Definition and Reuse of Analysis Patterns for Real-Time Applications

Hela Marouane, Saoussen Rekhis,
Rafik Bouaziz
Sfax University, BP 1088, 3018, Sfax, Tunisia
hela_marouane@yahoo.fr
{saoussen.rekhis, Raf.bouaziz}@fsegs.rnu.tn

Claude Duvallet, Bruno Sadeg
LITIS, UFR des Sciences et Techniques, BP 540,
76 058, Le Havre Cedex, France
{claude.duvallet, bruno.sadeg}@univ-lehavre.fr

***Abstract*— The analysis patterns improve the quality of products and the performance of development process. They have proven to be an effective means for capturing expert's knowledge and reducing the costs and the time of development. In this paper, we are interested in defining analysis patterns to model both the functional and non-functional requirements of Real-Time (RT) applications. The motivation behind the definition of these patterns is to facilitate the modeling of RT applications that must meet not only the accuracy of results, but also the time constraints related to the validity of data and the deadline of transactions. The proposed RT analysis patterns are illustrated through the modeling of two RT applications examples: the road traffic control and the medical control applications. These patterns are supported by a CASE toolset that both helps in RT analysis patterns representation and guides the patterns reuse.**

***Keywords*— Real-Time applications; analysis patterns; functional and non functional requirements.**

I. INTRODUCTION

Nowadays, Real-Time (RT) systems cover many sectors of activity: control of production lines, control of patients at home and medical assistance operations, control of road traffic, and so on. Generally, RT applications have common functionalities. Firstly, they acquire data from the environment by sensors. Then, they analyze the acquired data and provide results within the time constraints. Finally, they send orders to the environment via actuators. The design of these applications can be facilitated using reusable components that improve software quality and capture RT domain knowledge and design expertise.

There are different kinds of reusable components that can be applied in different levels of abstraction (analysis, design and implementation) such as software components, framework and patterns. Among these techniques, patterns have been the most widely used since they can be applied in different steps of the software development cycle. In order to benefit from the reuse at the first phase of development, several works [2] [3] [8] are interested in defining analysis patterns that provide facilities to model functional requirements of RT systems. The specification of functional requirements helps to understand the modularization of the structure of RT systems and to address the system's inputs, outputs, and their behavioral interrelationships. In addition, it is useful as a basis for RT systems design, test and documentation since the design of a developed system is evaluated from the functional point of view [7]. Nevertheless, requirements analysis must not exclude the modeling of non -functional aspects that define the general qualities of the intended product such as security, reliability, scalability, etc. That is, the concept of quality is also fundamental to software engineering, and the modeling of non-functional characteristics must be taken into consideration for early specification of restrictions and external constraints that RT systems must meet. Thereby, we interest in this paper to define RT analysis patterns that capture both functional and Non-Functional Requirements (NFRs) knowledge. In fact, to model the NFRs, UML profiles [15] [16] and the NFR Framework [1] expressed by Softgoals Interdependency Graph (SIG) can be used. The representation of NFRs with SIG makes their understanding easier. Moreover, the SIG is easy to adapt according to the systems evolution by adding softgoals and solutions through AND-decomposition and OR-decomposition. It is also easy to incorporate non-functional properties with functional requirements. For this reason, we adopt the NFR Framework [1] to model NFRs and we adopt UML use case diagram to represent the functional requirements of RT applications.

The remainder of this paper is organized as follows. Section 2 provides an overview of proposed analysis patterns that deal with the modeling of functional requirements of RT systems. Section 3 describes the definition of three analysis patterns to model the functionalities as well as the non-functional characteristics of RT applications. Section 4 illustrates the reuse of the proposed analysis patterns through the modeling of two examples of RT applications using our developed CASE toolset. Finally, we conclude in Section 5.

II. RELATED WORK

The term analysis pattern has been coined by Martin Fowler [4] for patterns which capture requirements in an application domain in order to allow reuse across applications. In this section, we present works on analysis patterns intended for the modeling of RT systems requirements. Among these works, there are the analysis patterns defined by Konard for the modeling of embedded systems [2] and "AMR" (Autonomous Mobile Robot) analysis pattern [3] used to model robot software.

The analysis patterns proposed by Konard [2] tend to have an inclination to focus primarily on either the structural or behavioral phase of object analysis. Therefore,

 ISBN: 978-1-61208-111-3

they can be classified accordingly as structural object analysis patterns or behavioral object analysis patterns. In the following, we describe briefly two commonly used object analysis structural patterns, found in automotive embedded systems development: “Actuator-Sensor” and “User-Interface” patterns. (i) The analysis pattern “Actuator-Sensor” specifies basic types of sensors and actuators in embedded systems. It differentiates between four types of sensors and actuators: real, boolean, integer and complex. (ii) The analysis pattern “User-Interface” specifies the interaction between the user and the system through indicators (i.e., a type of actuators) and controls (i.e., a type of sensors).

The analysis patterns “AMR” (Autonomous Mobile Robot) [3] aim to develop and to facilitate the reuse of knowledge of robots software. Each pattern is described using both structural model and RT behavior model. The classes presented in structural model are classified accordingly to their RT behaviors as passive class, active class, event class, or implementation dependence class.

