Elevator Management Class Model Descriptions

Author

Leon Starr

Document

mint.ev3.td.8 / Version 3.1.5 / November 5, 2017

Document organization

This document provide a description for each element of the class model. This includes classes, attributes, identifiers, relationship multiplicity and conditionality and constraints. Data types are described in a separate document since they span multiple subsystems.

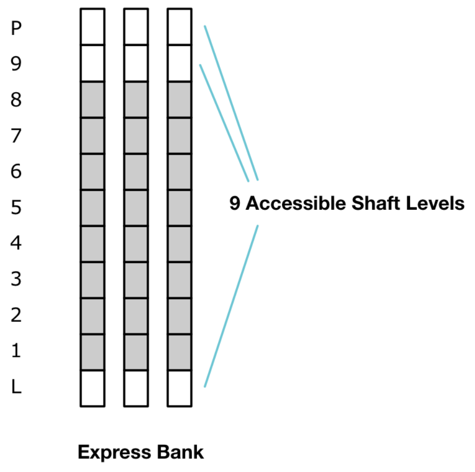
This information is ordered alphabetically first by class and then by relationship name (R-number). Whereas technical notes provide a functional or narrative driven explanation intended for sequential reading, class model descriptions index localized model element explanations and so are organized dictionary style for fast reference and ease of relocation as the model changes.

License / Copyright

Class Descriptions

Accessible Shaft Level

In a given Shaft, all Floors designated for service within the Shaft's Bank are considered accessible. In other words, it is possible for a Cabin to be directed to that Floor during normal usage. This means that for every Shaft in the same Bank, the same set of Floors is accessible. If a Bank provides service to all Floors, then, for each Shaft in the Bank, each of its Shaft Levels is an Accessible Shaft Level. If a Bank provides service to only the topmost and bottom Floors, then each Shaft in the Bank will have only two Accessible Shaft Levels.



If, for example, the **Express Bank** services only the **L – Lobby**, **9** and **P – Penthouse** Floors, only those associated Shaft Levels will be accessible for each Shaft in that Bank. Presumably, the excluded Floors are serviced by one or more Shafts in some other Bank.

Attributes

Floor

An Accessible Shaft Level is a Shaft Level made accessible by a Bank Level, so the Floors of each must match.

Shaft Level.Floor and **Bank Level.Floor**

Shaft

Shaft Level.Shaft

Bank

Bank Level.Bank

Stop requested

This setting corresponds to the concept of a button press inside of a Cabin requesting a stop at a particular Floor. So if set to **true**, the Cabin wants to stop at this level.

Type: **Boolean**

Identifiers

Floor + **Shaft**

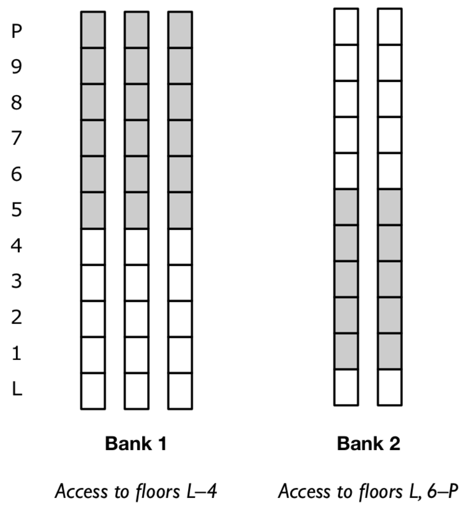
This is the standard identifier composition for a one to many association with the identifier attributes taken from the many side.

Bank

If you walk into the lobby of a tall building (more than 30 floors), you typically see the Shafts organized in service groups. You might see three Shafts going to the lower Floors only, while three other Shafts provide express service to the highest Floors. You might even see one or two Shafts dedicated to the parking garage.

A Bank is a group of Shafts and Cabins configured to provide identical service. Each Cabin in a Bank goes to the same Floors, moves with the same velocity / acceleration profile and operates Doors with the same open duration.

