13 Coordination and Integration

Boris Reuter and Jens Rohde

A strong *coordination* (i.e. the configuration of data flows and the division of planning tasks to modules) of APS modules is a prerequisite to achieve consistent plans for the different planning levels and for each entity of the supply chain. The same data should be used for each de-centralized planning task and decision. APS can be seen as "add-ons" to existing ERP systems with the focus on planning tasks and not on transactional tasks. In most cases an ERP system will be a kind of "leading system" where the main transactional data are kept and maintained. The data basis of APS is incrementally updated and major changes on master data are made in the ERP system. This task will be called *integration* of APS with ERP systems.

The coordination between the different planning modules described in Part II of this book is very important to derive dovetailed detailed plans for each supply chain entity. Section 13.1 will show which guidelines are given, which data are shared and how feedback is organized. Furthermore, one can see which modules are normally used centrally and de-centrally, respectively.

As we have already seen in Chap. 5, some decisions and tasks are left to the ERP system. These tasks and data which are used by APS but are kept in ERP systems are described in Sect. 13.2. The definition of the interface between ERP and APS has to determine which ERP data are used in APS and which data are returned. Moreover, *Data Warehouses* which keep important historical data and are mainly used by Demand Planning build interfaces to APS (see Chap. 7) as well as *business planning* scenarios that—in contrast to most APS—valuate the

B. Reuter
Ernst & Young GmbH, Wirtschaftsprüfungsgesellschaft, Willy-Brandt-Platz 5,
68161 Mannheim, Germany
e-mail: boris.reuter@de.ey.com

J. Rohde (✉)
SAP AG, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
e-mail: jens.rohde@sap.com

H. Stadtler, C. Kilger and H. Meyr (eds.), *Supply Chain Management and Advanced Planning*, Springer Texts in Business and Economics,
DOI 10.1007/978-3-642-55309-7__13, © Springer-Verlag Berlin Heidelberg 2015

output of planning results and set targets for APS and ERP systems. Though, recent research bridges the gap between analytical systems, like Data Warehouses, the transactional ERP systems and planning systems by performing all tasks on a single database with powerful *in-memory computing* technology (Plattner 2013) practice still has to deal with the integration of these different systems and the so-called ETL (Extract-Transform-Load) processes as described in this section. Besides on-premise analytical and transactional systems more and more cloud services need to be integrated as valuable data input for the planning tasks.

A detailed knowledge of the status of supply chain operations and the occurrence of events within the supply chain is getting more and more important. Thus, the concept of Supply Chain Event Management to effectively manage the different categories of events occurring in a supply chain is discussed in Sect. 13.3.

Modules of APS that support collaboration of supply chain entities as well as external customers and suppliers are part of Chap. 14 and will not be discussed in this chapter.

13.1 Coordination of APS Modules

A general structure for coordination of the different modules cannot be suggested. There are several architectures that range between individual planning modules, which can be used as stand alone systems, and fully integrated planning systems. A fully integrated system regularly has the advantage of an identical look-and-feel for all modules and accessibility to all modules by a single user interface. Furthermore, a single database provides data needed by every module and avoids redundancies and inconsistencies in the planning data caused by multiple databases. Different modules can interact via sending messages and exchanging data directly. In contrast, individual modules mostly do not have an identical look-and-feel and regularly no common data basis. An advantage of this architecture is that modules can easily be combined and chosen (if not all modules are needed) for a specific line of business. Most APS providers with such architectures provide special integration modules that enable controlled data and information exchange within the system (see also Chap. 18). Furthermore, an Alert Monitor is often responsible for the handling of alert situations from different APS modules in one central module.

The following paragraphs describe which guidelines are given and how feedback is organized to generate the different plans for a supply chain as a whole. Figure 13.1 gives a general view of the main interactions. The data flows are exemplified, as they can be different from one supply chain to another (see Chaps. 3 and 4). The main feedback is derived by periodic updates of plans while considering current data. Chapters 6–12 illustrate the interactions between APS modules in more detail.

