Context-Aware Elevator Scheduling

Thomas Strang
German Aerospace Center (DLR)
Institute for Communications and Navigation
D-82230 Wessling-Oberpfaffenhofen, Germany
Email: thomas.strang@dlr.de

Christian Bauer
University of Innsbruck (UIBK)
Institute for Computer Science
Innsbruck, Austria
Email: {thomas.strang|csad2789}@uibk.ac.at

Abstract— **Research on context-aware systems is usually user-centric and thus focussed on the context of a specific user to serve his or her needs in an optimized way. In this paper, we want to apply core concepts developed in research on context-awareness in a system-centric way, namely to elevator systems. We show with three different examples that the performance of an elevator system can be significantly improved if the elevator control has access to contextual knowledge. The first example demonstrates that an appropriate elevator group scheduling algorithm can be chosen and adjusted to the current and near-future passenger traffic situation. The second example incorporates a second predominant technology used in many ubiquitous computing environments, namely RFID tags, to predict the passengers destination. The third example shows that adaptation to a specific emergency situation can further increase the transportation capacity of an elevator system which is ultimately important to save as many lifes as possible in an emergency. Evidence for the increased efficiency is provided by using a comprehensive simulation of a diverse set of scheduling algorithms and scenarios.**

I. Introduction

The efficiency of an elevator service is vital for highly populated buildings as it is usually the only reasonable fast method of moving from one floor to another. Passengers expect elevators to respond to their calls virtually in no time and to reach their target floor as fast as possible. While we see this service as something not uncommon and absolutely naturally in everyday life, the task of assigning elevator cars to calls in such a way that the average journey time (AJT)[1] is indeed as small as possible is far from trivial. In fact, finding a schedule with the minimal number of stops is a NP-hard problem [SeKo99].

In a single elevator system a single controller handles all calls, whereas in the case of several elevators, each in its own shaft, a group controller decides to which elevator a call is assigned. Adaptation to changing traffic patterns or fluctuations in traffic can be achieved with artificial intelligence and fuzzy logic, usually based on statistical forecasts [Siik97].

Completely different from this approach, we propose to make use of contextual knowledge to provide controllers with otherwise non-accessible information. In particular, we propose to switch between different scheduling group algorithms as each of them has some advantages during a specific set of traffic patterns, and disadvantages during other traffic patterns. Thus predicting upcoming traffic demand not only based on statistics but much more accurately based on contextual knowledge such as the occurrence of a crowded meeting on a specific floor results in a better overall performance of the elevator system as we will show.

The remainder of this article is structured as follows: The subsequent subsections introduce our model of elevator installations in use in as well as the various traffic patterns that occur throughout the day. Section II gives an overview of scheduling algorithms used in current elevator control systems. Section III introduces context for elevator scheduling and how this context information could be acquired. Section IV explains the simulation and its parameters we've used and gives insight to the scenario details. Finally, section V presents the results obtained from the simulation runs.

A. Elevator Model

As elevator modelling and related parameters and algorithms are probably not common to the ubiquitous computing community, we will give a short introduction to the key components and how we use them in the following sections.

A building is defined as consisting of N floors (excluding the lobby) and L elevator cars, where these cars with index $\{0, 1, \ldots, L-1\}$ are either idle and standing on one of the floors $\{0, 1, \ldots, N\}$ or moving with direction UP or DOWN.

Floor calls are calls initiated by passengers waiting on a floor by pressing the buttons UP or DOWN, where the former indicates that a passenger would like to travel to a floor which is on a higher level of the building than the current level and the latter indicates that a passenger would like to travel to a floor which is on a lower level of the building than the current level. *Car calls* are calls initiated by passengers who boarded an elevator and press one of the floor buttons inside the elevator car, where the call floor corresponds to the passengers' desired exit floor.