The examples of analysis patterns presented in this section focus on modeling the structural and behavioral aspects of embedded systems [2] and robot software [3], using UML class and sequence diagrams. The patterns proposed by Konard [2] represent also the functionalities of embedded systems using UML use case diagram. In fact, the representation of functional aspects shows clearly why RT systems are needed. But, the description of functionalities is not useful without the necessary non-functional characteristics such as dependability, reliability and security [9]. We must take into account the definition of these characteristics for the improvement of RT system quality and longevity [9]. Also, the quality of software system can only be achieved by considering non functional requirements as early as possible. If the NFRs are not considered at the early stage of the software development process (i.e., analysis phase), it may be difficult and expensive to address them in final product and it can lead the failure of the development.

III. RT ANALYSIS PATTERNS DEFINITION

In this section, we define three analysis patterns to model both the functional and non functional requirements of RT applications. The first pattern aims to model the data acquisition from the environment using sensors. The second pattern allows to model the control of data acquired from environment. While the third pattern deals with the representation of corrective actions when a violation is found.

In order to simplify the understanding of proposed analysis patterns, we describe these patterns using the following four elements: name, context, intention and solution. The solution shows how to model the RT applications requirements using UML and Softgoals Interdependency Graph [1]. The NFRs represented by Softgoals are associated with four use case diagram elements: actor, use case, actor-use case association and the system boundary. For example, in Figure 1, Dependability NFR is represented as Dependability [Transmission System] softgoal (denoted by a light cloud icon) that is related to the association between ‘Sensor’ actor and ‘Receive data from the environment’ use case. NFR softgoals are named using Type[Topic] nomenclature where Type represents a specific NFR concept e.g. Dependability, Security and Topic represents the context of the NFR [10]. The leaf-nodes of the graph represent alternative solutions for the operationalization of the NFR softgoals. They are denoted by dark clouds in the graph. Their corresponding degrees of contribution, indicating how well these solutions achieve NFR softgoals, are represented by the following signs: (MAKE (++), HELP (+), HURT (-), or BREAK (--)) [10].

A. The “Data Acquisition” analysis pattern

• **Name:** “Data Acquisition”.
• **Context:** this pattern is applicable in all RT applications which manipulate important volumes of data during the data acquisition phase.
• **Intention:** this pattern describes the functions as well as the quality that RT systems must have when acquiring data from the environment.
• **Solution:** Figure 1 shows the “Data Acquisition” analysis pattern that describes the interaction between the system and sensors. The sensors i.e., radar, camera, acquire data from the environment. Then these data are stored in RT databases. In distributed RT applications, data are transmitted to the databases of different sites with minimum time and cost of communication.

This pattern describes also the Softgoals Interdependency Graph for achieving NFR dependability. This graph represents a comprehensive set of software quality attributes related to dependability of data transmission system. This latter must be operable and able to perform its required function at any instant during its specified operating time. The dependability can be further achieved by ensuring data security and data transmission reliability.

- Security [Data]: the transmitted data from sensors to RT system must be secured against unauthorized accesses. A RT system may be useless if it does not satisfy security property. The NFR security is composed of availability, integrity and confidentiality. Availability means guarding against the interruption of service [11]. It can be achieved by replicating data. Integrity means guarding against unauthorized updates or other tampering. Confidentiality means guarding against unauthorized disclosure, i.e., release of relevant data [11]. To ensure integrity property, the transmitted data must be complete and accurate. The accuracy is the ability of a measure (e.g. speed, altitude, temperature, etc.) to match the actual value of the quantity being measured.

