



## **EECS 4315 Final Report**



Jay Sharma - jksharma@my.yorku.ca - Student ID: 219358225  
Kedamawi Mengistu - Kedim22@my.yorku.ca - Student ID: 216339269  
Ruth Bezabeh - ruttkas@my.yorku.ca - Student ID: 216171795

# Table of Contents

Description of System & Features - (From First Progress Report).....	3
Architecture Schema - (From First Progress Report).....	3
Version 1: Original Architecture.....	4
Version 2: Partially Updated Architecture.....	5
Version 3: Mostly Updated Architecture.....	6
Version 4: Fully Completed Architecture.....	7
LNT Modeling Description - Different Versions of the Code as outlined later in the report briefly discuss element of B), C), and D).....	8
Description of the LNT Model - First Progress Report Description.....	8
A) LNT Model stats: NB types, NB functions, NB processes, NB channels, NB synchronised actions, LTS size.....	9
B) Main and specific features of your model (Final Report Summary).....	9
C) Abstraction/modelling choices (Final Report Summary).....	9
D) Modelling challenges (Final Report Summary).....	10
Verification: Different Code Versions & Issue Tracking.....	11
(Note To Professor: Version to Start reading from for final report assessment is Version 6: Trial641.lnt).....	11
Version 1: uae.lnt.....	11
Version 2: uae_version2.lnt.....	12
Version 3: uae_version3.lnt.....	13
Version 4: Trial6_1.lnt.....	16
Version 5: Trial6_4.lnt.....	20
Full LTS of Trial6_4.lnt.....	23
Min LTS of Trial6_4.lnt.....	25
Version 6: Trial641.lnt.....	25
Version 7: Trial641a0.lnt.....	29
Version 8: Trial641a1.lnt.....	31
Version 9: Trial641a12.lnt.....	36
Version 10: Trial641a2.lnt.....	41
Version 11: Trialx4.lnt.....	45
Model Checking.....	48
Description of Properties.....	48
Tested Properties.....	48
General Overview of Modifications to Properties From Progress Report.....	62
Verification.....	66
Difficulties Encountered.....	67
Feedback on CADP.....	68
References.....	68

## Description of System & Features - (From First Progress Report)

The TimeTree application is a robust system designed for collaborative scheduling and event management with mission critical communication features. TimeTree integrates comprehensive safety protocols to mitigate risks associated with collaboration and data sharing. This includes strict identification (ID) processes that ensure the confidentiality and integrity of user data, guarding against unauthorised access and data breaches. In addition to its emphasis on safety, TimeTree also prioritises effective communication channels to facilitate seamless interaction among users. Through its intuitive graphical user interface and real-time syncing capabilities, TimeTree enables swift dissemination of event updates and relevant information, promoting timely decision-making and coordination. Furthermore, the platform supports multi-directional communication channels, allowing users to exchange messages, comments, and attachments within the context of specific events, memos or calendars. This fosters a dynamic and collaborative workflow, empowering users to clarify details, resolve conflicts, and engage in productive dialogue within the framework of their shared schedules.

In summary, Time Tree is a secure and productive digital environment that empowers users to coordinate scheduling and communication effectively while prioritising data integrity and user engagement.

## Architecture Schema - (From First Progress Report)

TimeTree is a collaborative calendar scheduling app and the architecture has undergone fourth iterations with the first relatively simple design illustrated in Figure 1, and then partially updated in Figure 2 and again updated in Figure 3 and completed with Figure 4, which is a high-level architectural overview of TimeTree. It illustrates the system's main components, their relationships, and data flow between them, serving as a guide for developers and stakeholders to understand how the app functions and interacts with its users. TimeTree has a modular architecture with a separation of concerns for increased maintainability and scalability as it serves both mobile and web-client users.

## Version 1: Original Architecture

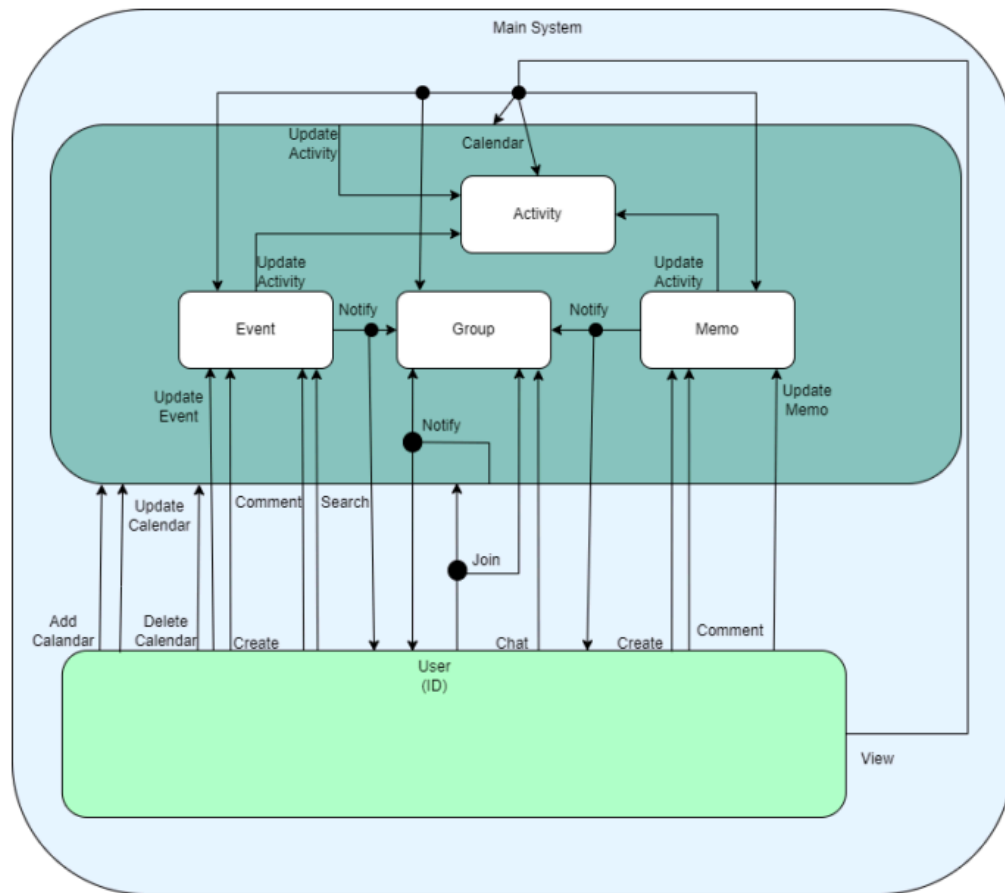


Figure 1: Architecture Model Version 1 of TimeTree Application

## Version 2: Partially Updated Architecture

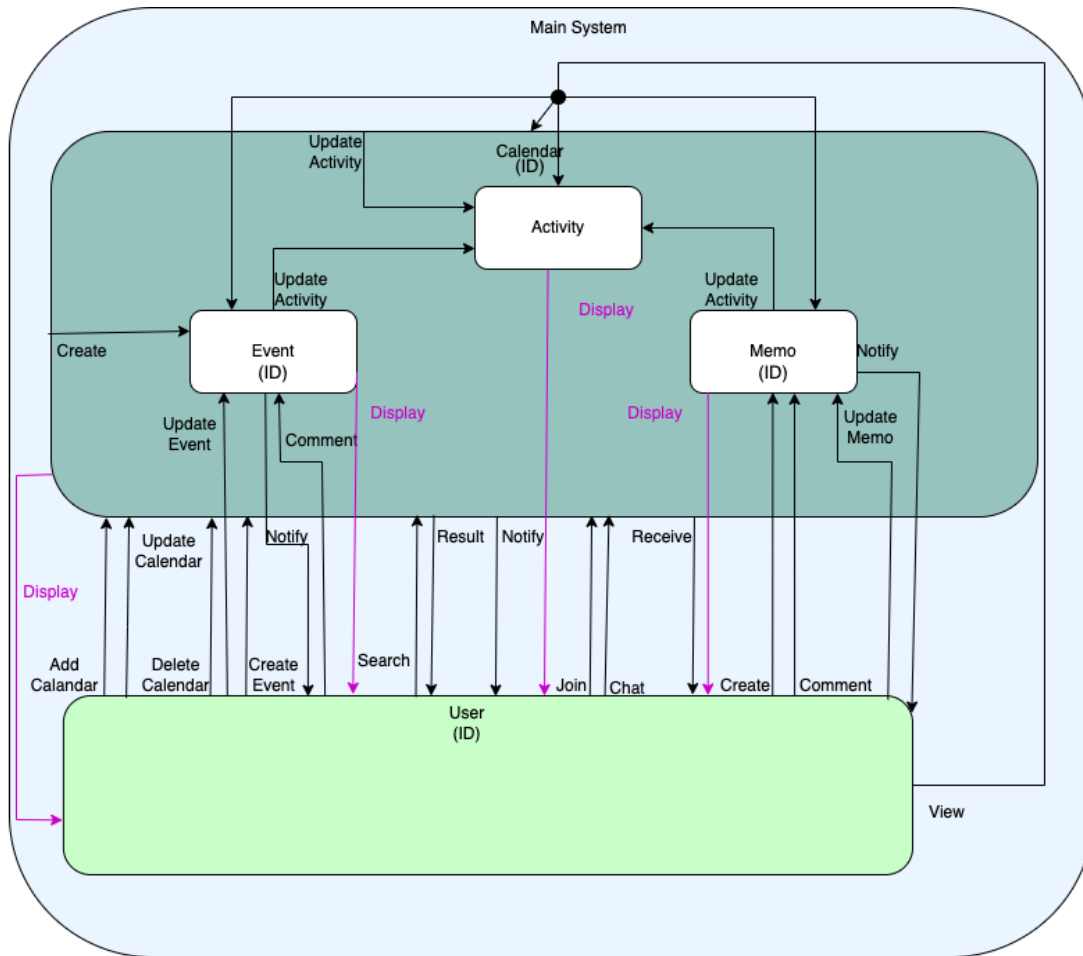


Figure 2: Architecture Model Version 2 of TimeTree Application no longer containing the Group Process

### Version 3: Mostly Updated Architecture

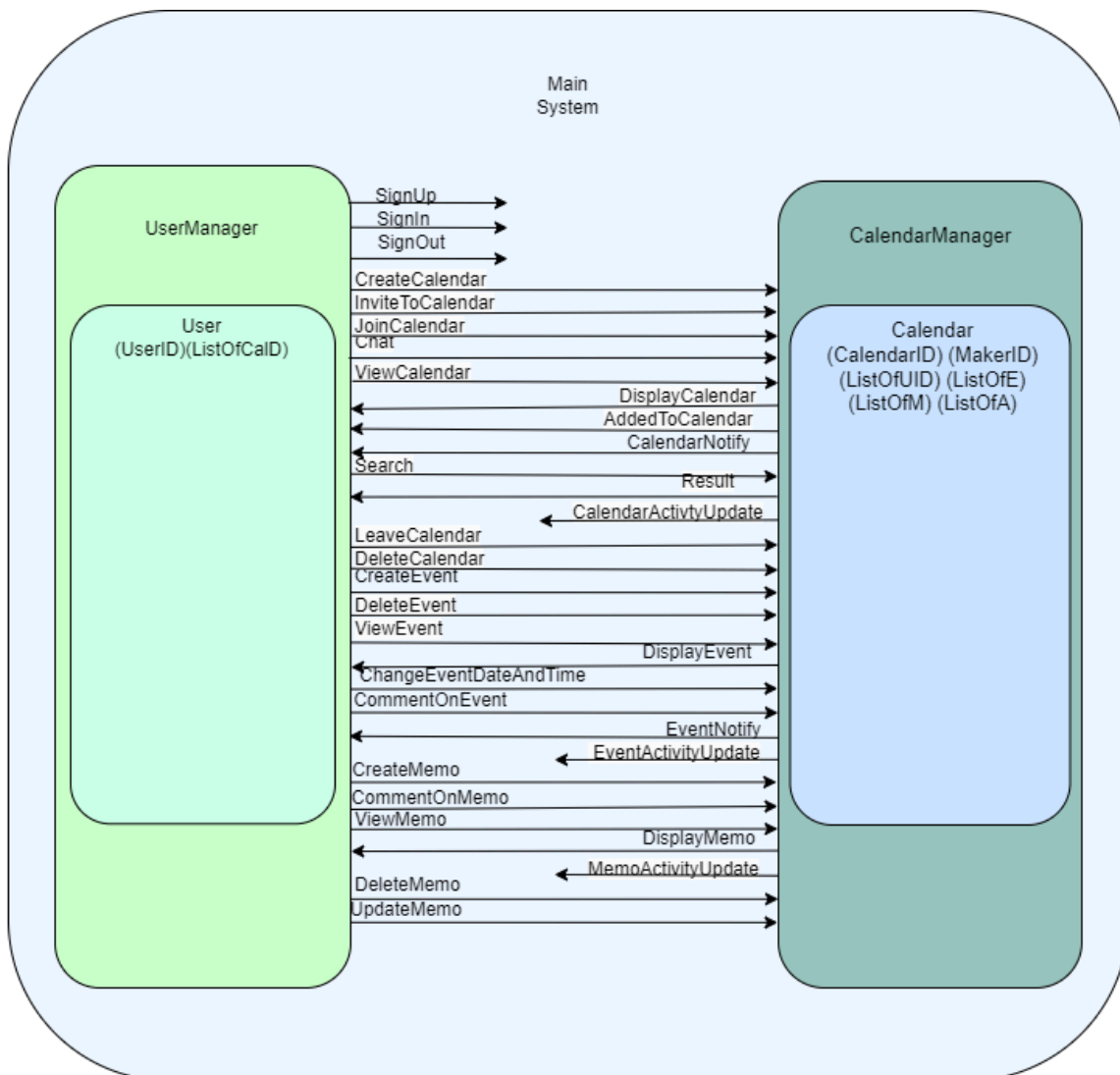


Figure 3: Architecture Model Version 3 of TimeTree Application with only user and calendar processes alongside user and calendar manager processes

## Version 4: Fully Completed Architecture

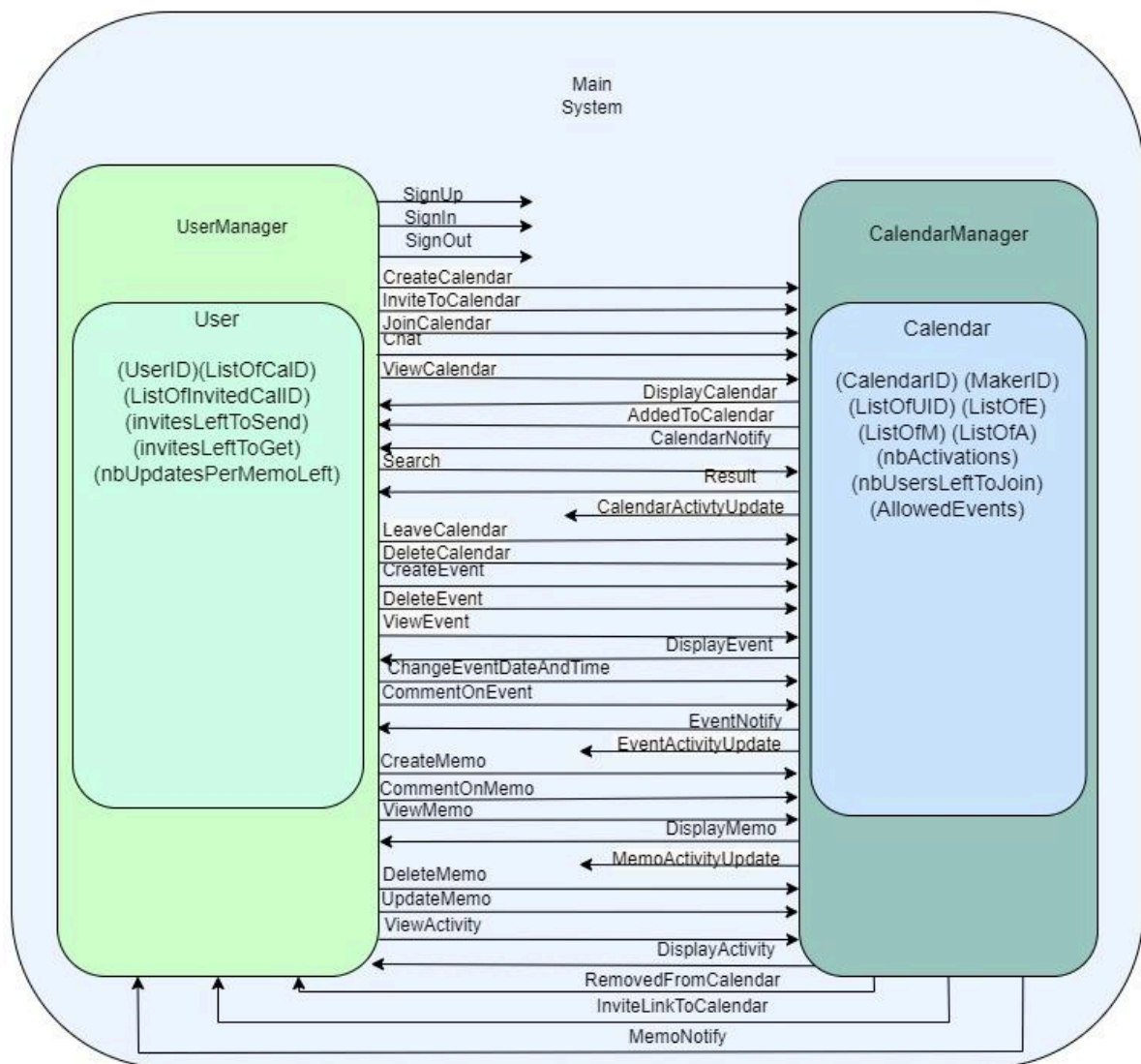


Figure 4: Architecture Model Version 4 of TimeTree Application with completed implementation of user and calendar processes alongside manager processes

LNT Modeling Description - Different Versions of the Code as outlined later in the report briefly discuss element of B), C), and D)

### **Description of the LNT Model - First Progress Report Description**

At the core of the architecture is the Main System, which acts as the central processing unit for the app. It manages the primary business logic, data handling, and orchestrates user interactions. The Main System is composed of four significant sub-components: Calendar, User, and its managers. Each instance of user has a unique ID and a list of unique calendar ids corresponding to different calendar instances, which each have an id, a maker id; the id of the user instance responsible for creating the calendar, and set of lists storing the member users of that calendars the events, memo's, and activities related to that calendar.

