Application of HLA in the Optimization of Rail Transport

T.W. van den Berg, R.E.J. Jansen, D. Middelkoop

TNO Defence, Security and Safety
PO Box 96864,
2509JG The Hague, The Netherlands
tom.vandenberg@tno.nl, roger.jansen@tno.nl,
dick.middelkoop@prorail.nl

Keywords:
Dynamic Traffic Management, Monte Carlo Simulation, HLA, Time Management, Rail Transport

ABSTRACT

The Dutch rail infrastructure manager – Prorail – utilizes simulation to perform research in a number of areas of rail-transport. One area that is of particular interest is that of the analysis of Dynamic Traffic Management (DTM) of trains. The aim of DTM is to manage plan/timetable deviations from daily train operations effectively in order to improve overall performance of the train service. To perform this analysis two existing (legacy) systems, FRISO and TMS, have been connected via the High Level Architecture (HLA).

FRISO is a train simulator that is used to investigate rail transport in the area of several (tens of) kilometers. TMS is a further development of the controller system COMBINE and is an advanced traffic control system that is used to predict and minimize route conflicts between trains in order to improve efficiency, reliability and quality in rail transport. Analysis involves the execution of stochastic (Monte Carlo) simulation, for which conservative Time Management is a critical issue. This paper focuses on the design of the TMS-FRISO federation, the federation agreements made, the lessons learned in making both legacy systems HLA enabled and possible future improvements.

Within this project TNO provided distributed simulation expertise and HLA software tools, like the TNO-RTI and the RCI middleware with code generator.

1. Introduction

The construction of timetables for trains is an off line process and done well in advance. Train order at crossings, junctions and platform tracks is fixed to prevent route conflicts and ensure capacity. When operating/executing a time table in real time unforeseen events may happen, causing a deviation from the planned time table. A Dynamic Traffic Management (DTM) system can alleviate this by allowing trains to arrive on alternative platforms, by allowing variations in train behaviour, by allowing changes in the order of trains and by allowing variation and synchronization of arrival and departure times of trains. A DTM system handles deviations from planning in real time with the aim to optimize overall performance of the rail network. In order to study the effects of DTM in a capacity bottleneck area like Den Bosch in The Netherlands, two existing systems are connected using the High Level Architecture: a train simulator (FRISO) that simulates train movements according to a time table using simple traffic management and an advanced Traffic Management System (TMS) that provides an adapted, optimized plan with advisory speed setpoints to the trains in the model area based on amongst others current position and speed [1].

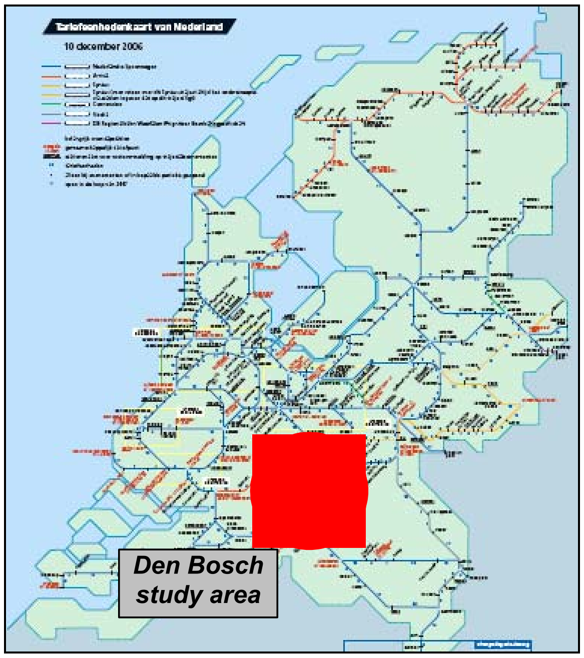

Figure 1. The Den Bosch study area in The Netherlands.

