RosterBuilder - An Architecture for an Integrated Airline Rostering Framework

Karl Doerner
University of Vienna
Department of Business Studies
Brünner Strasse 72
AUSTRIA – 1210 Vienna
karl.doerner@univie.ac.at

Gabriele Kotsis
Copenhagen Business School
Department for Informatics
Howitzvej 60
DENMARK – 2000 Frederiksberg
gk.inf@cbs.dk

Christine Strauss
University of Vienna
Department of Business Studies
Brünner Strasse 72
AUSTRIA – 1210 Vienna
christine.strauss@univie.ac.at

Contact for further correspondence:

Christine **Strauss**
University of Vienna
Department of Business Studies
Brünner Strasse 72
AUSTRIA – 1210 Vienna
christine.strauss@univie.ac.at

RosterBuilder – An Architecture for an Integrated Airline Rostering Framework

Abstract

Rostering is a crucial planning process in an airline's overall scheduling hierarchy. Besides addressing the issues of crew utilisation and crew costs (which represent two of the major expenses in running an airline), a rostering tool must also cover the question of quality requirements. The quality of service can be improved by having the rostering tool take into consideration the skills of individual flight attendants during the rostering process. The quality of work life that an individual employee enjoys depends on his or her ability to participate in the rostering process. To an extent, this participation involves the employee's ability to determine his or her own work schedule, which in turn also impacts his or her job performance. Timely and ubiquitous access to rostering information therefore represents a crucial benefit for employees, as it enables them to not only query data on schedules, but also to submit their preferences at all times and in any place.

In this paper, we present the architecture for RosterBuilder, a sample framework for flexible rostering at airlines and show how RosterBuilder can be integrated into modern computing and communication environments. We focus on deriving functional requirements that apply to most airlines using a rostering concept and specify those requirements in use case diagrams that follow the UML formalism. Our analysis considers the user groups that are involved: crew members, crew assigners and crew operators.

Considering the need for flexibility, scalability and ubiquitous access to information, we propose a component-based system architecture that is easily customisable and adaptable to end users' needs. We present and describe the integration of rostering algorithms as the core models and the interfaces to additional modules, which include support functions applied when generating rosters and mobile information access for crew members. We conclude by presenting the design for the proposed modular architecture.

Keywords airline application, requirements elicitation, requirements engineering, rostering, software architecture, UML-based modelling

Introduction

The tight market conditions in the airline sector are not only a consequence of economic turndowns triggered by market-external factors, but also reflect the long-term effects of liberalization, which has lead to intensified competition in the airline market. As a consequence, airlines are being forced to exploit and activate idle potentials by applying marketing strategies aimed at raising their turnovers, cutting down their expenses and maximizing synergies. The industry's increasingly tight profit margins, as well as a continuous rise in flight capacity and frequency, have increased the requirements placed on planning tools in this important economic sector with regard to their robustness, flexibility and accuracy.

The need for appropriate tools for rostering and its applications is especially urgent, because personnel costs in the labor-intensive airline industry are generally relatively high. In fact, the cost of personnel represents the second highest factor of direct operational costs next to fuel. Besides the relative importance of personnel costs, the growth of the industry's workforce provides another indicator for the urgent need for highly developed rostering tools. Given that rostering is primarily applied by European airlines, a short look at some statistics for the European market can help to substantiate the assumption of growth:

(1) For the years 1995 to 1999, the average annual percentage increase in the number of flight deck personnel was 5.6 percent, while that for the number of flight attendants was 6.5 percent (source: IATA, World Air Transport Statistics, data request 5975-00).

(2) The Association of European Airlines (AEA) reports that the volume of seat kilometers flown by its member airlines in the year 2000 was 4.5 higher than in 1999 (AEA, 2001, p. IV-3).

(3) An indicator for a continuation of this trend can be found in Boeing's forecast that the annual air travel growth rate for the European region will average 4.7 percent in the years until 2020. This forecast gives further credence to the assumption that the number of personnel being rostered will continue to grow (Boeing, 2001, p.10).

In addition, IATA expects total international passenger traffic to increase at an average annual rate of 3.5 percent between 2001 and 2005 (this figure already includes the impact of the 2.1 percent fall in passenger numbers in 2001; cf. Jeanniot, 2002, p. 11). Both the cost and volume factors imply that powerful approaches and solvers need to take additional requirements and attributes into consideration in order to use personnel resources more efficiently, i.e., disposing workforce closer to its regulative limits without decomposing the problem into "independent" subproblems, while fully exploiting the individual crew members' skills. This implies a complexity that is far too advanced to be managed manually and might even force existing tools to their limits.

Not only does the quality of service provided by airline employees depend to a considerable degree on the employees' satisfaction with their work schedules, airlines' ability to meet their staffs' expectations regarding work schedules provides a viable means of incentivizing employees and represents an alternative to monetary compensations.

Several factors contribute to the complexity associated with Airline Crew Rostering Problems: (1) a crew member's duty may cover a period of several days (up to 16), (2) each work schedule is tailored for a specific person with specific needs, (3) a varying number of crew members with specific skills are required, depending on the type of aircraft, and (4) the large size of real-world applications of airline rostering and the typically large sets of data associated with them make their highly efficient implementation a prerequisite. In addition to problem complexity, there are several external factors that hinder or prevent the development of a single, monolithic tool. Because requirements often vary widely (e.g., the terms of the work contracts differ highly from airline to airline), the constraints for a solution procedure need to be adjusted and customized in each implementation. Furthermore, the market for rostering tools in the airline sector is relatively small on account of alternative paradigms such as bidlines and preferential bidding. The development of a single monolithic tool does not appear to be economically feasible: a single provider would face too much risk, as the development of such an integrated software package would require high initial investment and remains cost-intensive. Moreover, each customer requires a specific solution.

Component-based, customizable software presents a possible means for confronting the drawbacks in this challenging application. Such a software approach allows multiple vendors to offer and integrate their products. It opens the market to those vendors who might already be successfully established in other markets and could provide useful non-airline specific modules. The products available on the market confirm this trend[1].

The "missing link" is a framework that specifies how such individual components can be "glued" together. This paper provides this missing link by presenting a reference architecture, which is the basis for customizable airline rostering software.

From the point of view of the tool builders, the benefits of a reference architecture include

- **Market assessment**: the elements of a reference architecture provide a basis on which to collect and structure information about which rostering(-related) modules are provided by whom and on how much of the market potential is already supplied/covered. As a result, tool builders can gain an insight into the state of competition and the number of competitors in individual market segments.
- **Product placement**: a reference architecture allows a broad variety of vendors - not necessarily only those from the airline rostering segment, but also tool providers from other business segments - to see where their products fit in among the market's offerings.
- **Product differentiation**: a market analysis based on a reference architecture can help to point out which individual tools will be harder or easier to differentiate from competitors' products.
- **Product diversification**: a reference architecture can offer the basis on which airline rostering tool builders evaluate their current degree of product diversification and provide an impetus for complementary products.
- **Integration**: a reference architecture supports the integration of add-on offerings. For example, a module for crew preferences or for crew tracking and monitoring can be integrated into an existing

[1] We refrain from mentioning specific products, brands or commercial software solutions, as providing a market overview would go beyond the scope of this paper.

implementation despite the fact that the details of the rostering kernel might not be in the public domain.