More specifically, this current architecture above shows the most critical properties of TimeTree, which are SignIn, SignUp, CreateCalendar, Join Calendar, CreateEvent, and CreateMemo and more. The model has the appropriate nodes and edges for the actions between processes mapped out. A brief description of the LNT model is that the model has the two primary processes that are the User and Calendar, where each has its collection of instances created and managed by the outer processes expressed as User Manager and Calendar Manager processes. Each of these processes interact with the other process via the following set of events/gates/actions: SignIn, SignOut, SignOut, CreateCalendar, DeleteCalendar, JoinCalendar, InviteToCalendar, Chat, LeaveCalendar, Chat, ViewCalendar, DisplayCalendar, CalendarNotify, AddedCalendar, CreateEvent, DeleteEvent, ChangeEventDataAndTime, CommentOnEvent, ViewEvent, DisplayEvent, EventNotify, CreateMemo, DeleteMemo, CommentOnMemo, ViewMemo, MemoNotify. etc. The overall model, through the individual and actionable features it offers, should reflect the properties related to safety, fairness, and liveness where below is a brief description of these properties and later on in the document each category, through different sets of mcl specified properties, will be tested for verification.



## **A) LNT Model stats: NB types, NB functions, NB processes, NB channels, NB synchronised actions, LTS size**

The final version of the model and comprises the following.

- 8 user defined types.
- 7 channels.
- 30 synchronised actions.
- 4553994 states, 13904868 transitions, 34679 labels, 2423616 deadlocks, and an average branching factor of 3.05 (This is also mentioned in version 11: Trialx4.lnt)

## **B) Main and specific features of your model (Final Report Summary)**

Below is a brief list of the main and specific features of the model, where some if not all were also briefly highlighted in the first progress report description above.

- The LNT model of TimeTree is specifically designed to simulate a collaborative calendar system with features like calendar creation, user to user interaction, event scheduling, and memo posting.
- It incorporates user actions such as signing in, signing out, creating, and joining calendars, as well as chatting and searching the calendars that the user belongs to.
- The model also details other non-calendar related user interactions like event and memo creation, as well as event and memo commenting, event date and time changing as well as updates to existing memos, ensuring comprehensive coverage of the application's functionality.
- There are also a number of different standard view and display actions for the calendar, the event, and the memo that is created and exists in the model.
- As well the system is reactive, where it allows for calendar related notifications; for chat, event related notifications; for event date and time change and comment on event, and memo related notifications; for comment on memo and memo update.
- It holds 2 users, where each can perform any of the actions mentioned above with respect to the singularly existing calendar instance in the calendar manager process, where these users can perform any of these actions simultaneously; this is a reflection of the system's concurrent nature.

## **C) Abstraction/modelling choices (Final Report Summary)**

To simplify the complexity, the system was abstracted in several different ways as mentioned below:

- Abstracted from managing multiple calendars to just one shared calendar instance; to reduce the overall number of actions and transitions by half.
- Activation of the calendar is restricted to only once now; the calendar has only one life now to prevent the initially seeming never ending cycle of actions that came with having two calendars having two activations so that now its more finite and reduced in terms of states and transitions once again.
- User interactions were further abstracted to limit each user to sending and receiving only one invite, to prevent an infinitely sized list of invites to be sent to a user.

- The model was constrained to allow the creation of only one event per calendar to prevent an infinite amount of event related actions from occurring simultaneously.
- Further abstraction was applied by limiting the system to only one event and one memo that could be created between the two users; if user 1 created event 1 then because all users can only create 1 they no longer can until it is deleted in spite of the fact that all they could ever create is event 1. The same logic follows for memo.

## **D) Modelling challenges (Final Report Summary)**

The following are the list of modelling challenges faced where they are more deeply discussed in the Difficulties encountered section.

- The model faced challenges in maintaining complete synchronisation between user and calendar managers due to an initially unaccounted gate not being synchronised amongst the sea of gates that existed already, leading to inconsistencies in user-calendar interactions. This is particularly referring to the issue of not having INVITELINKTOCALENDAR synchronised between the managers in the main process leading to occurrences of INVITELINKTOCALENDAR happening whenever; this is more deeply discussed in the problems encountered section as mentioned above.
- Efforts to minimise the model's size by limiting functionalities like the number of events and memos, although necessary, deviated from a realistic representation of the actual application. It was challenging to overly simplify the interactions in spite of being unsure of what should and should not be fully captured by the model's representation of the TimeTree application.
- Model Size reduction was an issue as the enormous size of gates defined, proved to be challenging to reduce but keep at the same time; data being sent across channels such as the event and memo id being set to one as the only option for any user to make use of had to be done to reduce the number of unique events and memos that could exist and the instance to scale down the model and improve compilation time.

## Verification: Different Code Versions & Issue Tracking

(Note To Professor: Version to Start reading from for final report assessment is Version 6: Trial641.lnt)

### Version 1: uae.lnt

- Data Type & Channel Definitions:

```
module uae with ==, !=, >= is

  type page is activity, group, memo, event, calendar end type
  type update is memoupdate, eventupdate, calendarupdate end type
  type updates is list of update with empty, head, tail, member end type
  type notification is n_eventupdate, n_eventstarting, n_memoupdate,
n_calendarupdate end type
  --type comment is comment(ID: nat) end type
  type commentpage is event, memo end type
  --type time is nat where time <= 24 end type

  channel P is (p: page) end channel
  channel U is (u: update) end channel
  channel N is (n: notification) end channel
  channel E is (ID: nat, t: nat, date: nat ) end channel
  channel C is (c: nat, p: commentpage ) end channel
  channel I is (ID: nat) end channel
  channel J is (CID: nat, UID: nat) end channel
```

- Activity Process Definition:

```
process Activity [View: P, Display: P, UpdateActivity: U] (in var
Lactivities: updates) is
```

- Event Process Definition:

```
process Event[View: P, Display: P, UpdateActivity: U, UpdateEvent: E,
DeleteEvent: I, Comment: C, Create: E] (ID: Nat) is
```

- User Process Definitions:

```
process User[View: P, Display: P, UpdateEvent: E, DeleteEvent: I, Create:
E, Comment: C] (ID: Nat) is
```

- Main Process Definitions:

```
process Main[View: P, Display: P, UpdateActivity: U, UpdateEvent: E,
DeleteEvent: I, Comment: C, Create: E] is
```

- Bug Output:

```
File Edit View Search Terminal Help
In process LOOP_1 [474]

- operation 1 used in value:
  1
  returns a result of undefined sort; its profile is
  -> (NUMBER [LNT_V1:1328] | INT [LNT_V1:286] | NAT [LNT_V1:95])
  (the sort of the result should be specified by ``of'')

- operation 1 used in value:
  1 DECNUM 0
  returns a result of undefined sort; its profile is
  -> (NUMBER [LNT_V1:1328] | INT [LNT_V1:286] | NAT [LNT_V1:95])
  (the sort of the result should be specified by ``of'')

- operation 0 used in value:
  1 DECNUM 0
  returns a result of undefined sort; its profile is
  -> (NUMBER [LNT_V1:1328] | INT [LNT_V1:286] | NAT [LNT_V1:95])
  (the sort of the result should be specified by ``of'')

- operation 2 used in value:
  ((2 DECNUM 0) DECNUM 2) DECNUM 4
  returns a result of undefined sort; its profile is
  -> (NUMBER [LNT_V1:1328] | INT [LNT_V1:286] | NAT [LNT_V1:95])
  (the sort of the result should be specified by ``of'')

- operation 0 used in value:
  ((2 DECNUM 0) DECNUM 2) DECNUM 4
```

- Bug Output Response:

There were several syntax errors related to the initial uae.lnt module which led to a debug session with the professor, resulting in a pivot involving the construction of process definitions and their critical sections; a particular important adjustment that should be noted is that the

Main process' parallel synchronisation drastically changed to synchronise User with a newly formed process called TimeTree, as seen in version 2 of the code. To add, unlike the newly added Time Tree process, other processes which were included in the original model architecture, but just not implemented yet, such as Memo and Calendar, were now introduced into the module.

---

## Version 2: uae\_version2.Int

### - Data Type & Channel Definitions:

```
module uae_version2 with ==, !=, >= is
-----
-- Types|
-----

type page is activity, group, memo, event, calendar end type
type update is memoupdate, eventupdate, calendarupdate end type
type updates is list of update with empty, head, tail, member end type
type notification is n_eventupdate, n_eventstarting, n_memoupdate, n_calendarupdate end type
--type comment is comment(ID: nat) end type
type commentpage is event, memo end type
--type time is nat where time <= 24 end type

-----
-- Channels
-----

channel P is (p: page) end channel
channel U is (u: update) end channel
channel N is (n: notification) end channel
channel E is (ID: nat, t: nat, date: nat ) end channel
channel C is (c: nat, p: commentpage ) end channel
channel I is (ID: nat) end channel
channel J is (CID: nat, UID: nat) end channel
```

### - Activity Process Definition:

```
process Activity [View, Display, CalendarUpdateActivity,
MemoUpdateActivity, EventUpdateActivity: none] is
```

### - Event Process Definition:

```
process Event[View, Display, UpdateActivity, UpdateEvent, DeleteEvent,
Comment, Create: none, Debug: any] (myId: Nat) is
```

### - Memo Process Definition:

```
process Memo[View, Display, UpdateActivity, Notify, DeleteMemo, Comment,
Create, UpdateMemo: none] is
```

### - Calendar Process Definition:

```
process Calendar [CalendarUpdateActivity, View, Create, Display,
DeleteCalendar, Search, Result, AddCalendar, UpdateCalendar, Join, Chat,
CreateEvent, CreateMemo: none] is
```

### - Time Tree Process Definition:

```
process TimeTree [View, Display, Notify, DeleteCalendar, Search, Result,
AddCalendar, UpdateCalendar, Join, Chat, CreateEvent, Create, CreateMemo,
CalendarUpdateActivity, MemoUpdateActivity, EventUpdateActivity,
UpdateEvent, DeleteEvent, Comment, UpdateMemo, DeleteMemo: none, Debug:
any] is
```

- User Process Definition:

```
process User[View, Notify, UpdateEvent, DeleteEvent, DeleteCalendar,
DeleteMemo, Comment, Search, Result, UpdateMemo, AddCalendar,
UpdateCalendar, Join, Chat, CreateEvent, CreateMemo: none] is
```

- Main Process Definition:

```
process Main[View, Display, Notify, UpdateEvent, DeleteEvent,
DeleteCalendar, DeleteMemo, Comment, Search, Result, UpdateMemo,
AddCalendar, UpdateCalendar, Join, Chat, CreateEvent, CreateMemo, Create,
CalendarUpdateActivity, MemoUpdateActivity, EventUpdateActivity: none,
Debug: any] is
```

- Bug Output: No Bugs (uae\_version2 compiles)

```
kedl@kedl-VirtualBox:~$ lnt.open uae_version2.lnt generator uae_version2.bcg
lnt.open: checking ``uae_version2.lnt''
lnt.open: translating ``uae_version2.lnt'' to ``/tmp/kedl_lnt.open_uSzYsQ/uae_version2.lotos'' ...
lnt.open: calling lotos.open for ``uae_version2.lotos'' ...
lotos.open: using ``(direct) caesar''
lotos.open: calling ``caesar.adt -debug -silent uae_version2''
lotos.open: calling ``caesar -silent uae_version2''
kedl@kedl-VirtualBox:~$ bcg_edit uae_version2.bcg
```

- As you can see a Debug gate was added by the professor to help extract the bugs found from the original existing code as in the process of pivoting to how the code was currently structured, where a TimeTree process was introduced, at that moment.

### Version 3: uae\_version3.lnt

- The central changes in this version of the code were those found in the critical section of TimeTree, Calendar, and Event and in the process definition of Main to incorporate the newly added gates. More specifically, Activate, Deactivate, ActivateEvent, and DeactivateEvent gates were introduced, where TimeTree was now responsible for creating and ultimately Activating and inversely Deactivating Calendar instances and similarly Calendar was now responsible for creating and Activating and Deactivating Event instances.

TimeTree Process - Creating Calendars and activating and deactivating them.

```
process TimeTree [View, Display, EventDisplay, ActivityDisplay, MemoDisplay,
Notify, DeleteCalendar, Search, Result, AddCalendar, UpdateCalendar, Join,
Chat: none, CreateEvent, CreateCalendar, CreateMemo, CalendarUpdateActivity,
MemoUpdateActivity, EventUpdateActivity, UpdateEvent: none, DeleteEvent, Comment,
UpdateMemo, DeleteMemo: none, Activate, Desactivate, ActivateEvent, DesactivateEvent: I,
EventView: none] is
  var calendatID1: nat in
    -- initialisation
    calendatID1 := 0;

  par
    loop
      -- Handling different calendars
      select
        CreateCalendar;
        Activate (calendatID1)
      []
        DeleteCalendar;
        Desactivate (calendatID1)
      []
        AddCalendar
      end select
    end loop
  ||
```

```

||
CalendarUpdateActivity, CreateMemo, EventUpdateActivity ->
  Calendar [CalendarUpdateActivity, View, Display,
            Search, Result, UpdateCalendar, Join, Chat, CreateEvent, DeleteEvent,
            CreateMemo, Activate, Desactivate, ActivateEvent, DesactivateEvent,
            EventView, EventDisplay, EventUpdateActivity, UpdateEvent, Comment] (calendarID1)

```

Calendar Process - Calendar Instances listen to be activated and deactivated, and to activate and deactivate Events that they create and delete.

```

process Calendar [CalendarUpdateActivity, View, Display,
                  Search, Result, UpdateCalendar, Join, Chat: none, CreateEvent, DeleteEvent,
                  CreateMemo: none, Activate, Desactivate, ActivateEvent, DesactivateEvent: I,
                  EventView, EventDisplay, EventUpdateActivity, UpdateEvent, Comment: none] (myID: nat) is
var calendarActivated: bool, nbEvents, nbEventsActivated, eventID1, eventID2: nat in

-- initialization
calendarActivated := false;
eventID1 := 0;
eventID2 := 1;
nbEvents := 2;
nbEventsActivated := 0;

par
  DesactivateEvent, ActivateEvent ->
  -- Main Calendar behaviour
  loop
    select
      var tempN: Nat in
        -- activate the Calendar
        Activate (?tempN) where tempN == myID;
        calendarActivated := true
    end var
  []
end var

```

```

[]
  if calendarActivated then
    DeleteEvent;
    if (nbEventsActivated > 0) then
      DesactivateEvent (nbEventsActivated - 1);
      nbEventsActivated := nbEventsActivated - 1
    end if
  end if
[]
  if calendarActivated then
    CreateEvent;
    if nbEventsActivated < nbEvents then
      ActivateEvent (nbEventsActivated);
      nbEventsActivated := nbEventsActivated + 1
    end if
  end if
[]

```

```

DesactivateEvent, ActivateEvent ->
par
  Event[EventView, EventDisplay, EventUpdateActivity, UpdateEvent,
        DesactivateEvent, Comment, ActivateEvent] (eventID1)
  ||
  Event[EventView, EventDisplay, EventUpdateActivity, UpdateEvent,
        DesactivateEvent, Comment, ActivateEvent] (eventID2)
end par

```

Event Process - Event instances listen to be activated and deactivated.

```
process Event[View, EventDisplay, UpdateActivity, UpdateEvent: none, Desactivated: I, Comment: none, Activated: I] (myId: Nat) is
  var eventActivated: bool in
    -- initialization
    eventActivated := false;
  loop
    select
      var tempN: nat in
        Activated (?tempN) where tempN == myID;
        eventActivated := true
      end var
    []
      var tempN: nat in
        Desactivated (?tempN) where tempN == myID;
        eventActivated := false
      end var
```

Main Process - Activate, Deactivate, Activate Event, and DeactivateEvent are all accounted for as gates in Main process definition.

```
process Main[View, CalendarDisplay, EventDisplay, ActivityDisplay, MemoDisplay,
  Notify, UpdateEvent: none, DeleteEvent, DeleteCalendar, DeleteMemo, Comment,
  Search, Result, UpdateMemo, AddCalendar, UpdateCalendar, Join, Chat,
  CreateEvent, CreateMemo, CreateCalendar, CalendarUpdateActivity, MemoUpdateActivity: none,
  EventUpdateActivity: none, Activate, Deactivate, ActivateEvent, DeactivateEvent: I, EventView: none] is
```

- Bug Output: No Bugs (uae\_version3 compiles)

```
keddi@keddi-VirtualBox:~$ lnt.open uae_version3.lnt generator uae_version3.bcg
lnt.open: checking ``uae_version3.lnt``
lnt.open: translating ``uae_version3.lnt`` to ``/tmp/keddi_lnt.open_H0cPTS/uae_version3.lotos`` ...
lnt.open: calling lotos.open for ``uae_version3.lotos`` ...
lotos.open: using ``(direct) caesar``
lotos.open: calling ``caesar.adt -debug -silent uae_version3``
lotos.open: calling ``caesar -silent uae version3``
```

