



NLR-TP-2005-415

Assessment of Galileo Key Dependability and Safety Parameters

A simulation-based approach

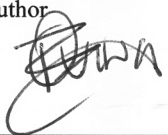
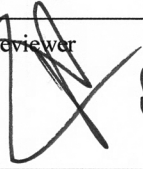

B.A. Oving

This report has been based on a paper presented at ENC-GNSS2005, at München (Germany) on 21-07-2005.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Customer: National Aerospace Laboratory NLR
Working Plan number: AS.1.I, AT.1.A, AT.2.D
Owner: National Aerospace Laboratory NLR
Division: Aerospace Systems & Applications
Distribution: Unlimited
Classification title: Unclassified
July 2005

Approved by:

Author  1-8-05	Reviewer  8/1/05	Managing department  8-9-05
---	---	--



Summary

For Galileo, a need for means for analysing performance of the navigation system with focus on dependability (reliability, availability and maintainability) and – to some extent – safety aspects has been identified by the European Space Agency (ESA).

NLR has supported Galileo Industries (GaIn) with dependability analyses, such as Functional Hazard Analysis (FHA), Failure Mode Effect and Criticality Analysis (FMECA) and Fault Tree Analysis (FTA), on Galileo system level.

The above-mentioned analysis methods each are useful for a particular problem area. They, however, also have their limitations. Most important drawback is that they are not capable of calculating certain dependability characteristics for Galileo, such as continuity and time-to-alert. These characteristics can only be derived using dynamic simulations where the time is an explicit parameter.

To target these dependability objectives, a modelling and simulation concept based on Dynamically Coloured Petri Nets (DCPN) is introduced in this paper to support dependability analysis on stochastic failures and statistic analysis in relation to top-level hazards. This tool is considered to provide an important, if not essential, contribution to the assessment of Galileo as a system with – among others – safety critical user communities.



Contents

1	Introduction	4
2	Requirements for Assessing Key Dependability Parameters	5
3	Dynamically Coloured Petri Nets	8
3.1	Introduction into Petri Nets	8
3.2	Petri Net Extensions	9
3.3	Petri Net Performance	9
4	DCPN Model of Galileo System	11
4.1	Modelling Basics	11
4.2	Overview of Galileo	13
4.3	Modelling of Galileo Measuring Chain	14
5	Conclusion	20
6	References	20



1 Introduction

For Galileo specific aspects related to the performance of Galileo system services need special attention, especially where these are linked to safety. Key aspects for the safety-of-life user are the confidence in a sufficiently low level of the Integrity Risk and Continuity Risk associated with the services provided.

During the Galileo development process extensive analysis is required to assess the navigation system with focus on these dependability and safety (i.e. RAMS) aspects. Traditional RAMS analysis, such as Functional Hazard Analysis (FHA), Failure Mode Effect and Criticality Analysis (FMECA) and Fault Tree Analysis (FTA), are only suited to partially cover the Galileo RAMS engineering needs. For this complex and dynamic system, techniques with appropriate tooling support are required that enable stochastic modelling and simulation of the system, for verification of dependability figures of merit and associated top-level hazards including Integrity Risk and Continuity Risk.

An FTA can be used for the verification of Galileo performance parameters: once actual availability numbers of the Galileo segments (ground segments and space segment) are known, the probabilities of the events listed in the trees can be modified accordingly. The trees will then compute the probabilities of top-level hazards, which can be evaluated against the top-level requirements. Results generated in this fashion can be used for re-allocation of error budgets to the segments or to modify the design (i.e. introduction of barriers against certain events or redundancies of critical elements). Despite the ease-of-modelling and fast calculation capability of Fault Trees, this technique has several limitations when analysing dynamic systems.

FMECA techniques analyse the criticality of effects that are a direct or indirect consequence of failure modes associated with functions or elements the system is composed of. Furthermore, FMECA has its limitations when assessing dynamic systems for dependability and safety.

2 Requirements for Assessing Key Dependability Parameters

The objective of dependability is to assess the impact of failures and feared events on the performance of the Galileo system. Key system analysis issues are:

- Integrity Risk,
- Continuity Risk,
- Availability of Services,
- Robustness (incl. redundancy at element level),
- Feared (or Hazardous) Events;
- Failure Tolerance, concerning operator errors and equipment failure (single and multiple), single point failures, common-cause/common-mode failure impact, and failure propagation.

The Galileo Mission Requirements Document (MRD) shows the Required Navigation Performance (RNP) concept from aviation in a diagram, clarifying the relationship between the performance parameters Accuracy, Integrity, Continuity and Availability, which are also crucial to Galileo RAMS requirements.

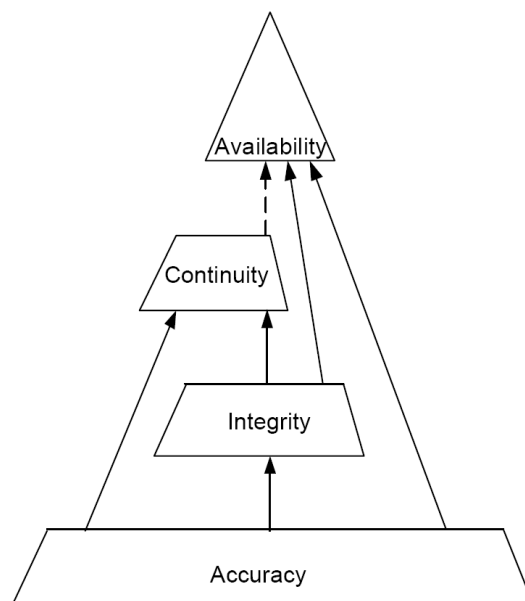


Fig. 1 RNP diagram for Galileo

The pyramid shows the following combinations:

- Availability of Accuracy + Integrity + Continuity,
- Continuity of Accuracy + Integrity,
- Integrity of Accuracy,

where

- Availability: percentage (e.g. 95%) of time the service is up,



- Integrity: probability of Hazardous Misleading Information (HMI) is below the user limit,
- Continuity: probability that the service will be up over the next interval.