Strategic Network Design. Strategic Network Design determines the configuration of the supply chain. This configuration consists mainly of locations for each supply chain entity and possible distribution channels. Long-term Demand Planning gives input about trends in future demand. Simulated master plans can provide

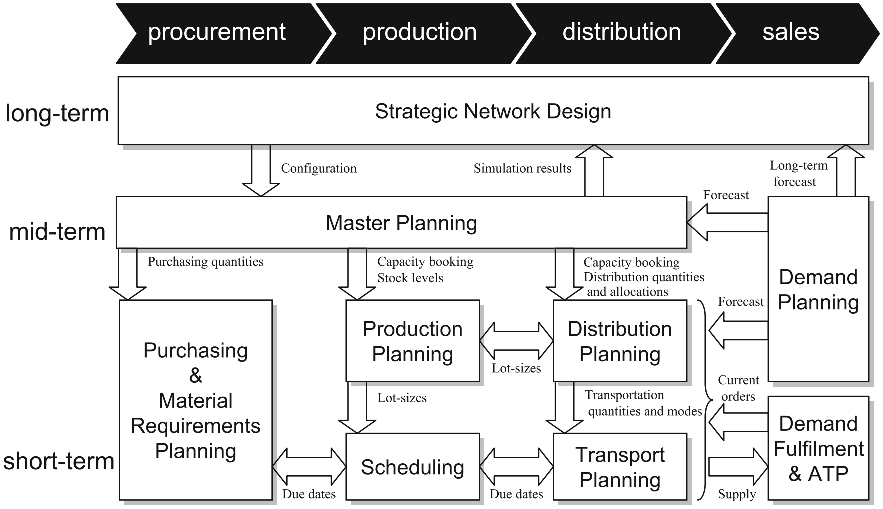


Fig. 13.1 Coordination and data flows of APS modules

useful hints for capacity enhancements. However, the strategic goals of a supply chain (i.e. market position, expanding into new regions and markets, etc.) specify the framework for this module.

Demand Planning. Demand Planning provides demand data for mid-term Master Planning as well as for short-term Production and Distribution Planning. The forecast for end products of a supply chain is input for Master Planning. The short-term planning modules use current, more accurate short-term forecasts from Demand Planning. Furthermore, de-centralized Demand Planning modules provide demand data for products not planned in Master Planning (e.g. non-critical components).

Master Planning. Master Planning determines a production, distribution and purchasing plan for the supply chain as a whole with given demand from Demand Planning on an aggregated level. Therefore, this task should be done centrally. The results provide purchasing guidelines for the Purchasing and Material Requirements Planning—like purchasing quantities from external suppliers, production guidelines for the de-central Production Planning and Scheduling—like capacity booking for potential bottlenecks and stock levels at the end of each period and distribution guidelines—like distribution channel chosen, and distribution quantities for de-central Distribution and Transport Planning. Feedback from short-term modules is derived by current stock levels, updated forecasts and current capacity usage. The average realization of given guidelines from short-term modules should be used for model adjustment in Master Planning.

Demand Fulfillment and ATP. For Demand Fulfillment and ATP demand data from Demand Planning, production quantities for disaggregated products and intermediates, due dates from Production Planning and Scheduling, distribution plans and detailed vehicle routes from Distribution and Transport Planning and purchasing due dates as well as selected suppliers from Purchasing and Material Requirements Planning are used. Furthermore, current inventory levels at each production and distribution stage are needed as input. To be able to influence production and distribution plans, unused capacity bookings have to be known, too.

Production Planning and Scheduling. The main guidelines from Master Planning are capacity bookings and stock levels for each period for the de-central units. Production Planning and Scheduling requires detailed, disaggregated information. Furthermore, current (short-term) forecasts and availability of production resources update the guidelines from Master Planning. Lot-sizes and due dates from this module are exchanged with Distribution and Transport Planning to coordinate production and transportation lot-sizes as well as with Purchasing and Material Requirements Planning to coordinate purchasing lot-sizes and due dates in a more detailed way than it is done by Master Planning.

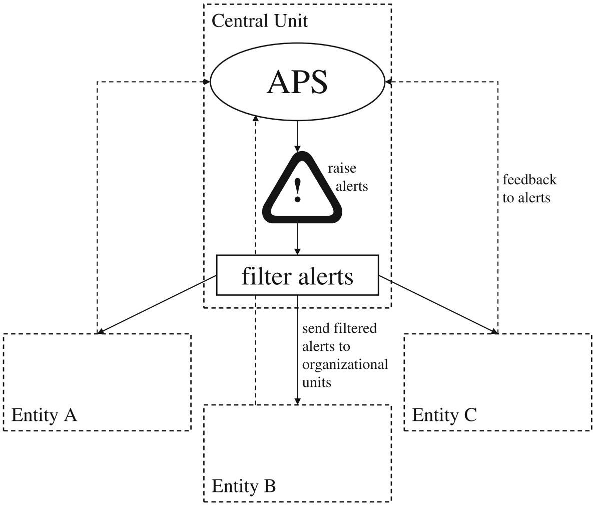
Purchasing and Material Requirements Planning. Purchasing quantities derived from Master Planning provide valuable input for mid-term supplier contracts and supplier selection. Based on these quantities discounts can be negotiated. Considering short-term production due-dates and lot-sizes as well as mid-term contracts for critical components feasible purchasing plans are obtained. Purchasing plans have to be aligned with production schedules to secure an adequate and timely supply of materials.