- As you can see this code version compiles without bugs, signifying the correct integration of Activate, Deactivate, ActivateEvent, and DeactivateEvent (integration was led by and occurred in meeting with the professor).
  - The problem here as it relates to continued development lies in the Activity process not being created as described by the original model architecture, where this signified an increased level of complexity that the team of students was unsure of how to tackle, hence leading to the entire restructuring of the model architecture and hence restructuring of the code.
-

## Version 4: Trial6\_1.Int

### - Data Type & Channel Definitions:

```
module Trial6_1 with ==, !=, >= is
-----
-- Types
-----
type Event is Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat) end type
type Memo is Memo (ID: Nat, MemoContent: Nat) end type
type Activity is Activity (ID: Nat, CalendarID: Nat, EventUpdate: bool, MemoUpdate: bool, ObjectID: Nat) end type
type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
-- type ListOfEvent is list of Event end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
-- type ListOfMemo is list of Memo end type
type ListOfActivity is list of Activity with empty, head, tail, member end type
-- type ListOfActivity is list of Activity end type

-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel

-----
-- Processes
-----
```

### - CalendarManager Process Definition:

```
process CalendarManager [CalendarActivityUpdate,
EventActivityUpdate, MemoActivityUpdate: none,
                        CreateCalendar: N2, DeleteCalendar: N2,
                        InviteToCalendar, JoinCalendar,
LeaveCalendar,
                        Chat,
                        ViewCalendar, DisplayCalendar,
                        CalendarNotify: none,
                        AddedToCalendar: N2,
                        CreateEvent, DeleteEvent,
                        ChangeEventDateAndTime,
                        CommentOnEvent,
                        ViewEvent, DisplayEvent,
                        EventNotify,
                        CreateMemo, DeleteMemo,
                        CommentOnMemo,
                        ViewMemo, DisplayMemo,
                        MemoNotify: none] is
```



- UserManager Process Definition:

```

process UserManager [SignUp, SignIn, SignOut: N,
                  CreateCalendar: N2, DeleteCalendar: N2,
                  InviteToCalendar, JoinCalendar,
                  LeaveCalendar,
                  Chat,
                  ViewCalendar, DisplayCalendar,
                  CalendarNotify: none,
                  AddedToCalendar: N2,
                  CreateEvent, DeleteEvent,
                  ChangeEventDataAndTime,
                  CommentOnEvent,
                  ViewEvent, DisplayEvent,
                  EventNotify,
                  CreateMemo, DeleteMemo,
                  CommentOnMemo,
                  ViewMemo, DisplayMemo,
                  MemoNotify: none] is

```

- Calendar Process Definition:

```

process Calendar[CalendarActivityUpdate, EventActivityUpdate,
MemoActivityUpdate: none,
                  CreateCalendarReceive: N2, DeleteCalendarReceive
                  N2,
                  InviteToCalendarReceive, JoinCalendarReceive,
                  LeaveCalendarReceive,
                  ChatReceive,
                  ViewCalendarReceive, DisplayCalendar,
                  CalendarNotify: none,
                  AddedToCalendar: N2,
                  CreateEventReceive, DeleteEventReceive,
                  ChangeEventDataAndTimeReceive,
                  CommentOnEventReceive,
                  ViewEventReceive, DisplayEvent,
                  EventNotify,
                  CreateMemoReceive, DeleteMemoReceive,
                  CommentOnMemoReceive,
                  ViewMemoReceive, DisplayMemo,
                  MemoNotify: none] (CalendarID: Nat, in var
                                MakerID: Nat, in var
                                ListOfUID: ListOfUserID, in
var
                                ListOfE: ListOfEvent, in var
                                ListOfM: ListOfMemo, in var
                                ListOfA: ListOfActivity) is

```

- User Process Definition:

```

process User[SignUp, SignIn, SignOut: N,
             CreateCalendar: N2, DeleteCalendar: N2,
             InviteToCalendar, JoinCalendar, LeaveCalendar,
             Chat,
             ViewCalendar, DisplayCalendarReceive,
             CalendarNotifyReceive: none,
             AddedToCalendarReceive: N2,
             CreateEvent, DeleteEvent,
             ChangeEventDataAndTime,
             CommentOnEvent,
             ViewEvent, DisplayEventReceive,
             EventNotifyReceive,
             CreateMemo, DeleteMemo,
             CommentOnMemo,
             ViewMemo, DisplayMemoReceive,
             MemoNotifyReceive: none] (UserID: Nat, in var
ListOfCalID: ListOfCalendarID) is

```

- Main Process Definition & Implementation:

```

process Main[SignUp, SignIn, SignOut: N,
    CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
    CreateCalendar: N2, DeleteCalendar: N2,
    InviteToCalendar, JoinCalendar, LeaveCalendar,
    Chat,
    ViewCalendar, DisplayCalendar,
    CalendarNotify: none,
    AddedToCalendar: N2,
    CreateEvent, DeleteEvent,
    ChangeEventDataAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none] is
par CreateCalendar, DeleteCalendar,
    InviteToCalendar, JoinCalendar, LeaveCalendar,
    Chat,
    ViewCalendar, DisplayCalendar,
    CalendarNotify,
    AddedToCalendar,
    CreateEvent, DeleteEvent,
    ChangeEventDataAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo,
    ViewMemo, DisplayMemo,
    MemoNotify in
    UserManager [SignUp, SignIn, SignOut,
        CreateCalendar, DeleteCalendar,
        InviteToCalendar, JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        CreateEvent, DeleteEvent,
        ChangeEventDataAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo,
        ViewMemo, DisplayMemo,
        MemoNotify]
||
    CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
        CreateCalendar, DeleteCalendar,
        InviteToCalendar, JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        CreateEvent, DeleteEvent,
        ChangeEventDataAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo,
        ViewMemo, DisplayMemo,
        MemoNotify]

```

- Bug Output: No Bugs (Trial6\_4 compiles)

```

kedi@kedi-VirtualBox:~$ lnt.open Trial6_1.lnt generator Trial6_1.bcg
lnt.open: file ``./Trial6_1.lnt`` does not exist
kedi@kedi-VirtualBox:~$ lnt.open Trial6_1.lnt generator Trial6_1.bcg

lnt.open: checking ``Trial6_1.lnt``
Trial6_1.lnt:158: warning: "in var" parameter ListOfE never used
Trial6_1.lnt:159: warning: "in var" parameter ListOfM never used
Trial6_1.lnt:160: warning: "in var" parameter ListOfA never used
Trial6_1.lnt:312: warning: "in var" parameter ListOfCalID never used

lnt.open: translating ``Trial6_1.lnt`` to ``/tmp/kedi_lnt.open_C49wnr/Trial6_1.lotos`` ...

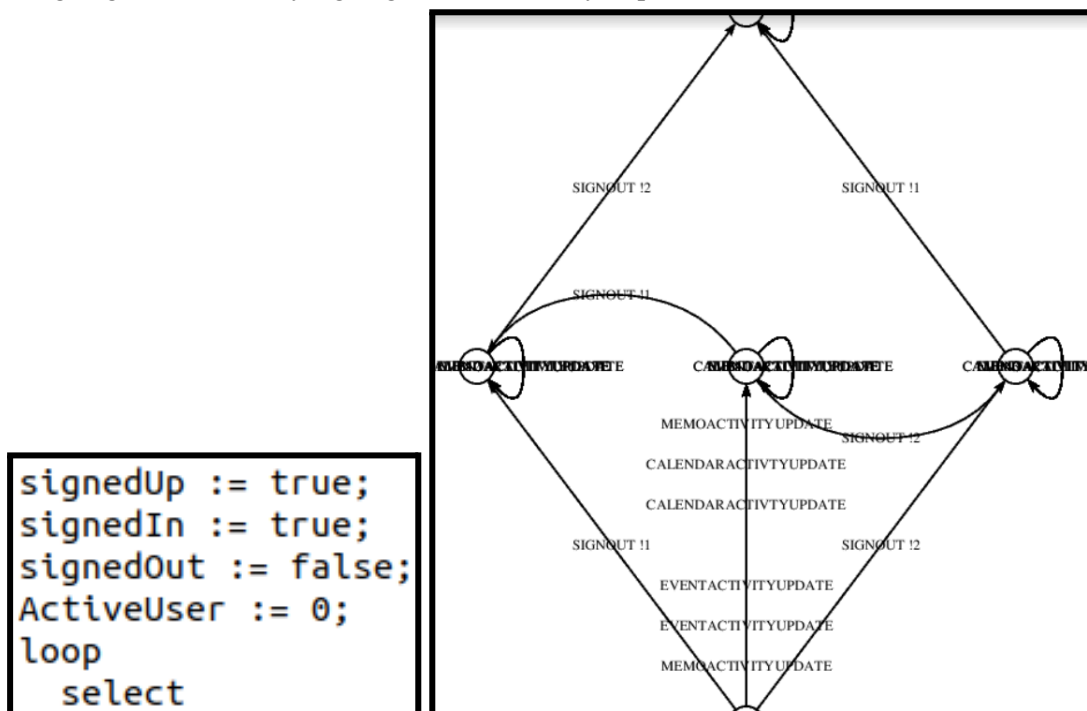
lnt.open: calling lotos.open for ``Trial6_1.lotos`` ...
lotos.open: using ``(direct) caesar``
lotos.open: calling ``caesar.adt -debug -silent Trial6_1``
lotos.open: calling ``caesar -silent Trial6_1``
kedi@kedi-VirtualBox:~$

```

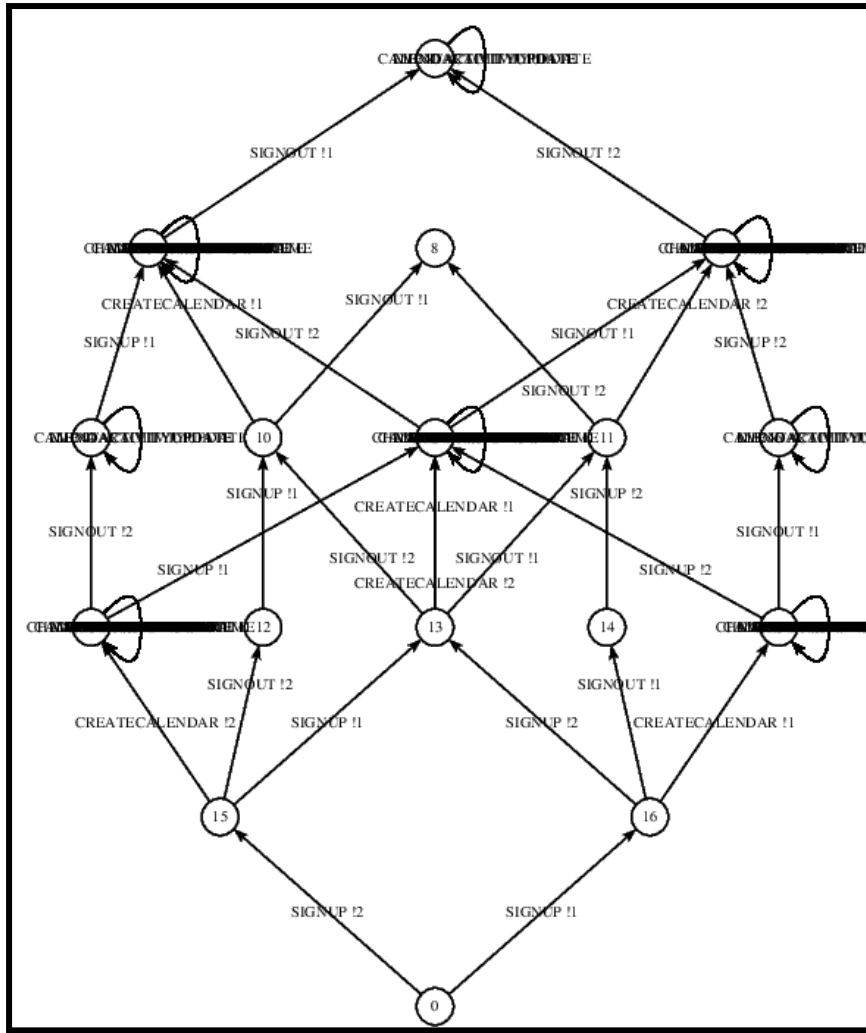
- This current version of the LNT model in similar fashion to the Trial6\_4.lnt is significantly different from the previous ones. As hinted, it like the Trial6\_1.lnt no longer has Event, Memo and Activity as processes but as lists in the Calendar process. This is to ensure a unique calendar keeps track of all events, memos and activities that correspond to it. By

removing these processes, it also eliminates the need to have gates to pass the active or inactive state of Event, Memo and Activity. Therefore they have been replaced by variables maintained inside the process Calendar and User.

- It includes new processed UserManager and CalendarManager to manage concurrency and synchronisation.
- It includes new gates - SignIn, SignUp, SignOut and UpdateMemo, and renames some old ones - UpdateEvent to ChangeDateAndTimeOfEvent, Comment to CommentOnMemo and CommentOnEvent, View to ViewMemo, ViewEvent and ViewCalendar.
- Bug Reporting:
- The issue here is not one that is a bug that can be seen in the syntax but is one that needs to be analysed from the LTS. As can be noted from the LTS none of the implemented actions at this point in the project like CreateCalendar, AddedToCalendar, and the DeleteCalendar action were being executed; only sign out and the non-implemented actions are executed. From the diagnosis it shows that the issue here is a matter of never being able to sign in and perform any of the TimeTree application functionalities, which we found was the fact that the signedup and signedin flags in the user process were initialised to be true; the combination of signedup, signedin, and signedout being true and true and false only leaves the option/branch of signing out, hence why signing out was the only implemented action



- Bug: signedUp and signedIn flags set to true initially, which lead to a useless model as shown in the LTS. The change lead to the following minimum LTS with useful actions:



## Version 5: Trial6\_4.Int

- This current version of the LNT model in similar fashion to the Trial6\_4.Int is significantly different from the previous ones. As hinted, it like the Trial6\_1.Int no longer has Event, Memo and Activity as processes but as lists in the Calendar process. This is to ensure a unique calendar keeps track of all events, memos and activities that correspond to it. By removing these processes, it also eliminates the need to have gates to pass the active or inactive state of Event, Memo and Activity. Therefore they have been replaced by variables maintained inside the process Calendar and User.
- It includes new processed UserManager and CalendarManager to manage concurrency and synchronisation.
- It includes new gates - SignIn, SignUp, SignOut and UpdateMemo, and renames some old ones - UpdateEvent to ChangeDateAndTimeOfEvent, Comment to CommentOnMemo and CommentOnEvent, View to ViewMemo, ViewEvent and ViewCalendar.
- As a result of a liveness model checking failure, it added a transition from a deadlocked state resulting in the system being deadlock free.

- Data Type & Channel Definitions:

```

module Trial6_4 with ==, !=, >= is
-----
-- Types
-----
type Event is Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat) end type
type Memo is Memo (ID: Nat, MemoContent: Nat) end type
type Activity is Activity (ID: Nat, CalendarID: Nat, EventUpdate: bool, MemoUpdate: bool, ObjectID:
Nat) end type

type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
-- type ListOfEvent is list of Event end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
-- type ListOfMemo is list of Memo end type
type ListOfActivity is list of Activity with empty, head, tail, member end type
-- type ListOfActivity is list of Activity end type

-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel
-----

```

- CalendarManager Process Definition:

```

process CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
CreateCalendar: N, DeleteCalendar: N2,
InviteToCalendar, JoinCalendar, LeaveCalendar,
Chat,
ViewCalendar, DisplayCalendar,
CalendarNotify: none,
AddedToCalendar: none,
Search, Result: none,
CreateEvent, DeleteEvent,
ChangeEventDateAndTime,
CommentOnEvent,
ViewEvent, DisplayEvent,
EventNotify,
CreateMemo, DeleteMemo,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo,
MemoNotify: none] is

```

- UserManager Process Definition:

```

process UserManager [SignUp, SignIn, SignOut: N,
CreateCalendar: N, DeleteCalendar: N2,
InviteToCalendar, JoinCalendar, LeaveCalendar,
Chat,
ViewCalendar, DisplayCalendar,
CalendarNotify: none,
AddedToCalendar: none,
Search, Result: none,
CreateEvent, DeleteEvent,
ChangeEventDateAndTime,
CommentOnEvent,
ViewEvent, DisplayEvent,
EventNotify,
CreateMemo, DeleteMemo,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo,
MemoNotify: none] is

```

- Calendar Process Definition:

```
process Calendar[CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
    CreateCalendarReceive: N, DeleteCalendarReceive: N2,
    InviteToCalendarReceive, JoinCalendarReceive, LeaveCalendarReceive,
    ChatReceive,
    ViewCalendarReceive, DisplayCalendar,
    CalendarNotify: none,
    AddedToCalendar: none,
    SearchReceive, Result: none,
    CreateEventReceive, DeleteEventReceive,
    ChangeEventDataAndTimeReceive,
    CommentOnEventReceive,
    ViewEventReceive, DisplayEvent,
    EventNotify,
    CreateMemoReceive, DeleteMemoReceive,
    CommentOnMemoReceive, UpdateMemoReceive,
    ViewMemoReceive, DisplayMemo,
    MemoNotify: none] (CalendarID: Nat, in var
    MakerID: Nat, in var
    ListOfUID: ListOfUserID, in var
    ListOfE: ListOfEvent, in var
    ListOfM: ListOfMemo, in var
    ListOfA: ListOfActivity) is
```

