

Tervezési feladat: Logisztika

Szoftvertechnológia

Dr. Simon Balázs
BME, IIT

Tartalom

- Követelmények
- Tervezési lépések: osztálydiagram
- Szekvenciadiagramok
- Tervezési alternatívák

Követelmények: Logisztika

Követelmények

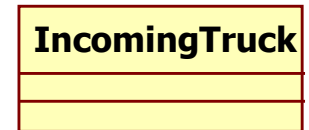
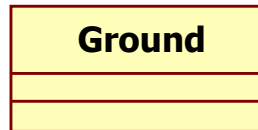
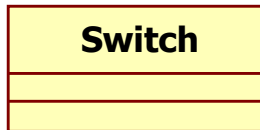
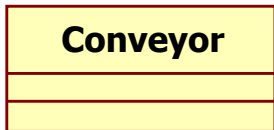
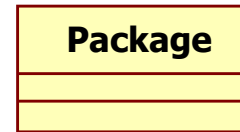
- A csomagelosztó állomáson a teherautókon beérkező csomagokat egy futószalag-rendszer közvetítésével más teherautókra rakják át. A játékos feladata a futószalag-rendszer irányítása.
- A teherautók az elosztó állomás be- és kilépő pontjaihoz kapcsolódva csomagokat hoznak és visznek. A csomagra jellemző a színe és a romlandósága. Az elosztó állomásra belépő csomagot arra a teherautóra kell továbbítani, amely az adott színű csomagokra vár. Ha a csomag romlandó (nem mindegyik az), akkor annak még a romlási idő lejártá előtt a teherautóra kell érkeznie. Ha a romlási idő lejártakor a csomag még nincs a teherautón, akkor a csomag felrobban.
- A belépő ponthoz álló teherautóról a csomagok véletlenszerű időközönként a pontot érintő (vagy onnan induló) futószalagra esnek. A futószalagról - annak végén - a csomag leesik. A leeső csomag eshet
 - egy másik futószalag valamely pontjára,
 - egy a futószalag végén álló teherautóra,
 - a földre (ha nincs ott teherautó, vagy a futószalag a semmibe vezet),
 - váltóra.

Követelmények

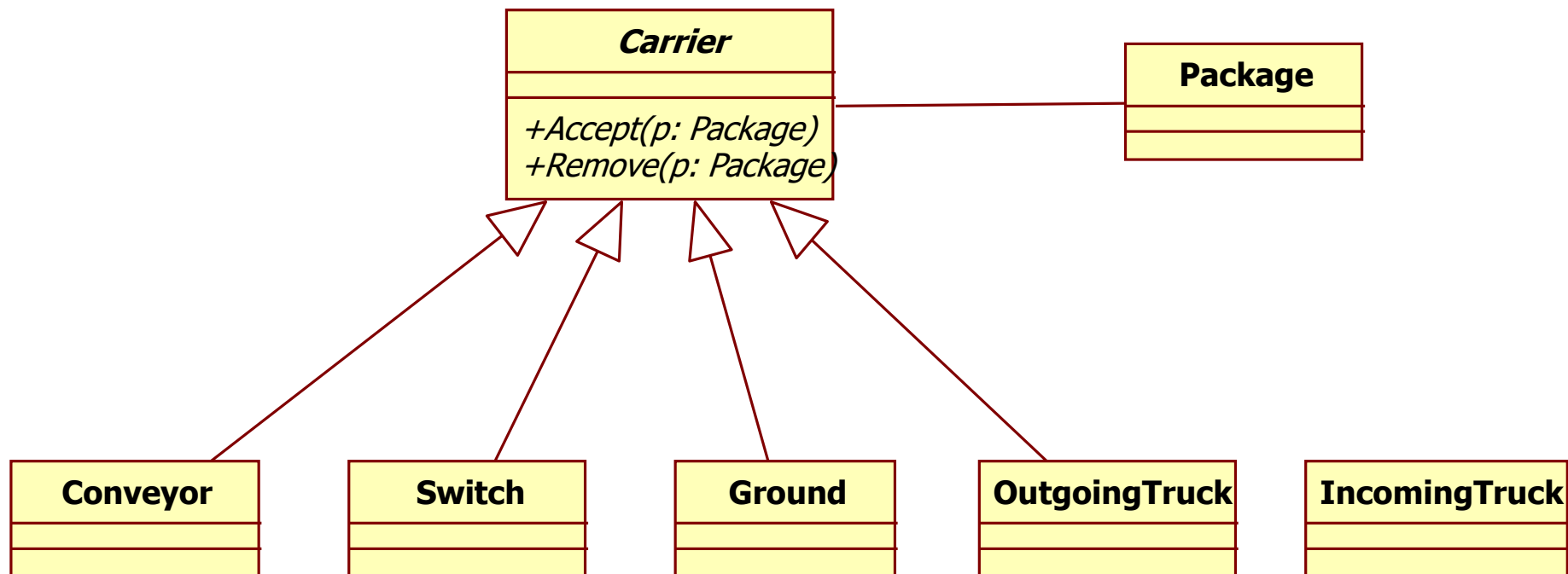
- A váltó célja a csomag irányítása a futószalagok között, hasonlóképp, mint a vasútnál. Egy váltóhoz egy vagy több futószalag hozza a csomagot, és mindig két vagy több futószalag egyike viszi el. A váltóra eső csomag némi várakozás után a beállítástól függő elmenő futószalagra esik rá. A játékos képes a váltók állítására.
- Pontot ér, ha a csomag olyan teherautóba esik, amelyik az adott színű csomagra (is) vár. Ha nem a színének megfelelő teherautóhoz lett a csomag irányítva, akkor az már hiba. Ha a csomag a földre esik, akkor összetörik. Ha a csomag futószalagra vagy váltóra esik és ott ütközik egy éppen ott tartózkodó csomaggal, akkor az ott tartózkodó csomag összetörik. A csomag összetörése nagy hiba.
- A beszállító teherautón található csomagok száma és minősége (színe, romlandósága) a játékos számára előre nem ismert - véletlenszerű. Ugyancsak véletlenszerű és a játékos által ismeretlen a kiszállító teherautó szállítási terve. A terv előírja, hogy a teherautóra milyen színű csomagból hányat kell felpakolni.