Both systems are fed with actual rail infrastructure and time table data of the Den Bosch area (see Figure 1). This is an area of tens of kilometers, from the city of Tilburg to Ravenstein and from the city of Boxtel to Geldermalsen. The analysis involves running a stochastic (Monte Carlo) simulation, using HLA features to manage simulation time and to coordinate single or multiple successive runs.

The TMS-FRISO federation is currently in the final stage of verification and a number of trial experiments are performed for testing. Verification is expected to complete in the first half of 2008.

This paper gives an overview of the TMS-FRISO HLA federation. Section 2. describes the top level architecture of the federation. The characteristics of the modes and states that the federation supports are described in Section 3. Section 4. briefly discusses the object model created for this federation. More federation agreements follow in section 5. This section also includes the agreements on the handling of time in the federation. Section 6. addresses a number of lessons learned in developing this federation and finally section 7. ends with conclusions.

2. Federation Architecture

The HLA federation consists of two federates, a FRISO federate and a TMS federate, interconnected via the HLA Run Time Infrastructure. FRISO (Flexible Rail Infra Simulation of Operations), from InControl in The Netherlands, is a simulator that is used to investigate and enhance rail transport in the area of tens of kilometers. It contains a model of the rail infrastructure and of trains that follow a time table.

FRISO models the following relevant functions of the rail infrastructure:

- track layout,
- signalling system,
- route setting and
- interlocking.

FRISO also models the following functions with respect to trains:

- train schedule (timetable),
- entry and exit of trains in to and out of the model area,
- disturbances in train entry time,
- stopping time and departure time,
- rolling stock combinations with a certain probability,
- planned stops and passings,
- train behavior (acceleration, deceleration and driving speed of trains) and
- communication delays (from train and infrastructure to TMS).

TMS (Traffic Management System), from CO.S.MO.S and On Air in Italy, is a controller system that is used to investigate and enhance efficiency, quality and reliability in rail transport. It optimizes on punctuality, energy consumption and/or throughput, and controls train speed and route booking on the basis of amongst others current train status information and schedule information. TMS models the following relevant functions:

- rail infrastructure,
- rollling stock characteristics,
- train schedule,
- detailed planning with goals to be achieved and routes to be booked,
- advisory speed setting,
- route booking,
- communication delays (from TMS to train and infrastructure).

TMS itself is composed of a number of components:

- CRS1 (Conflict Resolution System, level 1) is responsible for train scheduling and routing. Its purpose is to optimize traffic, developing and updating a current plan for all trains, which aims at reducing the final global delay.
- CRS2 (Conflict Resolution System, level 2) takes care of reconstruction of the system initial status, plan update management and route management.
- SR (Speed Regulator) is responsible for TMS plan execution. SR receives from the CRS1 the current plan, defined in terms of goals (arrival times, departure times, speed) for each train. The purpose of SR is to compute a speed profile for each controlled train that will achieve all the goals in a safe and energy consumption efficient manner.
- CS (Communication Server) is responsible of the routing of (TCP/IP) messages between the TMS components and interfaces with the mapper for outside communication.

Both federates interface to the RTI through the use of so called “mappers”. These mappers translate RTI service invocations to TMS socket messages or FRISO ActiveX Controls and visa versa. By using this solution the advantage is that all HLA RTI specifics are concentrated in relatively small software components, leaving much of the original systems intact.

In addition, each federate uses its own (proprietary) databases for logging information and retrieving information about the rail infrastructure, timetable, signals, routes and rolling stock. The infrastructure related data is imported from Prorail’s infrastructure database “Infra Atlas”. This data is converted pre-runtime to other formats.

Finally, for the Run Time Infrastructure the TNO-RTI is used. The TNO-RTI is a high performance and partial RTI implementation that supports most of the RTI management services.

An architectural overview of the federation is shown in Figure 2.

