Drop 2

By
Frode Jensen
Erik Nilsen Haga
Jenny Hougen

.

Part 1. System design

Design decisions
We have implemented BlindDate1 from INF5150Drop1Solution_v2.1, supplemented with a simplified make event functionality.

Assumptions
All assumptions made in INF5150Drop1Solution_v2.1 is retained, in addition we assume;
- No user will join more than one event at a time.
- No two events will have the same type.
- No more than 250 persons will sign up for our system.
- Event type does not contain white-space.

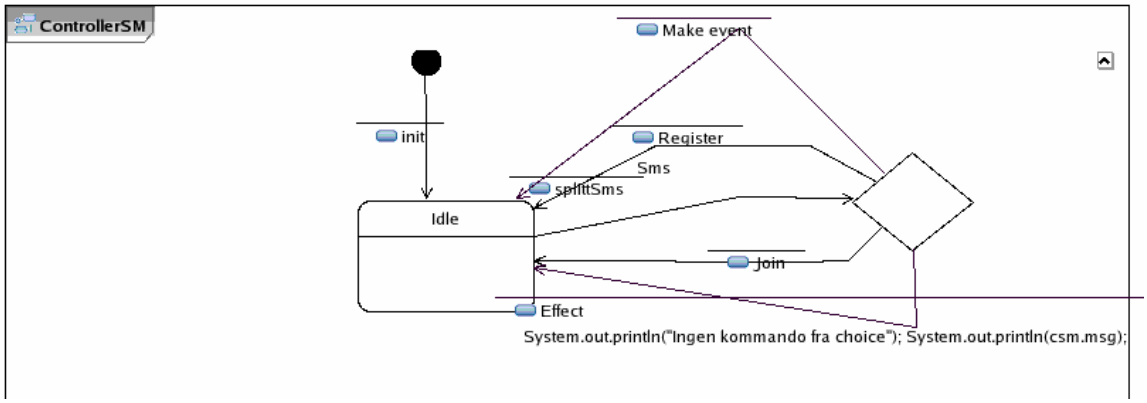
The system does not handle these assumptions in a robust manner.

Composite structures

The structure of our composites have not changed from BlindDate1.

