**4. Formal analysis using Alloy**

In this section, the Alloy model is given. Using it, some of the features of the system are specified and

explained in more details, with the focus being on the constraints:

Data4Help Constraints

* If status of request is “Received” no response should be created
* If request is approved or rejected there should be one response for this request. Status of response depends on the status of request.
* When request is approved response data should be the same with requested data
* Every response should be specific to a request.
* There can’t be more than one different user which has the same data record
* Every request should be peculiar to one third party.
* When a request is sent by a pre-confirmed third party, it should be approved, and requested data should be sent the third party.

Track4Run Constraints

* User can’t attend an organization as both runner and spectator.
* In order to attend runner individual must activate Track4Run. Also, third party must activate Track4Run service to organize a run.
* Organization must access the location data of it’s all runners.

AutomatedSOS

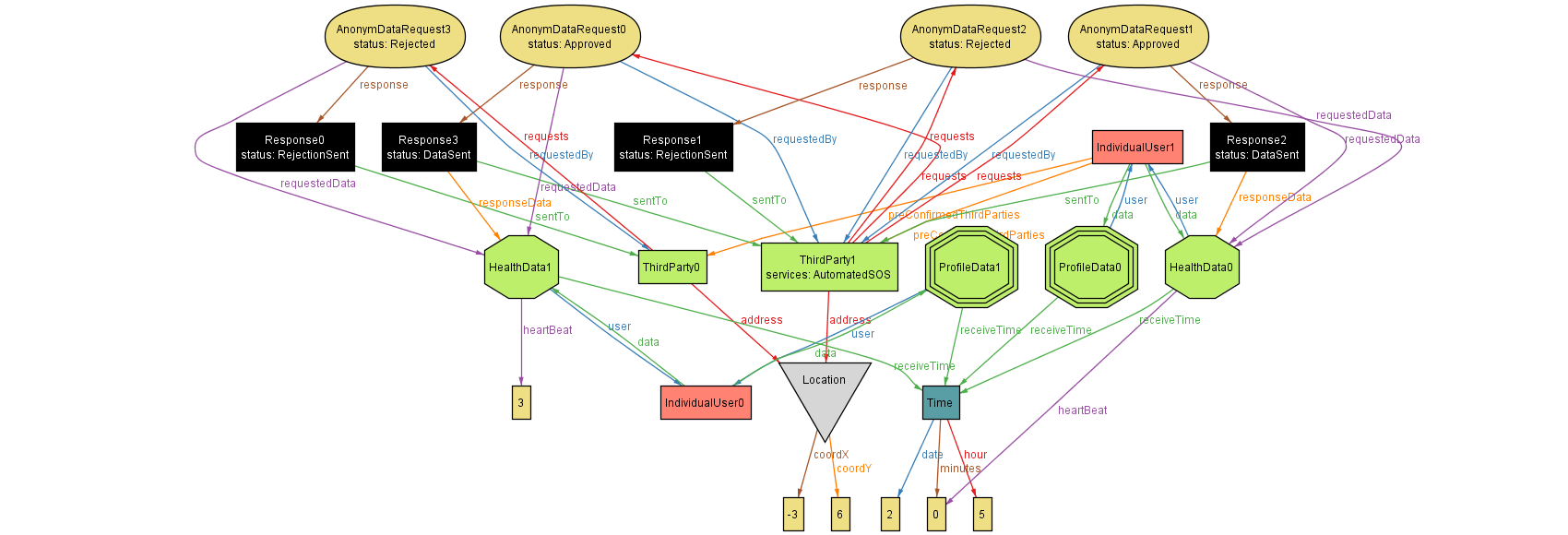
* Each AutomatedSOS user should have health and location data
* When health values are out of range a notification should be sent to a third party which serves AutomatedSOS service.
* For every exceed of threshold values one notification must be sent

It should be noted that in order to simplify modeling request time and respond time is excluded from modelling. Also, for monitoring health data we only model heartbeat. Heartbeat is modeled between 0 and 20 beats per minutes. Normal values are assumed as between 10 and 15 beats per minutes.