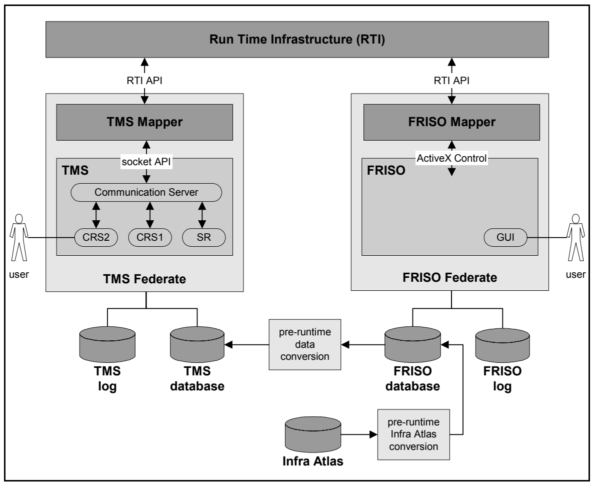

Figure 2. Federation Architecture.

3. Federation Simulation Modes and States

The federation design enables both a real time (possibly human operator in the loop) simulation and a faster than real time stochastic (Monte Carlo) simulation. Only the latter mode is supported in the current release of the federation.

The characteristics of both modes are:

1. **Manual**

- Both real time and non real time simulation;
- One replication (simulation run);
- During a replication the TMS federate may take/return control of train movement and route booking from/to the FRISO federate;
- The TMS federate may join and leave the federation during a replication.

2. **Batch**

- Faster than real time simulation;
- Multiple replications (simulation runs) with variable seeds;
- During one replication the TMS federate takes control of train movement and route booking after a warm up period of the FRISO federate;
- During one replication the TMS federate does not return control of train movement and route booking to the FRISO federate;

- The TMS and FRISO federates are present throughout the federation execution.

Manual mode is intended to mimic the situation in which TMS is connected to a “life system”. Trains continue to run, with or without the presence of TMS. When TMS is present it may take control of train movement. Batch mode is used for analyses where TMS is always present. This mode is for Monte Carlo simulation.

Figure 3 shows the federation states and state transitions for both modes. The states are:

1. Start up

State in which federates of the federation are launched, start up and discover each other.

2. Initialization

State in which the federation initializes. Model properties are exchanged and individual federates initialize.

3. Save (batch mode)

State in which the federation state is saved via the HLA Save for a future restore.

4. Replication

State in which the federation performs a replication (simulation run). During a replication the federation is in one of the following sub states:

- **Uncontrolled**: state in which the FRISO federate controls train movement and route booking.
- **Prepare to control**: state in which the FRISO federate still controls train movement and route booking, but where the TMS federate has indicated that it is about to take control of train movement and route booking.
- **Controlled**: state in which the TMS federate controls train movement and route booking.

5. Restore (batch mode)

State in which the federation state is restored via the HLA Restore for another replication.

6. Shutdown

State in which the federation shuts down.

The state transitions are controlled by the FRISO federate, so simulation control is implicitly with this federate.