- **Market awareness**: a reference architecture opens the airline rostering market to vendors from other sectors, because it specifies well-defined modules, some of which are not necessarily airline-specific and can be provided by vendors established in other sectors.
- **Customer acquisition**: the definition of core and auxiliary modules in a reference architecture enables the customer to perform a stepwise implementation. Medium and small airlines would rather invest into a limited, less cost-intensive start-up solution than in one big product.

Furthermore, a reference architecture supports the airlines in their roles as customers:

- **Gap analysis**: on basis of a reference architecture, an airline can evaluate its own standards, create a check list for analyzing the existing IT support in the company and perform a gap analysis.
- **Product evaluation**: a reference architecture is the basis on which airlines in their roles as customers of software vendors can be provided with a framework for evaluating various competing tools on the market. In addition, it helps to describe the quality of the offered tools on a component level and to establish to what degree the airlines' requirements and the functionality offered by a tool match.
- **Priorities of investment**: a reference architecture implies a hierarchy of components. It helps to identify basic functionalities and add-ons and supports the distinction between core functions, auxiliary functions and nice-to-have-modules. Furthermore, a reference architecture supports the customer in setting investment priorities when an existing system is updated or a new system is implemented.
- **Product variety**: as a reference architecture may attract vendors from other sectors, multiple vendors will offer solutions and the variety of products being offered on the market will rise; as a consequence, customers will have a broader variety of choices available to them.
- **Standardization:** a reference architecture supports the process of standardization, which provides a viable means for avoiding dependencies on single providers. For example, the definition of the open system architecture was a precondition for the development of LINUX.
- **End-User Orientation:** the reference architecture not only describes interfaces for crew assigners and crew operators, but also provides access functions to crew members. Crew members benefit from timely and ubiquitous access to all information related to their duties and can actively participate in the rostering process by stating their preferences on work schedules.

The remainder of this paper is structured as follows: Section 1 describes the planning hierarchy in the airline industry, highlights the difference between bidlines and rosters and presents related academic work. In Section 2, the requirements of a reference architecture are described by means of three use cases, namely, the main user categories "crew member", "crew assigner" and "crew operator". Section 3 focuses on RosterBuilder's reference architecture as a useful example and describes the core modules and the additional modules that address complexity and quality criteria. The methods implemented (e.g., pairing optimizers, rostering programs, preference evaluators) ought to be modular and should also conform to the software architecture. Section 4 of the paper concludes with a summary and an outlook on future work.

1 Rostering in Airline Applications

The term “rostering” derives from the practice of manually filling out a grid-lined sheet of paper in order to provide a graphical sketch of individual duty schedules. Wren (1996) gives a general definition on the term rostering as a specific type of scheduling by „...the placing, subject to constraints, of resources into slots in a pattern” (Wren, 1996, p. 53). Once shifts or duties showing the daily work of the personnel have been generated, these tasks are placed into a roster, which is a list of people with assigned duties for a particular day. Usually, the work pattern for an individual during any time period must conform to a set of rules, which are particularly manifold for flying personnel.

In airline applications, the rostering process is a planning activity embedded in a planning hierarchy, which is briefly described in the following paragraphs to give a rough insight into where the data that determines the rostering process of flight personnel is derived from (Barhart and Talluri, 1997; Barnhart et al., 1999).

The primary planning activity is the flight schedule design, which focuses on finding a set of scheduled flight legs (characterized by origin, destination and departure times) that maximize an airline's profit within the restrictions imposed by fleets, crews, regulatory agencies, etc.; and which may be solved with some assistance from automated profitability tools. Flight schedules are an airline's primary products, they are generated for six months or a whole year. During the second scheduling phase, called the fleet assignment, equipment types are assigned to flights, with the airline trying to find the profit-maximizing assignment of aircraft types to flights, given restrictions imposed by flight coverage, fleet balance, and maintainance. Many major airlines apply fleet assignment optimizers to solve this problem, a step usually taken three months ahead of the start of the schedule. The subsequent planning activity involves aircraft maintenance routing, during which the airline tries to find the set of aircraft rotations that maximizes through revenues given maintenance requirements; most major airlines use optimization models and algorithms during this phase.

Crew schedule planning falls into two more or less independent parts: crew pairing and crew assignment. The objective of crew pairing is to cover each flight the required number of times – depending on the aircraft type – by cost-minimizing work schedules given restrictions imposed by flight coverage, work restrictions and crew availability. A pairing is a sequence of flights beginning and ending at a crew base containing a set of daily work activities, called duties or duty periods, separated by rest periods. Every pairing is constructed so that a single crew member can legally perform all of the work activities it contains; pairings span time periods of some hours up to 16 days. Terms used synonymously for pairings are tours of duty, trips, or rotations. Extensive research in the last four decades yielded in efficient crew pairing optimizers.

Generally speaking, crew assignment accomplishes the “personalization” of the anonymous crew pairing solutions by linking several pairings with rest periods, training time, leave, etc., to form work schedules that cover the period of about one month and can be performed by a single crew member. Crew assignment takes into account the need to provide sufficient rest periods between flights and reflects the numerous restrictions

imposed by regulatory agencies, collective bargaining agreements and labor unions. Depending on the crew assignment strategy either lines-of-work or bidlines are used during the crew assignment process.
Widely used by the airline industry in North America, bidline generation involves constructing anonymous cost-minimizing bidlines and then letting the individual crew members express their preferences through a bidding process. The eventual assignment of a specific schedule to a specific person is often based on individual priority rankings (e.g., seniority).
Rostering is a commonly used assigment method in Europe and in some other countries (e.g., New Zealand). It constructs measure-made lines-of-work for each crew member and takes into account pre-assignments (e.g., training, observer flights) and crew-requests (e.g., days-off, annual leave, specific flights) when constructing individualized schedules for each crew member. Airline crew rostering's primary goal is to provide a fair-and-even distribution of the workload among all crew members and to maximize the crew members' satisfaction with their individual duty schedules.
Preferential bidding represents a compromise between the bidline and rostering approaches in that it generates personalized schedules that also take into account a set of bids that have been weighted to reflect the employees' preferences (cf. e.g., Gamache et al., 1998).
The reason for these two different assignment "philosophies" lies in the differing nature of working contracts in various parts of the world. In North America, work schedules involving high workload result in high pay and vice versa. By contrast, flight personnel in Europe operate under work contracts that guarantee payment for a certain minimum amount of flying hours (so-called block hours), no matter if the crew member actually performs these hours or not. An efficient crew planning approach must therefore avoid producing schedules whose workload is beneath the guaranteed lower bound set by the crew member's salary.
The rostering process itself, as well as all other described planning processes that rostering is based on, are anticipatory in nature; what-if-scenarios and variants can be tested, with some solutions being kept, while others are discarded. The point of time at which a roster is published is crucial insofar as an early publication might lead to many changes before the schedule actually starts, while a late publication is not convenient for the crew members. After the roster is published, duty swaps and minor changes might occur, as well as changes due to disruptions (e.g., maintenance problems, severe weather conditions or no shows). Crew recovery is a task performed by crew operations, which does not operate with several roster versions, but rather with **one** real time system.
Some approaches confront the drawbacks of sequentially solved airline schedule planning (i.e., incompatibilities, false infeasibilities) by simultaneously generating solutions, e.g., for fleet assignment and crew pairing (Barnhart et al., 1998).
After the overview over the planning activities, which determine the rostering process, we will in the following section describe key functions, which three types of users would expect to be able to perform with an intgrated rostering tool.