 ISBN: 978-1-61208-111-3

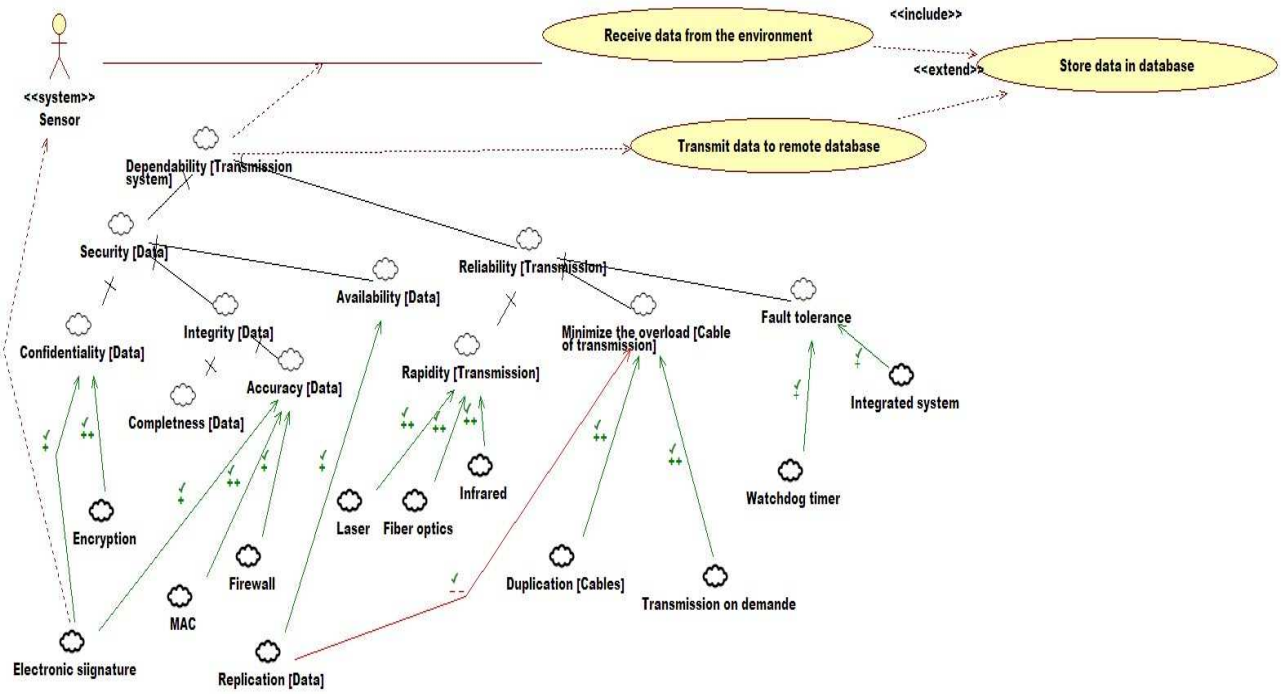

Figure 1. “Data Acquisition” analysis pattern

This constraint is ensured by the use of firewall that can block unauthorized accesses, by electronic signature based on encryption technology using public key and private key to authenticate the sender, or by using Message Authentication Code (MAC). The electronic signature can be also used to achieve confidentiality.

- Reliability [Transmission]: the transmission of data must be accomplished with minimum errors and time. In fact, the rapidity of data transmission is crucial in the RT applications since they must fulfill temporal constraints. It aims to ensure the respect of the validity of acquired data. If this quality is not satisfied, the data will not be fresh and therefore, lose their validity. Lasers, optical fiber and infrared are solutions to accelerate sensor data transmission. Besides, the degraded transmission mode which still corresponds to the specification is acceptable in RT system. This means that the imprecision of transmitted data is tolerable provided that they are received in time and they do not exceed the tolerable deviation. Fault-tolerance can be also applied by means of: (i) the watchdog timer, which is a device that triggers a system reset if the system does not respond due to some fault condition, (ii) and integrated system, which corresponds to a runtime equipment or software support for both real-time and fault tolerance.

B. The analysis pattern “Control of the environment”

• **Name:** “Control of the environment”.

• **Context:** this pattern is applicable in all RT applications which manipulate important volumes of data during the control phase.

• **Intention:** this pattern is used to model how control system monitors the acquired data and detects failures in RT applications.

• **Solution:** Figure 2 describes the shared and the varying functions of control system as well as their non functional characteristics. The variability is expressed through the generalisation and <<extend>> relationships. The generalisation relationship specifies different problems that can be detected in RT systems. Whereas the extension relationship specifies that the control of the environment functionality can be extended by the detection of errors related to data or actions. In fact, the control system can detect the non freshness of data if a measure’s value is used out of time interval during which it is considered valid. The control system can also detect an anomaly if boundary constraints are not fulfilled, i.e., a measure’s value is not between the minimum value and the maximum value defined by the user. For example, in freeway traffic management system, if the road segment density exceeds the limit, then the control system reports an anomaly. In addition, the control system can detect that the Quality of Data is not fulfilled (QoD) [12] if the maximum data error, defined to allow imprecise RT data, is exceeded. Note that the maximum data error is the upper bound of the difference between the value stored in the database and the new value acquired from sensor. Besides, the control system can detect an error if the deadline of an action is missed.

