Using formal methods in signal processing

AWWAMA EMAD, ALARSAN MOHAMED, KADI MOHAMMAD, KRAYEM SAID, LAZAR IVO, RIHAWI AHMAD
Department FAI, University UTB
Nad Stráněmi 4511, 760 05 Zlín
CZECH REPUBLIC
emad_awwama@hotmail.com, mkady@arcon.com.sa, drsaid@seznam.cz, Ivo.Lazar@centrum.cz,
amadr@ria-delta.com http://www.utb.cz/fai

Abstract: - Philosophy is centrally concerned with arguments. The first question to be asked of any argument (or inference) is whether or not it is valid: that is, does its conclusion really follow from the cited premises? Validity of inference is the central problem of deductive logic. While Rigorous descriptions promise to improve system reliability, design time and comprehensibility, they do so at the cost of an increased learning curve; the mathematical disciplines used to formally describe computational systems are outside the domain of a traditional engineering education. In addition, the meta-models used by most formal methods are often limited in order to enhance provability. There is a notable tradeoff between the need for rigor and the ability to model all behaviors. As formal methods become more common, engineers will have to learn type theory, modern algebra and proof techniques. Ultimately, engineers will have to think more like mathematicians.
Event-B is intended to be used to create complex models. Without appropriate tool support this would not be possible. This article presents justifications and explanations for the choices that have been made when designing the Event-B notation. It is used, for discrete systems modelling by refinement.

Key-Words: verification, rodin, formal modeling, model checking, specification, invariants, context, machine, events, refinement

1 Introduction

Formal verification of a program is the mathematical proof that it does what is expected of it. The 21st century has seen a vast worldwide interest in formal methods. Four journals (Automated Reasoning, Logic and Algebraic Programming, Formalized Mathematics, and Science of Computer Programming) and over a dozen yearly conferences, each of which has been held at least since 2000, are specifically devoted to these matters.

Centers of ongoing formal methods research include Argonne, Berkeley, Bialystok (Poland), Cambridge, Clemson, HP, INRIA, Iowa State, Karlsruhe, Lausanne, Microsoft, MITRE, Munich, NYU, Penn, Praxis, and SRI. Methods have been developed for Java (JML), Ada (SPARC), C#, C, and Eiffel (Spec#), Haskell, Ocaml, and Scheme (Coq), Pascal (Sunrise), Modula-3 (ESC), and a number of special-purpose languages.

Formal methods are techniques used to model complex systems as mathematical entities. By building a mathematically rigorous model of a complex system, it is possible to verify the system's properties in a more thorough fashion than empirical testing.

2 Formal methods

While Rigorous descriptions promise to improve system reliability, design time and comprehensibility, they do so at the cost of an increased learning curve; the mathematical disciplines used to formally describe computational systems are outside the domain of a traditional engineering education. In addition, the meta-models used by most formal methods are often limited in order to enhance provability. There is a notable tradeoff between the need for rigor and the ability to model all behaviours.

2.1 Using Formal methods

Formal methods are used in specifying software: developing a precise statement of what the software is to do, while avoiding constraints on how it is to be achieved.

2.1.1 A complex specification
Examples of these methods include ASM (Borger & Stark, 2003), B (Abrial, 1996), and VDM (Jones, 1990) [1,2,3]. A specification is a technical contract between programmer and client to provide them both with a common understanding of the purpose his software. The client uses the specification to guide application of the software; the programmer uses it to guide its construction. A complex specification may be decomposed into sub-specifications, each describing a sub-component of the system, which may then be delegated to other programmers, so that a programmer at one level becomes a client at another design by contract (Meyer, 1991) [4].

2.1.2 Framework – Event-B
We present our formal development framework – Event-B (see Figure 1). The Event-B formalism is a state-based formal approach that promotes the correct-by-construction development paradigm and formal verification by theorem proving. Event-B has been specifically designed to model and reason about parallel, distributed and reactive systems [7].