Distribution and Transport Planning. The coordination of Distribution and Transport Planning is similar to Production Planning and Scheduling. The short-term coordination by lot-sizes and due dates enables accurate production-distribution plans. The actual production quantities provide main input for the transport plans. Furthermore, time windows from customer orders are additional constraints for building and routing vehicle loads.

Alert Monitor. The alert monitor depicts the concept of *management-by-exception*. Management-by-exception is a technique to control guidelines. It differentiates between *normal cases* and *exceptional cases*. Here, the decision whether a situation is an exceptional case or not is delegated to the APS. The prerequisites for this concept are detailed information about tolerances for normal cases, exact definitions for reporting and delegation of decisions (along the lines of e.g. Silver et al. 1998 and Ulrich and Fluri 1995).

The APS raises alerts if problems or infeasibilities occur (see Fig. 13.2). To pass the right alerts to the right organizational units within a supply chain, it is necessary

Fig. 13.2 Alert monitor



to filter these alerts first. Afterwards, filtered alerts are sent to the responsible organizational unit of a supply chain entity. Specifying these responsibilities is part of an implementation project (see Chap. 17). Finally, these alerts have to be sent physically, e.g. by e-mail or an Internet based application.

The responsible units react on alerts by generating new plans, moving orders, using reserved teams, etc. The new or adjusted plans are then sent back to the APS. The APS has to process the changes made and propagate them to each unit affected by the changes.

13.2 Integration of APS

To use an APS effectively, it has to be integrated in an existing IT infrastructure (see Fig. 13.3). The main interactions exist between APS and *online transaction processing* (OLTP) systems, e.g. ERP and legacy systems. Another important system—especially for the demand planning task—is a *Data Warehouse* (DW). This "warehouse" stores major historical data of a business and supply chain, respectively. The next subsection will illustrate the integration of APS with OLTP systems. Afterwards, the integration with Data Warehouses and Business Planning (BP) to achieve a valuation of planning results is described. The integration of OLTP and Data Warehouses will not be subject of this book.

New middleware technology, subsumed as *Enterprise Application Integration* systems, provides a platform for integration of various tools and databases. The last subsection of this chapter will give a brief insight.

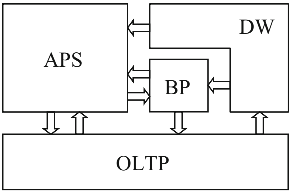


Fig. 13.3 APS integration

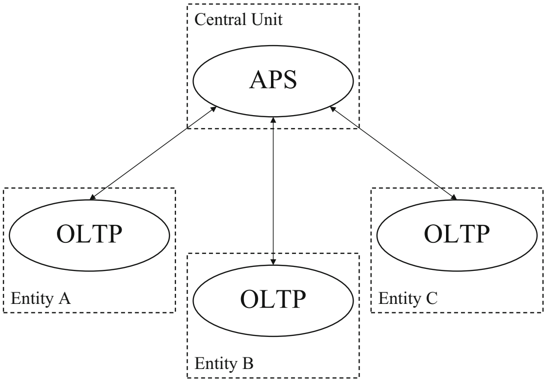


Fig. 13.4 Integration of several OLTP systems

13.2.1 Integration with OLTP

An APS does not replace an existing ERP or legacy system. On the contrary, APS extend them by additional planning functionality. Transactional data are kept and maintained in the OLTP system. An APS is only responsible for its specific data, more exactly, all data that are needed but are not part of the OLTP system's data basis.

As Fig. 13.4 shows, an APS regularly communicates with several OLTP systems of different supply chain entities. Furthermore, planning tasks like BOM explosion for non-critical materials and ordering of materials are mostly left to ERP systems (see Chaps. 4 and 5). The *integration model* defines which objects are exchanged, where they come from and which planning tasks are performed on which system. The *data exchange model* specifies how the flow of data and information between the systems is organized.

