Modeling Radio-Frequency Front-Ends Using SysML: A Case Study of a UMTS Transceiver

Sabeur Lafi, Roger Champagne, Ammar B. Kouki, and Jean Belzile

École de technologie supérieure, 1100 rue Notre-Dame Ouest, Montréal, H3C 1K3, Canada.

Abstract. Numerous engineering fields are nowadays dealing with complex systems. The analysis, design, testing and maintenance of such systems are crucial challenges. For this purpose, the OMG proposed SysML, an extension of UML, in order to address the issues of modeling complex systems in different engineering domains. This standard enables the elaboration of efficient tools allowing automated analysis, verification and validation of systems. The radio-frequency front-end's design is one of engineering fields which would benefit from such a technology to enhance the efficiency of the design and manufacturing process. In this paper, we discuss the provision and the limitations of both UML and SysML. We also present a case study consisting of the modeling of a Universal Mobile Telecommunications System (UMTS) transceiver using SysML and we discuss the advantages and the drawbacks of such a technology from the designer's point of view.

Keywords: SysML, UML, Modeling, Systems Engineering, UMTS, Transceiver.

1 Introduction

Engineering systems are increasingly growing in complexity, implying various design and testing challenges. Consequently, multiple fields of engineering are looking for a general-purpose and high-level methodology for systems' modeling. This can effectively enable an efficient design process from specifications all the way through to delivery and maintenance. One of these fields is radio-frequency (RF) and microwave engineering which mainly addresses the design and manufacturing of microwave radios and components. In fact, RF front-ends represent an important part in several embedded devices such as wireless sensors and smart radios.

Modern RF front-ends need to be modeled in parallel with the baseband hardware and software parts which carry out signal processing and support, in some cases, user applications. Software modeling can currently be achieved using Unified Modeling Language (UML) [1]. UML was originally defined by the Object Management Group (OMG) in order to enable the definition and modeling of complex software systems and was later used in other fields. On the other hand, complex engineering systems including software, electrical, hydraulic and mechanical hardware, can be modeled by the recent Systems Modeling Language (SysML) [2].

Because SysML is a recent standard addressing modeling in a wide range of domains, each engineering field must first evaluate its abilities to express and describe its specific particularities. Some studies had been already carried out in some fields such as sensor networks [3] and System-on-Chip/Network-on-Chip [4]. As far as RF design is concerned, case studies must be performed to test the usefulness of SysML for modeling RF systems. In this paper, we first focus on the modeling languages and we compare the use of OMG's UML and SysML in the modeling of software/hardware systems. We specifically discuss how RF front-ends can be modeled using SysML. Second, we present a case study in which a UMTS transceiver is modeled using SysML. Finally, we discuss the benefits and the limitations of using SysML in RF systems' design.

2 Modeling Languages in Modern Systems Engineering

The growing complexity of software systems led various organizations and research task groups, both in industry and academia, to investigate modeling techniques. One of these organizations, the OMG, has gathered several proposals with the intent of elaborating a standardized modeling language [5], [6], [7]. The outcome of this effort was the establishment of the Unified Modeling Language (UML). Therefore, UML, a visual specification language for object modeling, has emerged as a viable modeling language empowering software design, bringing high-level of abstraction and enabling different automated techniques such as code generation, verification and validation of software and allowed also data interchange and meta-modeling.

UML has proven its importance particularly in the field of software design. Its extensibility enhanced its scope to other domains. This advantage gives more appeal to UML which can then be used to model systems other than software. However, the provision of UML to systems engineering is limited. In fact, it is unable to express specific aspects of several domains. For example, it does not support efficiently the modeling of dynamically changing parameters causing the system to behave differently under different configurations [3]. It also expresses weakly the relationships between mixed systems composed of both hardware and software [3]. An interesting survey about the common limitations and defects of UML is given in [8]. In order to address these limitations and others, the OMG has specified and standardized another modeling language, SysML, which is intended to provide visual modeling support for a wide range of engineered systems. SysML is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include different types of components such as software, hardware, etc. [2]. It provides graphical representations with semantic foundations in order to model different aspects of a complex system such as its structure, behavior, requirements, and parametrics.

Despite the fact that SysML is a subset of UML, it differs remarkably from it. First, UML is a general-purpose modeling language while SysML, as a customized profile of UML 2.0 conceived for systems engineering, is domain-specific. Thus, SysML is smaller and easier to learn than UML since it removes many software-centric constructs. It expresses systems engineering semantics better than UML. Second, it is a precise language, including support for constraints and parametric analysis which allows models to be analyzed and simulated, greatly improving the value of system models compared to textual system descriptions. SysML also supports various diagrams that facilitate automated analyses, verification and validation. In addition, it is an open standard which is compliant with various data interchange formats such as XML, XMI (XML Metadata Interchange) and AP233 standards. Furthermore, SysML improves communication by using a formal language, namely Object Constraint Language (OCL), for sharing system information to all project engineers [9], [10].

