Title	Systems Modeling of Automotive Electrical and Electronic Architecture
Sub Title	
Author	東, 欣一(Azuma,Kinichi) 西村, 秀和(Nishimura, Hidekazu)
Publisher	慶應義塾大学大学院システムデザイン･マネジメント研究科
Publication year	2013
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2013年度システムエンジニアリング学 第123号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002013-0010

Master's Dissertation 2013

Systems Modeling of Automotive Electrical and Electronic Architecture

東　欣一

Kinichi Azuma

（Student ID Number：81233021）

Supervisor:

Profs. Hidekazu Nishimura

March 2014

Graduate School of System Design and Management,
Keio University
Major in System Design and Management

Summary of Master's Dissertation

Student Identification Number	81233021	Name	Kinichi Azuma

Title: Systems Modeling of Automotive Electrical and Electronic Architecture

Abstract:

In this thesis, systems modeling of automotive electrical and electronic architecture are discussed, treating a backdoor system, one of the access and protection systems, as an example of the E/E architecture. This study shows the systems engineering process to obtain the back door system architecture by using SysML (Systems Modeling Language) to solve the complication issue of the E/E architecture design.

To stay successful in today's globalized world, automotive companies have to bring their products according to the customers' needs and much faster to the market. Also companies must add variant functions to product according to each market, while reducing the cost of their product development resources and increasing the development efficiency. However, it is very difficult for engineers who develop product to solve these issues with the current development process. Although engineers of automotive companies understand necessary of systems engineering, they do not concretely find out the process of systems engineering.

First, the E/E architecture of an existing back door system consisting of a body control module and contactless sensor is analyzed to clarify the functions and the structure. Interaction between user and the existing back door system and that among the internal system are clarified using behavioral diagrams. Second, requirements for a new back door system are analyzed and user needs of user friendliness and "Omotenashi" are defined using the use case diagram and the sequence diagram. The E/E architecture of the new back door system, which can be operated by users' natural motion, is designed based on the model-based systems engineering (MBSE). And the functions of the new back door system can be derived by analyzing the requirements. Using the activity diagram, the functions are allocated to each component composing the system. The traceability of the requirement can be held in system models obtained by SysML.

From the interview to domain specialists it is showed that the process of E/E architecture design by MBSE must be effective to automotive companies. It is pointed out that the early stage of development process on the system level is very important and especially discussion between systems engineer and domain specialists through SysML diagrams is very useful.

Keywords:

Systems engineering, Electrical and Electric architecture (E/E), Mode-based systems engineering (MBSE), SysML

Table of contents

1. Introduction

1.1 Research background

1.1.1 E/E architecture plays an important role in automotive

Nowadays, science and technology develop very fast. No matter you want to admit it or not, people actually become much greedy for demanding for more and more function of products. This situation is quite obvious in automotive industry. The automotive industry has witnessed a huge increase in functionality based on electrical and electronic control units (ECUs) in recent years. Electrical and electronic are generally used in automotive system in order to provide huge improvement of functionality, performance and product properties. For instance, it is impossible to guarantee the safety, control the emission, provides users a comfortable interior environment of a vehicle without the use of electronic control units (ECUs). According to some resources, 80% of the innovation in a vehicle comes from electronics. On the other hand, 49.2% of car breaks down in Germany were based on the problems of the electrical and electronic architecture (E/E architecture). And most recalls from automotive industry are due to the failure of E/E architecture, Mercedes has $30 million recall of 680,000 cars in 2004 due to defects in brake-assist-by-wire system, which is included as a part of the E/E architecture.

1.1.2 E/E architecture is getting more and more complicated

In the early days, there are not so many functions required in a vehicle, so only a few ECUs have been installed in the vehicle and they are not connected with each other. Each ECU is worked as a stand-alone unit. It is simply enough for the engineers to develop and modify.

As a development of electronics and embedded system technology, and also with the requirement increasing, engineers have to plus more and more ECUs to realize the function requirement. Electronic content in vehicles has steadily increased. As a matter of fact, the E/E architecture gets complex substantially to implement a lot of functions. From 2004, the BMW 5 and 7 series have 70 networked ECUs.

Engineers then developed the In-Vehicle Network (IVN). In IVN, each ECU is connected with another, with point to point connection. Though this kind of connection provide a simply topology and it can make the functionality easily, the number of wiring which connect the ECUs is very huge. And the E/E architecture is hard to modify and extend. Besides this, the number of ECUs is continuously increasing as the function is added, the number of wiring and the connection nodes are increasing exponentially. So the weakness of IVN is not good for the development of the E/E architecture, and the production and the maintenance of the automotive.

In order to solve this problem, networked E/E architecture is developed by using bus architecture, at the first ten years of the twenty one century. In the networked architecture, each ECU has his own sensors and actuators, but networked architecture still has the limitation of providing flexibility and scalability. For instance, a premium-class automotive still contains around 100 million wires, running on 70 to 100 ECUs networked throughout, and the sensors and the actuators do not have their own network interface. So as the requirement of the automotive continuously increasing, the current networked E/E architecture will soon reach its limitation. Currently, the E/E architecture is in hierarchal structure.

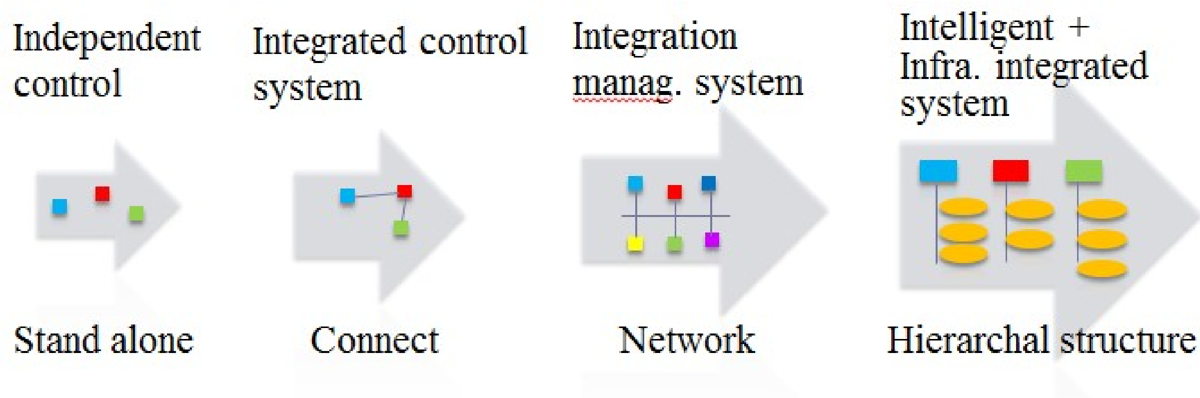


Fig 1.1 E/E architecture gets complicated due to growing sophistication of user needs

Moreover, not only the ECUs in the current E/E architecture is in hierarchal structure, but the E/E architecture itself integrates across other five subsystems which composing a vehicle. They are body subsystem, power train subsystem, chassis subsystem, ITS subsystem, IT subsystem. And each subsystem has its own design departments, body design team, power train design team, but they are relatively independent, while during E/E architecture design, E/E design team is involved in other five department. And currently there is no systems engineering process when designing the E/E architecture.

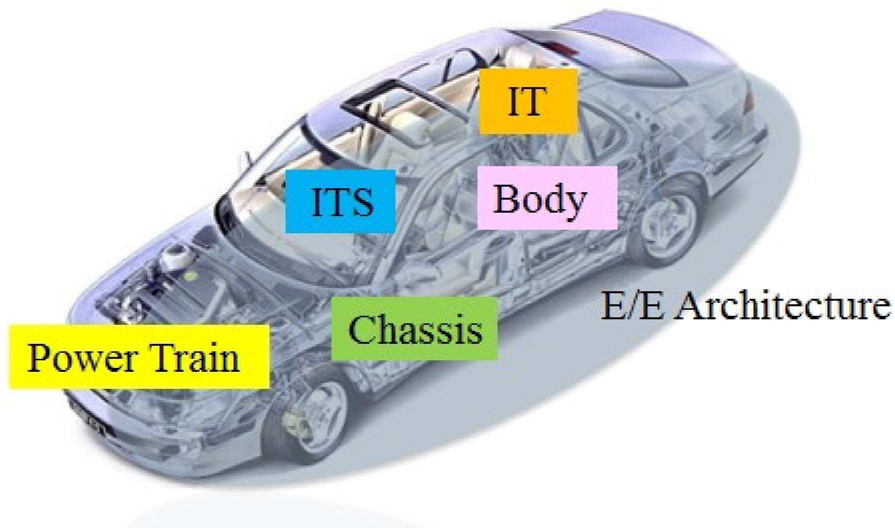


Fig 1.2 E/E architecture integrates across other five subsystems composing a vehicle

1.2 Problem Statement

In current situation, the development cycle of E/E architecture usually lasts for around three years, two years for the “product development declaration” (PDD) and one year for the “engineering completion”. In the stage of PDD, the function design and allocation, and the system architecture design should be completed. And in the stage of “engineering completion”, prototype will be produced to test whether the products can run well and meet the requirement of stakeholders. Though the time duration is decided, but in many cases, engineers can’t keep up with the time because of the function is always changed or added during the process, so they have to finish the design without seriously consideration in order to meet the deadline or even ask for prolong the time cycle.

Moreover, there are only component engineers in most of the automotive companies of Japan. They are only responsible for the parts of the system and they are much more focus on the optimization of the part design rather than on facilitating assembly design. There is totally no system level conception during the design of E/E architecture. ECU parts are continuously added to meet the increasing requirement. With the number of ECU parts increased, E/E architecture is becoming huge mass and hard to be modified. In addition, without view of system, it might also make the engineers to design the automotive with the function which can’t really meet the requirement of the stakeholders needs. And it is difficult for engineers to have documents left which record the traceability of system requirement by using current process.

In addition, the creation process of the automotive companies from Japan is bottom up oriented and part centered. In body-in-white design, at first, engineers design the feature part-spreading surfaces. Then they break the parts down into individual small parts and detailing is done on the part level. Parts re-assemble is considered only after each part has been designed. In other words, assembly design is seen as a process step which follows sequentially after part design. As a result, E/E architecture is not designed at the beginning but it is just formed after each part has been assembled. Because of there is no system engineers, the specification of the whole system is always changed and followed by cost increased.

Though the problems mentioned above draw the attention of the automotive industry, and the suppliers such as Bosch and Denso already provide the PREE Vision of systems engineering process to the automotive companies. But through the interview with the engineers from automotive company, they do not have the knowledge of what exactly should do by using the systems engineering.

1.3 Research Purpose

In this thesis, systems engineering process is clarified to obtain a back door system architecture, one of the access and protection systems involved in the E/E architecture by using SysML (systems modeling language). I will first analyze the E/E architecture of an existing back door system which users can operate with contactless sensor in order to clarify the functions and the structure. And then systems engineering process is shown to architect the new back door system which can be operated by user's natural motion, based on the model-based systems engineering (MBSE) in order to show the merit of designing E/E architecture from system level compared with the current development process. With the MBSE approach, user needs can be decomposed and functional requirements for the new back door system can be derived by using the behavioral diagrams such as the use case diagrams and the sequence diagrams provided by SysML. Then the functions of the new back door system can also be well derived by analyzing the requirements, and the functions can be allocated to each component using the activity diagram. Moreover, the traceability of the requirement can be held in system models such as use case diagrams, sequence diagrams and activity diagrams obtained by SysML. At last, the system models will be shown to the domain specialists from the automotive company, confirmation of effectiveness of MBSE in the early stage of development process on the system level and importance of discussion between systems engineers and domain specialists through SysML diagrams.

