Proposition of a guide for investigating, modeling and analyzing system operating modes: OMAG

Vincent Chapurlat, Nicolas Daclin

Laboratoire de Génie Informatique et d'Ingenierie de Production (LGI2P) – ENS Mines d'Alès - Site de l'Ecole des Mines d'Alès - Parc scientifique Georges Besse, f30035 Nîmes cedex 5, France – tél. : (+33) 466 387 066 – {surname.name}@mines-ales.fr

Abstract - This paper presents and illustrates an approach that allows designers exploring and reasoning, checking and then arguing the consistency of the operat- ing modes of a system. The goal is to help designers to build system's functional architecture by linking operating modes, allowed configurations and operational scenarios *i.e.* the set of functions displayed by the system in each mode in order to fulfill its mission taking into account its current configuration, requirements and environment. This approach is implemented as a guide called OMAG and is here illustrated on a vehicle design.

Keywords - System Engineering, System modeling, Operating Mode, Operational scenario, Verification

Introduction

System Engineering (SE) (INCOSE 2011) (SeBOK 2012) (Fiorèse *et al.* 2012) is a design approach approved and largely used in industry. Based on concepts and principles coming from system sciences *e.g.* (Féliot 2007), SE promotes simulta- neously a model based approach (INCOSE 2008) and a process oriented approach (ISO 2008) covering the whole cycle of a system design project. We consider here only technical activities related to architecture design (Sharman *et al.* 2004) (Blanchard *et al.* 2011). The goal is here to help designers' to make emerge poten- tial alternative solutions of functional architecture. We propose for this to cover some of designer's modeling and verification needs:

- To find what are the relevant operating modes of the system considering its mission and the moving environment in which this mission has to be fulfilled.
- To become able to imagine how the system can evolve from an operating mode to another one when considering various events (external coming from environment as internal coming from the system itself *e.g.* dysfunctions) and system configurations.

- To model the expected behavior of the system when considered in each retained operating mode. It is here question to model various operational scenarios for each mode, each operational scenarios showing what are the requested

functions of the system in this mode and how these functions are then dynamically processed.

- To precise then what are the links between modes, configurations and scenarios allowing then to improve the coherence of the entire model of the system.

- To analyse the resulting behavior of the system as proposed in (Chapurlat 2012) by 1) checking modeling expectations in order to detect modeling errors or mistakes (*e.g.* unwanted deadlock or model consistence), and 2) cheking some functional requirements as far as possible detecting then some potential omissions.

Considering modeling needs, designers often use their experience, know-how, sometimes approaches based on creativity *e.g.* brain storming or mental representation. They can also use, when they are formalized, best practices, design patterns (Schindel 2005) or some guide such as GEMMA (French acronym of Guide d'Etude des Modes de Marches et d'Arrêts (ADEPA 1981)). This guide helps manufacturing systems designers to determine the control part. So it is proposed here to develop an approach and to implenment it in a guide called OMAG (Oper- ating Modes Analysis Guide). OMAG promotes a graphical formalism facilitating its use by designers *e.g.* allowing them to select, decompose or refine an operating mode, a configuration or a scenario.

Considering analysis aspect, (Monin 2003) and (Grady 2007) propose using some formal approaches and particularly those focusing on properties proof (Da- guspta 2010) (Yahoda 2012). So a formal operational semantic[1] is proposed for OMAG and a technique based on property formalization and proof (Mallek et al. 2011) is used as described in (Chapurlat 2013a) (Chapurlat 2013b).

Last, this guide has to be fully interoperable with existing modeling languages and tools currently used in system engineering domain. This article presents the OMAG concepts and principles briefly illustrated on the case of a vehicle named VERECINT. Second, it presents the basics of the verification approach before concluding about perspectives and developments.