2 Requirements analysis

The requirements for a rostering reference architecture are determined by the functionality a typical user would expect from such a system and the specific activities he/she would perform. We distinguish between three archetypical user categories: (1) the crew member, who not only receives the duty plan through the system, but also communicates with the system with regard to duty determinants, (2) the crew assigner, who is responsible for the roster generation process, and (3) the crew operator, who is responsible for correctly updating the roster once it has been published.

Needless to say, the emphasis when implementing such a rostering tool in the airline sector is placed on reliability functions and recovering mechanisms, because very high upfront and follow-up costs may be associated with a shortage of personnel due to delays or no-shows of crew members. In addition, penalties and fines might be the incurred if work rules are ignored or work restrictions are violated.

The majority of users belongs to the "crew member" category, i.e., flightdeck crew (pilot and co-pilot) and flight attendants. From their viewpoint, the core function of a rostering tool lies in providing reliable data about their duties. Besides receiving data on their current duty, these users expect to be able to comment on them (by means of questions, feedback, complaints) and to query their past duties as well. Crew members are interested in knowing who their crewmates are and with whom they will be working. Therefore, the system should provide crew members with a query function that lists the names of all crew mates on a flight-leg-basis. Crew members should also be able to query the location of colleagues on a point-of-time basis, with the information provided conforming to the "actual" on-line version of the roster. Crew members should also be able to query data about transportation issues (e.g., transfer times from an airport to a hotel) and about accommodation (e.g., general hotel information, relevant contact details, earliest/latest check-in-time).

An important functionality addresses crew members' leave. Crew members expect to be able to declare their leave requests, change their requests and perform related activities, such as checking their current leave balance, leave status and entitlements. They also expect to receive an explanation if their leave requirement was turned down. Because crew members have personal preferences about duties (short hauls, long hauls, destinations, days-off, crew mates, etc.), a duty schedule that accurately reflects these preferences contributes substantially to a crew member's level of work satisfaction. For this reason, a rostering tool must provide crew members with the ability to declare his/her work preferences and to query, change and store them.

Training courses are another important design factor for a crew members roster. To keep or gain certain qualification levels, crew members must attend training units and lectures (e.g., flight deck crew members need to pass certain qualification tests periodically and attend flight, simulator and ground training courses). The rostering tool should therefore include an alert function that reminds the individual crew member to choose from among the training courses being offered and book him/herself a course. If the crew member does not use this option before a predefined deadline passes, then the system should automatically enroll that person in any course that is obligatory for them. Information about supervisors and course times and locations should also be provided. Crew members want to be able to query their own training record and to query, book and cancel course assignments. If a crew member has booked a course, he/she should also be able to check which of his/her colleagues will also be in the course; this information is analogous to the list of crewmates working together on a flight leg.

The other two user categories consist of crew assigners and crew operators, both of which involve few people in comparison to the number of crew members. The specific number of crew assigners required at any one airline depends on the technology being used (which in turn determines the degree of automation), as well as on the number and type of personnel for which the crew assigner is responsible. At a medium-sized airline with traditional technology, a crew-assigner usually handles 100 to 300 flight attendants or 50 to 150 persons for flightdeck. Traditional crew assignment activities are often split according to such factors as aircraft types and crew categories. With each crew assigner only bearing responsibility for the roster construction phase of a subproblem, this approach reduces the likelihood of creating a roster that is globally "good". From the viewpoint of a crew assigner, a rostering system should generate a set of pairings and, to a certain extent, provide an explicit control over the pairings generated (e.g., define desirable pairings). As the pairing process depends highly on alternative what-if scenarios, the system should not only generate suitable pairings, but also store, alter, drop and administer them on a perpairing basis, as well as on the basis of an individual scenario or a subsection of a scenario. Furthermore, the crew assigner should be given a choice among several rostering approaches or solvers and be able to define the one that seems most appropriate to him. He/she expects support in generating lines-of-work, in which pairings are combined to form a legal and valid duty for a specific person that conforms to that crew member's preferences. It should be possible for the crew assigner to query work rules and regulations, to define hard and soft rules and to make annotations. Given that the rostering process resembles the pairing process in relying heavily on a what-if scenarios technique, the crew assigner expects the system to create and manage both lines-of-work and roster alternatives. A rostering tool should also provide a function to quickly and reliably publish the final roster and pass it on to all crew members.

While the crew assigner works with a wide variety of roster versions and roster subsections, the crew operator only works with the one monolithic "real online version" of the roster. He distributes the released roster to all crew members and needs to take appropriate actions if a crew member has not confirmed that he/she has received his/her duties. In light of these requirements, the system should provide the crew operator with a list of those crew members who have not confirmed their duties by a certain deadline/due-date. As crew members are likely to comment on their duties or ask questions about them, the crew operator must be able to read these comments, compose answer messages and alter, delete and store the communications. A powerful query function about crew locations represents another prerequisite for making good decisions about changes to an existing roster.

Most of a crew operator's activities involve duty changes, which are increasingly time critical and have a decreasing set of solution alternatives the closer the start of a duty approaches. Less critical activities include duty swaps that crew members might propose right after roster dissemination in order to improve their schedules. In such cases, the crew operator checks conformity with work rules and regulations before performing the swap in the roster and notifying the relevant crew members about the performed swap. The critical activities in operations deal with crew recovery and aim at solving those crew plan disruptions that are due to such factors as maintenance problems or severe weather conditions. These disruptions often provide the reasons for very urgent and short term duty reassignments (e.g., stand-by's need to be activated, substitutes with the appropriate qualification for pilots who did not check-in on time have to be called in, etc.). The crew operator should be able to check the work rules and regulations and recent changes to them in a very efficient manner, as this ensures that they will be able to quickly reach a decision on such issues as overruling a soft constraint. Annotation and report functions should support the documentation process.