RF front-ends, as a complex engineering domain, require high-level modeling methodologies providing enough flexibility and abstraction in order to analyze, design, and validate RF systems. In practice, modeling would help the automation of several design aspects. Some of the most important are (i) verification, as the process of determining that a model implementation accurately represents the designer's conceptual description of the model and (ii) validation, as the process of determining the degree to which a model is an accurate representation of the real RF system from a functional perspective. In this context, we try to explore the provision and the limitations of SysML in this regard. For this purpose, we present in the next sections the results of a case study consisting of the modeling of a UMTS transceiver using SysML.

3 Case Study: Modeling a UMTS Transceiver Using SysML

To evaluate the benefits and the limitations of SysML in modeling RF and microwave front-ends, which represent an important interface between embedded systems and the real world, we propose to apply it to the modeling of a UMTS transceiver. This choice is motivated by the fact that a UMTS mobile phone is composed of three main parts: (i) software, carrying out the main signal processing, internetworking and user applications, (ii) digital hardware, such as digital signal processors to execute the signal processing algorithms and baseband operations and (iii) analog hardware, carrying out the transmission/reception of radio signals to/from the base station, known as node B in the UMTS terminology. If software can be modeled using UML and digital hardware using hardware description languages, analog hardware is still developed using classical techniques. As a result, RF design lacks flexibility and is still implemented manually. The prospect of modeling it using SysML is to achieve a high-level of abstraction and automation, particularly for such RF design tasks as verification and validation. In this section, we present a summary of the UMTS transceiver specifications and describe how we capture and model them using SysML. Due to space constraints, only a subset of the SysML diagrams we experimented with for this system is presented.

3.1 UMTS Transceiver Specifications

Universal Mobile Telecommunication System (UMTS) is a mobile standard conceived for third-generation mobile communications networks. This communication standard had specified different radio interfaces. In this section, we present a summary of the specification for a UMTS Terrestrial Radio

Access/Frequency Division Duplex (UTRA/FDD) compliant mobile transceiver. This summary is based on [11], [12] and [13].

An UTRA/FDD transceiver is a radio whose RF front-end is composed of three parts: (i) duplex filter, (ii) transmitter and (iii) receiver. All three are linked to the antenna as shown in Fig. 1.

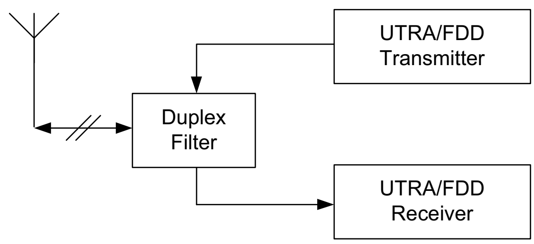

Fig. 1. A synoptic schematic of an UTRA/FDD Transceiver.

There are various architectures for implementing the transmitter and receiver blocks. Among these, direct up-/down-conversion architectures are being widely used in mobile communications, particularly in GSM and W-CDMA applications [11]. In fact, a direct up-/down-conversion transceiver is less complicated than classical architectures such as a superheterodyne radio. Fig. 2 shows a typical architecture of a direct conversion UTRA/FDD transceiver.

In the transmission phase, the baseband signal is modulated and data symbols are typically converted into two analog signals (in-phase, *I*, and in quadrature, *Q*) via digital-to-analog converters. The *I*/*Q* signals are then up-converted in the quadrature up-conversion stage (QUC) to the RF frequency determined by the local oscillator (LO) and then summed. The resulting signal is then amplified, by the variable-gain amplifier (VGA), in order to adjust its power level to the desired value, and filtered by the band-pass filter (BPF-T), in order to reduce intermodulation products. The final stage of the transmitter is the power amplifier (PA) which amplifies the transmitted signal to the high power level required for transmission. In the reception phase, the received signal is low power and its level is close to the noise floor. Therefore, it is first amplified by the low-noise amplifier (LNA1), which keeps the added noise to a minimum, and pass-band filtered by BPF-R in order to eliminate interferences. A second stage amplifier, LNA2, boosts the received signal further before it is divided into two signals which are mixed with the LO reference signal. One signal is mixed with the inphase LO while the other with the quadrature (90-degree phase-shifted) LO. The resulting *I*/*Q* signals are then amplified and filtered, to remove intermodulation products and non desirable signals, in the BFA1 and BFA2 blocks before they are digitized for baseband demodulation.

In a UTRA/FDD transceiver, both the receiver and the transmitter operate simultaneously. Any leakage between the transmitter and the receiver can either saturate the low-noise amplifier or disturb the transmitted signal. For this purpose, a duplex filter is added in order to isolate the received and the transmitted signals. The UTRA/FDD standard determines rigorously the specifications of the duplex filter and each component in the receiver/transmitter chains. For example, Table 1 presents the specifications of the duplex filter [11] while Table 2 presents some key specifications of a UTRA/FDD radio as stated in [13]. The duplex filter plays the role of a duplexer and a filter. It isolates the incoming and outgoing signals and also allows the rejection of out-of-band interferences. This duality in role implies severe constraints in terms of isolation and in-band attenuation.