Floor calls are registered with the *Group Elevator Controller*, which assigns the calls to one of the cars based on the result of a car selection algorithm. Assignment may be delayed for as long as no elevator with sufficient capacity is available.

Car calls are immediately assigned to the *Elevator Controller*, which is responsible for sorting calls assigned by the

[1] AJT is defined as the sum of the average waiting time (AWT - the time between registering a floor call and boarding the elevator) and the average ride time (ART or Service Time in queueing theory terminology - the time between boarding and unboarding the elevator)

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

IEEE
COMPUTER
SOCIETY

group elevator controller into the elevator's internal sequence list. If at least one call is assigned, the elevator moves into the target's floor direction, that is, the level of the first call of the sequence list.

Once the target floor is reached, the following operations are performed:

1) If the elevator's current floor is equal to the exit floor of a boarded passenger, the passenger leaves the elevator.
2) Passengers waiting on the current floor are boarding the elevator[2].
3) Calls which refer to the current floor (floor and/or car call) are removed from the sequence and a new target floor is set.

In case the sequence list is empty, the elevator is considered to be in idle state.

B. Traffic Patterns

Traffic demand is not static and changes over time, especially with respect to the time of day. Consider a typical situation in an office building as shown in figure 1. Here the first situation is the up-peak in the morning, with people entering the building in the lobby and travelling upwards to their offices. Nearly everyone arrives at the ground level and the destination calls are uniformly distributed over all (office-)floors.

Very similar, but reversed to the up-peak, is the down-peak in the late afternoon, with workers leaving their offices and heading downwards to the lobby. Passengers arrive uniformly distributed over all floors and for most of them the lobby is the target floor.

The third typical situation to be handled by the elevator control system is the mixed peak at lunchtime with people simultaneously leaving and entering the building. This situation can be decomposed into a decreasing down-peak, an increasing up-peak and a small, almost constant interfloor passenger volume, all over a defined period of about 2 hours (confer markers in figure 1).

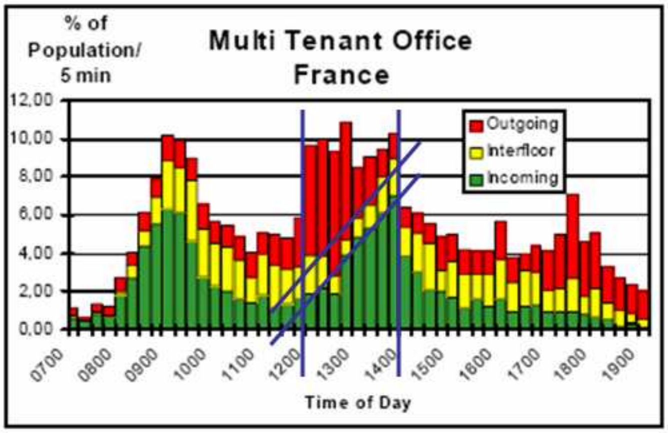

Fig. 1. Typical traffic pattern in a multi-tenant office building [SHS05]

II. Scheduling Algorithms

Elevator scheduling is performed on two levels, *Group Elevator Controller* and *Elevator Controller*. The group elevator controller is in charge of all elevator banks and performs the global planning. Once a call is assigned to an elevator, the appropriate elevator controller performs local planning in the scope of the single elevator.

The primary objective of group elevator scheduling is to assign elevator cars to floor calls in the most effective way, i.e. such that the passenger's average journey time becomes minimal. Many different group scheduling algorithms have been developed over time, ranging from simple Answer-The-Closest-Call-First up to highly sophisticated AI algorithms [KoOt02], [NiBr03]. We will give an overview of the basic layout of group elevator scheduling in the subsections below. The selection covers the major set of algorithms actually used in everyday installations.

Scheduling on the elevator controller level (local planning) is usually performed in a one-way fashion: The elevator answers all calls along the current movement direction until the last floor with calls - the reversal floor - is reached, where direction is reversed and all calls along the new direction are answered until the next reversal floor.