Modeling aspect

VERECINT is a vehicle allowing firemen and experts in the field of chemical, biological, radiological, and nuclear (CBRN) crisis to explore, to evaluate the different elements characterizing a crisis situation (data *e.g.* temperature or radiation level, information and expertise *e.g.* that follows the observation and the expertise of a phenomenonon on the site), and to communicate them to crisis managers hav- ing to decide actions to take. VERECINT has then to be able to act accordingly to these actions. At this stage, we assume that VERECINT mission, purpose and ob- jectives as environment (other systems in interaction all along its life cycle and enabling systems), requirements, life cycle and all or part of requirements have been defined. These element cannot be detailed in this article.

[1] The set of principles and rules allowing a model (defined here as an instance of a modeling language) execution

OMAG principles and elements

OMAG (Operating Modes Analysis Guide) is a graphical guide proposing a set of pre-defined Operating Modes and of pre-defined Transitions between Operating Modes considered as generally relevant and helpful for various kinds of systems. Applied in VERECINT case, OMAG is diagrammed in Appendix. Let's notice for instance that VERECINT as any a system to be designed must 1) fulfill its mission is some nominal Operating Modes, 2) have to ensure either the continuity of ser- vice of this mission as much as possible in non-nominal Operating Modes, and 3) assume security of goods, people and its immediate environment. Then, before presenting briefly the requested elements of OMAG, let's reformulate its goals as follows:

- To allow a designer to select and choose what are the relevant Operating Modes and Transitions (evolution conditions and events) by taking into account a list of Operating Modes and Transitions having generally to be taken into account. This choice has to be made accordingly to the available knowledge about system definition, the current state of the set of requirements and about system environment.
- To determine gradually what are the possible, unavoidable or interesting configurations of the system and having then to be considered.
- To facilitate the modeling of operational scenarios relevant in each mode *i.e.* to guide the research of the requested functions of the system and the description of how they have to be dynamically associated to describe the expected behavior of the system.
- To allow designers to trace these choices, to change or modify a choice during design activities and to check, evaluate and compare alternatives solutions of functional architectures induced by these choices.

Considering a system *S*, OMAG (see Figure 1) requires to define first a set *A* of data from time, shape or space nature. These ones are chosen and can be evaluated or estimated in order to characterize the temporal aspect (time), the structural as- pect (shape) and the situational aspect (space) of *S* and of its environment. They do not specify any candidate for architectural solution of *S*. They aim only to take into account as far as possible, for instance, dimensional constraints, specific at- tributes, expected delays, speed *i.e.* non functional requirements. *A* is initialized at the beginning of the design process and enriched bit by bit. Second, OMAG high- lights three main Phases named respectively Deployment, Exploitation and End of life. A Phase is a set of Operating Modes of *S* that are logically linked or depend- ent. The Exploitation Phase is it self divided into Operating, Maintenance and De- fault sub phases.

An Operating Mode of *S* is a state reachable by *S* during its life cycle exhibiting then particular behaviors. By hypothesis, *S* is in one and only one Operating Mode called active Operating Mode (conversely, disabled) at each moment of its evolu- tion. Each Operating Mode *O* of *S* is characterized by (E, C, OS_O, T) where:

- E is defined by a name and a set S_E A. It aims to describe the environment of *S*, even partial or simplistic *i.e.* the context in which *S* has to be able to execute various functions when *O* is active. For instance VERECINT may evolve the night, under snow or rain, or on roads that may be damaged due to the crisis.

- The set C contains one or several Configurations relevant for S in O. A configuration c determines and refines the description of the state of S when it is in O. VERECINT can be for example described by configurations named '*exercice*', '*operation on SEVESO site (site containing large quantities of dangerous substances)*' or '*operation on a city*'. A configuration c is defined by a name and a set S_c A defining the requested data from A to describe the configuration. At least one Configuration is required for S, then considered as *default configuration* c_0. Last, S can evolve from a Configuration c to another Configuration c' crossing a Transition as explained below.

