From Theory to Practice: AI Planning for High Performance Elevator Control (Extended Abstract)

Jana Koehler
Schindler Aufzüge AG

current address: IBM Zurich Research Laboratory, CH-8803 Rüschlikon, Switzerland
email: koe@zurich.ibm.com, http://www.informatik.uni-freiburg.de/~ koehler

Abstract. Offering an individually tailored service to passengers while maintaining a high transportation capacity of an elevator group is an upcoming challenge in the elevator business, which cannot be met by software methods traditionally used in this industry.
AI planning offers a novel solution to these control problems: (1) by synthesizing the optimal control for any situation occurring in a building based on fast search algorithms, (2) by implementing a domain model, which allows to easily add new features to the control software.
By embedding the planner into a multi-agent system, real-time interleaved planning and execution is implemented and results in a high-performing, self-adaptive, and modular control software.

1 Customization and Efficiency challenge Innovation

As many other industries, elevator companies are facing two main challenges today: (1) the pressure on building costs requires to improve the transportation capacity of elevator installations, (2) increasing competition challenges diversification and mass customization strategies, which aim at providing new and individually tailored services to passengers.

Schindler Lifts Ltd. has developed the Miconic-10™ elevator *destination control* where passengers input their destination before they enter the elevator. The Miconic-10™ control has been introduced into the market in 1996 and since then more than 100 installations have been sold worldwide. A 10-digit keypad outside the elevators allows passengers to enter the floor they want to travel to. After input of the destination, the control system determines the best available cabin and the terminal displays, which elevator the passenger should take. The current Miconic-10™ control uses a rule-based, heuristic allocation algorithm. It groups passengers with identical destinations together and thereby allows them to reach their destination faster and more comfortable.

By using identification devices such as smart cards, pin codes, or WAP phones, passengers can even be recognized at an individual basis and more individually tailored services can be implemented: *access restrictions* of certain passenger groups to certain zones in the building, *VIP service* to transport some passengers faster than others, e.g., medical emergency personal, and the *separation of passenger groups* that should not travel together in the same elevator.

Today, elevator systems can offer these services only in a very limited way by permanently or temporarily restricting the use of selected cabins. It is impossible to integrate and embed these functionalities directly into the usual normal operation of an elevator group and the restricted usage of some of the cabins dramatically reduces the transportation performance.

The work summarized in this abstract has been driven by a rigorous formal approach. In the early studies in 1998 and 1999, the complexity of the problem has been investigated and it has been proven that even in the simplest case, destination control is NP-hard [4]. Based on these results, a comparative analysis has been conducted, which modeled destination control as a *planning* problem, as a *scheduling* problem, and as a *constraint satisfaction* problem [3, 2]. Modeling the problem from a planning perspective seemed to be the most natural approach [1] and resulted in the *miconic10* domain, which has also been used in the AIPS-00 planning systems competition. The formal modeling of relevant domain properties in PDDL allowed to precisely define the services and prove the domain-specific algorithm to be sound and complete. It also helped in generating test problems that were used to verify the implementation.

There are two aspects of the problem: The *static, offline optimization problem* for one elevator, which requires to compute an optimal sequence of stops for a given, fixed traffic situation in a building. The *dynamic, online decision problem*, which needs to cope with the immediate and unknown changes of traffic situations.

2 The Offline Problem: A Case for AI Planning

The offline optimization problem for one elevator is given by a particular traffic situation in a building: the state and position of the elevator, the unanswered destination calls of passengers who are waiting in the building, and the destination calls that have already been picked up and where passengers are currently traveling in the elevator towards their destination. Given a number n of destination calls with board floor b and exit floor e as $(b_1, e_1), (b_2, e_2), (b_3, e_3), \ldots, (b_n, e_n)$ we need to compute a totally ordered sequence of stops $s_1, s_2, \ldots, s_k$ such that each s_i corresponds to a given board or exit floor (no unnecessary stops should be contained in the sequence) and where each b_i is ordered before each e_i (since passengers have to be picked up first and then delivered to their destination).

In practice, one is interested in finding stop sequences that minimize a given optimization criteria, e.g., minimizing the average waiting times of passengers or minimizing the overall time the passengers spend with the elevator from the moment they insert the call until they reach their destination.

The domain specific planning algorithm constructs a sequence of stops for a single elevator out of the stop actions that are applicable in a given situation. The following properties turned out to be key to success: (1) the state representation and the choice of data structures to implement it, (2) the search algorithm as a combination of a depth-first, branch-and-bound search with forward checking techniques from constraint reasoning, (3) an admissible heuristic function that estimates the distance to the goal state in such an effective way that the branching factor in the search space is reduced by 60 to 90 %. The planning system allows to find optimal plans of up to a length of 15 to 25 stops in less than 100 milliseconds even in search spaces containing between 10^{12} and 10^{15} states, which are frequently generated by data from real buildings with high traffic peaks. The planning system is able to deliver optimal plans given the tight real-time requirements because it works on the level of destination commands and returns an abstract sequence of stops. The execution of these plans requires to translate the stop sequences into the much more fine-grained level of elevator control commands.