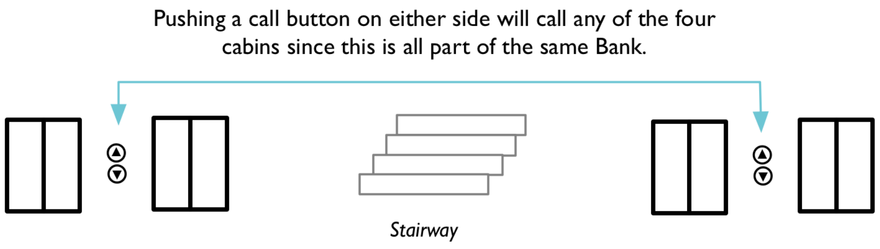
Consider an example where you organize five Shafts by defining two Banks.



Bank **B1** will serve the lower Floors while **B2** serves the higher Floors.

All Shafts in a Bank respond to the same floor up / down calls. If someone were to press the up button near Shaft **S1** in **B1** he or she would expect a lower floor servicing Cabin to arrive in response. While it may be a good idea to define Banks with Shafts that are physically near each other, Banks may be configured with any collection of one or more Shafts.

The Shafts in the next example belong to the same Bank since pushing the up or down call button on either side of the stairway may call a Cabin in any of the four Shafts:


Pushing a call button on either side will call any of the four cabins since this is all part of the same Bank.
Stairway

If there is another Bank nearby, it will ignore the request since each Bank services its own up / down floor call buttons.

Attributes

Name

Banks are typically named according to the intended purpose. A Bank titled *Tower* might provide express service to higher floors by skipping lower floors. A Bank titled *Freight* might be configured so that the Door stays open for a longer period and can access all Floors.

Type: **Name**

Passenger load time

The amount of time to hold doors open after arriving at a Floor before automatically closing again.

Type: **Duration**

Block clear time

The amount of time to wait after a **Door** is obstructed before attempting to close it again. This will probably be shorter than the **Passenger load time**.

Type: **Duration**

Max close attempts

The maximum number of consecutive attempts permitted to close an obstructed **Door**. If this value is exceeded, the Shaft is taken out of service.

Type: **Count**

Average cabin speed

We use this number to estimate a Cabin's arrival time at a calling destination. Since we are using it to roughly score Cabin choices during a floor call selection, this number need not be precise. In fact, the number can be fudged a bit to change the way that the bank select algorithm weights distance from a calling floor as a selection factor.

Type: **Speed**

Average stop duration

The average time that a **Cabin** spends resting at a **Floor** when a destination beyond that **Floor** in either direction is pending. In other words, we are measuring from the point in time that a **Cabin** arrives, to the point it starts moving again assuming that a destination is immediately pending.

This duration is considered during the bank select algorithm to estimate how long it will take for a **Cabin** to arrive at a calling floor given that there are one or more possible stops along the way.

Type: **Duration**

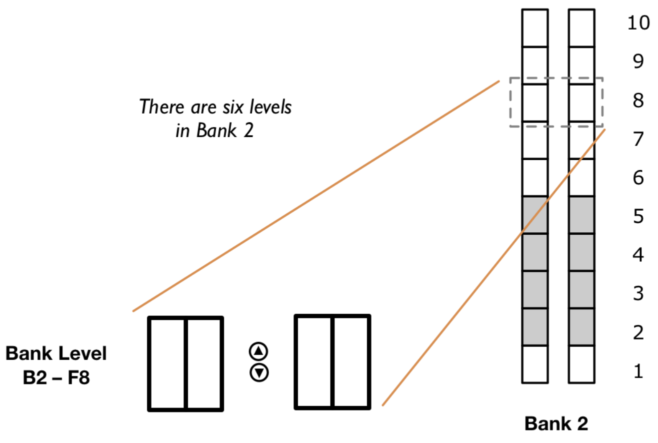
Identifiers

1. **Name**

Banks are named uniquely by policy.

Bank Level

A passenger at a Bank, **B2** let's say, presses the up or down call button to request the arrival of a Cabin at the current Floor. It doesn't matter which Cabin arrives, as long as the Cabin is going in the requested direction and belongs to the Bank. Thus, an up/down button pair is associated with a Floor-Bank intersection which we call a Bank Level.



When a Bank is defined, two or more adjacent or non-adjacent Floors are included as part of the Bank service. These Floors can be serviced by any of the Shafts included in the Bank. In the example above, Floors **1, 6..10** are included in the **B2** Bank. This establishes six instances of Bank Level: **B2-1, B2-6 .. B2-10**.

