Fig. 1

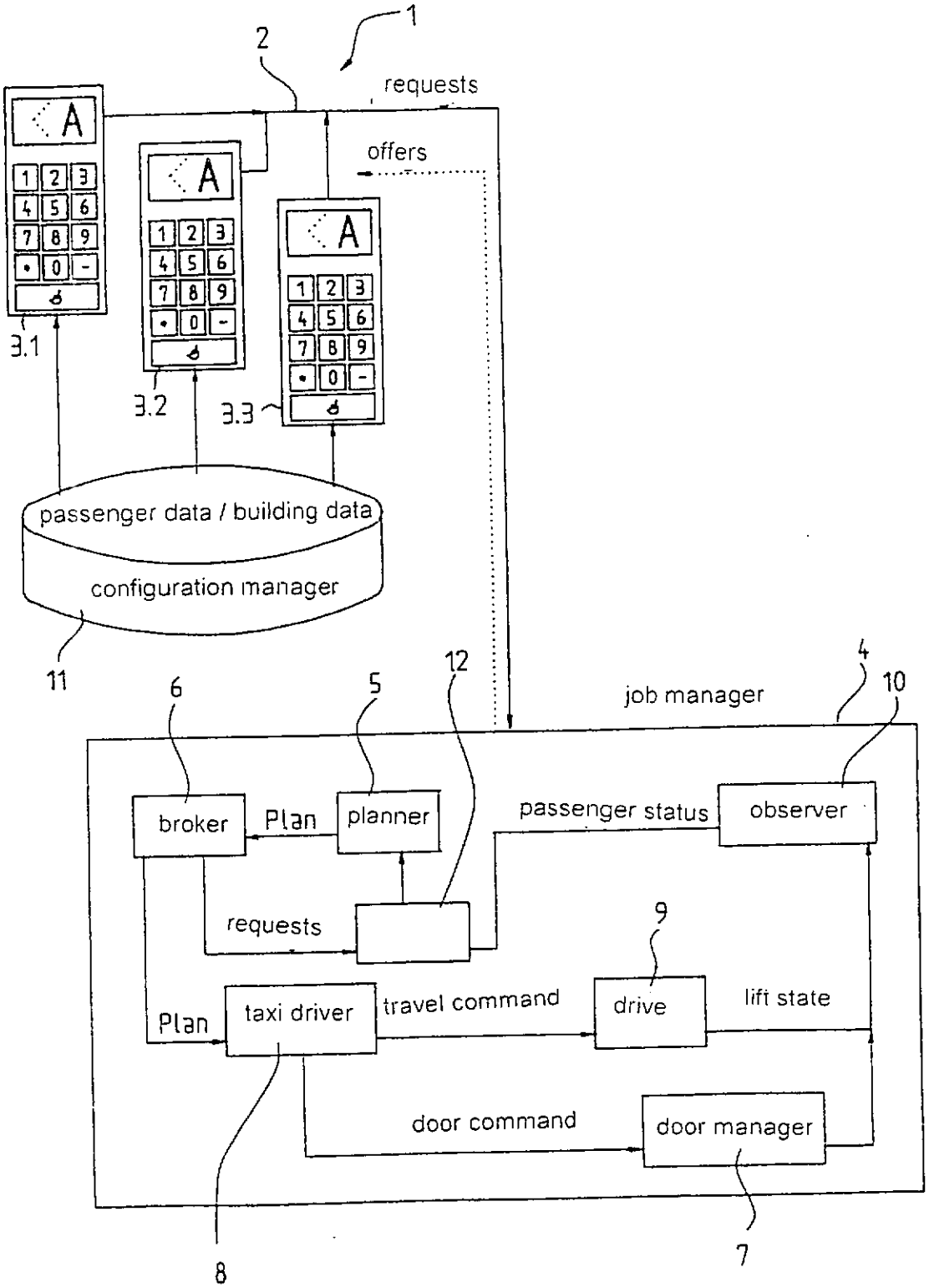

1
2
requests
offers
A
1 2 3
4 5 6
7 8 9
• 0 -
3.1
3.2
3.3
passenger data / building data
configuration manager
11
12
4
6
5
job manager
10
broker
Plan
planner
passenger status
observer
requests
9
Plan
taxi driver
travel command
drive
lift state
door command
door manager
8
7