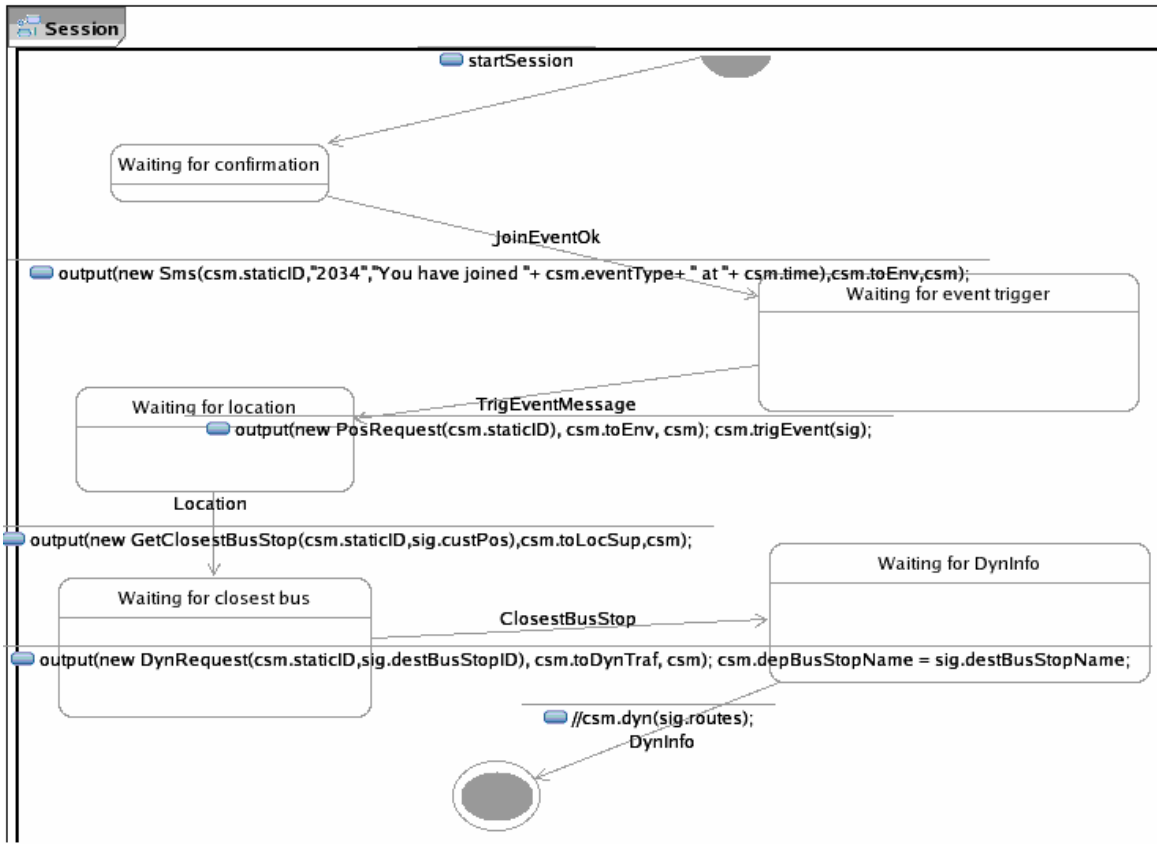
State-machines

ControllerSM
Is a Factory and Controller for sessions, that is it handles routing of messages to, and creation of, sessions. It also handles parsing and routing of incoming messages from the enviroment. This is a single-state machine in order to faciliate synchronisity.

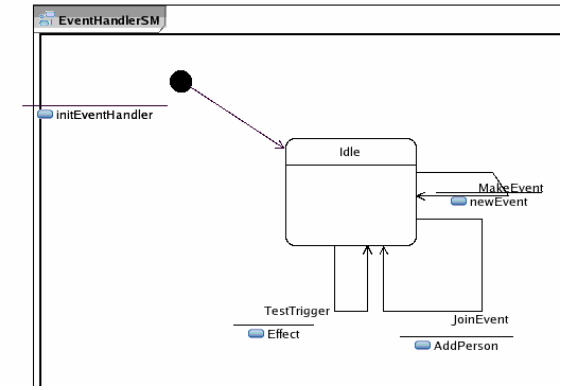


ControllerSM lives for the duration of the system run.

Session
Dynamicaly created state-machines. Handles communication to the enviroment for user-spesific messages. E.g. notification, getting routes from trafikanten and positioning of cellular phones.
A session lives from the user joins an event until notification is dispatched to the user.

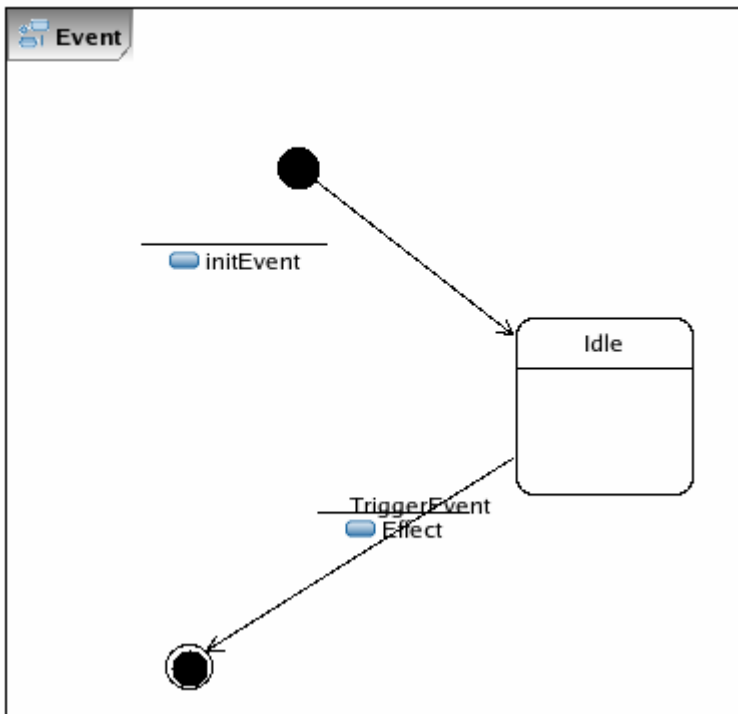


EventHandler
Is a Factory and controller for events, that is it handles routing of messages to, and creation of, events.
This is a single-state machine in order to faciliate synchronisity.