A. Round-Robin Group Elevator Scheduling

Round-Robin is a simple first-come-first-served approach, easy to implement with the idea of offering a not too bad service, especially for not too demanding situations. The goal of Round-Robin in group elevator scheduling is to achieve an equal load for all cars. Calls are assigned in the order they arrive in a sequential way to single elevators. Call 0 is assigned to car 0, call 1 to car 1, $\ldots$call L to car L, call $L+1$ to car 0, and so on.

B. Up-Peak Group Elevator Scheduling

Up-Peak scheduling is a special variant of Round-Robin for use in up-peak traffic situations, with the only difference being the use of a special parking policy: In case an elevator is idle, the group elevator controller creates a floor call with level 0 in order to relocate the car to the lobby. The idea is to reduce the waiting time for future passengers arriving at the lobby.

C. Zoning Group Elevator Scheduling

The idea of *zoning* is to split a (usually high-rise) building into several adjacent zones, with every elevator only serving floor calls that occur in the zone assigned to the respective car. The primary objective of this approach is to reduce the number of car stops and therefore the total journey time. A building served by m cars can be split into up to m zones, where these m zones are either disjoint or not. Cars in idle state are repositioned to the zone's center level, therefore minimizing waiting time for passengers on adjacent floors. Zoning may be either static, where zones are assigned permanently to a group of elevators, or dynamic, where zones are assigned temporarily and time-scheduled ([ChSo97]).

[2] Of course, passengers can only enter an elevator if the maximal passenger capacity has not already been reached. Otherwise, they initiate a new floor call and wait for the next elevator.

D. Three Passage Group Elevator Scheduling

The *Three Passage Algorithm* ([CGK99], [CGK01]) is a special variant of the *Estimated Time of Arrival (ETA) Based Elevator Group Control Algorithm* introduced in [RHL03]. It is used to determine the service order of the floor calls:

P1 Passage One calls can be served by the elevator along the current travel direction.
P2 Passage Two calls can be served after reversing the direction once.
P3 Passage Three calls require reversing the direction twice.

Floor calls with direction UP are only served by an elevator when travelling upwards, calls with direction DOWN only when travelling downwards.

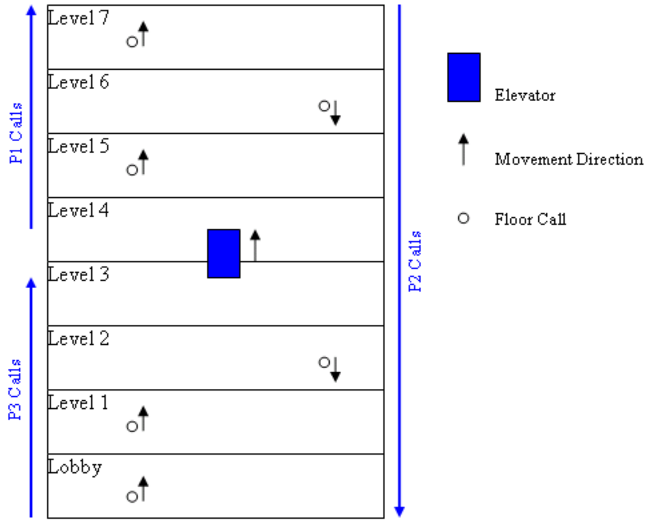

Fig. 2. Three Passage Concept

The principle of this approach is outlined in Figure 2, showing a high-rise building with a single elevator car currently moving upwards, about to reach floor 4, and several floor calls with the highest one occurring on level 7. The calls from floors 5 and 7 with direction UP can be served along the elevator's current movement direction and are therefore P1 calls. Level 7, being the highest floor with the last calls to serve, is the *reversal floor*, as the elevator will have to reverse its travel direction at this position in order to serve the remaining calls. The calls from level 6 and 2, both with direction DOWN, are therefore P2 calls, as direction is reversed once prior to serving. Starting from level 7 the elevator moves downward to the next reversal floor, the lobby, where direction is reversed for the second time before answering the calls from the lobby and level 1, therefore making these remaining calls P3 calls. Car calls are always either P1 or P2 calls, as they require none or one reversal at most.