In order to provide a synoptic overview of the functionalities expected by the user categories "crew member", "crew assigner" and "crew operator", figure 1 applies the graphical design elements of the unified modeling language UML (cf., Booch et al., 1999).

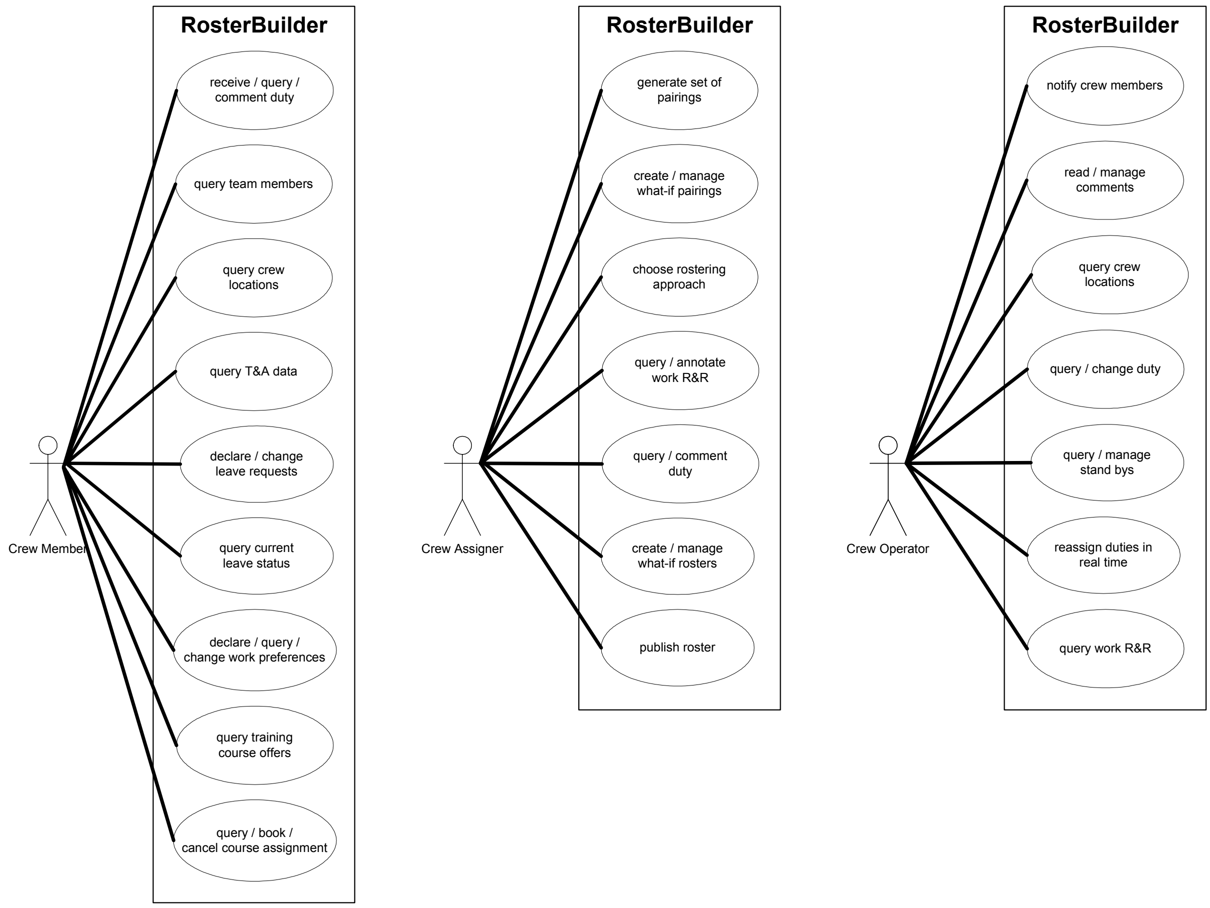

Figure 1: UML Use Cases for Actors „Crew Member", „Crew Assigner", and „Crew Operator"

3 Software Architecture

There is an observable trend towards component-based systems technology both in the commercial software market and in academic research (Herzum et al., 1999). Major software vendors have recognized the need for and the potential of component-based systems in providing customer-tailored solutions at lower costs. Research in software engineering, driven mainly by academia, has focused on the underlying core concepts, such as module interconnection languages, architecture description languages and architecture-driven system composition.

On the basis of the requirements analysis presented in section 2, we conclude that the airline sector would benefit from this approach and therefore propose a component-based reference architecture. The objective of a reference architecture is to describe the major components of the system, as well as their interdependencies and interfaces. A reference architecture should also address core functions and additional, optional functionalities, as well as security requirements and quality criteria such as performance (cf. Strauss, 2001).

3.1 Component and Deployment Diagram

A UML component and deployment diagram is used to illustrate the proposed software architecture (Figure 2). The component diagram describes the logical organization of the system. A component, represented by a rectangle with tabs, provides the physical building block of the system (a piece of code), and produces a group of operations by the means of interfaces, which are depicted as a line with a small circle. Dependencies between components are depicted as directed, dashed arrows.

The deployment diagram extends the component diagram in that it also includes physical resources. Physical resources, depicted as cubicles, are hosting (or executing) the components. Associations, which are physical connections between the resources, are depicted as solid lines.