2. An Overview of the Systems Modeling

2.1 Systems Engineering

Systems engineering is an approach to provide systems level solutions to technologically challenging and mission-critical problems. It can be used to develop balanced system solutions in response to diverse stakeholders needs. The solutions are generally cover hardware, software, people, facilities, etc. Moreover, systems engineering also provide the application of both management and technical processes to achieve balance of the system solutions and decrease the risks which may have the impact on the success of the project. Besides, the system engineering is based on the systems thinking. Systems thinking occur through discovery, learning, diagnosis, and dialog that lead to sensing, modeling, and talking about the real-world to better understand, define, and work with systems [1].

Systems engineering standards has been developing process standards that include EIA 632, IEEE 1220, and ISO 15288 [1]. These standards provide a foundation for establishing a systems engineering approach which addressing broad industry needs and reflect the fundamental tenets of systems engineering. Though the cover of life cycle of these process standards is different, the systems engineering process is the same which includes (1) system design, (2) system engineering management, (3) system analysis and evaluation. And the Vee model provides a useful illustration of the systems engineering activities during the life cycle stages. Fig 2.1 shows the Vee model. The Vee model illustrates the “Concept Stage”, “Development Stage” and “Production Stage” of ISO 15288. And IEEE 1220 is much more focus on the “Concept & Architecture Selection and Design to Specification” on the left side of the Vee model. And the IEEE 1220 systems engineering process of the “Concept & Architecture Selection and Design to Specification” is shown in Fig 2.2.

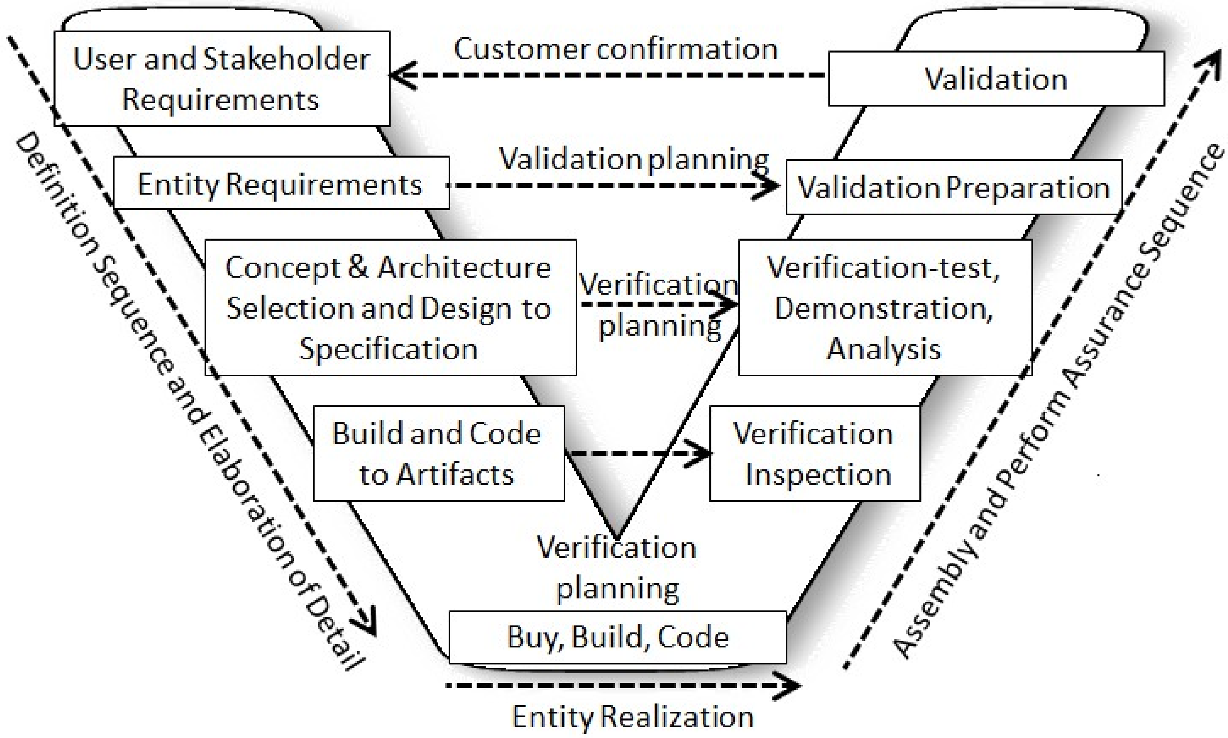


Fig 2.1 Vee model

Fig2.2 shows the IEEE 1220 systems engineering process which decomposes into eight sub-processes. Requirements analysis establishes system capabilities and product performance and defines the operational environments, human and system interfaces, physical characteristics, and other constraints that impact design solutions. The project team conducts various tradeoff and risk analysis to identify and resolve conflicts, ideally resulting in a requirements baseline that balances an operational view—how system products serve the users; analysis to identify and resolve conflicts, ideally resulting in a requirements baseline that balances an operational view—

how system products serve the users; a functional view—what analysis to identify and resolve conflicts, ideally resulting in a requirements baseline that analysis to identify and resolve

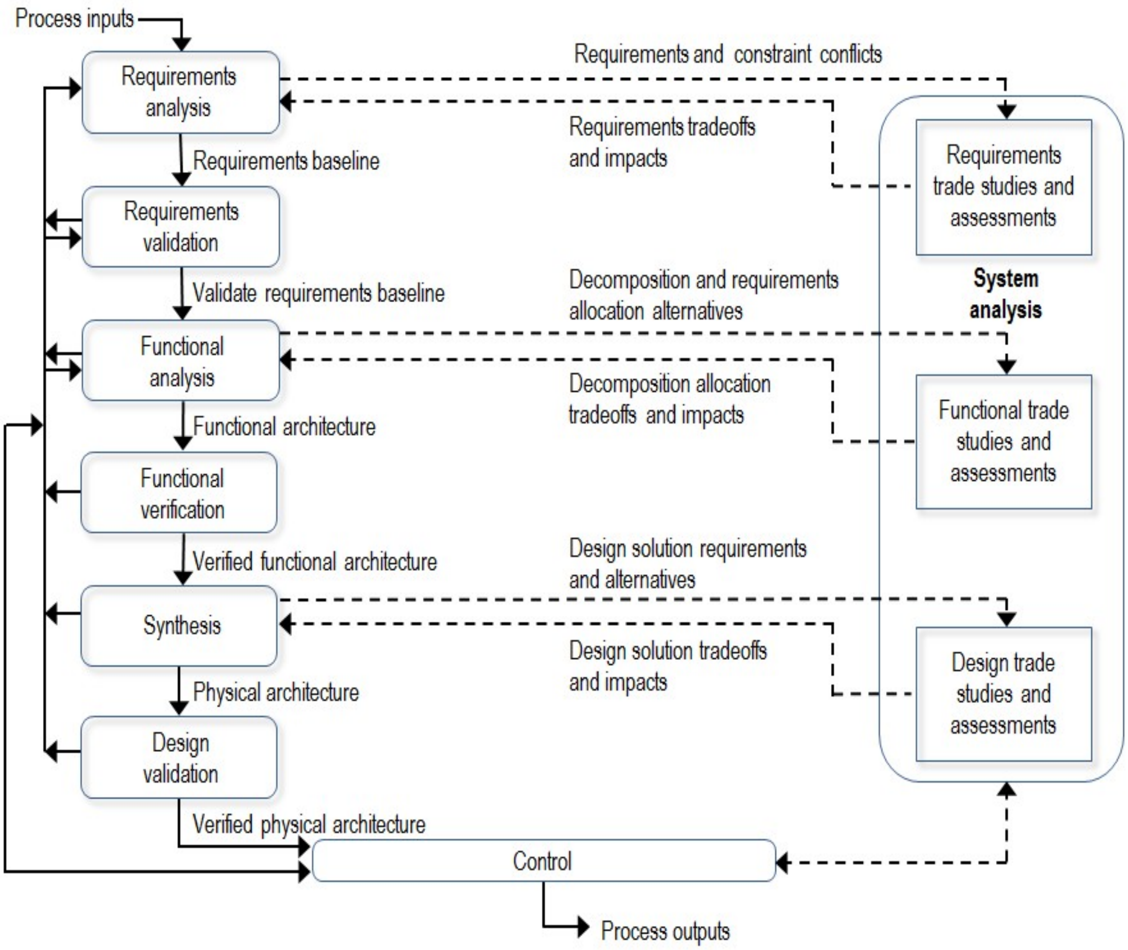


Fig 2.1 IEEE 1220 systems engineering process

conflicts analysis to identify and resolve conflicts, ideally resulting in a requirements baseline that balances an operational view—how system products serve the users; a functional view—what the products do; and a design view—design considerations. Requirements validation evaluates this baseline to ensure that it adequately addresses stakeholder expectations, enterprise and project constraints, external constraints, and system and life-cycle support considerations. If not, the project team repeats requirements analysis and requirements validation until achieving a satisfactory validated requirements baseline. Functional analysis refines the problem statement for a system solution— as defined by the requirements baseline—and breaks down the system functions to lower levels to satisfy system design elements. This process produces a functional architecture. Functional verification assesses completeness in satisfying the validated requirements baseline and yields a verified functional architecture. Synthesis translates the verified functional architecture into design architecture. It derives a preferred solution from a set of alternatives typically based on associated costs, schedule, performance, and other risk implications. Design verification assures traceability from the lowest level of the design architecture to the verified functional architecture, and also that the design [2].

In the process, the project shall perform the tasks of requirements analysis for the purpose of establishing what the system will be capable of accomplishing, how well system products are to perform in quantitative, measurable terms; the environments in which system products operate, the requirements of the human/system interfaces; the physical/aesthetic characteristics, and constraints that affect design solutions. The market needs, requirements, and constraints are derived from stakeholder expectations, project and enterprise constraints, external constraints, and higher-level system requirements. These are documented in a requirements baseline. The requirements baseline guides the remaining activities of the systems engineering process (SEP) and represents the definition of the problem to be solved. For each application of the SEP, the project refines previously defined requirements for upper levels of the system architecture, as appropriate, and defines requirements for the system under development. Then functional architecture is established to define the allocation of performance requirements from which design solutions should be determined via synthesis. Prior to synthesis, the functional architecture should be verified to assure that it meets the requirements of the validated requirements baseline. And then, physical architecture is designed to document the design solution and interfaces. The physical architecture includes the requirements traceability and allocation matrices, which capture the allocation of functional and performance requirements among the system elements. Physical architecture definitions should be documented in the integrated repository, along with trade-off analysis results, design rationale, and key decisions to provide traceability of requirements up and down the architecture. Verification of the physical architecture should be accomplished to demonstrate that the architecture satisfies both the validated requirements baseline and the verified functional architecture [2].

2.2 An Overview of Model-Based Systems Engineering

Model-based systems engineering is one of the supporting methods of systems engineering by using system model. A system model is the conceptual model that describes and represents a system with all the stakeholders. System model can be described by using several languages. SysML is one of these languages and is defined by OMG in 2006. Model-based systems engineering (MBSE) applies system modeling as part of the systems engineering process in order to support analysis, specification, design, and verification and validation [4]. MBSE begins in the conceptual design phase and continuing throughout development and later life cycle phases. The system model includes system specification, design, analysis, and verification information. The model is made of model elements which represent requirements, design, test cases, design rationale, and their interrelationships among these things.

In addition, using system model guarantee the design of a system can satisfies its stakeholders' requirement and meanwhile it can support the allocation of the requirements to the system's component. The system model includes the components interconnections and their interface, components interactions and the associated functions which the components must perform. And the component performance and the physical characteristics are also contained in the system model. As a result, the textual requirements for the components might be captured in the model and traced to system requirements.