 ISBN: 978-1-61208-111-3

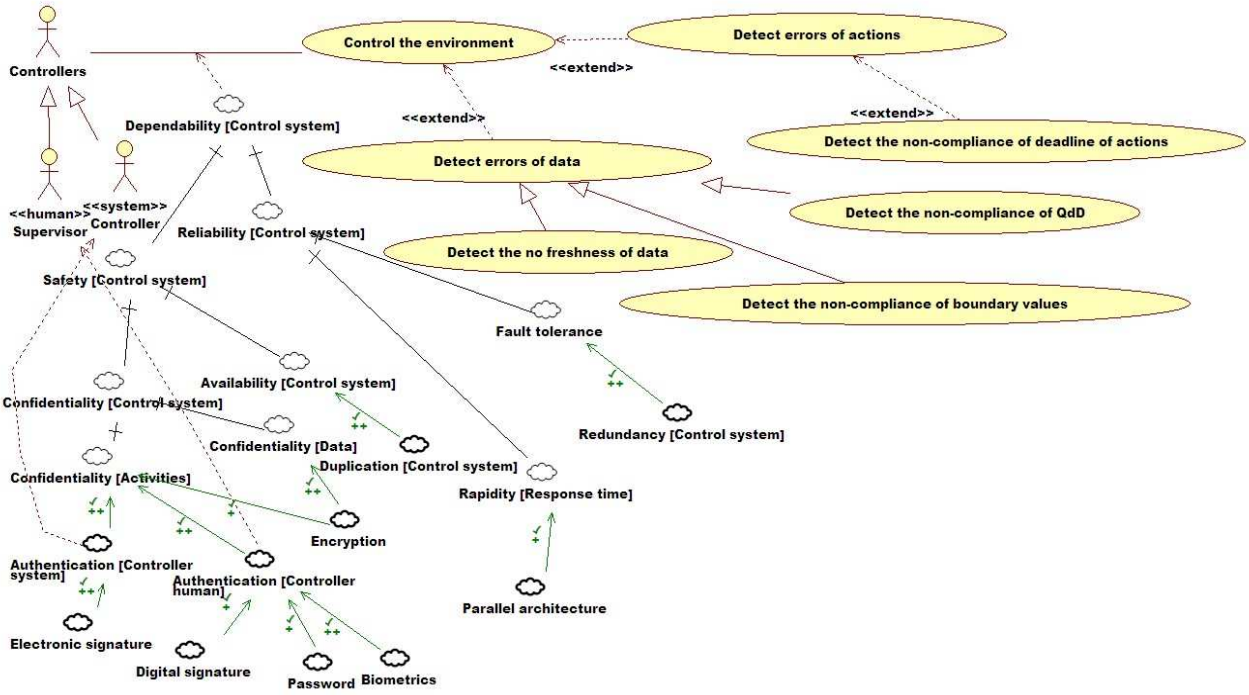

Figure 2. "Control of the environment" analysis pattern

When monitoring the environment, the control system must operate correctly. Thereby, it must fulfill the dependability property that is defined as the ability of the system to deliver specified services to the end users [13]. The dependability includes safety and reliability characteristics.

- Safety [Control system]: the control system must be protected against unauthorized accesses and must be available at all times. Availability means that the control system must be accessible at any time. For example, in case of failure, it must have a backup system. Confidentiality means that the control should be made by authorized members. This authorization is assigned by an authentication procedure. If the controller is a person, the authentication is ensured by password, biometrics or digital signature methods. Biometric can be implemented either by finger print verification or by voice or by face recognition. Digital signature is based on the technique of encryption with public key that is used to verify the signature and private key that is used to sign. Nevertheless, if the controller is a system, the authentication is ensured by electronic signature. Thus, only authorized controllers can exchange data and messages.

- Reliability [Control system]: The reliability is the ability of a system to consistently perform its intended or required function in time. Thereby, the control system must report the identified errors in a minimum time. In order to ensure the rapidity of response time, using parallel architectures is an effective solution. Besides, the reliability may be achieved with fault prevention by using different versions for the same processing.

C. The "Sending orders" analysis pattern

• **Name:** "Sending orders".

• **Context:** this pattern is used in RT applications when the control system reports an error that is occurred in the environment.

• **Intention:** this pattern aims to model the different recovery actions that are activated by actuators.

• **Solution:** Figure 3 illustrates the "Sending orders" pattern that describes the functional and non-functional requirements of actuator system. It allows to express variability since it shows different kinds of actions that can be triggered when the control system reports an error to the actuators. Indeed, actuators can trigger actions to achieve. It can also report alert messages such as voice messages, visual messages or alarms.

The non functional characteristics of actuator system are the same than those explained in the "control of environment" pattern.

IV. RT ANALYSIS PATTERNS REUSE

We have developed a toolset, called *AP-RT* (Analysis Pattern for RT applications), that deals with patterns representation and guides analysis patterns reuse. AP-RT allows the user to apply the proposed patterns in order to