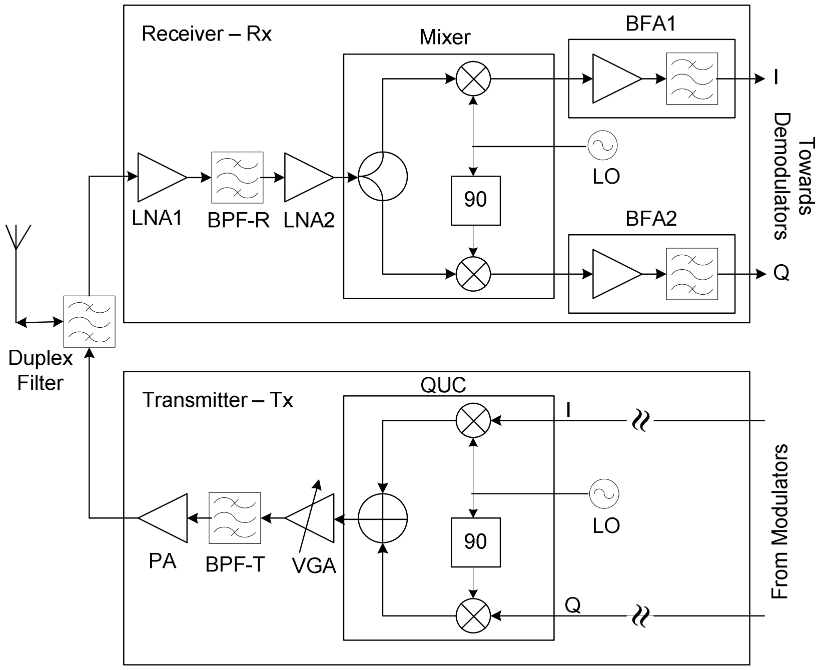

Fig. 2. A detailed view of a direct-conversion UTRA/FDD transceiver.

Table 1. Duplex filter requirements in terms of attenuation and isolation.

Tx – Antenna Attenuation (dB)	< 2
Rx – Antenna Attenuation (dB)	< 3
Tx – Rx Isolation / Tx-Band (dB)	> 50
Tx – Rx Isolation / Rx-Band (dB)	> 37

Table 2. Some of UMTS standard specifications.

Frequency Band (Up-link) (MHz)	1920 – 1980
Frequency Band (Down-link) (MHz)	2110 – 2170
Frequency Spacing (MHz)	4.4 – 5.2
Modulation	QPSK
Pulse Shaping	RRC / roll-off = 0.22
Chip Rate (Mc/s)	3.84
User Bit Rate (kbps) @ BER=10^{-3}	12.2
Power Control Frequency (Hz)	1500

Signal-to-Noise Ratio (dB)	6.0
Power Sensitivity (dBm)	-120.0
Input Noise Level (dBm)	-111.0
Reference LO Power (dBm)	6.0

The UMTS standard specifies the baseband characteristics such as the type of modulation, shaping filter and the chip rate. It also establishes the front-end parameters such as the input noise level, the signal-to-noise ratio, the power sensitivity, etc. These specifications are generally produced following extensive system level analysis. In the next section, we present how to capture the UTRA/FDD specifications using a SysML model.

3.2 UTRA/FDD Transceiver's SysML Model

We presented some of the specifications of a UTRA/FDD transceiver in the previous section. We chose the direct conversion architecture to implement the radio. For the detailed specifications, the reader can refer to [11], [12] and [13]. In this section, we present how to capture these specifications in a SysML model. In this model, we present the structure of the overall transceiver and we detail the internal blocks of the receiver. We also show how to capture some of the transceiver requirements.

The structure of a RF front-end incorporates the different components of which it is made. The SysML model can capture this structure using different diagrams at different levels of abstraction. Among these diagrams, we use the *package diagram* in order to give an overview of the general structure of the model packages, see Fig. 3. As shown in this Figure, the transceiver's SysML model is organized into four main packages: (i) Value Types, describing the measurement units used in the other packages (ii) Transceiver Structure, describing the structural components of the transceiver (iii) Transceiver Behavior, describing the signal flow inside the transceiver (iv) Transceiver Requirements, illustrating the requirements of the transceiver.

As in many disciplines, different measurement units are used in the radio-frequency domain. SysML allows their modeling using "*value types*". In the value types package, shown in detail in Fig. 4, we present the measurement units used in the specifications of a UTRA/FDD transceiver. All other packages using these measurement units have a "*dependency*" relationship with this package (see Fig. 3).

The structure package is composed of diagrams such as the *block definition diagram* and the *internal block diagram*. The former illustrates the structure of an object with blocks presenting its different components while the latter gives an insight of how a block is structured. The general structure of the UTRA/FDD transceiver can be modeled using a standard block definition diagram. This diagram captures the different components of the transceiver and organizes them into different levels of hierarchy. Fig. 5 shows the block definition diagram of the entire transceiver which is made up of a back-end consisting in an antenna, and a RF front-end which includes the duplex filter, the transmitter and the receiver. The duplex filter is an atomic component. However, the receiver and the transmitter are composed of several other components such as the local oscillator, the mixer, etc. To illustrate the structure of the transceiver for example, we can use the internal block diagram as shown in Fig. 6.

