's not all. The most important effort is the one that goes into integration of two or more business processes managed by different sub-systems so that they will perform as a whole, taking into account the common objective.

In reality, it is very difficult to make even the first two aspects of integration work properly. If the two programs that need to communicate with each other are designed by the same designer or the same IT vendor, with agreement between any communication protocols, there should be no problem. However, if the system does not know its future communication partners, and if they should be designed by different parties, there will be a question mark over the likelihood of suitable communication. Furthermore, even in a case where two different programs try to connect to a common database, the detailed information of the database schema is sometimes not available to read, or the information usually does not have any semantics, so that the program cannot find the correct data in the database.

This represents a kind of design problem, as well as a communication problem automatically produced by computers. In PSLX specifications, the semantics of each field of databases can be defined using PSLX ontology. Especially on plant floors, there can be a wide variety of terminologies depending on different company cultures, different divisions and personnel, even within a single enterprise. If a term used in each situation differs on the plant floor, PSLX ontology can locate the same information object by eliminating the ambiguity of the original definition. Furthermore, the APS domain model provides a standard terminology set, which can be useful in sharing information between different organizations without the need to translate using an ontology.

It is impossible to implement a system with full knowledge of all the requirements that will be arise in future communication with currently unidentified future partners. In such a case, the best approach is to define a common schema and protocol whereby all systems that wish to communicate with others support the interface. This involves legacy systems, as well as systems planned for future development. The PSLX-RDB common schema and the PSLX-XML common schema are really examples of this approach. Therefore, a system that has an interface of PSLX-RDB or PSLX-XML can communicate with other systems without any additional input.

Interface design for flexibility, as a component of large systems, is an important issue in developing an IT software package. For this purpose, the PSLX specification helps the package designer in defining the specification of the software. First, the target business activities are selected from the whole set of PSLX business activities. Then, collaborations supported by the package are decided. Third, a communication type is chosen; either a storage type communication by RDB, or a messaging type communication by XML technologies. Then, the appropriate part of the PSLX-RDB common schema or PSLX-XML common schema is selected. Finally, the system has an interface to accept immediate access from other systems, and publishes the information of the interface.

The case where a system only has the capability to access other systems that have PSLX interfaces when it needs to retrieve information is very easy to implement, because there is no need to constantly prepare unknown accesses. In this case, the system needs to have a PSLX-RDB or PSLX-XML common schema, and to only translate data between the internal format and the schema. This is very useful for many kinds of application software that do not have a special database, but instead have special algorithms for particular problems. They can be provided as plug-in software components. There are many personal plug-in business software packages, but only a few software packages in business applications for enterprises. Industrial business knowledge itself can be marketed under this approach in the future.