Most APS provide a *macro-language* to define these models and enable an automatic exchange of data. While OLTP systems are regularly older systems, the adjustment has to be done by APS. That is, an APS has to be able to match data items from the OLTP system to its implicit structure and to handle different import and export formats, because it is mostly not possible to do all adjustments needed on the OLTP system. Also, it must be possible to maintain specific data like penalty costs and aggregation rules (see Sect. 13.2.3 and Chap. 8) within the APS.

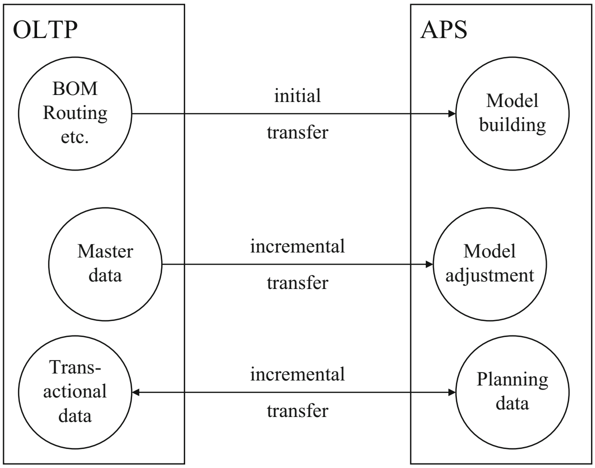


Fig. 13.5 Transferring data between OLTP and APS

Integration Model

Within the integration model, objects which are exchanged between OLTP and APS are defined. If, like it is done in most cases, not all products are planned in APS, the integration model has to define which products and materials are critical. Also, the potential bottleneck resources have to be defined. These objects are e.g.:

- BOMs
- Routings
- Inventory levels
- Customer orders.

Furthermore, the integration model has to define which data are exchanged with which OLTP system. A supply chain consists of several entities with local systems. The right data have to be exchanged with the right system. This assignment can be done by modeling different sites with their flows of materials in the Master Planning module (see Chap. 8).

The integration model also defines which results are returned to the OLTP systems and the planning tasks done by an APS or ERP system (e.g. performing BOM explosion of non-critical components in local ERP systems). By defining several integration models it is possible to simulate different alternatives of a division of labor between APS and ERP system.

Data Exchange Model

Data *which* have to be exchanged between OLTP and APS are mainly defined by the integration model. The data exchange model defines *how* these data are exchanged. The data transfer between OLTP and APS is executed in two steps (see Fig. 13.5).

The first step is the *initial data transfer*. During this step data needed for building the Master Planning, Production Planning and Scheduling and Distribution and Transport Planning models are transferred from OLTP to APS (e.g. the BOM and routings of critical products, properties of potential bottleneck resources, regular capacities, etc.). After the models are generated "automatically", it is necessary to maintain the APS specific data like optimizer profiles, penalty costs and aggregation rules.

In the second step, data are transferred *incrementally* between the systems. The OLTP system should only transfer changes that have been made on the data to the APS and vice versa (*net change*). The data exchanged are divided into *master data* and *transactional data*. Changes on master data require a model adjustment in the APS. For example, this could be the purchase of a new production resource or the long-term introduction of a second or third shift. Transactional data are transferred to and from APS as a result of planning tasks. For example, the following transactional data are sent incrementally to an APS:

- Current inventories
- Current orders
- Availability of resources
- Planned production quantities and stock levels, respectively.

Current inventories are needed for every APS module. For Master Planning they can be regarded as a feedback in an incremental planning process, the ATP module uses this data to perform online-promises, while in Distribution Planning it is necessary to calculate the actual distribution quantities, etc. Current orders are used to match planned orders. Those planned orders are a result of the Production Planning and Scheduling module if this planning task is performed on forecasts. All short-term planning modules have to consider the availability of resources like machines, vehicles, etc. to generate feasible plans. Planned production quantities and stock levels are for instance needed to perform BOM explosions for non-critical components in local ERP systems.