Fig. 2

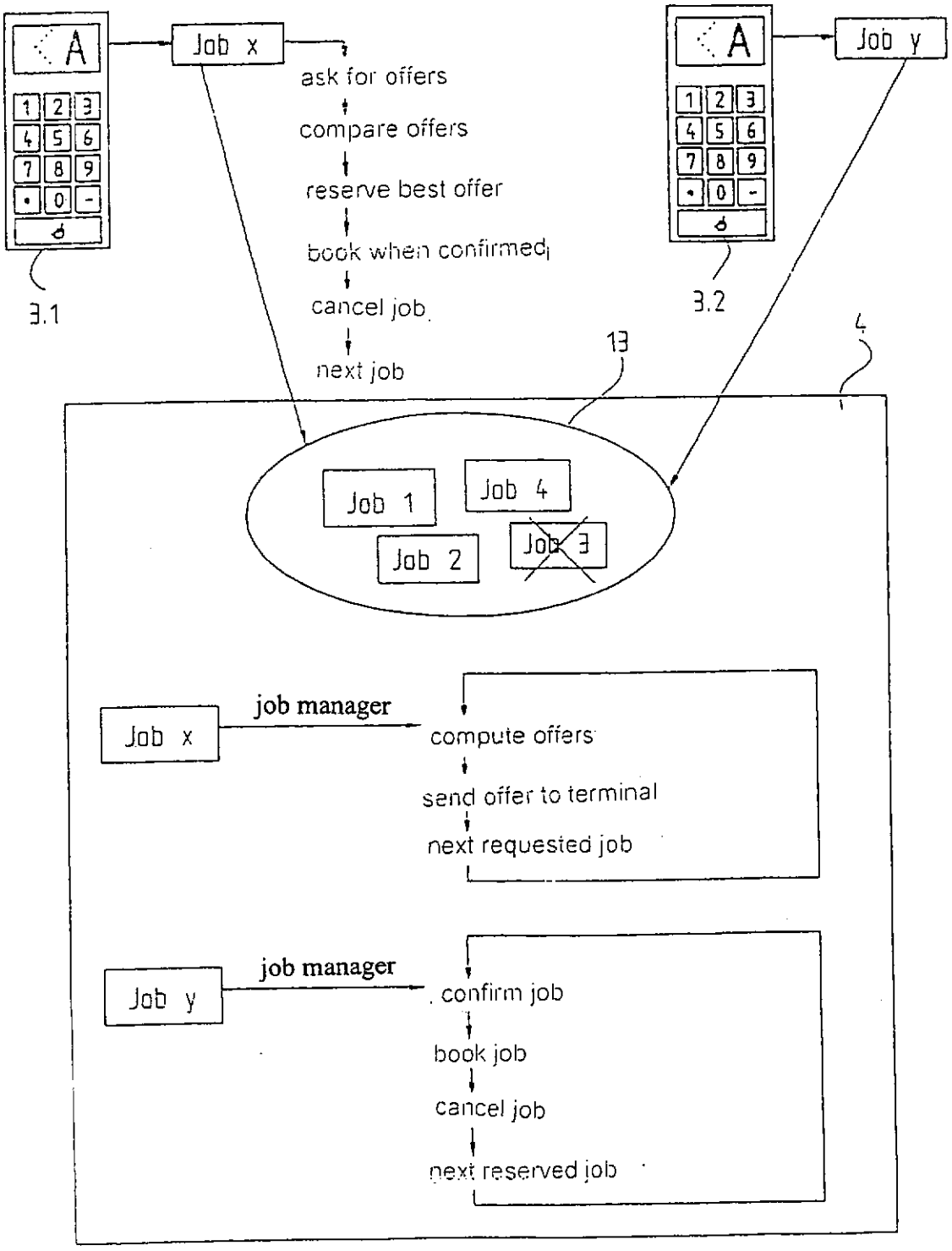

A
1 2 3
4 5 6
7 8 9
• 0 -
Job x
ask for offers
compare offers
reserve best offer
book when confirmed
cancel job
next job
3.1
A
Job y
3.2
13
4
Job 1
Job 4
Job 2
Job 3
job manager
Job x
compute offers
send offer to terminal
next requested job
job manager
Job y
confirm job
book job
cancel job
next reserved job