The actual combinations depend on the Galileo service used:

- Satellite-Only Navigation Services:
 - Open Services, providing navigation & timing,
 - Safety-of-Life Services, providing integrity & continuity,
 - Commercial Services, providing commercial ranging and data signals,
 - Public Regulated Services, providing restricted-access navigation signals.
- Other services:
 - satellite & local components, search & rescue, external integrity.

The most demanding Galileo service is the Safety-Of-Life Service for which the following requirements are set:

- Integrity Risk
the probability that a Position Error Alert is not given within the Time to Alert period, shall be less than 3.5×10^{-7} over any continuous period of 150 seconds; this includes a contribution of the user receiver of 1.5×10^{-7} .
- Continuity Risk
the probability that navigation & integrity is not provided over the following 15 sec interval shall be less than 1.0×10^{-5} ; this includes a contribution of the user receiver of 2.0×10^{-6} .

These requirements are very difficult to verify due to the low probabilities. For static analysis, FTA can be used to verify the probabilities, given dependability parameters such as Mean Time Between Failures (MTBF) and Mean Time To Recover (MTTR), and probabilities of environmental feared events.

However, for dynamic analysis, simulations must be used that are able to analyse internal failures and external feared events, in both a deterministic way to study the effects of such events at system level, complementing FMECA type analysis, and a stochastic way to study probabilistic issues in relation to system performance requirements, complementing FTA.

The Galileo System Simulation Facility (GSSF) is an example of a simulator that allows verification of the performance of the system, not only in terms of accuracy (algorithm performance), but also dependability characteristics such as availability.



Thus, the GSSF supports some system-level dependability analyses, but its design is based on the deterministic aspects only, i.e. no random failures nor statistical analysis. Furthermore, it is cumbersome to modify the design when investigating failure mitigation (redundancies, barriers). The resulting simulation software is relatively extensive and complex, which results in low faster-than-real-time performance, especially on desktop computers (factor 10^3).

A stochastic simulation tool should be able to calculate the number of Integrity and Continuity events over system life time, given the dependability parameters. Note that a single simulation is not representative, so that for statistical verification of Integrity Risk, Continuity Risk, and Availability of Service, Monte Carlo-type simulations are required.

The simulation needs to model the complete Galileo system (end-to-end), including the navigation and integrity chains. This means that a time acceleration factor in the order of 10^8 is required, where time-driven, fixed time step simulators achieve roughly a factor of 10^3 for a typical service volume scenario.

3 Dynamically Coloured Petri Nets

To achieve a high faster-than-real-time factor, Petri Net-based simulations can be used. Such simulations have been used in studies to find minimum distances between flying aircraft whilst maintaining safety requirements (collision risk).

3.1 Introduction into Petri Nets

A Petri Net is a graphical formalism for specifying system behaviour. It is used in Software Engineering to represent the dynamic states of the system. Unlike Finite State Machines, Petri Nets are suited to model systems with asynchronous and concurrent events.

A Petri Net is a graph of circles (named *places*), bars (named *transitions*) and arrows (named *arcs*). The arcs exist between places and transitions, and vice versa. The places represent possible discrete modes or conditions, the transitions represent possible actions, to be executed during the transition between the conditions.

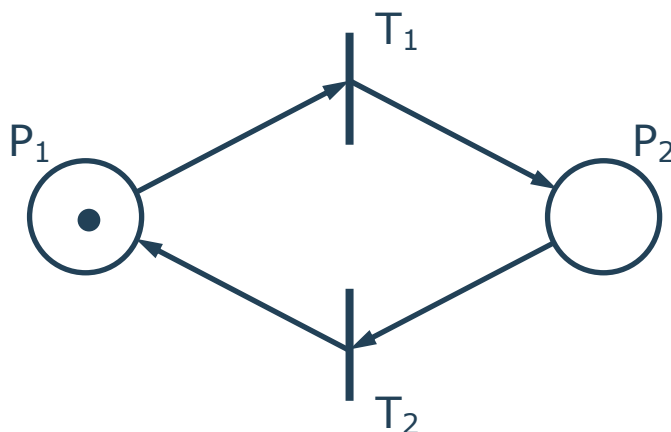


Fig. 2 Example Petri Net showing a two-state system, e.g. ON/OFF

A condition is current if a token (represented by a *dot*) is residing in a place, see place P_1 in figure 2. Several tokens can exist simultaneously in the various places of the Petri Net, representing compound conditions.

If a transition contains at least one token in each of the places to which it is connected by an incoming arc (these places are called input places), all of its preconditions hold. In that case, the transition may fire: it removes one token per arc from all of its input places and lays one token per arc in all of its output places.



3.2 Petri Net Extensions

The many advantages of Petri Nets and their extensions include their graphical representation, which makes it possible to easily model and observe a system with all of its components, and their applicability to dynamic process models.