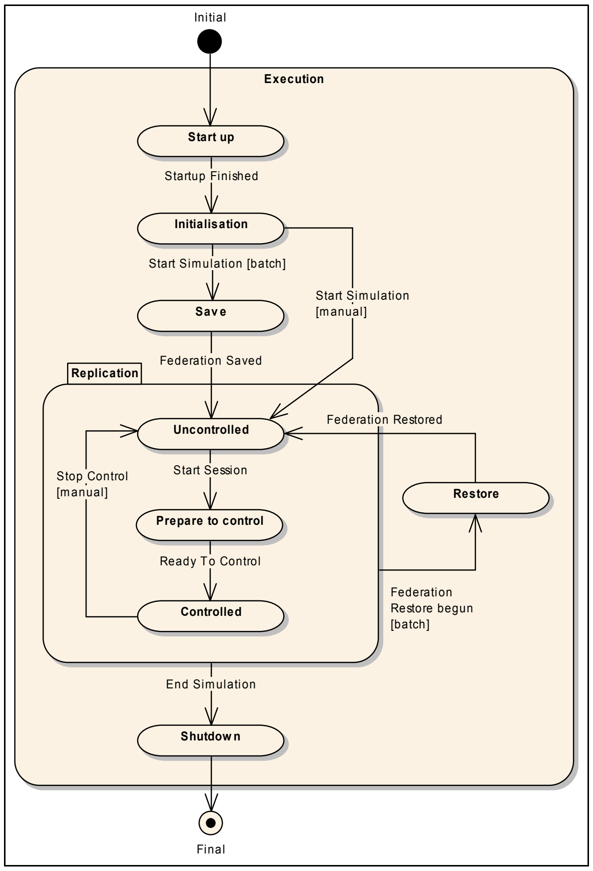

Figure 3. Federation simulation modes and states.

4. Federation Object Model

The Federation Object Model (FOM) defines the HLA objects and interactions that can be exchanged between the HLA federates. The FOM for the TMS-FRISO federation is in the current federation release based on the existing (legacy) TMS message interface for two reasons:

- There exists no reference FOM like the RPR-FOM that can be reused for the purpose of this federation.
- To simplify the integration between TMS and FRISO by reusing the TMS message definitions.

The FOM has two HLA object classes, namely ModelProperties and Train. The ModelProperties object class defines the experiment and scenario. The Train object class includes attributes for amongst others the train name, length, current position, current speed and current advisory speed. Modelling these as objects allows (in manual mode) a potential late joining federate like a viewer to retrieve the current situation from the federation.

All other TMS interface messages are modelled as HLA interaction classes. Interaction classes include plan (time table) related messages, route booking related messages, infrastructure related messages, and train target.

The plan related messages communicate time table information at run time to TMS. For example, an Initial Plan interaction is used to communicate a part of the timetable at the start of the simulation. Then, periodically after a defined interval (e.g. 15 minutes), a Supplemental Plan interaction contains the next part of the timetable. A modification to a previously sent plan is communicated by a PlanChanges interaction.

The route booking related messages deal with the booking of routes or partial routes, as explained in the next section.

The infrastructure related messages provide information about availability and status of signals, sections, blocks, speed restrictions, etc.

The train target provides the advisory speed for a controlled train. After receiving a target the train should accelerate or decelerate to the target speed specified, if allowed by the safety system (signals, speed restriction).

5. More Agreements

5.1. Train Entry and Exit

The FRISO federate simulates train movement over the infrastructure according to a time table. When a train "enters" the controlled area FRISO creates the associated HLA train object in the federation execution and TMS will receive a Discover Object message from the RTI. When a train leaves the area FRISO deletes the associated train object from the federation execution and TMS will receive a Delete Object message from the RTI. Train objects are created (and their attributes updated) a few simulation minutes before they actually enter the area to ensure that TMS is "aware" of their presence.

5.2. Route Booking

In the time table trains are allocated to routes. A route is further composed of blocks and sections to increase the granularity of locking. A route must be booked (reserved) by a train and set by the route booking system before the train may enter the route. If a route is not set, the train must wait. The authorization to proceed is provided by a signal (not showing red if authorized). When a train leaves a section the (partial) route is released again and available for new route booking and setting.

The FRISO federate manages the routes in the infrastructure. Route booking requests are (in the state **Replication.Controlled**) initiated by the TMS federate as interactions during the simulation cycle. Depending on, amongst others, the current simulation time and the time parameters in the route booking request FRISO may set the route in the future, set the route immediately or cancel the request. The TMS federate may also cancel an earlier issued route booking request.

5.3. Time Management

Since the TMS-FRISO federation is mainly intended as an analysis tool performing many simulation runs, repeatability is a highly desirable characteristic. This allows unexpected or unusual analysis results to be investigated in greater detail to explain the underlying phenomenon. Repeatability requires a conservative Time Management strategy using Time Stamp Ordered (TSO) messages.