Additionally, prior to assigning a call to a car, the group elevator controller submits the call to every elevator controller which perform a cost analysis. This analysis estimates the costs that would result from assigning the new call to the elevator, as the necessary amount of time needed for serving all calls could increase because of an additional stop and a longer passage. Stop costs are static and include the period of time necessary for opening the door, unloading and loading each one passenger and closing the door. Finally, the call is assigned to the elevator with the lowest costs.

In case every elevator already reached 80% load (number of calls $> 80\%$ car capacity), calls will not get assigned until at least one elevator falls below this mark.

III. Context-Aware Elevator Systems

Elevator systems become context-aware in the sense of Dey [Dey99] through the use of context information for whatever reason (here: to increase efficiency). The goal of enabling an elevator to be context-aware is to enhance the service provided to passengers by reducing the average waiting and ride time. Obviously, a well designed context model is a key accessor to the context in any context-aware system, cf. [Stra04] for a survey of the most relevant approaches to modelling context. Usually context information is either directly obtained from some kind of sensor or derived by fusing two or more context information.

In this paper context information in the elevator model is viewed as knowledge that is not accessible to controller systems currently in use. This information consists of specific passenger and environmental data, both of individual and global range, and is retrieved with the help of the general concept of sensors. Depending on the type of sensor output, either the group or elevator scheduler can benefit from and adapt to additional input parameters, or a completely different algorithm, one more appropriate for the current traffic situation, may be chosen. In the following we will give three different examples where the integration of context information showed a significant improvement of the elevator systems efficiency in our simulations.

A. Example I - Context-Aware Group Scheduler Switching

We found out that switching to the scheduling algorithm which is optimal for the current and near future traffic situation results in a better performance compared to switching based just on statistics. This decision could be made by an additional switching controller, hierarchically above the group elevator controller, with direct access to the context information.

One example where this becomes illustrative obvious is the correlation between the weather and the passenger traffic pattern during lunchtime in high-rise office buildings which do contain lunch-time facilities like restaurants. Assuming a restaurant is located at the top level as indicated in figure 3, offering a roof-deck for guests, we could conclude that office workers will head to the top level during lunch hours on a sunny summer day, therefore overlaying the usual lunch hour traffic pattern (section I-B). If we can conclude from some weather sensors that the roof-deck has high acceptance, the passenger traffic demand conforms to a up-peak instead of lunch-peak. Contrary, rain, snowfall, high wind speed or low

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

temperature will have no effect on the standard lunch-peak demand with people visiting a mid-level canteen or leaving the building seeking for lunch somewhere else.

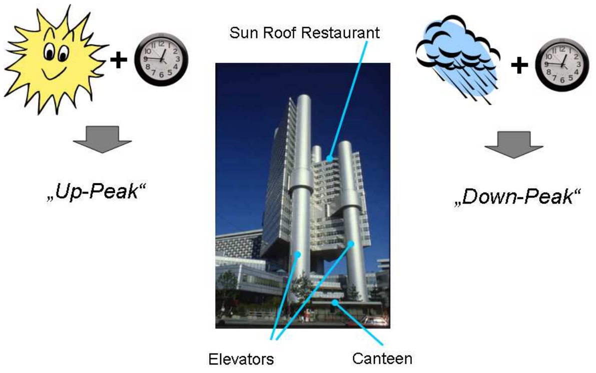

Fig. 3. Correlation between weather and passenger traffic pattern.