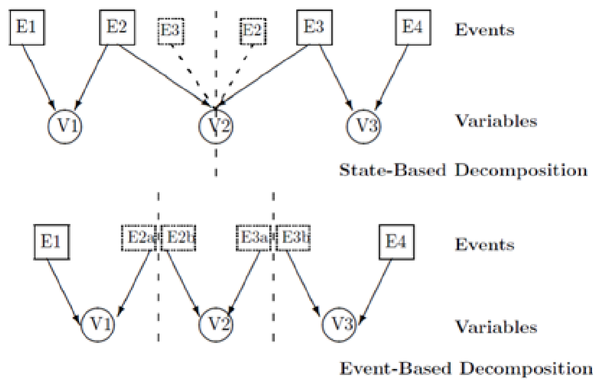

Fig.1 Comparing Event-Based and State-Based Decomposition

3 Development tool
Formal methods are the application of mathematics to model and verify software or hardware systems (Storey, 1996).

Mathematically based languages can be used to write specifications of systems using precise rules. The specification can be interpreted unambiguously and can be formally verified to ensure consistency and correctness. Formal methods have been used to develop applications such as air traffic control (Hall, 1996), railway signaling (Behm et al., 1999) and transaction processing systems (Houston & King, 1991) [5,6].

Formal verification involves the application of mathematical proofs to every possible behavior allowed by a specification (Abrial, 1996) [7]. In a state-based specification the behavior is a transformation of the system moving from one state to another. Proof obligations are generated using the specification and the language rules.

3.1 Modelling elements
These proof obligations then need to be discharged using properties of the specification.

Event-B is a mathematical approach for developing formal models of systems (Abrial & Hallerstede, 2006) [7,17]. An Event-B model is constructed from a collection of modelling elements. These elements include invariants, events, guards and actions. The modelling elements have attributes that can be based on set theory and predicate logic. Set theory is used to represent data types and the manipulation of data. Logic is used to apply conditions on the data (Hallerstede, S. 2007) [21].

3.2 State model
The development of an Event-B model goes through two stages; abstraction and refinement. The abstract machine specifies the initial requirements of the system0 Refinement is carried out in several steps with each step adding more detail to the system, generally, but not exclusively, in a top-down manner Reactive systems (Harel & Pnueli, 1985) are systems that continually respond to changes in their environment.

The focus on atomic events in Event-B creates a representation of a reactive system (Jones, 2005). The model transitions are triggered by changes in the state of the model, which can represent the system's environment. The guard on an event will allow or prevent an event from occurring depending on the state of the model.

When none of the guards are true the system is deadlocked. Event-B is designed for modelling distributed systems (Abrial & Hallerstede, 2006) [7]. It implements the theory of discrete transition systems. Discrete transition systems, or action systems, model atomic actions that can be performed in parallel providing the actions do not affect the same state variables One method for specifying concurrency in Event-B is to model each update as a group of potentially interleaving atomic events (Edmunds & Butler, 2008) [9].

This allows the model to specify how concurrent execution can be dealt with by the system being modelled specifying a distributed system in Event-B takes a global approach. Rather than creating a specification for each component of the system it is modelled as a whole along with its environment (Abrial, J.-R., Butler, M., Hallerstede, S., & Voisin, L. 2006) [17]. The model is closed in that it reacts only to changes in its internal state. Initially states are modelled abstractly with the events that describes the main goal of the system. Detail is added through refinement to describe the final distributed system. The ability to add new events and refine single events into multiple concrete events allows the functionality of the system to expand beyond that modelled in the abstract machine. Refinement ensures that the refined models are consistent with the abstract machine.

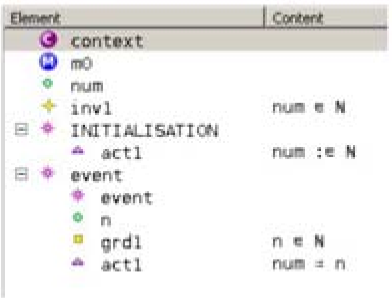

Fig.3 Context event-B

4 Event-B

In Event-B, a system model is specified using the notion of an abstract state machine (Abrial, 2010). An abstract state machine encapsulates the model state represented as a collection of model variables, and defines operations on the state, i.e. it describes the dynamic behavior of the modelled system.