<table>
<tr><th colspan="2">Phases</th><th colspan="2">Operational Modes</th></tr>
<tr><td colspan="2" rowspan="3">System deployment</td><td>D1</td><td>System is ready and waiting for deployment</td></tr>
<tr><td>D2</td><td>Operational retirement</td></tr>
<tr><td>D3</td><td>System functions for tests, maintainance, or traning out of operational site</td></tr>
<tr><td rowspan="12">System exploitation</td><td rowspan="5">Operationnal</td><td>O1</td><td>System is deployed and operational on site, waiting</td></tr>
<tr><td>O2</td><td>Preparing the system to assume its mission in nominal mode</td></tr>
<tr><td>O3</td><td>System functions in nominal mode</td></tr>
<tr><td>O4</td><td>Preparing the system to end normally its mission</td></tr>
<tr><td>O5</td><td>System functions for tests, maintainance, or traning on operational site</td></tr>
<tr><td rowspan="3">Dysfunctioning / security</td><td>DS1</td><td>Stop after a default or dysfunction</td></tr>
<tr><td>DS2</td><td>Diagnosis for default detection</td></tr>
<tr><td>DS3</td><td>System functions in non-nominal mode</td></tr>
<tr><td rowspan="4">Maintainance</td><td>M1</td><td>Diagnosis and Corrective Maintenance</td></tr>
<tr><td>M2</td><td>Diagnosis and Preventive Maintenance</td></tr>
<tr><td>M3</td><td>Diagnosis and Adaptative Maintenance</td></tr>
<tr><td>M4</td><td>Diagnosis and Evolutive Maintenance</td></tr>
<tr><td colspan="2" rowspan="2">Cessation and dismantling</td><td>C1</td><td>Retract</td></tr>
<tr><td>C2</td><td>Dismantling</td></tr>
</table>

Figure 1: Phases and Operating Modes of a system in OMAG

- The set OS_O contains one or several Operational Scenarios describing expected behavior of S when S is characterized by the current configuration. An Operational Scenario is defined by a functional model compose of a set of functions dynamically linked *i.e.* a part of S functional architecture describing how S must fulfill its mission. The behavior of VERECINT can be for instance specified by a scenario '*To observe and to measure specific values from crisis area*'.

- The set T contains one to several input and output Transitions. A Transition links a source object here an Operating Mode O (or a configuration C) to a target object *i.e.* an Operating Mode O' (or configuration C'). A input Transition describes how O' (resp. C') can be reached (O' or C' are then activated). Conversely, an output Transition describes how O or C are deactivated. A Transition is then formalized by a 6-uplet (*source*, *destination*, c, e, d, [op]) where *source* and *destination* describes respectively source object and destination object, c is the tiggering condition (computed taking into account data from A), e is the initiating event (internal to S or coming from the environment) and d is the delay (null by default, but allowing to describe for instance duration of a requested reconfiguration time) under which S can evolve from the source to the target (from the same Phase or not). Last, a Operational Scenario op describing the behavior of S when it is reconfigured can be associated to the Transition.

OMAG: modeling principles

Establishing an OMAG model for a system *S* begins by defining a first version of set *A* i.e. by defining the set of space, shape and time data with approximate values. Then designer selects what are the appropriate Phases and Operating Modes. Indeed, those proposed by default (see Figure 1) can be selected by the de- signer or conversely may be rejected considering they are irrelevant regarding the mission, objectives, requirements, and context of *S*. If an Operating Mode *O* is se- lected:

- The set of Transitions allowing to activate *O* (input transitions *T* for which the source Operating Mode has been also selected) and to desactivate O (output transitions *T'* for which the destination Operating Mode has been also selected) are selected and defined by giving the 6-uplet (*source*, *destination*, *c*,*e*,*d*,[*op*]) which characterizes *T*.
- Configurations of *S* reachable in *O* and Operational Scenarios authored by these configurations can be determined. Then, transitions *T'* between Configurations have to be determined by determing the 6-uplet (*source*, *destination*, *c*,*e*,*d*,[*op*]) which characterizes *T'* where the firing event e can be linked to one of the proposed Operational Scenarios that can induce a modification of the current configuration.