The TMS-FRISO federation uses the HLA Time Management services [2] to control the advancement of time and ordering of messages and to simulate the communication channel delay between TMS, train and infrastructure, as explained in the following subsections.

5.3.1. Advancing Time

FRISO and TMS are time regulating and time constrained federates as they both send and receive TSO messages.

Every time step (also called "simulation cycle") the FRISO federate processes the received train targets and route booking requests, calculates the new train state for each train, updates the train states in the federation and advances simulation time by issuing a Time Advance Request (TAR) with a time step "dt". The receipt of a Time Advance Grant (TAG) marks the start of a new simulation cycle.

The TMS federate runs "event driven". It "follows" the federation time by issuing a Next Message Request (NMR) supplying the value "infinite" as logical time value. This means that the TMS federate waits for the next message from the federation.

When the TMS federate "wakes up" it consumes the received messages (e.g. train state, route settings). The receipt of the TAG marks the end of the batch of incoming messages and the beginning of the "planning cycle". In the planning cycle TMS calculates, amongst others, new train targets and sends these into the federation. Once the planning cycle is completed TMS waits again for the next message from the federation.

Figure 4 illustrates the working of a simulation cycle. The figure shows two simulation cycles #1 and #2 (indicated by the square box). To simplify the sequence diagram only train state messages (from FRISO) and train target messages (from TMS) are shown. But in reality many other messages are communicated, including plan and route booking messages, and discover and delete object messages for train entry and exit.

In Figure 4 a one second simulation time step dt is used. The epsilon values are used to ensure that train state and train target can be exchanged in the same simulation cycle. That is, when epsilon2 ≤ dt – epsilon1 a train target from TMS shall be received by FRISO before the start of the next simulation cycle (where dt, epsilon1 and epsilon2 > 0).

The RTI simulation time starts at zero for each new replication. Since the train time table is constructed in another time frame than the RTI simulation time each mapper corrects for this by adding the scenario start time to the RTI simulation time value.

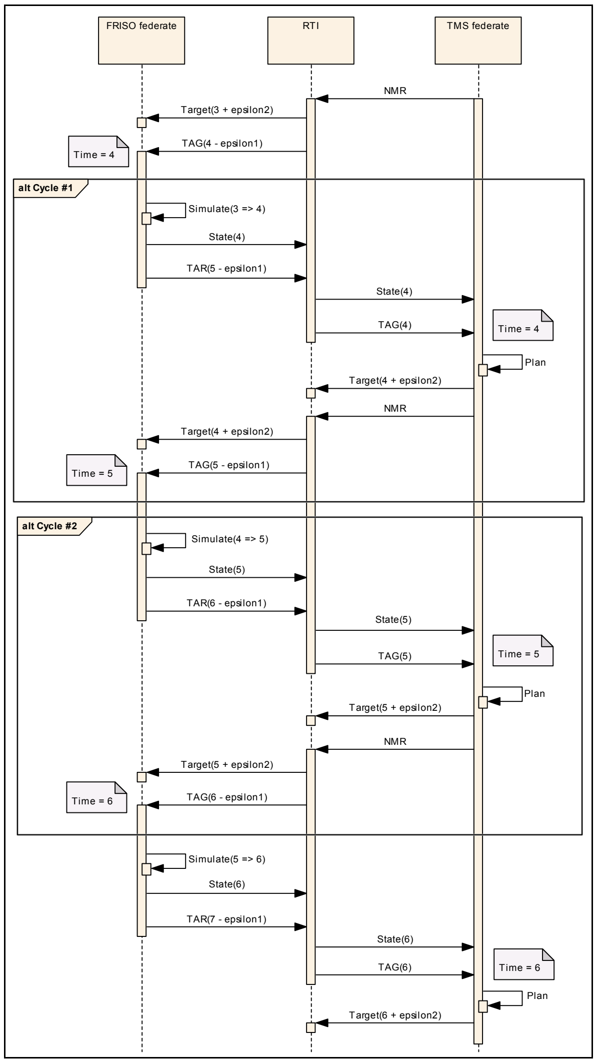

Figure 4. Simulation cycles.