- User Process Definition:

```
process User[SignUp, SignIn, SignOut: N,
    CreateCalendar: N, DeleteCalendar: N2,
    InviteToCalendar, JoinCalendar, LeaveCalendar,
    Chat,
    ViewCalendar, DisplayCalendarReceive,
    CalendarNotifyReceive: none,
    AddedToCalendarReceive: none,
    Search, ResultReceive: none,
    CreateEvent, DeleteEvent,
    ChangeEventDataAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEventReceive,
    EventNotifyReceive,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemoReceive,
    MemoNotifyReceive: none] (UserID: Nat, in var ListOfCalID: ListOfCalendarID) is
```

- User updated implementation section:

```
[]
if (ActiveUser != 0) then
  if ((signedUp == true) and (signedIn == true) and (signedOut == false)) then
    SignOut(UserID);
    signedUp := true;
    signedIn := false;
    signedOut := true;
    ActiveUser := 0;
    select
      SignIn(UserID);
      signedUp := true;
      signedIn := true;
      signedOut := false;
      ActiveUser := UserID
    []
    null
  end select
end if
end if
[]
```

- Main Process Definition:

```
process Main[SignUp, SignIn, SignOut: N,
    CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
    CreateCalendar: N, DeleteCalendar: N2,
    InviteToCalendar, JoinCalendar, LeaveCalendar,
    Chat,
    ViewCalendar, DisplayCalendar,
    CalendarNotify: none,
    AddedToCalendar: none,
    Search, Result: none,
    CreateEvent, DeleteEvent,
    ChangeEventDateAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none] is
```

- Bug Output: No Bugs, some warnings (Trial6\_4 compiles)

```
rl03 434 $ lnt.open Trial6_4.lnt generator Trial6_4.bcg

lnt.open: checking ``Trial6_4.lnt``
./Trial6_4.lnt:200: warning: useless assignment to ListOfA
./Trial6_4.lnt:199: warning: useless assignment to ListOfM
./Trial6_4.lnt:198: warning: useless assignment to ListOfE
./Trial6_4.lnt:159: warning: "in var" parameter ListOfE overwritten before used (should be a local variable)
./Trial6_4.lnt:160: warning: "in var" parameter ListOfM overwritten before used (should be a local variable)
./Trial6_4.lnt:161: warning: "in var" parameter ListOfA overwritten before used (should be a local variable)
./Trial6_4.lnt:336: warning: "in var" parameter ListOfCalID never modified (should be an "in" parameter)

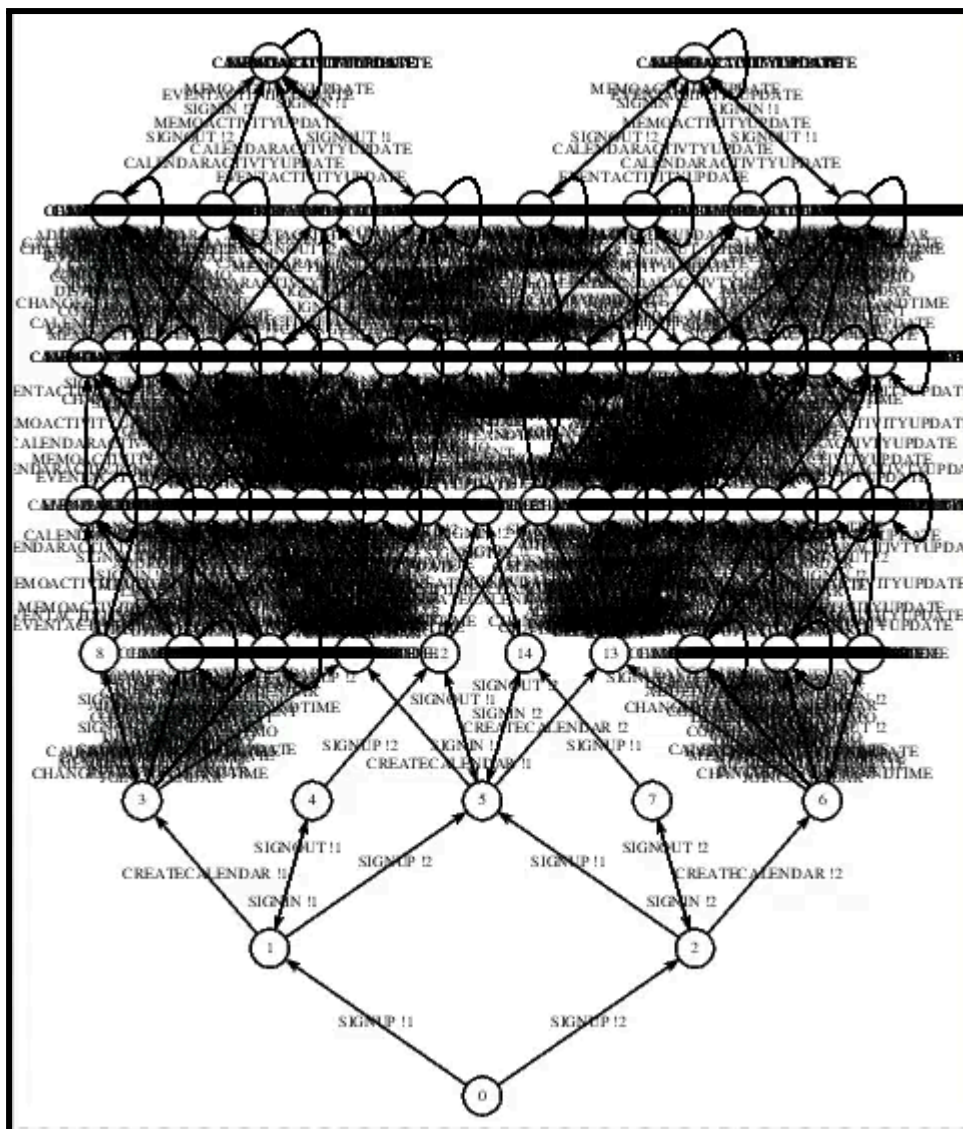
lnt.open: translating ``Trial6_4.lnt`` to ``/tmp/ruttkas_lnt.open_CssfWA/Trial6_4.lotos`` ...

lnt.open: calling lotos.open for ``Trial6_4.lotos`` ...
lotos.open: using ``(direct) caesar``
lotos.open: calling ``caesar.adt -debug -silent Trial6_4``
lotos.open: calling ``caesar -silent Trial6_4``
rl03 435 $
```

- As you can see the model is able to compile but gives warnings for both overwritten before being used variable assignments and unused variable assignments.
  - Both warnings are due to the unimplemented features on the model and will be addressed in the next report.
-

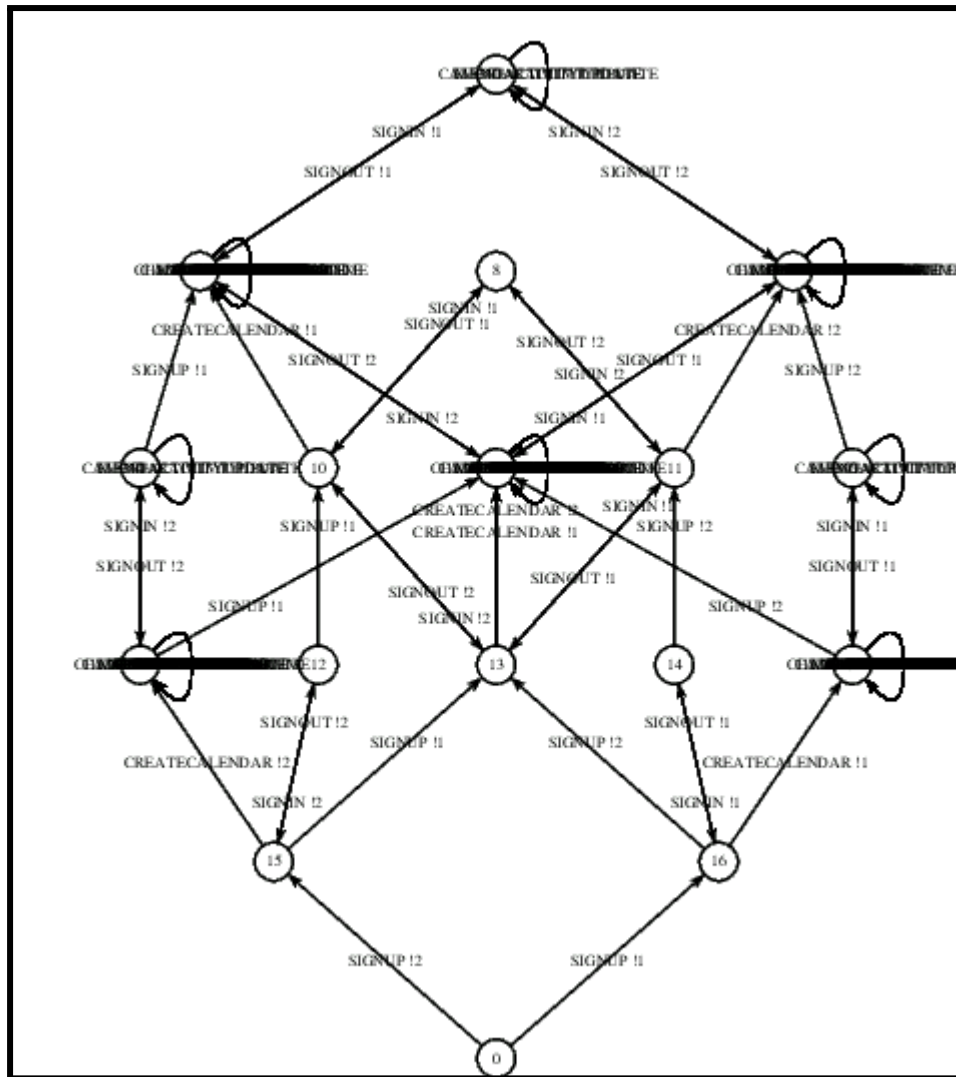


# Full LTS of Trial6\_4.Int





## Min LTS of Trial6\_4.lnt



## Version 6: Trial641.lnt

- Building upon previous LNT models, we complete the full functionality of signin, signup, signout, and added features. Bug Output: No Bugs, some warnings (Trial641 compiles). The model does compile with warnings which are caused by unimplemented features on the model and is addressed in the next LNT iteration of the model.

- Data Type & Channel Definitions:

```

module Trial641 with ==, !=, >= is
-----
-- Types
-----
type Event is Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat) end type
type Memo is Memo (ID: Nat, MemoContent: Nat) end type
type Activity is Activity (ID: Nat, CalendarID: Nat, EventUpdate: bool, MemoUpdate: bool, ObjectID: Nat) end type

type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
-- type ListOfEvent is list of Event end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
-- type ListOfMemo is list of Memo end type
type ListOfActivity is list of Activity with empty, head, tail, member end type
-- type ListOfActivity is list of Activity end type

-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel
channel N3 is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat) end channel

```

#### - CalendarManager Process Definition:

```

process CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
                        CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
                        InviteToCalendar: N3, InviteLinkToCalendar: N2,
                        JoinCalendar: N2, LeaveCalendar,
                        Chat,
                        ViewCalendar, DisplayCalendar,
                        CalendarNotify: none,
                        AddedToCalendar: none,
                        Search, Result: none,
                        CreateEvent, DeleteEvent,
                        ChangeEventDataAndTime,
                        CommentOnEvent,
                        ViewEvent, DisplayEvent,
                        EventNotify,
                        CreateMemo, DeleteMemo,
                        CommentOnMemo, UpdateMemo,
                        ViewMemo, DisplayMemo,
                        MemoNotify: none, Debug: any] is
par
  Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDataAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [1, 0, {}, {}, {}, {}, 2]
||
  Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDataAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [2, 0, {}, {}, {}, {}, 2]
end par
end process

```

- UserManager Process Definition:

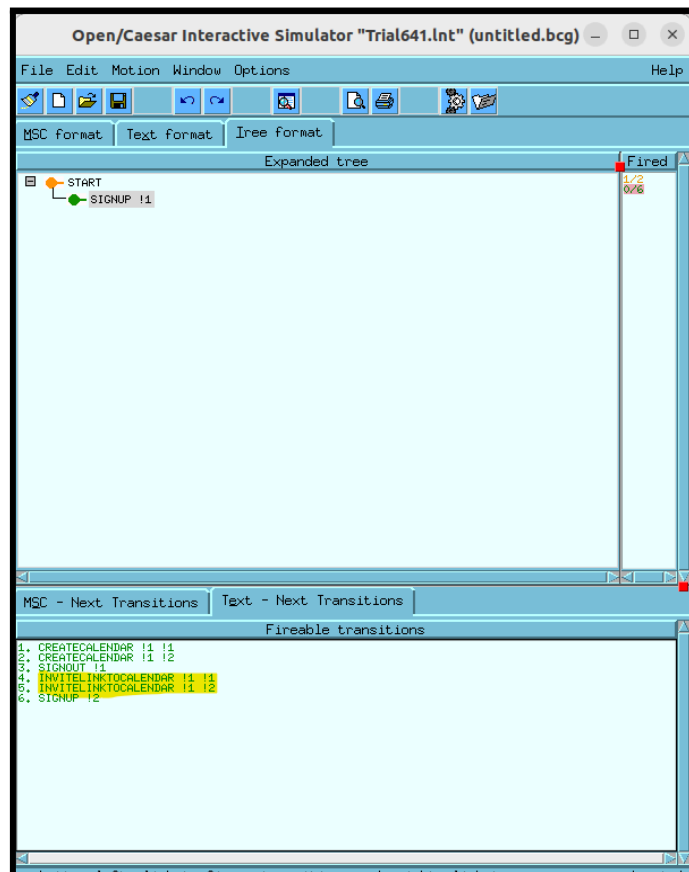
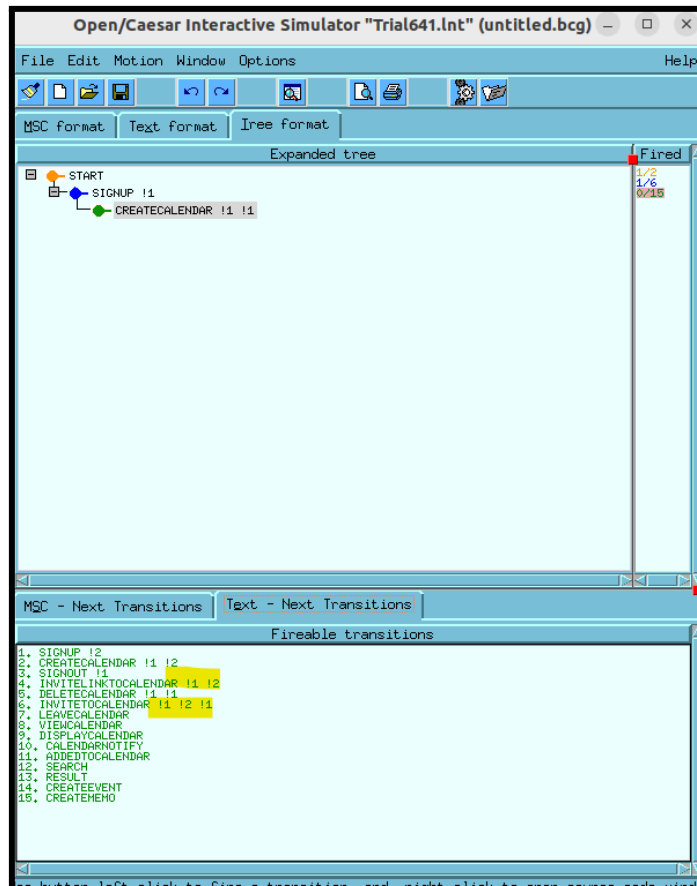
```

process UserManager [SignUp, SignIn, SignOut: N,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar,
    Chat, ViewCalendar, DisplayCalendar,
    CalendarNotify: none,
    AddedToCalendar: none,
    Search, Result: none,
    CreateEvent, DeleteEvent,
    ChangeEventDataAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none, Debug: any] is
    par Chat in
        User [SignUp, SignIn, SignOut,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDataAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [1, {}, {}] -- user 1 with no calendars currently
    ||
        User [SignUp, SignIn, SignOut,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDataAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [2, {}, {}] -- user 2 with no calendars currently
    end par
end process

```

- During the evolution of the model, problems were encountered with deadlocks where certain gates like InviteLinkToCalendar should always be executed after InviteToCalendar but that doesn't seem to be the case, likewise JoinCalendar appears in the labels but when running using OCIS it does not appear as a gate that should be accessible to be chosen for execution; it should appear at the very least as an optionable action after InviteLinkToCalendar is executed. Therefore, using the OCIS tool and debugging information about line numbers of the code where the error was occurring we were able to resolve the problem (InviteToCalendar-496, InviteToCalendarReceive-233, InviteLinkToCalendar-238, InviteLinkToCalendarReceive-505, JoinCalendar-522, JoinCalendarReceive-247).
- Screenshots of OCIS are shown below for clarification:

OCIS screenshots with highlighted section of selection error causing deadlock:



---

## Version 7: Trial641a0.Int

- Here we addressed several issues in the calendar code to ensure smooth functioning. Firstly, the order had to be corrected in the order of appearance where invitelinktocalendar was showing before invitetocalendar. Additionally, the problem of the code getting stuck during compilation had to be resolved. Although the compilation time is still around 10 minutes, the code now compiles without any issues.
- Next there was implementing the Invite System which was developed to enhance user interaction within the calendar application. Users are now able to send a maximum of 2 invites to others and receive a maximum of 2 invites from other users. Furthermore, users can only join the calendars corresponding to the invites they have received. This system ensures effective management of calendar invitations.
- There was also the matter of addressing capacity of the model and so we introduced a limit of 2 users per calendar. When a user creates a calendar, the available spots decrease from 2 to 1, indicating that one spot is taken. Similarly, when a non-creating user joins the calendar, the available spots decrement to 0, indicating that the calendar is full. If the creator deletes or leaves the calendar, the available spots increment back to 2. Likewise, when a non-creating user leaves the calendar, the available spots increment by one, indicating an available spot.
- Data Type & Channel Definitions:

```
module Trial641a0 with ==, !=, >= is
-----
-- Types
-----
type Event is Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat) end type
type Memo is Memo (ID: Nat, MemoContent: Nat) end type
type Activity is Activity (ID: Nat, CalendarID: Nat, EventUpdate: bool, MemoUpdate: bool, ObjectID:
Nat) end type

type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
-- type ListOfEvent is list of Event end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
-- type ListOfMemo is list of Memo end type
type ListOfActivity is list of Activity with empty, head, tail, member end type
-- type ListOfActivity is list of Activity end type

-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel
channel N3 is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat) end channel
```

- CalendarManager Process Definition:

```

process CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar,
    Chat,
    ViewCalendar, DisplayCalendar,
    CalendarNotify: none,
    AddedToCalendar: none,
    Search, Result: none,
    CreateEvent, DeleteEvent,
    ChangeEventDateAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none, Debug: any] is
    par
        Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDateAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [1, 0, {}, {}, {}, {}, 2]
    ||
        Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDateAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [2, 0, {}, {}, {}, {}, 2]
    end par
end process

```

- UserManager Process Definition:

```

process UserManager [SignUp, SignIn, SignOut: N,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar,
    Chat, ViewCalendar, DisplayCalendar,
    CalendarNotify: none,
    AddedToCalendar: none,
    Search, Result: none,
    CreateEvent, DeleteEvent,
    ChangeEventDateAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none, Debug: any] is
    par Chat in
        User [SignUp, SignIn, SignOut,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDateAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [1, {}, {}] -- user 1 with no calendars currently
    ||
        User [SignUp, SignIn, SignOut,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDateAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] [2, {}, {}] -- user 2 with no calendars currently
    end par
end process

```

---

## Version 8: Trial641a1.lnt

- This file added essential functions such as viewcalendar, displaycalendar, calendarnotify, search, and result. However, a significant bug hinders the compilation process, causing it to never finish. This issue is likely a result of the file containing an excessive number of states and transitions, overwhelming the compilation process, thus likely state explosion. As a consequence, the code fails to compile, severely impacting the functionality of the calendar-related features in the software.
- This compilation runtime bug needs to be debugged by carefully reviewing the code to identify the specific issue causing the compilation to hang. It was determined to be caused by the search and results features being added and their inherently complicated nature with multiple loops, potentially causing multiple deadlocks. Until this bug is addressed, the software's calendar-related functionalities will remain compromised, potentially affecting the user experience and overall usability of the software. Therefore, resolving this issue promptly is essential to maintain the software's integrity and functionality.
- The next version of Trial641a1.lnt involved making the nbActivations - an element in the search and results section. This value was altered to 1 instead of in order to resolve state explosion and make the model manageable in compilation time and viewing for verification purposes.
- This version of the code corrected for the nbActivations and built upon previous work with a total compilation time of 10 minutes, which is manageable. The number of states in the model was 275703 states, 1934726 transitions, 69 labels, 0 deadlocks and an average branching factor of 7.02.

```
kedi@kedi-VirtualBox:~$ bcg_info Trial641a1.bcg
./Trial641a1.bcg:
created by caesar
    275703 states
    1934726 transitions
    69 labels
    initial state: 0
    no deadlock state
    branching factor: average = 7.02, minimal = 2, maximal = 68
    67480 transition(s) with a hidden label ("i")
    non-deterministic behavior for:
```

- We then set everything to 1, so there is only 1 user, 1 id, 1 calendar, etcetera for all variables and changed nbActivations to 2 so that we got 645235 states, 45267870 transitions, 69 labels, 0 deadlocks and an average branching factor of 7.02.

```
kedi@kedi-VirtualBox:~$ bcg_info Trial641a1.bcg
./Trial641a1.bcg:
created by caesar
    645235 states
    45267870 transitions
    69 labels
    initial state: 0
    no deadlock state
    branching factor: average = 7.02, minimal = 2, maximal = 68
    151832 transition(s) with a hidden label ("i")
    non-deterministic behavior for:
```

- Lastly, in the debugging process we tried out setting everything to 2 and nbActiations to 1 and got 2724681 states, 19003286 transitions, 69 labels, 0 deadlocks and an average branching factor of 6.97.

```

kedi@kedi-VirtualBox:~$ bcg_info Trial641a1.bcg
./Trial641a1.bcg:
created by caesar
  2724681 states
  19003286 transitions
  69 labels
  initial state: 0
  no deadlock state
  branching factor: average = 6.97, minimal = 2, maximal = 68
  607344 transition(s) with a hidden label ("i")
  see deterministic behavior for:

```

- Data Type & Channel Definitions:

```

module Trial641a1 with get, set, ==, !=, >= is
-----
-- Types
-----
type Event is
  Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat)
end type
type Memo is
  Memo (ID: Nat, MemoContent: Nat)
end type
type Activity is
  Activity (ID: Nat, CalendarID: Nat, EventUpdate: bool, MemoUpdate: bool, ObjectID: Nat)
end type

type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
type ListOfActivity is list of Activity with empty, head, tail, member end type

-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel
channel N3 is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat) end channel
channel N2E is (NatNum1: Nat, NatNum2: Nat, EventNum1: Event) end channel

```



- CalendarManager Process Definition:

```

process CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar: N2,
    Chat: N3,
    ViewCalendar: N2, DisplayCalendar: N2,
    CalendarNotify: N3,
    AddedToCalendar: N2,
    Search: N3, Result: N2E,
    CreateEvent: N2E, DeleteEvent,
    ChangeEventDateAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none, Debug: any] is
par
    Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
        CreateCalendar, DeleteCalendar, RemovedFromCalendar,
        InviteToCalendar, InviteLinkToCalendar,
        JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        Search, Result,
        CreateEvent, DeleteEvent,
        ChangeEventDateAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo, UpdateMemo,
        ViewMemo, DisplayMemo,
        MemoNotify, Debug] (1, 0, {}, {}, {}, {}, 1, 2, 2)
||
    Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
        CreateCalendar, DeleteCalendar, RemovedFromCalendar,
        InviteToCalendar, InviteLinkToCalendar,
        JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        Search, Result,
        CreateEvent, DeleteEvent,
        ChangeEventDateAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo, UpdateMemo,
        ViewMemo, DisplayMemo,
        MemoNotify, Debug] (2, 0, {}, {}, {}, {}, 1, 2, 2)
end par
end process

```

- UserManager Process Definition:

```

process UserManager [SignUp, SignIn, SignOut: N,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar: N2,
    Chat: N3,
    ViewCalendar: N2, DisplayCalendar: N2,
    CalendarNotify: N3,
    AddedToCalendar: N2,
    Search: N3, Result: N2E,
    CreateEvent: N2E, DeleteEvent,
    ChangeEventDateAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none, Debug: any] is
par
    User [SignUp, SignIn, SignOut,
        CreateCalendar, DeleteCalendar, RemovedFromCalendar,
        InviteToCalendar, InviteLinkToCalendar,
        JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        Search, Result,
        CreateEvent, DeleteEvent,
        ChangeEventDateAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo, UpdateMemo,
        ViewMemo, DisplayMemo,
        MemoNotify, Debug] (1, {}, {}, 1, 1, 1, 1, 1, 1, 1, 1) -- user 1 with no calendars currently
||
    User [SignUp, SignIn, SignOut,
        CreateCalendar, DeleteCalendar, RemovedFromCalendar,
        InviteToCalendar, InviteLinkToCalendar,
        JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        Search, Result,
        CreateEvent, DeleteEvent,
        ChangeEventDateAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo, UpdateMemo,
        ViewMemo, DisplayMemo,
        MemoNotify, Debug] (2, {}, {}, 1, 1, 1, 1, 1, 1, 1, 1) -- user 2 with no calendars currently
end par
end process

```

- User to Calendar Process definition implementation:

```

-----
-- From User to Calendar (Basic)
if (ActiveCal == 0) and (MakerID == 0) and (nbActivations != 0) then
  var UserID: Nat, CalID: Nat in
  CreateCalendarReceive(?UserID, ?CalID) where CalID == CalendarID;
  -- calendar activated if not created and the CalID input is the ID of cal instance.
  -- maker of calendar must not exist prior to user creating calendar.
  if ((MakerID == 0) and (CalendarID == CalID)) then
    ActiveCal := CalendarID;
    MakerID := UserID;
    ListOfUID := cons(MakerID, ListOfUID);
    nbUsersLeftToJoin := nbUsersLeftToJoin - 1
  end if
end var
end if
[]
if (ActiveCal != 0) then
  if (ActiveCal == CalendarID) then
    var TCalID: Nat, TUserID: Nat in
    DeleteCalendarReceive(?TUserID, ?TCalID) where ((TCalID == CalendarID) and (TUserID == MakerID));
    if ((ActiveCal == TCalID) and (MakerID == TUserID)) then
      ActiveCal := 0;
      ListOfE := {};
      ListOfM := {};
      ListOfA := {};
      var CurrentList: ListOfUserID, CurrentUserID: Nat in
      CurrentList := ListOfUID;
      while (empty(CurrentList) == false) loop
        CurrentUserID := head(CurrentList);
        RemovedFromCalendar(CurrentUserID, TCalID);
        CurrentList := tail(CurrentList);
        nbUsersLeftToJoin := nbUsersLeftToJoin + 1
      end loop
    end var;
    -- Lina NEW: to limit number of Activation
    nbActivations := nbActivations - 1;
    MakerID := 0;
    ListOfUID := {};
    if nbActivations == 0 then
      i -- Lina NEW to avoid deadlock
    end if
  end if
end var
end if
[]
if (ActiveCal != 0) then
  if (ActiveCal == CalendarID) then
    var UserID: Nat, InvitedUserID: Nat, CalID: Nat in
    InviteToCalendarReceive(?UserID, ?InvitedUserID, ?CalID) where
      ((CalID != 0) and (CalID == CalendarID) and
      (UserID != InvitedUserID) and
      (member(UserID, ListOfUID) == true) and
      (member(InvitedUserID, ListOfUID) == false));
    if ((MakerID != 0) and (UserID != 0)) then
      InviteLinkToCalendar(InvitedUserID, CalID)
    end if
  end var
end if
end if
[]
if (ActiveCal != 0) then
  if (ActiveCal == CalendarID) then
    var UserID: Nat, CalID: Nat in
    JoinCalendarReceive(?UserID, ?CalID) where ((CalID == CalendarID) and (UserID != MakerID) and
      (nbUsersLeftToJoin > 0));
    if (member(UserID, ListOfUID) == false) then
      ListOfUID := cons(UserID, ListOfUID);
      AddedToCalendar(UserID, CalID);
      nbUsersLeftToJoin := nbUsersLeftToJoin - 1
    end if
  end var
end if
end if
[]

```

- Calendar to User Process definition implementation:

```

-- From Calendar to User (Basic)
[]
if (ActiveCal != 0) then
  if (ActiveCal == CalendarID) then
    var UserID: Nat, CalID: Nat, EventID: Nat in
      SearchReceive(?UserID, ?CalID, ?EventID) where {(member{UserID, ListOfUID}) and
                                                {CalID == CalendarID}};
    -- Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat)
    var CurrentList: ListOfEvent,
        CurrentEvent: Event,
        EventExists: bool,
        PlaceholderEvent: Event in
      CurrentList := ListOfE;
      EventExists := false;
      PlaceholderEvent := Event(0, 0, 0, 0, 0);
      while (empty(CurrentList) == false) loop
        CurrentEvent := head(CurrentList);
        if (CurrentEvent.ID == EventID) then
          EventExists := true;
          PlaceholderEvent := CurrentEvent
        end if;
        CurrentList := tail(CurrentList)
      end loop;
      if (EventExists == true) then
        Result(UserID, CalID, PlaceholderEvent)
      end if;
      if (EventExists == false) then
        PlaceholderEvent := Event(0, 0, 0, 0, 0);
        Result(UserID, CalID, PlaceholderEvent)
      end if
    end var
  end var
end if
end if
[]

-----
-- From User to Calendar (Event)
if (ActiveCal != 0) and (AllowedEvents > 0) then
  var UID: Nat, CalID: Nat, EventID: Nat, newEvent: Event in
    CreateEventReceive(?UID, ?CalID, ?newEvent) where {(CalID == CalendarID) and
                                                        {member(UID, ListOfUID) == true}};
  var CurrentList: ListOfEvent, CurrentEvent: Event, EventExists: bool in
    CurrentList := ListOfE;
    EventExists := false;

    while (empty(CurrentList) == false) loop
      CurrentEvent := head(CurrentList);
      if (CurrentEvent.ID == newEvent.ID) then
        EventExists := true
      end if;
      CurrentList := tail(CurrentList)
    end loop;

    if (EventExists == false) then
      Debug("event created");
      ListOfE := cons(newEvent, ListOfE);
      AllowedEvents := AllowedEvents - 1
    end if
  end var
end var
end if
[]
if ((ActiveCal != 0) and (ListOfE != {})) then
  DeleteEventReceive
end if
[]
if ((ActiveCal != 0) and (ListOfE != {})) then
  ChangeEventDataAndTimeReceive
end if
[]
if ((ActiveCal != 0) and (ListOfE != {})) then
  CommentOnEventReceive
end if
[]
if ((ActiveCal != 0) and (ListOfE != {})) then
  ViewEventReceive
end if

```

- The updated version of the code implements restrictions on the frequency of various actions including createevent, viewevent, commentonevent, changeevent, chat, viewcal, and search. These restrictions are designed to control the number of times these actions can be performed within a specified period. The implementation of these restrictions aims to optimize system resources, enhance user experience, and ensure fair usage of the application's functionalities.

## Version 9: Trial641a12.Int

- This version of the code built upon previous work with a total compilation time of 15 minutes. Trial641a12 aimed to diagnose an issue causing infinite compilation when introducing search and result functions into the code. To achieve this, a simplified version of the search and result functionality was created, removing all data. The objective was to reduce compilation time to 10-15 minutes for efficient diagnostic testing. Despite the simplification, the code still failed to compile within the desired time frame, indicating that the issue lies deeper than the complexity of the data. Further investigation is required to identify and address the root cause of the compilation problem. This trial provided valuable insights into the nature of the compilation issue, helping to narrow down potential areas for investigation. The diagnostic process will continue, with a focus on identifying any underlying structural or algorithmic issues within the code. Results from this trial will inform the next steps in resolving the compilation problem, ensuring the code's functionality and efficiency are restored.

- Data Type & Channel Definition:

```
module Trial641a12 with get, set, ==, !=, >= is
-----
-- Types
-----
type Event is
  Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat)
end type
type Memo is
  Memo (ID: Nat, MemoContent: Nat)
end type
type Activity is
  Activity (ID: Nat, CalendarID: Nat, EventUpdate: bool, MemoUpdate: bool, ObjectID: Nat)
end type

type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
type ListOfActivity is list of Activity with empty, head, tail, member end type

-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel
channel N3 is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat) end channel
channel N2E is (NatNum1: Nat, NatNum2: Nat, EventNum1: Event) end channel
```

- CalendarManager Process Definition:

```

process CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar: N2,
    Chat: N3,
    ViewCalendar: N2, DisplayCalendar: N2,
    CalendarNotify: N3,
    AddedToCalendar: N2,
    Search, Result,
    CreateEvent, DeleteEvent,
    ChangeEventDateAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none, Debug: any] is
    par
        Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDateAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] (1, 0, {}, {}, {}, {}, 2, 2)
    ||
        Calendar[CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
            CreateCalendar, DeleteCalendar, RemovedFromCalendar,
            InviteToCalendar, InviteLinkToCalendar,
            JoinCalendar, LeaveCalendar,
            Chat,
            ViewCalendar, DisplayCalendar,
            CalendarNotify,
            AddedToCalendar,
            Search, Result,
            CreateEvent, DeleteEvent,
            ChangeEventDateAndTime,
            CommentOnEvent,
            ViewEvent, DisplayEvent,
            EventNotify,
            CreateMemo, DeleteMemo,
            CommentOnMemo, UpdateMemo,
            ViewMemo, DisplayMemo,
            MemoNotify, Debug] (2, 0, {}, {}, {}, {}, 2, 2)
    end par
end process

