APPENDIX F

A Controller for a Compact Dynamic Bus Station

F.1 Case Description

A bus station consists of a number of platforms at which buses can stop. Buses run along circular routes that start and end at a platform. Each such route is called a line. In a bus station with static bus allocation, each platform has a number of lines allocated to it so that a bus that runs the route of this line invariably stops at this platform. At any point in time, most platforms are empty because at any point in time, most buses are out, driving a route. This is a waste of space. To save space in densely populated areas, compact dynamic bus stations have been invented.

A compact dynamic bus station consists of a small number of platforms to which buses are allocated when they approach the bus station. The allocation is made when a bus approaches the bus station, so that optimal use of platforms can be made. This appendix describes the design of a Bus Allocation System (BAS) for a compact dynamic bus station. We start by giving more information about compact dynamic bus stations and the requirements for BAS.

A compact dynamic bus station consists of a number of platforms, sensors, and information screens. BAS has access to sensors in the approaching roads that can sense whether a bus is approaching. Sensors at the entries and exits of the station can tell BAS whether buses are entering the bus station or are departing from it, and sensors at the platforms can tell BAS whether buses are stationary at a platform. One platform is reserved for buses that, for some reason, could not stop at their intended platform. This platform is called the buffer. Each bus contains a screen on which messages to the driver can be displayed. This is called a driver instruction screen. BAS has a wireless connection to this screen. The bus station has several passenger information screens on which information for the passengers can be displayed. BAS is connected to these screens too.

The routes driven by buses are partitioned into lines. Each line has exactly one route, which starts and ends at the bus station. When a bus makes a trip, it follows the route of exactly one line. Each day, one or more buses can make several trips of one line. For each line there is a preferred platform, which is reserved for that line during particular time slots during a day. BAS should allocate an incoming bus as much as possible to this platform, that is, to the platform preferred for the trip that the bus is now making. BAS has access to a linear programming package called CPLEX that can compute optimal allocations. When a bus enters the station, BAS should instruct the driver to drive to the platform allocated to the trip and it should inform the passengers of this allocation.

When a bus is out making a trip, it is expected to arrive at a certain time. If it is delayed, BAS must announce this to the passengers. When the bus is approaching, BAS must announce this too and the driver should be told to which platform the bus is dynamically allocated. If upon entry that platform is nevertheless full, the bus has to wait at the buffer and go to the platform when it becomes free.

This case is based on a formal specification of a bus allocation system done by Bas van Vlijmen [1, 2].

F.2 Mission and Functions

Figure F.1 shows the mission statement of BAS and Figure F.2 the function refinement tree. BAS is a controller because it has a directive functionality; it tries to make people act in a certain way. It does this by coordinating the allocation of buses to platforms (done by CPLEX) with the events in the life of a bus trip and with the messages to be displayed on the information screens. The name and acronym chosen in Figure F.1 have a more prosaic motivation: The person who first designed the system is called Bas.

Name: Bus Allocation System

Acronym: BAS

Purpose: Coordinate the allocation of buses to platforms with the movement of the buses and the announcements to the drivers and passengers.

Responsibilities:

- Monitor the state of a bus trip.
- Periodically recompute the optimal allocation of buses to platforms.
- Announce allocations to drivers in approaching buses and to passengers waiting at platforms.
- Allow editing of schedules, timetables etc.

Exclusions:

- The system does not handle reservation of platforms for buses; rather, given a set of preferences, it will dynamically compute an allocation of buses to platforms that is optimal with respect to these preferences.
- The system does not deal with linked trips (i.e., those for which passengers of one trip must be able to transfer to another trip).

Figure F.1 Mission statement of BAS.

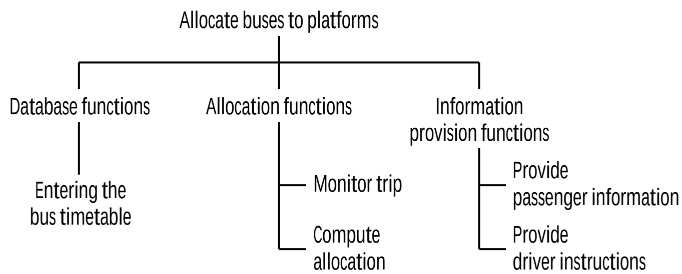

Figure F.2 Function refinement tree.