Fig. 3

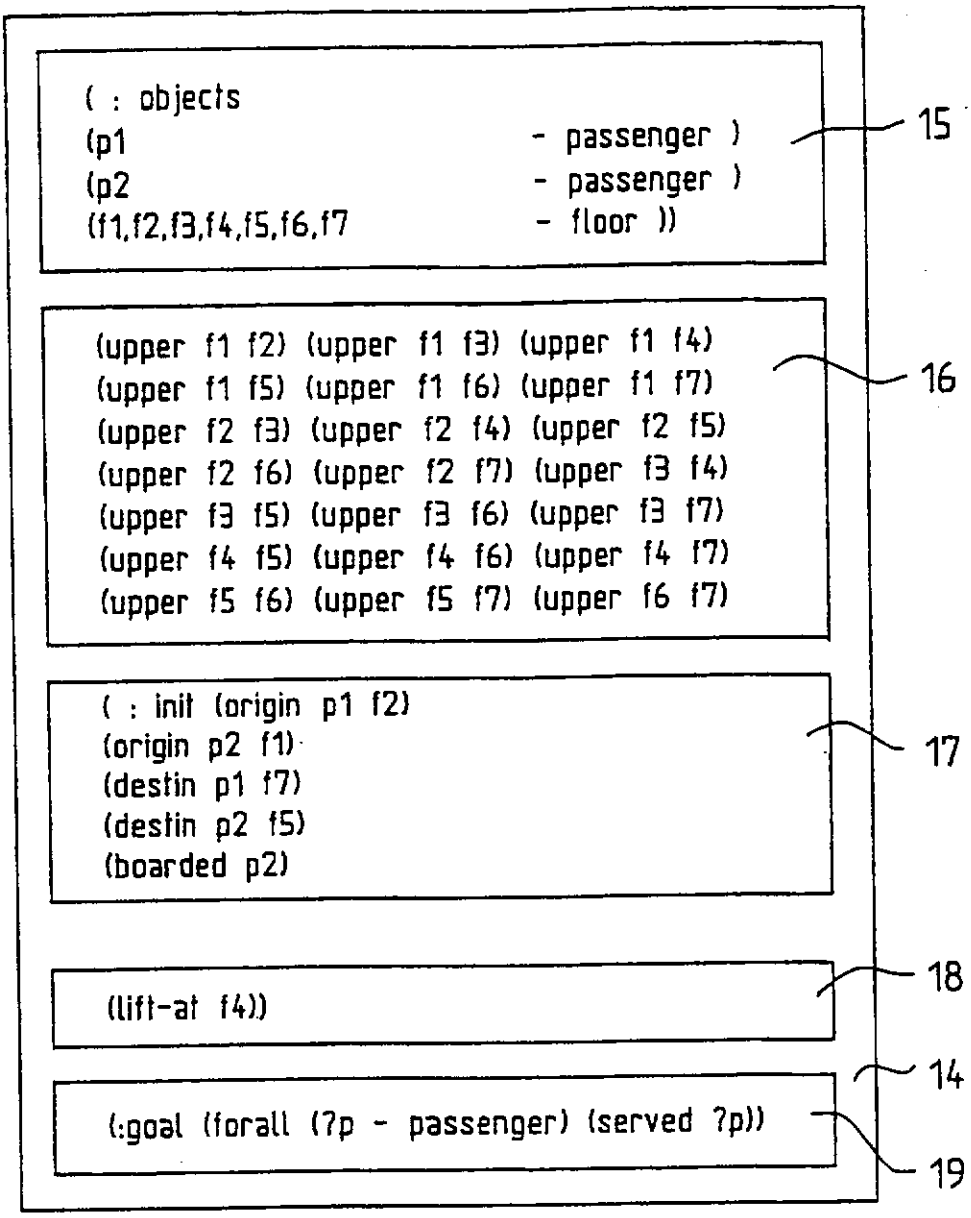

(: objects
(p1 - passenger)
(p2 - passenger)
(f1,f2,f3,f4,f5,f6,f7 - floor))
15
(upper f1 f2) (upper f1 f3) (upper f1 f4)
(upper f1 f5) (upper f1 f6) (upper f1 f7)
(upper f2 f3) (upper f2 f4) (upper f2 f5)
(upper f2 f6) (upper f2 f7) (upper f3 f4)
(upper f3 f5) (upper f3 f6) (upper f3 f7)
(upper f4 f5) (upper f4 f6) (upper f4 f7)
(upper f5 f6) (upper f5 f7) (upper f6 f7)
16
(: init (origin p1 f2)
(origin p2 f1)
(destin p1 f7)
(destin p2 f5)
(boarded p2)
17
(lift-at f4))
18
(:goal (forall (?p - passenger) (served ?p))
14
19