However, the separated data basis for APS modules poses problems of redundancies and inconsistencies. These problems have to be controlled by the data exchange model. Even though redundancy of data enables the APS to simulate different plans without affecting OLTP systems, it is very difficult to ensure that all systems have the correct data. Changes in each system have to be propagated to avoid an inconsistent data basis. That is, every modification has to be recorded and sent to the relevant systems. If too many changes are made, too many data are transferred between systems and data updates paralyze the APS. The trade-off between 100 %—consistency and paralyzation of the APS has to be considered during the implementation process (see Chap. 17). That is to say, it is possible to perform updates in predefined time intervals to avoid trashing by data transfer but with a reduction in consistency.

13.2.2 Integration with Data Warehouses

While OLTP systems depict the current state of a supply chain entity a *Data Warehouse* is its "memory". Nearly all data is available—but not information. The goal of a Data Warehouse is to provide *the right information at the right time*. The Data Warehouse has to bring together disparate data from throughout an organization or supply chain for decision support purposes (Berry and Linoff 2004, p. 473).

Several years ago the terms *knowledge discovery in databases* (KDD) and *data mining* kept arising in combination with Data Warehouses. The term KDD is proposed to be employed to describe the whole process of extraction of knowledge from data. Data mining should be used exclusively in the discovery stage of this process (Adriaans and Zantinge 1996, p. 5). The complete set of these tools including data storage in DW is usually known as *Business Intelligence* (Loshin 2003).

Since data is changing very fast and the amount of data is growing faster than the possibility to store it new approaches were elaborated to access and analyze large data sets. One issue in Data Warehouses is the predefined structure of master data and metrics that has to be adopted when changes occur. In case of analyzing not yet modeled structures the Data Warehouse is not able to provide ad hoc mechanism for this type of analyze. There are approaches known as *Analytics* or *Data Discovery* that are typically based on so-called in-memory technologies with specific methods to access, model and store data (Plattner 2013). Visualization components combined with fast data access help users to analyze their data from different angles simultaneously. Prognosis algorithms combined with this type of data management is also known as predictive analytics.

The interaction between APS and the Data Warehouse is a read-only process—the Data Warehouse is updated incrementally by transactional data from OLTP systems. The main use of Data Warehouses is in Demand Planning where statistical tools are applied on historical data to find patterns in demand and sales time-series (see Chaps. 7 and 29). Business Intelligence, especially data mining and predictive analytics provide important input supporting model building for all modules of an APS. While data mining tools focus on *finding patterns by using more complex algorithms* in data, data discovery and analytics tools are fast and powerful tools for *reporting and exploring the results*. Data discovery tools have begun to substitute the former *online analytical processing* (OLAP) tools to provide a fast way for APS to access data not only of the Data Warehouse. The conventional way by queries (esp. SQL) is also possible to access data (see Fig. 13.6).

SCM is a new challenge for the design of Data Warehouses and Business Intelligence. The supply chain inherent dynamic of changing data structures (e.g. organizational changes, new products, new sales channels) drives huge efforts, thus the aim of integrating cross-company supply chain wide master and transactional data is reserved to selected applications and business objects (see e.g. Chap. 14).

13.2.3 Integration with Business Planning

Business Planning is often referred to as a value based planning process which includes the key performance indicators of an enterprise. In the organization of a company, Business Planning is mostly driven by the finance and controlling department. One can distinguish long-term strategic planning, 1 to 3-year-planning

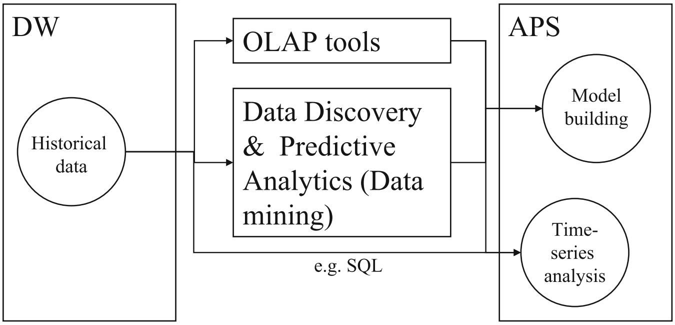


Fig. 13.6 Integration of Data Warehouse

(budgeting) and monthly or mostly quarterly forecasts. As opposed to an OLTP where detailed transactions are focused on, predominantly aggregated values are being planned. Business Planning systems often use the structures of a Data Warehouse, as the reporting structures (legal entities, segments, regions) for controlling purposes are ideally congruent with the planning structures.