The important thing is this: An up/down call is specific to a single Bank Level.

Attributes

Bank

Bank.Name

Floor

Floor.Name

Identifiers

1. **Bank** + **Floor**

Standard identifier composition for a many to many association with the identifier attributes taken from both sides.

Bottom Bank Level

This is the lowest Floor serviced by a Bank. From here a passenger can request only up service.

Attributes

Bank

Bank Level.Bank and **Bank.Name**

A Bottom Bank Level is a subclass of Bank Level and a Bank must specify exactly one lowest level.

Floor

Bank Level.Floor

Calling up

Corresponds to the idea of an up arrow button at some Floor. If set, it represents a request for a Cabin to arrive providing upward service.

Type: **Boolean**

Identifiers

1. **Bank** + **Floor**

This is the standard use of a reference to the superclass identifier.

Building

A physical structure serviced by a mechanical, software controlled lifting system such as a hotel, skyscraper, parking garage or even a really big house.

Attributes

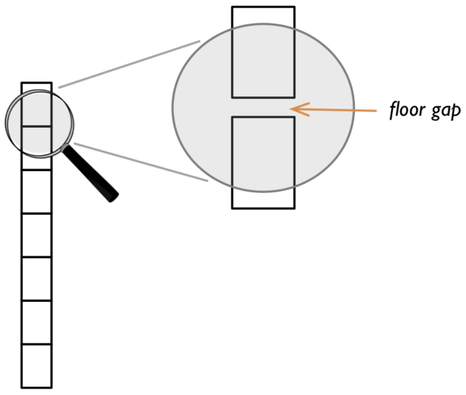
Name

Any kind of name is okay here. **My Really Huge Building**, for example. Since there is only one instance in this release, we could have used a Nominal type, but the door is open to a future release where there are multiple instances.

Type: **Name**

Average floor gap

This is the average vertical distance between two adjacent Floors.



This distance is used by the bank select algorithm when it is gauging distances up and down the Shaft to estimate travel time. This is a derived attribute in theory, but since it never changes and need not be precise, we just enter a reasonable value during initialization.

Type: **Distance**

Identifiers

1. **Name**

It is assumed, in this release anyway, that there is only ever one instance of Building, so it must be unique.

Cabin

The compartment that conveys passengers up and down a Shaft is known as a Cabin. It is sometimes (in the United States, at least) informally referred to as an "elevator". Our British friends call it a "lift". Since either term may also refer to the entire, er... lifting system, we use Cabin as the formal term for the enclosed box of people that goes up and down.

Attributes

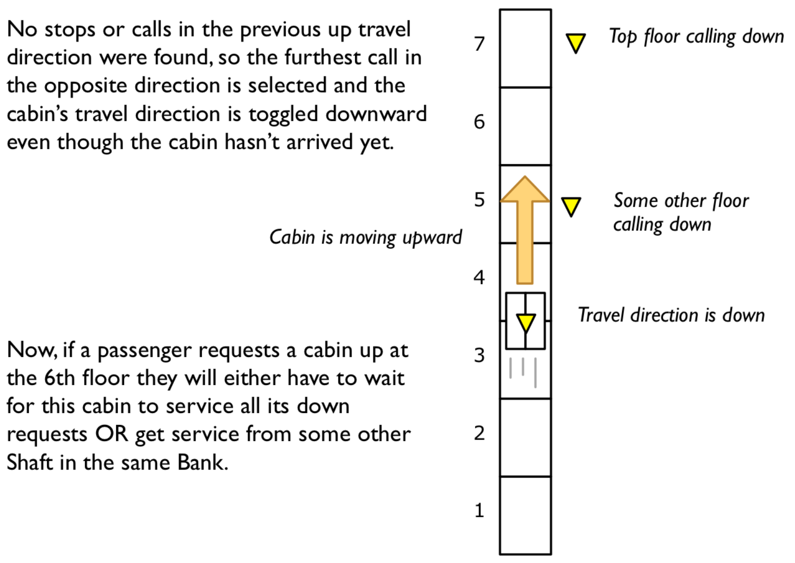
Shaft

Shaft.ID

Travel direction

This is the *intended* travel direction of the Cabin which biases selection of the next destination Floor. It is sometimes, but not always the current direction of Cabin movement.