For analysing complex, dynamic systems featuring piecewise deterministic Markov processes, NLR has developed the Traffic Organisation and Perturbation AnalyZer (TOPAZ) methodology. This methodology provides designers of advanced Air Traffic Management (ATM) with safety feedback following on a (re)design cycle (Blom et al., 1998). With TOPAZ, complex and highly distributed systems can be assessed for their accident risk.

TOPAZ uses the concept of Dynamically Coloured Petri Net (DCPN), which acts as the basis for a dependability assessment of the system. This extension to the normal Petri Net concept features coloured tokens, which means that the tokens carry extra information that can be used during the firing of transitions. The colouring of the tokens may change in time using stochastic differential equations, such that this introduces the dynamical concept into the modelling.

Time dependency is introduced by adding delay transitions, which only may fire when they are enabled and the current time has progressed beyond the delay since enabling. This can be used for the generation of feared events or item failures, especially when the delay is calculated using stochastic probability functions.

A further extension is the introduction of firing functions that specify the firing behaviour of a transition using a user-definable algorithm, e.g. randomly add tokens in output places, or create a coloured token using the tokens from input places. For example, in the Galileo simulation model, they are used, among others, for determining the visibility of satellites from ground stations.

3.3 Petri Net Performance

A DCPN kernel has been implemented in Java to investigate the potential time acceleration. The simple two-state Petri Net from figure 2 was used to benchmark the kernel's ability for faster-than-real-time performance. The duration of the simulation was set to roughly one year (20 million seconds), the delay of transition T_1 was set to 1000 hours (MTBF), and the delay of transition T_2 was set to 10 hours (MTTR).

The first run used a fixed time step of one second; the second run used the dynamic time step approach, jumping from event to event.

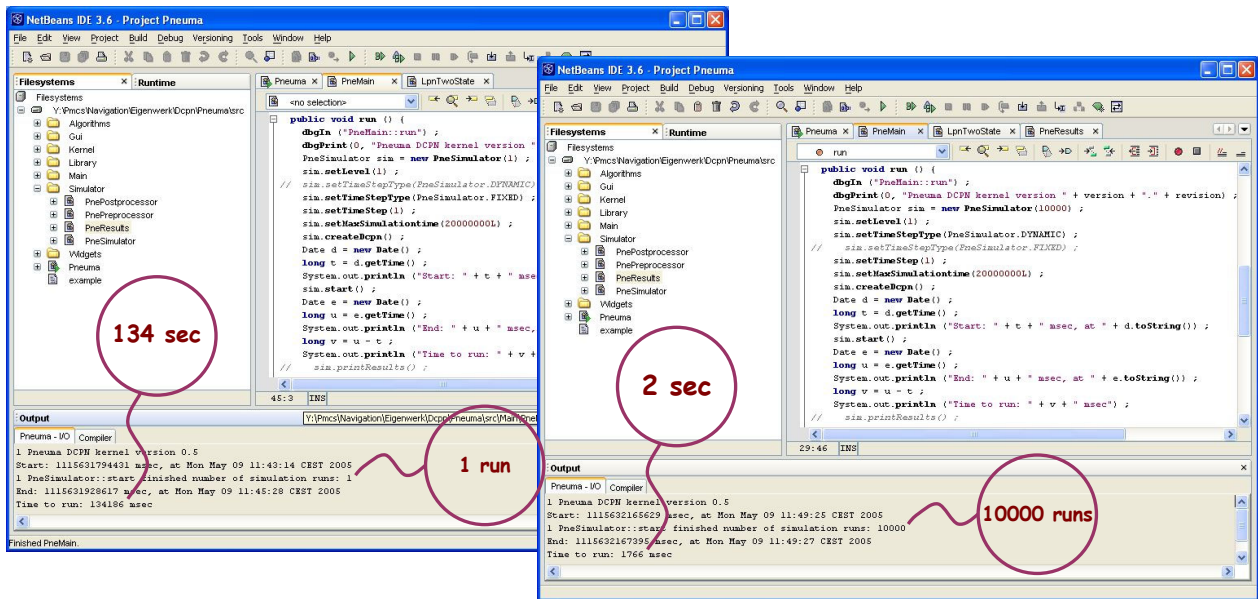


Fig. 3 Fixed time step versus dynamic time step simulation

The first run with is very simple model achieves an acceleration factor of simulation duration ($20 \cdot 10^6$ s) divided by 134 s equals 150,000. However, only 12 events have been recorded during this run. The second run shows an acceleration of (10,000 runs) times ($20 \cdot 10^6$ s) divided by 2 s equals 1011 for an average of 12 events per run. The dynamic time step run thus yields a boost of 106 over the fixed time step run. Note however that the first depends on the number of events, whereas the latter does not. For a more complex model generating lots of events, the difference will be less.

4 DCPN Model of Galileo System

4.1 Modelling Basics

Boolean logic can be easily be modelled in Petri Nets, see figure 4:

- OR: either of two events triggers a third event (cause - effect).
- AND: two events must happen simultaneously to trigger P_3 .