5.3.2. Communication Channel Delays

The HLA Time Management functions are used to simulate the communication delay that exists between TMS, train and infrastructure. Two communication channels are defined:

- Between TMS and infrastructure (for route booking, route setting and infra status) and
- Between TMS and train (for train state and train target).

The communication time between TMS and infrastructure is negligible.

The TMS federate models the communication time from TMS to train driver and back to TMS as a “control loop delay” (see Figure 5). When TMS (SR component) receives speed and position data for the circulating trains, it calculates an advisory speed for each train. The computation (or planning) time is a parameter t_1 depending on the TMS internal algorithms efficiency. Another delay t_2 is introduced to communicate this advisory speed to each “train driver”. An important delay t_3 is associated to the driver reaction time. Other delays are generated by the train on-board systems responsible for estimating the train position and speed (parameter t_4) and to communicate this information (parameter t_5) to TMS.

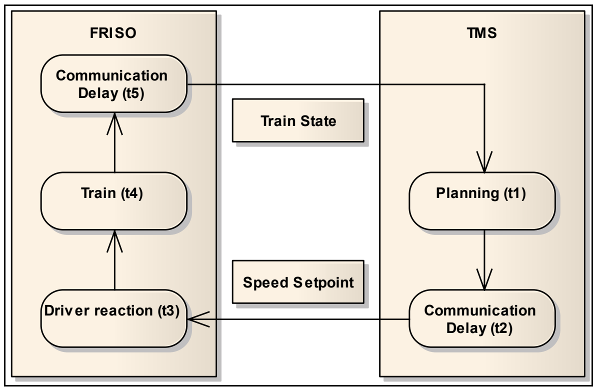

Figure 5. Control loop delay.

So, let *t* be the total Control Loop Delay. The parameter *t* is composed by a set of parameters that are all independent of the TMS, except for parameter t_1 (besides, compared with the other parameters, t_1 is negligible):

- TrainToTMSDelay;
- TMSToTrainDelay;
- driver reaction time.

From the viewpoint of the TMS performance, the only relevant parameter is the total delay *t*. The effect of the delay control loop could cause a reduced capacity of the TMS to control the trains. As a consequence, to reach a proper control of the trains, the TMS must estimate position and speed of the trains at time *t*+*t*, on the basis of the knowledge of the position and speed of the trains at time *t*, and on an estimation of the parameter *t*.

Simulation of the communication channels is as follows.

1. TMS – infrastructure communication channel:

Since the delay time is negligible this channel is not simulated (in effect the value zero is assumed as delay time).

2. TMS – train communication channel:

This channel is simulated by both federates.

- The **TMSToTrainDelay time** is simulated by the TMS federate. The TMS mapper adds "TMSToTrainDelay" to the RTI timestamp of each TrainTarget message.
- The **TrainToTMSDelay time** is simulated by the FRISO federate. The FRISO mapper adds "TrainToTMSDelay" to the RTI timestamp of each Train State message.
- The **driver reaction time** is simulated by the FRISO federate.

5.3.3. TMS Planning Time

As mentioned in the previous paragraph the TMS planning time is generally negligible with respect to the TMS – train communication channel delay. Planning time is not negligible in the following cases:

- Upon the "initial planning" when TMS receives the initial plan, infra status and train states (just before the Ready To Control).
- Upon a "re-planning" by TMS.

The first case is relevant in manual mode, where the TMS federate may join and leave the federation any time. Upon each join TMS will perform an initial planning. In batch mode initial planning happens only once at the start of each replication. The second case happens once in a while, because TMS looks well ahead in planning.

For the current federation release, TMS planning time is neglected. This means that the TMS federate issues an **NMR** after processing each batch of messages.