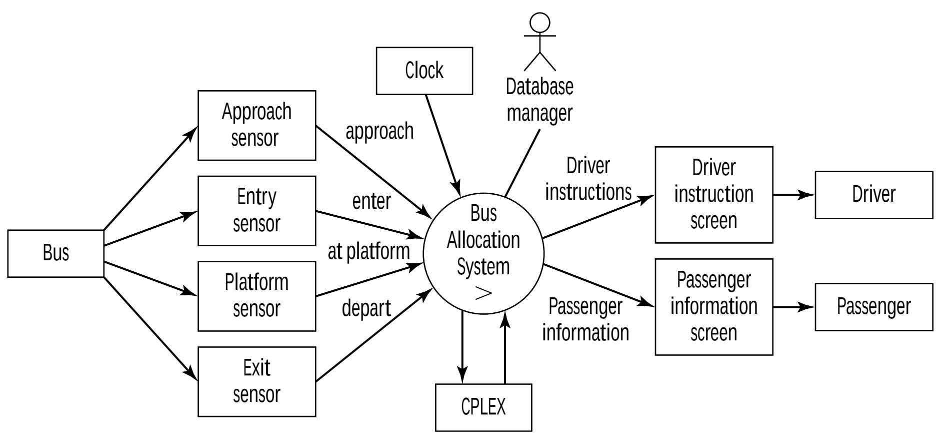

Figure F.3 BAS context diagram.

F.3 Context

Figure F.3 shows the context diagram of BAS. It shows that BAS primarily coordinates the CPLEX package with what happens at the sensors and with what should happen on the screens.

F.4 Subject Domain

Figure F.4 shows the subject domain ERD and Figure F.5 shows some dictionary entries. Only a few dictionary entries are listed, namely those that need precise definition.

Figure F.6 represents the behavior of a trip. The attribute status of a trip represents the state in this diagram. It is stated in terms of events that can be observed by the system by means of its sensors. The STD represents normal behavior, not actual behavior, because it is very possible that a bus that starts a trip will not take a path

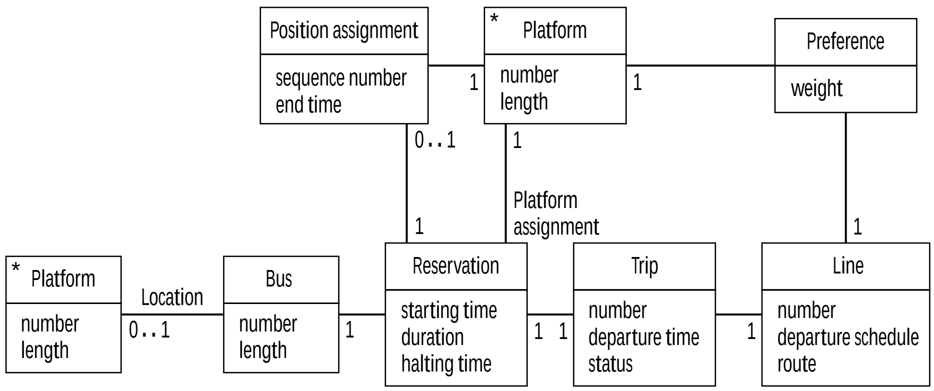

Figure F.4 ERD of the BAS subject domain.

Departure interval For each reservation, this is the interval
[starting time + halting time, starting time + halting time + 2 minutes].

Sequence number Attribute of Position assignments. The order of parking of buses at one platform. An assignment with a lower sequence number indicates a location in front of assignments with higher sequence numbers.

Starting time interval For each reservation, this is the interval
[starting time, starting time + 2 minutes].