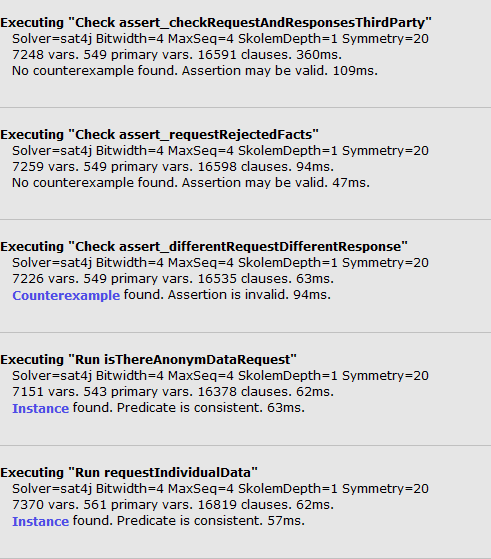
In order to increase readability of diagram we show three different model instances. First one focuses on Data4Help request response relation. The second one focuses on only Track4Run and the last one focuses on AutomatedSOS.

**Data4Help**

**Model**

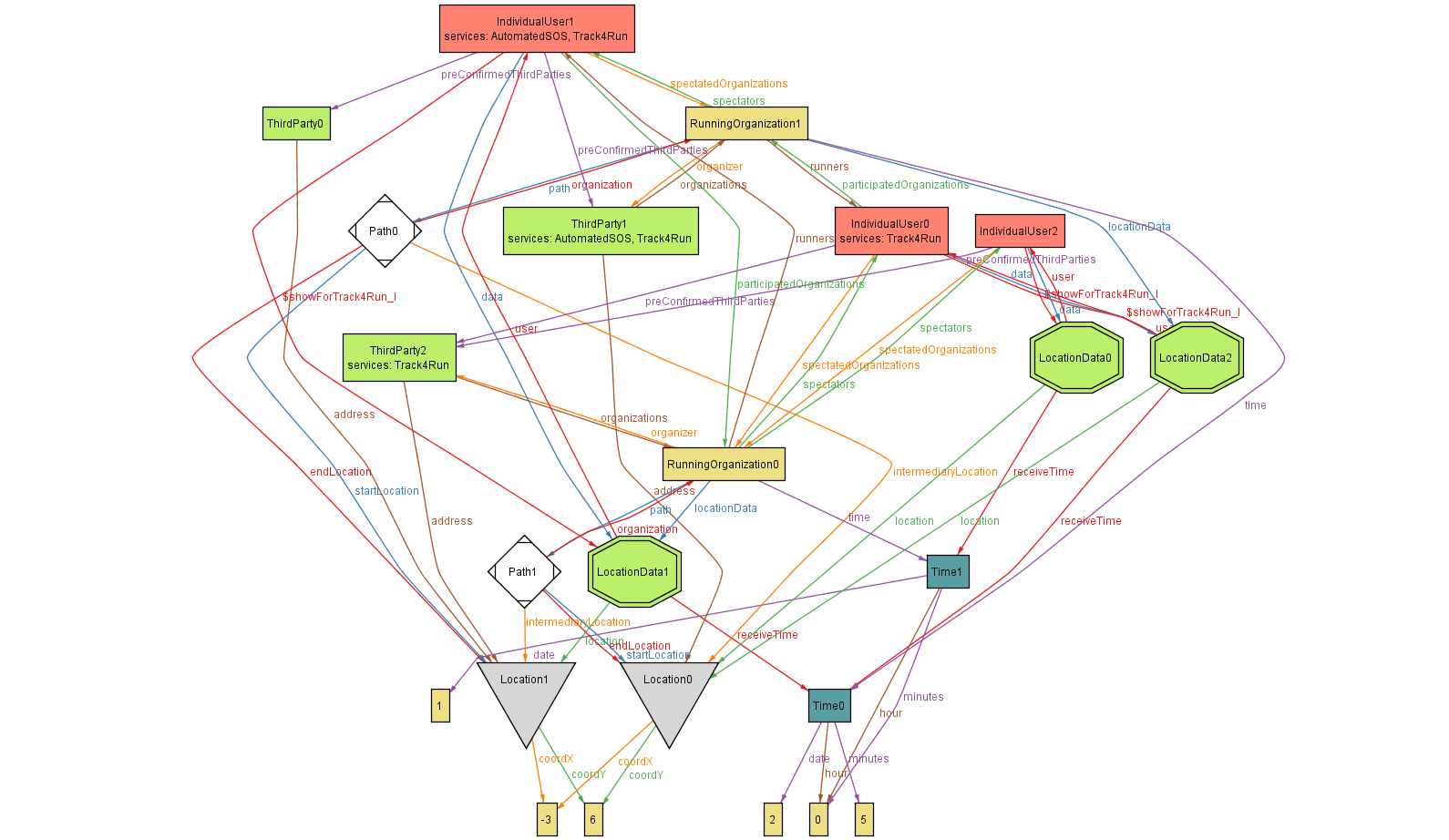


**Asserts**

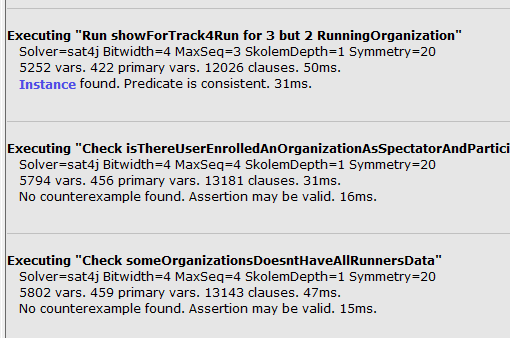
****

**Track4Run**

**Model**

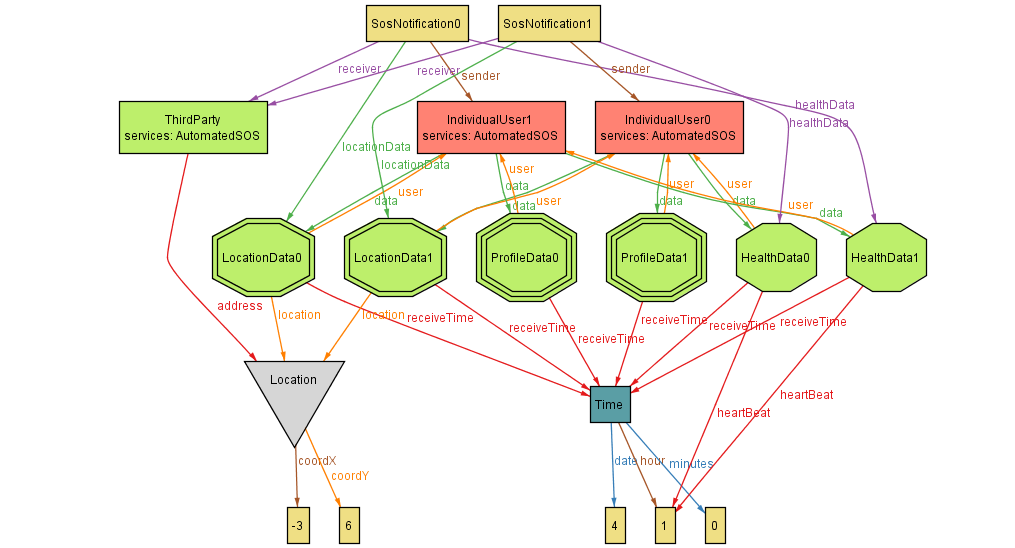


**Asserts**

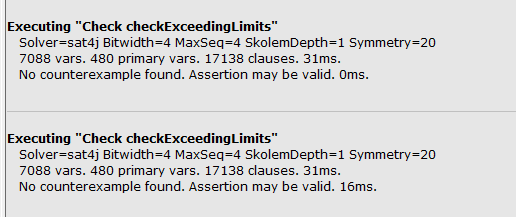


**AutomatedSOS**

**Model**



**Asserts**



**4.1. Alloy model**

**open** util**/integer**

sig **Time{**

**date:** one **Int,**

**hour:** one **Int,**

minutes**:** one **Int**

**}**

**{** **hour** **>=**0 **and** **hour<=**5 **and**

minutes **>=**0 **and** minutes**<=**5 **and**

**date>=**1 **and** **date** **<=**2

**}**

sig Location **{**

coordX**:**one **Int,**

coordY**:**one **Int**

**}**

--{ coordX >= -90 and coordX <= 90 and coordY >= -180 and coordY <= 180 }

**{**coordX **>=**-3 **and** coordX**<=** -3 **and** coordY **>=**-6 **and** coordY**=**6 **}**

abstract sig **Data{**

**user:** one IndividualUser**,**

receiveTime**:**one **Time**

**}**

**{**

this **in** **user.data**

**}**

sig LocationData **extends** **Data{**

location**:**one Location

**}**

sig ProfileData **extends** **Data** **{**

**}**

sig HealthData **extends** **Data{**

heartBeat**:** one **Int**

**}**

**{**

heartBeat**>=**0 **and** heartBeat**<=**20