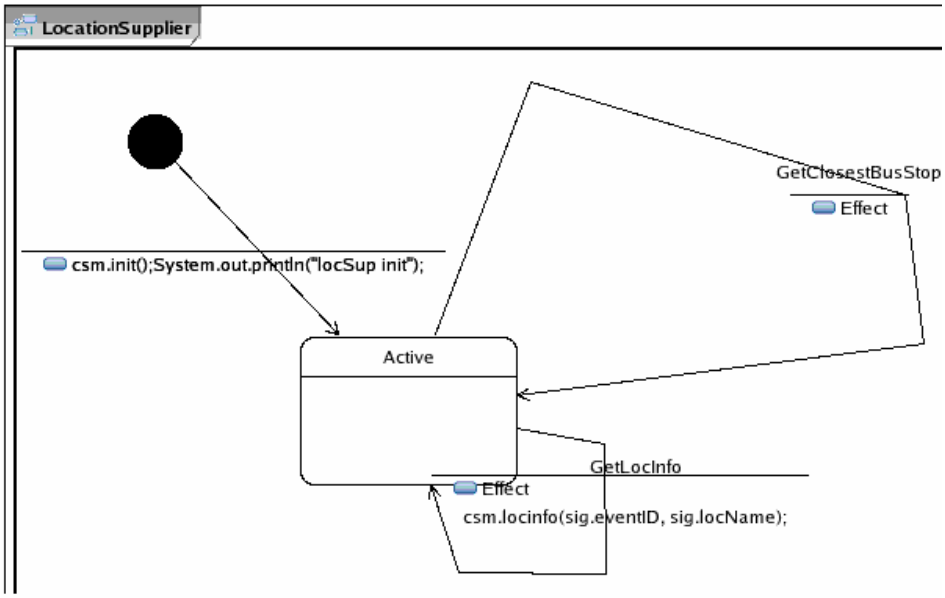


EvenHandler lives for the duration of the system run.

Event
Dynamically created state-machines. Handles creation of event-spesific messages. Keeps track of recivers of these.
An event lives from it is made until the event dispatches notifcations to the users signed up for it.



Location supplier
Transforms locations and positions to bus-stops.



Location supplier lives for the duration of the system run.

Other UML diagrams

For the other UML diagrams we refer to INF5150Drop1Solution_v2.1.

Drop2 as a refinement of BlindDate1.

This discussion bases itself upon the system as we have intended it, but we are not quite there yet.

BlindDate1 defines no negative traces, therfore our statemachines accept no traces that would be negaitve in BlindDate1.

BlindDate1 has 3 modes of interating with the system, registerCustomer, JoinEvent and NotifyUsers.

RegisterCustomer

This Use case is represented by two sms's in the Sequence diagrams, one for the user to register and one for the system to send confirmation. This corresponds to the sms transitions and csm.msg.startswith("register") guard in the controllerSM. With effects the wanted updates in the system and sends the confirmation to the user.

JoinEvent

This is represented by the sequence diagrams JoinEvent, BD_JoinEvent and BD_Controller_JoinEvent. The initial sms corresponds to the sms transition and csm.msg.startswith("join") guard in the controllerSM, which causes a new session to be created. Upon creation session will dispatch an JoinEvent message to EventHandlerSM, which will route this forward to the appropriate event. Upon recieving an AddPerson Event will dispatch an AddOk to EvenHandler, which will send it on to ControllerSM for there to be sent to the appropriate session. The session is at this stage in the waiting for confirmation state, and has a transition on AddOk, causing the session to dispatch a reciept to the user and going to the waiting for event trigger state. This matches the message flow in the JoinEvent Sequence diagrams.