Examples of more dynamic context compared to the weather include the conclusions of large meetings or gatherings of groups of people in hallways. If the elevator control system has access to this knowledge, it can proactively switch to the respective algorithm which is more suited for the expected pattern. In particular, the control system is highly adaptive to changes in the context. One might reflect on the case that the rooftop restaurant begins offering lunch specials that overcome the concerns about bad weather. This modification is dynamically handled by the system and does obviously not require any system administrator intervention to reconfigure the elevator controller.

B. Example II - Location-Aware Scheduling Algorithm

We found out that the Zoning scheduling algorithm as introduced in section II-C performs better in some scenarios if the zones are assigned according to the destination floors of passengers. (Note that regular Zoning groups adjacent source floors for elevator journeys). We call this *Destination Zoning* or *Context Zoning* as the destination is a personal location and location-awareness is a key aspect of context-awareness.

A similar observation leaded to the *destination control system* invented by Schindler [Schi06], where each passenger enters the number of his destination floor in to a keypad located in every floor. Passengers are assigned to a specific elevator car through some audio/visual feedback system which serves only destination floors within its zone, therefore reducing the number of extra stops that increase the ride time especially in heavy traffic situations.

The problem of the usually unknown destination floor of a passenger may also be addressed in a more innovative way: This information may be gathered by another popular ubiquitous computing technology, namely Radio Frequency Identification (RFID). RFID has become an important and widespread technology, especially for logistics and access systems. Assuming RFID tags would be used in office buildings for the sake of identity proving, the destination floor of the passenger may be estimated by the elevator system as the floor of the office location of the identified passenger. Obviously this approach has the disadvantage that it is an *estimate* of the passenger's destination, which might be right or might be wrong. On the other side it has two advantages compared to the Schindler system: a) it acknowledges the destination of every single passenger and thus calculates with true capacities and b) it eliminates the interaction complexity with the passengers as it is performed automatically and invisible for the user.

C. Example III - Emergency Context

We found out that Zoning which performed best for down-peak demands could be improved for evacuation scenarios. During an evacuation which is caused by a reason which does not prevent to use any elevators (such as a fire), directly detected by some sensor (e.g. gas) or derived from some more complex context information, the primary objective is to rescue as many of the building population as possible. In a traditional down-peak pattern every call in the current movement direction is answered, even if the maximal capacity of the elevator car is already reached and no space for additional passengers is available. Hence, ride time will unnecessarily increase and newly arrived passengers will have to wait for the next car anyway as they are unable to board. The main difference between *Evacuation Zoning* and traditional Zoning is that the controller ignores all current and future yet not already answered calls if the maximal number of passengers has boarded and moves directly to the *recall floor*, usually the lobby, without any stops. After unloading has been performed, the elevator continues serving calls, beginning from the highest not yet served floor call below the last served one. See figure 4 for an example.

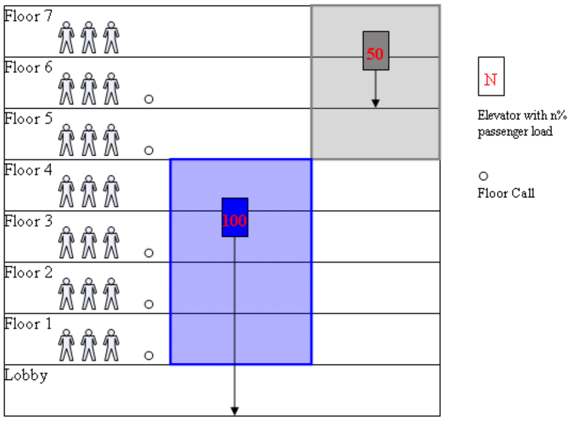

Fig. 4. Evacuation Zoning

IV. Simulations

A new simulation tool was implemented along the core elements of the elevator model introduced in section I-A which is outlined in figure 5. The tool offered various parameters to adjust the algorithms under different scenarios and contextual conditions.