A machine may also have the accompanying component, called context (see Figure 2). A context might include user-defined carrier sets, constants and their properties, which are given as a list of model axioms. In Event-B, the model variables are strongly typed by the constraining predicates called invariants.

Moreover, the invariants specify important properties that should be preserved during the system execution. A general form of Event-B models is given in (see Figure 3)

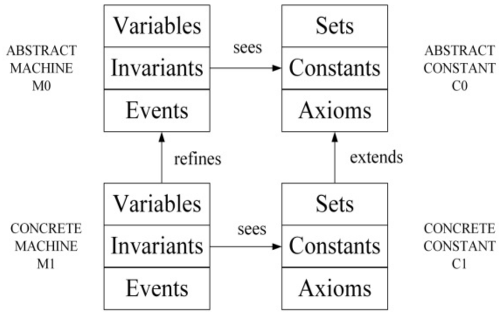

Fig.2 Machine and Context relationship

We outline the general structure of an Event-B specification. A specification consists of a static part, specified in a context, and a dynamic part, specified in a machine (Abrial, J.-R., Butler, M., Hallerstede, S., & Voisin, L. 2006) [17].

4.1 Event-B context

An Event-B context contains the following elements:

- Sets: Abstract types used in specification to distinguish various entities
- Constants: Logical variables whose value remain constant
- Axioms: Predicates that specify assumptions about the constants.

4.2 Event-B machine

The dynamic behavior of the system is defined by the set of atomic events specified in the Events clause. Generally, an event can be defined as follows introduction to the event-B method and the Rodin.

An Event-B machine contains the following elements:

- Variables: State variables whose values can change
- Invariants: Predicates that specify properties about the variable that should always, remain true.
- Initialization: Initial values for the abstract variables
- Events: Guarded actions specifying ways in which the variables can change. Events may have parameters.

A machine may see the static elements defined in a context meaning that these elements are visible within the machine. The structure of a specification is outlined (see Figure 3).

4.2.1 Behavior machine – Event-B

The machine is uniquely identified by its name M. The state variables, ***v*** are declared in the Variables clause and initialized in the Init event. The variables are strongly typed by the constraining predicates. I given in the Invariants clause. The invariant clause might also contain other predicates defining properties that should be preserved during system execution (Abrial, J.-R., Butler, M., Hallerstede, S., & Voisin, L. 2006) [17].

4.2.2 Refinement machine – Event-B

An Event-B machine M1 may be declared to be a refinement of some other Event-B machine M0 (see Figure 5, Figure 6 & Figure 10). In this case we refer to M0 as the abstract machine and M1 as the refined machine. Machine M1 is said to be a correct refinement of M0 if any behavior that may be exhibited by M1 is also a possible behavior of M0 (see Figure 11).

Refinement represents our expectation that the behavior of M1 should conform to the behavior of M0 (see Figure 10). Of course declaring that M1 refines M0 does not on its own guarantee the correctness of a refinement. Rather the declaration gives rise to proof obligations that need to be discharged in order to guarantee the correctness of a refinement.

When refining a machine, it is common to specify new types and constants to be used in the refinement. This is achieved by specifying a new context for the refined machine. If the specification of any new types and constants depend on the types and constants used by the abstract machine, the new context is declared to be an extension of the context of the abstract model. The relationships between machine and its refinement, as well as their respective contexts, is illustrated by Figure 3 [17,18].

This figure shows the refinement declaration from M1 to M0, together with the relationships with their contexts [20,21].

A refined context C1 is declared as an extension of the abstract context C0 (see Figure 4 & Figure 9) meaning context C1 may refer to types and constants specified in context C0. The dashed line from machine M1 to context C0 indicates that M1 implicitly see definitions in C0 (via C1) (Abrial, J.-R. 2010) (Jastram, M., & Butler, M. 2004) [19].

5 Case study – signal processing

The purpose of our work is to study and model the spread of DVB-T signal between the transmitter and the antenna using the Event B method and acquire new measurement knowledge.