Fig. 4

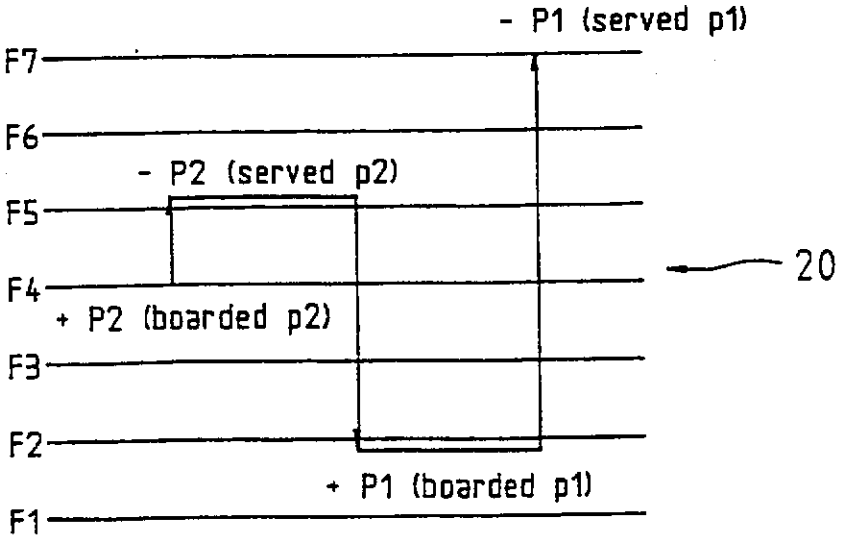

- P1 (served p1)
F7
F6
- P2 (served p2)
F5
F4
20
+ P2 (boarded p2)
F3
F2
+ P1 (boarded p1)
F1