**}**

sig ChronicDisease **extends** **Data{**

drugs**:set** UsedDrugs

**}**

sig UsedDrugs**{**

dosePerDay**:**one **Int**

**}**

**{**dosePerDay **>**0**}**

sig IndividualUser**{**

**data:some** **Data,**

services**:** **set** Service**,**

preConfirmedThirdParties**:** **set** ThirdParty**,**

receivedRequests**:** **set** IndividualDataRequest**,**

participatedOrganizations**:set** RunningOrganization**,**

spectatedOrganizations**:** **set** RunningOrganization

**}**

sig ThirdParty**{**

requests**:** **set** Request**,**

services**:** **set** Service**,**

organizations**:** **set** RunningOrganization**,**

address**:** one Location

**}**

abstract sig Service**{}**

one sig AutomatedSOS **extends** Service**{}**

one sig Track4Run **extends** Service**{}**

one sig RunningOrganization**{**

organizer**:** one ThirdParty**,**

runners**:** **some** IndividualUser**,**

spectators**:** **set** IndividualUser**,**

**path:** one **Path,**

**time:** one **Time,**

locationData**:some** LocationData

**}**

**{** this **in** organizer**.**organizations

**}**

sig **Path{**

organization**:**RunningOrganization**,**

startLocation**:** one Location**,**

intermediaryLocation**:** one Location**,**

endLocation**:** one Location

**}**

**{**

**no** l**:** intermediaryLocation **|** startLocation**=**l **or** endLocation**=**l

**}**

abstract sig Request**{**

--requestTime: one Time,

requestedData**:** one **Data,**

status**:** RequestStatus**,**