Moreover, the system model is used to specify the requirements of component and the system model itself can be an evident which shows the agreement between the system level designers and the subsystem designers. With the system model, the subsystem designers could get to know the subsystem requirements by acknowledged the documentation which is automatically generated from the model. Then the designers can continuously provide the information about how the subsystem design could satisfy the requirements in a similar way as in the upper design level. Designers can maintain the traceability between system and subsystem requirements and at the same time, enabling to specify and integrate subsystems into the system by using the system models.

Based on the system models, MBSE enhances the ability to capture, analyze, share, and manage the information associated with the complete specification of a system. And the system models is equal to the architecture in the IEEE 1220 SEP mentioned in the last section.

MBSE approach improved the communications among the development stakeholders such as the customers, program managers, system engineers, hardware and software developers and testers, etc. From the perspective of different people, a system model can be viewed from multiple ways, thus increase the ability to manage the system complexity and become easily to analyze the impact of changes. And the quality of the product will also be improved because of using the unambiguous and precise model of the system which can be evaluated for consistency, correctness, and completeness. For the subsequent development, MBSE approach can also enhance knowledge capture and reuse the information by capturing information in more standardized ways and leveraging built in abstraction mechanisms inherent in model driven approaches. Therefore, systems modeling is very important.

2.3 Systems Modeling Language

SysML is one of the languages of model-based system engineering. SysML is used as a standard from the Object Management Group (OMG) in May, 2006. It is based on the Unified Modeling Language (UML) which is also a standard for software engineering developed within the OMG. SysML is a modeling language which can represent the architectures of the systems and the products as well as their behavior and functionalities. The architecture shows the elements which can realize the functional aspect of their products. Activity diagrams and block diagrams are reused from the UML and were extended in SysML while state machines diagram, internal block diagram and use case diagram are used from UML 2.0 without modification. Requirements diagram, parametric diagram are the new diagrams available only in SysML. As a result, SysML contains nine diagrams as follows:

(A) Requirement diagram

(B) Activity diagram

(C) Sequence diagram

(D) State machine diagram

(E) Use case diagram

(F) Block definition diagram

(G) Internal block diagram

(H) Parametric diagram

(I) Package diagram

The nine diagrams introduced above can be arranged into four pillars. It shows the system model represents the key system aspects as defined in SysML, including system's "Requirements", "Structure", "Behavior" and "Parametrics". (A) Requirements diagrams was included in "Requirements" while (F) block definition diagram, (G) internal block diagram and (I) package diagram were included in "Structure". And "Behavior" contains (B) activity diagram, (C) sequence diagram, (D) state machine diagram and (E) use case diagram while (H) parametric diagram was included in "Parametrics". SysML shows the architecture of the system by using these four pillars. Designers can choose the suitable diagrams to design the systems abided by the systems engineering process, no matter the sequence of using the diagrams and not all the nine diagrams must be used in the process. Moreover, since the system model is an interconnected set of model elements, the multiple cross-cutting relationships between the model elements enable the system model to be viewed from many different perspectives and the traceability also can be found among the four pillars.

3. Analysis of One Example of Existing Access & Protection System

3.1 Problem of Existing Access &Protection System

In this chapter, a current access and protection system is analyzed to clarify the function of each ECU and the interfaces. Detailed model that describes many aspects of this system was given. It is shown that as the function is added to the existing E/E architecture the E/E architecture becomes more and more complicated. First, the definition of the access and protection system is the systems which contain the driver door system, rear passenger door system and the back door system. And the current access and protection system analyzed in this chapter is the back door system. In order to distinguish the current back door system with the new back door system which discussed in the next chapter, the current back door system is named as “Contactless Back Door System”.

The “Contactless Back Door System” analyzed is a kind of backdoor which there is no need for directly touching the backdoor to open it. Users can open it by contactless way since there is a sensor embedded in the backdoor which can detect the users’ gesture for opening the back door.

Throughout the meeting with the engineers from associated company, I came to know that “Contactless Back Door System” which with the function of detecting users’ hands by sensor is a derivative product based on the initial backdoor system and the second generation backdoor system. So, let me introduce these two back door system at first. In Fig 3.1, a picture of an initial backdoor system is shown. The part included in the red rectangle is the initial back door system.

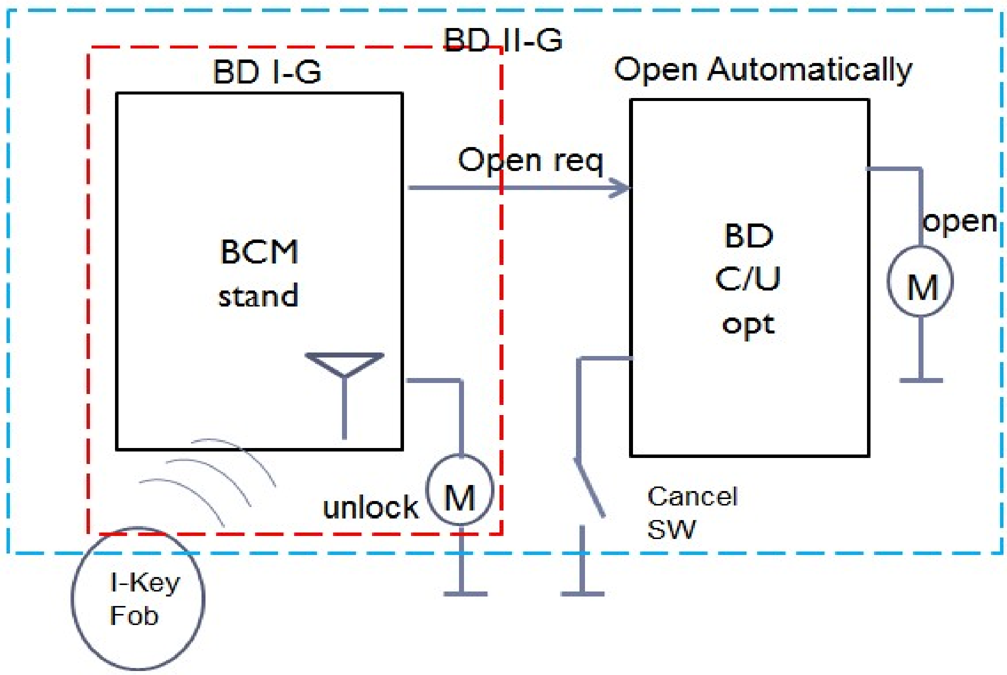


Fig 3.1 the initial back door system and the second generation of the back door system

The E/E architecture of the initial backdoor system consists of only a body control module (BCM). In automotive electronics, body control module is a generic term for an ECU responsible for monitoring and controlling various electronic accessories in a vehicle's body. Typically in a car the BCM controls the power windows, power mirrors, air conditioning, immobilizer system, central locking, etc. The BCM communicates with other on-board computers via the car's vehicle bus, and its main application is controlling load drivers-actuating relays that in turn perform actions in the vehicles such as locking the doors. So in the initial backdoor system, engineers only provide users with a way of plugging in the key or pushing the button on the key fob to unlock the backdoor and open the backdoor by themselves.

Then, as the users' requirement increased, people want the backdoor open automatically instead of opening by their hands. Therefore, engineers added an ECU which named "Backdoor

Control Unit" (BD_CU) to realize this function. In Fig 3.1, a picture of the second generation backdoor system is shown. The part included in the blue rectangle is the second generation back door system. The BD_CU will control the backdoor to open automatically once received the signal from BCM. And the second generation back door system act as the "Vehicle 2013" which is one of the system components of the new back door system discussing in chapter four.

Now, let me turn back to the "Contactless Back Door System". Like the second generation backdoor system introduced above, engineers added an ECU called "Sensor Control Unit" (SENSOR_CU) to meet the requirement of users who do not want to open the backdoor by operating the I-key fob or users with the cargo who are not convenient to operate the I-key fob. In Fig 3.2, a picture of the "Contactless Back Door System" is shown.

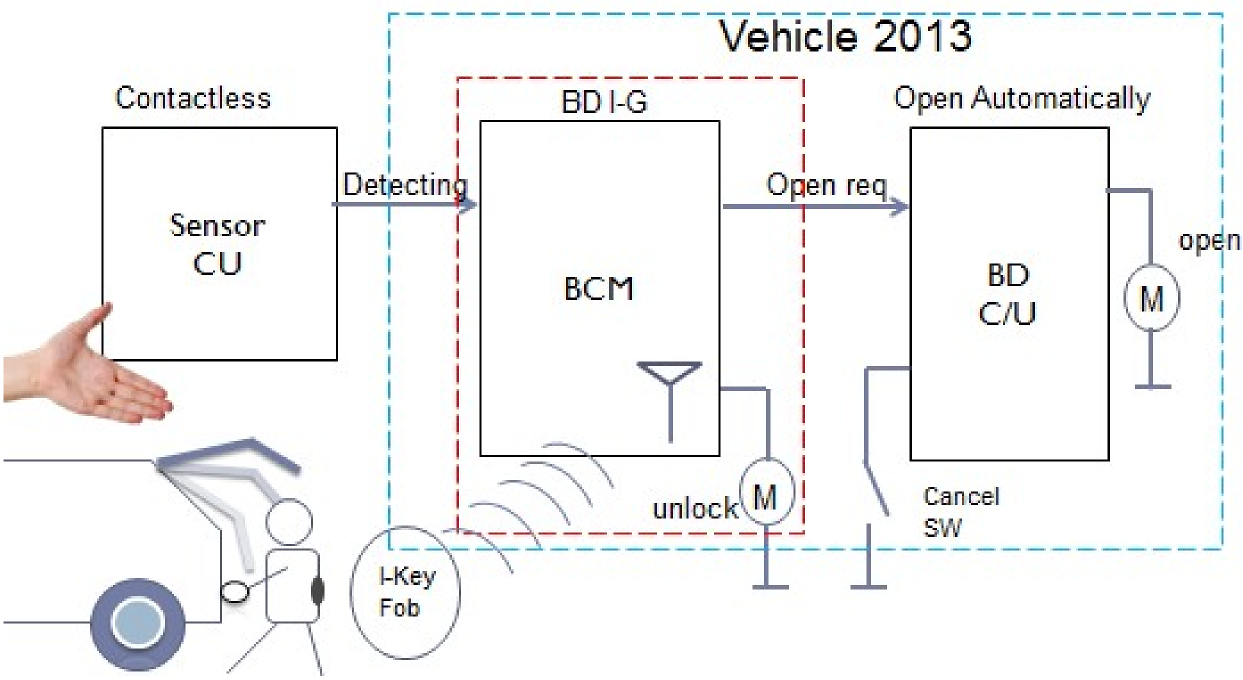


Fig 3.2   Contactless Back Door System

The back door can be opened once the user's hand is detected by the sensor, but is it really reflects the user-friendliness? If you want to open the back door when you are carrying cargo on your hands, it is definitely inconvenient. Thus, the design of “Contactless Back Door System” is not user-friendliness and in Japanese, it is not Omotenashi. And from Fig 3.1 and Fig 3.2, I notice that “Contactless Back Door System” is almost developed based on the former generation of backdoor by adding ECU with corresponding function. Due to this situation, I guess that with the requirement of automotive increased, the number of ECUs would also be increased, and finally, the E/E architecture would be very complicated. I get the systems diagrams of “Contactless Back Door System” from associated company’s engineers in order to verify my hypothesis. Unfortunately, from these diagrams, it is very hard to get a whole view of the system and the function of each component and the interfaces among components are also not easy to clarify. Therefore, at first, I try to analyze the E/E architecture of “Contactless Back Door System” from a model based systems engineering view by using SysML in order to clarify the functions and interface between each components.