Obviously, as OMAG, all the possible Configurations that can appear in the Operating Modes and the set of Transitions between Configurations forms a new state model that allows to decompose S from a different point of view, here by re- fining its possible Configurations whatever may be the Operating Mode on which *S* is.

VERECINT application

Applied to VERECINT system, selected Phases are:

- **Deployment**: the system is subjected to operations to ensure its storage and deployment onto a site on which its mission can begin. This phase is relevant for VERECINT which is involved in activities such as waiting, preparation, adaptation, training, exercise, or regulatory maintenance.
- **Exploitation:** the system is ready and deployed in operational conditions. This phase is relevant for VERECINTand means that it can be used by stakeholders, here firemen and experts having to explore and evaluate a crisis site. In order to avoid any interpretation, this phase is split up into three Families as follows:

Operating (O): the system is in nominal condition being able to fulfil its mission in coherence with specified requirements, hypothesis, and planed resources. In this Family, VERECINT fulfils efficiently its mission and exhibits nominal behaviours and configurations.

Maintainance (M): the system undergoes operations to restore the operating conditions due to anticipation, failure diagnosis, request for modification, adaptation, evolution, etc. In this Family, we choose to consider VERECINT in predictive or currative maintainance that may be done eventually on the operational site where the crisis occurs.

Default (DS): the system is in a safe state or degraded operation following order, failure, damage, or more generally to an internal or external interference that may cause damage. In this Family, VERECINT has to check default and to decide what maintenance is requested considering it must continue as possible to fulfil its mission eventually with loss of performance but a loss of security of users cannot be acceptable.

- **End of life**: the system is removed from service being concerned by decommissioning, partial resuse or reprocessing of part of all of its components and subsystems for possible future use. The feedback, uses and special cases of operational scenarios 'lived' by the system are finalized, indexed and stored to feed the design of possible future releases. This phase is also selected for VERECINT.

The possible Operating Modes of S and those that are retained by a designer for VERECINT are then the followings:

- **D1 - System is ready and waiting for deployment**: VERECINT is available but stopped and possibly stored out of operational site, ready and packed to be deployed on site and then exploitable by stakeholders. By hypothesis, D1is defined as the initial Operating Mode of the studied system (graphically denoted by a box with large borders in Appendix) here VERECINT.
- **D2 - Operational retirement**: VERECINT has to be removed from the site and repackaged or prepared to be redeployed afresh on another site.
- **D3 - System functions for tests, maintenance, or training out of operational site**: VERECINT, although not deployed, is operating, possibly in a degraded or reduced testing environment for functional tests, training, or regulatory service beyond operational site.
- **O1 - S is deployed and operational on site**: VERECINT is operational on site, ready to fulfill its mission in identified operating environments.
- **O2 - Preparing the system to assume its mission in nominal mode**: VERECINT requires preparation before it can fully perform its operational mis- sion on site (*e.g.* preheat, audit checklist usage, etc.).
- **O3 - S functions in nominal mode**: VERECINT fulfill its mission maximizing its performance on specified operating environment.
- **O4 - Preparing S to end normally its mission**: VERECINT requires to be prepared before stopping its mission normally so various Operational Scenarios can be expected in O4 for VERECINT *e.g.* cleaning, decontamining, etc.
- **O5 - S functions for tests, regulatory maintenance, or training on operational site**: VERECINT functions, possibly in a degraded or reduced functional coverage for testing, training, regulatory maintenance on the operational site where VERECINT is currently deployed.
- **DS1 - Stop after a default or a dysfunction**: VERECINT has to be put in safety due to an internal dysfunction or a default detected which threatens its own integrity and safety or the integrity and safety of the environment.
- **DS2 - Diagnosis for default or failure detection**: VERECINT is submitted to tests and procedures (led by itself or by one or more contributors systems) of assessment and diagnosis of failures of its functions and components.
- **DS3 - S functions in non-nominal mode**: VERECINT suffers the consequences of an internal failure or external events affecting its operational capabili- ties but continues to fulfill its mission staying in a range of acceptable values in terms of risk, performance or respect of some chosen non-functional characteris- tics - safety, security, survivability, maintainability, interoperability, ... called "- *ilities*" (Weck *et al.* 2012.).
- **M1 - Diagnosis and corrective maintenance**: VERECINT has to undergo operations permitting to restore a specified configuration so that it is able to ensure its operational mission again.