The elevator dispatching algorithm tries to keep each Cabin going in the same direction until there are no more Floors to service in that direction. The Cabin then reverses its **Travel direction**.


No stops or calls in the previous up travel direction were found, so the furthest call in the opposite direction is selected and the cabin's travel direction is toggled downward even though the cabin hasn't arrived yet.
Cabin is moving upward
Now, if a passenger requests a cabin up at the 6th floor they will either have to wait for this cabin to service all its down requests OR get service from some other Shaft in the same Bank.
7
6
5
4
3
2
1
Top floor calling down
Some other floor calling down
Travel direction is down

In the above example, the Cabin starts off with an *up* **Travel direction**. But a search upward finds only calls in the opposite direction, so the furthest is selected and the **Travel direction** is toggled downward, even though the Cabin will be moving upward. The Cabin has effectively decided to ignore upward calls until all downward calls are serviced. So we can see that the **Travel direction** truly represents a Cabin's decision of which direction it wants to travel.

Type: **Direction**

Current floor

Shaft Level.Floor

Identifiers

1. **Shaft**

The concept of a single Cabin in a Shaft is unlikely to change. So, for convenience, the Shaft ID is appropriated as the Cabin ID. (It's easier to keep track of a population if you can refer to the **S2** Door, Cabin and Shaft, for example).

Door

Each Cabin has a pair of door panels under software control which normally separate when the Cabin is at rest on a Floor. Each Shaft Level has its own set of passive outer panels. During separation, the Cabin door panels mechanically engage the corresponding Shaft Level outer panels. We refer to the interior pair of software controlled Cabin door panels as "the Door".

During its lifecycle, a Door opens, closes and, if it cannot close after repeated attempts, indicates blockage and shuts down the Shaft for service.

Attributes

Shaft

Cabin.Shaft

Close attempts

The number of consecutive times the Door has attempted to close without success.

Type: **Count**

Lock requested

This soft lock guards against a potential race condition where the Door is triggered to open just as the Cabin begins to move. When true, the Door moves into a state where passenger requests to open the Door are ignored, thus the Cabin may move. See the Door/Cabin state models for more details.

Type: **Boolean**

Held

A passenger may want to hold the Door open for an indefinite duration. If the Doors are ready to close and **Held** is set, the Door will remain open until **Held** becomes false.

Type: **Boolean**

Blocked

This is set when the Door tries to close but encounters some obstacle. This value remains true until the Door successfully closes.

Type: **Boolean**

Emergency hold

During an emergency such as a fire or earthquake, or during repair, the Emergency hold status will prevent the Door from closing under any circumstances.

Type: **Boolean**

Identifiers

1. **Shaft**

The Door is identified by its associated Shaft. This works since there is only one Door per Cabin.

Floor

A Floor is a horizontal level intersecting at least one Shaft in a Building. It must be physically possible for a Cabin to arrive at a Shaft-Floor intersection to transfer passengers. While Bank policies may preclude a Cabin's access to a Floor, arrival is still *physically* possible. So a Floor is not some level where Cabin access is physically impossible, reachable only by stairs, for example.

Attributes

Name

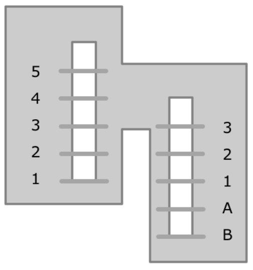
A letter, number or short label (**P1, L, MEZZ, 1, 2, 3** for example) typically appearing on buttons inside the Cabin.

Type: **Level Name**

Height

The distance of the Floor from the bottom of the Building. This is measured so that each Floor has a positive value, even if it is below ground. Ih the following scenario it may appear that two floors can be at the same height. By policy, however, each stretch will have the same name if they are at the same height.

Oddly shaped building



Here is a Building with an overlapping parking garage. Note that apparently distinct floors at the same height have the same name. This means that it is not possible, within a Building, to have two Floors at the same height, as they would, by policy be the same Floor, even if there is a physical separation.

Type: **Distance**

Identifiers

1. **Name**
2. **Height**

I1

Names must be unique to keep the passengers from getting confused about what floor they are actually on.