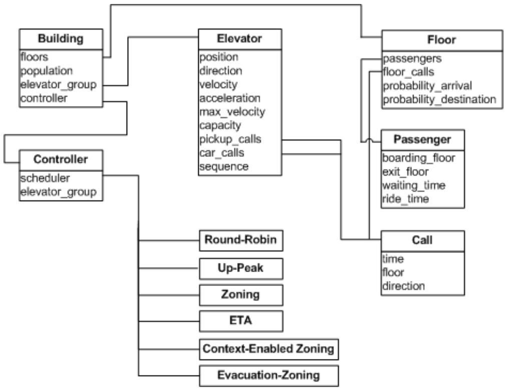

Fig. 5. Core classes of elevator model in our simulator

A. Setup and Parameters

Most of the parameters used in the setup of our simulation were derived from [Cibs05], [Schi06], including information on reference buildings. Besides trivial parameters such as the *number of floors* N, the *number of elevators* L and the *building population* U. The passenger arrival rate is modelled by a poisson distribution in a way that a given percentage (e.g. 6%, 9% or 12%) of the building population was generated in average in a 5 minute window.

Another adjustable main parameter is the *scheduling algorithm* - the four of them relevant for the next section were presented in section II.

Other changeable, mostly elevator-specific, characteristics are *acceleration* a, *rated speed* v, *single floor flight time* $t_f(1)$, *passenger capacity* with corresponding average number of passengers P, *average interfloor height* d_f, *passenger loading / unloading time* t_l and t_u, and finally *passenger delay time* t_{sd}. An explanation and tables with typical values are omitted here due to space restrictions but can be found in [Cibs05], which was also used to estimate the precise scenario parameters.

Passengers are modelled in the simulation as they behave in the real world, i.e. they may not be cooperative in the sense of the optimization task of the controller: They sometimes enter a car that has arrived without paying attention whether it is the called one and whether it is travelling in the desired direction or not.

B. Scenarios

Algorithms behave differently in buildings of various sizes, elevator models and configurations. Therefore several different scenarios covering specific situations were used, all having in common that they deal with office buildings and are based on existing buildings [Schi06]. The precise parameters for all test scenarios are defined in Table I. The average interfloor height d_f for all buildings was defined as 3m, passenger loading t_l and unloading time t_u as 1s. Building 3 consists of a small number of occupants which results in low number of passengers even for high demand, whereas building 1 offers a medium traffic scenario. Buildings 2 and 4 are very similar, as they house a large number of occupants and mainly only differ in the number of floors.

TABLE I
TEST SCENARIOS

Index	Building	N	U	L	$t_f(1)$	v	a	P
1	1180 Avenue	23	1978	8	5	3.0	1.00	7.6
2	Penn Avenue	9	2408	6	6	1.6	0.65	6.3
3	Corning	10	430	3	6	1.6	0.70	3.6
4	AAAS	13	2408	6	6	1.6	0.75	6.3

Passenger arrival rate was 12%, 9% and 6% of the building population, which results in three simulation runs per algorithm and traffic pattern. A single run simulated one hour of traffic and was performed 100 times to calculate an average value in order to eliminate statistical deviations, summing up in over 36.000 simulation runs.

V. Evaluation and Findings

As to be expected, our simulations showed that every scheduling algorithm performs well in some scenarios and less good in others. Unsurprisingly the scheduling algorithms which were designed for a specific demand (e.g. Up-Peak Group Scheduling for up-peak demand, see figure 6) performed best for this kind of demand. Likewise, ETA performed best for lunch-peak and Zoning performed best for down-peak demands.