- **M2 - Diagnosis and Preventive Maintenance**: VERECINT has to undergo operations for the replacement, revision or repair of one or more of its components before the dreaded occurrence of a default respecting a maintenance plan.

- **M3 - Diagnosis and adaptive maintenance**: The system has to undergo operations to adapt, for example due to the possibility of using new technologies to better fulfill its initial operational mission. This includes reengineering activities. This Operating Mode is not relevant for VERECINT so M3 and each input and output Transition of M3 are the graphically identified by a red line has diagrammed in Appendix.

- **M4 - Diagnosis and evolutionary maintenance**: The system has to undergo operations to make it evolve, for example due to the possibility to extent its functional coverage to respond to new requirements or modify its initial operational mission. This includes also reengineering activities. As for M3, this Operating Mode is not relevant for VERECINT.

- **F1 - Retirement**: VERECINT has to be taken out of service permanently.

- **F2 - Dismantling**: VERECINT has to be dismantled and its various components and subsystems may be stored, packaged or stored for reuse, conversion, reprocessing.

A set of generic Transitions between Operating Modes is given in Figure 2. These transitions are quoted as follows: Classic Transitions (T_i, i = 1 to 20), Stop (S_i, i = 1 to 3), Application Maintenance (AM_J, j = 1 to 4) End of Maintenance (EM_j, j = 1 or 2) or Fault Detection Security (FDS). Only input and output Transi- tions of selected Operating Modes have to be specified by the designers. For more information, the reader can find all conditions and events having to be specified for each selectable Transition in OMAG in (Chapurlat *et al.* 2013c).

	D1	D2	D3	O1	O2	O3	O4	O5	DS1	DS2	DS3	M1	M2	M3	M4	C1	C2
D1		S2	T2	T4					S1			AM1	AM2	AM3	AM4	S3	
D2	T1								S1			AM1	AM2	AM3	AM4	S3	
D3	T3	S2							S1							S3	
O1		S2			T7	T5		T11	S1		FDS					S3	
O2		S2				T8			S1		FDS					S3	
O3		S2		T6			T9		S1		FDS					S3	
O4		S2		T10					S1		FDS					S3	
O5		S2		T12					S1		FDS					S3	
DS1		S2								T15						S3	
DS2		S2							S1			T16	T17	T18	T19	S3	
DS3		S2							T14	T13						S3	
M1	EM2	S2		EM1					S1							S3	
M2	EM2	S2		EM1					S1							S3	
M3	EM2	S2		EM1					S1							S3	
M4	EM2	S2		EM1					S1							S3	
C1																	T20
C2																	

Figure 2: Selectable transitions between Operating Modes in OMAG

Verification aspect: OMAG semantic

The operational semantic of OMAG is formalized for two reasons. First, it allows describing without ambiguity how a model OMAG can be interpreted and executed and then to define and implement OMAG simulation mechanisms. Second, it allows to formalize what are then expected modeling properties (Chapurlat 2013b) that must be satisfied in order to help designers to improve the quality of OMAG *e.g.* absence of modeling errors, but not its relevance or adequation with the modeled system. In this case, OMAG transformation rules are proposed in or- der to transform an OMAG model into a formalism authorizing proof of a- temporal and temporal properties as proposed for instance in (Mallek *et al.* 2012). In this case, techniques are based on the use of Conceptual Graphs for a-temporal properties and on Model Checking techniques for temporal properties. The reader interested by these two complentary techniques can find definitions and illsutra- tions in the referenced articles.