In a future release planning time is not neglected. The TMS federate manages the planning time ("pt") as follows. Since the TMS federate has become "time aware" the federate should not merely issue an **NMR**, but instead issue:

- a **TAR**(T+pt) when the TMS federate received and processed messages (thus spent pt > 0 time on planning);
- an **NMR**(inf) when the TMS federate just received a TAG without any messages (thus spent pt = 0 time on planning).

As can be seen, the management of time by the initial and later TMS federate release is compatible.

6. Lessons learned

This section summarises some of the lessons learned in the development of the TMS-FRISO federation. The lessons learned have been based on feedback from the parties involved in the project (On Air, CO.S.MO.S, Prorail, InControl and TNO).

6.1. HLA Concepts

An important statement to start with is that the suppliers of TMS and FRISO are all new to HLA. For them HLA provides many new concepts for simulation interoperability, amongst which the new concepts for Time Management and object/declaration management are the most important ones for the TMS-FRISO federation. Previously, interoperability with TMS was mainly built on the concept of a real time point-to-point TCP socket connection, without much time synchronisation. The new HLA concepts were initially hard to grasp, partly due to the lack of simple examples to start with.

6.2. HLA Architecture

HLA structures simulation models in federates, interconnected by an RTI for run-time information exchange. It is a general good practice to have some form of 'mapping function' or 'conversion module' between the simulation model and the RTI to prevent mixing of RTI specifics with the simulation model. This approach was followed for TMS-FRISO federation as well, resulting in so called 'Mapper' components between simulation model (TMS and FRISO) and RTI. This 'design decision' was also made to preserve the existing TCP socket connection among TMS components and to prevent just too many changes in one release. Identified further improvements include:

- Extend the use of HLA to the individual TMS components, replacing the TCP socket interface by the RTI. The consequence is that the RTI becomes part of TMS and thus of a potential operational TMS, unless variants of TMS are created. Note that each individual TMS component will still have a 'conversion module' or 'Mapper'.
- A consideration is to split also FRISO in smaller building blocks (federates) to promote reuse across simulation applications, e.g. route booking simulation, safety system simulation, train (and train driver) simulation.

6.3. Time Management

Time Management was problematic with earlier integrations of train simulators with TMS. Time synchronisation was performed by passing a timestamp in each message. Problems included:

- Time is only synchronized when messages are transmitted;
- No guarantee that a scenario is reproducible, as the timing of message receipts is dependent on the operating system functions and the network topology and bandwidth;
- No guarantee that an "as fast as possible" running scenario provides a correct outcome as each simulator runs at its own pace;

- No concept of simulation time; everything is in wall-clock time.

HLA Time Management was the hardest HLA topic to tackle in developing the TMS-FRISO federation, because:

- The concept of time is a difficult topic anyway, regardless of the simulation application;
- FRISO and TMS were initially not prepared for Time Management by an external software component (the RTI), in particular for time stepped simulation.

Once the required behaviour was specified, both FRISO and TMS were adapted to time stepped HLA federates. With the federation both real time and faster than real time simulation is possible, without effects on the outcome of the simulation. Also scenarios are reproducible.

It was also noted that the added value of HLA Time Management becomes really visible with three or more federates participating in a simulation. With two federates Time Management is still relatively easy to oversee.

6.4. HLA Middleware

TNO's HLA middleware, the Run-time Communication Infrastructure (RCI), provides the federate developer with the necessary functionality to incorporate the federate into a federation and supports reuse of federate components [3]. The RCI library provides all kind of services that are independent from the FOM, while the RCI code generator takes care of all the federate's FOM dependent interface functionality.

The RCI library provides the federation developer a simplified programming interface to the RTI. The RCI programming interface deviates slightly from the RTI programming interface due to the simplification of programming interface. One of the benefits of the RCI is the very easy transformation from HLA 1.3 to IEEE1516, since the specifics of both interfaces is "hidden" by the RCI.