NotifiyCustomers

This is represented by the sequence diagrams NotifyCustomers, BD_NotifyCustomers, and BD_Controller_Ncust. It is initialized by one or more TrigEventMessage from an event to controllerSM, this message is in our system caused by a message sent from the console.

Upon recieving a TrigEventMessage controllerSM will forward it to the appropriate session, which, since each user belongs to at most one event, is in the state waiting for event trigger. From this state TrigEventMessage causes a transition to waiting for location and a getlocation sent to PATS. Provided PATS return a Location the controller whill send this to the session we are considering, as the staticID functions as an identifier. The session will transit to waiting for closest bus stop and send a GetClosestBusStop to LocationSuplier. Location Supllier will iterate over all busstops and send closestBusStop with the closest bus stop to controllerSM using staticID to identify the session. Controller will pass this on to session and the session will transit to waiting for DynInfo as well as sending a Dyn request. The Dynrequest will be intercepted by controllerSM which will pass it to session, causing session to finish by sending route info to the customer.

The above matches one part of the par opperator. The other instances will run because event sends an TrigEventMessage for every person that has been added to it.

What went wrong?

JoinEvent causes a null pointer exception in the method forward of the Mediator class. We suspect that this is the error that is described in the FAQ, caused by wrong order of the ports, but fail to see how the solution suggested can be carried through, as the ownedPorts only list ports directly belonging to the composite, and not to the gates of the parts in said composite. We see no way to use this method to change the interal order of linked input/output ports as they will never be in the same ownedPorts list. A possibly fix is to delete all the ports an recreate them in a reasonable order, but with an hour left before deadline and some minor tweaks that has been deferred until we had a working system, we feel that we would not be able to finish this.

Part 2: Blind Date System security analysis

1. Context identification

The case for our analysis is The Blind Date System (BDS). It is a mobile system where persons could send an SMS to join an particular event.

The client for our security analysis is the system developer of BDS. He has asked us to find potential threats, both technical and environmental threats, to the system. In this chapter we are going to describe the system and its environment, the analysis context. We describes the target of evaluation, find the assets and a way to measure potential threats.

1.1. Target of analysis

This section describes the target of our analysis.

Target	The Blind Date System – A mobile application to meet groups of people.
Objective	Provide a service for meeting people.
Service/Function	- Register customer - Join event - Notify customer
Security aspects	Data integrity, availability

Tab1e Target of evaluation

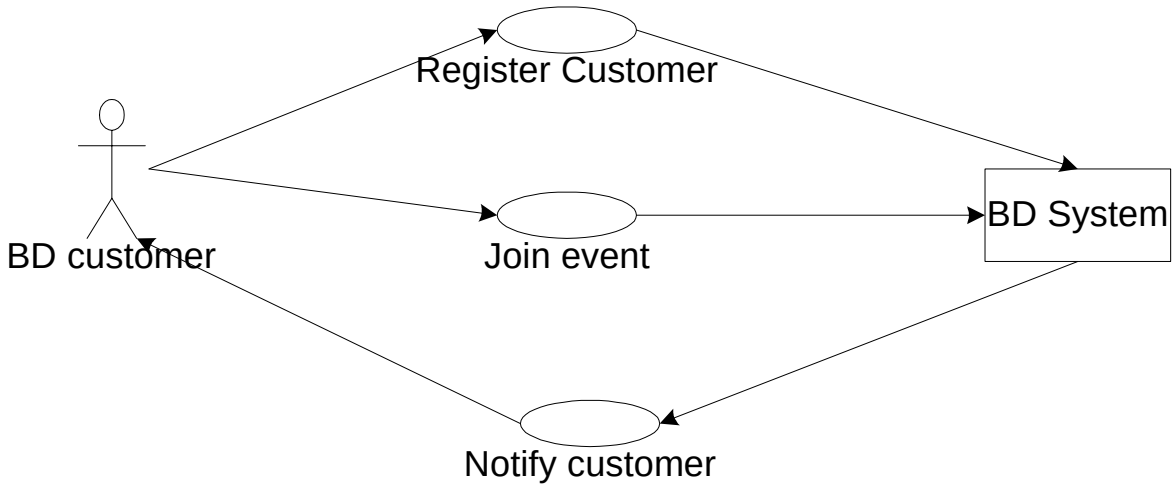


Figure 1 Use case of BD system

Figure 1 gives an overview of the functionalities in the BD system. It is possible for the customer to register himself with phone number and name. He could also join a particular event by sending an SMS with type and time for event. The system will send SMS, notify customer, to al user that are signed up for a particular event. All customer communication with the system goes through SMS.