Fig. 5

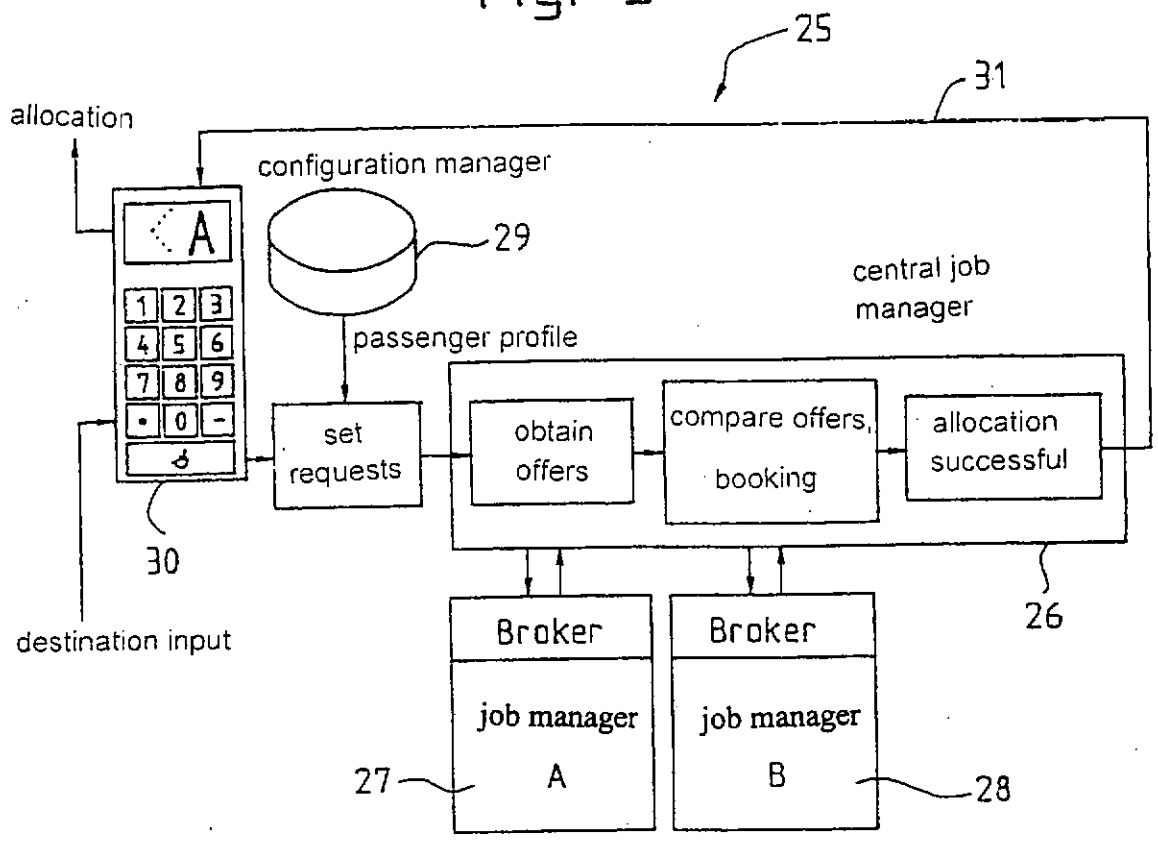

25
31
allocation
configuration manager
A
1 2 3
4 5 6
7 8 9
• 0 -
29
central job manager
passenger profile
set requests
obtain offers
compare offers, booking
allocation successful
30
26
destination input
Broker
job manager
A
27
Broker
job manager
B
28

Fig. 6

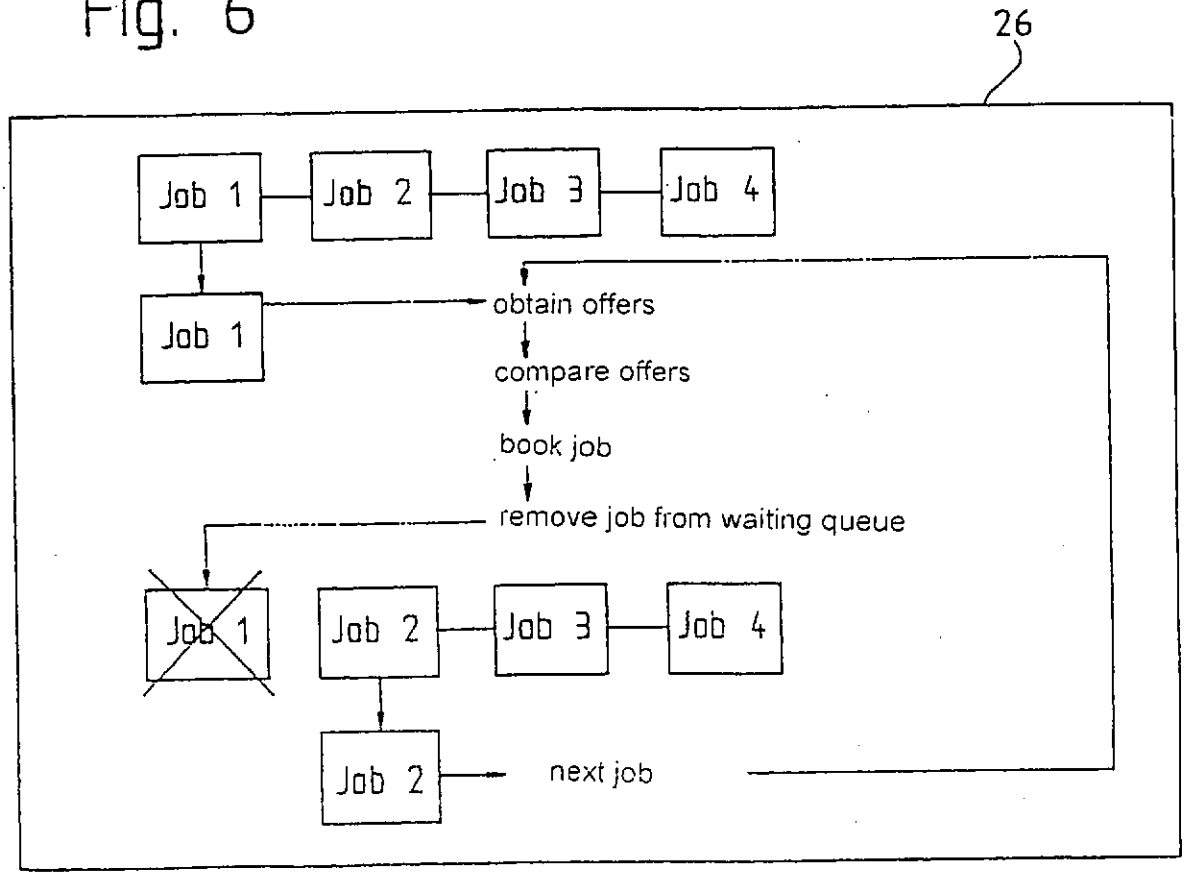

26
Job 1
Job 2
Job 3
Job 4
Job 1
obtain offers
compare offers
book job
remove job from waiting queue
Job 1
Job 2
Job 3
Job 4
Job 2
next job