Weight The weight of a preference is a natural number that represents the preference for (any position at) a platform. The number 0 indicates highest preference. Higher natural numbers indicate increasingly lower preferences.

Figure F.5 Definitions.

through this diagram: It may get stuck, it may run in circles and loop over one particular sensor any number of times, the driver can ignore the instructions, and so on. The diagram, therefore, describes the behavior of a trip as it should ideally take place. The diagram is nevertheless useful because it defines the meaning of the states of a trip.

Figure F.7 lists a number of subject domain properties in column 1. Column 2 indicates whether a property is always true or whether it may be violated sometimes by the subject domain. A property is always true when it is a law of nature or when it is an analytical truth that follows from the meanings of the words used in it. The properties in Figure F.7 that may be violated are aspects of the desired situation in the subject domain. When a desired property can be violated, we may decide that the system must register it or must not register it. This is a system requirement, listed in column 3. Finally, if the system must register it, then we may decide whether the system must do something about this violation. This is, again, a system requirement, listed in column 4. (Columns 3 and 4 are therefore not part of

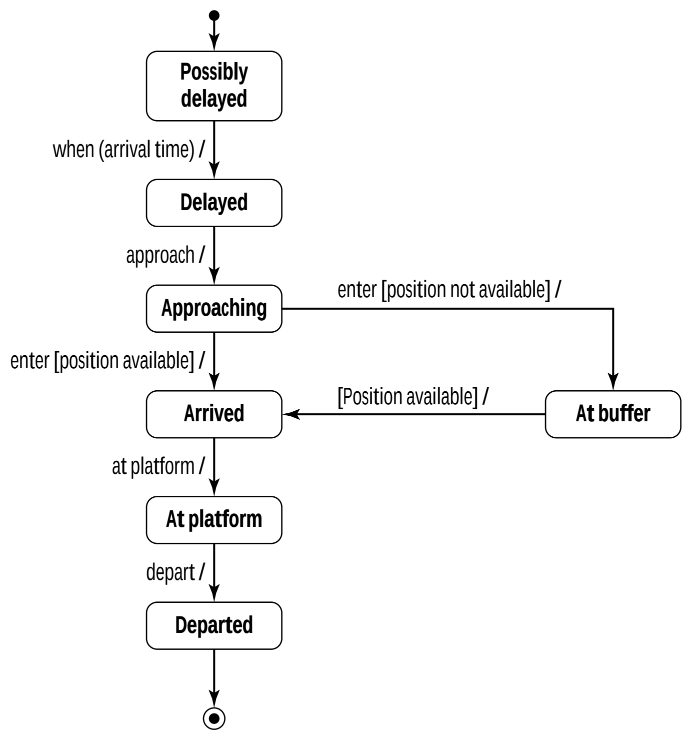

Figure F.6 Mealy diagram of the status of a trip.

the subject domain model but of the system requirements. They are listed here for convenience.)

F.5 Desired Behavior of BAS

Figure F.8 lists the relevant subject domain events to which BAS must respond, the stimulus by which BAS determines that this event has occurred, and the desired response. For traceability, the functions as part of which this event-response sequence can occur are listed too. The figure does not list the current and next states of BAS. These are shown in the Mealy diagram in Figure F.9.

The Mealy diagram in Figure F.9 shows that, for each trip, BAS must represent the states of the trip. The stimuli received by BAS trigger state transitions that mirror those of the ideal trip shown in Figure F.6 and that produce the responses prescribed by the desired functions of BAS (Figures F.2 and F.8).

This behavior specification makes a number of assumptions about the subject domain and the connection of BAS with the subject domain, listed in Figure F.10.