1.2. Value definition

To be able to measure our assets and risks we need to suggest different values to use. These value types are filled inn the value of definition table. The different types in this table are described in their own tables.

Type	Qualitative/Quantitative	Domain	Allowed values
Frequency	Qualitative	Occurence/time	Very rare, rare, uncommon ,common, usual
Consequence	Qualitative	NOK	Neglible, light, moderate, high severe
Asset	Qualitative	NOK	Low, medium, high

Table 2 Value definition table

Table 2 gives an overview of the value we use for this analysis. Table 3, 4 and 5 gives more detailed information about each value.

Asset value type	Description
Low	Less than 1000 NOK
Medium	1000 – 10 000 NOK
High	More than 10 000 NOK

Table 3 Asset value description table

Frequency value type	Description
Very rare	Not expected more than once every second year
Rare	Expected between twice a year and once every two years
Uncommon	Expected between once a week and twice a year
Common	Expected between once a day and once a week
Usual	Expected at least every day

Tabel 4 Frequency value description table

Consequence value type	Description
Neglible	Less than 10 NOK
Light	10 – 100 NOK
Moderate	100 – 700 NOK
High	700 – 2000 NOK
Severe	More than 2000 NOK

Tabel 5 Frequency value description table

1.3. Assets

In this section we find the assets in the BD system. The assets will get an ID and a value (as defined in table 3). The asset value represents how important an asset is.

Asset ID	Asset	Asset value
A1	Customer repudation	Medium
A2	BD Income	High
A3	Personal information (phone nr)	Low
A4	BD System (system functionalities)	High

Table 6 Asset table

1.4. Risk Evaluation

We fill in risk evaluation table to be able to estimate the loss of asset we are willing to accept.

Criteria ID	Asset ID	Description
C1	A1	Ok if risk value <= Low
C2	A2	Ok if risk value <= Low
C3	A3	Ok if risk value <= Low
C4	A4	Ok if risk value <= Low

Table 7 Risk evaluation criteria table

2. Risk identification

In this chapter we are going to identify potential threat in the BD system. To find risks we need to identify threats, vulnerabilities and unwanted incidents. This is done by doing a HazOp (hazard and operability) analysis.

2.1 HazOp analysis

We started our risk identification by doing an 15 minute long brain storming where the users, system developers and security experts mention everything they could think of from the word security and risks. From the list of words we sorted out relevant words. From these words, we came up with the threats and unwanted incidents.

- Hacking
- Virus/worm
- Data confidentiality
- Integrity
- Availability
- Cracking
- Intruder
- Different access point
- System failure
- Hijack attack
- Not available resources
- Hash/encryption
- Spam
- Trust
- Repudiation
- The law
- SQL injection
- Performance
- Consistent error messages
- Interference

With these words in our mind we went through the sequence diagrams for the BD system and created the HacOp table. The HazOp table gives an overview of potential risks in the system.Based on the HazOp table we create threat diagrams.

Risk ID	Asset ID	Diagram reference	Attribute	Scenario
R1	A2	Figure 1	Availability	System not available

R2	A1	Figure 1	Spamming	SMS spamming
R3	A3	Figure 1	Confidentiality	Information tapping
R4	A1	Figure 1	Integrity	Message modification
R5	A1	Figure 1	System failure	Location changed
R6	A1	Figure 1	Delay	Delayed message
R7	A4	Figure 1	Sytem failure	Loss of data

Table 8 HazOp table

Figur 1 Threat diagram

3. Risk level estimation

In this chapter we give an estimation of the risks identified in chapter 2. We evaluate the frequency and consequence of the unwanted incidents.