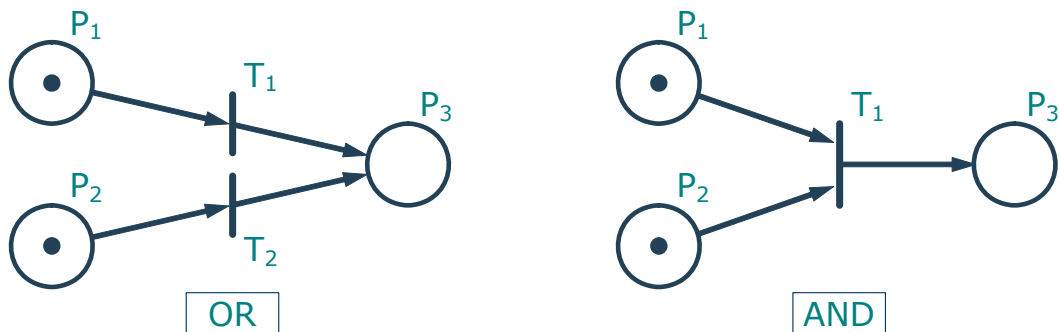


Fig. 4 Petri Net examples of Boolean logic

Using these basics, specific dependability modelling can be defined, such as redundancy. In principle, there are two kinds of redundancy:

- hot redundancy: two systems are running simultaneously, see figure 5;
- cold redundancy: one system is running, a spare is standing by.

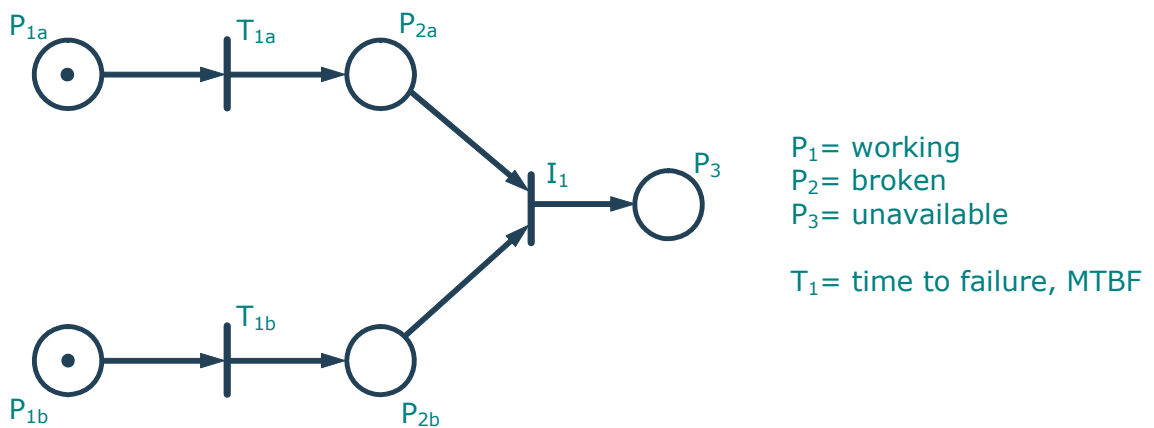


Fig. 5 Petri Net examples for two hot redundant components without repair

When modelling components of a system, it may be cumbersome to always draw all of the arcs and transitions for a standard situation. A shorthand notation is used to graphically define often-used models.

For instance, if the functioning of a component (b) depends on that of another (a), then the shorthand notation in figure 6 is used. If component (a) is down, then the transition from nominal to down of component (b) is enabled. Likewise, in case component (a) is nominal, the transition from down to nominal is enabled. This introduces the concept of *enabling arcs*, which do not remove the token from their input place, while still ensuring that the transition only fires once. The enabling arcs allow parts of the Petri Net to keep their own tokens, thus representing local components.