Tervezési lépések: Osztálydiagram

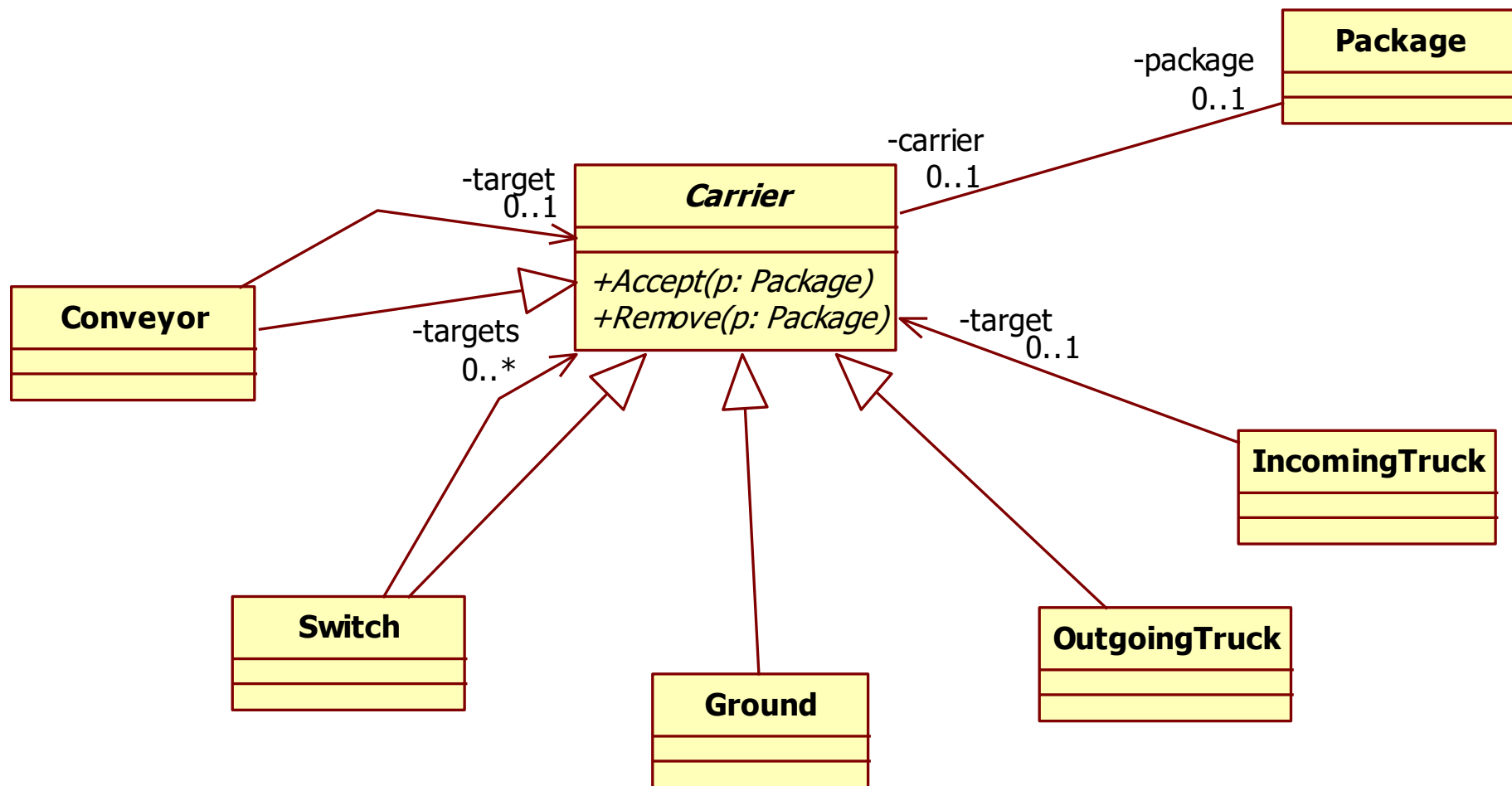
Legfontosabb osztályok



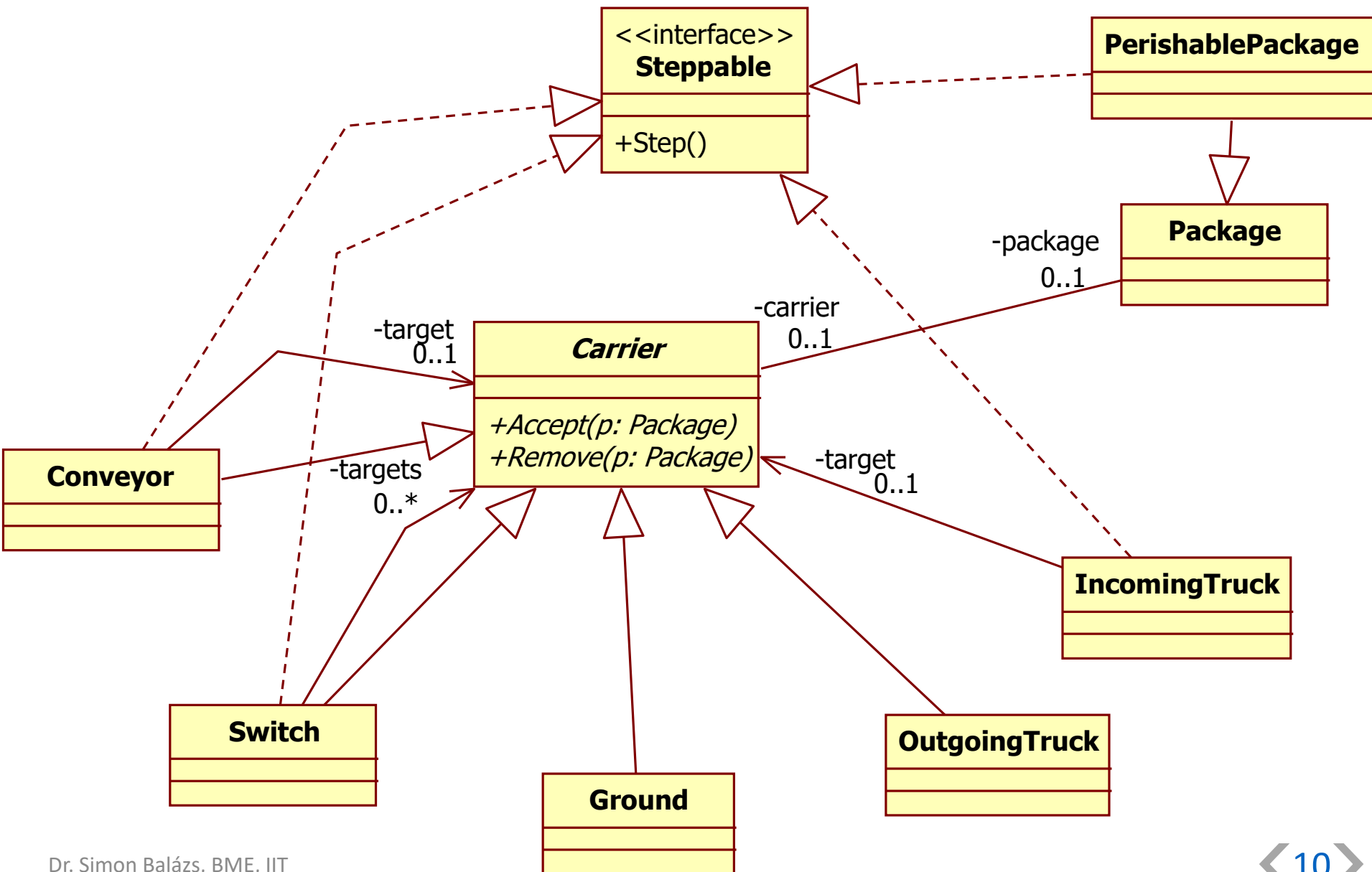
Közös űs, kapcsolat a csomaggal



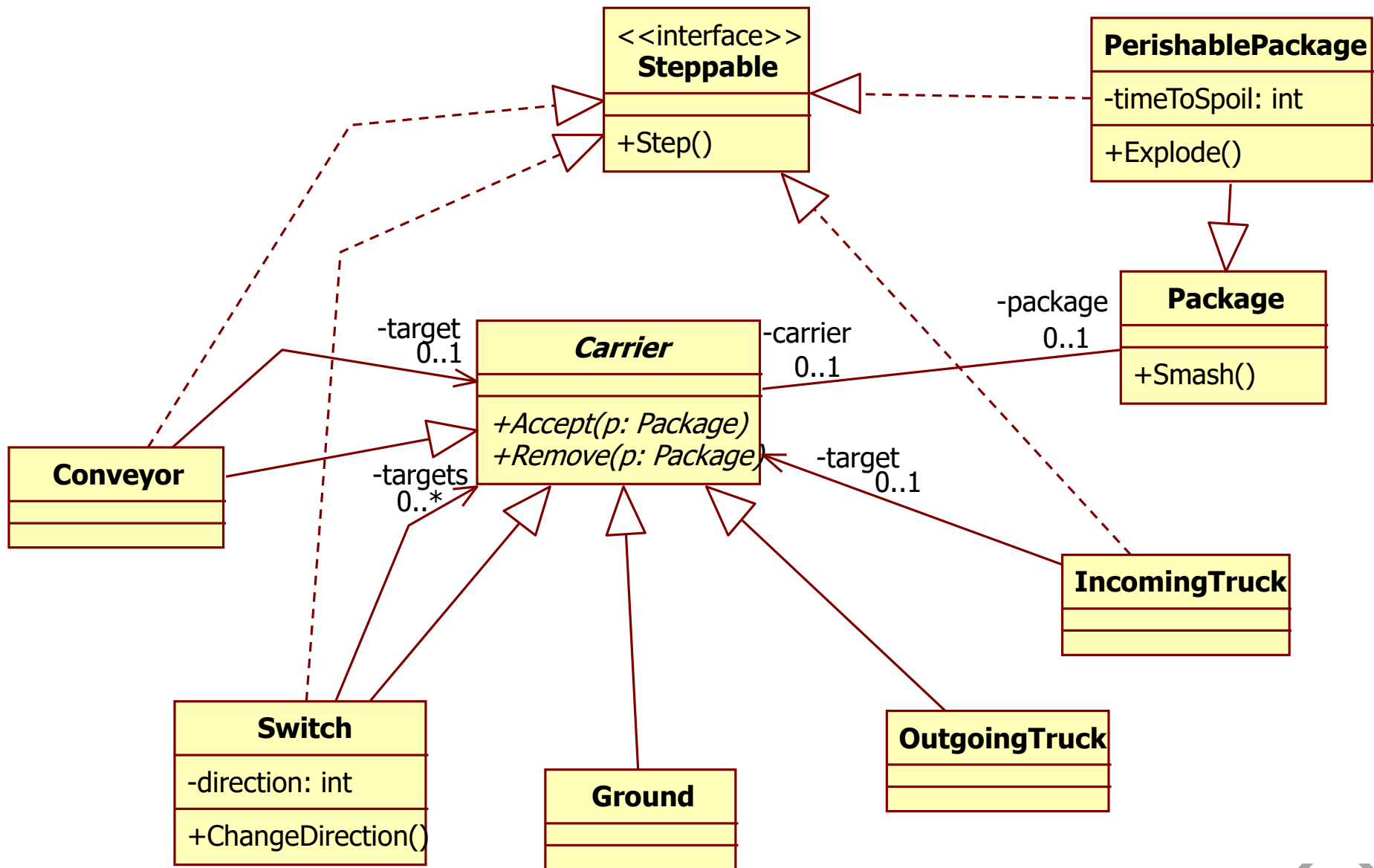
Következő szállító elem



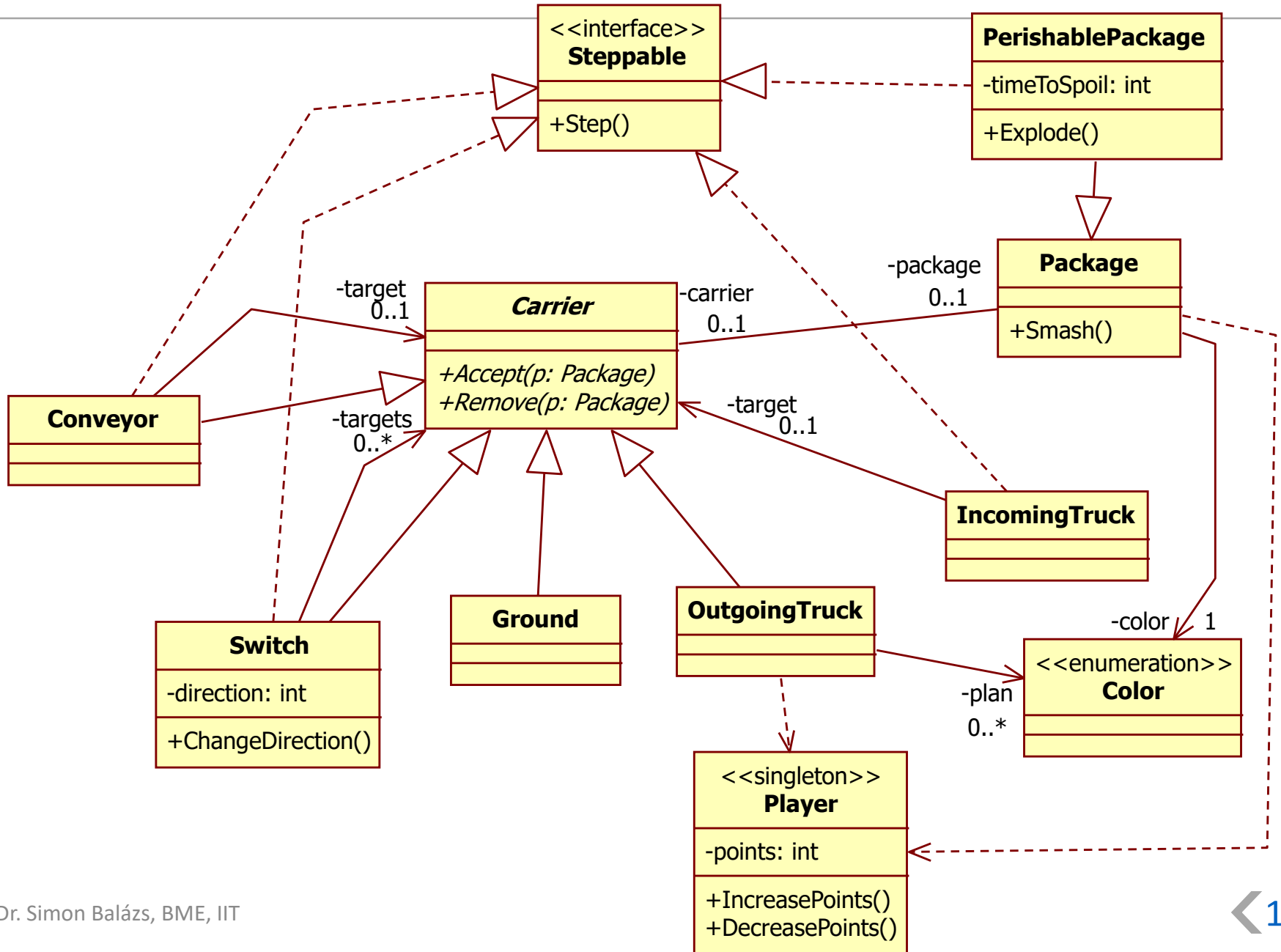
Léptethető elemek



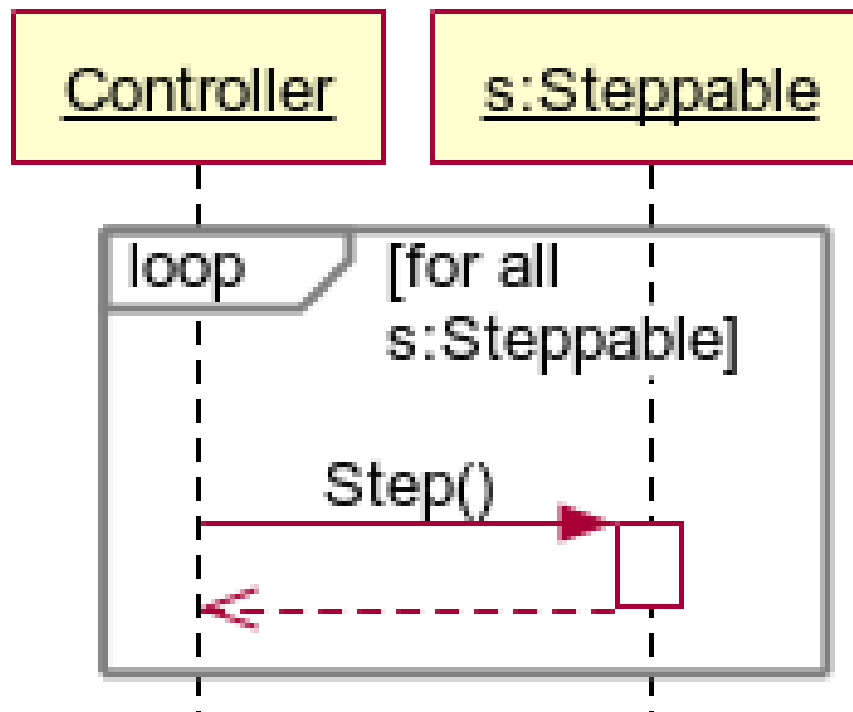
Maradék függvények és mezők



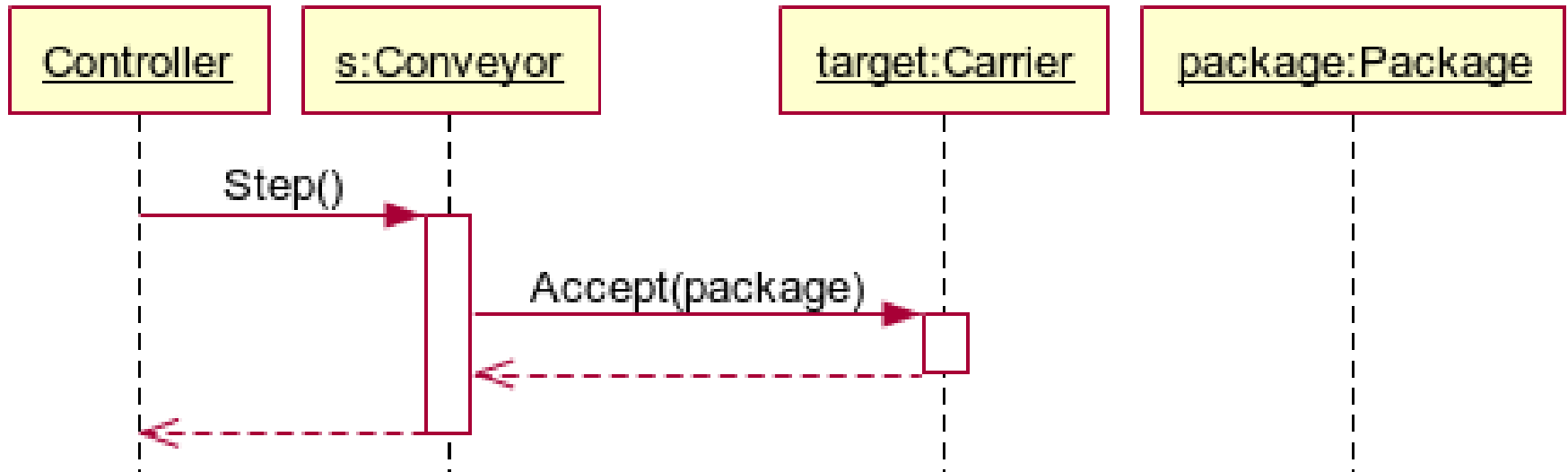
Játékos, színek



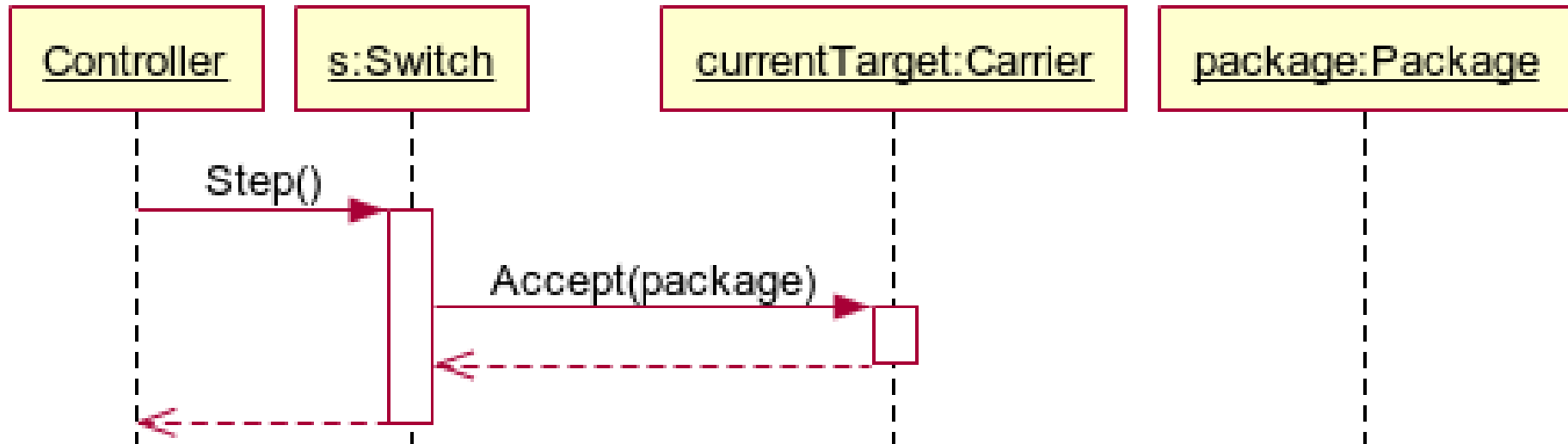
Szekvenciadiagramok



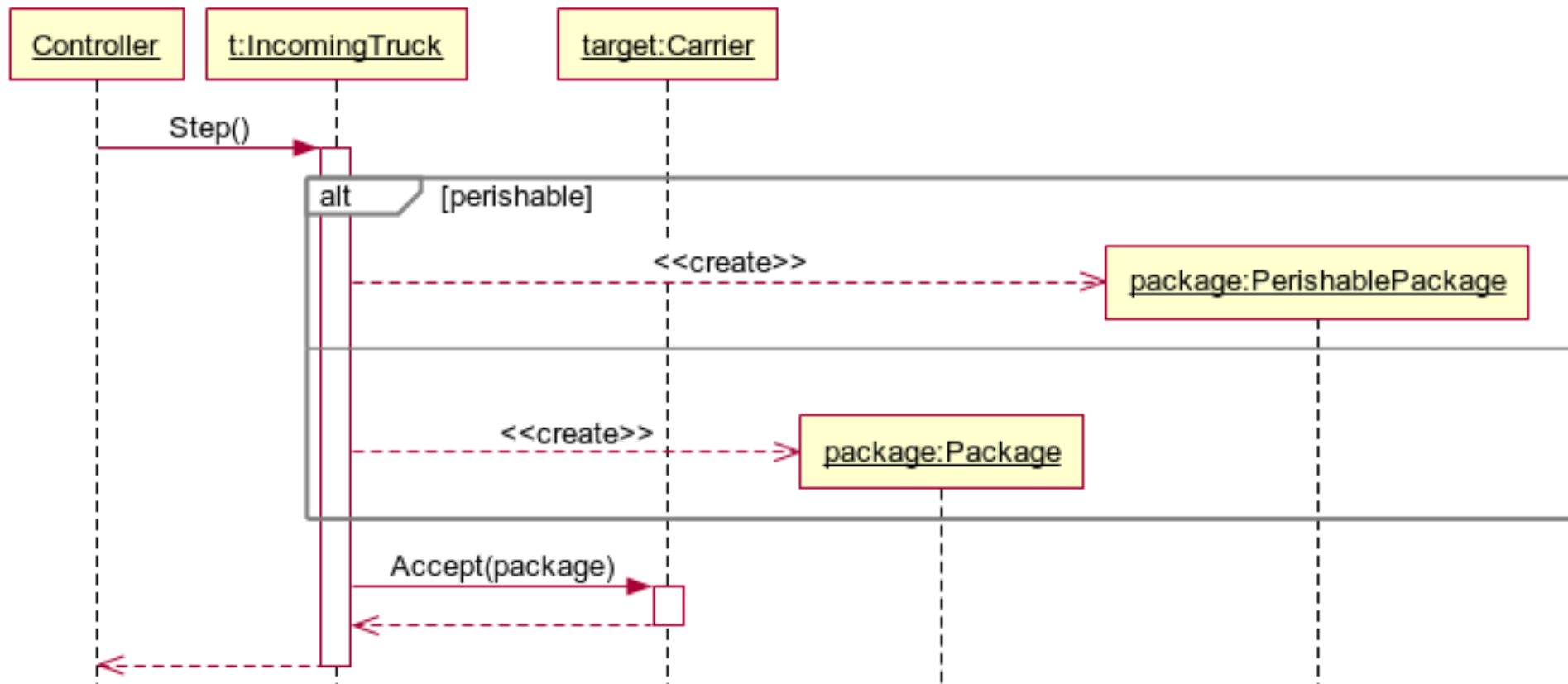
Futószalag lép



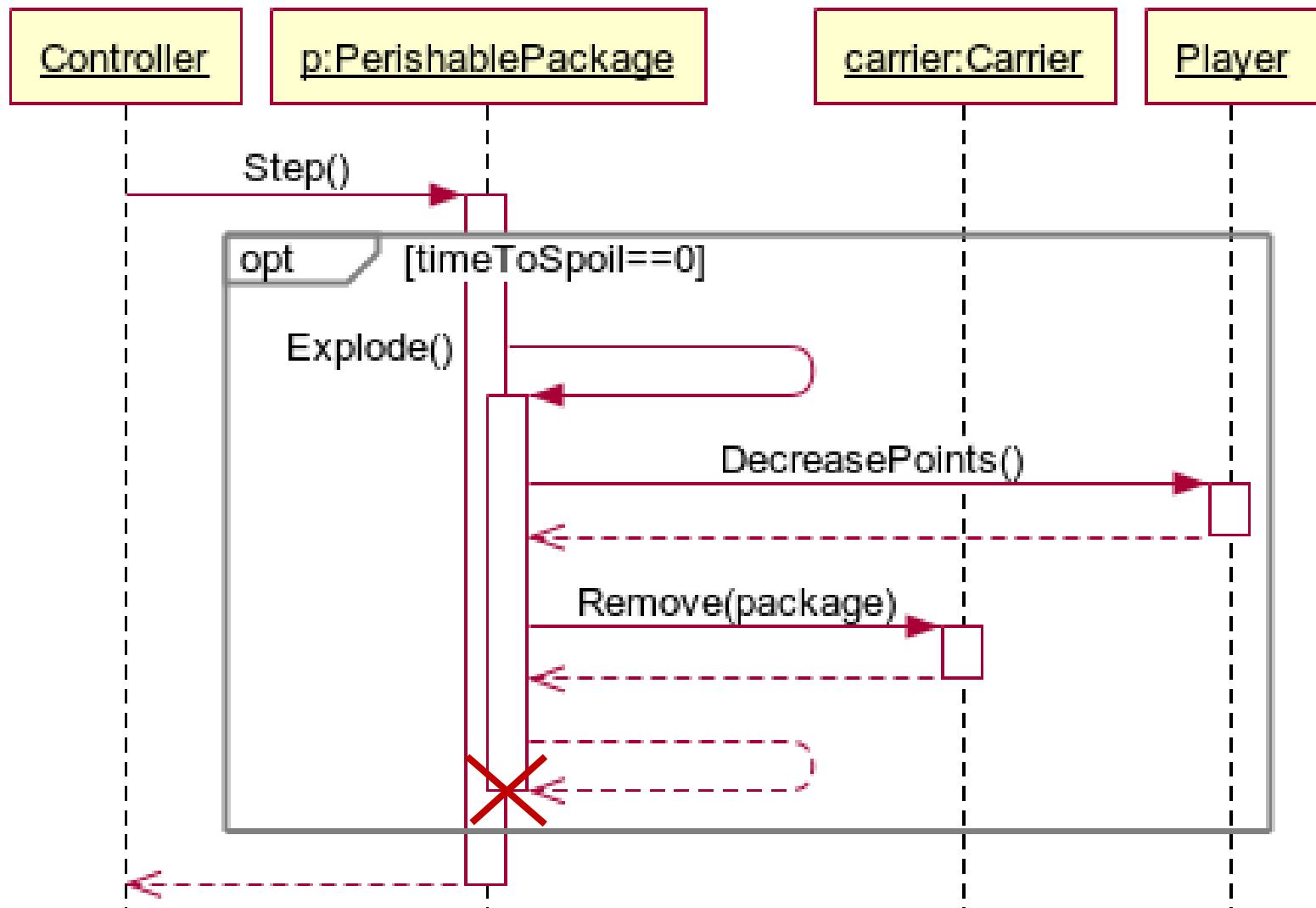
Váltó lép



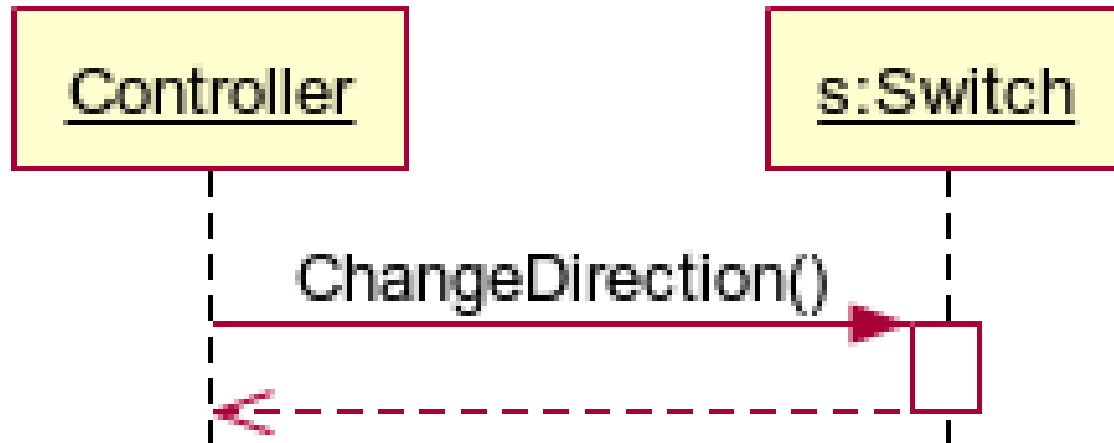
Beérkező teherautó lép



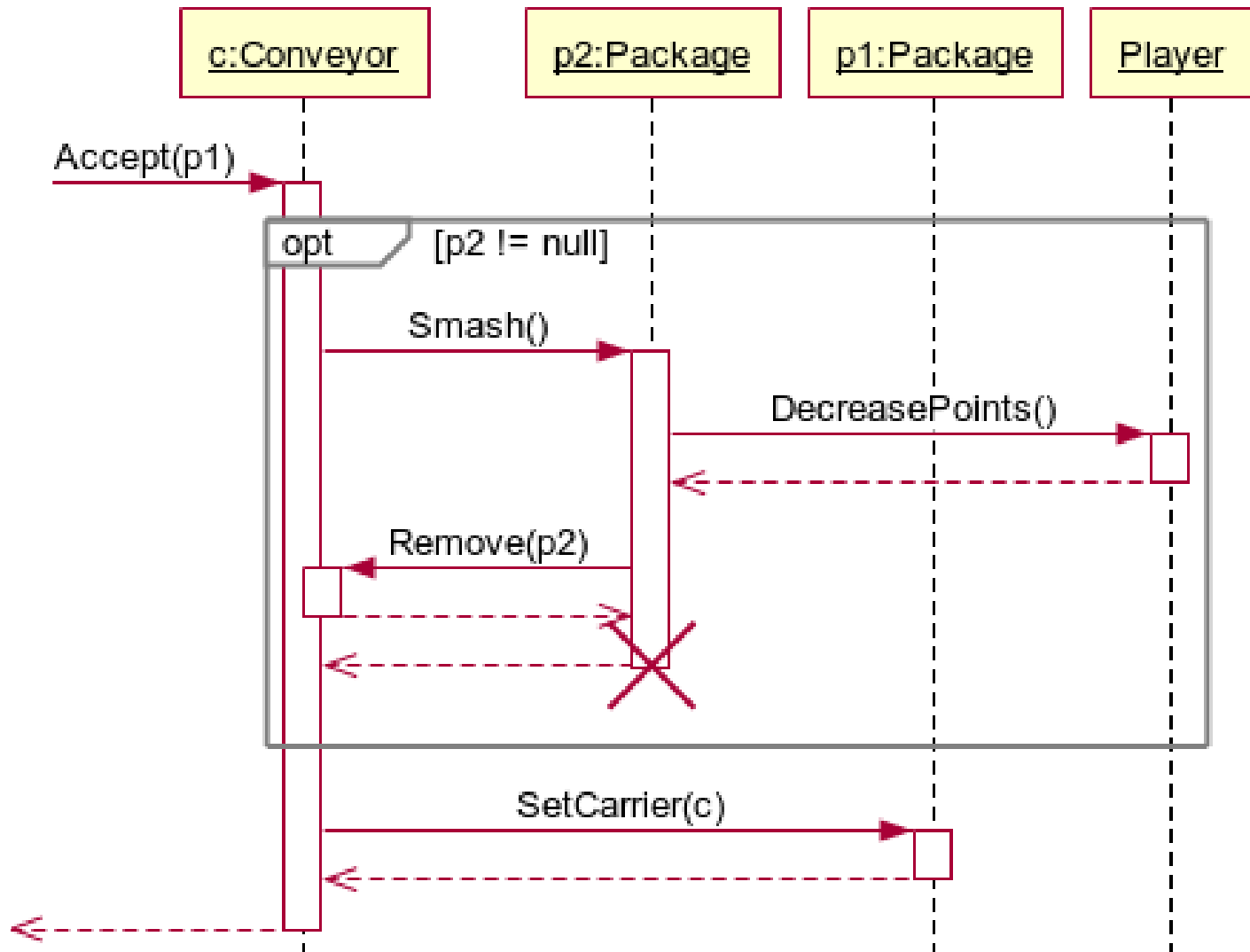
Romlandó csomag lép



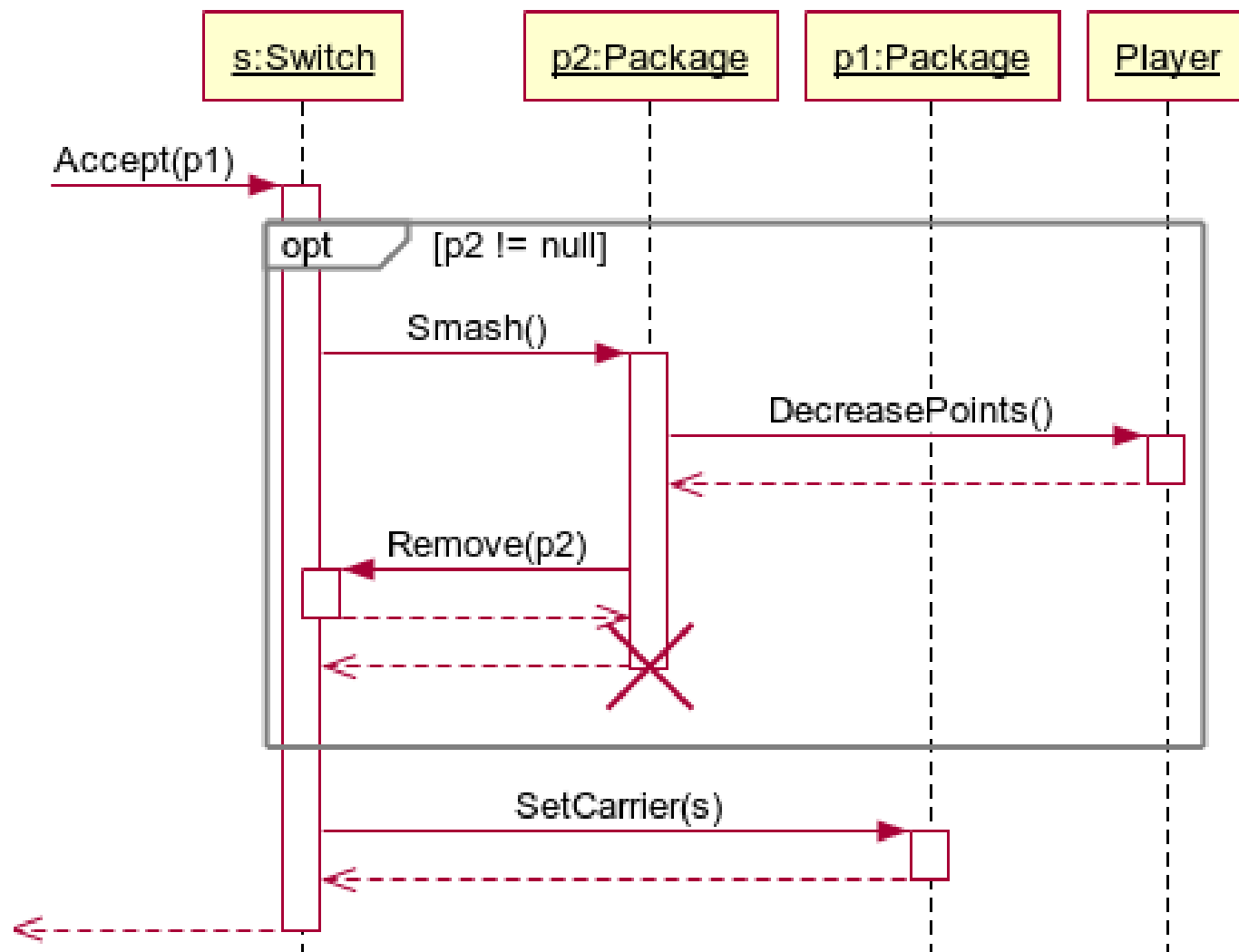
Váltó irányának állítása



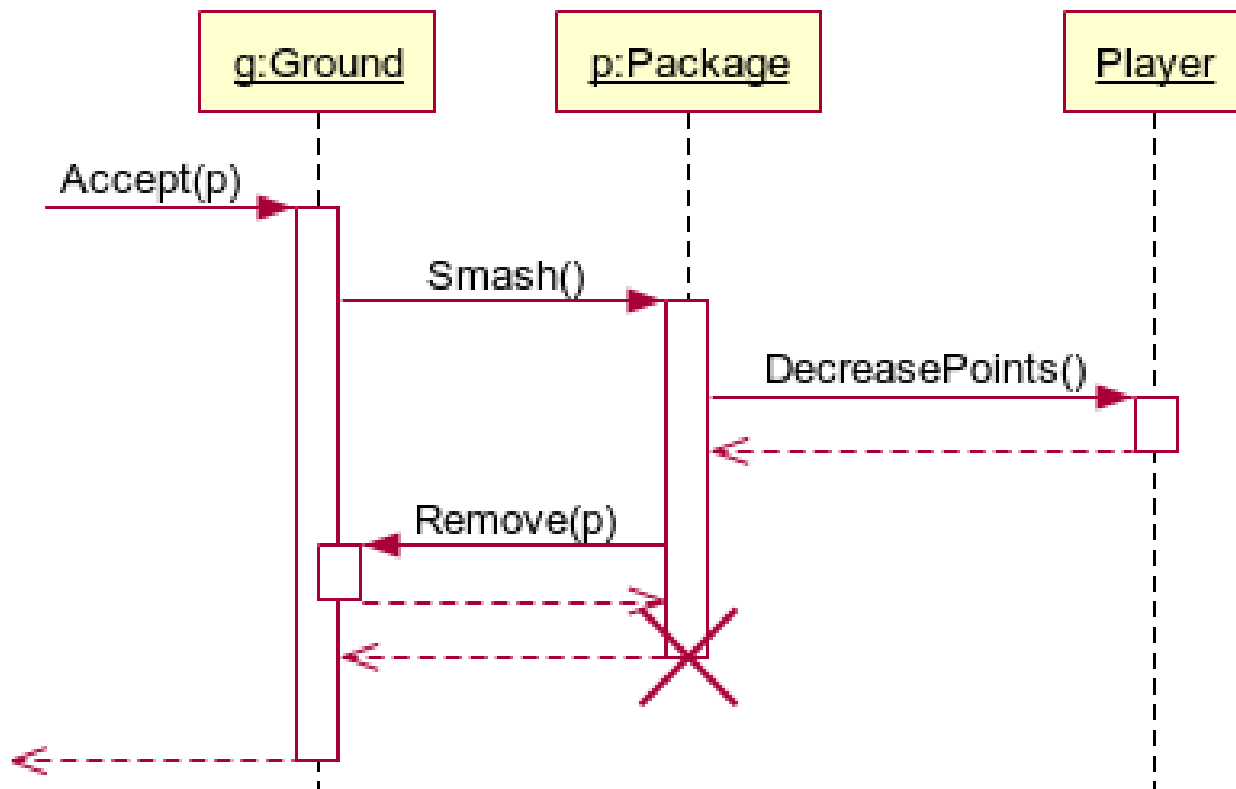
Csomag futószalagra érkezik



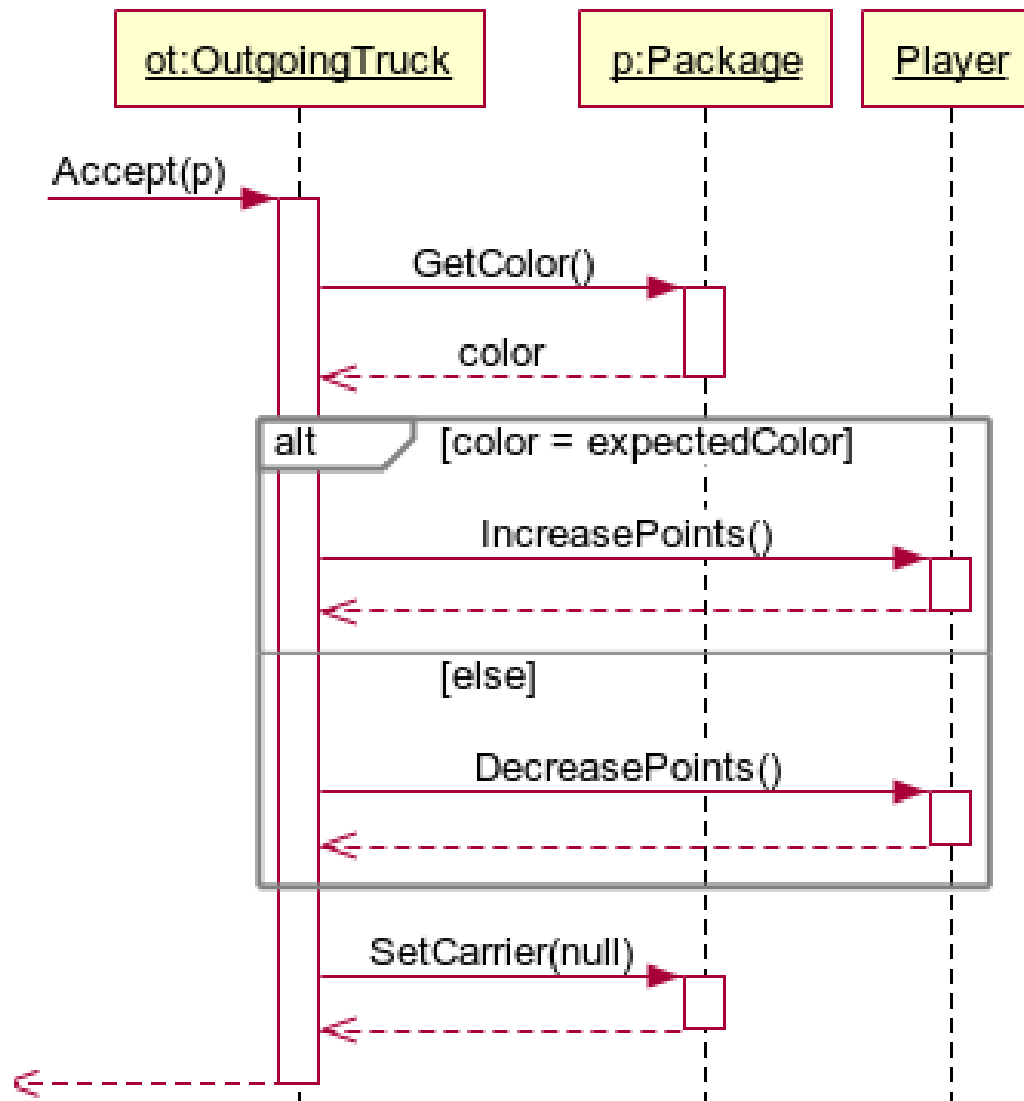
Csomag váltóra érkezik



Csomag földre esik



Csomag kimenő teherautóra érkezik