Fig. 7

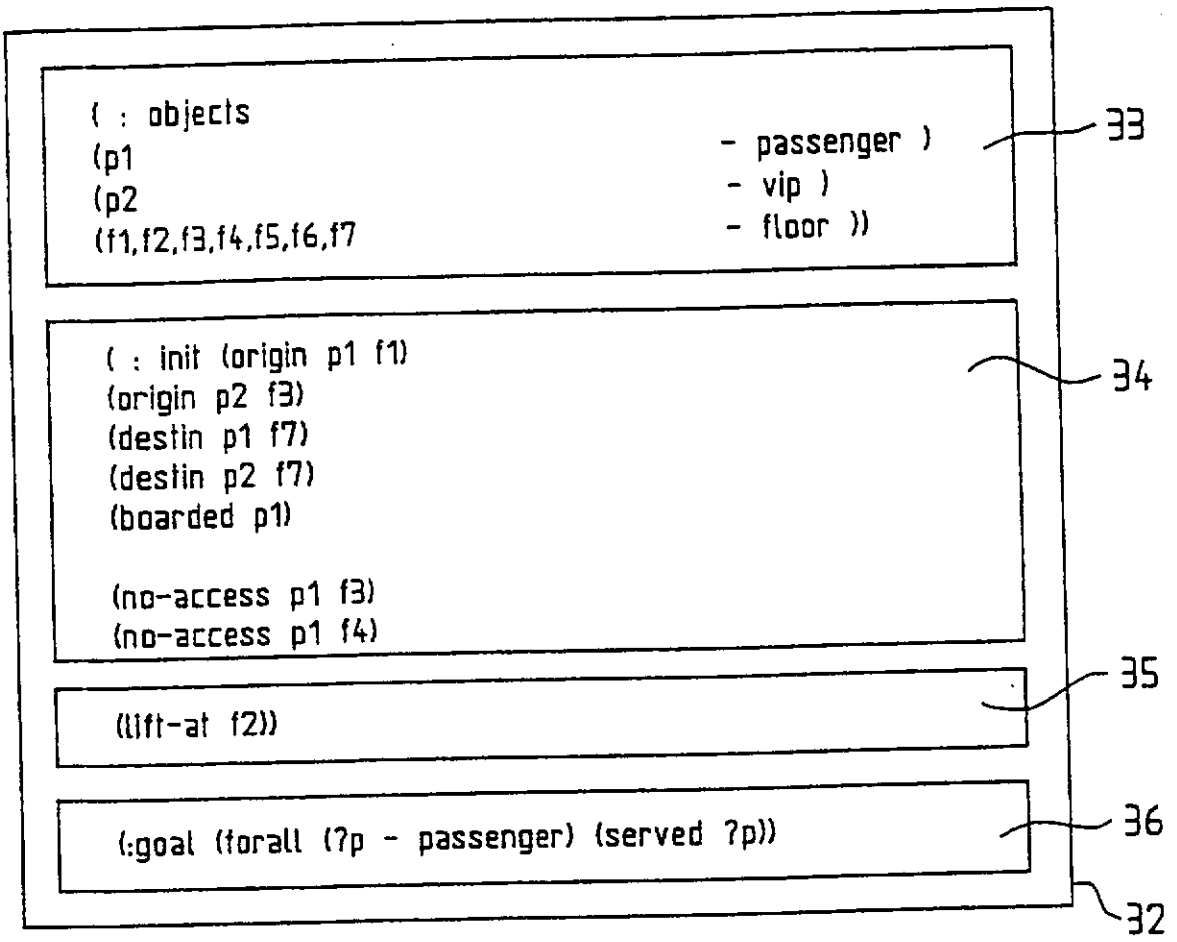

(: objects
(p1
(p2
(f1,f2,f3,f4,f5,f6,f7
- passenger)
- vip)
- floor))
33
(: init (origin p1 f1)
(origin p2 f3)
(destin p1 f7)
(destin p2 f7)
(boarded p1)
(no-access p1 f3)
(no-access p1 f4)
34
(lift-at f2))
35
(:goal (forall (?p - passenger) (served ?p))
36
32