 ISBN: 978-1-61208-111-3

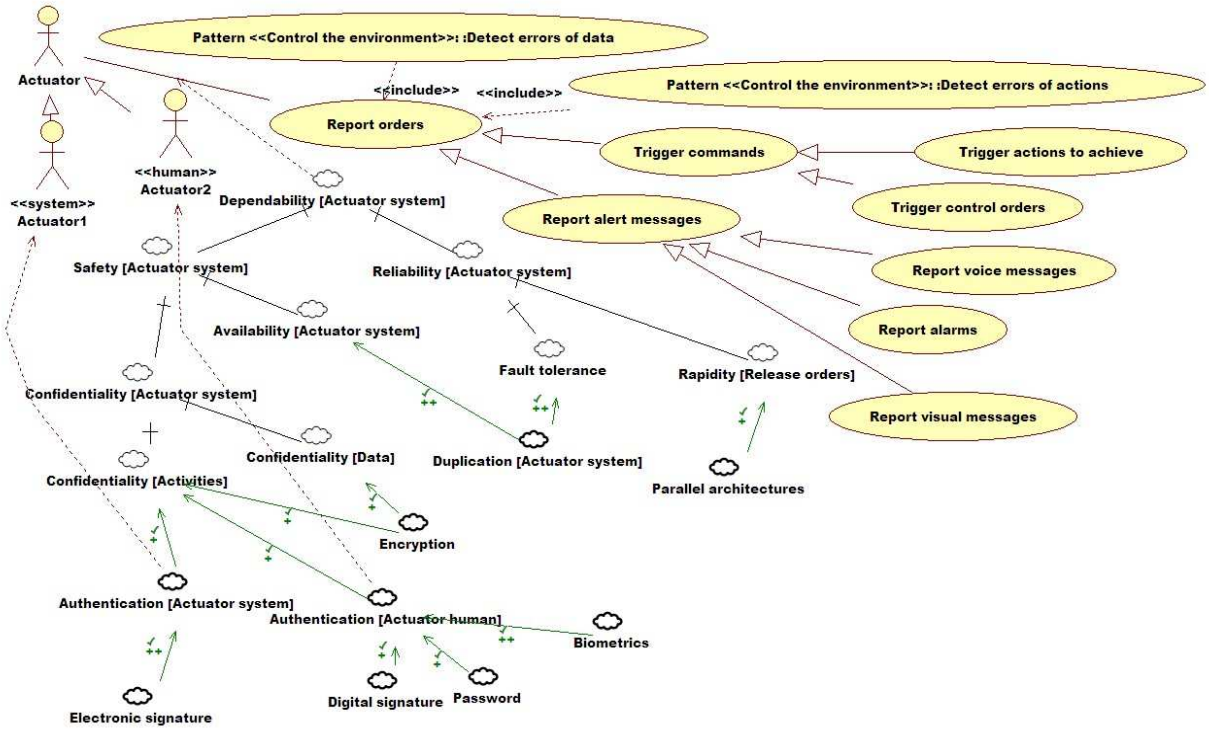

Figure 3. "Sending orders" analysis pattern

facilitate the modeling of functional and non-functional requirements of specific RT applications. As illustrated in Figure 4, the pattern can be created within AP-RT tool by means of the menu bar shown in the left of the figure. In addition to presenting patterns, AP-RT assists the designer in instantiating an analysis pattern selected and adapted via the menu bar shown in the top of Figure 4. After creating an initial application model that instantiates a pattern, the designer can continue developing the application model by adding, updating and removing various model elements using the AP-RT tool. Note that the tool does not allow the remove of pattern elements that are shared between all RT applications (i.e., pattern fundamental elements).

In the following, we describe the instantiation of defined analysis patterns using AP-RT tool. We illustrate the reuse of "Data Acquisition" pattern through the modeling of a road traffic control system. Also, we reuse the "Data Acquisition" and "Control of the environment" patterns in order to model a medical control application.

A. Example of a road traffic control system modeling

The road traffic management systems have become an important task intended to improve safety and provide a better level of service to motorists. We focus on modeling the acquisition data subsystem of a road traffic control application [6] and we explain how this design issue can be facilitated by the reuse of the "data acquisition" pattern. Current road state is obtained from the essential sources: radars, inductance loop detectors and supervision cameras. This system uses radars to measure vehicle velocity and to acquire vehicle crossing time of red light. It uses also inductance loops to measure traffic density (i.e., number of vehicles in a road segment). The supervision cameras are used to supplement and to confirm the data received through the vehicle detector stations and to provide information on local conditions which affect the traffic flow. The road traffic management system analyses the acquired data by the Central Computer System and informs drivers in real time about the state of circulation using variable message signs.

In order to ensure the security of data transmission, the road traffic control system maintains the confidentiality and the data integrity characteristics using electronic signature. In addition, the system maintains the rapidity through data compressing and laser technology. The data compressing is also used to minimize the overloaded transmission.

Figure 4 shows how to adapt the "Data Acquisition" pattern to model this system. The "Sensor" actor is instantiated by "Radar", "Inductance loop" and "Camera". Then, the "Receive data from the environment" use case is instantiated by the corresponding road traffic system functions which are: (i) "Receive vehicles speeds and crossing times" use case associated to the "Radar" actor, (ii) "Receive road segment density" use case associated to "Inductance loop" actor and (iii) "Receive image of vehicles" use case associated to the "Camera" actor.

 ISBN: 978-1-61208-111-3

The operationalizing softgoals, associated to road traffic control system, are: “Electronic signature”, “Laser” and “Compression data”. These desirable leaf-node solutions are labelled with (√) sign. Whereas, the proprieties that do not have Operationalizing Softgoals are labelled with (×) sign.

B. Example of a medical telesurveillance system modeling

Telemedicine for patient in residence, called “Televigilance”, concerns elderly persons, people with cardiac pathologies and persons in convalescence after hospitalisation, all needing a close medical supervision. The MEDIVILLE system [5] for telesurveillance of patients at home allows a more reactive medicalisation remotely released by urgency units (diagnosis, intervention).

This system is composed of three main components: (1) a terminal placed on the patient, continuously recording his physiological data, (2) an in-door reception base-station, processing physiological signals to detect emergency situation and create an alarm, which is retransmitted to the (3) third component corresponding to a remote medical monitoring server hosted in the televigilance centre exploiting all these data to decide any intervention. The patient’s terminal is coupled to actimetry and pulse sensors, indicating respectively the attitude of the patient (vertical/horizontal positions, activity) and his heart rate (pulse measurement). The base station continuously receives the emission signals from the patient’s terminal through a VHF radio link. On the other side the base station is connected to the remote server of the Surveillance centre through an IP channel using a VPN (Virtual Private Network) protocol.

During normal operation, the local home system is solicited at regular intervals (every 30 seconds) by the remote server in order to transmit the totality of recent sensors data. In an alarm case, the local system communicates with the central server and transmits, simultaneously to the alarm, the latest available data.