Tervezési alternatívák

Tervezési alternatívák (mind jók!)

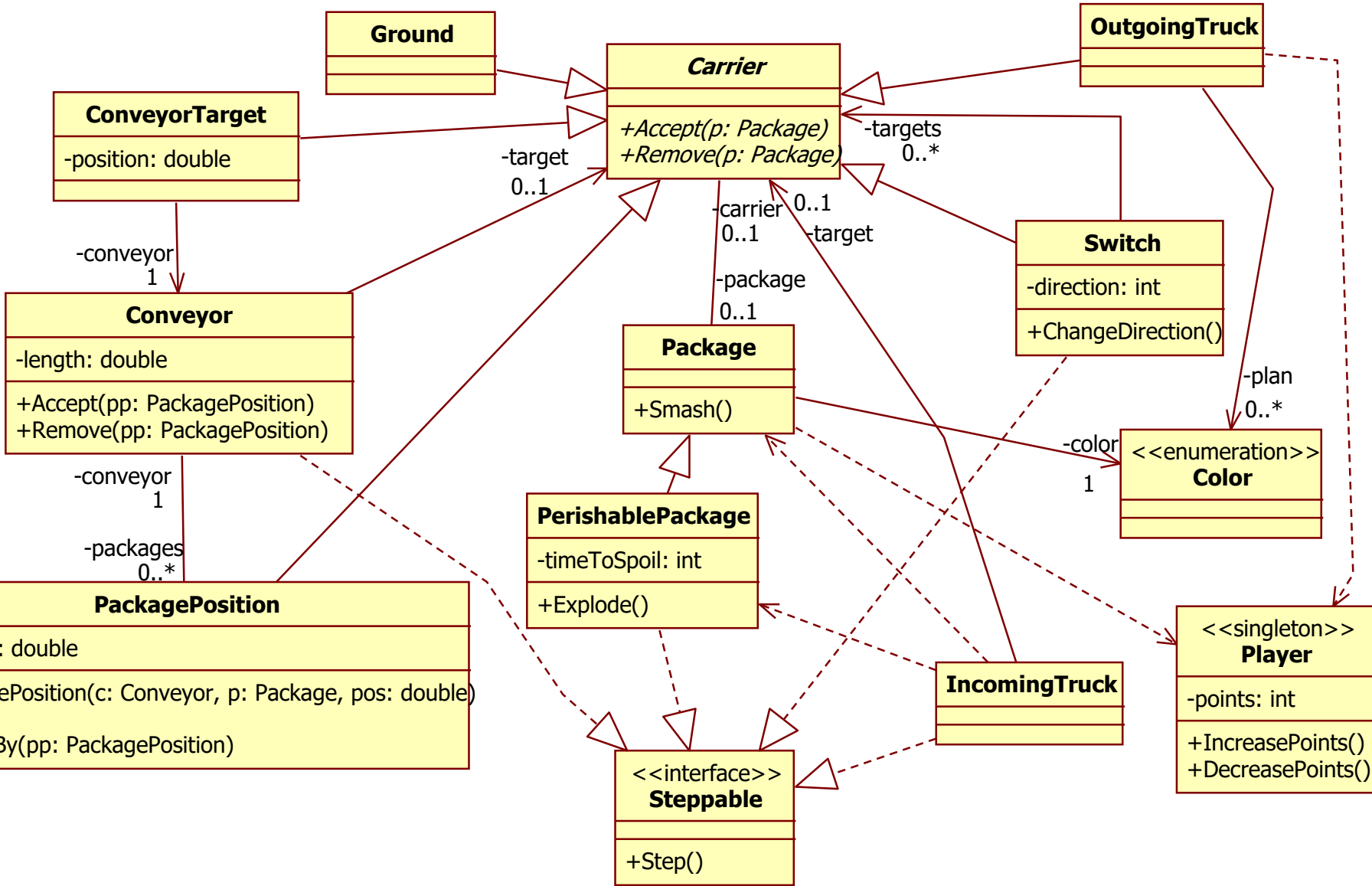
- Váltó a futószalag leszármazottja
 - mivel a szekvenciáik nagyon hasonlóak
 - a váltó nem indexként tárolja az irányt, hanem az ős futószalag célja határozza meg az irányt
 - a fordított eset (a futószalag a váltó leszármazottja) hibás: a váltónak több felelőssége van, így neki kell lennie a leszármazottnak
- Általános gráf helyett mezők
 - de párhuzamos rétegek nélkül: a futószalag, váltó, föld a mező leszármazottai, nem pedig a mezőn vannak rajta
 - *nem kell a Pacman-szerű HitBy callback*: nem kell tudni a csomagok típusát
- A futószalagnak van hossza, és több csomag is lehet rajta
 - kell egy pozíció, de csak a futószalagon: kell valami csomagoló osztály a csomaghoz, ami tárolja a pozíciót
 - nem a futószalag (Conveyor), hanem a csomagoló osztály a Carrier leszármazottja
- A futószalag több darabból áll, egy darabon egy csomag lehet
 - nincs szükség pozícióra