3 The Online Problem: Real-Time Interleaved Planning and Execution with Failure Recovery Mechanisms

For each elevator, a so-called *jobmanager* has been developed, which controls a single cabin with its drive and various doors. A jobmanager is a holon of agents responsible for different tasks in the control, see Fig. 1. The agents communicate via an asynchronous messaging system supporting publish/subscribe mechanisms and allowing a peer-to-peer communication between the control and hardware components such as drives, doors, and terminals.

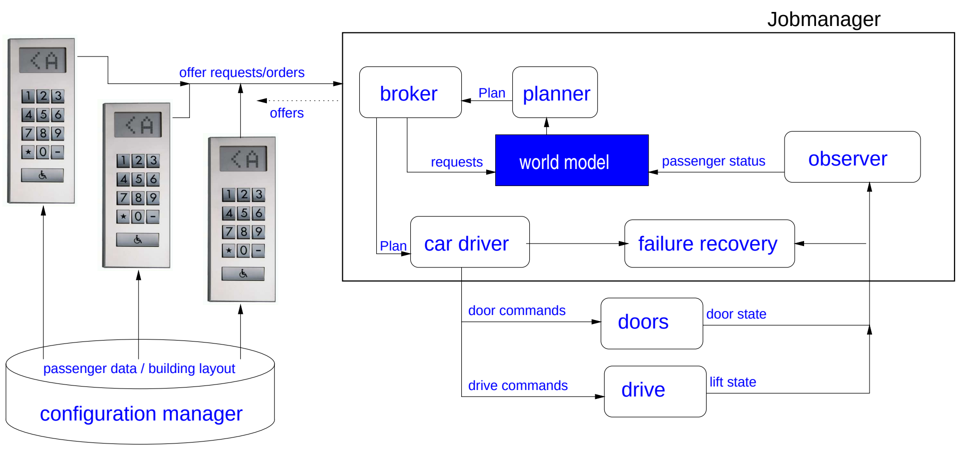

Fig. 1. The jobmanager as a multi-agent system implementing interleaved planning and execution for a single elevator.

The **broker** receives the offer requests from the terminals and adds the new calls to the world model, which is part of the initial state representation for the planner. It initiates the planning process and evaluates the returned plan. Based on the evaluation, it sends offers to the requesting terminals, which book passengers via an *auctioning* process. The **car driver** is responsible for executing the plans. Given an abstract sequence of stops, the car driver maps these into a fine-grained temporal sequence of activities, e.g., accelerating, moving, landing, opening doors, etc. The **observer** updates the world model of the planner solely based on information that it receives from the doors and the drive. This model updating independently of the car driver's actions is very important to keep world model and reality in accordance. The **failure recovery** maps the activities of the car driver to the information it receives from drive and doors and verifies whether the planned activities are correctly executed. It implements a very flexible approach to deal with hardware failures, situations that made world model and reality fall apart, and passengers who behave not as assumed. The **configuration manager** provides information about the building layout, i.e., the number of floors, access zones, passenger groups and access rights, active services, etc.

Each component is a self-acting agent that initiates activities when certain events occur. This can trigger several agents simultaneously and their activities can run in parallel. The distributed control is able to deal with such interfering events.

The developed control software has achieved two major results: First, we obtain a much more modular and compact code comprising only 8000 lines in an object-oriented language. The underlying agent-based architecture leads to very clear interfaces and allows it to further develop the different agents independently of each other. The planning algorithm is able to unify the software for various elevator platforms, e.g., cabins with multiple decks.

The situation-dependent optimization improves the transportation capacity of elevator systems significantly. The currently used heuristic allocation algorithm already improved performance by 50 to 80 % when compared to conventional control systems where passengers press the destination buttons inside the cabin after they have been picked up. With the intelligent planning system, performance is increased by another 10 to 30 % depending on the traffic pattern. Average waiting times remain roughly identical, but travel times within a cabin get reduced by up to 35 %.

References

1. J. Koehler and K. Schuster. Elevator control as a planning problem. In S. Chien, S. Kambhampati, and C. Knoblock, editors, *Proceedings of the 5th International Conference on Artificial Intelligence Planning and Scheduling*, pages 331–338. AAAI Press, Menlo Park, 2000.
2. Jana Koehler. Von der Theorie zur Praxis: Verkehrsplanung für Hochleistungsaufzüge. *Künstliche Intelligenz*, 2, 2001.
3. B. Seckinger. Synthese von Aufzugssteuerungen mit Hilfe von Constraintbasierten Suchverfahren. Master's thesis, Albert-Ludwigs-Universität Freiburg, 1999.
4. Bernhard Seckinger and Jana Koehler. Online-Synthese von Aufzugssteuerungen als Planungsproblem. In *13. Workshop Planen und Konfigurieren, Interner Bericht des Instituts für Informatik der Universität Würzburg*, pages 127–134, 1999.