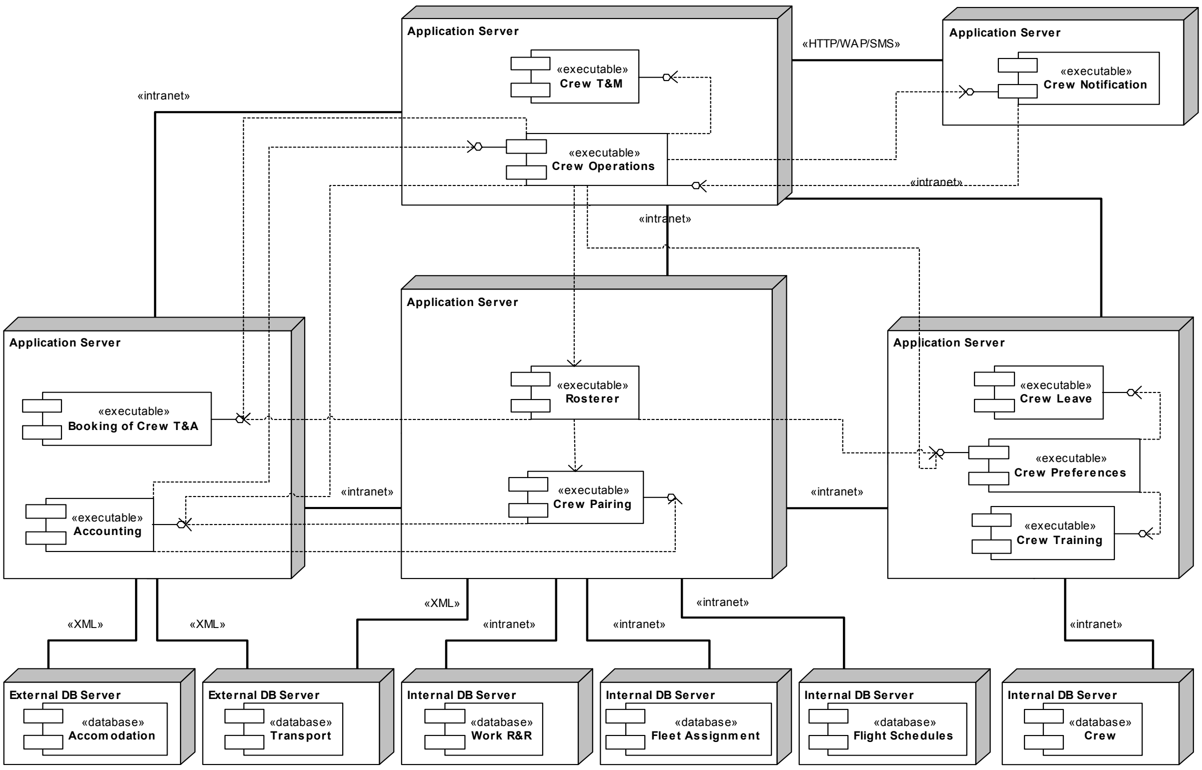

Figure 2: Component and Deployment Diagram of RosterBuilder

Rosterer and *Crew Pairing* (corresponding to the lowest level of the planning hierarchy as discussed in section 1) constitute the core components of the system. A small to medium-sized airline would typically host those two core components on a single application server, while larger airlines might consider hosting them on single servers for performance reasons. Both the rosterer and the crew pairing component would be accessed by the crew assigner and offer the functions illustrated in the use case involving the crew assigner in Figure 1. The other components, which have interfaces to internal and external databases support, are not directly accessible for the crew assigner, but offer services for the pairing and rostering processes. The *Crew Preferences*, *Crew Leave* and *Crew Training* components all interface with the *Crew* database, which contains the personnel data and demands a high level of security and privacy in its role as one the enterprise's core databases. As a consequence, the components accessing this database ought to be hosted on a separate server on which special security mechanisms are implemented. Both crew pairing and rostering are related to the planning phase, during which access to the *Accounting* system is needed in order to also include cost planning. Access to the services offered by the booking module (*Booking of Crew T&A*) enables the rosterer to make reservations for crew transport and accommodation.

The operational business starts once the roster for a certain time period has been published and involves two core supporting components: *Crew Operations* and *Crew Notification*. The *Crew Notification* component is the interface to the crew members, who can either actively query information on assignments (pull service) or receive notification from the system via push services. All functions described in the UML use case related to the crew member are provided via this component. The server deploying the *Crew Notification* component must be accessible in a multi-modal way, including internet access (wired and wireless), notification and query by SMS or other mobile communication technologies. *Crew Operations* provides the services for managing a roster once it has been published. This includes all functions related to communications with the crew members (see the first three functions in the UML use case involving the crew operator, Figure 1) and all functions related to changes on duties (either on demand from crew members or in real-time, e.g. due to emergencies).

As can be seen in the reference architecture, ten of the components are executables, i.e., they are (pieces of) code providing functions and interfaces to other components or end users. Four of those components (namely, crew pairing, rosterer, crew operations and crew notification) provide functionality to the end-users of RosterBuilder and are therefore called public executable components (providing interfaces to the public). The other six components are private[2] in the sense that they only offer their services via interfaces accessible to other components within the system.

The database components, hosted on database servers, store information needed by the executable components. Typically, information on accommodation and transport will reside on external databases (e.g., a tourism information system), while the other databases are most likely to be located within the airline's IT infrastructure.

3.2 Description of Public Executable Components

3.2.1 Crew Pairing

The service expected from the crew pairing component can be described in a simple sentence: taking flight schedules and fleet assignments as input to generate a set of pairings. However, the complexity of this service is significant with respect to qualitative and quantitative requirements.

There are three major considerations to be observed in generating pairings: first of all, feasible solutions must be generated, i.e., the scheduled flights must be serviced with sufficient and qualified personnel according to fleet assignment. As finding all possible pairings for given flight schedules and fleet assignments is typically a complex problem, the pairing component must assist the crew assigner in this process. Therefore, the *degree of automation* offered in finding pairings represents a central quality criterion. Solutions range from manual approaches, in which tool support is restricted to storing and visualizing the generated pairings, to semi- or fully-automated generation of pairings. The latter include functions for re-applying pairings for following time periods, fixing some pairings and re-planning of others, generating alternatives, automatically correcting overbooked pairings, performing downgrades, etc. As the generation of pairings is a *computationally hard problem*, performance is another critical criterion: the tool must provide feasible solutions in a reasonably short amount of time. Some current research efforts are addressing the possibility of parallelizing the pairing process as a means of increasing its speed, see for example the Parrot Project (Parrot, 1999).

Secondly, the pairing procedure must consider work rules and regulations. A good tool support would control conformity to work rules and regulations, but would also allow deviations if they are necessary. The control function should not only send out alerts when violations of rules and regulations occur, but also provide warnings if some conditions are close to be violated (signaling function).

Finally, crew pairings are selected mainly according to cost criteria. The crew pairing component must have an interface to an accounting component, as costs are the most crucial criterion in evaluating the quality of the generated pairings. The ideal crew pairing component would not only generate sets of feasible pairings, but would include *optimization techniques*, suggest what-if scenarios and make it possible to query generated pairings according to different attributes (e.g., date, flight number, sector, fleet, route, flight time or other operational figures) to support planning, as well as provide a history or annotation function for tracking changes in pairings.

The output of the crew pairing process is the set of pairings, which constitutes the starting point for the rostering process. In addition, the pairing component offers an interface for reporting operating figures (e.g., costs and utilization), which are used by the accounting component.

---

[2] Please, note, that we are not using the term "private" in the strict sense as defined in UML, according to which a private function is only accessible within the component. We use the terms "private" and "public" to denote interfaces accessible from the outside or within the system.

3.2.2 Rosterer

The rosterer component supports the crew assigner in generating lines-of-work and in matching the workforce demand and capacity, i.e., selecting for each crew member exactly one duty out of a large number of alternatives (lines-of-work) which were generated for this specific crew member. While costs were the major criterion in pairing, the rostering process aims at generating "fair" assignments of lines-of-work, considering preferences of crew members where possible.