 - nem a futószalag (Conveyor), hanem a futószalag-darab a Carrier leszármazottja
- Váltó csak futószalagra dobhat csomagot
- A teherautók be- és kilépési pontja is modellezve van
 - a teherautók nem közvetlenül csatlakoznak a futószalagokhoz/váltókhoz/...
- A romlandó csomag és a sima csomag összevonható egy osztállyá
- stb.

Egyéb tervezői döntések

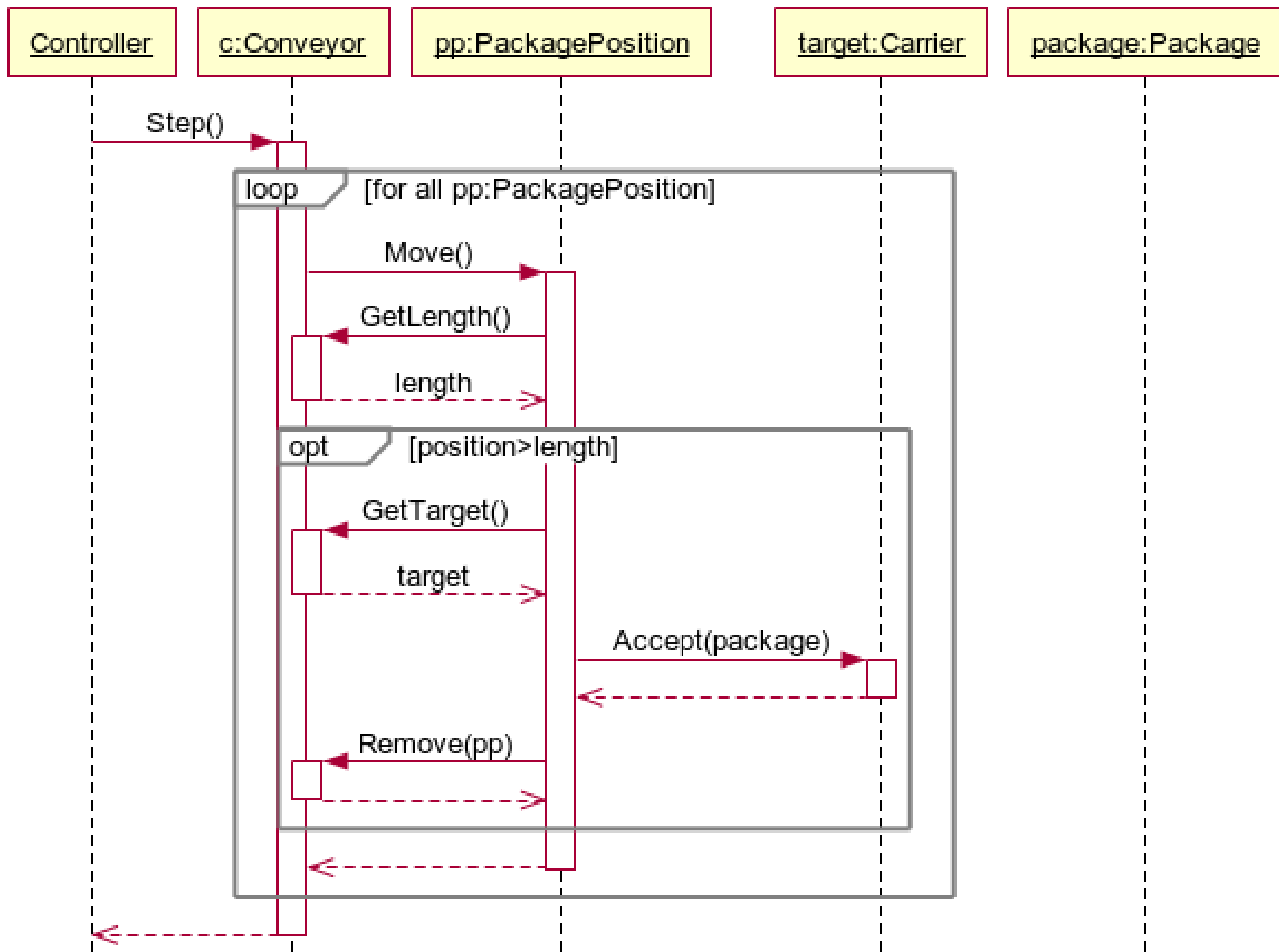
- Mi van, ha a teherautók tárolják a csomagokat?
 - a csomag osztálynak még kell függvény: `LeaveTruck()`, `ArriveToTruck()`
 - ezekkel lehet elindítani és leállítani a megromlást
 - bejövő teherautó: nem hoz létre csomagokat, egyszerűen csak továbbítja őket, és meghívja a `LeaveTruck()` függvényt
 - kimenő teherautó: a csomag szállítója a teherautó lesz, és meghívja az `ArriveToTruck()` függvényt
- Mi van, ha egy teherautó a helyétől függően lehet egyszer bejövő, máskor pedig kimenő?
 - csak egy teherautó osztály van, ami a `Carrier` leszármazottja, a kétfajta teherautó függvényeit össze kell fésülni