I2

By definition, a given height where a Cabin can stop represents a Floor. Consequently, within the same Building, two stretches of Floor at the same height constitutes a single Floor with the same name.

Floor Service

A passenger at a Floor may request a Cabin going up or down. No particular Cabin can be chosen by the passenger, so an appropriate Cabin within the local Bank will be selected using an algorithm. The algorithm should pick a Cabin likely to arrive as soon as possible going in the requested direction. Once the selection is made by this algorithm, a Floor Service represents the choice of an Accessible Shaft Level at the calling Floor within the local Bank.

In other words, this is an Accessible Shaft Level designated to handle a call from a Floor.

Attributes

Bank

Accessible Shaft Level.Bank and **Bank Level.Bank –** Both are part of the same Bank.

Floor

Accessible Shaft Level.Floor and **Bank Level.Floor –** Both are at the same Floor.

Shaft

Accessible Shaft Level.Shaft

Direction

The passenger has requested a Cabin moving in this direction.

Type: **Direction**

Identifiers

1. **Bank** + **Floor** + **Shaft**

Standard formulation from referential attributes on a many-many association.

Middle Bank Level

This is a Floor between the highest and lowest serviced by a Bank. From here a passenger can request either up or down service.

Attributes

Bank

Bank Level.Bank

Floor

Bank Level.Floor

Calling up

Corresponds to the idea of an up arrow button at some Floor. If set, it represents a request for a Cabin to arrive going up.

Type: **Boolean**

Calling down

Corresponds to the idea of an up arrow button at some Floor. If set, it represents a request for a Cabin to arrive going down.

Type: **Boolean**

Identifiers

1. **Bank** + **Floor**

This is an identifier since it is a subclass reference to a superclass identifier.

Shaft

A Shaft is a vertical conduit through which a Cabin traverses. A system of motors and cables or some other mechanical system installed in each Shaft makes this motion possible.

Attributes

ID

Each Shaft is numbered uniquely.

Type: **Nominal**

Bank

Bank.Name

In service

When **true**, this Shaft is available for passenger transport. Otherwise, the Shaft is out of service until maintenance is complete.

Type: **Boolean**

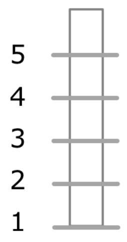
Identifiers

1. **ID**

Unique by policy

Shaft Level

A Shaft physically intersects one or more Floors. At each Floor-Shaft intersection there is an opening where a Cabin may come to rest, open its Door and transfer passengers. This intersection is called a Shaft Level.



Shaft with five Shaft Levels on Floors 1–5

Attributes

Floor

Floor.Name

Shaft

Shaft.ID

Identifiers

1. **Floor** + **Shaft**

Standard composition of an identifier for a many to many with combined references from each side.

Top Bank Level

This is the highest Floor serviced by a Bank. From here a passenger can request only up service.

Attributes

Bank

Bank Level.Bank and **Bank.Name**

A Top Bank Level is a subclass of Bank Level and a Bank must specify exactly one highest level.

Floor

Bank Level.Floor

Calling down

Corresponds to the idea of a down arrow button at some floor. If set, it represents a request for a Cabin to arrive going down.

Type: **Boolean**

Identifiers

1. **Bank** + **Floor**

This is an identifier since it is a subclass reference to a superclass identifier.

Transfer

A Cabin can only go one place at a time. A Transfer represents the current destination, if any, of a Cabin. In transit, new stop requests and floor calls may result in the choice of a more optimal destination. For example, the cabin might be moving from the 2nd to the 10th Floor when a passenger at the 7th floor pushes the up button. If this new destination is selected by the routing algorithm, the Transfer will coordinate with the Transport service to accommodate the reroute. If successful, the Transfer will update the current destination, even if the Cabin is in motion.

Attributes

Destination floor

Accessible Shaft Level.Floor

Shaft

Accessible Shaft Level.Shaft and **Cabin.Shaft –** Can't go to a Shaft Level in a different Shaft!

Identifiers

1. **Destination floor**
2. **Shaft**

Standard formation of identifiers in a one to one association class where the reference from each side constitutes an identifier.