Business Planning generally consists of the following elements:

- Planning of guidelines, assumptions and premises
- Sales planning: Based on the services offering (mix, quantity and quality) and price policy
- Cost planning: based on cost elements directly allocatable to a service activity and indirect costs on cost centers; through internal cost allocations, these indirect costs are allocated to activities
- Investment planning: planning of payment flows resulting from longer-term investment decisions
- Profitability planning
- Profit & loss and balance sheet planning
- Cash flow statement.

Methods within sales planning are often congruent with practices of demand planning. For cost planning, predominantly distribution keys or simple allocation methods are in use. More complex allocations are performed in the OLTP based on the exact allocation rules after upload (so-called retraction) of the planned cost center values. Basic amortization calculations or cost comparison calculations are utilized to calculate investments. Capitalized values are not so widespread.

Most significant is the integration of sales and demand planning. Sales targets can be converted top-down into quantity targets using price targets or independent price planning. Sales targets can then be distributed over lower organizational levels or time intervals using distribution keys (see Chap. 20). Sales figures are then reconciled, bottom-up aggregated and evaluated. Most common issues can be found in the reconciliation processes of diverse hierarchy levels to commit to a binding plan. This is because the plan often represents part of the manager's goal-sheet.

Apart from that, price planning and distribution keys used in top-down plans are significant challenges (due to (lack of) stability and completeness of historical data, price elasticity of certain markets etc.). Commitment to a sales plan requires its plausibility. In a case of restricted resources, this means there needs to be a master planning. From the resulting resource and material usage direct costs can be derived. Cost allocations of indirect costs can be calculated from the cost center planning. Both represent a complete bottom-up cost calculation.

One of the integration issues of cost planning and APS is that most of the cost elements are not appropriate to derive optimization penalties directly from them (e.g. for master planning), because of the portion of indirect costs and not relevant direct costs. Investment planning often is done separately either in strategic network design or in an external investment program plan. Thus, the integration of increasing capacities by investments into the supply chain vice versa has to be done manually.

From the plans illustrated, a planned profitability calculation can be derived. Including also non operating costs and income, and financing and investment planning, finally it is possible to generate a plan balance sheet, a plan profit and loss account and a plan cash flow statement.

A Business Planning prepared using all the planning elements of this method represents a closed loop.

13.2.4 Enterprise Application Integration

The growing number of different systems within a single organization makes point-to-point integration no longer applicable. Integrating the various systems of an organization or even the entire supply chain, called *Enterprise Application Integration* (EAI), is a challenging task that could not be performed without powerful middleware systems. According to the task these systems have to perform they are called EAI systems. The goal of those systems is the decoupling of applications. Figure 13.7 visualizes the difference between point-to-point integration and decoupled applications.

Independent of the underlying software component architecture like *Enterprise Java Beans* (EJB), *CORBA* or Microsoft *DCOM*, an architecture for EAI has to be identified. Such an identification provides essential input for software selection and implementation. Lutz (2000) distinguishes the following five EAI architecture patterns:

Integration Adapter. Via this architecture an existing (server) application interface is converted into the desired interface of one or more clients. The client application will then exclusively invoke services of the server application through the interface of the integration adapter. An interface is dedicated to a single server application, but multiple clients can access the server application by using a common integration adapter. Changes in the server application no longer reflect each (accessing) client; only the adapter has to be adjusted. The integration adapter does not provide any logic. Solely, a mapping of the server API (application programming

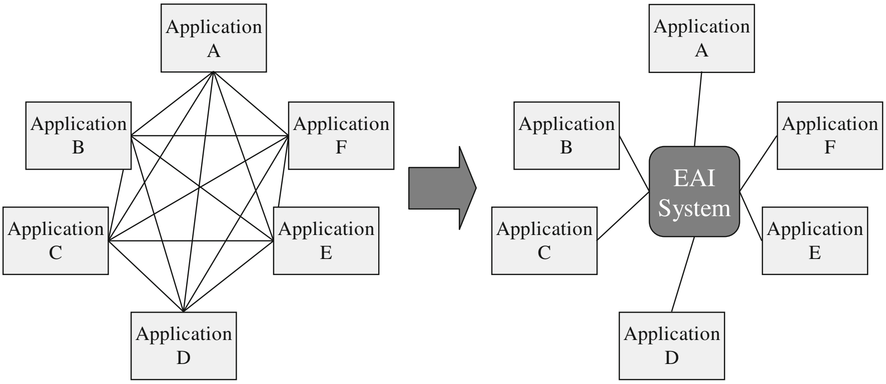


Fig. 13.7 Point-to-point integrated vs. decoupled applications

interface) to the API provided by the adapter is performed. Usually, the integration adapter does not know of the existence of clients and the server application does not know about the existence of the integration adapter unless the server application needs adjustment to be able to provide desired services.