The components of the blocks are called "*parts*". One of the advantages of this representation is its ability to capture how they are connected and which types of information or signals can be exchanged between them.

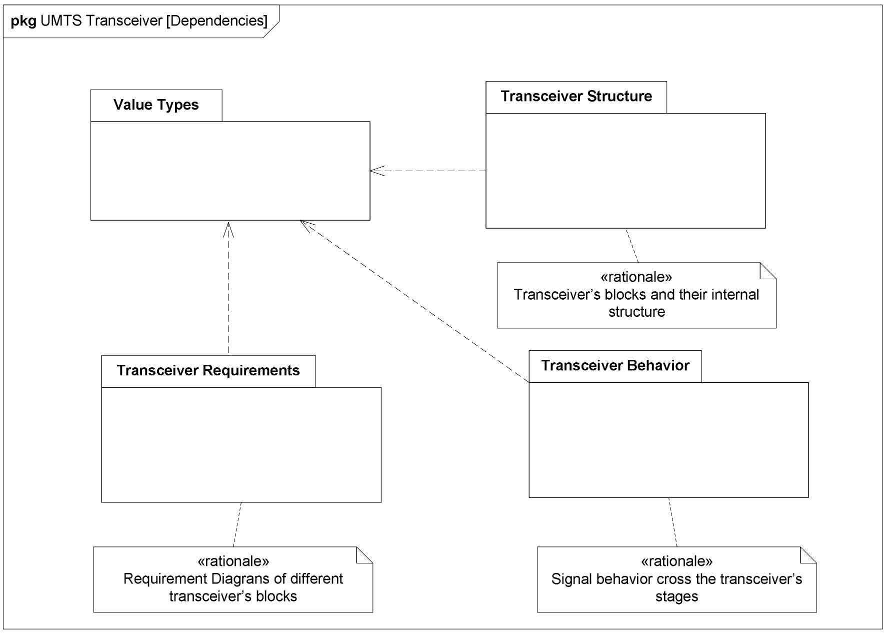

Fig. 3. Package diagram of the UTRA/FDD transceiver's SysML model.

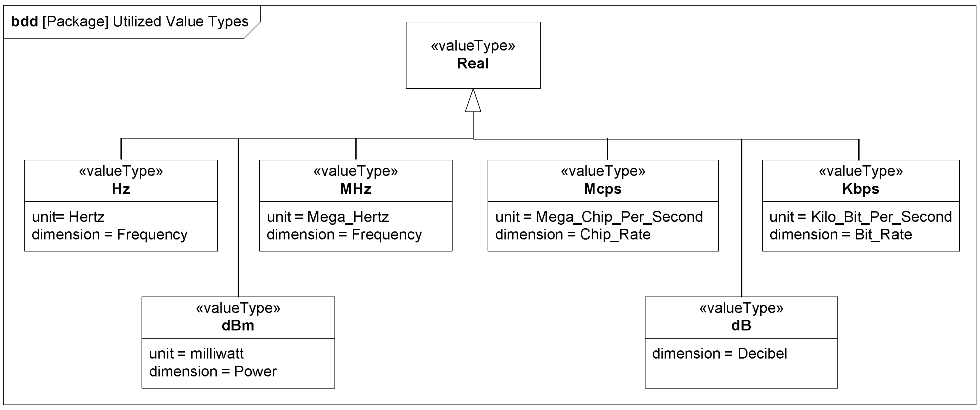

Fig. 4. Value types package.

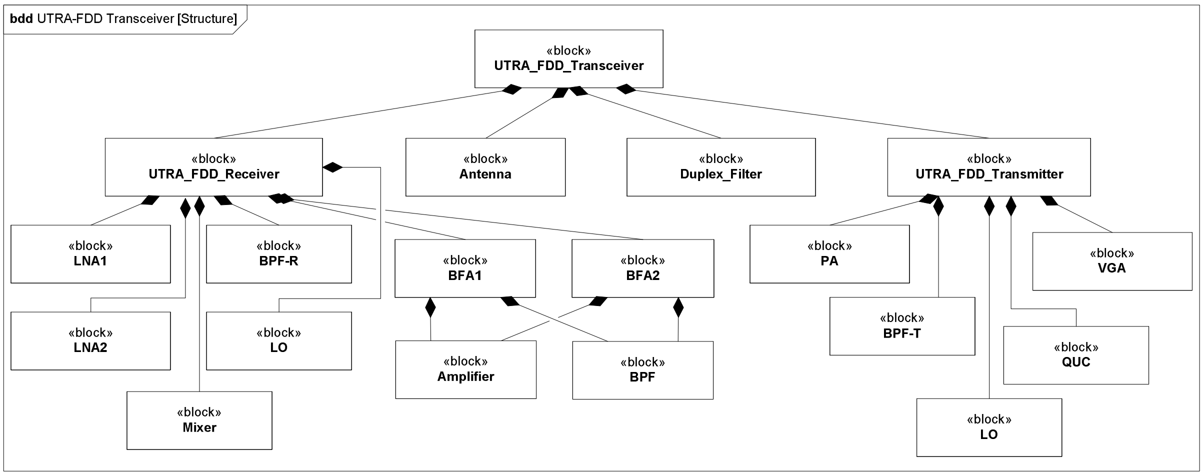

Fig. 5. The block definition diagram of the UTRA/FDD transceiver showing the hierarchy of its components.