Relationship Descriptions

R1

Bank defines service features of *at least one* **Shaft**

Shaft has service features defined by *one* **Bank**

All Shafts in a Bank provide the same type of service. This includes access to the same Floors and consistent Door behavior. If you want a certain Shaft to skip **Floor 3**, let's say, then you would establish a Bank that excludes that Floor and then assign the Shaft to that Bank.

A Bank has no utility if it does not define policy for at least one Shaft.

Since we don't want two conflicting sets of policies defined for a Shaft, the Shaft must belong to exactly one Bank.

Formalization

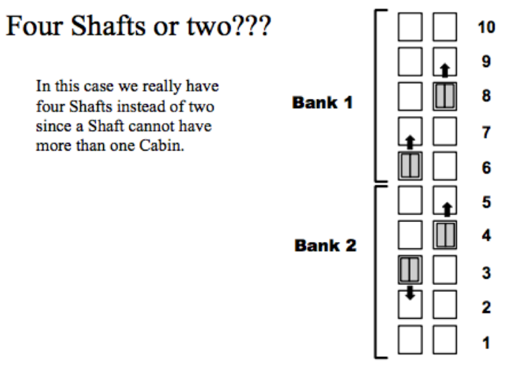
Shaft.Bank

R2

Cabin travels through *one* **Shaft**

Shaft is conduit for *one* **Cabin**

A Cabin moves up and down its Shaft to deliver passengers from one Floor to the next. But imagine the following case where we split a building vertically into two Banks covering two Shafts.


Four Shafts or two???
In this case we really have four Shafts instead of two since a Shaft cannot have more than one Cabin.
Bank 1
Bank 2
10
9
8
7
6
5
4
3
2
1

It's tempting to consider the possibility of a Shaft containing more than one Cabin. A simpler formulation, however, is to say that we have four Shafts two upper Shafts stacked on two lower Shafts. So by definition, a Shaft really does enclose the travel path of only one Cabin.

The case where it might be useful to think of a multi-Cabin Shaft in some future release would be a situation where two Cabins actually service the same set of Floors, but there is no requirement in this release to address the complexity of such an arrangement.

An empty Shaft is of no interest or utility in our system so a Shaft is defined to contain exactly one Cabin.

Formalization

Cabin.Shaft

R3

Bank Level enables access to *at least one* **Shaft Level**

Shaft Level is accessible in *zero or one* **Bank Level**

A Shaft belongs to a single Bank and all of its Shaft Levels belongs to that same Bank. But only some of those Shaft Levels will be accessible with the rest of them being skipped. The skipped Floors are presumably serviced by a different Bank.

So, for a given Shaft Level it either is or isn't accessible in the current Bank configuration. Furthermore, all Shaft Levels at the same Floor in the same Bank are either accessible or skipped.

Formalization

Accessible Shaft Level(**Floor** + **Shaft**)

R4

Door is passenger entry for *one* **Cabin**

Cabin passengers enter via *one* **Door**

Every Cabin is built with one Door. Cabins with more than one opening (front and back) are not considered in this release. A Door is built into exactly one Cabin.

Regarding inner vs. outer doors, see the Door class description.

Formalization

Door.Shaft

R5

Top Bank Level is the highest floor serviceable by *one* **Bank**

Bank provides service up to *one* **Top Bank Level**

A Bank is constrained such that it must have a highest Floor where it provides service. By definition, a Top Bank Level must be that Floor for its Bank.

Formalization

Top Bank Level(**Bank**)

R6

Bottom Bank Level is the lowest floor serviceable by *one* **Bank**

Bank provides service down to *one* **Bottom Bank Level**

A Bank is constrained such that it must have a lowest Floor where it provides service. By definition, a Bottom Bank Level must be that Floor for its Bank.

Formalization

Top Bank Level(**Bank**)

R28

Shaft intersects and opens onto *at least one* **Floor**

Floor is intersected and accessed by *at least one* **Shaft**

A Shaft must intersect and open onto at least two Floors to be useful. The two-ness constraint is ignored here since it is handled by the definition of Bank Levels. If a level cannot be reached by any Shafts, then it is not a Floor as far as the elevator system is concerned (see Floor class description). So, by definition, every Floor can be reached by at least one Shaft. Multiple Shafts may open onto the same Floor.