Property	Always true in subject domain	Violation must be registered by BAS	BAS must take corrective action when violation is registered
P1. The total length of buses at platform never exceeds the length of the platform.	No	Yes	No
P2. The total length of buses at platform never exceeds the length of the platform plus the maximum length of a bus.	Yes		
P3. For each reservation, the starting time interval is disjoint from the departure interval.	No	No	
P4. For different reservations allocated to one platform, their departure and arrival intervals must be disjoint.	No	No	
P5. For any pair of reservations allocated to the same platform, the ordering of their departure intervals equals the ordering of their arrival intervals.	No	No	
P6. The departure time of a trip is one of the departure times of the line of the trip.	Yes		
P7. The halting time of a reservation does not exceed the reservation duration.	No	No	
P8. The end time of a position assignment lies between the starting time of its reservation and the end time (= starting time + duration) of its duration.	No	No	
P9. For every line-platform combination there is at least one preference.	No	Yes	Yes
P10. For every line, there is at least one preference with weight 0.	No	Yes	Yes
P11. Initially, the platform assignment assigns a trip to the preferred platform for the line of this trip.	No	No	
P12. A bus cannot be at a buffer and a platform at the same time.	Yes		
P13. A reservation can only have a position assignment to a position at the platform to which the reservation is assigned.	No	No	

Figure F.7 Actual domain properties, desired domain properties, and requirements of BAS.

Some of these assumptions are violated very easily by the environment of the system. Assumption A1, for example, is too restrictive for a system that relies on it to be useful. A bus may pass a sensor when it is brought from or returned to its garage, when it is towed by another bus, when it is pushed by its passengers to its destination, when it is hijacked, or when it is otherwise occupied in a way that differs from the ideal trip behavior and that makes it meaningless for the system to direct it to a platform of the bus station. Only buses in working order traveling on a scheduled trip should be directed to platforms.

Event	Stimulus	Desired response	Function
Bus approaches	approach	• Update passenger information board	• Monitor trip • Provide passenger information
Bus enters bus station	enter	• Update passenger information board • Instruct driver to drive to platform or buffer	• Monitor trip • Provide passenger information • Provide driver instructions
Bus arrives at platform	at platform	• Clear driver instructions	• Monitor trip • Provide driver instructions
Bus departs from bus station	depart	• Clear passenger information board • Recompute position assignment	• Monitor trip • Provide passenger information
Scheduled arrival is 15 minutes from now		• Update trip status to reserved	• Monitor trip
Scheduled arrival time is now		• Trip status becomes delayed	• Monitor trip • Provide passenger information
One minute has passed since last computation		• Compute optimal allocation	• Compute allocation
Decision to update database	update request	• Update the database	• All database functions

Figure F.8 Events to which the system must react and their desired responses.

On the other hand, concerning assumption A2, we cannot require that a sensor be able to sense the trip that a bus is making, so the system must indeed perform some event recognition activity to figure out which trip a bus is taking. When a bus passes a sensor, the system should use its database to check whether this bus currently is out on a particular trip or not. Only when it is can we make the assumption that the bus is in normal working order and can BAS direct the bus to the platform allocated to the trip.

F.6 Requirements-Level Architecture

Figure F.11 shows a requirements-level decomposition of BAS. It uses subject-domain-oriented decomposition because subject domain entities are represented by system components and functional decomposition because functions are represented by components, too. Most subject domain entities are represented in databases. One subject domain entity, a trip, is represented by an object, which represents the current state of the trip. The Mealy diagram of desired system responses shown in Figure F.9 is the behavior of instances of the Trip monitor class. The three functions are represented by data transformations.