Példa egy alternatívára: a futószalagnak van hossza

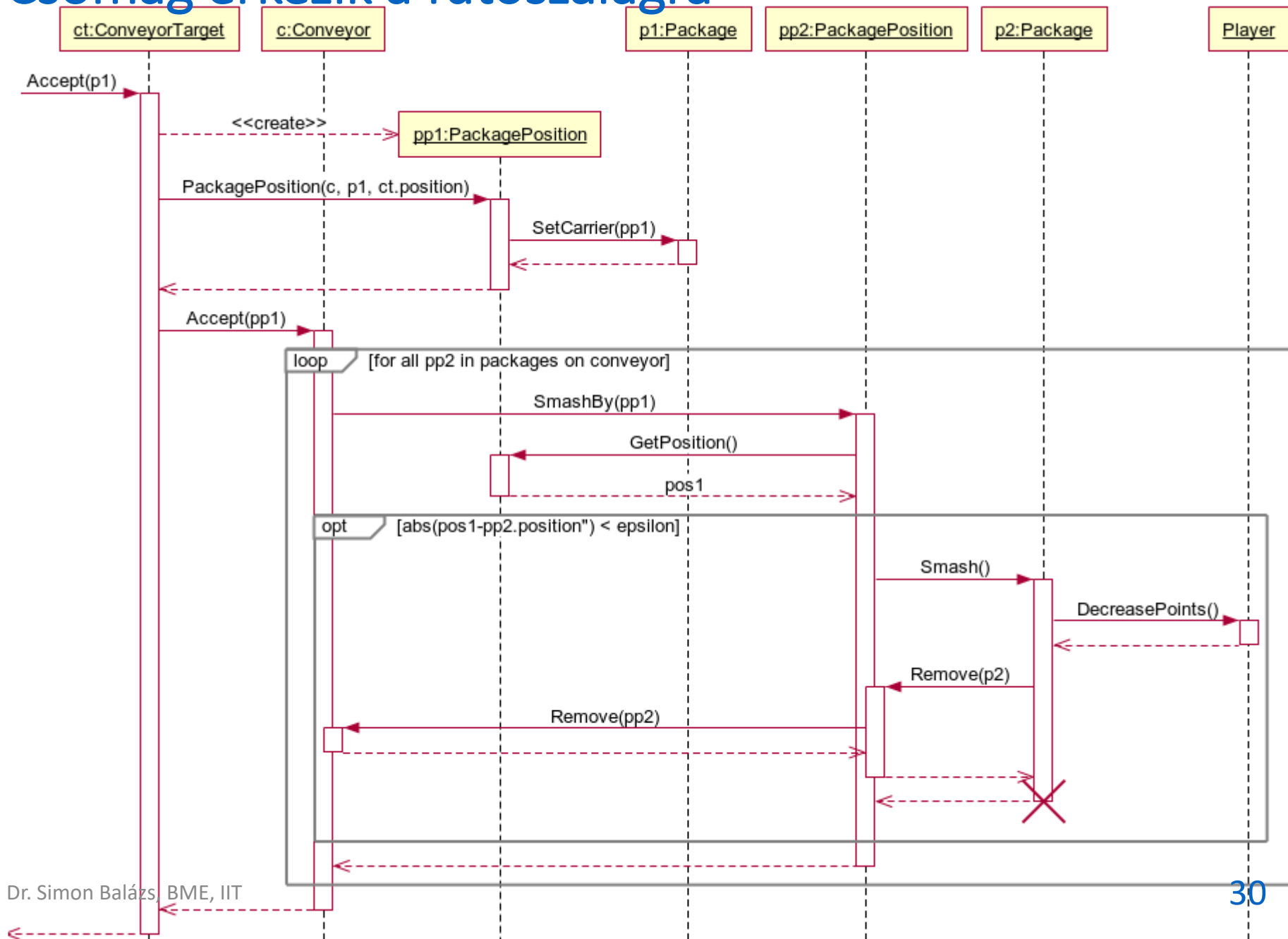
Futószalagnak van hossza, több csomag is lehet rajta



Futószalag lép



Csomag érkezik a futószalagra



A többi szekvencia

- Ugyanaz, mint korábban:
 - Léptetés
 - Váltó lép
 - Beérkező teherautó lép
 - Romlandó csomag lép
 - Váltó irányának állítása
 - Csomag földre esik
 - Csomag váltóra érkezik
 - Csomag kimenő teherautóra érkezik

Design task: Bank Robber

Software engineering
Szoftvertechnológia

Dr. Balázs Simon
BME, IIT

Outline

- Requirements for Bank Robber
- CRC cards for Bank Robber
- Class diagram
- Sequence diagrams

Requirements: Bank Robber

Requirements

- In a small town there has been a bank robbery. The robbers try to flee using a car, however, there are other cars on the road and also the police is after them. The player controls the car of the robbers.
- The roads in the town are all one-way roads, and all the rules are signaled by signs (there are no right-handed junctions). All cars follow the roads and keep the rules. They stop at red lights. They also stop at STOP signs and give way to other cars if they are near enough. The cars go in a random direction when they arrive at a junction. Cars do not bump into each other. Since the roads are too narrow, cars cannot take over each other, and this is true even for the robbers and the police. If a faster car catches up with a slower car, it also slows down, but when it can go freely, it speeds up again. The cars arrive from the edges of the city, and they vanish when they leave the city.
- The robbers and the police can break the rules (go through red light and stop signs). But they have to be careful not to hit another car. The robbers should avoid being caught by the police, before they reach their hiding place.

Functional requirements I.

Identifier	Requirement
R01	The robbers try to flee using a car
R02	There are cars on the road
R03	The police is after the robbers
R04	The roads in the town are all one-way roads
R05	All the rules are signaled by signs (there are no right-handed junctions)
R06	All cars follow the roads and keep the rules
R07	Cars stop at red lights
R08	Cars stop at STOP signs and give way to other cars if they are near enough
R09	Cars go in a random direction at a junction
R10	Cars do not bump into each other
R11	Cars cannot take over each other
R12	The robbers and the police cannot take over cars

Functional requirements II.

Identifier	Requirement
R13	If a faster car catches up with a slower car, it also slows down
R14	When a car can go freely, it speeds up again
R15	Cars arrive from the edges of the city
R16	Cars vanish when they leave the city
R17	The robbers and the police can break the rules (go through red light and stop signs)
R18	The robbers and the police have to be careful not to hit another car
R19	The robbers should avoid being caught by the police
R20	The robbers try to reach their hiding place

Additional requirements

Identifier	Requirement
R21	The city is a graph of roads
R22	The vertices of the graph are junctions
R23	The edges of the graph are roads
R24	The police and the robbers also follow the one-way roads
R25	Traffic light switches between incoming roads
R26	Hiding place is at an end of a road
R27	Robbers win if they reach their hiding place
R28	Robbers lose if they hit another car
R29	Robbers lose if they are caught by the police

CRC cards for Bank Robber

Entities

- Potential entities (nouns):
 - Town, Robber, Car, Road, Police, Player, Rule, Sign, Junction, Red light, Stop sign, Direction, Edge, City, Hiding place
- Synonyms:
 - Player = **Robber**
 - Town = **City**
 - Sign = **Rule**
 - Edge = **City**
 - Red light = **Traffic light**
- Final list of potential entities:
 - Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place
- Kind-of relationships:
 - Rule: Traffic light, Stop sign
 - Car: Robber, Police

R01. The robbers try to flee using a car

Responsibility: flee

Robber	
Flee	

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R02. There are cars on the road

Responsibility: road has cars

Road	
Has cars	Car

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R03. The police is after the robbers

Responsibility: police steps towards the robbers

Police	
Step	

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R04. The roads in the town are all one-way roads

R06. All cars follow the roads and keep the rules

R24. The police and the robbers also follow the one-way roads

Responsibilities:

cars move on the road only in one direction

cars have positions on the road

Car	
Move	Road
Has position	Position
Robber	
Move (instead of Flee)	
Position	
Has road	Road
Has distance	
Road	
Has cars	Car
Has length	

R05. All the rules are signaled by signs (there are no right-handed junctions)

Responsibilities:

roads meet at junctions

there are a signs at junctions

Junction	
Has sign	Sign
Has incoming roads	Road
Has outgoing roads	Road

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R07. Cars stop at red lights

Responsibility: cars can stop

Car	
Move	Road
Has position	Position
Stop	Red light
Traffic light	
Stop car	Car

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R08. Cars stop at STOP signs and give way to other cars if they are near enough

Responsibilities:
cars can stop
give way at stop sign

Car	
Move	Road
Has position	Position
Stop	Red light, Stop sign
Stop sign	
Stop car	Car, Road

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R09. Cars go in a random direction at a junction

Responsibility: cars move in a random direction at a junction

Car	
Move	Road
Has position	Position
Stop	Red light, Stop sign
Step	Junction

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R10. Cars do not bump into each other

R11. Cars cannot take over each other

R13. If a faster car catches up with a slower car, it also slows down

Responsibilities:

cars have speeds

a car has to slow down if there is another car in front of it

Car	
Move	Road
Has position	Position
Stop	Red light, Stop sign
Step	Junction
Has speed	
Slow down	Car

Road	
Has cars	Car
Has length	
Is car in range	Car

R11. The robbers and the police cannot take over cars

Responsibility: robbers and police cannot take over cars (nothing new)

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R14. When a car can go freely, it speeds up again

Responsibility: cars can speed up

Car	
Move	Road
Has position	Position
Stop	Red light, Stop sign
Step	Junction
Has speed	
Slow down	Car
Speed up	

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R15. Cars arrive from the edges of the city

Responsibility: cars arrive

City	
Has edges	Road
Create car	Road, Car

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R16. Cars vanish when they leave the city

Responsibility: cars vanish (nothing new)

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R17. The robbers and the police can break the rules (go through red light and stop signs)

Responsibility: robbers and police do not have to stop

Robber	
Move	Road
Stop	Traffic light, Stop sign

Police	
Move	Road
Step	
Stop	Traffic light, Stop sign

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R18. The robbers and the police have to be careful not to hit another car

Responsibilities:
robbers and police may hit other cars
when they do, they lose

Car	
Hit	Car
...	

Robber	
Move	Road
Stop	Traffic light, Stop sign

Police	
Move	Road
Step	
Stop	Traffic light, Stop sign

R19. The robbers should avoid being caught by the police

Responsibilities:
robbers and police may hit other cars
when they do, they lose

Robber	
Move	Road
Stop	Traffic light, Stop sign
HitBy	Police
Police	
Move	Road
Step	
Stop	Traffic light, Stop sign
Hit	Car

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R20. The robbers try to reach their hiding place

Responsibility: robbers can reach hiding place

Hiding place	
Reach	Robber

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R21. The city is a graph of roads

R22. The vertices of the graph are junctions

R23. The edges of the graph are roads

Responsibilities:
vertices are junctions
edges are roads

City	
Has edges	Road
Create car	Road, Car
Has roads	Road
Has junctions	Junction

R25. Traffic light switches between incoming roads

Responsibility: traffic light can switch

Traffic light	
Stop car	Car
Switch	

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R26. Hiding place is at a junction

Responsibility: hiding place is at a junction (nothing new)

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R27. Robbers win if they reach their hiding place

Responsibility: robbers can win

Hiding place	
Reach	Robber
Robber	
Move	Road
Stop	Traffic light, Stop sign
HitBy	Police
Win	Hiding place

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R28. Robbers lose if they hit another car

Responsibility: robbers can lose

Robber	
Move	Road
Stop	Traffic light, Stop sign
HitBy	Police
Win	Hiding place
Lose	Car

Car	
HitBy	Robber
...	

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

R29. Robbers lose if they are caught by the police

Responsibility: robbers lose if they are caught (=hit by) the police

Robber	
Move	Road
Stop	Traffic light, Stop sign
HitBy	Police
Win	Hiding place
Lose	Car, Police

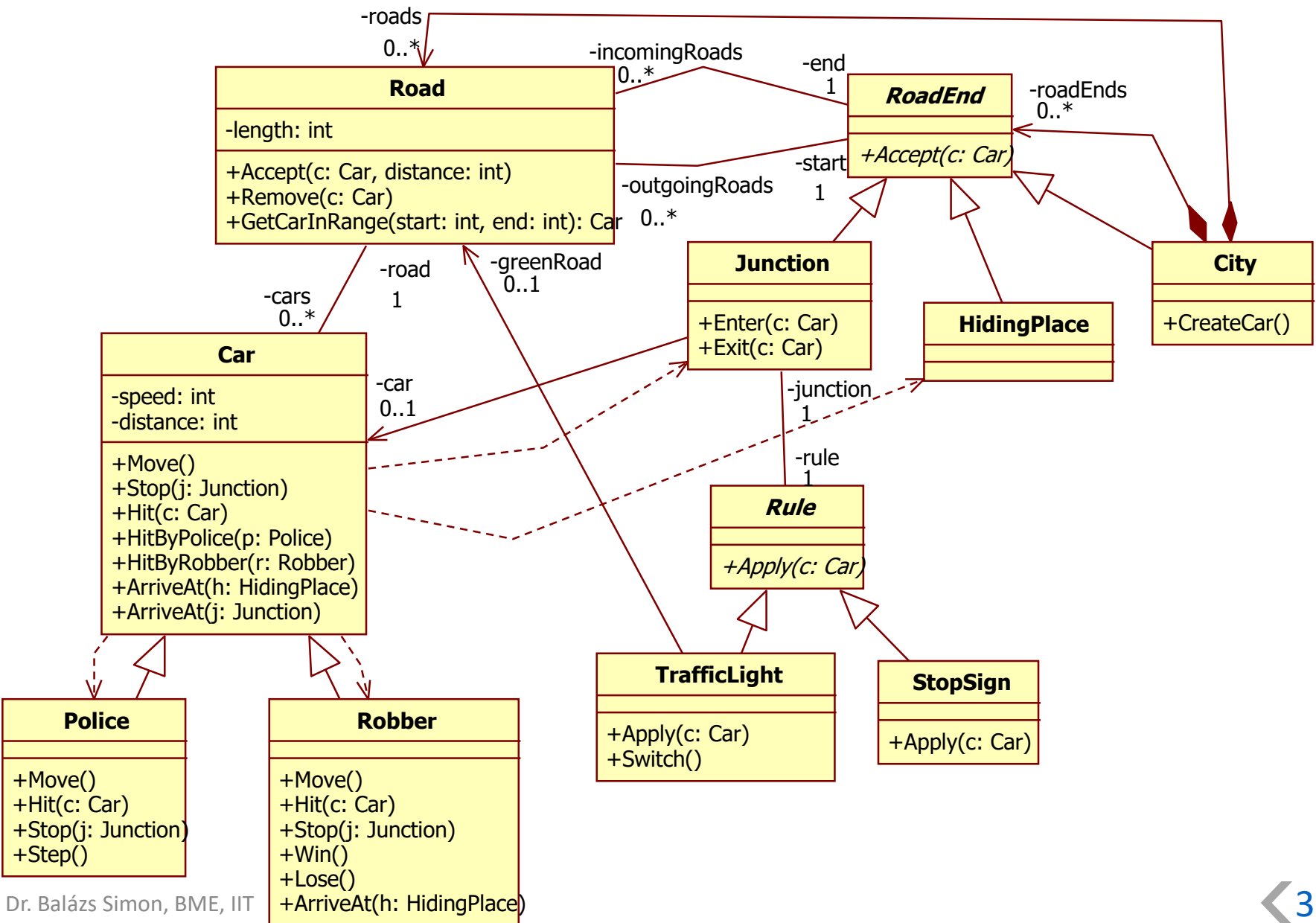
Police	
HitBy	Robber
...	