While one or more Shafts may physically intersect a Floor, that Floor may still be unreachable if none of the enclosing Banks service that Floor as a Bank Level.

Formalization

Shaft Level(**Floor** + **Shaft**)

R29

Bank services *at least one* **Floor**

Floor is serviced by *zero, one or many* **Bank**

When a Bank is configured, two or more Floors are selected for inclusion in the Bank. This means that these Floors can be visited by any of the Bank's Cabins. Each inclusion of a Floor results in the creation of a Bank Level.

If a Bank is intended to service Floors **1** and **21-40**, for example, 20 Bank Levels would be created and linked. By definition then, a Bank Level is part of a single Bank. Since a Cabin must be able to visit at least two Floors to be useful, a Bank must service two or more Floors.

One way to block all access to a given Floor, would be to exclude it from any Bank in the Building. In this case the Floor is not serviced by any Bank. At some later time the Floor might be linked into one or more Banks for service.

Formalization

Bank Level(**Bank** + **Floor**)

R38

Bank Level *is a* **Top, Middle or Bottom Bank Level**

The constraint that a Bottom Floor can only call Cabins going up and that a Top Floor can only call Cabins going down while Middle Floors can call either direction is enforced by this specialization of Bank Level.

For a given Floor, the top, middle, bottom status is determined for each Bank membership. The highest Floor included in a Bank will be a Top Bank Level, the lowest a Bottom Bank Level and all others will be Middle Bank Levels.

We can imagine a case, for example, where the **20th floor** is simultaneously the Top Bank Level in Bank 1 and the Bottom Bank Level of Bank 2.

Formalization

Bank + **Floor** in each subclass

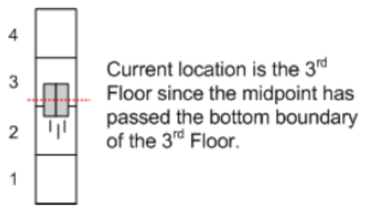
R43

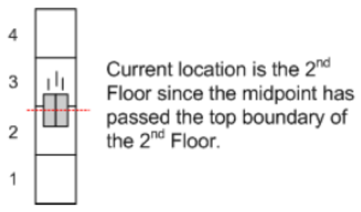
Cabin is at *one* **Floor**

Floor is location of *zero, one or many* **Cabin**

As a Cabin travels up and down it crosses Floor boundaries. The Transport service will update the current Floor location as the Cabin passes each Floor.

When a Cabin is between Floors moving up, the current Floor location will be the latest lower Floor boundary passed by the approximate midpoint of the Cabin. (The exact location can vary depending on where the sensors are located).


4
3
2
1
Current location is the 3rd Floor since the midpoint has passed the bottom boundary of the 3rd Floor.


4
3
2
1
Current location is the 2nd Floor since the midpoint has passed the top boundary of the 2nd Floor.

Going down, the current Floor location will be the latest upper Floor boundary passed by the approximate Cabin midpoint.

This means that the Cabin position will always be the last updated location. So a Cabin will always have a current location.

A given Shaft Level may or may not be the current location of the Shaft's Cabin.

Formalization

Cabin (**Shaft** + **Current floor**)

R49

Bank Level call has chosen for service *zero, one or many* **Accessible Shaft Level**

Accessible Shaft Level will service call from *zero, one or many* **Bank Level**

When a passenger requests a Cabin going up or down at a Floor for some Bank any Shaft in that Bank can potentially service the request. The Bank Level must apply some algorithm to choose the appropriate Cabin. Factors such as Cabin speed, location, direction and current workload can be taken into account. The choice is made immediately after the up/down request and registered as a Floor Service in the requested direction.

Formalization

Floor Service(**Bank** + **Floor** + **Shaft**)

R53

Cabin is going to *zero or one* **Accessible Shaft Level**

Accessible Shaft Level is current destination of *zero or one* **Cabin**

At any given point in time, a Cabin may or may not have a next destination. If there are no pending floor service requests and no stop requests for a given Shaft, the associated Cabin will have no destination.

A Cabin is directed to one destination at a time, so an Accessible Shaft Level is the current destination of a Cabin only if the Cabin has been dispatched to that location.

Formalization

Transfer (**Shaft** + **Destination floor**)