By definition, OMAG is conform to the Interpreted Sequential Machine (ISM) described in (Larnac *et al.* 1999) and is then an extension of a State Machine. Operating Modes and Configurations are formalized by *states* and Transitions are described as conditioned *transitions* between *states*. By hypothesis, an Operational Mode or a Configuration can be decomposed giving then a new ISM. The ISM operational semantic is then enriched by model decomposition rules as proposed by (Harel 1987) for Statecharts. This semantic can be summarized in the next. There always exists an initial Operating Mode *i.e.* initial state for each level of decomposition and the next hypothesis of behavioral determinism are required:

- For each moment in the evolution of the OMAG, the same input vector applied to the same active state *S* induce always the same resulting output vector and the same next reached state *S'*.
- A transition *T* is triggered at a null time *i.e.* there is no potential event *e* (internal as well as external) that can be omitted during triggering of T.
- For a given state *S*, the evolution condition associated to output transitions of *S* are exclusive *i.e.* only one transition *T* can be triggered at each moment.

Thus, the transition triggering is done in two stages. If the condition *c* is true and the trigger event *e* appears (always occurring by default if it is not specified), then the Operating Mode or the Configuration is deactivated. This induces eventually to be able to stop current Operational Scenarios that are associated with this one. It can also require the execution of the so called *reconfiguration operational scenario op* which describes what the required functions are allowing *S* going from the current Operating Mode or the current Configuration to the next one. After the delay *d* (equal to null by default), the targeted Operating Mode or Configuration is activated. In the case of an Operating Mode, a default configuration is de- fined and then, is activated. In the case of the activation of a Configuration, authorized Operational Scenarios are launched and executed. Conversely, any Operational Scenario may induce a modification of Configuration, possibly causing the trigger a new transition between this configuration and the next one.

Conclusion

An abstract and a concrete syntax are under development by using DIAGRAPH tool box (Pfister *et al.* 2012) under Eclipse modeling framework. It provides a graphical user interface allowing to handle operating modes, configurations and operational scenarios. By hypothesis, eFFBD (enhanced Functional Flows Block Diagram) (DoD 2001) modeling language is here adopted to describe operational scenarios. So, a first perspective consists to use operational semantic of eFFBD proposed by (Seidner 2006) for synchronizing OMAG and Operational Scenarios evolution. Last, the modeling tool will aim to be interoperable with various SE tools. For this, transformation rules and mechanisms are studied by using ATL (ATL 2006).

Verification techniques have now to be adapted and tested on complex examples. The perspective is to enrich these two techniques (Conceptual Graphs and model checking) by considering extensions proposed by (Thierry-Mieg *et al.* 2004) allowing to gain in performance and relevance when facing problematic of growing up models' size and complexity (*e.g.* due to number of refinements lev- els) to be analyzed during verification activities.

References

ISO/IEC 15288:2008(E) / IEEE Standards 15288.2008 – Systems engineering – System life cycle processes (2nd edition), February 2008

INCOSE, System Engineering (SE) Handbook Working Group, System Engineering Handbook, A Guide For System Life Cycle Processes And Activities Version 3.2.1, INCOSE TP 2003 002 03.2., 2011

BKCASE Project, System Engineering Book of Knowledge, SEBoK v1.0, http://www.sebokwiki.org/ (last visit 2013-04), 2012

S.Fiorèse, J.P.Meinadier, Découvrir et Comprendre l'Ingénierie Système, CEPADUES Editions, ISBN 978.2.36493.005.6, May 2012

C.Fèliot, Toward a formal theory of systems, Colloque d'Automne du LIX 2007 - CAL07 Complex Systems: Modelling, Verification and Optimization, Paris, Carré des Sciences, http://www.lix.polytechnique.fr/~liberti/cal07/presentations/ (last visit 2012-01)

INCOSE, Survey of Model-Based Systems Engineering (MBSE) Methodologies, Model Based Systems Engineering (MBSE) Initiative from International Council on Systems Engineering (INCOSE), 10 June 2008