Entities:

Town, Robber, Car, Road, Police, Player, Rule, Junction, Traffic light, Stop sign, Direction, City, Hiding place

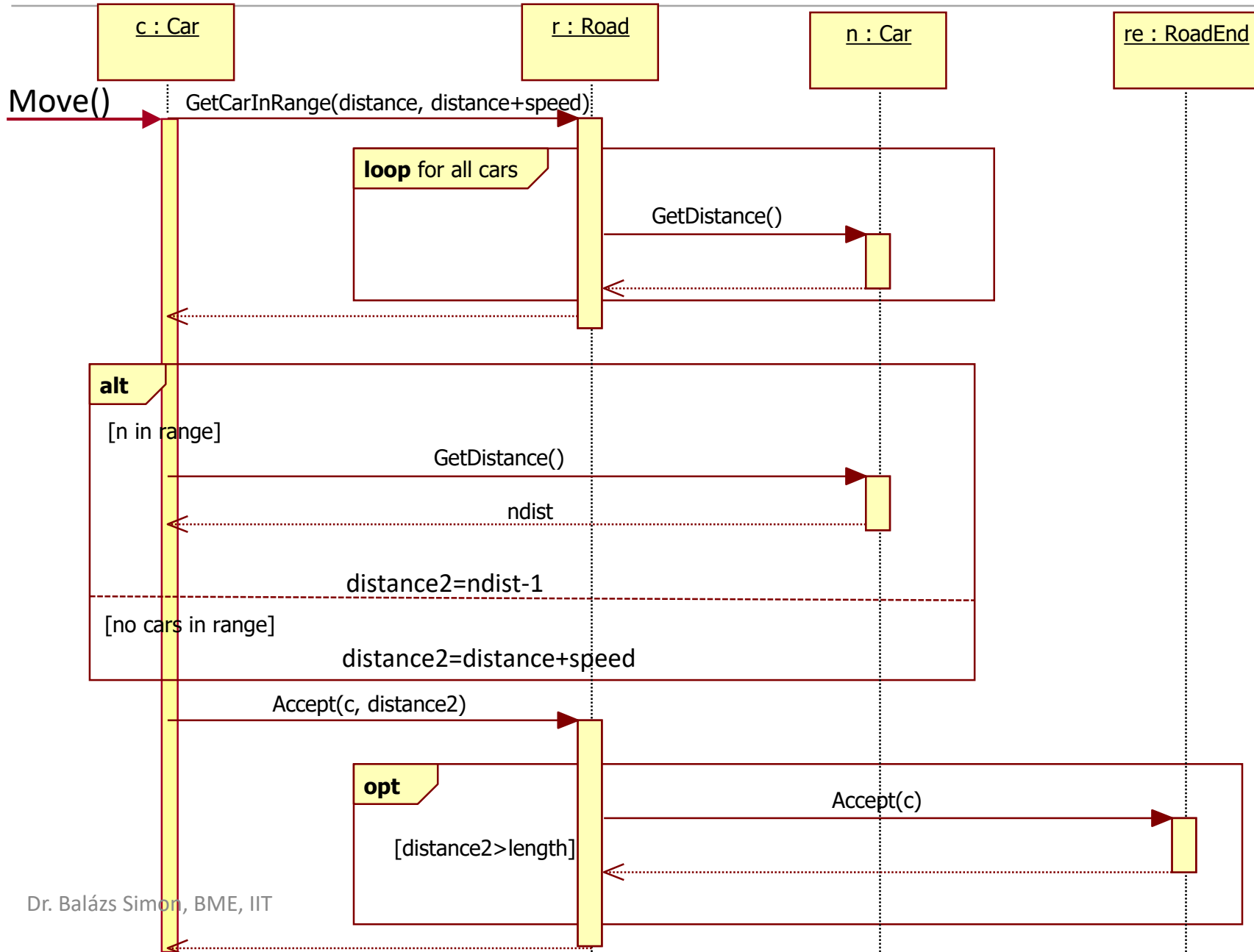
Class diagram for Bank Robber

Class diagram

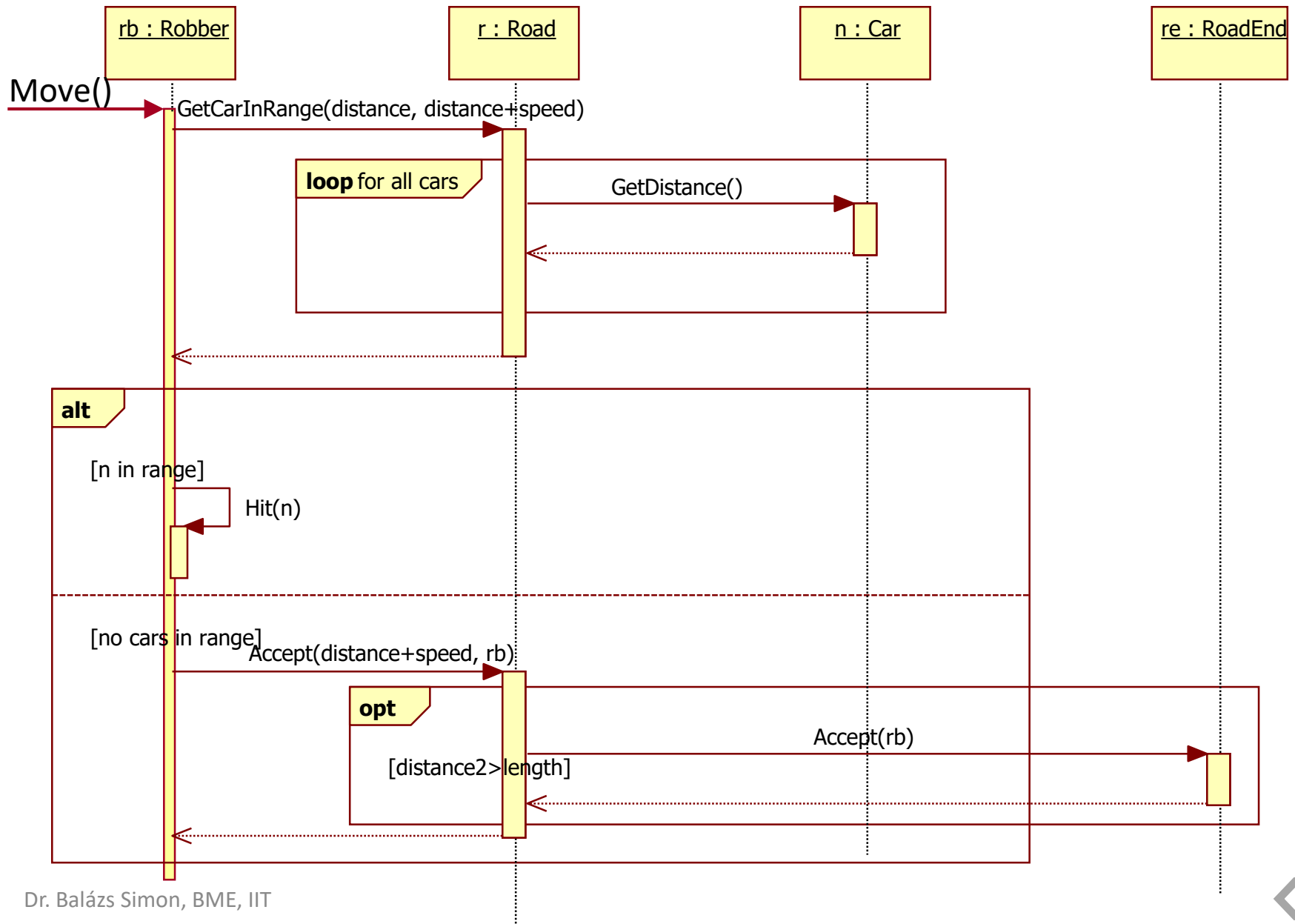


Sequence diagrams for Bank Robber

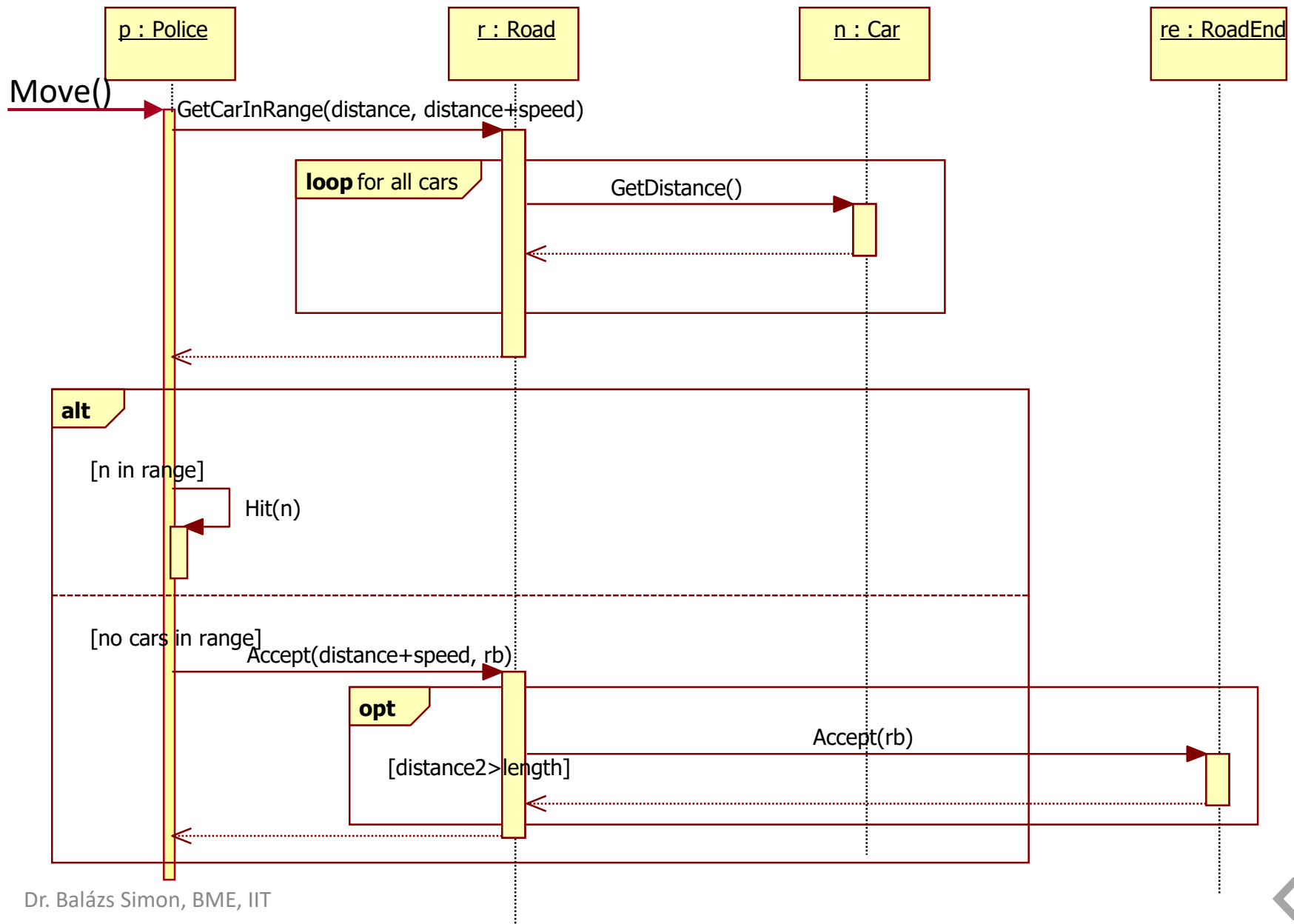
Car moves (Car.Move)



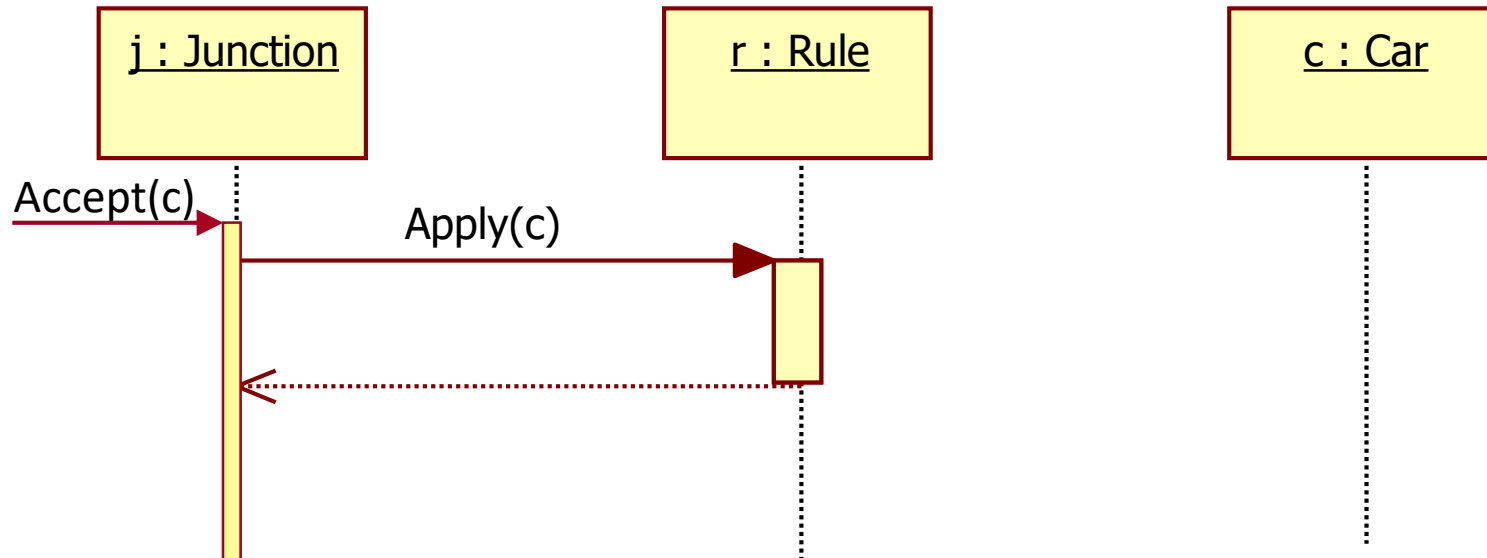
Robber moves (Robber.Move)