3.2 Analysis of Existing Back Door System

3.2.1 Analysis in terms of use case

Since the “Contactless Back Door System” is current existed, we just focus on “Behavior” and “Structure”, which are the two of the four pillars of the SysML, so the requirements and the parametrics is not analyzed in this chapter. However, systems engineering process is abided in analyzing the “Behavior” and “Structure”.

First of all, it is important to think about how the users use the system, and from this, requirement can also be captured. Use case diagram included in “Behavior” could help us to describe the functionality of a system in terms of how it is used to achieve the goals of its various users. In Fig 3.3 shows a use case diagram as the top level of the system, containing the key diagram elements, system boundary shows the boundary of the system, and a main use case as "OPEN_BD" which includes three other use case, and three actors. In the use case diagram, actors are used to represent the users of a system. Actor is used to represent the role of a human, an organization, or any external system that participates in the use of some system. Actors may interact directly with the system or indirectly through other actors. At the beginning, there is no "external environment" as an actor in Fig 3.3. It could be harmful for engineers to design the system at the very early step. Since "external environment" also should be considered as a stakeholder of the system, which may include the weather like such as rainy, snowing, and may also include the road condition, the external noise, etc. These kinds of factors might change the functional requirements of the system.

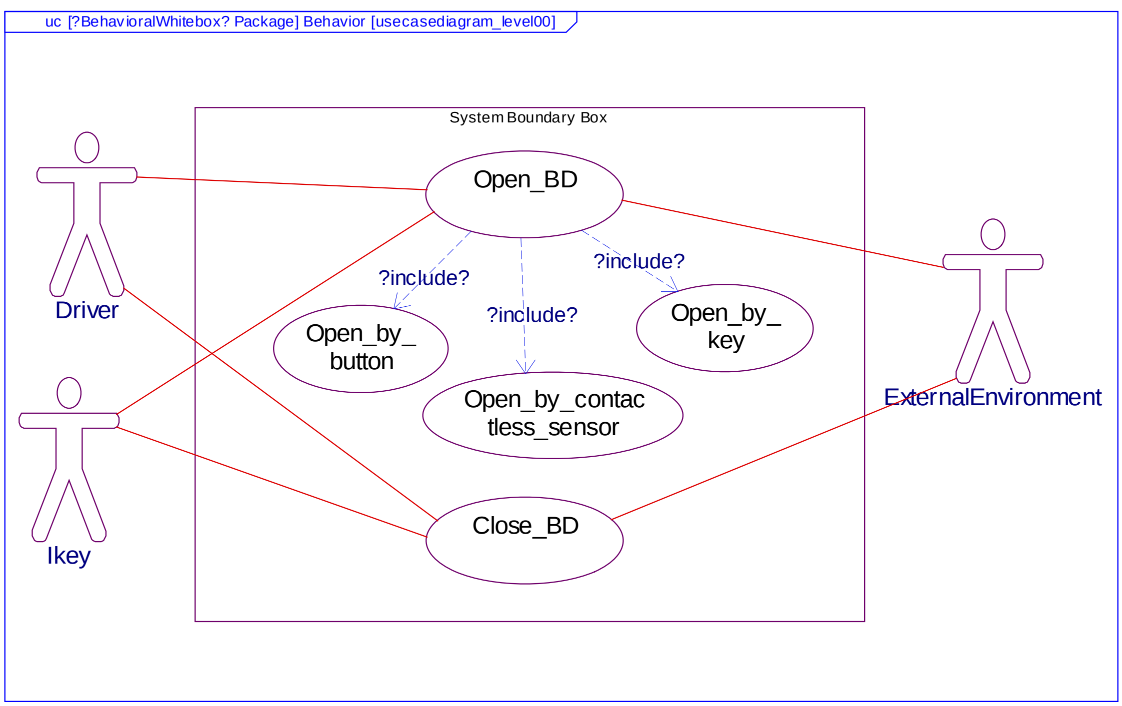


Fig 3.3 Use case diagram of the top level of the context of operating back door

In Fig 3.3 the inclusion relationship means it allows one use case referred to as the base use case. Here, “OPEN_BD” is the base use case. Base use case includes the functionality of another use case which called the included use case, as part of its functionality when performed. Here, “OPERATE_BY_KEY”, “OPEARTE_BY_BUTTON” and “OPERATE_BY_CONTACTLESS_SENSOR” are the included use cases. They are always performed when the “OPEN_BD” use case is performed. A behavior that realizes the “OPEN_BD” use case often references the behavior of the included use cases. Included use cases are not supposed to show a functional decomposition of the base use case. But on the other hand, a base use case and its included use cases often represent different aspects of the required functionality. In next section, “OPERATE_ BY_CONTACTLESS_SENSOR” is analyzed since it is the main required functionality of the “Contactless Back Door System” which we want to visualize.

3.2.2 Analysis in terms of Interactions

In the last section, system’s behavior is illustrated by using use case diagram, but the interactions between the use case and the actors are not analyzed. In this section, sequence diagrams are used to represent the interaction between the backdoor system and its environment and the users. A message can represent the invocation of a service on a system or the sending of signal. Sequence diagram is now widely used in modeling system behaviors, an interaction can be drawn as a specification of how components of a system should interact. Fig 3.4 is the sequence diagram of modeling system level behaviors which is corresponded to the use case diagram introduced in the last section. In Fig 3.4, the three actors, I-key fob; users outside the car; external environment and back door system block are represented by lifelines on a sequence diagram. A life line is shown by using a rectangle with a dash line descending from its tail. The name is contained in the rectangle.

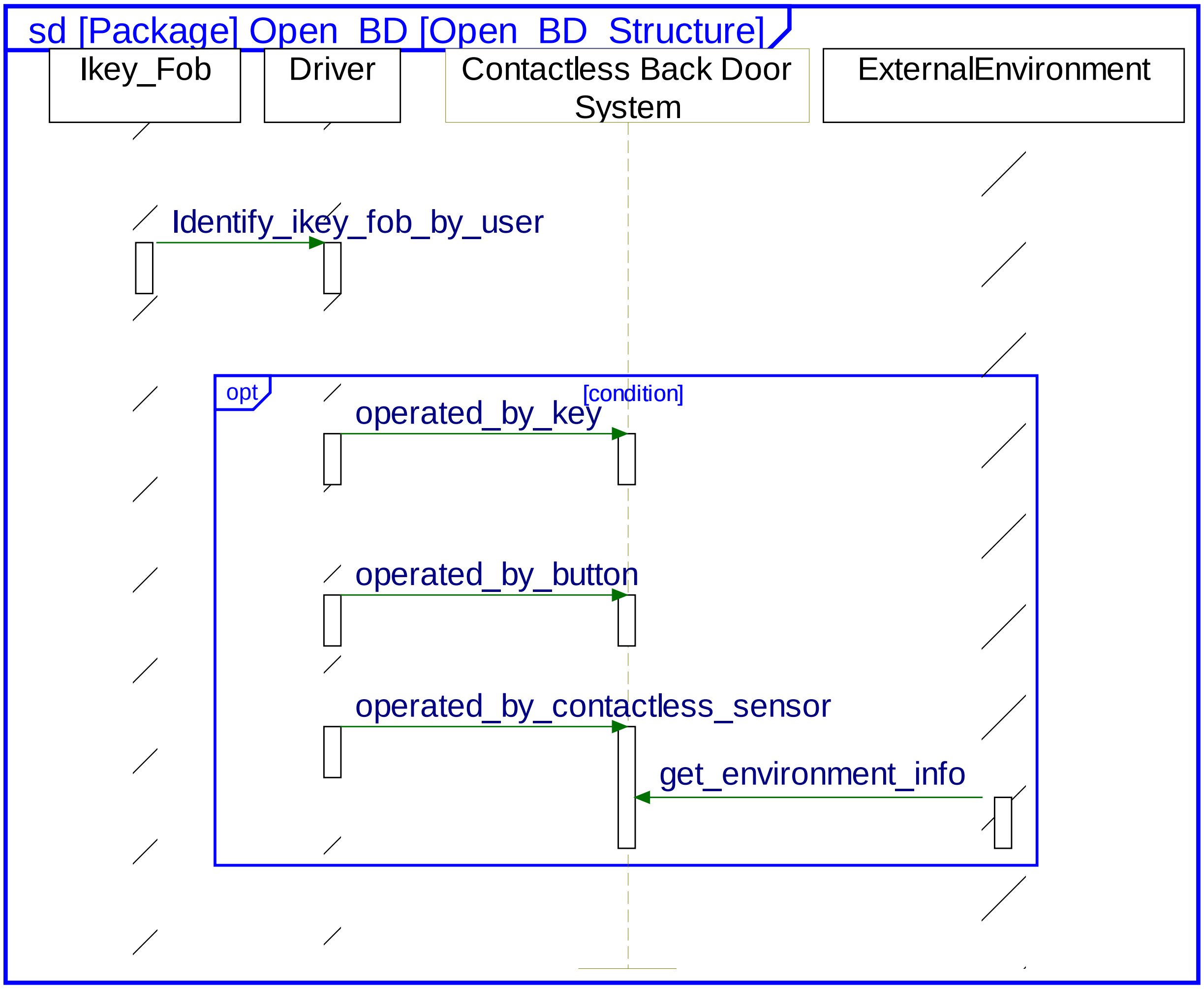


Fig 3.4 Sequence diagram of high level of the "Open BD"

The interactions are described between these four lifelines as an order series of occurrence messages, or the start and the end of behavior executions. The receiving instance should start the execution of a behavior that implements the operation or signal reception referenced in the message once it received a message. For instance, once "Contactless Back Door System" receives the message of "operate by contactless sensor", it will start the function to implement the operation in order to meet the request from “Driver”. Fig 3.5 shows the definition of the arrow of this article. It means that component “A” have a request of implement of function “C” and function “C” will be realized by component “B”.

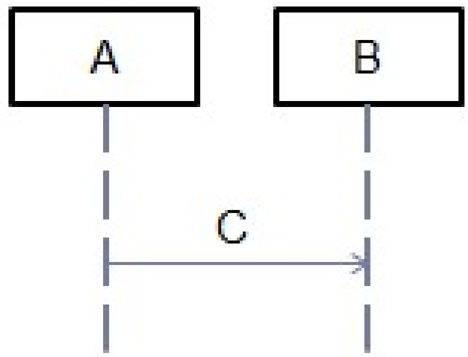


Fig 3.5 definition of the arrow of the sequence diagram

In Fig 3.4, the “Contactless Back Door System” is as a black box, so how the operations are executed is not clear, so we need to flow down to the next level.

With the sequence diagrams, designers can get more information than use case diagrams. Especially the interaction and the messages are clearly illustrated. This is also the merit of the MBSE, designers can view the system from different system models and all of these diagrams can be saved as an evidence to show the discussion result.

What I want to mention is that since the “Contactless Back Door System” analyzed in this chapter is a current existing system, the system components are existed. So we assume that a set of subsystems and components are determined through the requirements engineering process based on stakeholders needs. Therefore, we finish the sequence diagrams directly by using these components in order to visualize the E/E architecture of the “Contactless Back Door System”. Fig 3.6 shows the system components of “Contactless Back Door System”.