B.S.Blanchard, W.J.Fabricky, Systems Engineering and analysis, Prentice Hall International Series in industrial and systems engineering, fifth Edition, Pearson Coll., 2011

D.Sharman, A.Yassine, Characterizing Complex Products Architectures, Systems Engineering, 7(1), pp. 35-60, 2004

B. Schindel, Pattern-Based Systems Engineering: An Extension of Model-Based SE, INCOSE 2005, TIES 4

ADEPA, Guide d'Etude des Modes de Marches et d'Arrêts, http://www.technologuepro.com/cours-automate-programmable-industriel/GEMMA/Gemma-original.pdf (last visit 2012-12), 1981 [*in French*]

J-F. Monin, Understanding Formal Methods, Springer-Verlag, ISBN: 1-85233-247-6, 2003

J.O.Grady, System Verification: proving the design solutions satisfies the requirements, Academic Press, Elsevier editor, ISBN: 978-0-12-374014-4, 2007

P.Dasgupta, A roadmap for formal property verification, Springer, ISBN: 978-90-481-7185-9, 2010

Yahoda, formal verification tools overview web site, http://http://anna.fi.muni.cz/yahoda/ (last visit 2012-01)

S.Mallek, N.Daclin N., V.Chapurlat, An Approach for Interoperability Requirements Specification and Verification. The International IFIP Working Confer- ence on Enterprise Interoperability, Stockholm, Sweden, 2011

V.Chapurlat, (2013a) UPSL-SE: A Model Verification Framework for Systems Engineering, Computers in Industry, Computers in Industry 64 (2013), pp. 581–597

V.Chapurlat (2013b), Property concept and modeling languages for Model-Based Systems Engineering (MBSE) context, Internal Research Report (access on demand),

M. Larnac, J. Magnier, V. Chapurlat, B. Chenot, Extended Temporal Proof of Properties using the Interpreted Sequential Machine Model, proc. of International Congress IEEE SMC'99, Tokyo, Japan, 1999, pp I-974/I-979

V.Chapurlat (2013c), Proposition d'un Guide d'Etude des Modes Opérationnels des Systèmes (GEMOS) pour l'Ingénierie Système, Conférence Francophone de Génie Industriel, La Rochelle, June 12st -14st, 2013 [*in French*]

D.Harel, Statecharts: a visual formalism for complex systems, Science of comput- er programming, 8, 1987, pp 231-274

C.Seidner, Vérification des EFFBDs: Model checking en Ingénierie Système, Thèse de Doctorat Université de Nantes, 3 Novembre 2009 [*in French*]

O.L. de Weck, A.M.Ross, D.H. Rhodes, Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (-ilities), third International Engineering Systems Symposium CESUN 2012, Delft University of Technology, 18-20 June 2012

F.Pfister, V.Chapurlat, M Huchard, C.Nebut, A proposed tool and process to design domain specific modeling languages (pp. 31-35), http://www.lgi2p.ema.fr/~urtado/Slides/RR12_Lab_002.pdf (last visit 2012-09)

DoD, Systems Engineering Fundamentals. Defense Acquisition University Press, http://www.dau.mil/pubscats/PubsCats/SEFGuide 2001-01.pdf (last visit 07/07/2012), 2001

ATL: Atlas Transformation Language - ATL User Manual, version 0.7, ATLAS group LINA & INRIA, Nantes, 2006

S.Mallek, N.Daclin, V.Chapurlat, The application of interoperability requirement specification and verification to collaborative processes in industry Computers in Industry, n°63 (2012) 643–658, 2012

Y. Thierry-Mieg, S. Barrir, A. Duret-Lutz, F. Kordon, Nouvelles techniques de Model Checking pour la vérification de systèmes complexes, Génie Logiciel, 69, pages 17-23, 2004 [*in French*]

Appendix: OMAG graphical representation for VERECINTexample