5.1 Digital broadcasting

Digital broadcasting replaced in the last years the previous analogue one and compared to that, it brings many advantages. It provides many facilities and enables a far more efficient use of the available radio frequency spectrum than the previous analogue transmissions. DVB-T is the most widely used digital television standard in use around the globe for terrestrial television transmissions.

Digital broadcasting replaced in the last years the previous analogue one and compared to that, it brings many advantages. It provides many facilities and enables a far more efficient use of the available radio frequency spectrum than the previous analogue transmissions. DVB-T is the most widely used digital television standard in use around the globe for terrestrial television transmissions [13,14,15].

5.1.1 Antenna parameters

The primary parameters of the antenna radiation pattern shown in diagrams usually depending on the azimuth (0–360 degrees) and elevation angle. Other important parameters are the beam angle, bandwidth, polarization, etc., [10,11,12].

Antenna gain is different from the amplifier gain because the antenna has no active circuitry and cannot therefore increase the signal strength. Antenna gain is measured in decibels. The shape of the radiated field differs from the ideal isotropic antenna (dBi unit) [12].

The ideal isotropic antenna transmits or receives evenly into (from) all directions, and of course in the real world, it cannot be construct. Sometimes also compares the gain of a dipole antenna (isotropic ant.), which can be already constructed and used as a reference.

5.1.2 Frequency spectrum

In the past, the frequency spectrum was divided into channels with a fixed difference of frequencies — in the Czech Republic the frequency bands of 8MHz were used. This range was large enough for one analog television broadcast to be transmitted through. Each such a channel has its fixed central frequency. For DVB-T channels 21-69 can be used, which cover wholly the UHF (ultra high frequency) spectrum 470–862 MHz (European Telecommunications Standards Institute 2009) [10,11].

Digitalization of signals and their further compression bring the pleasant effect that now more TV and other signals can go through one channel. Such sets of signals are called multiplexes and they contain compressed television broadcasts, audio data and other supplementary services like superteletext, EPG (Electronic Program Guide), etc.

These are multiplexed into MPEG program streams (MPEG-PSs). Currently used MPEG format is MPEG-2 [12].

5.1.3 Transport stream

One or more MPEG-PSs are joined together into an MPEG transport stream (MPEG-TS) and this is the basic digital stream which is being transmitted and received by TV sets or home Set Top Boxes. Whereas DVB-T transmitter applies on multiplexes coding by orthogonal frequency-division multiplexing (COFDM or OFDM) modulation, set top boxes are awaited to do the opposite process of demodulation and decoding (Fisher, W. 2010) [14].

The contents of each multiplex is given by agreement and licences. Nowadays in the Czech Republic four statewide and four regional multiplexes are established. For example MUX1 contains four TV and eight audio signals [11,12]. The concrete channel, where each multiplex is transmitted through, depends on the concrete transmitter according to the map of coverage. However, orthogonal frequency-division multiplexing also facilitates the design of single frequency networks, where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be combined constructively, rather than interfering as would typically occur in a traditional single-carrier system (Czech Telecommunications Office, 2008) [12].

In the past, the frequency spectrum was divided into channels with a fixed difference of frequencies — in the Czech Republic the frequency bands of 8MHz were used. This range was large enough for one analog television broadcast to be transmitted through. Each such a channel has its fixed central frequency. For DVB-T channels 21-69 can be used, which cover wholly the UHF (ultra high frequency) spectrum 470–862 MHz. Digitalization of signals and their further compression bring the pleasant effect that now more TV and other signals can go through one channel [12].

5.2 Mutiplexes

Such sets of signals are called multiplexes and they contain compressed television broadcasts, audio data and other supplementary services like superteletext, EPG (Electronic Program Guide), etc.

These are multiplexed into MPEG program streams (MPEG-PSs) (Handley, M. 2010). Currently used MPEG format is MPEG-2. One or more MPEG-PSs are joined together into an MPEG transport stream (MPEG-TS) and this is the basic digital stream which is being transmitted and received by TV sets or home Set Top Boxes.