The MEDIVILLE system ensures a good quality of service. Firstly, an original noise reduction algorithm implemented in the microcontrollers aims to reduce the variations of pulse measurement and then to improve data accuracy. Secondly, the access to stored information is allowed only for authorised users: patient agent, practitioner agent and server manager. Thirdly, the importance and complexity of the functions taken over by this server strongly require a reliable system. The system contains a sufficient capacity and redundancy in order to cater for these conditions. Finally, the WS-DSAC *(Web Servers – Differentiated Services Admission Control)* [14] mechanism is used to accelerate response time. This mechanism is based on the balancing of imposed load among a certain number of computers to improve performance and on the use of admission control mechanisms to allow differentiated allocation of resources for specific service classes [14].

Reuse of the “Data acquisition” pattern: Figure 5 shows the reuse of the pattern “Data acquisition” to model the medical telesurveillance system.

The “Actimetry_Sensor” and “Pulse_Sensor” represent the instances of “Sensor” actor. These sensors are associated respectively to the following use cases: “Receive the attitude of the patient” and “Receive the pulse measurement”.

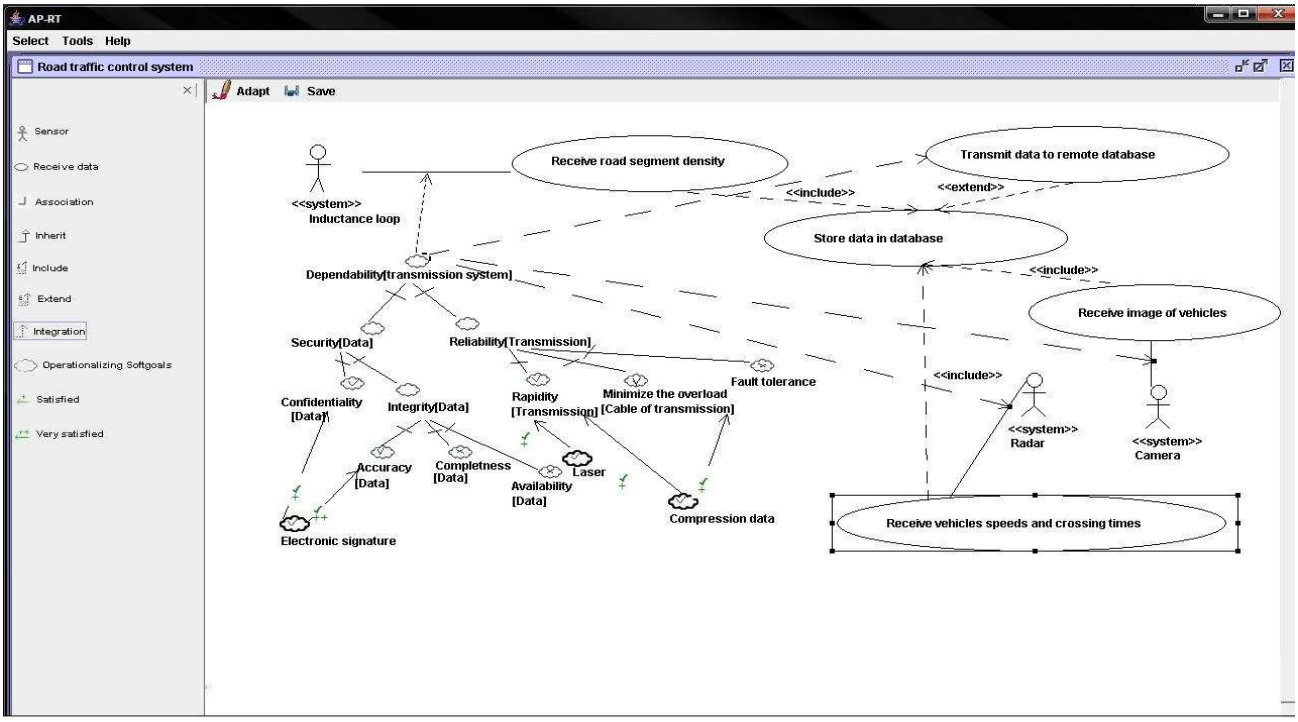

Figure 4. Data acquisition analysis model of Road traffic control system.

These functional requirements constitute the instances of the general use case “Receive data from the environment”. Furthermore, “VPN protocol” and “Noise reducing processing” constitute the operationalizing softgoals of the non functional requirements fulfilled by data transmission of MEDIVILLE system, which are: confidentiality and accuracy.

Reuse of the “Control of the environment” pattern: Figure 6 represents the control of patient information analysis model reusing the pattern “Control of the environment”. In fact, the “Controller” and “Control the environment” pattern’s elements are instantiated respectively by “Medical surveillance team” actor of MEDIVILLE system and “Control the patients at home” use case. Moreover, the in-door base station of MEDIVILLE system can detect patient’s falls and heart problems. Thereby, “Control the patients at home” use case can be extended by “detect falls” and “detect heart problems” use cases. These latter correspond to the instantiation of the pattern’s use case “detect the non-compliance of boundary constraints”.

Figure 6 represents the non-functional requirements that are fulfilled by the control system of the medical telesurveillance application. Confidentiality, rapidity of response time and reliability represent components of the quality of services offered by the MEDIVILLE system. These properties are ensured respectively by (i) “password” Operationalizing Softgoal, that allows to ensure the authentication of the medical surveillance members, (ii) “WS-DSAC mechanism”, that allows to perform admission control and load-balancing on a distributed platform, (iii) and “replication of processors”, that allows to prevent fault and then to ensure reliability of the system.