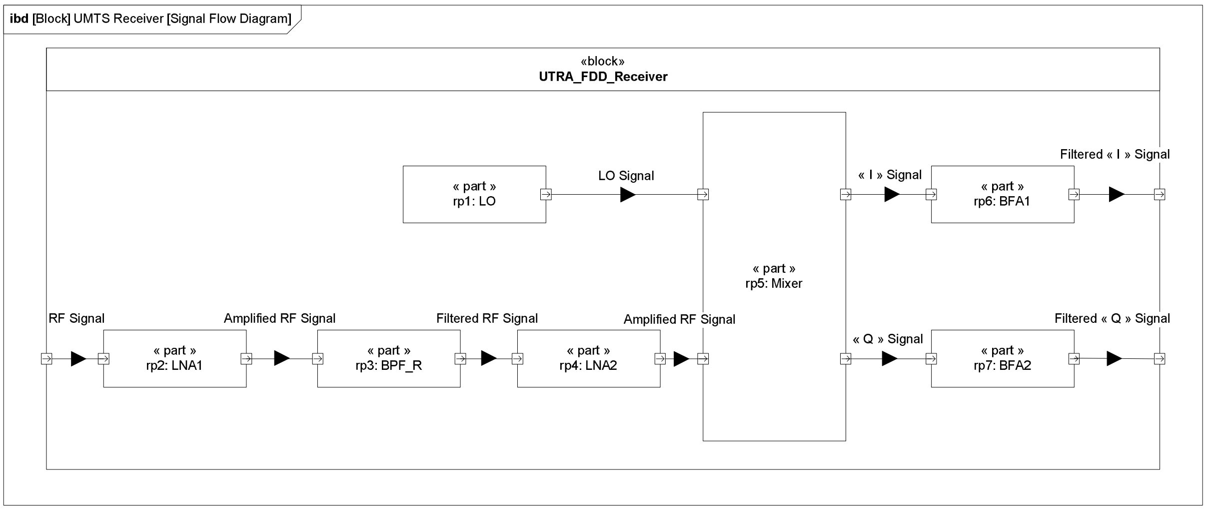

Fig. 6. The internal block diagram of the receiver presenting how its different parts are linked as well as the signal flow between them.

Different analog signals of different origins travel between these components. For example, two types of signals are required inside the mixer block. Both the baseband and the LO reference signals are needed in order to achieve the down-conversion operation. These signals are communicated to the mixer via its input ports. The flow of signals can also be captured by the internal block diagram as shown in Fig. 7. The RF signal is communicated to the mixer by its RF input port and then divided in two signals: one is in-phase and another is 90-degree phase-shifted. Both of them are down-converted according to the LO reference frequency received from the LO output port. The result is the *I* and *Q* signals, each carrying a part of the information.

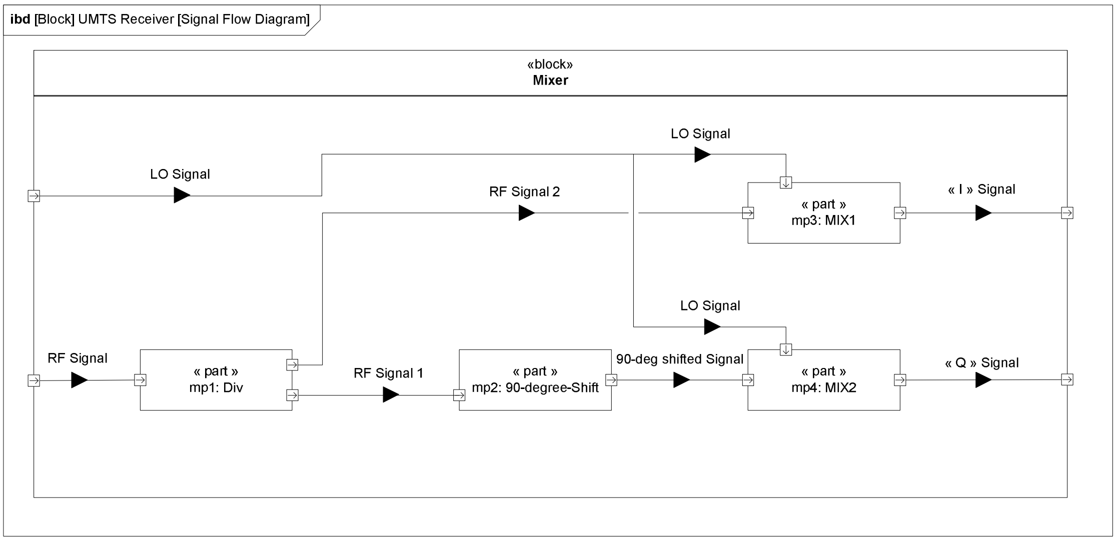

Fig. 7. The internal block diagram of the receiver's mixer and the signal flow inside it.