Whereas DVB-T transmitter applies on multiplexes coding by orthogonal frequency-division multiplexing (COFDM or OFDM) modulation, set top boxes are awaited to do the opposite process of demodulation and decoding [14,15].

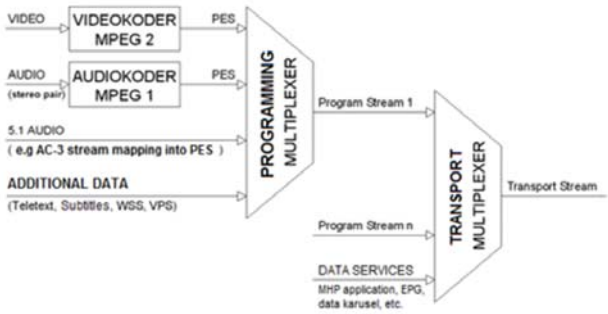

Fig.4 Multiplexer into stream

The contents of each multiplex is given by agreement and license. Nowadays in the Czech Republic four statewide and four regional multiplexes are established. For example MUX1 contains four TV and eight audio signals. The concrete channel, where each multiplex is transmitted through, depends on the concrete transmitter according to the map of coverage. However, orthogonal frequency-division multiplexing also facilitates the design of single frequency networks, where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be combined constructively, rather than interfering as would typically occur in a traditional single-carrier system [12].

5.2 Transmitters

The transmitter transmits a signal to the ether with a certain frequency, modulation and power. Physical principles of electromagnetic spreading waves in the case of digital video broadcasting (DVB-T) is the same as the analog.

Reception range of the transmitting antenna (transmitter) is limited in proportion with its performance. Transmitter power is measured in effective radiated power ERP [W], the signal is distributed in a horizontal or vertical polarization. On the receiving antenna signals fall with different polarizations, distant and close transmitters with different levels quality and modulation (see Table 1).

The receiving antenna must have sufficient earnings in [dBi] (to the isotropic radiator), directionality [12,14].

5.3 System behavior using Event-B

The context trsnsmitt_0 contains eight constants representing main table 1. The set column contains the values for these attributes [18,20].

```
context transmitt_0

constants chan freq center_f band_range mux trans_name erp r
sets columns

axioms
@column_partition partition(columns,{chan},{freq},{center_f},
                            {band_range},{mux},{trans_name},{erp})
@transmitt_name r ⊆ columns↔columns
```

Fig.5 Context of the project Transmitter

```
machine Machine_0 sees transmitt_0//

variables v

invariants
@inv1 v∈ columns

events
  event INITIALISATION
then
@act1 v :∈ columns
  end

  event transmit
  when
   @grd1 v ∈ columns
  then
  @act1 v≔ trans_name
  end
end
```

Fig.6 Abstract Machine for project Transmitter

6 Conclusion

Communication and negotiation are very important characteristics of distributed systems; and in this paper we have presented some of the basic concepts in formal method using Event-B also we have presented verification of protocols in distributed systems., so we can say, event-B allows us to define a kind of modeling methodology by write the correct mathematical notions; wherefore we can apply event-B in modeling many different complex projects, but we should choose carefully invariants and variables to ease effort of proof.

As well as the Rodin tool offers reactive environment for constructing and analyzing models as do most modern integrated development environments, and provides integration between modeling and proving whereas this is important feature for the developers to focus on the modeling task without switch between different tools to check proving in same time.

The intent of this paper to give some insights on modelling and formal reasoning using Event-B method.

Contexts are used to model static properties of a model, things that do not change over time — contains constans, sets and axioms. A context describes the static elements of a model [21,22].

A machine describes the dynamic behavior of a model by means of variables whose values are changed by events. A central aspect of modelling a machine is to prove that the machine never reaches an invalid state [19,20,21].

Proof obligations are generated using the specifications and the language rules. These proof obligations then need to be discharged using properties of the specifications.

In this case, the model used for writing Camille editor. Allows easy editing and writing mathematical symbols. Camille editor it is necessary to install as additional plug-in in the Rodin tool [22].