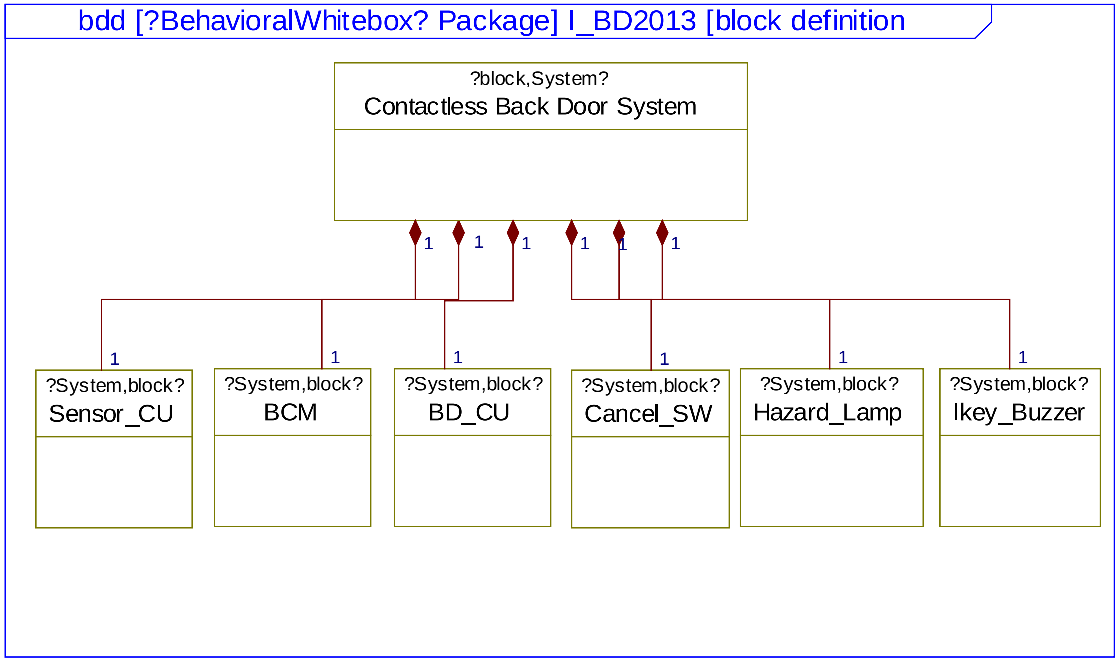


Fig 3.6 The components of the "Contactless Back Door System"

Fig 3.7 shows the function which should be performed included in the "operate by contactless sensor". Though it is also a system level sequence diagram, but it shows the reactions in term of the time and the functions necessary for the "Contactless Back Door System". The sequence diagram shows the scenario which is as follows,

1) Driver set the cancel switch to "OFF" to activate the function of sensor.
2) "Contactless Back Door System" will identify the state of the cancel switch
3) "Contactless Back Door System" will then "identify driver motion" and "identify driver I-key" only in the condition of the state of cancel switch is "OFF".
4) "Contactless Back Door System" will then "unlock back door" only in the condition of both driver motion and I-key are detected.
5) "Contactless Back Door System" will then "open back door" only in the condition of the step of "unlock back door" is finished.

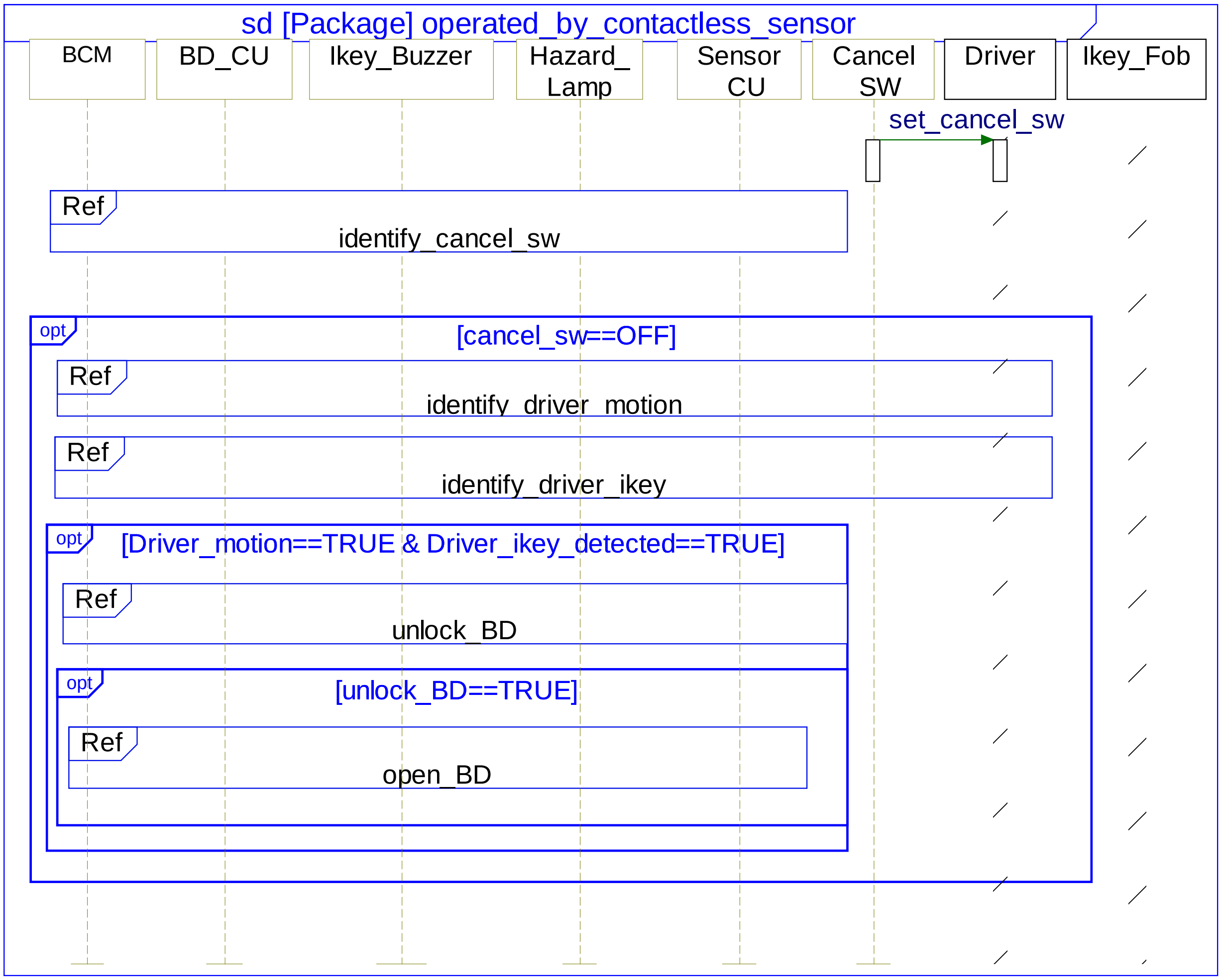


Fig 3.7 the sequence diagram of high level of operating "Contactless Back Door System"

Through Fig 3.7, how the function is implemented is still not displayed. As an example, Fig 3.7 describes the interaction between actors and the back door system to realize the "Operate by contactless sensor" use case in Fig 3.3. It represents the main function of the "Contactless Back Door System". But from this sequence diagram, which function should be implemented by which component is not visualized, for example, which component should realize the function of "Identify user motion" is not clear, so we should flow down again to check more in details about the interaction between the components.

And through the analysis in the next level, even though the system is current existed, it also has good impact on clarifying the function of the system component and the interfaces. For an example, Fig 3.8 shows the earlier edition sequence diagram of “Open BD”, the interactions are then completed through the discussion with the engineers from association automotive company. The final edition is shown in Fig 3.9.

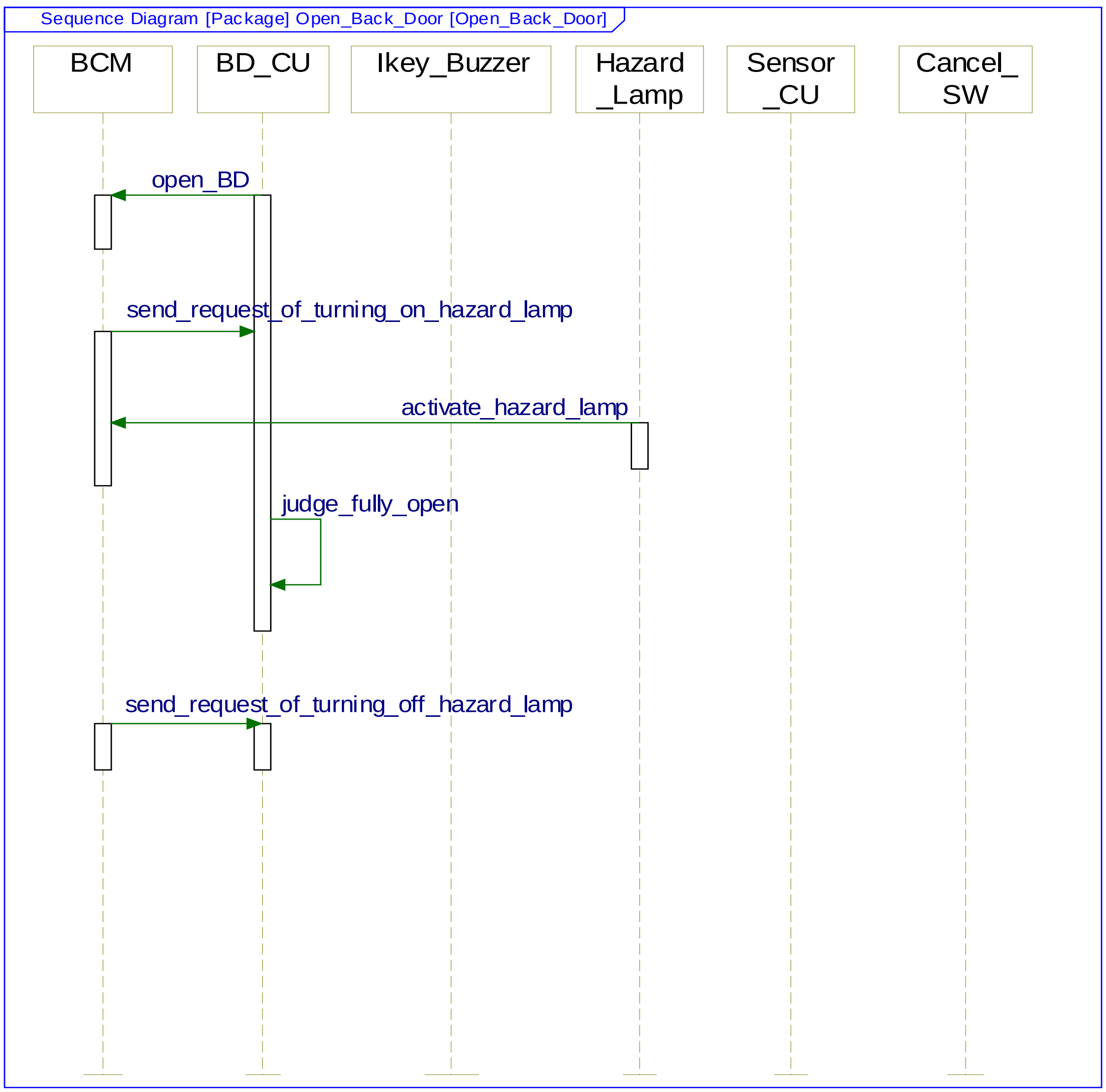


Fig 3.8 Earlier edition of sequence diagram of “Open BD”

It shows that “Open BD” is not the function which should be performed of BCM, but instead, it is one of the functions which belong to BD_CU. And the engineers also tell me that in the company they also draw the diagram which is similar to the sequence diagram, but the problem is not in such details, not only the which function is belonged to which components is not considered but the terms of the message are also in a very simply way. In other word, the engineers they have recognitions of what the system should be performed, but not in that details like drawing the sequence diagram. Therefore, during the process of drawing the sequence diagram, the functions of the components of the “Contactless Back Door System” is becoming more and more clearly.