3.1 Consequence and frequency table

Risk ID	Consequence value	Consequence rationale	Frequancy value	Frequency rationale

Risk ID	Consequence value	Consequence rationale	Frequancy value	Frequency rationale
R1	Light	We will just have to reboot. Should not affect our cash flow too much.	Common	System is not robust
R2	Severe	Spam is annoying	Common	Availiable Internet access.
R3	Light	Little confindential information stored	Rare	Limited output of system
R4	High	Could turn customers avay permanently	Rare	Rather difficult.
R5	Moderate	Can cause annoyance among customers	Very Rare	Can't really see how this should occur.
R6	Light	Should usually not be a big deal	Common	TTT
R7	Moderate	Is a hassle to restore	Rare	Data generally does not go poof.

Table 9 Consequence and frequency table

4. Risk evaluation

In this chapter we determine what risks that need treatment. It is done by identify the level of risk associated wit the unwanted incidents and decide whether the level is acceptable.

4.1 Risk estimation

Frequency Consequence	Very rare	Rare	Uncommon	Common	Usual
neglible	Safe	Safe	Safe	Light	Light
Light	Safe	Safe	Light	Moderate	High
moderate	Light	Light	Moderate	High	Critical
High	Light	Moderate	High	Critical	Ekstreme
severe	Moderate	Moderate	High	Critical	Extreme

Tabel 10 Risk matrix

We will treat all risks that are moderate, high, critical and extreme. That gives us following risks to treat:

- R1: System is not available
- R2: SMS spamming
- R4: Message modification
- R6: Delayed message

5. Risk treatment

This chapter is a out address the treatment of the identified risks and how to prevent the unacceptable risks. Based on the risk matrix in chapter 4 we suggest a way to treat those risks.