One of SysML novelties is the *requirement diagram* whose role is the capture of specifications and requirements of an engineering system in a simple and standard manner. In RF engineering, the requirement diagram is an important tool that can help the designer to represent and communicate the specifications to the other designers of the team. For example, the information of Table 1 can be represented in a requirement diagram as shown in Fig. 8. Requirements can be organized in a hierarchal fashion. They can be copied, derived or traced. Test cases can be added in order to verify the system at the end of the design cycle. In the example of Fig. 8, the duplex filter has two main properties: attenuation and isolation. Its final design must satisfy the requirements expressed in Table 1.

Requirements can be organized in a nested structure, as shown in Fig. 9. This allows more clarity in the representation. They can also be grouped into constraints, test cases, etc. For example, in Fig. 9 the user bit rate is considered 12.2 kbps only if the corresponding bit error rate (BER) is equal to 10^{-3}.

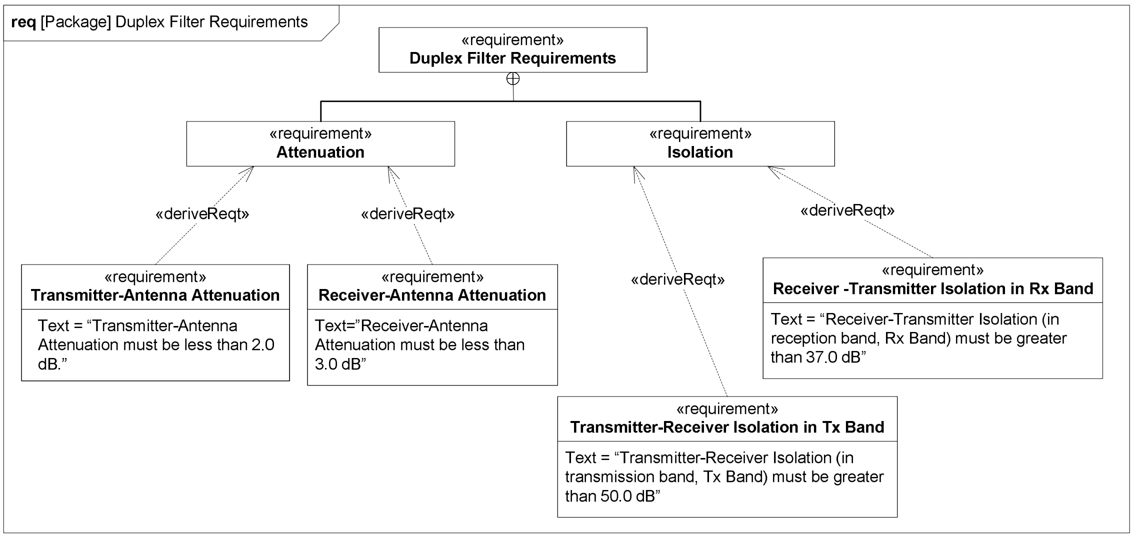

Fig. 8. The requirement diagram of the duplex filter.

4 Discussion

In the previous section, we presented a model of a UTRA/FDD transceiver. We have shown particularly how the different aspects of a RF system's specifications can be captured using a SysML model. In this section, we discuss the consistency and coherency of such a methodology for modeling RF front-ends. We specifically focus on the provision and the limitations observed in this case study.

SysML is a language, inspired from UML, aiming to offer a powerful standard supporting rigorous modeling of various systems. It addresses this issue in a wide range of engineering domains. This mission is not easy because each engineering field has its own particularities. Despite the fact that SysML is defined as a new language, it retains and extends many concepts of UML. A legitimate question is: is this property a provision or a limitation?

In the radio-frequency domain, semantics are very important. The notations and the representations are needed by RF designers and engineers for easily expressing, understanding and sharing designs. For example, the symbols used to represent RF components such as mixers and LOs are fixed by a consensus. This facilitates the understanding and the interpretation of RF schematics. However, their SysML representation, being currently limited to a restrictive set of notations, lacks the flexibility of customized symbolic representation. Consequently, RF engineers will find it difficult to read and interpret SysML diagrams and impractical to work with its notation. Adding customizable symbols to the SysML standard representation of blocks and parts will go a long way towards making SysML more easily accepted by

RF engineers. To illustrate this, we reproduce the block definition diagram of Fig. 5 with added RF symbols as shown in Fig. 10. This enhances considerably the readability of this SysML representation of the transceiver. A possible solution to adopt in this regard is the creation of a SysML profile for modeling RF and analog components. Such a profile would define the stereotypes and constraints which enable the specifications, analysis and verification of RF systems.