response**:** lone Response**,**

requestedBy**:** one ThirdParty

**}**

**{**

this **in** requestedBy**.**requests

**}**

sig IndividualDataRequest **extends** Request**{**

**user:** one IndividualUser

**}**

sig AnonymDataRequest **extends** Request**{**

**}**

abstract sig RequestStatus**{**

**}**

one sig Received **extends** RequestStatus**{}**

one sig Rejected **extends** RequestStatus**{}**

one sig Approved **extends** RequestStatus**{}**

sig Response**{**

responseData**:** lone **Data,**

status**:** one ResponseStatus**,**

sentTo**:** one ThirdParty**,**

--responseTime :one Time

**}**

**{**

one r**:**Request**|**r**.**response**=**this

**}**

abstract sig ResponseStatus**{**

**}**

one sig RejectionSent **extends** ResponseStatus**{}**

one sig DataSent **extends** ResponseStatus**{}**

sig SosNotification**{**

sender**:** one IndividualUser**,**

receiver**:**one ThirdParty**,**

healthData**:** one HealthData**,**

locationData**:**one LocationData

**}**

**{**

--Time of healthData and locationData must be the same

healthData**.**receiveTime**=**locationData**.**receiveTime **and**

--Notification should only be sent when healt values are out of range

**(**healthData**.**heartBeat**<**10 **or** healthData**.**heartBeat**>**15**)** **and**

healthData **in** sender**.data** **and**

locationData **in** sender**.data**

**}**

--<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Data4Help Facts start>>>>>>>>>>>>>>>>>>>