Next step (refining) model would consist in terms of BER DVB-T signal error conditions and QAM reception of terrestrial broadcasting local and remote reception [14].

Channel	Frequency (MHz)	band center frequency (MHz)	Band, pol.	MULTIPLE X	TRANSMITTER	ERP (kW)
C 21	470 – 478	474	IV/H	ORS MUX B (Austria)	Jauerling	100
C 24	494 – 502	498	IV/H	ORS MUX A (Austria)	Wien – Kahlenberg	100
C 25	502 – 510	506	IV/H	MUX 3 (CZ)	Zlín – Tlustá Hora	100
C 27	518 – 526	522	IV/V	MUX 3 (SK)	Bradlo, Dubník, Holíč, Kamzík (SFN)	0,11/22 /0,324/ 50
C 29	534 – 542	538	IV/H	MUX 1 (CZ)	Děvín – Mikulov	25
C 33	566 – 574	570	IV/H	MUX 1 (CZ)	Zlín – Tlustá Hora	100
C 34	574 – 582	578	IV/H	ORS MUX B (Austria)	Wien – Kahlenberg	100
C 35	582 – 590	586	IV/H	Regional – Skalica (SK)	Malého – Skalica	?
C 40	622 – 630	626	V/H	MUX 2 (CZ)	Brno – Kojál, Mikulov – Děvín (SFN)	100/25
C 41	630 – 638	634	V/H	Reg. MUX 7	Uherské Hradiště	10
C 42	638 – 646	642	V/H	MUX 4 (CZ)	Uh. Hradiště, Zlín (SFN)	10/10
C 46	670 – 678	674	V/H	MUX 4 (CZ)	Hodonín - Kaplansko	10
C 49	694 – 702	698	V/H	MUX 2 (CZ)	Zlín – Tlustá Hora	100
C 55	742 – 750	746	V/V	MUX 1 (SK)	V. Javořina (SK)	50
C 56	750 – 758	754	V/V	MUX 2 (SK)	V. Javořina (SK)	50
C 57	758 – 766	762	V/V	MUX 3 (SK)	Javořina (SK)	50
C 58	766 – 774	770	V/H	Reg. MUX 8	Zlín – Tlustá Hora	2
C 59	774 – 782	778	V/H	MUX 3 (CZ)	Brno – Kojál,	100

Table 1. Table UHF IV and V band — frequences and channels (For region — South Moravien, district Hodonín)

```
machine Machine_1
refines Machine_0 sees transmitt_0//
variables   v Q

invariants
@act1 Q ⊆ columns
@act2 trans_name ∈ Q

events
  event INITIALISATION
  then
  @act1 v:∈ columns
  @act2 Q≔columns
  end

  event transmit refines transmit
  any x
  where
  @grd1 x ∈ Q
  @grd2 Q = {x}
  then
  @act1 v≔x
  end

event transmitter_name
  any x
  where
  @grd1 x ∈ columns ∧ x≠chan ∧ x≠freq ∧ x≠center_f ∧ x≠band_range ∧ x≠mux ∧ x≠erp
  then
 @act1  v≔x
  end
end
```

Fig.7 First refinement Machine

References:

[1] Borger, E., & Stark, R. (2003). *Abstract State Machines: A Method for High-Level System Design and Analysis*. Springer, ISBN 978-3-642-18216-7.

[2] Abrial, J.R. (2005). *The B-Book: Assigning Programs to Meanings*. Cambridge University Press; 1 edition (November 3, 2005), 978-0521021753.

[3] Jones, C.B. (1990). *Systematic software development using VDM*. New York: Prentice Hall

[4] Meyer, B. (1991). In *Advances in Object-Oriented Software Engineering*, D. Mandrioli and B. Meyers, Eds. Prentice Hall, 1-50.

[5] Behm, P., Benoit, P., Faivre, A., & Meynadier, J.-M. (1999). météor: A successful application of B in a large project. In *World Congress on Formal Methods*, J. M. Wings, J. Woodcock, & J. Davies, Eds. Lecture Notes in Computer Science, vol. 1708. Springer, Berlin, Heidelberg, 369–387.