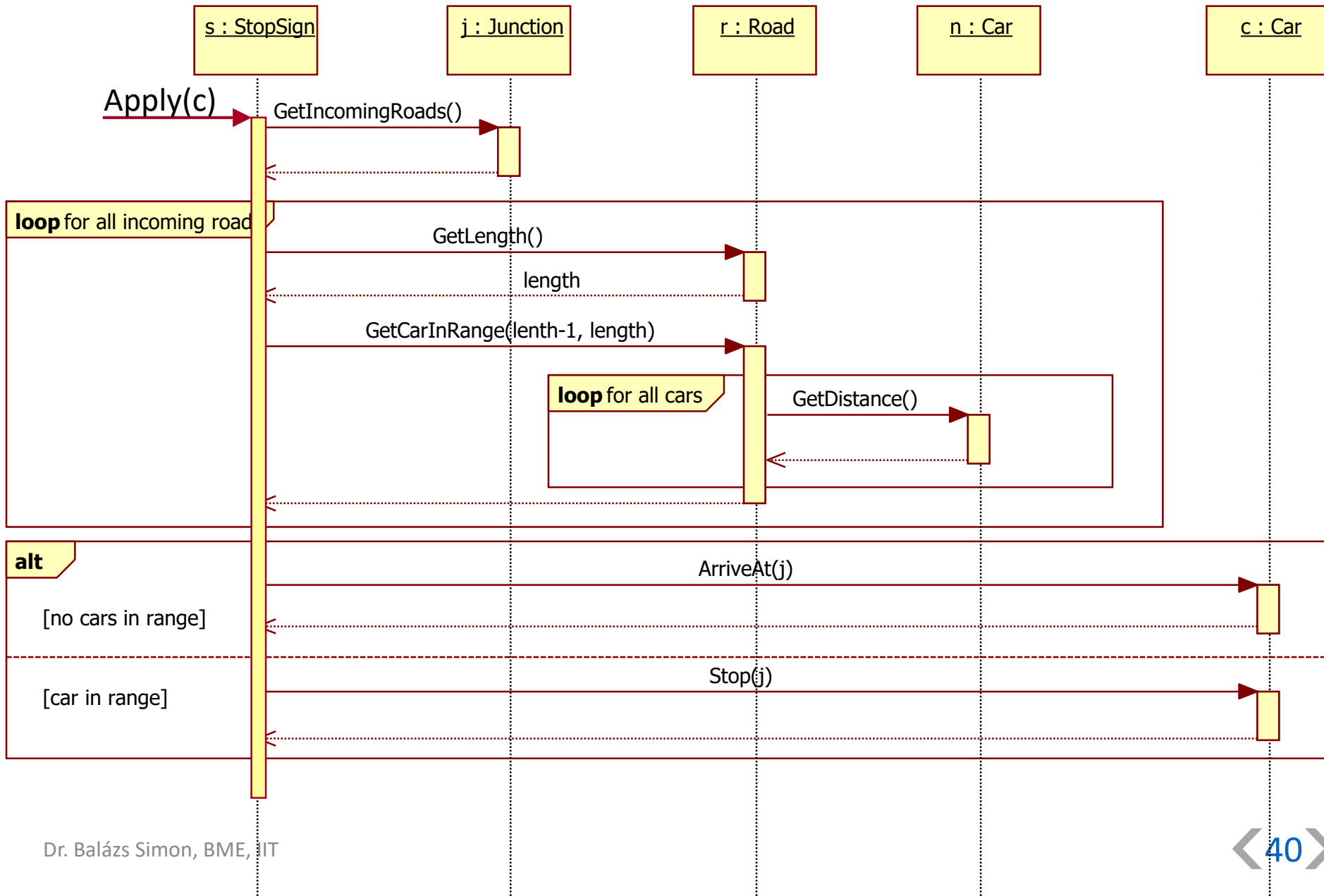
Police moves (Police.Move)



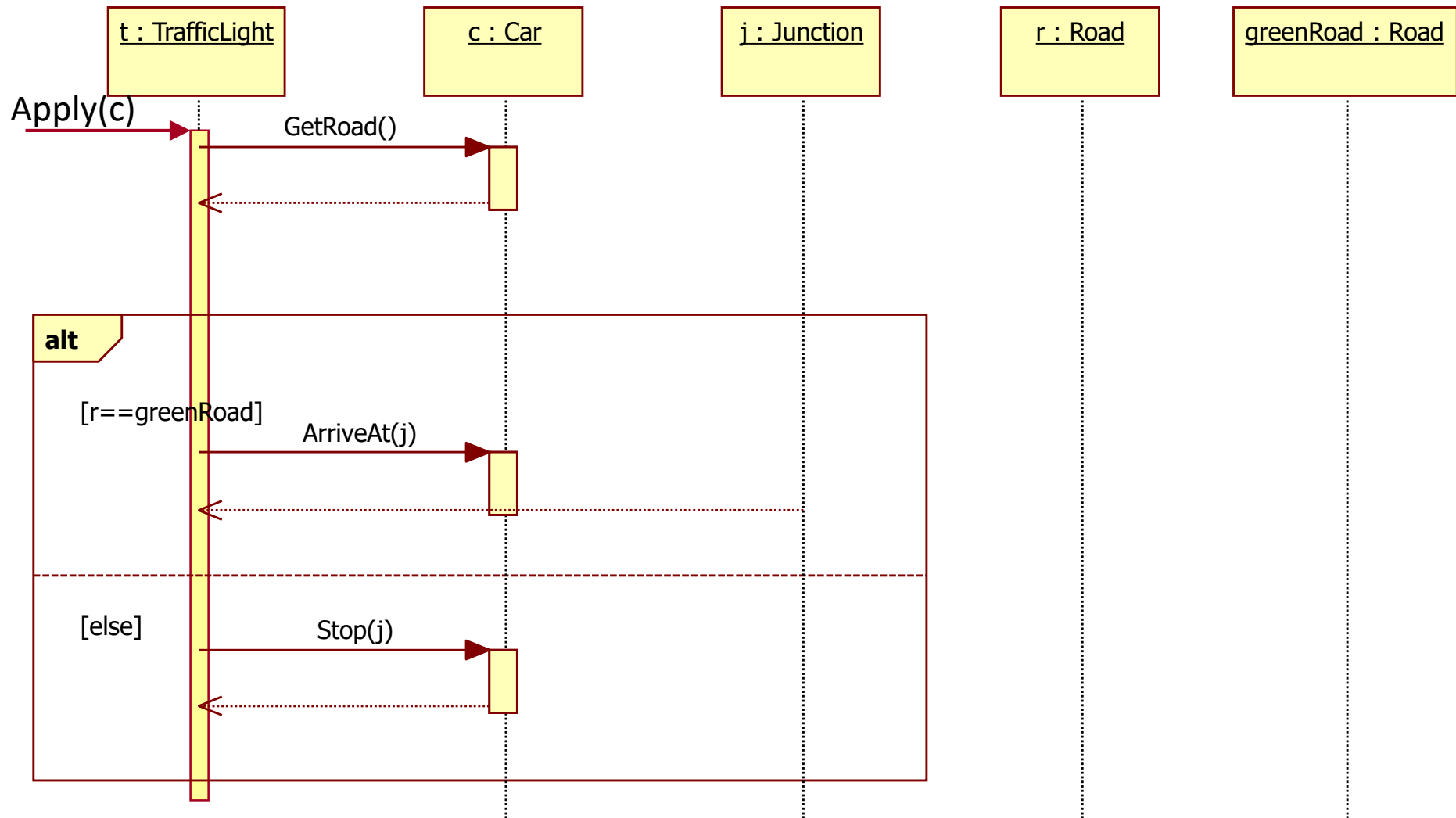
Car arrives at junction (Junction.Accept)



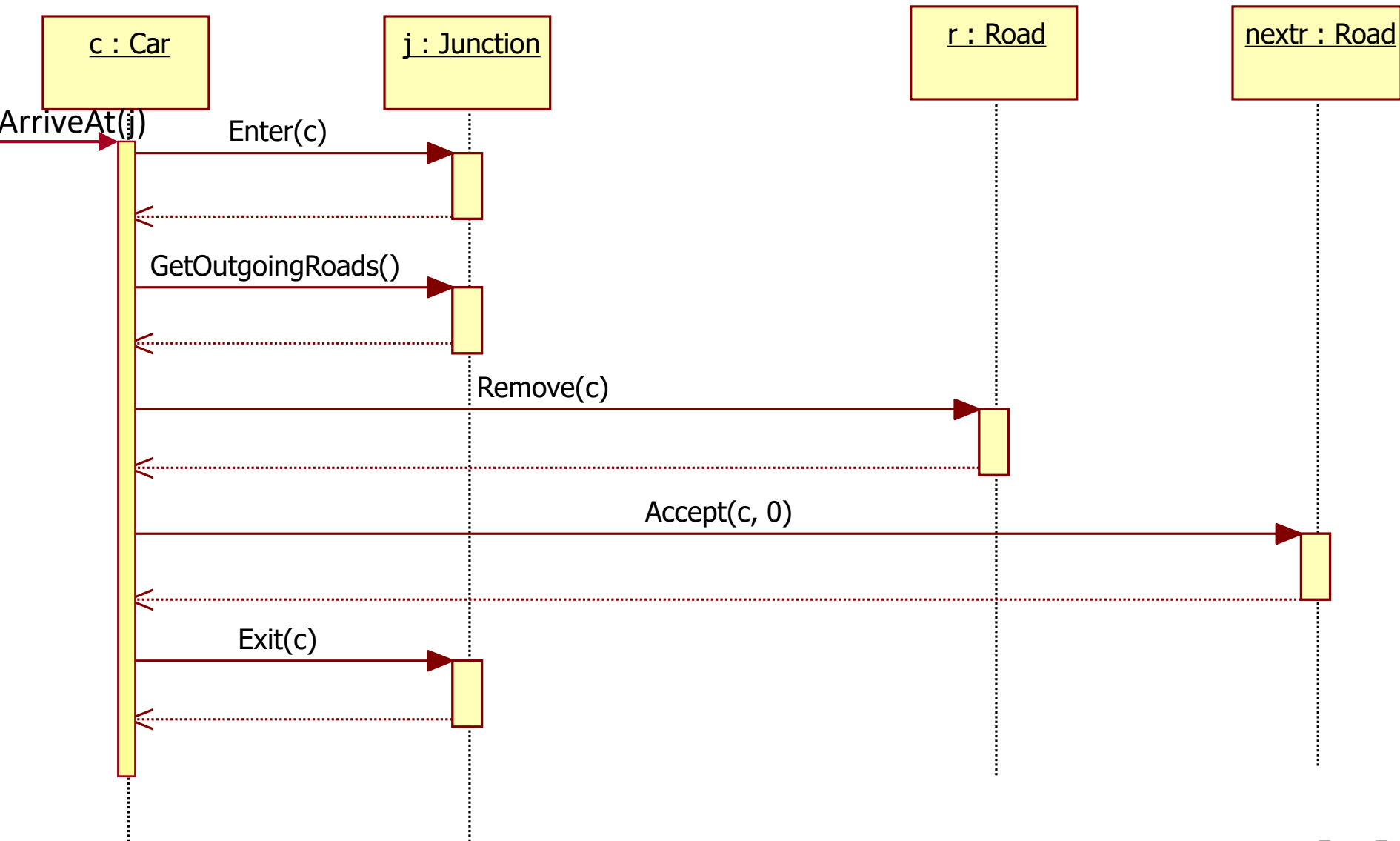
Car arrives at stop sign (StopSign.Apply)



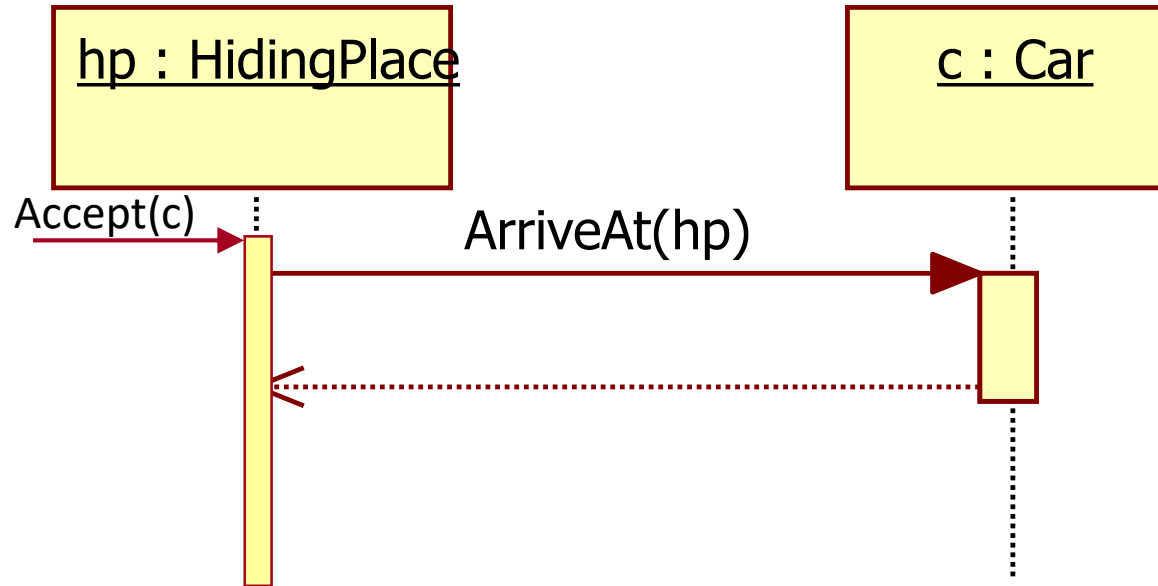
Car arrives at traffic light (TrafficLight.Apply)



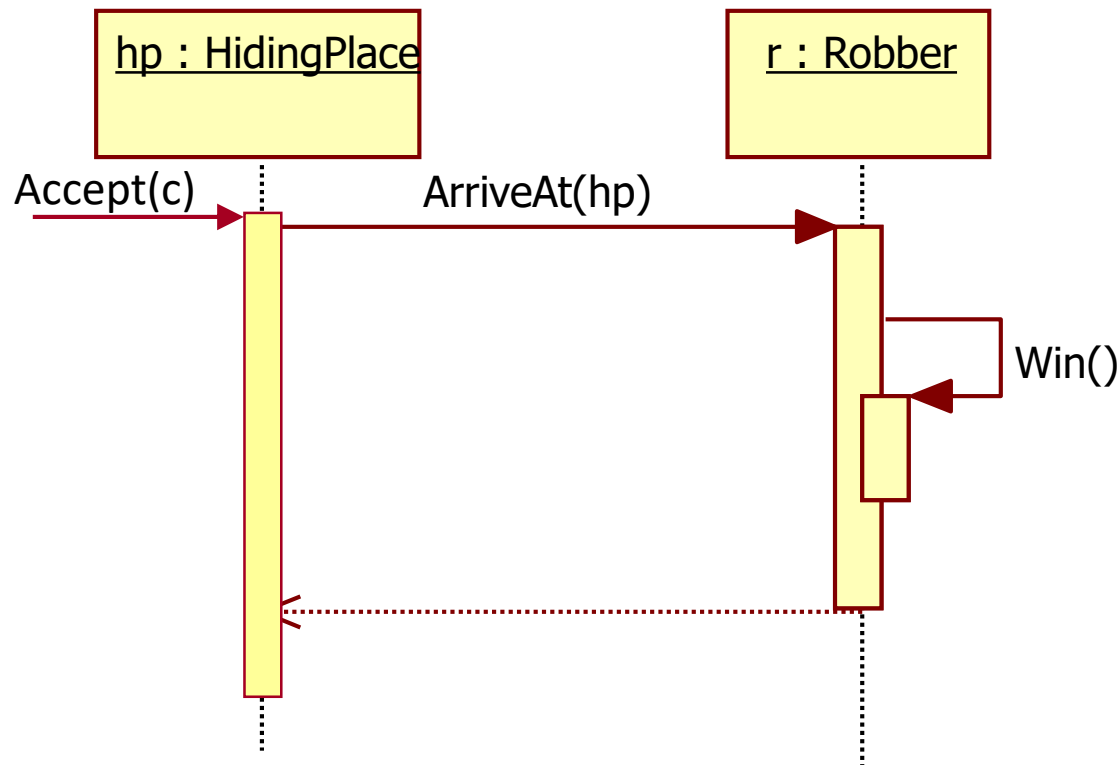
Car goes through junction (Car.ArriveAt)