-- If status of request is “Received” no response should be created

fact requestStatusReceivedResponseRelation**{**

**all** r**:**Request**|** r**.**status**=** Received **=>** #**(**r**.**response**)=**0

**}**

fact requestStatusAppRejectedResponseRelation**{**

**all** r**:**Request**|(** r**.**status**=** Approved **or** r**.**status**=**Rejected**)** **=>** r**.**response **in** Response **and** #**(**r**.**response**)=**1

**}**

--rejected Requests' responses should be rejection sent and responseData should be empty

fact requestRejectedFacts**{**

**all** req**:** Request**,**res**:**Response**|** req**.**response**=**res **and** req**.**status**=**Rejected **=>** res**.**status**=**RejectionSent **and** #**(**res**.**responseData**)=**0 **and** res**.**sentTo**=**req**.**requestedBy

**}**

--approved Requests' responses should be DataSent and responseDataShould be equal to RequestedData

fact requestApprovedFacts**{**

**all** req**:** Request**,**res**:**Response**|** req**.**response**=**res **and** req**.**status**=**Approved **=>** res**.**status**=**DataSent **and** res**.**responseData**=** req**.**requestedData **and** res**.**sentTo**=**req**.**requestedBy

**}**

--every request should have it's own response

fact differentRequestDifferentResponse**{**

**no** disj r1**,**r2**:**Request**|** r1**.**response**=**r2**.**response

**}**

--Every user's data should be specific

fact differentUserDifferentData**{**

**no** disj u1**,**u2**:**IndividualUser**|** u1**.data=**u2**.data**

**}**

--There should be bidirectional relation between IndividualDataRequest and user

fact allIndividualRequestsAreRelatedToUser**{**

**all** r**:**IndividualDataRequest**,** u**:**IndividualUser**|** r**.user=**u iff r **in** u**.**receivedRequests

**}**

fact bidirectionalRelationBetweenDataAndUser**{**

**all** d**:Data,**u**:**IndividualUser**|** d**.user=**u iff d **in** u**.data**

**}**

-- Requested data of IndividualDataRequest and User should be related

fact DataRelationBetweenUserAndIndReq**{**

**all** r**:**IndividualDataRequest**,** u**:**IndividualUser**|** r**.user=**u iff r**.**requestedData**.user=**u

**}**

--Every usedDrugs should be related with at least one disease

fact checkUsedDrugs**{**

**all** d**:** UsedDrugs**|** **some** c**:**ChronicDisease**|** d **in** c**.**drugs

**}**

--Every request should be peculiar to it's ThirdParty

fact differentRequestDifferentThirdParty**{**

**no** disj t1**,**t2**:** ThirdParty**|** #**(**t1**.**requests **&** t2**.**requests**)>**0

**}**

--all Individual request which are requested by preconfirmedThirdparty should be approved

fact IndReqOfPreConfirmedShouldBeApproved**{**

**all** i**:**IndividualDataRequest **|**i**.**requestedBy **in** i**.user.**preConfirmedThirdParties**=>** i**.**status**=**Approved

**}**

--Each user should has only one ProfileData

fact eachUserHasOnlyOneProfileData**{**

**all** u**:**IndividualUser**|** #**(**ProfileData **&** u**.data)=**1

**}**

--<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Data4Help Facts end>>>>>>>>>>>>>>>>>>>

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--

--<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Track4Run Facts start>>>>>>>>>>>>>>>>>>>