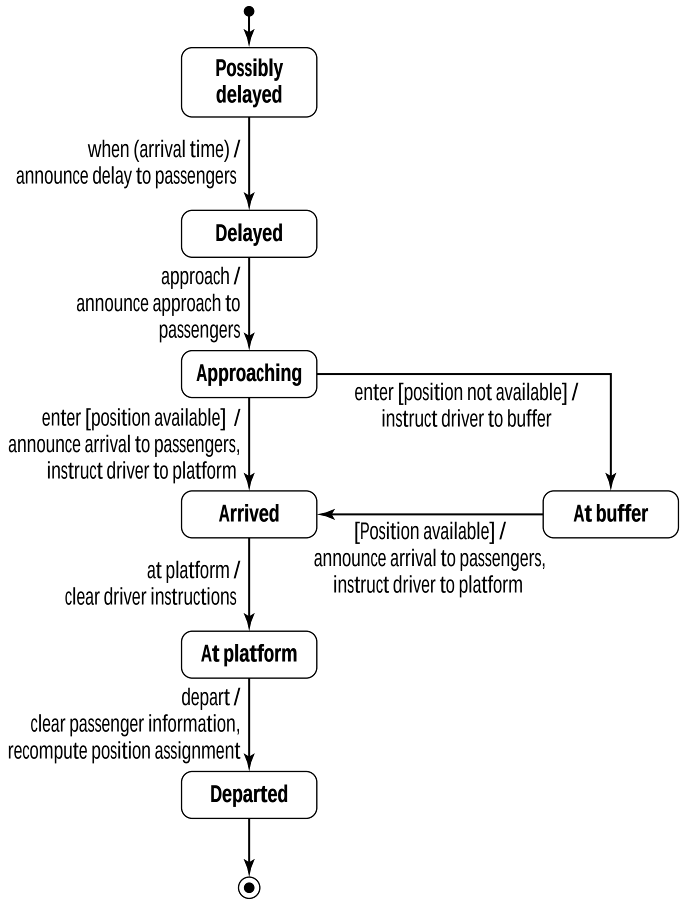

Figure F.9 Mealy diagram of the desired response of the system to changes in trip status.

- **A1** Bus trips behave in the ideal manner shown in Figure F.6.
- **A2** Each entry sensor can sense which particular bus passes it.
- **A3** Drivers always follow the instructions of the system. For example, they never stop at another platform than the one they are allocated to.

Figure F.10 Assumptions made about the subject domain.

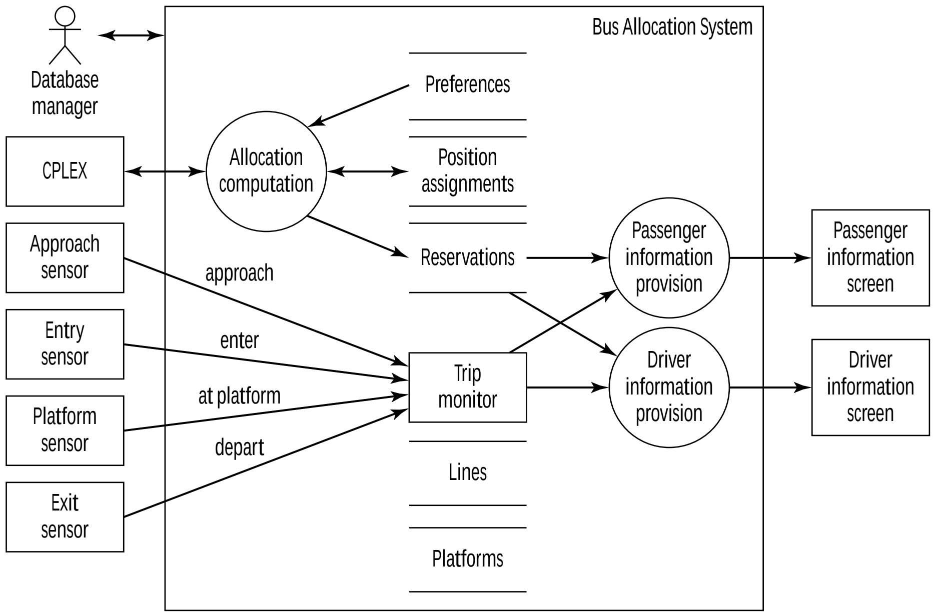

Figure F.11 Communication diagram of a requirements-level architecture.

Note that the database manager has access to all the databases. In order not to clutter up the diagram, functions for the database manager are not represented. Cardinality properties of the databases and the Trip Monitor class are taken from the subject domain ERD.

References

1. van Vlijmen, S. F. M. (1998). Algebraic Specification in Action. PhD diss. University of Utrecht.
2. Wieringa, R. J. (1999). *Using the Tools in TRADE III: A Controller for a Compact Dynamic Bus Station*. Technical report, CTIT Technical Report Series.