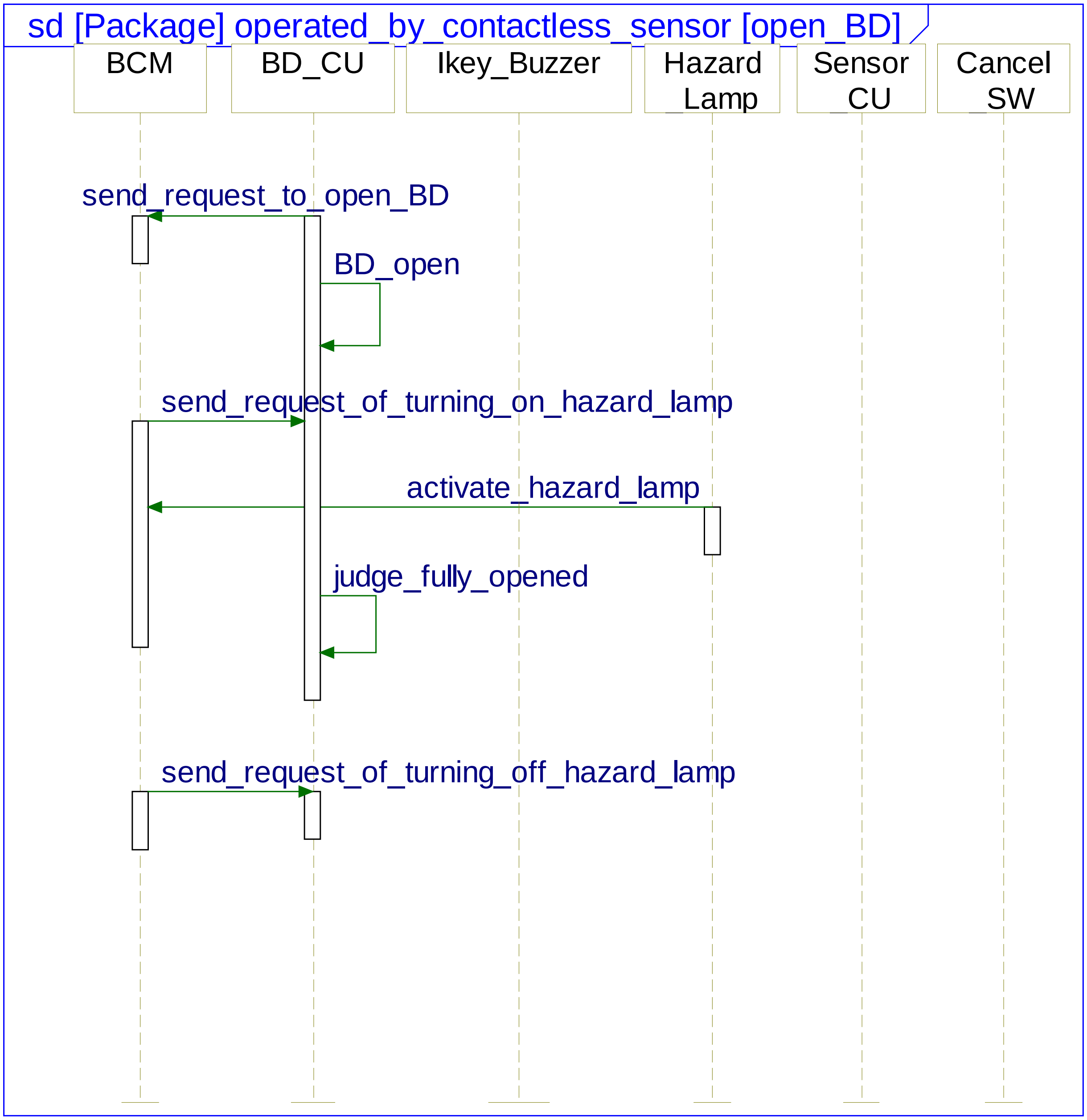


Fig 3.9 Sequence diagram of "Open BD"

Based on the discussion with engineers, the remaining functions "Identify cancel switch", "Identify driver motion", "Identify driver I-key", "Unlock back door" and "Open back door" are analyzed in terms of the reactions between the systems' components. The sequence diagrams realize a behavioral aspect of the "Contactless Back Door System". The sequence diagrams are shown in the Appendix. However, in some cases, it does not mean the interactions between each component are one hundred percent correct. Sometimes, problems are found during the process of analyzing the activity of the system by drawing the activity diagram, in that situation, we should return back and check the sequence diagram again.

3.2.3 Analysis in terms of Activities

In the last section, message-based behavior with interaction of the “Contactless Back Door System” is analyzed and the sequence diagrams are effective for representing discrete types of behavior which focus on control flow. However, continuous types of behaviors associated with the interactions to “Identify cancel switch”, “Identify driver motion” and the other functions of the “Contactless Back Door System” can sometimes be more effective represented with the activity diagram. The activity diagram is similar with the functional flow diagram which has been widely used for modeling system behavior.

Compared with the functional flow diagram, activities provide enhanced capabilities such as to express the relationship between the structural aspects of the system such as parts and blocks. And the activity diagram can also be used to model continuous flow behaviors. Fig 3.10 shows the activity diagram of the “Contactless Back Door System”. And it shows the actions should be taken of the system components to realize “Open by contactless sensor”. Activity diagram are made up of three basic elements which are “Activity node”, “Activity edge” and “Region”. The “Region” is activity partitions or we also call it swim-lanes represent the system components as shown in Fig 3.6, they are “Cancel switch”, “Sensor CU”, “BD CU”, “BCM”, ”Hazard Lamp” and “I-key Buzzer” included in the “Contactless Back Door System”. The “Activity node” in the swim-lanes specifies functional requirements that the system components must perform. The “Activity edge” element has two main types which are “Control flow” (red dotted line in Fig 3.10) and “Object flow” (green line in Fig 3.10). A “Control flow” is used to show the main routes through the activity diagram and connects together one or more “Activity node”. An “Object flow” is used to show the flow of information between one or more “Activity node”. The action is shown in the “Activity node”, and the action here is equal to the function which should be performed by the component which the name is shown in the swim lane.

Fig 3.11 is the earlier edition of the sequence diagram depicts the all actions occurred in the sequence diagrams. Compared the two activity diagrams, we found that the function “Search I-key” is invocated only in the condition that driver hand is detected, but in the earlier edition diagram, it is shown that “Search I-key” and “Detect driver hand” are performed simultaneously once the state of the cancel switch is off. This example tell us, though the functions defined in the sequence diagrams are located in the same swim lanes in both Fig 3.10 and Fig 3.11, the process of the activity should also be carefully considered in order to grasp how the system is correctly running. By drawing the activity diagram, the functions should be performed by which components are all gathered in the corresponding swim lane, so it is very clear to have an overall view not only from the aspect of the whole system, but also from the aspect of each component. It is not only helpful for the system engineers, but also helps the components designer to know how his components are performed in the whole system.

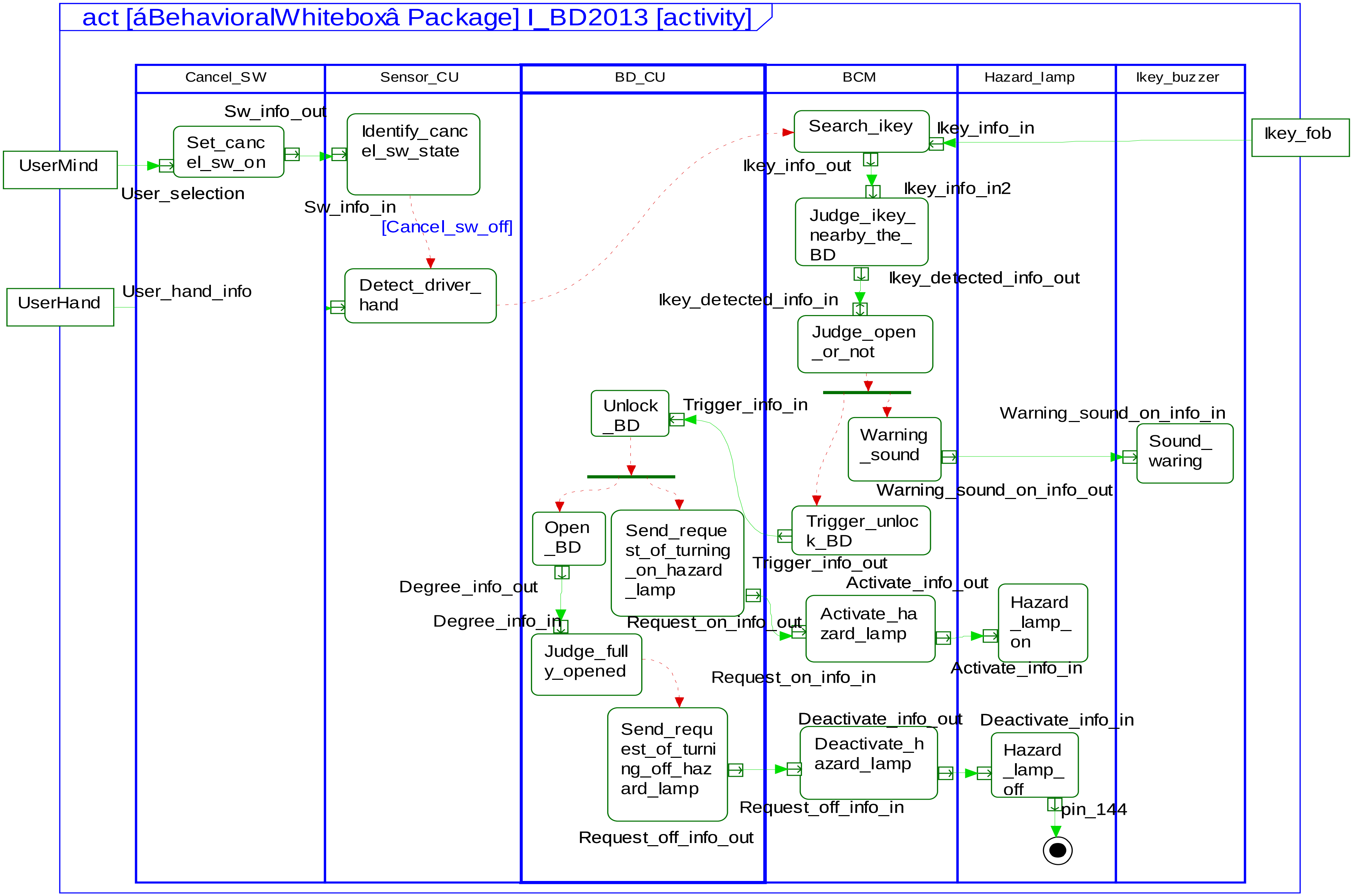


Fig 3.10 Activity diagram of "Contactless Back Door System"