[6] Houston, I., & King, S. (1991). Experiences and results from the use of Z in IBM. In VDM '91: *Formal Software Development Methods*. Lecture Notes in Computer Science, vol 551. Springer, Berlin, Heidelberg, 558–595.

[7] Abrial, J-R & Hallerstede, S. (2006). *Refinement, decomposition, and instantiation of discrete models: Application to Event-B*. IOS Press, 1–28.

[8] Harel, D., & Pnueli, A. (1985). On the development of reactive system. *Logics and Models of Concurrent Systems*. Krzysztof R. Apt. Spring-Verlag, Berlin, Heidelberg, New York, Tokyo. pp. 477–498.

[9] Edmunds, A., & Butler, M. (2008). Linking Event-B and concurrent object-oriented programs. *In: Proc. Refine 2008 – Internacional Refinement Workshop (2008)*. Retrieved from http://eprints.ecs.soton.ac.uk/16003/

[10] (European Telecommunications Standards Institute 2009, *Digital Video Broadcasting (DVB)*, ETSI TS 101 154 V1.9.1, Accessed on: 2009-09 Retrieved from http://www.etsi.org/deliver/etsi_ts/101100_101199/101154/01.09.01_60/ts_101154v010901p.pdf

[11] European Telecommunications Standards Institute 2009, *Digital Video Broadcasting (DVB)*, ETSI TS 102 591-2 V1.1.1, Accessed

on: 2010-02 Retrieved from http://www.etsi.org/deliver/etsi_ts/102500_102599/10259102/01.01.01_60/ts_10259102v010101p.pdf

[12] Czech Telecommunications Office, *163 DECREE on theme thod for determining terrestrial television broadcasting signal coverage,* Accessed on: 2008-04-30 Retrieved from https://www.ctu.eu/sites/default/files/obsah/stranky/62476/soubory/annualreport-2008.pdf

[13] Handley, M. *System Streams*, (n.d). Retrieved from http://www0.cs.ucl.ac.uk/teaching/GZ05/10-system-streams.pdf

[14] Fisher, W. (2010). *Digital Video and Audio Broadcasting Technology*, Springer, ISBN: 978-3-642-11611-7 (Print) 978-3-642-11612-4 (Online), Springer**.**

[15] Fisher, W. (2010). *Measuring DVB-T Signals*, *Springer*,**153**Volumes from **2002 – 2017 (March 2010),** 2010. pp 421-450, ISBN: 978-3-642-11611-7 (Print) 978-3-642-11612-4 (Online).

[16] Abrial, J.-R. (2010). *Modeling in Event-B*, Swiss Federal University (ETH), Zürich, 2010. ISBN: 978-0-521-89556-9

[17] Abrial, J.-R., Butler, M., Hallerstede, S., & Voisin, L. (2006). *An open extensible tool environment for Event-B*, Springer, Vol. 4260 of*LNCS*, 2006, pp 588–605, ISSN 0302-9743.

[18] Abrial, J.-R. (2010). *Modeling in Event-B*, Cambridge, 2010. ISBN-13 978-0-511-72976-8, CAMBRIDGE UNIVERSITY PRESS

[19] Jastram, M., & Butler, M. (2004). *Rodin User's Handbook*: Covers Rodin v.2.8, CreateSpace Independent Publishing Platform, 2014. ISBN 10: 1495438147 ISBN 13: 9781495438141, Retrieved from https://www3.hhu.de/stups/handbook/rodin/current/pdf/rodin-doc.pdf.

[20] Rodin User's Handbook v.2.8, Accessed on: 2014-02-04, Retrieved from https://www3.hhu.de/stups/handbook/rodin/current/html/introduction.html.

[21] Hallerstede, S. (2007), *Justifications for the Event-B Modelling Notation*. B: Formal Specification and Developmentin B, *Springer*, Vol. 4355, 2007., pp 49-63., ISSN 0302-9743

[22] *Rodin Platform 3.2 Release Notes*, Accessed on: 2015-11-05 Retrieved from http://wiki.event-b.org/index.php/Rodin_Platform_3.2_Release_Notes