```

- UserManager Process Definition:

```

process UserManager [SignUp, SignIn, SignOut: N,
                    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
                    InviteToCalendar: N3, InviteLinkToCalendar: N2,
                    JoinCalendar: N2, LeaveCalendar: N2,
                    Chat: N3,
                    ViewCalendar: N2, DisplayCalendar: N2,
                    CalendarNotify: N3,
                    AddedToCalendar: N2,
                    Search, Result,
                    CreateEvent, DeleteEvent,
                    ChangeEventDateAndTime,
                    CommentOnEvent,
                    ViewEvent, DisplayEvent,
                    EventNotify,
                    CreateMemo, DeleteMemo,
                    CommentOnMemo, UpdateMemo,
                    ViewMemo, DisplayMemo,
                    MemoNotify: none, Debug: any] is
  par
    User [SignUp, SignIn, SignOut,
          CreateCalendar, DeleteCalendar, RemovedFromCalendar,
          InviteToCalendar, InviteLinkToCalendar,
          JoinCalendar, LeaveCalendar,
          Chat,
          ViewCalendar, DisplayCalendar,
          CalendarNotify,
          AddedToCalendar,
          Search, Result,
          CreateEvent, DeleteEvent,
          ChangeEventDateAndTime,
          CommentOnEvent,
          ViewEvent, DisplayEvent,
          EventNotify,
          CreateMemo, DeleteMemo,
          CommentOnMemo, UpdateMemo,
          ViewMemo, DisplayMemo,
          MemoNotify, Debug] (1, {}, {}, 2, 2) -- user 1 with no calendars currently
  ||
    User [SignUp, SignIn, SignOut,
          CreateCalendar, DeleteCalendar, RemovedFromCalendar,
          InviteToCalendar, InviteLinkToCalendar,
          JoinCalendar, LeaveCalendar,
          Chat,
          ViewCalendar, DisplayCalendar,
          CalendarNotify,
          AddedToCalendar,
          Search, Result,
          CreateEvent, DeleteEvent,
          ChangeEventDateAndTime,
          CommentOnEvent,
          ViewEvent, DisplayEvent,
          EventNotify,
          CreateMemo, DeleteMemo,
          CommentOnMemo, UpdateMemo,
          ViewMemo, DisplayMemo,
          MemoNotify, Debug] (2, {}, {}, 2, 2) -- user 2 with no calendars currently
  end par
end process

```

- User Process definition and implementation:

```

process User[SignUp, SignIn, SignOut: N,
  CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendarReceive: N2,
  InviteToCalendar: N3, InviteLinkToCalendarReceive: N2,
  JoinCalendar: N2, LeaveCalendar: N2,
  Chat: N3,
  ViewCalendar: N2, DisplayCalendarReceive: N2,
  CalendarNotifyReceive: N3,
  AddedToCalendarReceive: N2,
  Search, ResultReceive,
  CreateEvent, DeleteEvent,
  ChangeEventDataAndTime,
  CommentOnEvent,
  ViewEvent, DisplayEventReceive,
  EventNotifyReceive,
  CreateMemo, DeleteMemo,
  CommentOnMemo, UpdateMemo,
  ViewMemo, DisplayMemoReceive,
  MemoNotifyReceive: none, Debug: any] (UserID: Nat, in var
  ListOfCalID: ListOfCalendarID, in var
  ListOfInvitedCalID: ListOfCalendarID, in var
  invitesLeftToSend: Nat, in var
  invitesLeftToGet: Nat) is
var signedUp: bool, signedIn: bool, signedOut: bool, ActiveUser: Nat in

-- user created in user manager essentially means the user is signed up & ultimately signed
-- in the TimeTree Application; create implies signed up which further implies signed in.
signedUp := false;
signedIn := false;
signedOut := false;
ActiveUser := 0;
loop
  select
    -- From User to none
    if (ActiveUser == 0) then
      if ((signedUp == false) and (signedIn == false) and (signedOut == false)) then
        SignUp(UserID);
        signedUp := true;
        signedIn := true;
        signedOut := false;
        ActiveUser := UserID
      end if
    end if
    []
    if (ActiveUser == 0) then
      if ((signedUp == true) and (signedIn == false) and (signedOut == false)) then
        SignIn(UserID);
        signedUp := true;
        signedIn := true;
        signedOut := false;
        ActiveUser := UserID
      end if
    end if
    []
    if (ActiveUser != 0) then
      if ((signedUp == true) and (signedIn == true) and (signedOut == false)) then
        SignOut(UserID);
        signedUp := true;
        signedIn := false;
        signedOut := true;
        ActiveUser := 0;
        select
          SignIn(UserID);
          signedUp := true;
          signedIn := true;
          signedOut := false;
          ActiveUser := UserID
        []
        null
      end select
    end if
  end select
end if
[]

```

```

if (signedIn == true) then
  if (ActiveUser == UserID) then
    var CalID: Nat, UID: Nat, InvitedUID: Nat in
      CalID := any Nat where ((CalID == 1) or (CalID == 2)) and ((member(CalID, ListOfCalID)) == true));
      UID := ActiveUser;
      InvitedUID := any Nat where ((InvitedUID == 1) or (InvitedUID == 2)) and (InvitedUID != UID);
      if (invitesLeftToSend > 0) then
        InviteToCalendar(UID, InvitedUID, CalID);
        invitesLeftToSend := invitesLeftToSend - 1;
      end if
    end var
  end if
end if
[]
if (signedIn == true) then
  if (ActiveUser == UserID) then
    var CalID: Nat, UID: Nat in
      InviteLinkToCalendarReceive(?UID, ?CalID) where
        ((UID == UserID) and
         ((member(CalID, ListOfCalID)) == false) and
         ((CalID == 1) or (CalID == 2)) and
         (invitesLeftToGet > 0));
      ListOfInvitedCalID := cons(CalID, ListOfInvitedCalID);
      invitesLeftToGet := invitesLeftToGet - 1;
    end var
  end if
end if
[]
if (signedIn == true) then
  if (ActiveUser == UserID) then
    if (empty(ListOfInvitedCalID) == false) then
      var CalIDToJoin: Nat, UID: Nat in
        CalIDToJoin := any Nat where (member(CalIDToJoin, ListOfInvitedCalID));
        UID := ActiveUser;
        if (member(CalIDToJoin, ListOfCalID) == false) then
          JoinCalendar(UID, CalIDToJoin);
        end if
      end var
    end if
  end if
end if
[]
if (signedIn == true) then
  if (ActiveUser == UserID) then
    var UID: Nat, CalIDToLeave: Nat in
      UID := ActiveUser;
      CalIDToLeave := any Nat where (member(CalIDToLeave, ListOfCalID));
      LeaveCalendar(UID, CalIDToLeave);
    end var
  end if
end if
[]
if (signedIn == true) then
  if (ActiveUser == UserID) then
    var UID: Nat, CalID: Nat, MessageContent: Nat in
      UID := ActiveUser;
      CalID := any Nat where (member(CalID, ListOfCalID));
      MessageContent := UID;
      Chat(UID, CalID, MessageContent);
    end var
  end if
end if
[]
if (signedIn == true) then
  if (ActiveUser == UserID) then
    var UID: Nat, CalID: Nat in
      UID := ActiveUser;
      CalID := any Nat where (member(CalID, ListOfCalID));
      ViewCalendar(UID, CalID);
    end var
  end if
end if
[]

```



- Main Process:

```

process Main[SignUp, SignIn, SignOut: N,
    CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: none,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar: N2,
    Chat: N3,
    ViewCalendar: N2, DisplayCalendar: N2,
    CalendarNotify: N3,
    AddedToCalendar: N2,
    Search, Result,
    CreateEvent, DeleteEvent,
    ChangeEventDataAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify: none, Debug: any] is
par
    CreateCalendar, DeleteCalendar, RemovedFromCalendar,
    InviteToCalendar, InviteLinkToCalendar,
    JoinCalendar, LeaveCalendar,
    Chat,
    ViewCalendar, DisplayCalendar,
    CalendarNotify,
    AddedToCalendar,
    Search, Result,
    CreateEvent, DeleteEvent,
    ChangeEventDataAndTime,
    CommentOnEvent,
    ViewEvent, DisplayEvent,
    EventNotify,
    CreateMemo, DeleteMemo,
    CommentOnMemo, UpdateMemo,
    ViewMemo, DisplayMemo,
    MemoNotify in
    UserManager [SignUp, SignIn, SignOut,
        CreateCalendar, DeleteCalendar, RemovedFromCalendar,
        InviteToCalendar, InviteLinkToCalendar,
        JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        Search, Result,
        CreateEvent, DeleteEvent,
        ChangeEventDataAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo, UpdateMemo,
        ViewMemo, DisplayMemo,
        MemoNotify, Debug]
||
    CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
        CreateCalendar, DeleteCalendar, RemovedFromCalendar,
        InviteToCalendar, InviteLinkToCalendar,
        JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        Search, Result,
        CreateEvent, DeleteEvent,
        ChangeEventDataAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo, UpdateMemo,
        ViewMemo, DisplayMemo,
        MemoNotify, Debug]
end par
end process

```

## Version 10: Trial641a2.lnt

- This version of the code built upon previous work with an approximate average compilation time of 5 to 10 minutes. The number of states in the model was 719083 states, 8141145 transitions, 141 labels, 0 deadlocks and an average branching factor of 11.32.

```

lotos.open: calling ``caesar -silent Trial641a2''
rl09 514 $ bcg_info Trial641a2.bcg
./Trial641a2.bcg:
created by caesar
  719083 states
  8141145 transitions
  141 labels
  initial state: 0
  no deadlock state
  branching factor: average = 11.32, minimal = 2, maximal = 59
  7464 transition(s) with a hidden label ("i")
  non-deterministic behavior for:
    label "SIGNUP !1" at state(s): 26 85 96 98 249 251 252 ... (8164 states in total)

```

- All event-related gates have been implemented, with the exception of eventnotify and eventactivityupdate. Additionally, the second calendar has been commented out, along with event-related process parameters. Despite these modifications, the code still compiles within 5-10 minutes. These changes were made to streamline the code and optimize its performance. The implementation of event-related gates enhances the functionality of the application, providing users with a comprehensive set of features. By commenting out the second calendar and related process parameters, the code's complexity is reduced, contributing to faster compilation times. Overall, these adjustments contribute to the efficient operation and maintenance of the codebase. Further optimizations may be explored to improve performance and usability.

#### - Data Type & Channel Definition:

```

module Trial641a2 with get, set, ==, !=, >= is
-----
-- Types
-----
type Event is
  Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat)
end type
type Memo is
  Memo (ID: Nat, MemoContent: Nat)
end type
type Activity is
  Activity (ID: Nat, CalendarID: Nat, EventUpdate: bool, MemoUpdate: bool, ObjectID: Nat)
end type

type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
type ListOfActivity is list of Activity with empty, head, tail, member end type

-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel
channel N3 is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat) end channel
channel N2E is (NatNum1: Nat, NatNum2: Nat, EventNum1: Event) end channel
channel N3M is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat, Message: string) end channel

```

- CalendarManager Process Definition:

```

process CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: any,
CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
InviteToCalendar: N3, InviteLinkToCalendar: N2,
JoinCalendar: N2, LeaveCalendar: N2,
Chat: N3,
ViewCalendar: N2, DisplayCalendar: N2,
CalendarNotify: N3,
AddedToCalendar: N2,
Search: N3, Result: N2E,
CreateEvent: N2E, DeleteEvent: N3,
ChangeEventDateAndTime: N2E,
CommentOnEvent: N3M,
ViewEvent: N3, DisplayEvent: N2E,
EventNotify: none,
CreateMemo: N3, DeleteMemo: N3,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo: none,
MemoNotify: N3, Debug: any] is
--par
  Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
CreateCalendar, DeleteCalendar, RemovedFromCalendar,
InviteToCalendar, InviteLinkToCalendar,
JoinCalendar, LeaveCalendar,
Chat,
ViewCalendar, DisplayCalendar,
CalendarNotify,
AddedToCalendar,
Search, Result,
CreateEvent, DeleteEvent,
ChangeEventDateAndTime,
CommentOnEvent,
ViewEvent, DisplayEvent,
EventNotify,
CreateMemo, DeleteMemo,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo,
MemoNotify, Debug] (1, 0, {}, {}, {}, {}, 1, 2, 2)
--||
  Calendar[CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
CreateCalendar, DeleteCalendar, RemovedFromCalendar,
InviteToCalendar, InviteLinkToCalendar,
JoinCalendar, LeaveCalendar,
Chat,
ViewCalendar, DisplayCalendar,
CalendarNotify,
AddedToCalendar,
Search, Result,
CreateEvent, DeleteEvent,
ChangeEventDateAndTime,
CommentOnEvent,
ViewEvent, DisplayEvent,
EventNotify,
CreateMemo, DeleteMemo,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo,
MemoNotify, Debug] (2, 0, {}, {}, {}, {}, 1, 2, 2)
--end par

```

- UserManager Process Definition:

```

process UserManager [SignUp, SignIn, SignOut: N,
CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
InviteToCalendar: N3, InviteLinkToCalendar: N2,
JoinCalendar: N2, LeaveCalendar: N2,
Chat: N3,
ViewCalendar: N2, DisplayCalendar: N2,
CalendarNotify: N3,
AddedToCalendar: N2,
Search: N3, Result: N2E,
CreateEvent: N2E, DeleteEvent: N3,
ChangeEventDateAndTime: N2E,
CommentOnEvent: N3M,
ViewEvent: N3, DisplayEvent: N2E,
EventNotify: none,
CreateMemo: N3, DeleteMemo: N3,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo: none,
MemoNotify: N3, Debug: any] is
  par
    User [SignUp, SignIn, SignOut,
CreateCalendar, DeleteCalendar, RemovedFromCalendar,
InviteToCalendar, InviteLinkToCalendar,
JoinCalendar, LeaveCalendar,
Chat,
ViewCalendar, DisplayCalendar,
CalendarNotify,
AddedToCalendar,
Search, Result,
CreateEvent, DeleteEvent,
ChangeEventDateAndTime,
CommentOnEvent,
ViewEvent, DisplayEvent,
EventNotify,
CreateMemo, DeleteMemo,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo,
MemoNotify, Debug] (1, {}, {}, 1, 1) -- user 1 with no calendars currently
  ||
    User [SignUp, SignIn, SignOut,
CreateCalendar, DeleteCalendar, RemovedFromCalendar,
InviteToCalendar, InviteLinkToCalendar,
JoinCalendar, LeaveCalendar,
Chat,
ViewCalendar, DisplayCalendar,
CalendarNotify,
AddedToCalendar,
Search, Result,
CreateEvent, DeleteEvent,
ChangeEventDateAndTime,
CommentOnEvent,
ViewEvent, DisplayEvent,
EventNotify,
CreateMemo, DeleteMemo,
CommentOnMemo, UpdateMemo,
ViewMemo, DisplayMemo,
MemoNotify, Debug] (2, {}, {}, 1, 1) -- user 2 with no calendars currently
  end par
end process

```

- While working on this model, Trials x2\_2.lnt and Trialx2.lnt were produced where search and memocreate have been removed. This modification aims to simplify the codebase and streamline the decision-making process within the application. By focusing solely on these two conditions, the code becomes more concise and easier to manage. Additionally, this change reduces the complexity of the code, potentially improving its performance and maintainability. It is anticipated that these adjustments will enhance the overall efficiency and reliability of the application. Further testing and optimization may be conducted to ensure the desired outcomes are achieved.

- The result of manipulations to optimise the code base resulted in the model having 197691 states, 1727394 transitions, 74 labels, 0 initial states and no deadlocks, with an average branching factor of 8.74.

```

kedi@kedi-VirtualBox:~$ bcg_info Trialx2_2.bcg
./Trialx2_2.bcg:
created by caesar
  197691 states
  1727394 transitions
  74 labels
  initial state: 0
  no deadlock state
  branching factor: average = 8.74, minimal = 2, maximal = 45
  6132 transition(s) with a hidden label ("i")
  non-deterministic behavior for:

```

---

## Version 11: Trialx4.Int

- This version of the model contains all processes, definitions and functions as per the refined architecture originally mapped out in this report and can be compiled in under 2 hours. It contains 4553994 states, 13904868 transitions, 34679 labels, 0 initial states and 2423616 deadlocks, with an average branching factor of 3.05. This version of the LNT was compiled for 6 hours and it was still running, so we stopped it due to lack of time and computing resources and the following information regarding Trialx4 is based on the compiled file.

```

rl04 504 $ bcg_info Trialx4.bcg
./Trialx4.bcg:
created by caesar
  4553994 states
  13904868 transitions
  34679 labels
  initial state: 0
  list of deadlock state(s): 2130378 2130379 2130380 2130381 2130382 21303
83 2130384 ... (2423616 states in total)
  branching factor: average = 3.05, minimal = 0, maximal = 35
  40852 transition(s) with a hidden label ("i")
  non-deterministic behavior for:
    label "SIGNUPI" at state(s): 25 79 87 196 211 213 214 ... (209843 s
tates in total)

```

- The current version of the model represents the culmination of the refined architecture outlined in this report. It includes all final architecture processes, namely MemoNotify, InviteLinkToCalendar, RemovedFromCalendar, and DisplayActivity, as illustrated in Figure 4. The model has been designed to compile and we ran it for 6 hours and it was still compiling. hours, making it efficient for development and testing purposes. With 4553994 states, 13904868 transitions, 34679 labels, 2423616 deadlocks, and an average branching factor of 3.05, this implementation provides a detailed and comprehensive representation of the system.

- The main features of the model include the full implementation of final architecture processes and its ability to compile within the specified timeframe. The large LTS size, though containing a significant number of deadlocks, ensures thorough testing and analysis. Abstraction and modelling choices have been made with careful consideration to ensure an accurate representation of the system. Utilising the LNT Model has enabled a detailed depiction of the architecture, allowing for efficient compilation and analysis.

- Modelling challenges primarily revolved around managing the vast number of states and transitions while ensuring the accurate representation of all architecture processes. Additionally, dealing with a high number of deadlocks required careful consideration and optimization strategies to maintain model efficiency.