The RCI library and RCI code generator were used in the construction of the "mappers". The mappers can be seen as the glue code between the generated HLA interface code and the higher-level functionality of the federate [4]. When the FOM was changed or extended during the project, the HLA interface code could simply be regenerated and the developer could concentrate on the actual functionality of the federate. The general conclusion was that the RCI has saved time in the construction of the mappers and federates.

6.5. Federation Object Model

It was relatively easy to create a FOM from the TMS message definitions. As a side effect however, some time-related fields are also present in the FOM. As HLA Time Management is used, these (redundant) fields should be removed from the FOM.

The question remains if the FOM (based on TMS) is generic enough and future proof. Probably not. As additional federates may be added to the federation (or existing ones are replaced) the FOM will change, improve and mature. One simulator that already has been suggested for addition to the federation is the train cabin simulator "MATRICS". Another candidate is a simulator of the safety system, called "BITS".

One proposed improvement for the FOM is to split the object model into smaller parts, so called Base Object Models. For example, models for routes, infrastructure, safety, planning. These Base Object Models are the building blocks for extending the federation with additional federates.

6.6. Databases

InfraAtlas is the source database for the infra data. InfraAtlas data is converted to FRISO format data. And FRISO format data is converted to TMS format data. Because of the various transformations to different database formats inconsistencies are possible.

Ideally, InfraAtlas should provide an exchange data format that is readable by both TMS and FRISO. This makes TMS independent from FRISO when TMS is reused in another federation configuration. In addition, other federates (such as a viewer) can use the same exchange format. A possible (pre-runtime) exchange format is RailML [5]. The purpose of the RailML is to define XML-based standards for simplified data exchange between railway applications.

7. Conclusion

The main conclusions are:

- HLA concepts may be difficult to understand for first time users. Training and good training or reference materials are therefore important.
- Time Management is hard topic to tackle. Sufficient time and effort should be allocated for Time Management agreements and for the identification of required changes in existing systems.
- FOM based code generation speeds up development. An example of supporting tools is the TNO RCI and codegenerator.
- A good design pattern is to use a "mapper" component to prevent the mixing of RTI specifics with the simulation model.
- Both TMS and FRISO can be split in smaller HLA based building blocks, creating a component based federate.
- The TMS-FRISO FOM should be improved by removing TMS specifics and by dividing and standardizing the FOM in several Base Object Models.

- RailML may be a way to standardize the exchange of pre-runtime infrastructure and time table data.

8. References

[1] Simulation of traffic management with FRISO, A.D. Middelkoop, L. Loeve. Computers in Railways X: Computer System Design and Operation in the Railway and Other Transit Systems. 2006.

[2] IEEE Std 1516.1-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification, March 2001.

[3] Marco Brasse, Wim Huiskamp, Olaf Stroosma, "A Component Architecture for Federate Development", Simulation Interoperability Workshop, Fall 1999 (99F-SIW-025).

[4] Roger Jansen, Louwrens Prins, Wim Huiskamp, "Template Driven Code Generator for HLA Middleware", Simulation Interoperability Workshop, Fall 2007 (07F-SIW-038).

[5] RailML, http://www.railml.org.

9. Author Biography

Tom van den Berg is scientist in the M&S department at TNO Defence, Security and Safety, The Netherlands. He holds an M.Sc. degree in Mathematics and Computing Science from Delft Technical University. His research area includes distributed processing and simulation systems, software architectures and software process improvement.

Roger Jansen is a member of the scientific staff in the M&S department at TNO Defense, Security and Safety in the Netherlands. He holds an M.Sc. degree in Computing Science and a Master of Technological Design (MTD) degree in Software Technology, both from Eindhoven University of Technology, The Netherlands. He works in the field of distributed simulation and his research interests include distributed computing and simulation interoperability.

Dick Middelkoop is working at ProRail, department of Rail Development and responsible for decision support tools in the field of capacity analysis and optimisation of the rail network.