Similar to the pairing process, we can identify the minimum degree of support as assisting the crew assigner in finding at least a feasible roster, i.e., a way of covering all lines-of-work with qualified, available personnel. Restrictions to be considered are amongst the already mentioned fairness criterion, personal preferences of crew members (regarding preferred destinations, team members, schedules, etc.), work rules and regulations, crew leave (holidays, illness, temporary deficiency, etc.) and crew training.

In order to find a feasible solution, the actual qualification of crew members available at the home base must be considered. A good rostering component will be able to consider crew members with more than one home base and might include suggestions for transfer of personnel to avoid shortages. The minimum support necessary is the identification and visualization of those lines-of-work that have not yet been (adequately) assigned, while the best support once again consists of a (semi-)automated optimization procedure. To support fairness, the ideal rostering component would retrieve crew preferences in an anonymized way in order to keep decisions from being based on any possible personal preferences that a crew assigner might have for specific crew members; ideally, the rostering component would consider also crew preferences in an automated rostering process. An interface to a booking module is a desirable feature, as it allows the crew assigner to make appropriate reservations (and to also consider crew preferences) for crew transport and accommodation. Crew trainings should not only be considered as a period in time during which personnel is not available for regular work duties, but the rostering component should also pro-actively schedule mandatory trainings and observe stipulations on the number of hours that crew members have to work on certain aircraft types in order to maintain their respective licenses.

The rostering component must support different time scales for planning, ranging from a whole season to a couple of days and last-minute changes before publication. When the roster has been finalized, it should be automatically published to both all crew members and crew operations in order to ensure consistency. An electronic signature should be used to ensure authentification, integrity and validity of the published roster.

The usability of the rostering component represents a major concern in addition to the issues of privacy and security. The crew assigner is not typically an IT expert and needs process-driven support in his work. A good rostering tool would guide the crew assigner through the necessary steps in the rostering process. In the future, we might find learning and adaptive tools on the market that are able to capture and understand the working behavior of the crew assigner and would therefore provide a better level of support. Existing rostering procedures (e.g., Butchers et al., 2001; Dawid et al., 2001; Day and Ryan, 1997; Desaulniers et al., 1997; Gamache and Soumis, 1998; Gamache et al., 1999; Ryan, 1992;) could - after adaptation - be integrated into this module. Some approaches focus on skills (e.g., language skills) of the crew members to

raise the service level on board and the passengers content (König and Strauss, 2000a) or focus on qualifications – allowing short term downgrading of crew members widenes the spectrum of options for crew assigners and crew operators (König and Strauss, 2000b). Categorization schemes should be developed to quickly identify the abilities of the approach (in analogy to e.g., Krishnamoorthy and Ernst, 2000).
Due to the complexity of functions involved in the rostering process, this component itself must be customizable (cf. e.g., Mason and Nielsen, 1999).

3.2.3 Crew Operations

Once a roster has been published, the rosterer component no longer has access to it; all subsequent modifications and changes are performed with support of the crew operations component. Two major support functions are distinguished: (1) support for changes according to explicit requests from crew members and (2) support for ad-hoc changes necessary due to unforeseen circumstances (e.g., delays, maintenance reasons). These two functions are separated because of their very different demands. When changes are to be made on the basis of crew requests, basically the same functions must be performed as in the planning phase when generating the roster: appropriate alternatives must be sought and evaluated under the aspect of fairness; all changes must be documented and traceable.
In contrast, time is the critical resource in ad-hoc changes. Although the same type of functions are involved, the system must allow much greater flexibility, such as allowing violations of soft work rules and regulations. The main operative need is to find a solution as quickly as possible.
The Crew Operations component also provides a security function in that it is able to validate the integrity of the published rosterer and of subsequent modified versions and can always provide the authorized up-to-date version of the roster. In addition, access rights are administered in this component. Crew members who want to retrieve information on their rosters will need to identify themselves and their identity is subsequently checked in the crew operations module by means of electronic signatures.

3.2.4 Crew Notification

The crew notification component is the interface to and for crew members and as such must be able to inform crew members in a timely and ubiquitous (any time any place) manner (cf. e.g., Kotsis et al., 2001). Given these demands, the crew notification module must provide multi-modal access consisting not only of internet access and access via mobile phones, but also the integration of such traditional media as fax or wired phones. We generally distinguish between so-called push services, which involve information being sent from the system to the crew member (with the trigger for sending the information ariseing within the system) and pull services, in which the crew members trigger the delivery of information by explicitly querying information. The past few years have seen the presentation of research prototypes for such reachability and information delivery architectures (cf. e.g., Appenzeller et al, 1996). In the future, we might find commercial offers for such services on the market, and this function could potentially be outsourced as a result.

We propose that only the crew operations component ought to interact with crew notification, i.e., that all correspondence to and from crew members should be routed via this component. This principle of modularization is very common in software engineering. It not only enhances flexibility, but also contributes to security, as all identification and authorization procedures can be performed in an efficient, centralized manner. The type of information crew members will have access to is governed by their role and function in the roster. For example, in order to ensure privacy, location information is available for crew members in the same team, but not for other crew members. The notification component is also the communication media for changing rosters among crew members, although we expect this function to become less important by considering preferences already in the rostering process.

3.3 Description of Private Executable Components

3.3.1 Crew Preferences

The crew preference component administers all information related to crew personnel. This information includes crew availability (or unavailability due to leave or training), crew qualifications, home base(s) of crew members or crew preferences (regarding vacancies, buddy requests, destinations, number of spare days in between flights, etc.). Both the rosterer component in the planning phase and the crew operations component in the deployment phase access this information when assigning lines-of-work to crew members.

The quality of service can be improved by considering individual skills of flight attendants during the rostering process; furthermore the quality of work life of an employee depends on his or her possibility to participate in the rostering process, i.e., determine to a certain extent his or her own work schedule, which in turn also effects his or her performance.

The minimum expected support function provides information on preference requests to the crew assigner, i.e., highlighting lines-of-work where preferences have (not) been met or proposing "good" assignments, i.e., assignments in conformance with preferences. The preference component must keep track of the assignments where preferences have (not) been considered in order to ensure fairness, at least in the long-term. Managing crew preferences, especially in the planning and optimization phase, is a highly complex problem and cannot be represented in models in a straightforward way. Ongoing research work on finding formal representation of the preferences can be included in the framework. The ideal preference component would allow the crew assigner to formulate preferences as conditions to be considered in an automated rostering process.

The preference component also manages the preferences submitted from the crew members. As already discussed, the access for crew members is established via the notifications module and routed through the crew operations component. A direct interface offered from crew preferences to crew notifications is not desirable, considering that the notification component could well be outsourced. It is better to use the security functions embedded in crew operations to ensure safe access to crew preferences.