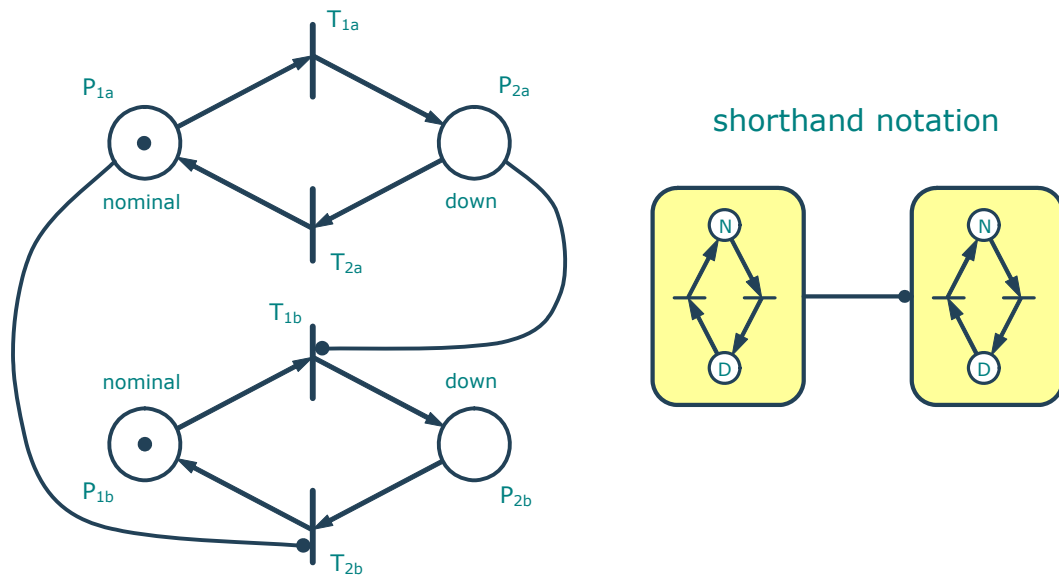


Fig. 6 Shorthand notation for a component that depends on another component

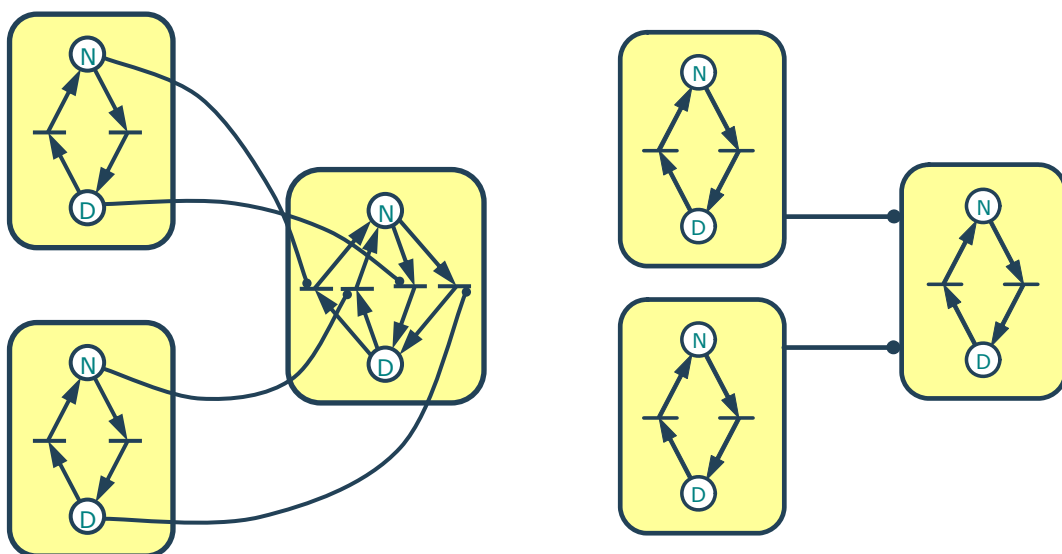


Fig. 7 Shorthand notation for a component that depends on multiple components

4.2 Overview of Galileo

The Galileo system basically consists of a number of satellites, disseminating the Signal-In-Space (SIS) containing timing and navigation information, and a ground infrastructure to monitor the satellites, measure their position, and update the information. Some satellites also broadcast integrity information, which –like the navigation information– is produced by the ground segment.

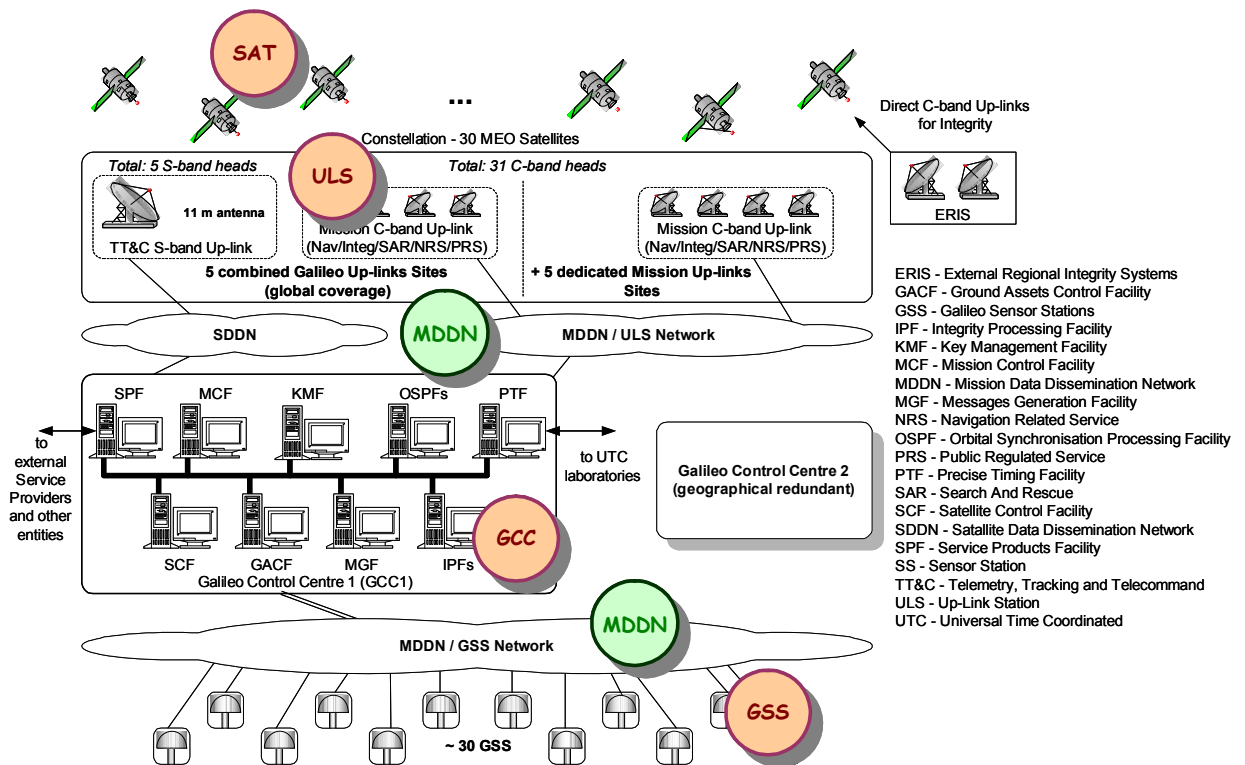


Fig. 8 Overview of Galileo architecture

The continuous measuring of the satellites' positions and the consequently updating of the navigation data represent a loop that must be modelled by the dependability simulation.