--User can't enroll a run as a both spectator and runner

fact userCantEnrollAsSpectatorToParticipatedRun**{**

**all** u**:**IndividualUser**|** #**(**u**.**participatedOrganizations **&** u**.**spectatedOrganizations**)=**0

**}**

--Spectators and runners of a run is all different

fact runnerSpectatorRelation**{**

**all** r**:**RunningOrganization**|** #**(**r**.**spectators **&** r**.**runners**)=**0

**}**

--All runners in a run must participate it

fact runnerUserRelation**{**

**all** r**:**RunningOrganization**,** u**:**IndividualUser**|** u **in** r**.**runners iff r **in** u**.**participatedOrganizations

**}**

--All spectators in a run must spectate it

fact spectatorUserRelation**{**

**all** r**:**RunningOrganization**,** u**:**IndividualUser**|** u **in** r**.**spectators iff r **in** u**.**spectatedOrganizations

**}**

-- All organizations must be in the list of it's organizer

fact organizationOrganizerRelation**{**

**all** r**:** RunningOrganization**,** t**:**ThirdParty**|** r **in** t**.**organizations iff r**.**organizer**=**t

**}**

--Each run has it's specific path

fact differentRunDifferentPath**{**

**no** disj o1**,**o2**:**RunningOrganization**|** o1**.path=**o2**.path**

**}**

fact organizationPathRelation**{**

**all** r**:** RunningOrganization**,** p**:Path|** r**=**p**.**organization iff r**.path=**p

**}**

--Organizer of run should activate Track4run

fact organizerTrack4RunConstraint**{**

**all** t**:**ThirdParty**,** tr**:**Track4Run**|** #**(**t**.**organizations**)>**0 iff tr **in** t**.**services

**}**

--Runner should activate Track4Run

fact runnerTrack4RunConstraint**{**

**all** t**:**IndividualUser **|** #**(**t**.**participatedOrganizations**)>**0 iff Track4Run **in** t**.**services

**}**

--Runner should have at least 1 locationData

fact runnerLocationDataConstraint**{**

**all** t**:**IndividualUser**|** **some** l**:**LocationData**|** #**(**t**.**participatedOrganizations**)>**0 **=>** t**.data=**l

**}**

--#LocationData and #Runners should be equal in Organization

fact locationNumberAndRunnerConstraint**{**

**all** o**:**RunningOrganization**|**#**(**o**.**runners**)=**#**(**o**.**locationData**)**

**}**

--Organization should access to it's runners' location. So, spectators and organizers can access it

fact organizationShouldHaveLocationDataOfRunners**{**

**all** r**:**RunningOrganization**,** u**:**IndividualUser**|** u **in** r**.**runners implies one l**:**LocationData**|** l**.user=**u **and** l **in** r**.**locationData

**}**

fact differentOrganizationDifferentData**{**

**no** disj r1**,**r2**:** RunningOrganization**|** #**(**r1**.**locationData **&** r2**.**locationData**)>**0

**}**

--<<<<<<<<<<<<<<<<<<<<<<<<<<<<<Track4Run Facts end>>>>>>>>>>>>>>>>>>>

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--

--<<<<<<<<<<<<<<<<<<<<<<<<<<<<<AutomatedSOS Facts start>>>>>>>>>>>>>>>>>>>

--All AutomatedSOS users should have location and health data

fact sosUserShouldHaveLocationAndHealthData**{**

**all** u**:**IndividualUser**,** s**:**AutomatedSOS**|** #**(**u**.**services**&**s**)=**1 **=>** **some** l**:**LocationData**,** h**:**HealthData**|** l**.**receiveTime**=**h**.**receiveTime **and** l**.user=**u **and** h**.user=**u

**}**

--sender of notification must be AutomatedSosUser

fact senderMustBeSosUser**{**

**all** n**:**SosNotification**,** u**:**IndividualUser**|** n**.**sender**=**u**=>** #**(**u**.**services**&**AutomatedSOS**)=**1