5.1 Risk treatment diagrams

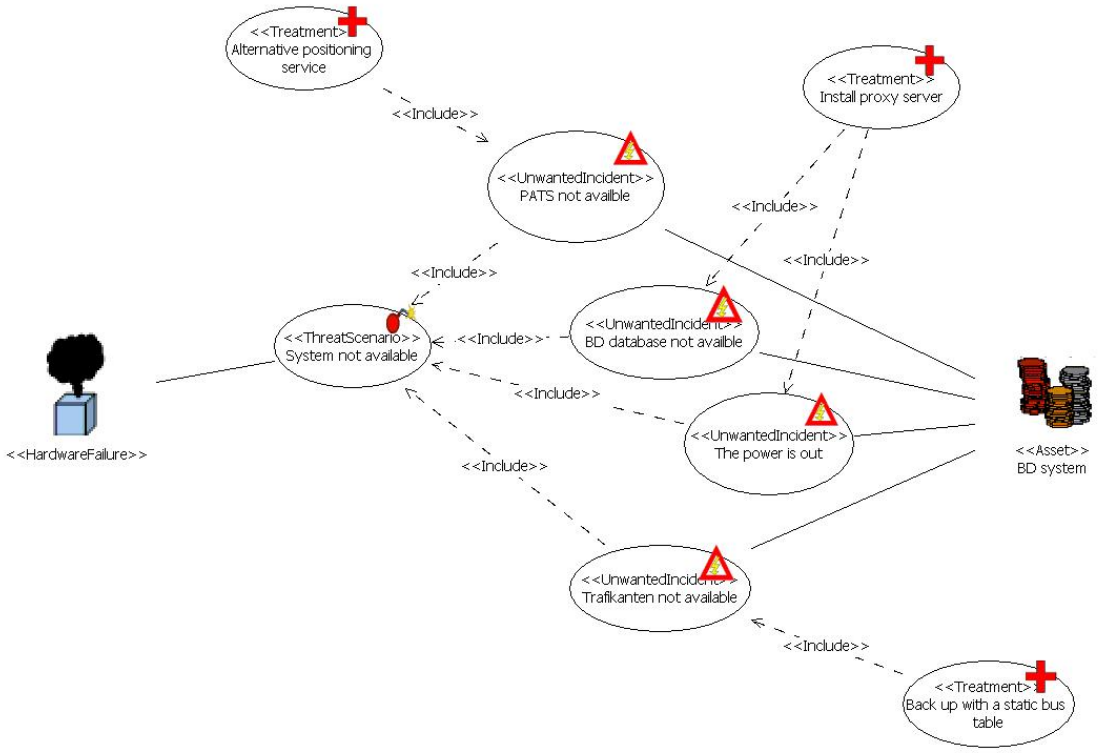


Figure 2 R1: system not available

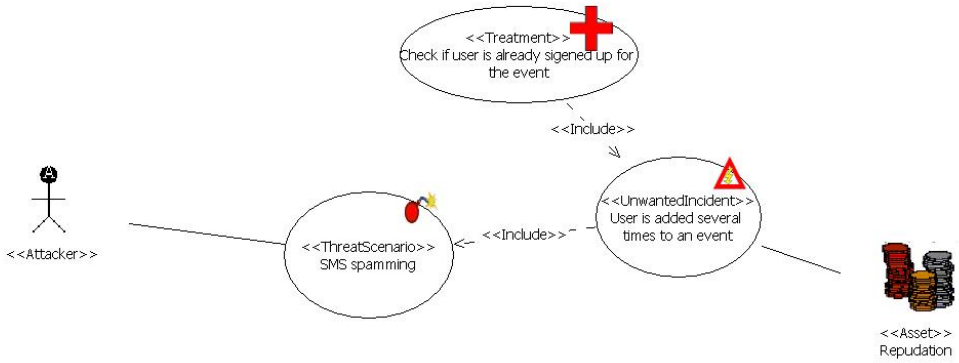


Figure 3 R2: SMS spamming

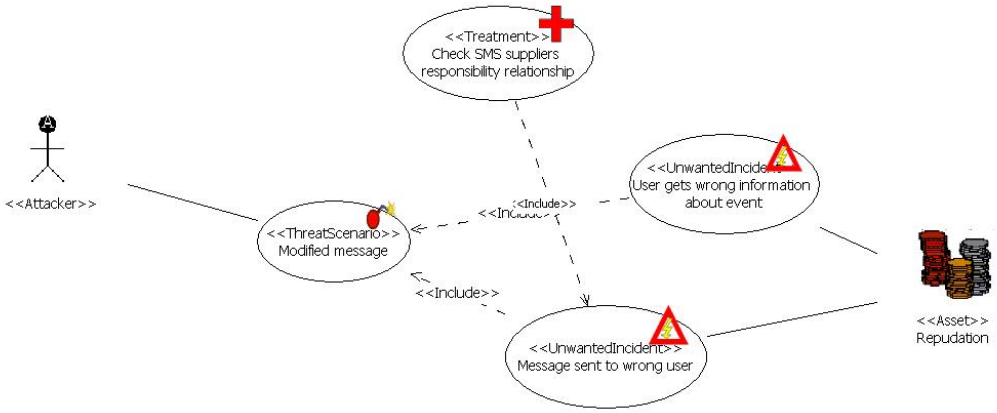


Figure 4 R4: Modified message

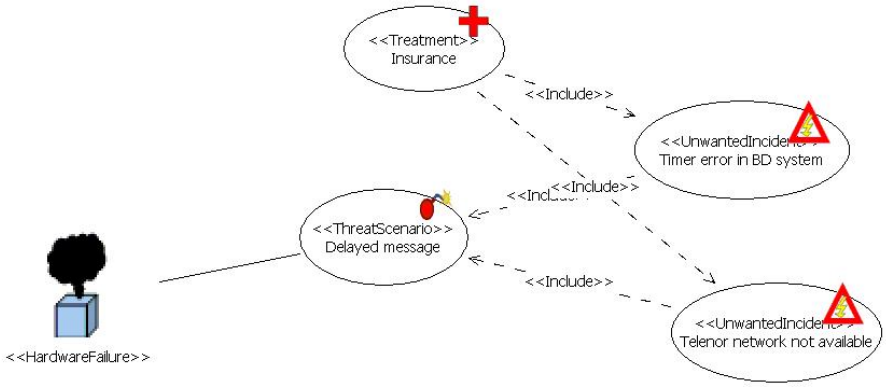


Figure 5 R6: Delayed message