3.3.2 Crew Leave

The crew leave component administers all kinds of vacancies and offers an interface to query and modify the availability status of any crew member at any time. In the proposed reference architecture, this interface is not public to all other components, but rather is only accessible from the crew preference component in order to ensure security and privacy; any other component requiring information on availability must access this function via the crew preference module. In addition, crew members entering vacancy requests or requests for change of vacancies do not have direct access to the leave component, but must rather access their services via the notification component. Such a design is chosen for two reasons: on the one hand, the availability of a crew member is not only determined by its leave due to vacancies, but also on account of participation in training courses. Therefore, a centralized unit such as the crew preference module can aggregate the true availability status of crew members based on both crew leave and crew training. Combining all three components into a single component would also be an option, but at the costs of a loss in flexibility. If the components are separated, it is easier to perform modifications, updates or changes within each component without necessarily influencing the other components. The minimum support includes the ability to query the number of vacancy days available/spent, while a good optional component would also allow comments, specify if vacancy requests have not been considered and provide mechanisms for controlling the fair assignment of vacancies.

The consistency and timeliness of the information represents a crucial requirement in this component. The leave status of all crew members must always be up-to-date.

3.3.3 Crew Training

The crew training component keeps track of the training history and status (level) of each crew member. It schedules mandatory training courses for crew members. For flight-deck crew, mandatory regulations on the number of hours on certain aircraft types to maintain a certain qualification level must also be considered. The crew training component provides this information to the rosterer component, which must plan accordingly.

Optionally, this component can offer functions for administering internal training courses, which would include schedule timing, the assignment of course instructors and course participants (trainees) and the tracking of participation and successful completion.

3.3.4 Booking of Crew Transportation and Accommodation

The booking of crew transportation and accommodation (T&A) component offers an interface to external booking systems that is accessible by the rosterer and by the crew operations component. The component must know about up-to-date rates, check-in/check-out times, travel distances, etc., as all these parameters influence the costs incurred by generating pairings and lines-of-work.

From a technical point of view, the key challenge in the component involves using the appropriate interface to the external booking systems (Bichler et al., 2001). While EDI/EDIFACT has not really proven itself as a vital, worldwide standard, current initiatives to extend the use of XML (Extended Markup Language) to electronic commerce may offer a solution to this problem. By using XML, accommodation and transportation information can be exchanged in a standardized - and thus, machine-processable - manner.

3.3.5 Accounting

The accounting component offers cost-tracking services. It will have an interface to the other accounting programs within the airline, including salary payment systems

It must provide specific functions, such as calculating profit on routes, calculating travel costs to/from home base, accounting for costs on a per home base basis, providing cost estimates for future alternative flight schedules and allowing comparisons of what-if scenarios on costs.

3.3.6 Crew Tracking and Monitoring

The crew tracking and monitoring (T&M) component allows the crew operator to locate and view the (graphical) position of crew members. From a technical point of view, techniques such as GPS or the mobile phone-based location services could be used, although privacy remains a critical consideration. In any case, an interface to the airport's crew check-in systems is needed to verify that the complete crew arrived on time. Location information is also needed in order to track – to provide an example - those flights that are over- or under-staffed, or to establish the current status of crew member's block hours. The crew operator’s attention must be drawn to such problems, and he should be able to request alternative suggestions from the system. Thus, he must also be allowed to ignore them if no other solution can be found.

A subset of location information is also provided to crew members, who should be able to view and track the location of their team members if needed. In fact, crew members might be a valuable source of information for T&M information. For example, a pilot should be able to report immediately to crew operations, if his or her co-pilot does not show up in time for the preparatory meetings.

3.4 Description of Database Components

3.4.1 Internal Databases

Crew. The crew database contains all information on crew personnel, including name, address, reachability, qualifications, etc. This information must be treated as personal and confidential. Therefore, only authorized entities must be allowed access to this database. As a single interface contributes to security, only the crew preference module accesses this database in the proposed architecture.

Work rules and regulations. Both the work rules and regulations defined by external authorities (e.g., government) and the internal (i.e., airline-specific) rules and regulations are stored in this database[3]. This

[3] Current research focuses on reducing the complexity of explicit rules and regulations by defining few restriction categories (cf. e.g., Unger, 2002).

information is accessed by the preference module, which the rosterer can use to make annotations to rules and regulations (e.g., identifying hard and soft rules).

Flight Schedules and Fleet Assignment. The flight schedules are maintained in an airline-specific database that contains information on which flights have to be operated and when. The fleet assignment represents an additional airline-specific database and contains information on the assignment of aircraft types to specific flights. As information on schedules and assignments determines the number and qualifications of crew members needed, this database must be accessible to the crew pairing component, which annotates the generated pairings accordingly. Any subsequent allocation of individual crew members during planning (rosterer) or operations does not access these databases, but rather uses the annotations in the rosterer where all necessary information can be found. This approach decouples the access to databases and might increase the performance during rostering and operations.

3.4.2 External Databases

Information on available hotels, discount rates, check-in/out times, location, booking conditions, booking interface, options for transportation, including flights, train, bus, taxi etc., must be available in the planning phase and in operations. A direct, on-line interface to external hotel booking systems and transportation databases is a desirable feature. The most crucial issue is the definition of appropriate protocols for the exchange of information. We propose to use XML as a means of describing this information in a standardized way. This choice is motivated by observable trends in electronic commerce towards the deployment of XML for exchange of business information (c.f., Bichler et al., 1998).

4 Conclusions

Having recognized the crucial role of rostering in the airline business, the need for appropriate support tools becomes apparent. Crew assigners need fast and flexible tools and methods in the planning phase, crew operators need reliable real-time support and services for *ad hoc* changes in the operational phase, and crew members demand interactive involvement into the rostering process, the fulfilment of their personal preferences and timely and ubiquitous access to rostering information. This diversity of user requirements is our first argument in favor of a component-based software architecture, as opposed to a monolithic approach. By implementing a component-based software architecture, each component can be tailored to specific needs of the end users involved.

The structure of the market provides another factor promoting the deployment of components. Given the comparatively small number of customers and their specific needs, it is neither technically nor economically feasible for a single software vendor to develop and market "the" rostering tool. A component-based approach gives sufficient flexibility for customization or for the integration of products from other sectors. By selling components and adaptation services instead of off-the-shelf software, software vendors will be able to address a larger number of potential customers and to meet the demands of their customers more effectively.

In this paper, we have offered a detailed presentation of the specific requirements placed on a rostering framework and have derived a generic software architecture for RosterBuilder on the basis of this process of determining requirements. In the proposed architecture, crew pairing, rosterer, crew operations and crew notifications are the core components accessed by the users of the system, which consist of the crew assigner, the crew operator and the crew member. Enhanced functionalities can be provided through the integration of additional components, such as booking or accounting components in the crew pairing process. This specific integration supports the crew assigner in performing what-if scenarios and in evaluating the costs of alternatives, which represents the most important selection criterion in crew pairing. An interface to a crew preference component in the crew rostering process allows crew members to participate in this process: crew members can state their preferences, receive feedback on why their preferences were or were not taken into consideration; comment on rosters or even use the system to swap duties (in an operations-controlled way) once the roster has been published. These functions contribute to transparency of the crew rostering process and help to ensure its fairness, thus contributing directly to the crew members' level of satisfaction and indirectly improving work performance levels.