Integration Messenger. Communication dependencies between applications are minimized by this approach. Here, the application interaction logic is not decoupled. The integration messenger delivers messages between applications and provides location transparency services, i.e. distributed applications do not have to know about the actual location of each other to be able to communicate. The design of the integration logic is still left to the applications. One example for an integration logic is *remote method invocation* where an application is able to perform "direct" method calls on another one. In this case both applications have to provide the required services and public interfaces for remote method invocation. A change on the integration logic in one application still affects the other ones while application location changes only concern the integration messenger.

Integration Facade. The interface to several server applications is simplified by providing an integration facade. Server functionality is abstracted to make the back-end applications easier to use. The integration facade has to perform the mapping of its own interface to the interfaces of the server application. Furthermore, internal logic has to be provided to enable the abstraction. For example, an ATP request (see Chap. 9) invokes services on various systems to get information about product availability. The different system are e.g. inventory management systems of different distribution centers. Without an integration facade the client has to invoke these services on each server application. Whereas the integration facade is aware of the different systems. It provides a thin interface to the client(s) and takes over the invocation of the services on different systems as well as processing the information.

Server applications are unaware of the existence of the integration facade while the integration facade itself does not know of the existence of clients.

Integration Mediator. This architecture pattern encapsulates interaction logic of applications and decouples this logic from the participating applications. In contrast to the integration facade the participating applications are aware of the existence of the integration mediator. No direct communication between these applications is permitted. Each interaction has to invoke the integration mediator. Via this pattern dependencies between applications are minimized and maintenance is simplified owing to the centralized interaction logic. The interaction logic encapsulated by the integration mediator includes message content transformation (e.g. mapping of different product IDs) and controlling message destination(s). In stateless scenarios this logic is only dependent on the current content of a message. In contrast, stateful scenarios are additionally dependent on previous application interactions (e.g. accumulation of events). Stateful integration mediators are much more complex to handle as they usually need state management and state persistence to span shutdown situations, for example.

Process Automator. The goal of this architecture is to minimize dependencies between process automation logic and applications. It automates sequencing of activities in a process. The process automator pattern consists of a process controller, activity services and applications providing desired services. The sequencing activity logic of a process is implemented by the process controller. The activity services abstract from the applications and provide request-based services to the process controller, i.e. all system interactions are hidden. The activity service is a specialty of the integration facade pattern that abstracts interactions to the level of an activity. By providing such a simplified and uniform interface to the service applications, the process controller is decoupled from the special APIs of the services. The application integration logic is encapsulated.

The different architecture patterns can be combined. For example, integration adapter and integration messenger can be combined in such a way that the integration adapter provides the interfaces that the integration messenger is expecting. This architecture decouples server APIs and the application itself from the API of an integration messenger.

A major part of the EAI had been performed on so-called on-premise systems that are in the responsibility of the company using these systems. These days cloud services are gaining importance. The model of *cloud computing* operates data and computation power in a collection of data centers owned and maintained by a third party—the so-called *cloud*. Many cloud services follow a pay-per-use model or are delivered to customers by e.g. a monthly fee. Depending on the services offered three different categories are build in general (Jin et al. 2010):

IaaS (Infrastructure as a Service). This is usually the lowest level of cloud service offerings. It provides access to e.g. storage and computing power (e.g.

Dropbox). A consumer of this service has control over operating systems, storage and deployed applications but does not manage the underlying infrastructure itself (like needed when running systems on-premise).

PaaS (Platform as a Service). Offering PaaS implies providing tools for developers to build and host web applications that can be consumed web-based and more and more on mobile devices. An example here is the *Google App Engine*.

SaaS (Software as a Service). The highest level of cloud service offerings is SaaS. The software offered as a service has to be accessible from different devices incl. mobile devices via so-called *thin clients* (e.g. web browsers). Especially, software services offered for mobile devices need to take care of lower bandwidth and computing power on these devices implying that the major part of the computing has to take place on the server offering the service.