Figure 9 shows this loop with on the left hand side the measuring chain and on the right hand side the update chain. Central components are the satellites (SAT) and the ground control centres (GCC) that contain the processing facilities. The Galileo sensor stations (GSS) and up-link stations (ULS) form the interfaces between them.

The environment (ENV) and the on-ground data dissemination network (MDDN) represent other components that may introduce feared events or equipment failures. Especially, the environment is the prime source of navigation errors.

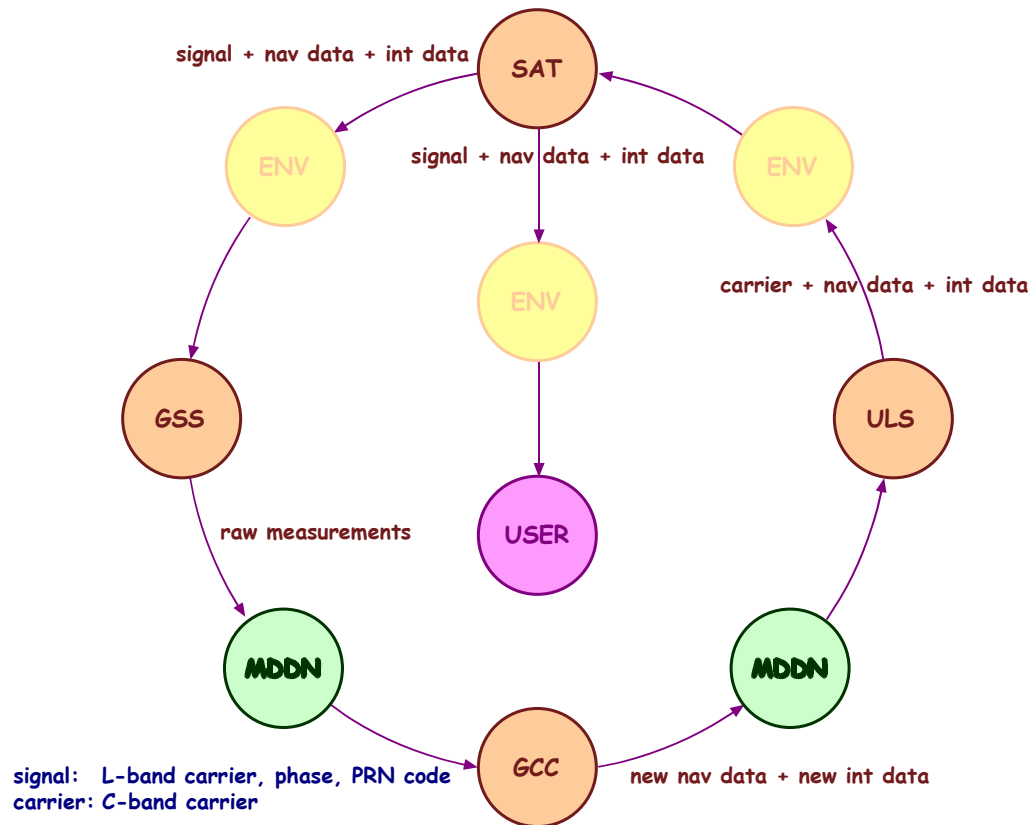


Fig. 9 End-to-end model of Galileo showing the data loop and the user as end point

In the middle of figure 9, the user is depicted, who receives the signal through the environment. Each may introduce feared events that are outside the scope of the Galileo system, but add to the Integrity and Continuity Risk.

4.3 Modelling of Galileo Measuring Chain

This paper will not describe the complete DCPN model of Galileo, but will focus on the measuring chain, as this is the most complex to model. There are i satellites, k sensor stations, and 1 active control centre (with hot backup of a second centre to be modelled as well).

Each of the components will influence the measurements in one way or another: the satellites may be down, i.e. generating no SIS whatsoever, the environment may disturb the signal introducing measurement errors, the sensor stations may be down as well (but other GSSs may

have the same satellites in view), and finally the control centre may be down or produce incorrect data.

The relation between these components and the measurements is modelled in the following (local) DCPNs.

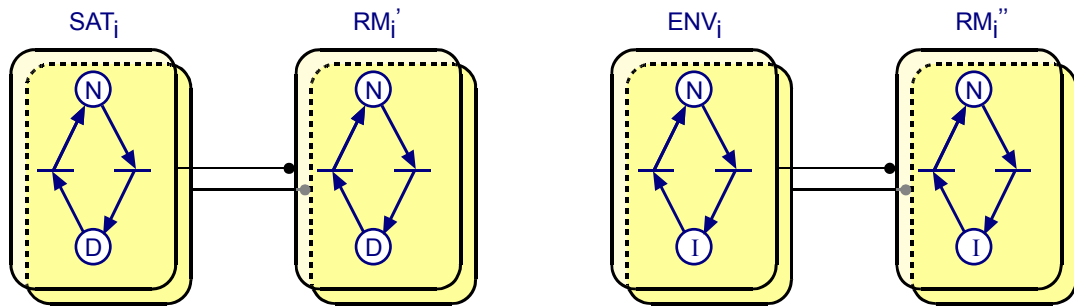


Fig. 10 One-on-one relations between satellites and raw measurements, and between environment and raw measurements

The diagrams in figure 10 show that the state of each component (satellite or local environment) is directly reflected on the state of its corresponding raw measurement. The difference is that a satellite can go down (D) whereas the environment can cause incorrect measurements (I). Note that other feared events that can cause incorrect measurements, e.g. satellite clock instability, may be treated the same as the environment.