Fig. 8

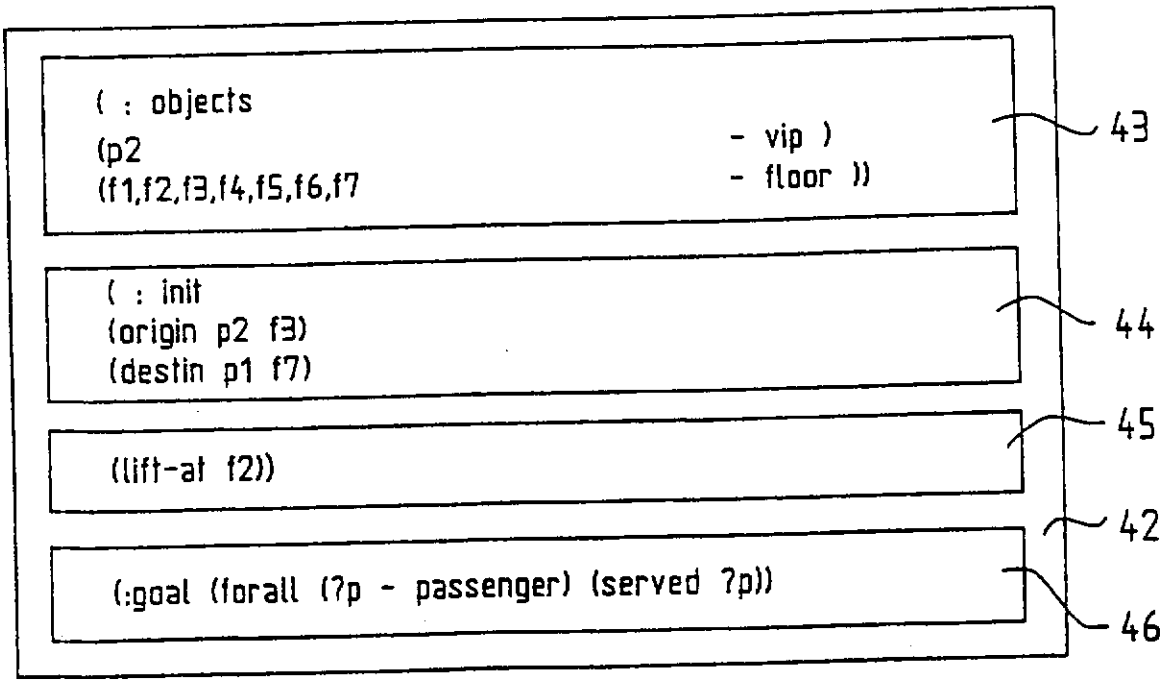

(: objects
(p2
(f1,f2,f3,f4,f5,f6,f7
- vip)
- floor))
43
(: init
(origin p2 f3)
(destin p1 f7)
44
(lift-at f2))
45
42
(:goal (forall (?p - passenger) (served ?p))
46