We expect the reference architecture introduced in this paper to be of value for airlines, which can use the framework as a comparison schema on which to match their needs with the products on the market. A survey of rostering tools could also make use of this framework to provide a general classification of existing components and modules on the market.

Future work will be twofold: one research direction will focus on a formal definition of interfaces among the components, which is an important prerequisite for tool builders interest in integrating their products into a general roster framework. A clear distinction should be made between those functions and data inputs that are mandatory and those that are optional.

Another research direction will concentrate on advanced approaches to cooperation, such as ensuring that the module managing crew preferences provides mechanisms that support the crew members in their decision-making in the "market of preferences" (e.g., auctions). Another advanced approach that contributes to crew satisfaction can be found in the concept of partially autonomous and self-organizing crews, which could be achieved by providing a team portal as a part of the notification module.

References

AEA - Association of European Airlines (2001). *Yearbook 2001*. http://www.aea.be/Yearbook/download/ Yearbook01.pdf

Appenzeller, G., Lai, K., Maniatis, P., Roussopoulos, M., Swierk, E., Zhao, X. and Baker, M. (1996). *The Mobile People Architecture*. Technical report, Stanford University, 1996.

Barnhart, C. and Talluri, K.T. (1997) "Airline Operations Research." In A. McGarrity and C. Revelle (eds.), *Design and Operation of Civil and Environmental Engineering Systems*. New York: Wiley, p. 435-469.

Barnhart, C., Lu, F. and Shenoi, R. (1998). "Integrated Airline Schedule Planning." In G. Yu (ed.): *Operations Research in the Airline Industry.* Norwell: Kluwer, p. 384-403.

Barnhart, C., Johnson, E. L., Nemhauser, G. L. and Vance, P. H. (1999). "Crew Scheduling." In R. W. Hall (ed.), *Handbook of Transportation Science*. Norwell, Kluwer, p. 493-521.

Bichler, M., Segev, A. and Zhao, J.L. (1998). "Component-Based E-Commerce: Assessment of Current Practices and Future Directions", *ACM Sigmod Record: Special Section on Electronic Commerce* 27, 7-14.

Bichler, M., Hansen, H.R., Kotsis, G. and Neumann, G. (2001). "Collaboration on Electronic Markets, in Information Age Economy." In Buhl, Huther, and Reitwieser (eds.), *Proceedings of the WI-Tagung*, Physica.

Boeing Commercial Airplane Group (2001): 2001 Current Market Outlook. Meskill, T., Griffiths, J. (eds.), Boeing Commercial Airplane Group, Seattle, June 2001.

Booch, G., Rumbaugh, J. and Jacobson, I. (1999). *The Unified Modeling Language User Guide*, Addison-Wesley.

Butchers, E.R., Day, P.R., Goldie, A.P., Miller, S., Meyer, J.A., Ryan, D.M., Scott, A.C. and Wallace, C.A. (2001) "Optimized Crew Scheduling at Air New Zealand." *Interfaces* 31, 30-56.

Dawid, H., König, J. and Strauss, C. (2001). "An enhanced rostering model for airline crews." *Computers and Operations Research* 28, 671-688.

Day, P. R. and Ryan, D. M. (1997). "Flight attendant rostering for short-haul airline operations." *Operations Research* 45, 649-661.

Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S., Rioux, B., Solomon, M. M. and Soumis, F. (1997). "Crew pairing at Air France." *European Journal of Operational Research* 97, 245-259.

Gamache, M. and Soumis, F. (1998). "A method for optimally solving the rostering problem". In G. Yu (ed.): *Operations Research in the Airline Industry.* Norwell: Kluwer, p. 124-157.

Gamache, M., Soumis, F. and Villeneuve, D. (1998). "The Preferential Bidding System at Air Canada." *Transportation Science* 32, 246-255.

Gamache, M., Soumis, F., Marquis, G. and Desrosiers, J. (1999). "A Column Generation Approach for Large-scale Aircrew Rostering Problems." *Operations Research* 47, 247-263.

Herzum, P. and Sims, O. (1999). *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise*, John Wiley & Sons.

Jeanniot, P.C. (2002). *Annual Report*. International Air Transportation Association for the 58th Annual General Meeting, Shanghai. http://www1.iata.org/NR/ContentConnector/CS2000/SiteInterface/pdf/ar02.pdf

Kotsis, G., Deiters, W. and Ferscha, A. (2001): "Pervasive Computing and Information Logistics. " In K. Bauknecht, W. Brauer, and T. Mück (eds.), *Proceedings of the GI/OCG-Jahrestagung 2001*, Vienna: OCG Schriftenreihe, Bd. 157.

König, J. and Strauss, C. (2000a). "Supplements in Airline Cabin Service." In D.R. Fesenmaier, S. Klein and D. Buhalis, (eds.), *Information and Communication Technologies in Tourism 2000*. Wien: Springer, p. 365-374.

König, J. and Strauss, C. (2000b). "Rostering-integrated Services and Crew Efficiency." *Information Technology & Tourism* 3, 27-39.

Krishnamoorthy, M. and Ernst, A. (2000). "The Personnel Task Scheduling Problem." In X. Q. Yang, K. L. Teo and L. Caccetta (eds.), *Optimization Methods and Application*, Norwell: Kluwer.

Mason, A.J. and Nielsen, D. (1999). *PETRA: A Programmable Optimisation Engine and Toolbox for Personnel Rostering Applications*. Presented at the 15th Triennial International Federation of Operational Research Societies Conference IFORS 99, August 16-20, 1999, Beijing, China; available as School of Engineering Technical Report 593, 1999.

Parrot Project (1997). Executive Summary, Esprit 24960, 1997, http://www.upb.de/~parrot/

Ryan, D. M. (1992). "The solution of massive generalized set partitioning problems in air crew rostering." *Journal of the Operational Research Society* 43, 459-467.

Strauss, C. (2001). *Quantitative Personaleinsatzplanung im Airline Business*. Frankfurt/Main: Peter Lang Verlag.

Unger, G. (2002). Modellierung von Arbeitszeitregeln im Airline Crew Rostering mit Constraint Programming, diploma thesis, University of Vienna (in progress, supervisor C. Strauss).

Wren, A. (1996). Scheduling, Timetabling and Rostering – A Special Relationship? In G. Goos, J. Hartmanis, J. van Leeuven (eds.), *Practice and Theory of Automated Timetabling*. Berlin: Springer, p. 46-75.