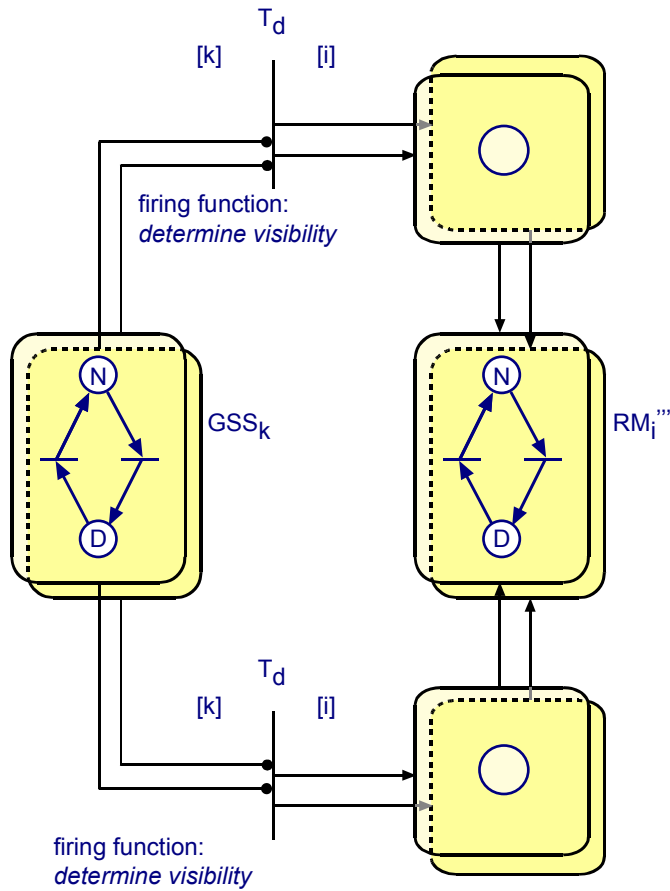


Fig. 11 k-to-i relations between sensor stations and raw measurements

The difference in numbers of sensor stations (k) and raw measurements of the satellites (i) require dedicated transitions for nominal and down states, with firing functions using pre-defined visibility tables. A coloured token contains the number of GSSs that are producing raw measurements for a satellite.

Now, the three separate DCPNs for raw measurements must be combined into a DCPN that reflects the overall state of the measurements for each satellite.

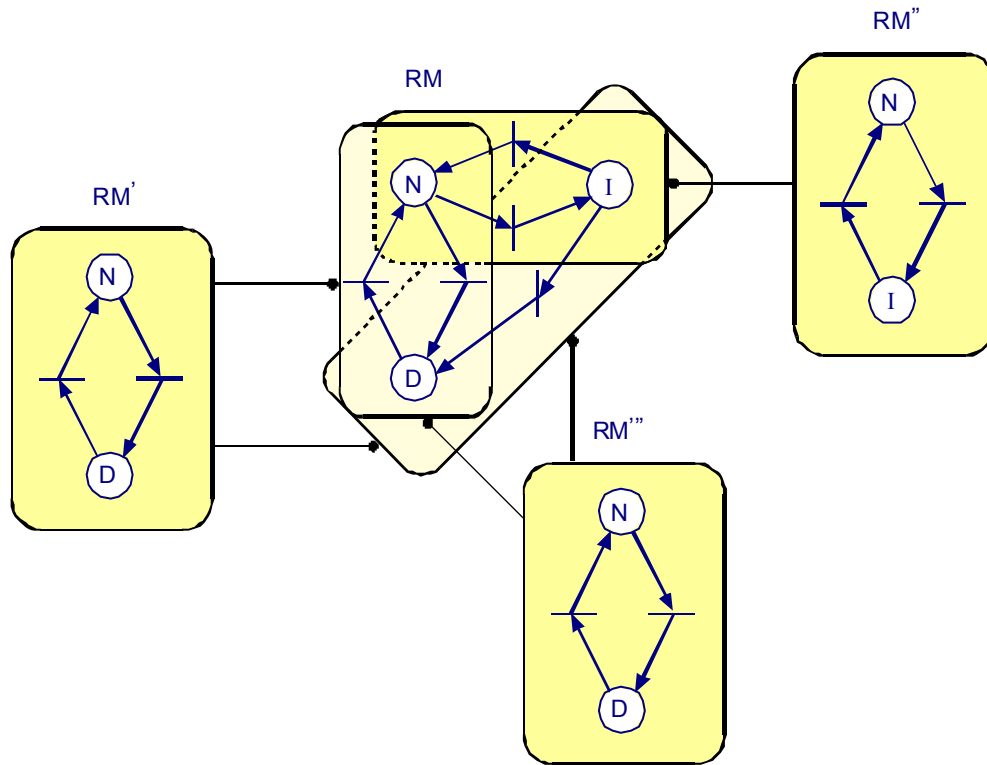


Fig. 12 Combination of the component raw measurements into one measurement

Figure 12 in fact shows a rather complicated diagram with multiple dependencies. Without the shorthand notation, the diagram would be full with arcs and transitions (see figures 6 and 7 for an explanation of the shorthand notations).