- Data Type & Channel Definition:

```
module Trialx4 with get, set, ==, !=, >= is
-----
-- Types
-----
type Event is
  Event (ID: Nat, Date: Nat, Year: Nat, TimeHour: Nat, TimeMin: Nat)
end type
type Memo is
  Memo (ID: Nat, MemoContent: Nat)
end type
type Activity is
  Activity (EventUpdate: bool, MemoUpdate: bool, ActivityInfo: String)
end type

type ListOfCalendarID is list of Nat with empty, head, tail, member end type
type ListOfUserID is list of Nat with empty, head, tail, member end type
type ListOfEvent is list of Event with empty, head, tail, member end type
type ListOfMemo is list of Memo with empty, head, tail, member end type
type ListOfActivity is list of Activity with empty, head, tail, member end type
-----
-- Channels
-----
channel N is (NatNum: Nat) end channel
channel N2 is (NatNum1: Nat, NatNum2: Nat) end channel
channel N3 is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat) end channel
channel N2E is (NatNum1: Nat, NatNum2: Nat, EventNum1: Event) end channel
channel N3M is (NatNum1: Nat, NatNum2: Nat, NatNum3: Nat, Message: String) end channel
channel N2L is (NatNum1: Nat, NatNum2: Nat, ListOfA: ListOfActivity) end channel
channel NMNMN is (NatNum1: Nat, Message1: string, NatNum2: Nat, Message2: string, NatNum3: Nat) end channel
```

- CalendarManager Process Definition:

```
process CalendarManager [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate: any,
    CreateCalendar: N2, DeleteCalendar: N2, RemovedFromCalendar: N2,
    InviteToCalendar: N3, InviteLinkToCalendar: N2,
    JoinCalendar: N2, LeaveCalendar: N2,
    Chat: N3,
    ViewCalendar: N2, DisplayCalendar: N2,
    CalendarNotify: N3,
    AddedToCalendar: N2,
    Search: N3, Result: N2E,
    CreateEvent: N2E, DeleteEvent: N3,
    ChangeEventDateAndTime: N2E,
    CommentOnEvent: N3M,
    ViewEvent: N3, DisplayEvent: N2E,
    EventNotify: NMNMN,
    CreateMemo: N3, DeleteMemo: N3,
    CommentOnMemo: N3M, UpdateMemo: N3,
    ViewMemo: N3, DisplayMemo: N3,
    MemoNotify: N3, Debug: any,
    ViewActivity: N2, DisplayActivity: N2L] is
--par
    Calendar [CalendarActivityUpdate, EventActivityUpdate, MemoActivityUpdate,
        CreateCalendar, DeleteCalendar, RemovedFromCalendar,
        InviteToCalendar, InviteLinkToCalendar,
        JoinCalendar, LeaveCalendar,
        Chat,
        ViewCalendar, DisplayCalendar,
        CalendarNotify,
        AddedToCalendar,
        Search, Result,
        CreateEvent, DeleteEvent,
        ChangeEventDateAndTime,
        CommentOnEvent,
        ViewEvent, DisplayEvent,
        EventNotify,
        CreateMemo, DeleteMemo,
        CommentOnMemo, UpdateMemo,
        ViewMemo, DisplayMemo,
        MemoNotify, Debug,
        ViewActivity, DisplayActivity] (1, 0, {}, {}, {}, {}, 1, 2, 2)
```

# Model Checking

Description in English and MCL formalization of all your properties, discussion on specific choices for formalizing the properties as you did, properties formalization challenges

## Description of Properties

The following are several properties specified in English and Model Checking Language (MCL) for our LNT Model of TimeTree:

1. Liveness:
  - Liveness property expressing the existence of a sequence leading to an action.
2. Safety:
  - Safety expresses that something bad never happens.
3. Fairness:
  - Fairness property expresses reachability of actions by considering only fair execution sequences.

## Tested Properties

In the order the properties appear in the Demo.svl file, here are their property identifiers, descriptions, and their results from running the svl Demo.svl command in the terminal.

Property ID	Description	Result
PROPERTY_1_ORIGINAL_WITH_USER1	Safety: user1 cannot sign up immediately after they just have.	PASS
PROPERTY_2_ORIGINAL_WITH_USER2	Safety: user2 cannot sign up immediately after they just have.	PASS
PROPERTY_3_ORIGINAL_WITH_USER1	Safety: user1 cannot sign in immediately after they just have.	PASS
PROPERTY_4_ORIGINAL_WITH_USER2	Safety: user2 cannot sign in immediately after they just have.	PASS
PROPERTY_5_ORIGINAL_WITH_USER1	Safety: user1 cannot sign out immediately after they just have.	PASS



PROPERTY_6_ORIGINAL_WITH_USER2	Safety: user2 cannot sign out immediately after they just have.	PASS
PROPERTY_7_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot use any application functionality without being signed up or signed in.	PASS
PROPERTY_7v2_USER1	Safety: user 1 cannot use any application functionality without being signed up or signed in.	PASS
PROPERTY_7v3_USER2	Safety: user 2 cannot use any application functionality without being signed up or signed in.	PASS
PROPERTY_8_ORIGINAL_WITH_GENERAL_DATA	Safety: A user must have signed up in order to have signed out.	PASS
PROPERTY_8v2_USER1	Safety: user 1 must have signed up in order to have signed out.	FAIL
PROPERTY_8v3_USER2	Safety: user 2 must have signed up in order to have signed out.	FAIL
PROPERTY_10_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot create calendar without signing up or in.	PASS
PROPERTY_10v2_USER1	Safety: user 1 cannot create calendar without signing up or in.	FAIL
PROPERTY_10v3_USER2	Safety: user 2 cannot create calendar without signing up or in.	FAIL

PROPERTY_11_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot delete calendar without signing up or in.	PASS
PROPERTY_11v2_USER1	Safety: user 1 cannot delete calendar without signing up or in.	FAIL
PROPERTY_11v3_USER2	Safety: user 2 cannot delete calendar without signing up or in.	FAIL
PROPERTY_12_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot delete calendar without it being created.	PASS
PROPERTY_12v2_USER1	Safety: user 1 cannot delete calendar without it being created.	FAIL
PROPERTY_12v3_USER2	Safety: user 2 cannot delete calendar without it being created.	FAIL
PROPERTY_13_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot create an event or a memo without calendar being created.	PASS
PROPERTY_13v2_USER1	Safety: user 1 cannot create an event or a memo without calendar being created.	FAIL
PROPERTY_13v3_USER2	Safety: user 2 cannot create an event or a memo without calendar being created.	FAIL
PROPERTY_15_ORIGINAL_WITH_GENERAL_DATA	Liveness: Should be able for some user to sign up followed by some user signing up immediately after.	PASS

PROPERTY_15v2_USER2	Liveness: Should be able for user 2 to sign up followed by user 1 signing up immediately after.	PASS
PROPERTY_15v3_USER1	Liveness: Should be able for user 1 to sign up followed by user 2 signing up immediately after.	PASS
PROPERTY_16_ORIGINAL_WITH_GENERAL_DATA	Liveness: Should be able for some user to sign in followed by some user signing in immediately after.	PASS
PROPERTY_16v2_USER2	Liveness: Should be able for user 2 to sign in immediately after following user 1 signing in.	PASS
PROPERTY_16v3_USER1	Liveness: Should be able for user 1 to immediately sign in after following user 2 signing in.	PASS
PROPERTY_17_ORIGINAL_WITH_GENERAL_DATA	Liveness: Should be able for some user to sign out followed by some user signing out immediately after.	PASS
PROPERTY_17v2_USER2	Liveness: Should be able for user 2 to sign out followed by user 1 signing out immediately after.	PASS

PROPERTY_17v3_USER1	Liveness: Should be able for user 1 to sign out followed by user 2 signing out immediately after.	PASS
PROPERTY_20_ORIGINAL_WITH_GENERAL_DATA	Liveness: Any user leads to sign up after a finite number of actions	PASS
PROPERTY_20v2_USER1	Liveness: user 1 leads to sign up after a finite number of actions	FAIL
PROPERTY_20v3_USER2	Liveness: user 2 leads to sign up after a finite number of actions	FAIL
PROPERTY_21_ORIGINAL_NO_DATA	Liveness: Display calendar is inevitable if user selects to view calendar.	FAIL
PROPERTY_21v2_WITH_GENERAL_DATA	Liveness: Display calendar is inevitable if user selects to view calendar.	FAIL
PROPERTY_21v3_USER1	Liveness: Display calendar 1 is inevitable if user 1 selects to view calendar 1.	FAIL
PROPERTY_21v4_USER2	Liveness: Display calendar 1 is inevitable if user 2 selects to view calendar 1.	FAIL
PROPERTY_22_ORIGINAL_WITH_GENERAL_DATA	Liveness: Display event is inevitable if user selects to view event.	FAIL

PROPERTY_22v2_USER1	Liveness: Display event 1 is inevitable if user 1 selects to view event 1.	FAIL
PROPERTY_22v3_USER2	Liveness: Display event 1 is inevitable if user 2 selects to view event 1.	FAIL
PROPERTY_23_ORIGINAL_NO_DATA	Liveness: Display memo is inevitable if user selects to view memo.	FAIL
PROPERTY_23v2_WITH_GENERAL_DATA	Liveness: Display memo is inevitable if user selects to view memo.	FAIL
PROPERTY_23v3_USER1	Liveness: Display memo 1 is inevitable if user selects 1 to view memo 1.	FAIL
PROPERTY_23v4_USER2	Liveness: Display memo 1 is inevitable if user selects 2 to view memo 1.	FAIL
PROPERTY_24_ORIGINAL_NO_DATA	Liveness: Search result is inevitably returned to user when they submit search.	FAIL
PROPERTY_24v2_WITH_GENERAL_DATA	Liveness: Search result is inevitably returned to user when they submit search.	FAIL
PROPERTY_24v3_USER1	Liveness: Search result is inevitably returned to user 1 when they submit search for event 1 in calendar 1.	FAIL

PROPERTY_24v4_USER2	Liveness: Search result is inevitably returned to user 2 when they submit search for event 1 in calendar 1.	FAIL
PROPERTY_25_ORIGINAL_WITH_GENERAL_DATA	Fairness: After user completes signing up it is fairly possible to sign out*.	FAIL
PROPERTY_26_ORIGINAL_WITH_DATA_USER 1	Fairness: After user1 completes signing up it is fairly possible to sign out*.	FAIL
PROPERTY_26a_USER1	Fairness: After user1 completes signing up it is fairly possible to sign out*.	PASS
PROPERTY_27_ORIGINAL_WITH_DATA_USER 2	Fairness: After user2 completes signing up it is fairly possible to sign out*.	FAIL
PROPERTY_27a_USER2	Fairness: After user2 completes signing up it is fairly possible to sign out*.	PASS
PROPERTY_28_ORIGINAL_WITH_GENERAL_DATA	Fairness: After user completes signing in it is fairly possible to sign out*.	FAIL
PROPERTY_28a_WITH_GENERAL_DATA	Fairness: After user completes signing in it is fairly possible to sign out*.	PASS
PROPERTY_29_ORIGINAL_WITH_DATA_USER 1	Fairness: After user1 completes signing in it is fairly possible to sign out*.	FAIL

PROPERTY_29a_USER1	Fairness: After user1 completes signing in it is fairly possible to sign out*.	PASS
PROPERTY_29b_USER1	Fairness: After user1 completes signing in it is fairly possible to sign out*.	FAIL
PROPERTY_30_ORIGINAL_WITH_DATA_USER2	Fairness: After user2 completes signing in it is fairly possible to sign out*.	FAIL
PROPERTY_30a_USER2	Fairness: After user2 completes signing in it is fairly possible to sign out*.	FAIL
PROPERTY_30b_USER2	Fairness: After user2 completes signing in it is fairly possible to sign out*.	FAIL
PROPERTY_31_ORIGINAL_NO_DATA	Fairness: After sending a chat, all fair execution sequences will lead to a calendar notification.	FAIL
PROPERTY_31v2_WITH_GENERAL_DATA	Fairness: After sending a chat, all fair execution sequences will lead to a calendar notification.	FAIL
PROPERTY_31v3_USER1	Fairness: After user 1 sends a chat to calendar 1, all fair execution sequences will lead to a calendar 1 notification.	FAIL

PROPERTY_31v4_USER2	Fairness: After user 2 sends a chat to calendar 1, all fair execution sequences will lead to a calendar 1 notification.	FAIL
PROPERTY_32_ORIGINAL_NO_DATA	Fairness: After creating an event, deleting an event, changing the date and time of an event or commenting on an event all fair execution sequences will lead to an event notification and the event update being reflected in activity.	FAIL
PROPERTY_32v2_WITH_GENERAL_DATA	Fairness: After creating an event, deleting an event, changing the date and time of an event or commenting on an event all fair execution sequences will lead to an event notification and the event update being reflected in activity.	FAIL



PROPERTY_32v3_USER1	Fairness: After user 1 creates event 1, deletes event 1, changes the date and time of event 1 or comments on event 1 all fair execution sequences will lead to an event notification and the event update being reflected in activity.	FAIL
PROPERTY_32v4_USER2	Fairness: After user 2 creates event 1, deletes event 1, changes the date and time of event 1 or comments on event 1 all fair execution sequences will lead to an event notification and the event update being reflected in activity.	FAIL
PROPERTY_33_ORIGINAL_NO_DATA	Fairness: After creating a memo, deleting a memo, or commenting on a memo all fair execution sequences will lead to a memo notification and the memo update being reflected in activity.	FAIL

PROPERTY_33v2_WITH_GENERAL_DATA	Fairness: After creating a memo, deleting an memo, or commenting on a memo all fair execution sequences will lead to a memo nofication and the memo update being reflected in activity.	FAIL
PROPERTY_33v3_USER1	Fairness: After user 1 creates memo 1, deletes memo 1, or comments on memo 1 all fair execution sequences will lead to a memo nofication and the memo update being reflected in activity.	FAIL
PROPERTY_33v3_USER2	Fairness: After user 2 creates memo 1, deletes memo 1, or comments on memo 1 all fair execution sequences will lead to a memo nofication and the memo update being reflected in activity.	FAIL
PROPERTY_34_ORIGINAL_NO_DATA	Fairness: After a user completes creating a calendar or joining a calendar all fair execution sequences will lead to the user being added to the calendar.	PASS

PROPERTY_34v2_WITH_GENERAL_DATA	Fairness: After a user completes creating calendar 1 or joining a calendar 1 all fair execution sequences will lead to the user being added to the calendar.	FAIL
PROPERTY_34v3_USER1	Fairness: After user 1 completes creating calendar 1 or joining calendar 1 all fair execution sequences will lead to the user being added to the calendar.	FAIL
PROPERTY_34v4_USER2	Fairness: After user 2 completes creating calendar 1 or joining calendar 1 all fair execution sequences will lead to the user being added to the calendar.	FAIL
PROPERTY_37_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot comment on an event before it has been created.	FAIL
PROPERTY_37v2_USER1	Safety: user 1 cannot comment on event 1 before it has been created.	FAIL
PROPERTY_37v3_USER2	Safety: user 2 cannot comment on event 1 before it has been created.	FAIL

PROPERTY_38_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot change the date or time of an event before it has been created.	FAIL
PROPERTY_38v2_USER1	Safety: user 1 cannot change the date or time of event 1 before it has been created.	FAIL
PROPERTY_38v3_USER2	Safety: user 2 cannot change the date or time of event 1 before it has been created.	FAIL
PROPERTY_9_ORIGINAL_WITH_GENERAL_DATA	Safety: A user must sign out in order to have signed in.	FAIL
PROPERTY_9v2_USER1	Safety: user 1 must sign out in order to have signed in.	FAIL
PROPERTY_9v3_USER2	Safety: user 2 must sign out in order to have signed in.	FAIL
PROPERTY_14_ORIGINAL	Liveness: Timetree is deadlock free; there should always be some action occurring.	FAIL
PROPERTY_18_ORIGINAL_WITH_USER1	Liveness: User1 can potentially create a calendar after signing up or in.	PASS
PROPERTY_19_ORIGINAL_WITH_USER2	Liveness: User2 can potentially create a calendar after signing up or in.	PASS

PROPERTY_35_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot comment on a memo before it has been created.	N/A
PROPERTY_35v2_USER1	Safety: user 1 cannot comment on memo 1 before it has been created.	N/A
PROPERTY_35v3_USER2	Safety: user 2 cannot comment on memo 1 before it has been created.	N/A
PROPERTY_36_ORIGINAL_WITH_GENERAL_DATA	Safety: A user cannot update a memo before it has been created.	N/A
PROPERTY_36v2_USER1	Safety: user 1 cannot update memo 1 before it has been created.	N/A
PROPERTY_36v3_USER2	Safety: user 2 cannot update a memo before it has been created.	N/A
PROPERTY_39	Safety: A user must receive an invitation link first to join a calendar.	N/A
PROPERTY_40	Safety: Memo notification cannot be executed without a memo being created, deleted, updated, or commented on.	N/A
PROPERTY_41	Liveness: DisplayActivity should be able to be executed after a user requests to view it.	N/A

PROPERTY_42	Safety: A user must send an invitation for another user to receive the link.	N/A
-------------	--	-----

## General Overview of Modifications to Properties From Progress Report

The comparison between the properties from the original progress report and the newly modified properties reveals several changes focused on specificity and model parameterization. Below is a brief outline of these differences:

- **Specific User Parameters:** The new version of the old properties all use !1, !2, etc., to specify actions for particular users (e.g., SIGNUP !1), whereas the original properties used more general terms without specifying users directly (e.g., {SIGNUP ...}) or not specifying data at all due to the tested gates not being implemented. This change makes the properties more precise, targeting specific user interactions. Below is a table showcasing which properties are using user 1 and or user 2 data in there modified versions:
  - **To Note:** Several of these modified properties user both user 1 and user 2 data however these versions were named according to the user the original property was being tested for.