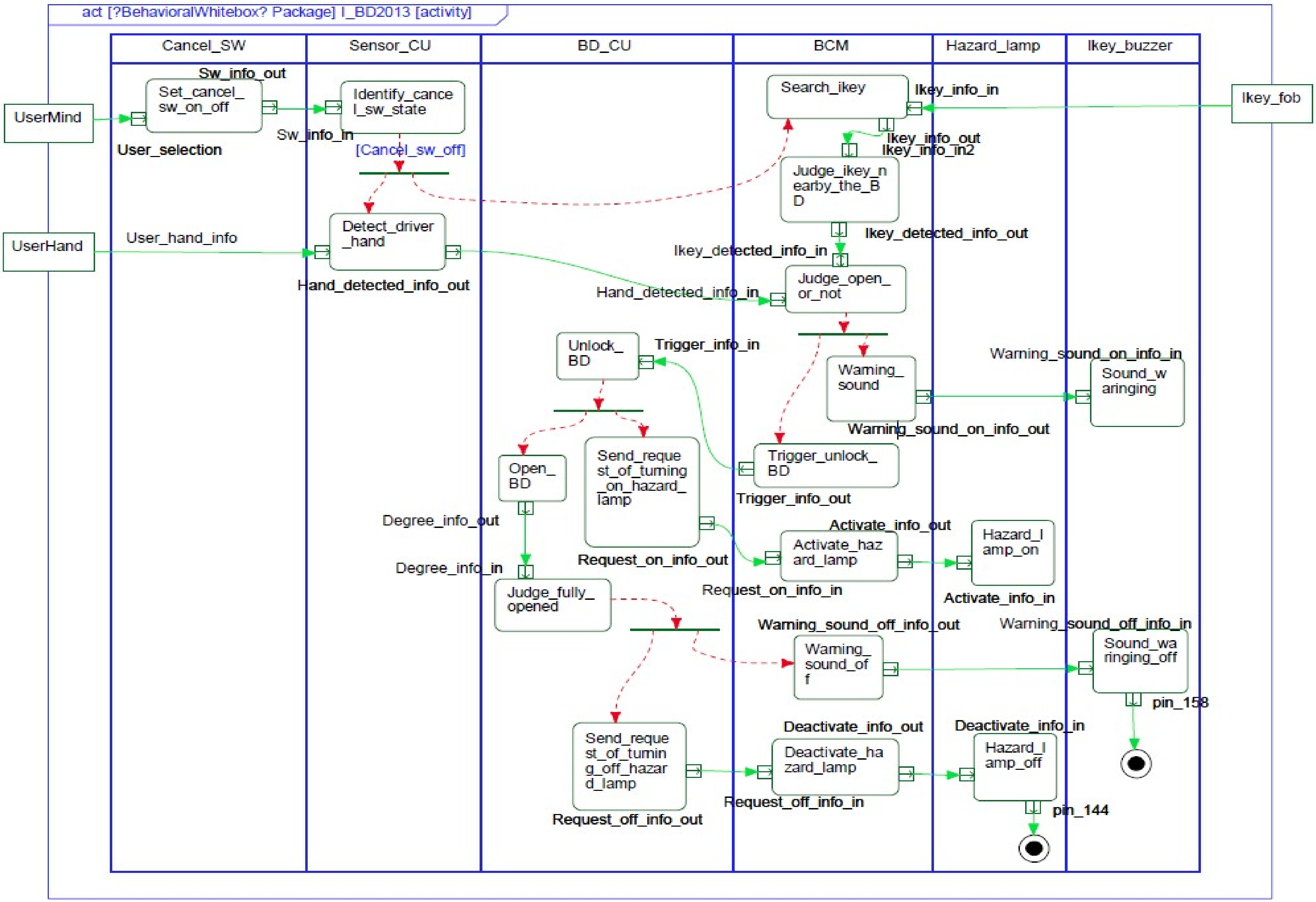


Fig 3.11 Earlier edition of the activity diagram of "Contactless Back Door System"

3.2.4 Analysis in terms of Structure

In last section, we visualize the activity occurred when realize the function of "Operate by contactless sensor", however the structure and the interfaces of the system and the system components are not shown neither in sequence diagrams nor in activity diagrams. In this section, Block definition diagram will be used to describe the structure of the system. Block is defined as a basic structural element in SysML whose aim is to represent any type of components of the system. Blocks can be assembled to form architecture that represents how different components can co-exist in the system.

Block definition diagram is the very easy way to show the structure of the system. It can be used to represent the system decomposition and their relationships. Features such as properties and operations of the components or the blocks can be displayed in the block definition diagram. Block definition diagrams are made up of two basic elements which are blocks and relationships.

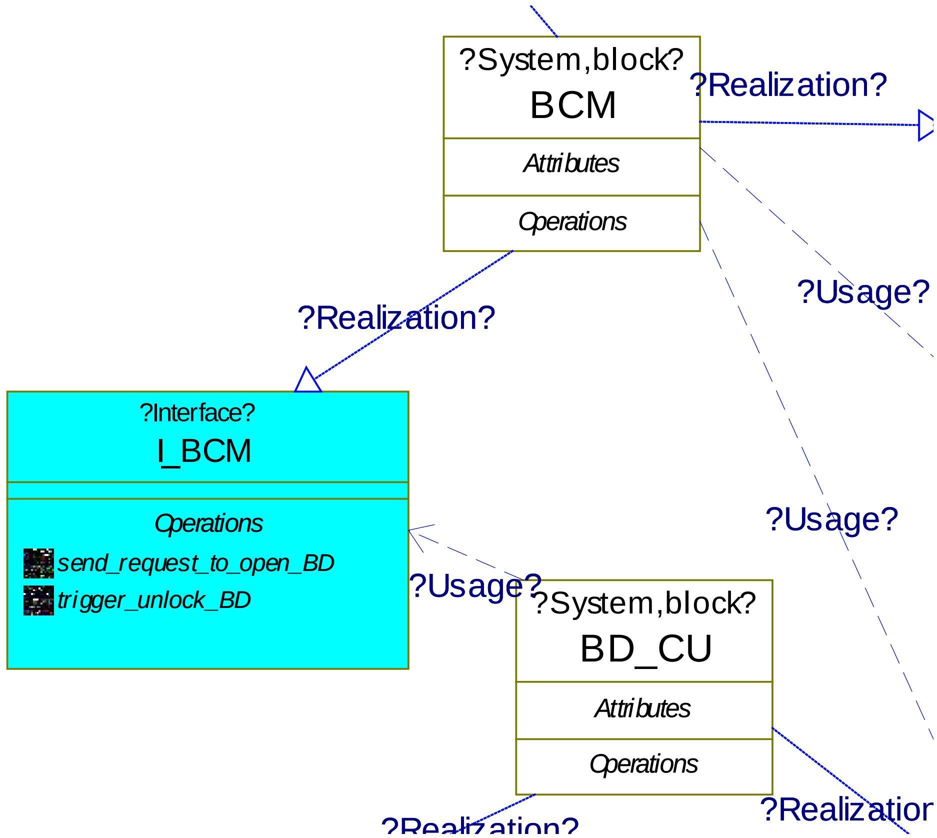


Fig 3.12 Expansion of the block definition diagram

Fig 3.13 shows the level01 block definition diagram of the "Contactless Back Door System", the interfaces are focused on and shown as the blue box. The operations shown in the interfaces are the operations which should be implemented once the request is arrived from the other component. For example, Fig 3.12 shows part of the block definition diagram, it shows the relationship between the BCM and the BD_CU. The interface specifies the set of behavioral features required by BD_CU, in other words, the relationship between the interface and the BD_CU is "usage". And the behavioral features are provided by the BCM, in other words, the relationship between the interface and the BCM is "realization". Based on this example, return back to the Fig 3.13, we can now read the relationships between the components and the interfaces. The block definition diagram provides another aspect of viewing the system. It is important to decide what kind of interface is necessary in the high level of the development processes. If the interfaces are not clarify such as in the Fig 3.13, then in the next level or the level after next, the design of interfaces of the system might be in confusion. On the other hand, now we analyze the interfaces which are necessary in a high level of the architecture design, the block definition diagram itself might be a documentation which provides the traceability of the interface between the components when designers flow down to next several levels.

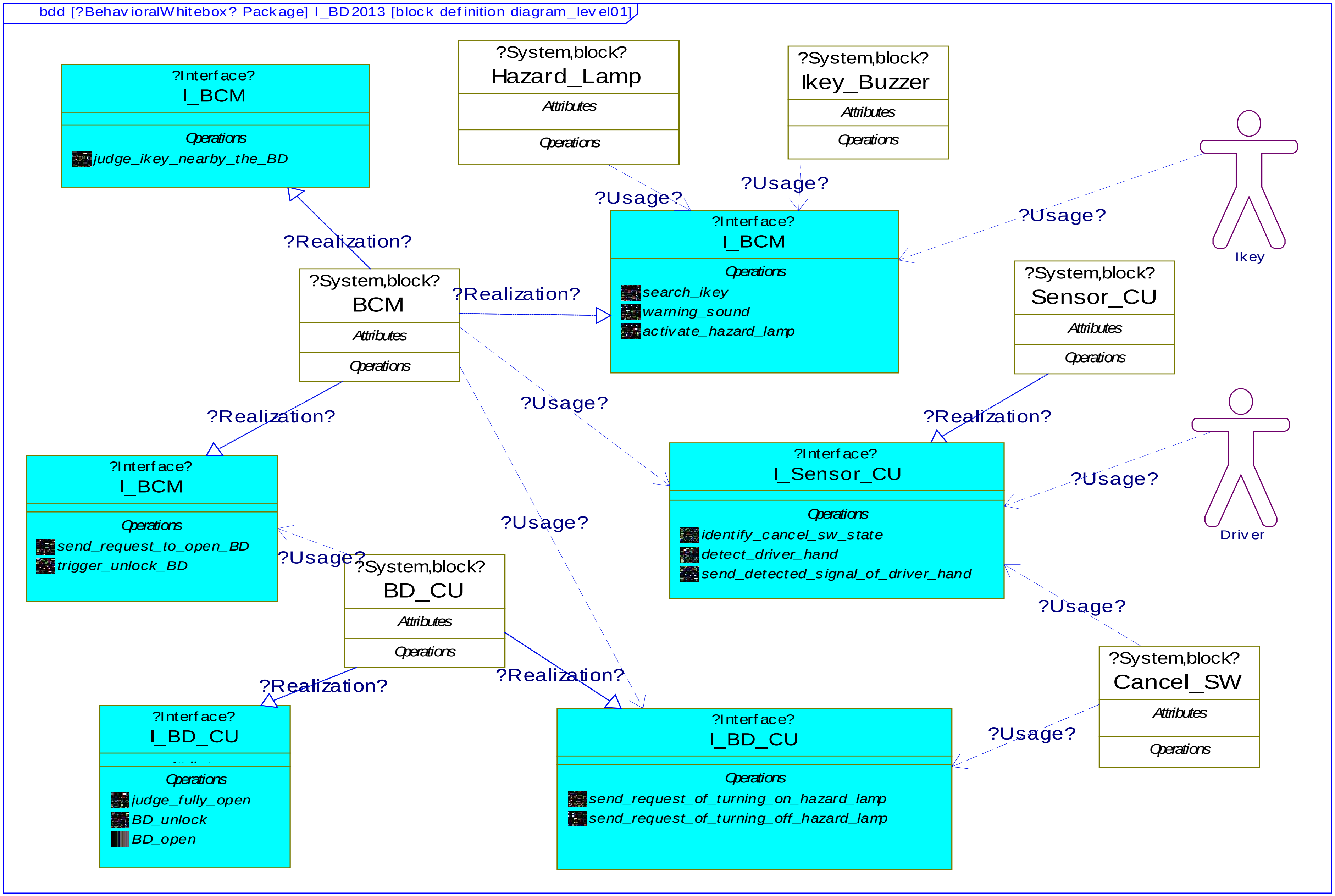


Fig 3.13 Level01 Block Definition Diagram of the "Contactless Back Door System

4. Architecture of "One-Step Back Door System"

4.1 Context Analysis

In this chapter, a design process of the architecture of a new back door system by using SysML is discussed. The new back door system is named as "One-step back door system".

In this section, the requirement and the scenario of operating the "One-step back door system" is analyzed. At first, brainstorming is used to get new and innovative ideas of how the future back door system should be performed to meet the needs of the stakeholders. And three main requirements were defined which shall be performed in the new back door system. These three main requirements specify a capability or condition that the new back door system should be satisfied. The requirements of the new back door system are shown as follows:

1) Customers with I-key are able to open and close the back door without using their hands from the position where he or she is approaching to the backdoor.
2) Back door open and close features with one action, natural motion, speedy and without dirtying the customers' hands and clothes.
3) Back door shall show the "Omotenashi" attitude to the customers.