Integrating these services into the company's system landscape is also achieved by EAI but relies strongly on the integration technologies offered by PaaS providers—usually the same technologies as used by software developers to connect their software to the platform service. Most commonly used technologies are: Remote Procedure Call (RPC), Service-Oriented Architecture (SOA), Representational State Transfer (REST) and Mashup (an overview of these technologies and the cloud service categorization can be found in Jin et al. 2010).

13.3 Supply Chain Event Management

The task of *Supply Chain Event Management* is to *manage* planned and unplanned events in a supply chain. The effectiveness of the supply chain is to be improved while reducing costs by handling events. Managing events does not only mean to react to events, but also to affect or even prevent their occurrence. Following Otto (2004), SCEM can be characterized as a management concept, a software solution and a software component. Here, we will focus on SCEM as a management concept.

To understand SCEM Alvarenga and Schoenthaler (2003) define the terminology given in Table 13.1. A supply chain event can occur on all levels of detail in the supply chain (from broader cycles to detailed tasks). To be able to manage these events efficiently a reasonable grouping into event categories is inevitable. By giving the probability of an event to occur events can be classified into standard and nonstandard events, assuming that nonstandard events are generally more costly to manage. A documented reaction to supply chain events avoids ad hoc decisions under pressure of time. Planned events are generally less costly to manage.

A reduction in cost by SCEM can be achieved in two ways. Firstly, the reaction to an event can be accomplished more efficiently. Secondly, more costly events can be shifted to less costly events by defining EMPs for so far unplanned events or by eliminating events following the idea of continuous improvement.

When changing the category of events from unplanned to planned the tradeoff between costs for defining an EMP and the probability of occurrence and the

Table 13.1 SCEM definitions (Alvarenga and Schoenthaler 2003)

Supply chain event	Any individual outcome (or nonoutcome) of a supply chain cycle, (sub) process, activity, or task
Supply Chain Event Management	The application of statistical, process, and technology identification and control solutions to standard and nonstandard supply chain events
Event category	A logical grouping of supply chain events
Event probability index (EPI)	The statistical measure, on a scale of 0 to 1, of the tendency of an event to occur within the supply chain [. . .] over a given time interval
Standard event	An event that tends to occur within the supply chain; that is, has an EPI of 0.50× or higher
Nonstandard event	An event that tends not to occur within the supply chain; that is, has an EPI of 0.49× or lower
Event management plan (EMP)	A documented process that outlines the steps taken to control or react to an event
Planned event	An event for which a documented EMP exists
Unplanned event	An event for which a documented EMP does not exist

resulting ad hoc decision should be taken into account. Furthermore, it should be inspected whether external influences like material shortages can be affected, e.g. by using the concept of vendor managed inventory. Thus, probably shifting an event from a standard to a nonstandard event makes a single occurrence more costly, but in sum less costly due to less occurrences.

A software based approach to SCEM has to provide online access to predefined supply chain events. Furthermore, events have to be categorized and management of event shifts has to be provided. By making categorized events and actions to be taken in case of occurrence accessible throughout the complete supply chain a valuable *supply chain visibility* is created. Here, an Alert Monitor offers an important platform for identification of events and notification of their respective owners.

References

Adriaans, P. & Zantinge, D. (1996). *Data mining*. Harlow: Addison-Wesley.

Alvarenga, C. & Schoenthaler, R. (2003). A new take on supply chain event management. *Supply Chain Management Review, 7*, 28–35.

Berry, M. & Linoff, G. (2004). *Data mining techniques: For marketing, sales, and customer support* (3rd ed.). New York: Wiley.

Jin, H., Ibrahim, S., Bell, T., Qi, L., Cao, H., Wu, S., & Shi, X. (2010). Tools and technologies for building clouds. *Cloud computing: Principles, systems and applications*. Berlin/Heidelberg/New York: Springer.

Loshin, D. (2003). *Business intelligence* (1st ed.). Amsterdam/Boston: Morgan Kaufman.

Lutz, J. (2000). EAI architecture patterns. *EAI Journal, March*, 64–73.

Otto, A. (2004). Supply chain event management: Three perspectives. *International Journal of Logistics Management, 14*(2), 1–13.

Plattner, H. (2013). *A course in in-memory data management*. Berlin/Heidelberg/New York: Springer.

Silver, E., Pyke, D., & Peterson, R. (1998). *Inventory management and production planning and scheduling* (3rd ed.). New York: Wiley.

Ulrich, P. & Fluri, E. (1995). *Management: Eine konzentrierte Einführung* (7th ed.). Bern: Haupt.