Modified Properties Using User 1 Data:	Modified Properties Using User 2 Data:
<ul style="list-style-type: none"> <li>• PROPERTY_7v2_USER1</li> <li>• PROPERTY_8v2_USER1</li> <li>• PROPERTY_10v2_USER1</li> <li>• PROPERTY_11v2_USER1</li> <li>• PROPERTY_12v2_USER1</li> <li>• PROPERTY_13v2_USER1</li> <li>• PROPERTY_15v3_USER1</li> <li>• PROPERTY_16v3_USER1</li> <li>• PROPERTY_17v3_USER1</li> <li>• PROPERTY_20v2_USER1</li> <li>• PROPERTY_21v3_USER1</li> <li>• PROPERTY_22v2_USER1</li> <li>• PROPERTY_23v3_USER1</li> <li>• PROPERTY_24v3_USER1</li> <li>• PROPERTY_26a_USER1</li> <li>• PROPERTY_29a_USER1</li> <li>• PROPERTY_29b_USER1</li> <li>• PROPERTY_31v3_USER1</li> <li>• PROPERTY_32v3_USER1</li> <li>• PROPERTY_33v3_USER1</li> <li>• PROPERTY_34v3_USER1</li> </ul>	<ul style="list-style-type: none"> <li>• PROPERTY_7v3_USER2</li> <li>• PROPERTY_8v3_USER2</li> <li>• PROPERTY_10v3_USER2</li> <li>• PROPERTY_11v3_USER2</li> <li>• PROPERTY_12v3_USER2</li> <li>• PROPERTY_13v3_USER2</li> <li>• PROPERTY_15v2_USER2</li> <li>• PROPERTY_16v2_USER2</li> <li>• PROPERTY_17v2_USER2</li> <li>• PROPERTY_20v3_USER2</li> <li>• PROPERTY_21v4_USER2</li> <li>• PROPERTY_22v3_USER2</li> <li>• PROPERTY_23v4_USER2</li> <li>• PROPERTY_24v4_USER2</li> <li>• PROPERTY_27a_USER2</li> <li>• PROPERTY_30a_USER2</li> <li>• PROPERTY_30b_USER2</li> <li>• PROPERTY_31v4_USER2</li> <li>• PROPERTY_32v4_USER2</li> <li>• PROPERTY_33v3_USER2</li> <li>• PROPERTY_34v4_USER2</li> </ul>

<ul style="list-style-type: none"> <li>• PROPERTY_35v2_USER1</li> <li>• PROPERTY_36v2_USER1</li> <li>• PROPERTY_37v2_USER1</li> <li>• PROPERTY_38v2_USER1</li> <li>• PROPERTY_9v2_USER1</li> </ul>	<ul style="list-style-type: none"> <li>• PROPERTY_35v3_USER2</li> <li>• PROPERTY_36v3_USER2</li> <li>• PROPERTY_37v3_USER2</li> <li>• PROPERTY_38v3_USER2</li> <li>• PROPERTY_9v3_USER2</li> </ul>
--	--

- Inclusion of Additional Scenarios: Some of the new properties include additional sequences not present in the original properties, to reflect the concurrent and dual nature of the system, where for several properties in the updated set they involved both user 1 and user 2. Below are some examples.

```

property PROPERTY_37v2_USER1 "Safety: user 1 cannot comment on event 1
before it has been created." is
  "Trialx4final.Int"
  |= with evaluator4
  library standard.mcl end_library
  macro NEVER (R) = [ (R) ] false end_macro
  macro NOT_1_BEFORE_2 (A, B) = NEVER ((not (B))* . (A)) end_macro
  NOT_1_BEFORE_2({COMMENTONEVENT !1 !1 !1 !"1"},
  ({CREATEEVENT !1 !1 !EVENT (1, 0, 0, 0, 0)} or {CREATEEVENT !2 !1 !EVENT
  (1, 0, 0, 0, 0)}));
  expected TRUE;
end property

```

- The highlighted part of this property is there to signify that either user 1 or user 2 must execute the create event gate to create an event in calendar 1 with an event id of 1 and date, year, hour, and minute of value 0 so user 1 is able to comment on that event with a message of "1"; it completely evaluates fairness for commenting by ensuring to check that commenting should be fairly reachable from when user 1 or user 2 creates the event, which cover both paths leading to user 1 commenting.

```

property PROPERTY_38v3_USER2 "Safety: user 2 cannot change the date or time
of event 1 before it has been created." is
  "Trialx4final.Int"
  |= with evaluator4
  library standard.mcl end_library
  macro NEVER (R) = [ (R) ] false end_macro
  macro NOT_1_BEFORE_2 (A, B) = NEVER ((not (B))* . (A)) end_macro
  NOT_1_BEFORE_2({CHANGEEVENTDATEANDTIME !2 !1 !EVENT (1, 13,
  24, 8, 18)}, ({CREATEEVENT !1 !1 !EVENT (1, 0, 0, 0, 0)} or {CREATEEVENT !2
  !1 !EVENT (1, 0, 0, 0, 0)}));
  expected TRUE;
end property

```

- The highlighted part of this property is there to signify that either user 1 or user 2 must execute the create event gate to create an event in calendar 1 with an event id of 1 and date, year, hour, and minute of value 0 so user 2 is able to change the date and

time of event 1 so that it has date:13, year 24, hour:8 minute: 18; it completely evaluates fairness for changing the date and time of an event by ensuring to check that changing date and time is fairly reachable from when user 1 or user 2 creates the event, which cover both paths leading to user 1 this event update.

- Alternating versions for properties and potential property categorization swapping: Some properties were adjusted from using the more complexly defined macros to a more simplified representation of the property, as a means to guarantee that an oddly failing property does in fact hold to pass, which in turn resulted in some properties switching from one type to another. Below are several examples:

```
property PROPERTY_28_ORIGINAL_WITH_GENERAL_DATA "Fairness: After
user completes signing in it is fairly possible to sign out*." is
  "Trialx4final.lnt"
  |= with evaluator4
  library standard.mcl end_library
  macro SOME (R) = < (R) > true end_macro
  macro FAIRLY (A) = [ (not (A))* ] SOME (true* . (A)) end_macro
  macro AFTER_1_FAIRLY_2 (A, B) = [ true* . (A) ] FAIRLY (B) end_macro
  AFTER_1_FAIRLY_2 ({SIGNIN ...}, {SIGNOUT ...});
  expected TRUE;
end property
```

```
-----
property PROPERTY_28_ORIGINAL_WITH_GENERAL_DATA
| Fairness: After user completes signing in it is fairly possible to sign out*.

FAIL
```

```
property PROPERTY_28a_WITH_GENERAL_DATA "Fairness: After user
completes signing in it is fairly possible to sign out*." is
  "Trialx4final.lnt"
  |= with evaluator4
  library standard.mcl end_library
  <(true*.{SIGNIN ...}.true*.{SIGNOUT ...})> true;
  expected TRUE;
end property
```

```
-----
property PROPERTY_28a_WITH_GENERAL_DATA
| Fairness: After user completes signing in it is fairly possible to sign out*.

PASS
```

- PROPERTY\_28a\_WITH\_GENERAL\_DATA is a simplified representation of the Fairness property defined by PROPERTY\_28\_ORIGINAL\_WITH\_GENERAL\_DATA which makes use of macros defined in the standard.mcl library document. The property failing is not attributed to the use of macros but possibly may be due an incorrect use or application of the gates within this implementation or even an incorrect lnt model, where as a result the team decided to check fairness by ensuring that signing out is a reachable action when signing in; to note on the topic of property swapping this new



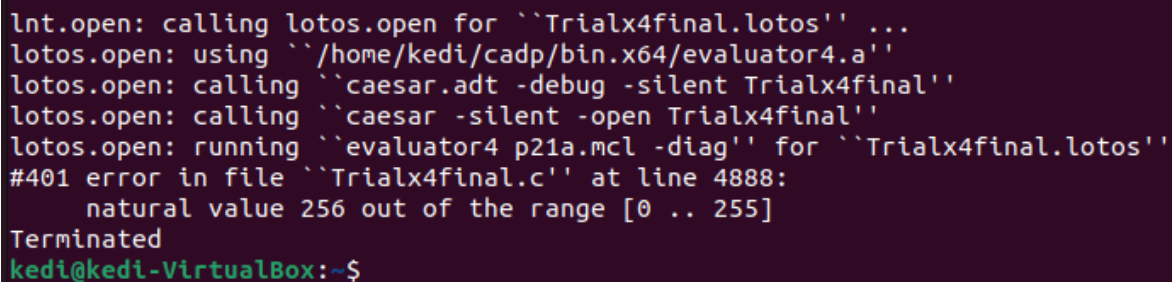
implementation of property 28 admittedly better reflects Liveness than it does Fairness, which is something we took note of in the midst of the properties verification process but did not account for in the property description.

- The list of other properties that underwent the same process include the following:
  - property PROPERTY\_26a\_USER1
  - property PROPERTY\_27a\_USER2
  - PROPERTY\_28a\_WITH\_GENERAL\_DATA
  - PROPERTY\_29a\_USER1
  - PROPERTY\_30a\_USER2
  - PROPERTY\_31v2\_WITH\_GENERAL\_DATA
  - PROPERTY\_31v3\_USER1
  - PROPERTY\_31v4\_USER2
  - PROPERTY\_32v2\_WITH\_GENERAL\_DATA
  - PROPERTY\_32v3\_USER1
  - PROPERTY\_32v4\_USER2
  - PROPERTY\_33v2\_WITH\_GENERAL\_DATA
  - PROPERTY\_33v3\_USER1
  - PROPERTY\_33v3\_USER2
    - Each of these properties correspond to an original fairness property, where in the Demo.svl file the group denoted them as Fairness properties, when in fact their different implementation from the originals they correspond to lead them to more accurately fit under Liveness properties.
- Total Stats for properties: Below is a detailed set of the criteria that represents the set of properties.
  - Number of Safety: 34
  - Number of Liveness: 30 - 14 modified versions swapped out → 16
  - Number of Fairness: 29 +14 modified versions swapped into → 43
  - Number of Unaccounted for Property Results (they could not generate a pass or fail due to running the svl properties using the lnt file): 10
  - Number of Failing: 61
  - Number of Passing: 31
  - Test Batch: 92
    - Failure Rate: 33.6956521739 %
  - Total: 102

# Verification

Describe your debugging process using MCL with evaluator4 and issues found with OCIS, describe the multiple versions of your LNT model, and how you addressed bugs

The debugging process for the MCL properties was conducted at the end of the codes completion, where in hindsight it may have been better to perform property verification earlier to ensure a smaller amount of non-passing properties. Initially, the team planned to use the evaluator 4 diagnosis command to analyse the produced evaluator bcg file, however because of the size of the code at the end of its completion, which is when the process of property verification began, this made it difficult to generate this bcg file; the completed code undergoes infinite compilation due to the millions of states and transitions that occur which produces an error when trying to run the evaluator 4 diagnosis command for a specific mcl property that wants to be debugged. Below is an image of an example of this issue:



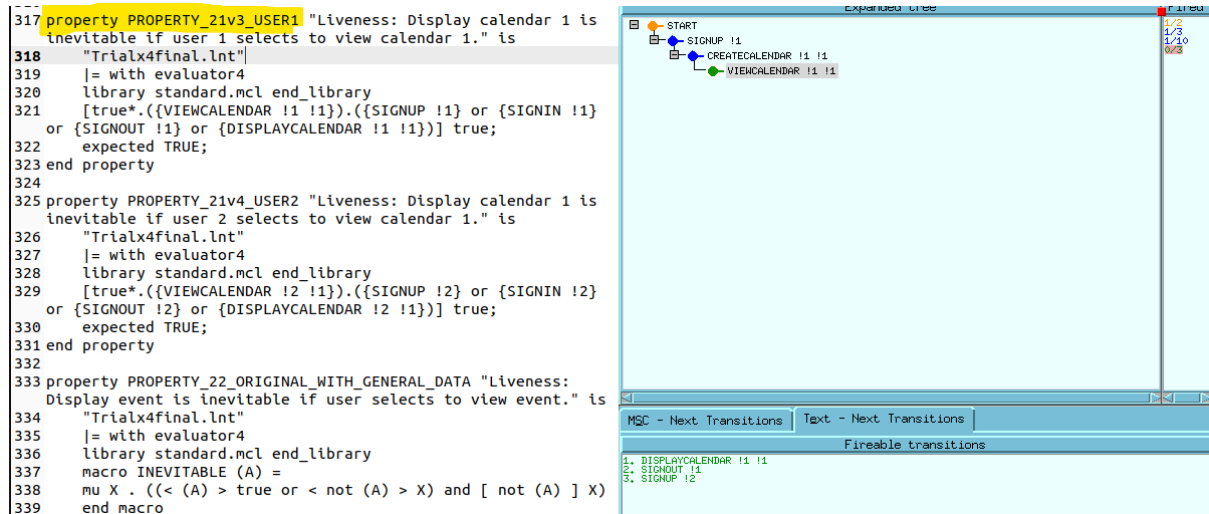
```
lnt.open: calling lotos.open for `Trialx4final.lotos' ...
lotos.open: using `/home/kedi/cadp/bin.x64/evaluator4.a'
lotos.open: calling `caesar.adt -debug -silent Trialx4final'
lotos.open: calling `caesar -silent -open Trialx4final'
lotos.open: running `evaluator4 p21a.mcl -diag' for `Trialx4final.lotos'
#401 error in file `Trialx4final.c' at line 4888:
    natural value 256 out of the range [0 .. 255]
Terminated
kedi@kedi-VirtualBox:~$
```

- You can see here that when trying to run the evaluator4 -diag command for the specific failing property in the p21a.mcl file using the Trialx4final.lnt code, that an error indicating an out of range value occurs and most likely due to the gigantic nature of the model.

Likewise, to address issues of properties failing the team analysed the property in the svl file to understand the sequence of gate being claimed to exist or not exist under the pretext of being a safety, fairness, or liveness property and used ocis to visually determine what the bug contributing to failing was. For instance in the image below, you can see that PROPERTY\_21v3\_USER1 which was a test for inevitability in terms of the Display Calendar action being an inevitably executed action after View Calendar action taken by user 1 for Calendar 1, however what we saw from the ocis is that it was very much possible for other actions such as user 1 signing out via SIGNOUT !1 or user 2 signing up via SIGNUP !1, which in turn informed the team of several things:

- The property is improperly defined for the current implementation of the model.
- The property should be adjusted to expect FALSE instead.
- A Fail in this case signifies that the model is working as it was designed to work and accurately represents a 2 user multi-action calendar sharing application.
  - The team should have modified the property to expect false or explicitly denote in the property description that a fail is expected, which was not

completed in spite of the focus on minimising the enormous scale of the model.



## Difficulties Encountered

As far as the main issues encountered for the final version of the code, these mainly refer to the issue of synchronisation and working to minimise the scale of the lnt model. To address issues with synchronisation, there was a period of time where because the model contained so many gates we experienced the issue of a gate such as INVITELINKTOCALENDAR always being available to the user as an action to execute when it in fact is the gate that is fed data sent by INVITETOCALENDAR, which should always be committed first. The issue here was that we had missed its placement in the par block in the MAIN process which synchronised the USERMANAGER and CALENDARMANAGER processes on several gates with the exception of INVITELINKTOCALENDAR. We had several emails and meetings with the professor regarding this issue where it was suggested to debug via generating a mcl file that tests for the safety property claiming it should never be the case where the INVITETOCALENDAR gate follows the INVITELINKTOCALENDAR along some sequence of actions.

To address the matter of the scale of the model, we were able to do so over the several iterations of the code we generated, where we put limits on the number of invites a user can send or get, and also putting limits on the number of calendar instances being managed so that the two users of the system could only ever perform an action on the singularly available calendar instance. Nearing the end of the codes completion however, when integrating the event and memo related gates for the Timetree model, we noticed the compilation time taking longer and longer until it eventually started undergoing infinite compilation. There were many solutions attempted such as restricting the number of times a user could chat, search, view calendar, display calendar, and comment or makes changes to events or memos, or even making it so that the users could only create an event and memo with id's of 1. This was to ensure that once created no other events or memos could be created as calendar only accepts

unique event and memo ids for event and memo creation, where the plan was to reduce the number of transitions and therefore the number of states, however this proved to be ineffective where the compilation time remained to be infinite.

To add, the trickle down effect of infinite compilation is the problem of not being able to obtain the bcg to be used for property verification, where several properties remained idle in their verification process because they required an extensive search of the model, which when infinitely compiling to be millions and millions of states and transitions is technically impossible. This was the main reason for the groups pivot to an Int based verification style, where of course what was compounded on top as a further difficulty was the issue of not being able to use evaluator 4 diagnosis command as a result of out of bound errors.

## Feedback on CADP

Construction and Analysis of Distributed Processes, better known as CADP, is a powerful piece of software for designing asynchronous concurrent systems with communication protocol and distorted systems, such as the TimeTree application detailed in this report. Some feedback regarding CADP is that while it is very useful with valuable features such as displaying models with a minimised form or OCIS which helps in debugging, it could benefit from more user-friendly documentation and tutorials on YouTube to make it more accessible to a wider range of users. Additionally, improving the user interface and experience to be more intuitive and graphical could enhance the overall user experience. Integrating more advanced debugging tools and error reporting features could also make CADP even more powerful for designing and analysing complex systems. Overall, CADP is extremely useful and will likely gain mainstream adoption for its utility and accessibility. We enjoyed working on the CADP project a lot. Thanks for all the help and for a great semester professor Marssó!

## References

1. <https://en-blog.timetreeapp.com>
2. <https://cadp.inria.fr/man/mc15.html>
3. <https://arxiv.org/pdf/2203.09885.pdf>
4. <https://cadp.inria.fr/man/evaluator4.html>
5. <http://vasy.inria.fr/ftp/presentations/Thivolle-EVALUATOR-08.pdf>