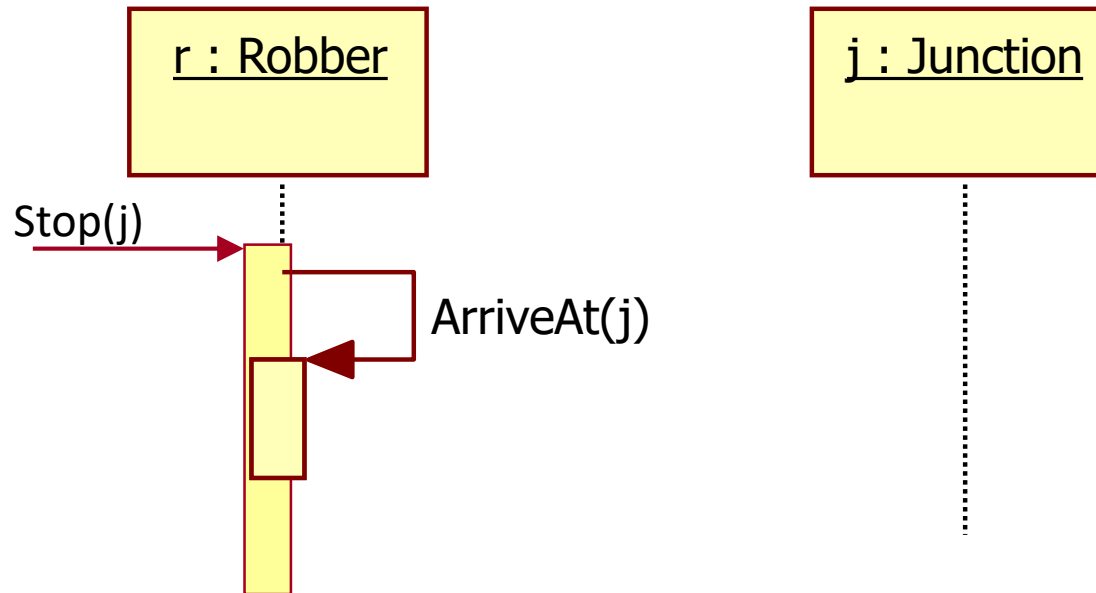
Car arrives at hiding place (HidingPlace.Accept)



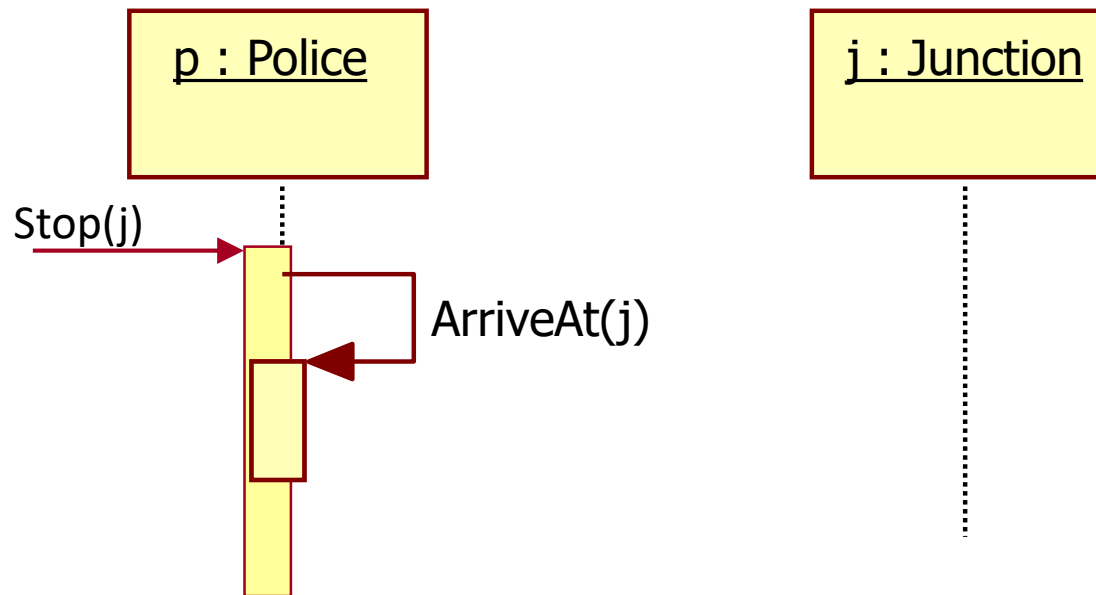
Robber arrives at hiding place (HidingPlace.Accept)



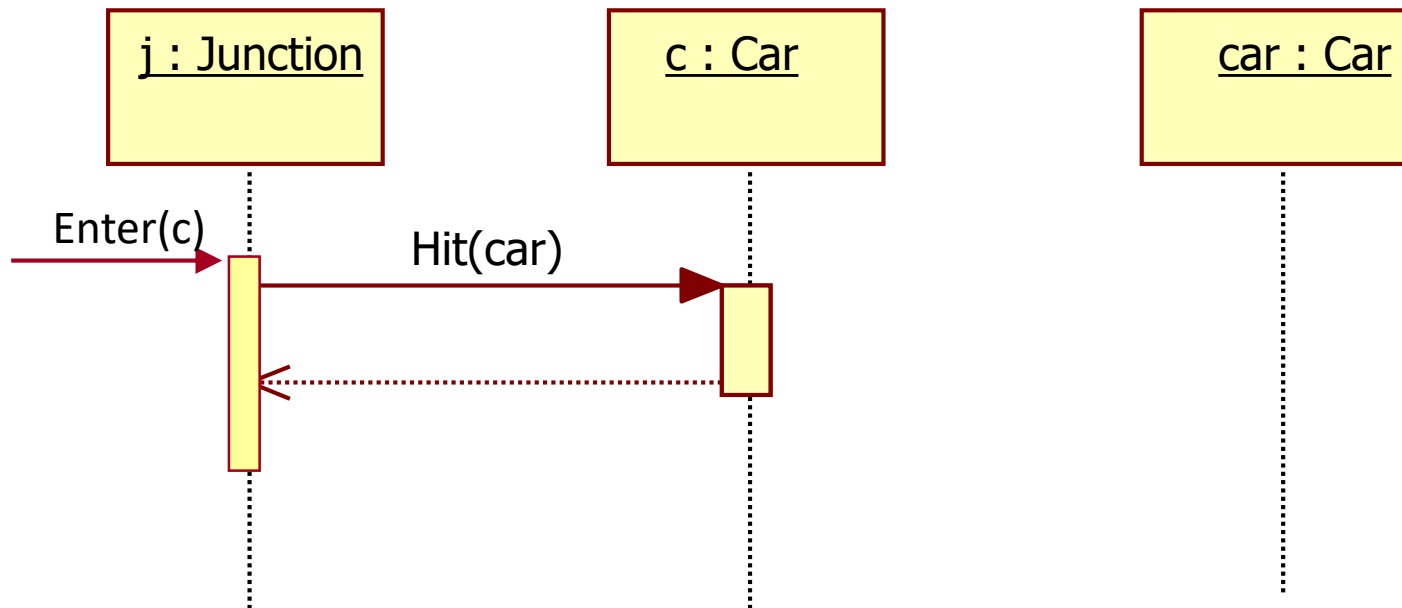
Robber does not stop (Robber.Stop)



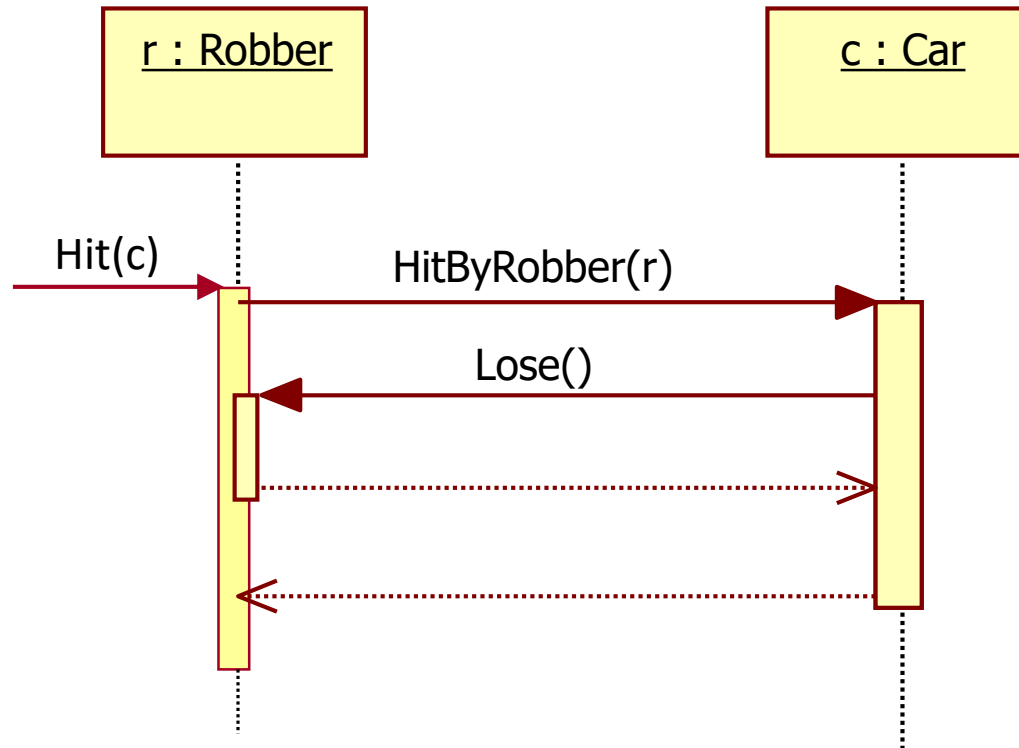
Police does not stop (Police.Stop)



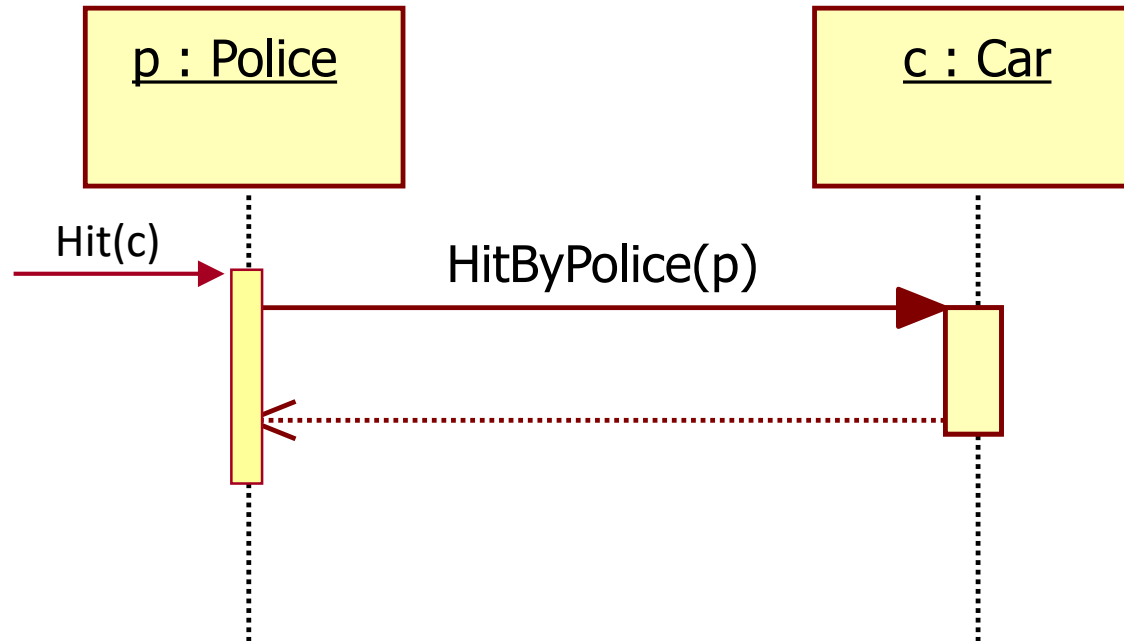
Cars collide at junction (Junction.Enter)



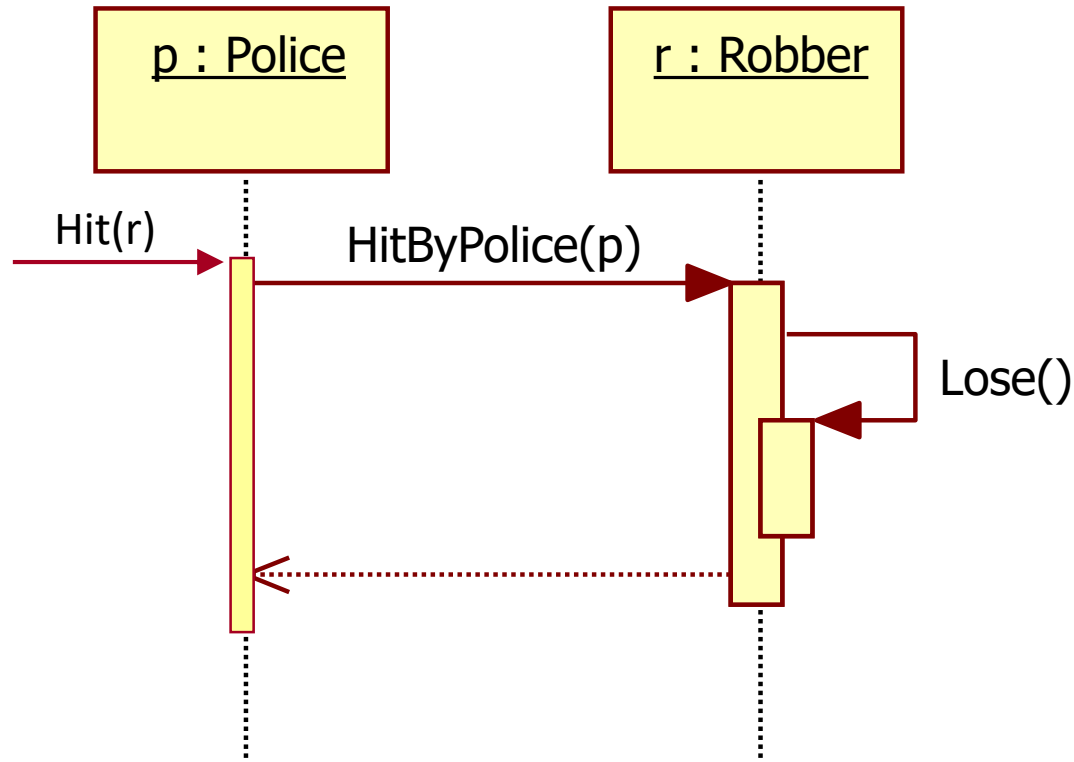
Robber collides with a car (Robber.Hit)



Police collides with a car (Police.Hit)



Police catches robber



City creates car