V. CONCLUSION

The main objectives of our work, described in this paper, are the definition of three analysis patterns and their reuse in RT applications. The first pattern represents the functional aspects of data acquisition system as well as the non-functional aspects of RT data transmission system. The second pattern shows the different kinds of anomalies that can be identified when time constraints related to the validity of data and deadline of transactions are not fulfilled. These anomalies are represented respectively by the use case “Detect the non freshness of data” and the use case “Detect the non-compliance of deadline of actions. The third pattern describes the different recovery actions that are activated by actuators. These patterns aim to facilitate the specification of RT applications analysis models and to improve their quality since they capture both past experience and best practices of RT systems designers.

Our future works include: (1) the definition of analysis patterns composition techniques in order to create a generic model that describes the common and the difference functions between RT applications, (2) the integration of the analysis patterns in the context of model driven architecture in order to add more assistance when defining the relationship between the proposed RT analysis and design patterns.

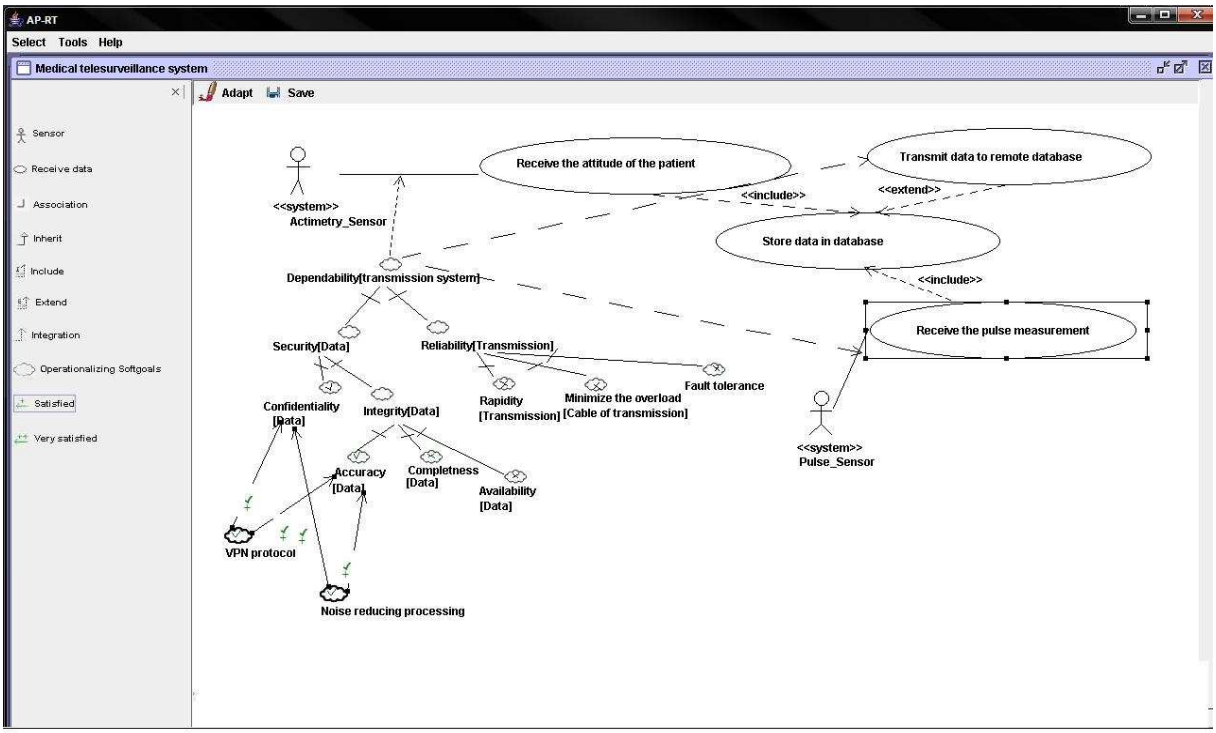

Figure 5. Data acquisition analysis model of medical telesurveillance system

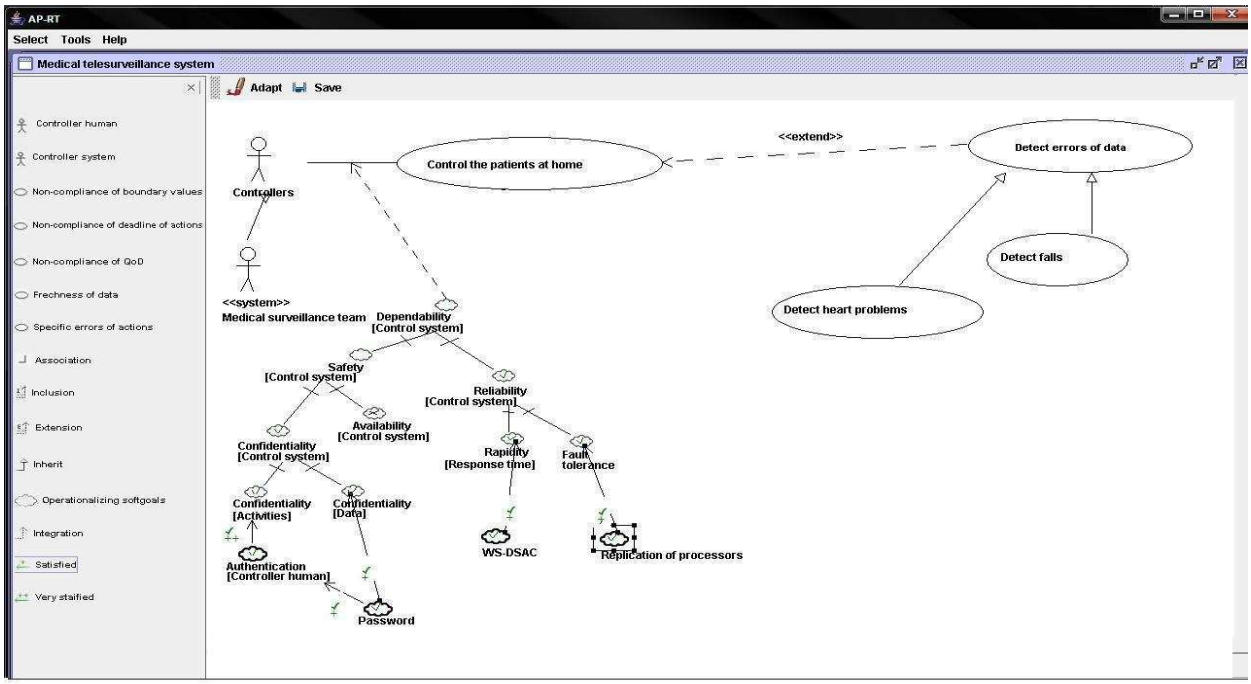

Figure 6. Data control analysis model of medical telesurveillance system

This could bring new benefits and impulse for both the knowledge capturing techniques and the software development process quality.

REFERENCES

[1] Chung L. and Supakkul S., Representing NFRs and FRs: A goal-oriented and use case driven approach. *Software Engineering Research and Applications (SERA2004)*, LNCS, 3647(29-41), 2005.

[2] Konard S.J., Cheng B H.C. and Campbell L. A., "Object Analysis Patterns for Embedded Systems", IEEE Transactions on Software Engineering, Vol. 30, No. 12, December 2004.

[3] Jawawi D., Deris S. and Mamat R., Software Reuse for Mobile Robot Applications Through Analysis Patterns, The International Arab Journal of Information Technology, Vol. 4, No. 3, 2007.

[4] Fowler M., Analysis Patterns – Reusable Object Models, Addison-Wesley, 1997.