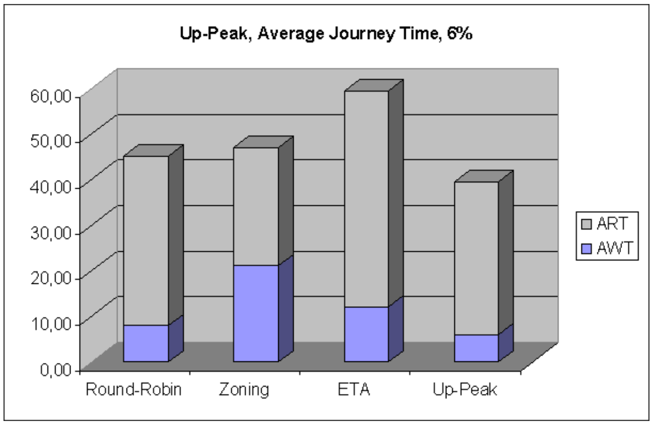

Fig. 6. AJTs of an up-peak scenario

However, the type of pattern - even if itself already not easy to detect and less easy to predict - is just one out of several parameters which determine whether the scheduling currently in use is optimal for the current demand, or if another algorithm would be better. This can be seen if you compare figure 6 with figure 7: Both show the average journey times for the same up-peak traffic pattern; they differ mainly in the passenger arrival rate. It is easy to see that Up-Peak scheduling performs best at the 6% level, but is almost worst if about twice as many passengers want to use the elevators of the building in the same time interval (12% level). Thus it is obvious that a good estimate of all the influencing parameters is essential to reach an optimal efficiency. This includes in particular any contextual knowledge such as the one illustrated in the weather example in section III.

Our simulations showed also that for high up-peak demands algorithms such as Zoning and Context-Zoning performed best, as one can also see in figure 7. However, traditional Zoning is strongly affected by a high amount of not served passengers, so that Context-Zoning delivered the best trade-off between a low AJT and a low number of passengers that has not been served.

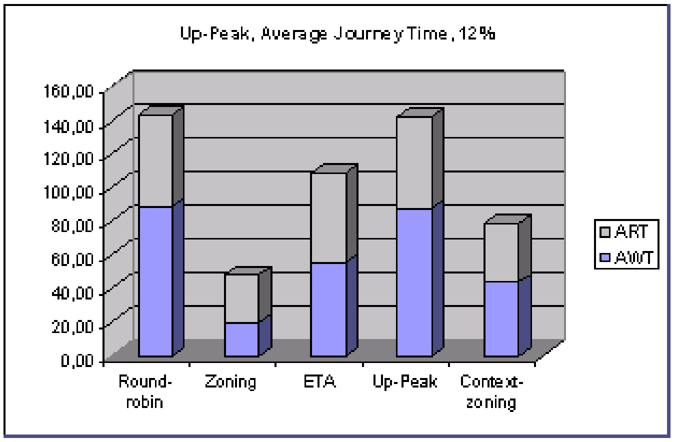

Fig. 7. AJTs for building 4 with 12% population

A similar argument holds for Context-Zoning in case of high down-peak demands, with evacuation as the most extreme scenario for a high down-peak demand. As shown in figure 8, Context-Zoning performed best concerning the AJT, but Evacuation-Zoning was almost as good as Context-Zoning concerning the AJT having about have as many not served passengers in average.