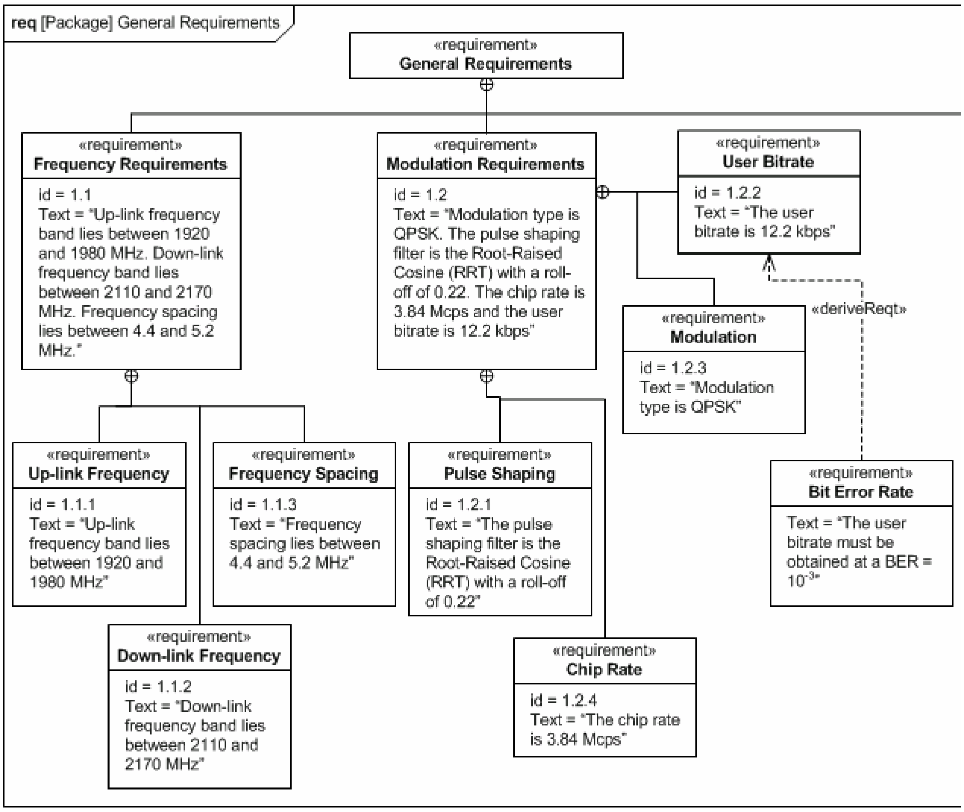

Fig. 9. A portion of the requirement diagram expressing the specifications of Table 2.

On the modeling level, some important questions remain open, namely, what is the right depth of a SysML model? In other words, is there any definition of the granularity concept? At this stage, it is difficult to formulate a definite answer since (i) SysML, like UML, is a notation and not a methodology, (ii) a SysML model is not unique and (iii) many experiments and case studies have to be carried out in order to learn how the depth of a model relates to the hierarchical levels of the modeled system. Consequently, a SysML model depends on the level of experience of the system modeler and may need several iterations before reaching acceptable results.

In the light of the above case study, one can argue that the provision of SysML to the design of RF systems lies in three main levels: (i) abstraction, (ii) flexibility and (iii) requirements. First, on the abstraction level, SysML can represent the structure of a RF system in different ways. Aspects such as hierarchy, containment, and multiplicity can be expressed rigorously. This allows the masking of some levels of the SysML model. Such a mechanism can be very useful in RF systems. In fact, one of the issues in RF systems is the absence of an abstraction mechanism which controls the level of details of the system meaning that the designer can choose the level of abstraction and the granularity at which he wants to carry out the analysis of the system. Such a mechanism can really empower a hardware abstraction strategy allowing automatic design and synthesis of RF components and systems. Second, our experience has shown that SysML formalism and notations are flexible enough to express most of a RF system's aspects. For example, the port is considered as the lowest level of abstraction in a RF system. SysML allows describing the properties of RF ports, the flows travelling between them and the connectors relating them. Third, SysML can capture and express requirements in an organized and simple way. For several years, designers have been experiencing difficulties communicating the specifications among themselves. Ambiguities and forgotten details usually lead to serious negative effects on the design, test, integration and validation times. SysML presents an important evolution from traditional requirements management tools to UML/SysML models which offer a rich language for expressing the context, behavior and constraints of an engineering system. Therefore, the requirement diagram of SysML can be a useful tool to help RF designers and engineers to organize their specifications in a rigorous way.