[5] Baldinger J.L., Boudy J., Dorizzi B., Levrey J.P., Andreao R., Perpère C., Delavault F., Rocaries F., Dietrich C. and Lacombe A., Tele-surveillance System for Patient at Home: The MEDIVILLE System, Book chapter in Computers Helping People with Special Needs, Springer Berlin, LNCS 3118, 2004.

[6] Fasel W., On-line traffic surveillance. Strasse and Verkehr revue, ISSN 0039-2189, vol 89, N°4, pp. 31-35, 2003.

[7] Stone R. and Wood K., Development of a Functional Basis for Design. Journal of Mechanical Design, 122(4): 359-370, 2000.

[8] Esfahani N., Mirian-Hosseinabadi S.H. and Rafati K., A Real-Time Analysis Process Patterns, Book chapter in Advances in Computer Science and Engineering, Springer-Verlag, CCIS 6, pp: 177-181, 2008.

[9] Chung L. and Sampaio J.C., On Non-Functional Requirements in Software Engineering, Book chapter in Conceptual Modeling: Foundations and Applications, Springer-Verlag, LNCS 5600, pp. 363–379, 2009.

[10] Chung L. and Supakkul S., Capturing and Reusing Functional and Non-functional Requirements Knowledge: A Goal-Object Pattern Approach, Proceedings of IEEE International Conference on Information Reuse and Integration, 10.1109/IRI.2006.252471, pp. 539 – 544, 2006.

[11] Tonu S.A., Incorporating Non-Functional Requirements with UML Models, Phd thesis presented to the University of Waterloo, Ontario-Canada, 2006.

[12] Amirijoo M., Hansson J., and Son S. H., Specification and management of QoS in real-time databases supporting imprecise computations. IEEE Transactions on Computers, 55(3), 2006.

[13] J. Laprie, "Dependable Computing and Fault-Tolerant Systems", Depndability: Basic Concepts and Terminology in Eng-lish, French, German, Italian and Japanese. Vol 5. Springer-Verlag, 1992.

[14] Serra A., Gaïti D., Barroso G. and Boudy J., Assuring QoS Differentiation and Load Balancing on Web Servers Clusters, Proceedings of the IEEE Conference on Control Applications, pp. 885 – 890, Toronto, Canada, August 28-31, 2005.

[15] OMG, UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms V 1.1, April 2008.

[16] Zhu L. and Gorton I., UML Profiles for Design Decisions and Non-Functional Requirements, International Conference on Software Engineering, Proceedings of the Second Workshop on SHAring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent, page 8, ISBN 0-7695-2951-8, 2007.

 ISBN: 978-1-61208-111-3