**}**

--receiver of notification must be AutomatedSosUser

fact receiverMustBeSosUser**{**

**all** n**:**SosNotification**,** t**:**ThirdParty**|** n**.**receiver**=**t **=>** #**(**t**.**services**&**AutomatedSOS**)=**1

**}**

fact forEveryExceedOfThresholdOnlyOneNotificationMustBeSent**{**

**no** disj n1**,**n2**:** SosNotification**|** #**(**n1**.**healthData **&** n2**.**healthData**)>**0

**}**

--<<<<<<<<<<<<<<<<<<<<<<<<<<<<<AutomatedSOS Facts end>>>>>>>>>>>>>>>>>>>

--Each response should be peculiar to a request

assert assert\_differentRequestDifferentResponse**{**

**no** r1**,**r2**:**Request**|** r1**.**response**=**r2**.**response

**}**

--

assert assert\_requestRejectedFacts**{**

**all** req**:** Request**,**res**:**Response**|** req**.**response**=**res **and** req**.**status**=**Rejected **=>** res**.**status**=**RejectionSent **and** #**(**res**.**responseData**)=**0

**}**

assert assert\_requestApprovedFacts**{**

**all** req**:** Request**,**res**:**Response**|** req**.**response**=**res **and** req**.**status**=**Approved **=>** res**.**status**=**DataSent **and** res**.**responseData**=** req**.**requestedData

**}**

assert assert\_checkRequestAndResponsesThirdParty**{**

**all** req**:**Request**,** res**:**Response**|** req**.**response**=**res**=>** req**.**requestedBy**=**res**.**sentTo

**}**

pred requestIndividualData**[**u**:**IndividualUser**,**d**:Data,**s**:**RequestStatus**,**r**:**Response**,**rb**:**ThirdParty**,**/\*t:Time,\*/ i**:**IndividualDataRequest**]{**

i**.user=**u

i**.**requestedData**=**d

i**.**status**=**s

i**.**response**=**r

i**.**requestedBy**=**rb

-- i.requestTime=t

**}**

assert checkExceedingLimits**{**

**no** n**:**SosNotification**|** **(**n**.**healthData**.**heartBeat**=**11**)**

**}**

pred showSosUsers**{**

**some** SosNotification

**}**

pred isThereAnonymDataRequest**{**

**some** AnonymDataRequest

**}**

--Organization asserts

assert isThereUserEnrolledAnOrganizationAsSpectatorAndParticipator**{**

**no** u**:**IndividualUser**|** #**(**u**.**spectatedOrganizations **&** u**.**participatedOrganizations**)** **>**0

**}**

assert someOrganizationsDoesntHaveAllRunnersData**{**

**all** o**:**RunningOrganization**,** u**:**IndividualUser**|** u **in** o**.**runners **=>** **(**u**.data** **&** LocationData**)** **in** o**.**locationData

**}**

pred showForData4Help**{**

**(some** u**:**IndividualUser**|**#**(**u**.**preConfirmedThirdParties**)>**0**)** **and**

**(some** t**:**ThirdParty**|**#**(**t**.**requests**)>**0**)** **and**

**(some** AnonymDataRequest**)** **and**

**(some** IndividualDataRequest**)** **and**

**(no** RunningOrganization**)** **and**

**(no** SosNotification**)** **and**

#**(**Location**)=**2

**}**

pred showForTrack4Run**{** one RunningOrganization**}**

run showForTrack4Run

run showForData4Help **for** 5 but 3 IndividualUser**,**2 ThirdParty**,**1 Received**,** 1 AnonymDataRequest**,** 1 Approved**,** 1 Rejected

**check** isThereUserEnrolledAnOrganizationAsSpectatorAndParticipator

**check** assert\_checkRequestAndResponsesThirdParty

**check** assert\_requestRejectedFacts

**check** assert\_differentRequestDifferentResponse

**check** someOrganizationsDoesntHaveAllRunnersData

run isThereAnonymDataRequest

run requestIndividualData

**check** checkExceedingLimits

pred show**{**

**}**

run show