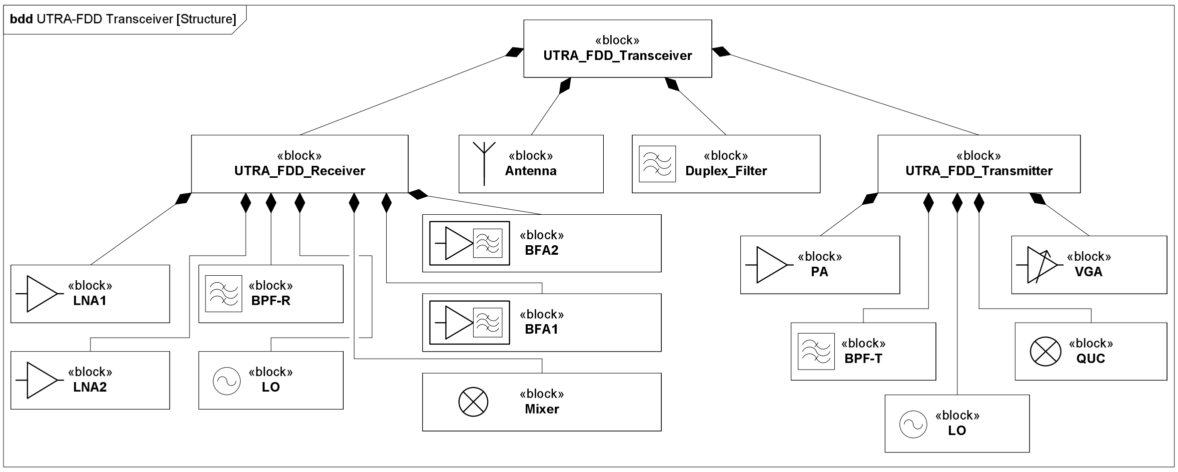

Fig. 10. An example of block definition diagram incorporating notations belonging to RF engineering domain.

Additionally, on the verification and validation level, though not illustrated in our case study, SysML allows the representation of flows, the choice of their type and the corresponding ports such that the designer can model the signal flow of the RF system and carry out an automatic check of the model leading to automated verification and validation (AV&V) of RF systems, which would be of great use in RF engineering.

Another equally useful SysML concept for RF design is the parametric diagram since RF design relies heavily on mathematical models with multiple parameters. We believe that the parametric diagram can help in building models for customized RF components and systems.

Finally, one can observe that SysML is a new language that surely needs more refinement and revision. This said, one cannot deny the importance and the consistency of the modeling concepts it presents. SysML can help RF designers to automate at least some design tasks. However, only few tools currently truly support SysML. Furthermore, the majority of them are either not sufficiently mature or were originally designed to support UML. This situation hinders significantly the widespread adoption of SysML.

5 Conclusion

SysML is a modeling language recently standardized by the OMG. It was introduced in order to address modeling issues in systems engineering. In this paper, we investigated the possibility to use this language to model RF front-ends. We first discussed the scope of UML and the emerging SysML. Then, we studied how a typical RF front-end such a UTRA/FDD transceiver can be modeled using the latter. We finally discussed the provision and the limitations of SysML to RF systems design.

This work allows us to conclude that SysML is useful in RF front-ends' design. In fact, modeling RF systems using tools that implement this language can provide significant flexibility to designers because it allows the abstraction of certain RF subsystems. Such tools can also help automating some design tasks, especially the coherence verification of the model and even the validation of its resulting implementations. This modeling approach can also enhance productivity because it captures the requirements and the constraints imposed to the system. However, great efforts must be deployed in order to enhance the SysML-supporting tools and ensure their widespread acceptance in various engineering fields.

References

1. Object Management Group: Unified Modeling Language Superstructure. Specification V 2.1.2, (2007).
2. Object Management Group: Systems Modeling Language (SysML). Specification V 1.0 (2007)
3. Belloir, N., *et al.*: Utilisation de SysML pour la modélisation des réseaux de capteurs: 14ème Colloque International sur les Langages et Modèles à Objets. Montréal, Canada (2008)
4. Vanderperren, Y., Dehaene, W.: UML 2 and SysML: An Approach to Deal with Complexity in SoC/NoC: Design, Design, Automation and Test in Europe. pp. 716 – 717 (2005)
5. Booch, G., Rambaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. pp. 10 – 12. Addison Wesley (1998)
6. Weilkiens, T.: Systems Engineering with SysML/UML: Modeling, Analysis, Design. Morgan Kauffmann OMG Press (2008)

7. Medvidovic, N., *et al.*: Modeling Software Architectures in the Unified Modeling Language. ACM Transactions on Software Engineering and Methodology, vol. 11, no. 1, pp. 2 – 57 (2002)
8. Lange, C. F. J.: Chaudron, M. R. V., Muskens, J.: In Practice: UML Software Architecture and Design Description, IEEE Software Journal, pp. 40 – 46 (2006)
9. What are the benefits of using SysML?, www.embeddedplus.com/SysML.php
10. What is the relationship between SysML and UML?, www.sysmlforum.com/FAQ.htm
11. Madsen, P., *et al.*: UTRA/FDD RF Transceiver Requirements, Wireless Personal Communications Magazine, pp. 55 – 66 (2002)
12. Third Generation Partnership Project (3GPP): UE Radio Transmission and Reception (FDD), Technical specification 25.101 v.4.0.0 (2001)
13. W-CDMA/UMTS, FDD Technical Summary, www.umtsworld.com/technology/wcdma.htm