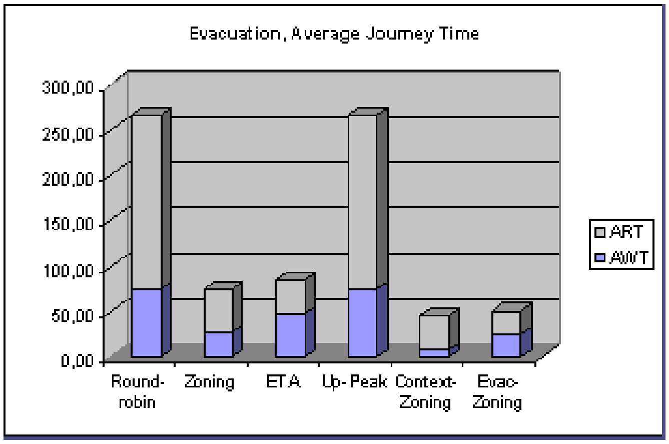

Fig. 8. AJTs for the evacuation scenario in building 1

VI. Summary and Conclusion

This paper applied ubiquitous technology, namely the concept of context-awareness, to an unusual example of a mobile system – an everyday transportation device that supports the mobility of people by its moving components. We introduced the application of context-awareness to elevator scheduling, which resulted in a novel elevator group scheduling algorithm and two novel elevator scheduling algorithms, namely Context-Zoning and Evacuation-Zoning. These new algorithms responded to its context by deploying scheduling strategies that optimize current behavior based on the present demands upon the system. This dynamic selection and adaptation in elevator scheduling on both the group and the single controller level contrasts with a classical generalized strategy that has the same behavior in all contexts.

The emphasis of much research in context-aware systems has been on the display and selection of information. In contrast, our unusual application demonstrates how in a well-known theoretic domain – that of scheduling – and an everyday problem – elevator service – context can be used as support technology towards ”intelligent buildings”. Similar context-aware computational strategies can certainly be implemented in other systems.

In the case of the work presented in this paper, we did not focus on a specific architecture for retrieving sensor information and mechanisms used to derive high-level knowledge in this paper. For the purpose of the simulation, the context information was directly accessible by the control system.

References

[CGK99] Cho, Y.C., Gagov, Z. and Kwon, W.H.: Timed Petri net based approach for elevator group controls. Technical report No. SNU-EE-TR-1999-3, School of Electrical Engineering, Seoul National University.

[CGK01] Cho, Y.C., Gagov, Z. and Kwon, W.H.: Improved concept for derivation of velocity profiles for elevator systems. IEEE International Conference on Robotics and Automation, Proceedings 2001 ICRA, vol. 3, p. 2419-2423.

[Cibs05] CIBSE Guide D: Transportation systems in buildings. The Chartered Institution of Building Services Engineers, 2005.

[ChSo97] Chan, W.I. and So, A.T.P.: Comprehensive Dynamic Zoning Algorithms. Elevator World, September, 1997, p. 99-103.

[Dey99] Dey, A.K. and Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. GVU Technical Report GIT-GVU-99-22.

[GrSt01] Grimmett, G.R. and Stirzaker, D.R.: Probability and random processes, 3rd edition, Oxford University Pr., 2001.

[KoOt02] Khler, Ottiger: An AI Based Approach to Destination Control in Elevators. AI Mag., vol 23, 2002

[NiBr03] Nikovski, D. and Brand, M.: Decision-Theoretic Group Elevator Scheduling. ICAPS’03, Trento, Italy, June 9-13, 2003

[RHL03] Rong A., Hakonen H. and Lahdelma R.: Estimated Time of Arrival (ETA) Based Elevator Group Control Algorithm with More Accurate Estimation. Technical Report 584, TUCS - Turku Centre for Computer Science, Turku, Finland, 2003.

[Schi06] Schindler Group: Miconic 10 Reference Installations. http://www6.schindler.com/SEC/websecen.nsf/pages/elev-MHR-Mic10-07, 22 May 2006.

[SeKo99] Seckinger, B. and Köhler, J.: Online Synthese von Aufzugssteuerungen als Planungsproblem. 13. Workshop ”Planen und Konfigurieren”, Technical Report, University of Würzburg, p. 127-134, 1999.

[Siik97] Siikonen, M.-L.: Elevator Group Control with Artificial Intelligence. Research Reports A67, Helsinki University of Technology, Systems Analysis Laboratory, October 1997.

[SHS05] Sorsa, J., Hakonen, H. and Siikonen, M.-L.: Elevator selection with destination control system. In Elevcon Asia, 2005.

[Stra04] Strang, T. and Linnhoff-Popien, C.: A Context Modelling Survey. Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England, September 2004.