The diagram shows that the RM can go down when either RM' or RM''' is down and it can be incorrect when RM'' gets incorrect. Furthermore, a state transition from incorrect to down is possible, e.g. if a satellite fails, there simply are no measurements, even when the environment is introducing a feared event. The state transition from down directly to incorrect is not necessary.

After the measurements have been collected, the GCC processes the data and produces derived navigation quantities, such as Signal-In-Space Accuracy (SISA). Furthermore, it provides timing information from its Precision Timing Facility. The updating of this data needs to be done at least every 100 minutes.

For the integrity loop, similar quantities are produced, but here the updating is every second. The integrity loop acts as a barrier for the navigation service: once an event has an effect on the navigation performance, the integrity loop will come into play.



Prior to the processing, the GCC applies barriers to filter outliers in the measurements and performs several checks for data consistency. This is handled by the firing function of transition T_1 in figure 13. This transition, of course, is only enabled if the GCC is working. The level of incorrectness is derived from the duration of having a reduced number of measurements relative to the navigation batch size.

The scheduling of the produced navigation and integrity information is handled by the firing function of transition T_2 .

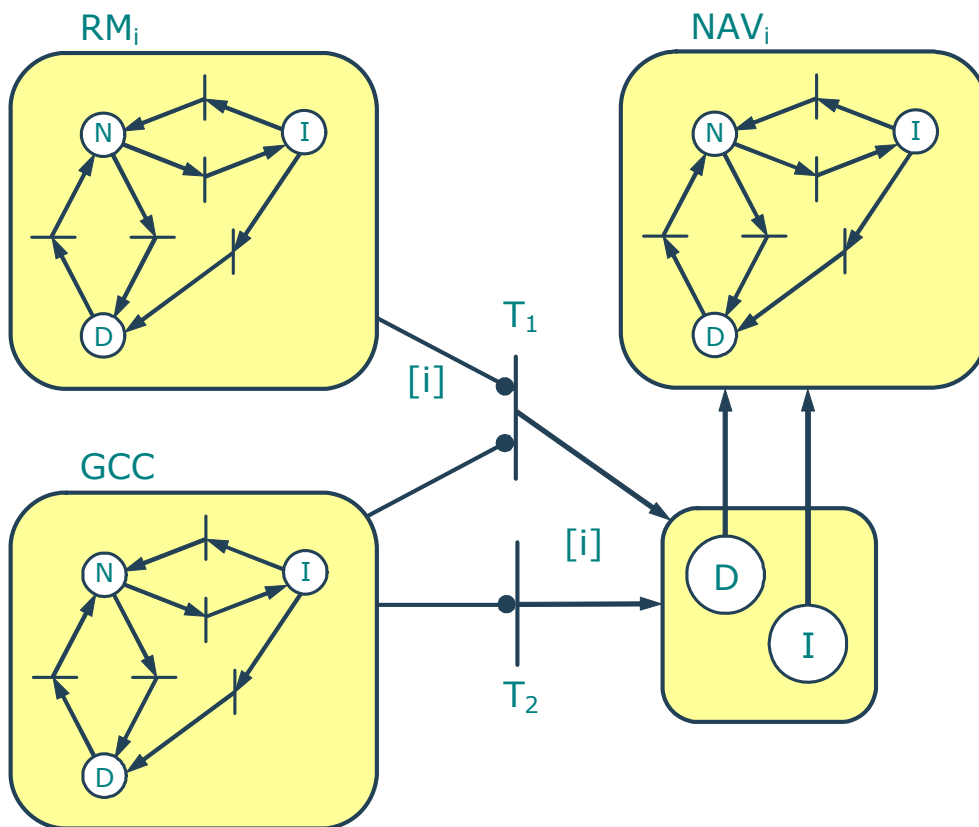


Fig. 13 Transition from raw measurements into navigation data

The GCC also provides the up-link schedule as part of the update chain. This chain will only have an impact on the (timely or not) arrival of the navigation and integrity data, as it is assumed that there will be interface barriers, such as cyclic redundancy checks or parity bits, to mitigate any message corruption during transport.



Each of the models can be extended to incorporate more feared events (e.g. events from a component fault tree) and to include internal redundancy.



5 Conclusion

This paper presented a Petri Net simulation approach for dependability parameters of Galileo, based on the Netherlands Valileo 2002 initiative. This work has continued in 2004 and 2005 to create a Dynamically Coloured Petri Net (DCPN) kernel, which has shown its potential for faster-than-real-time (factor 10^8), long-duration simulations, ready for Monte Carlo analysis.

Furthermore, a detailed Galileo DCPN model has been defined for measuring chain of the navigation loop. This has been partly implemented (work in progress).

For the future, it is foreseen to extend the Galileo DCPN model with the uplink of navigation data, and the generation and uplink of integrity data.

6 References

1. Everdij, M.H.C.; Blom, H.A.P., *Piecewise deterministic Markov Processes represented by Dynamically Coloured Petri-Nets*, NLR-TP-2000-428, 12/1999, revised in 2003, <http://www.nlr.nl/public/library/2000/2000-428-dcs.html>
2. Oving, B.A., *VALILEO 2002 - WP 4100 - Dependability Tool Definition*, NLR-CR-2002-456, V/TN/02/D4100/1/NLR, December 2002.
3. Park, M. van der, *VALILEO 2002 - DCPN based Galileo Model for Dependability Assessment*, NLR-CR-2002-456, V/TN/02/D4200/1/NLR, December 2002.