Fig. 9

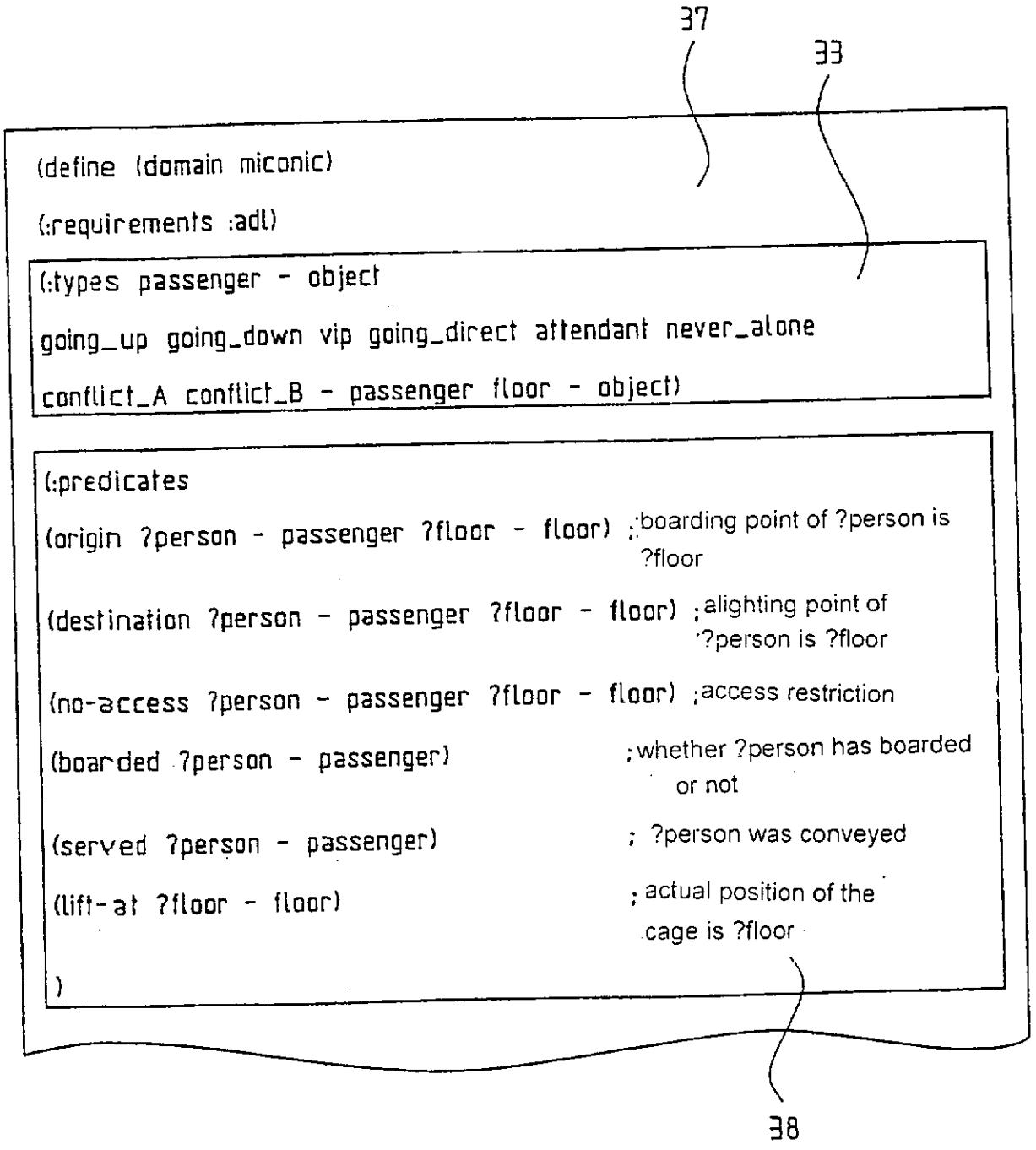

37
33
(define (domain miconic)
(:requirements :adl)
(:types passenger - object
going_up going_down vip going_direct attendant never_alone
conflict_A conflict_B - passenger floor - object)
(:predicates
(origin ?person - passenger ?floor - floor) ;boarding point of ?person is ?floor
(destination ?person - passenger ?floor - floor) ;alighting point of ?person is ?floor
(no-access ?person - passenger ?floor - floor) ;access restriction
(boarded ?person - passenger) ;whether ?person has boarded or not
(served ?person - passenger) ; ?person was conveyed
(lift-at ?floor - floor) ; actual position of the cage is ?floor
)
38

Fig. 9

39

```
(:action stop
 :parameters (?f - floor)
 :precondition (and (lift-at ?f)
 (imply (exists (?p - conflict_A)
 (or (and (not (served ?p)) (origin ?p ?f)) (and (boarded ?p) (not
 (destination ?p ?f))))))
 (forall (?q - conflict_B) (and (not (boarded ?q) (or (served ?q)
 (not (origin ?q ?f))))))
 (imply (exists (?p - conflict_B)
 (or (and (not (served ?p)) (origin ?p ?f)) (and (boarded ?p) (not
 (destination ?p ?f))))))
 (forall (?q - conflict_A) (and (not (boarded ?q)) (or (served ?q)
 (not (origin ?q ?f))))))
 (imply (exists (?p - never_alone) (or (and (origin ?p ?f)
 (not (served ?p)))
 (and (boarded ?p) (not (destination ?p ?f))))))
 (exists (?q - attendant) (or (and boarded ?q) (not (destination ?q ?f))
 (and (not (served ?q)) origin ?q ?f))))))
 (forall (?p - going_direct) (imply (boarded ?p) (destination ?p ?f)))
 (or (forall (?p - vip) (served ?p)) (exists (?p - vip) (or (origin ?p ?f)
 (destination ?p ?f))))
 (forall (?p - passenger) (imply (no-access ?p ?f) (not (boarded ?p)))))
 ;effect (and (forall (?p - passenger) (when (and (boarded ?p)
 (destination ?p ?f))
 (and (not (boarded ?p)) (served ?p))))
 (forall (?p - passenger) (when (and (origin ?p ?f) (not (served ?p)))
 (boarded ?p)))))
```