And we divide the main three requirements into five groups from the perspective of "Operation", "Omotenashi", "Safety", "Security" and "Constraints" in order to describe the requirements in much more details. Moreover, "Operation" is divided into "User friendliness", "Environment" and "Speedy". And five more requirements are contained in "User friendliness" to make the requirement of the "User friendliness" in more details. In order to show the hierarchy of the requirements and to make the relationship becomes more clearly. In SysML, containment can be used to represent how a compound requirement can be divided into several simpler requirements without adding or changing their meaning.

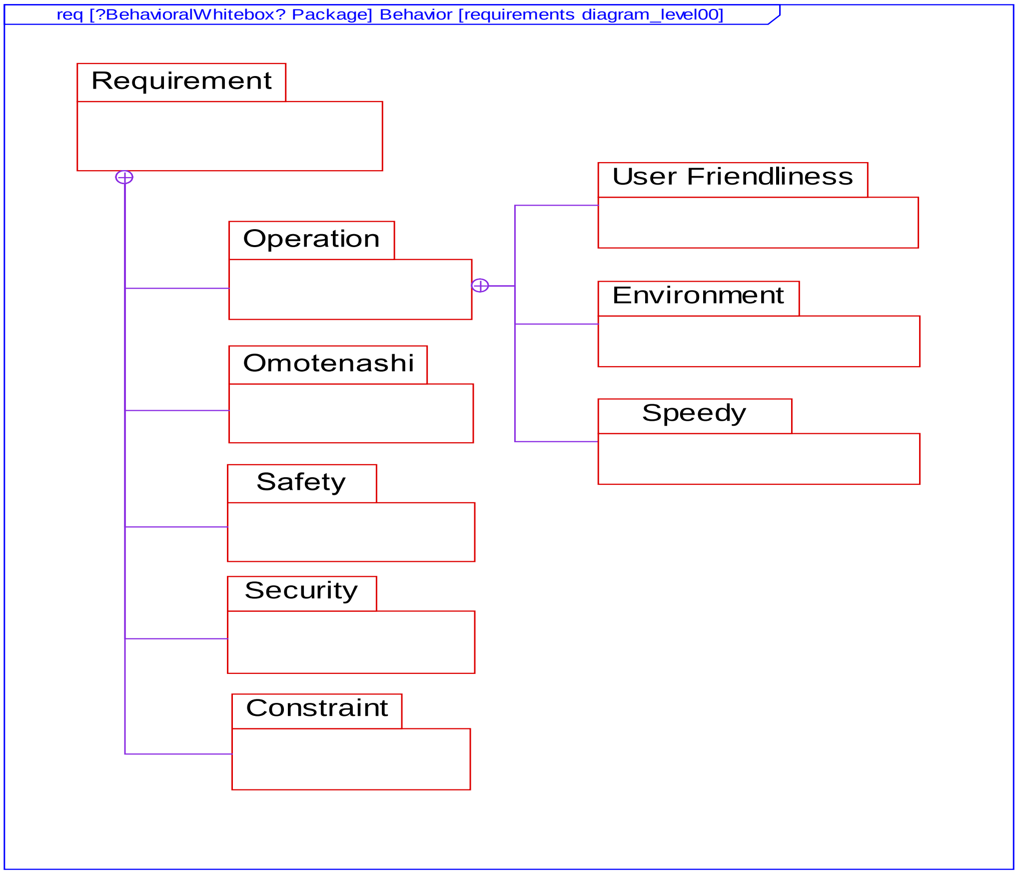


Fig 4.1 Package structure for organizing requirements

Fig 4.1 shows a requirement diagram with simple containment hierarchy and the whole view of the relationship of the system's requirement. The "Requirement" package shows a top-level specification that serves as a container for all other stakeholder generated requirements. The partitioned parts of the compound requirement help establish full traceability and show how individual requirements are the basis for further derivation.

However, we noticed that the requirements are described with the very natural sentences. It is easy for people to understand, but on the other hand, it is easy to make misunderstanding among engineers due to the way of thinking of everybody is different. Therefore, each individual requirement should be expressed in a clear and unambiguous terms, which clear enough for the engineers to implement a system that can really meets the stakeholders' needs. So, if the terms are too long for engineers to make a clear decision of what functional requirement is necessary for the system, the terms should be cut into several short terms.

For instance, the requirement of the "User friendliness" is "Customers can open and close the back door from the side of the back door which is intuitively recognizable with natural motion and one action without using their hands, and dirtying their clothes." This term is difficult for the engineers to determine which functional requirement should be performed. Therefore, four individual requirements are partitioned as follows:

1) UF1: Customers can open and close the back door from the side of back door.
2) UF2: Customers can open and close the back door with natural motion and one action without dirtying their hands and their clothes.
3) UF3: Customer can open/close BD from the back of BD without stepping back from BD moving area.
4) UF4: Customers can open and close the back door with one action even the car is locked.
5) UF5: Operation of back door should be intuitively recognizable.

(UFX means No. X requirement of "User friendliness".)

By analyzing the requirements of the other packages shown in Fig 4.1 using the process discussed above, the requirements are organized in the requirement diagram of the context level which shown as Fig 4.2.

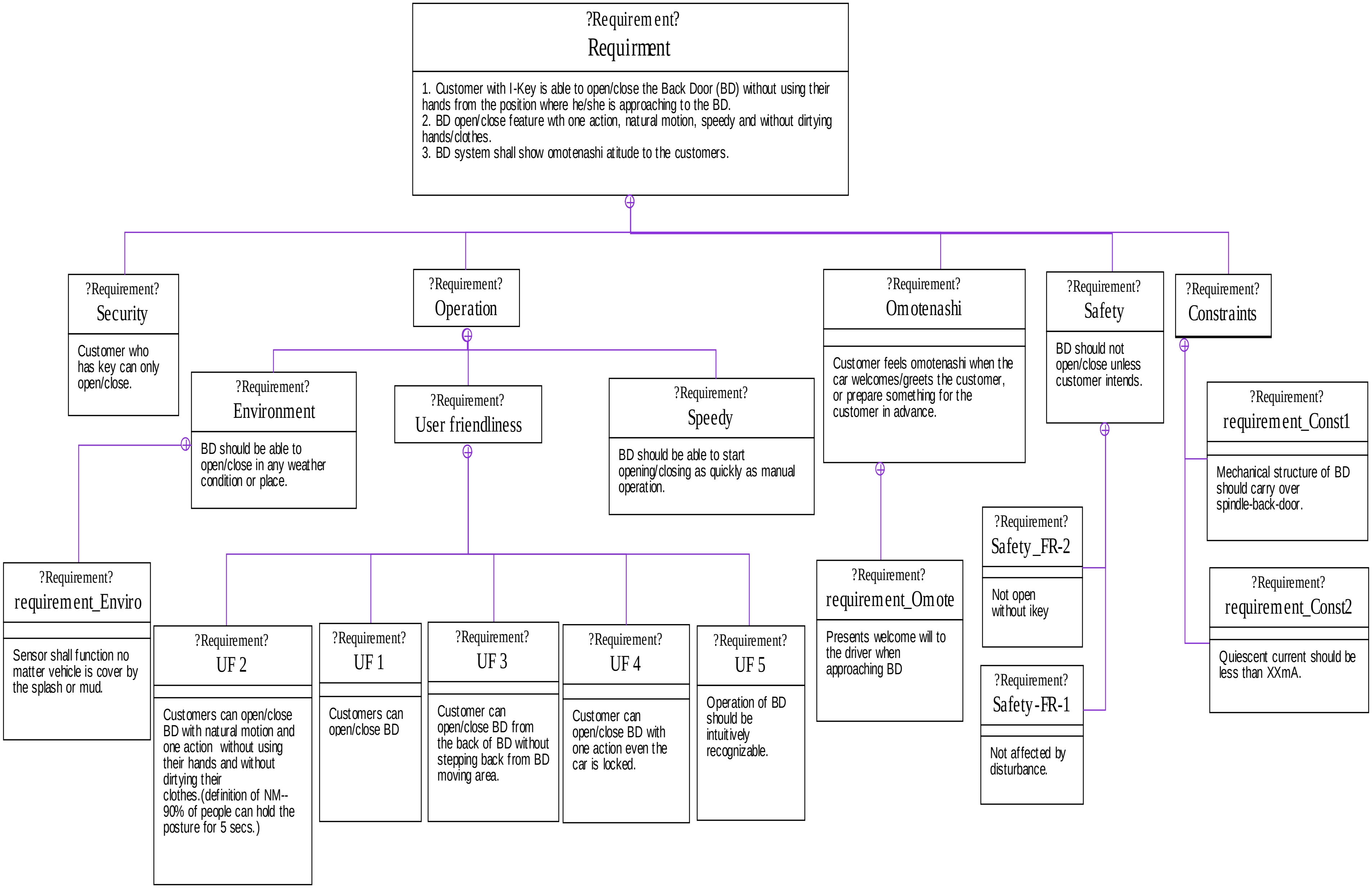


Fig 4.2 Requirement diagram of context level of "One-Step Back Door System"

At the same time, based on the analysis of the requirements discussed above, a roughly idea is carried out, which driver can open and close the back door by using their foot which meet the requirement of “Customers with I-key are able to open and close the back door without using their hands from the position where he or she is approaching to the backdoor.” At this time we give a temporarily name of the new back door system which is “One-Step Back Door System”. And a scenario will be show as follows with several figures are illustrated to show the scenario.

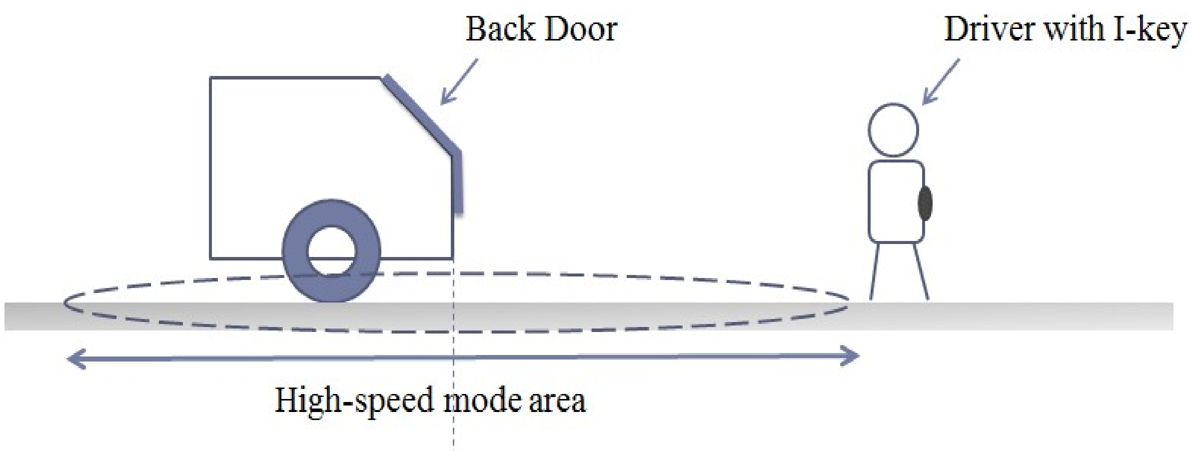


Fig 4.3 No.1 scenario graph of “One-Step Back Door System”

At first, we consider the scenario of “open back door”, there are two areas which are the area of “High-speed mode area” and the area out of it. The “High-speed mode area” is the area once the driver with I-key is detected, the “One-Step Back Door System” will start to perform. And based on the requirement of “Presents welcome will to the driver when approaching BD” of “Omotenashi” and the requirement of “UF5: Operation of back door should be intuitively recognizable” of “User friendliness”. LED light is supposed to be used in order to show the “welcome will” and realize the requirement of “intuitively recognizable”.

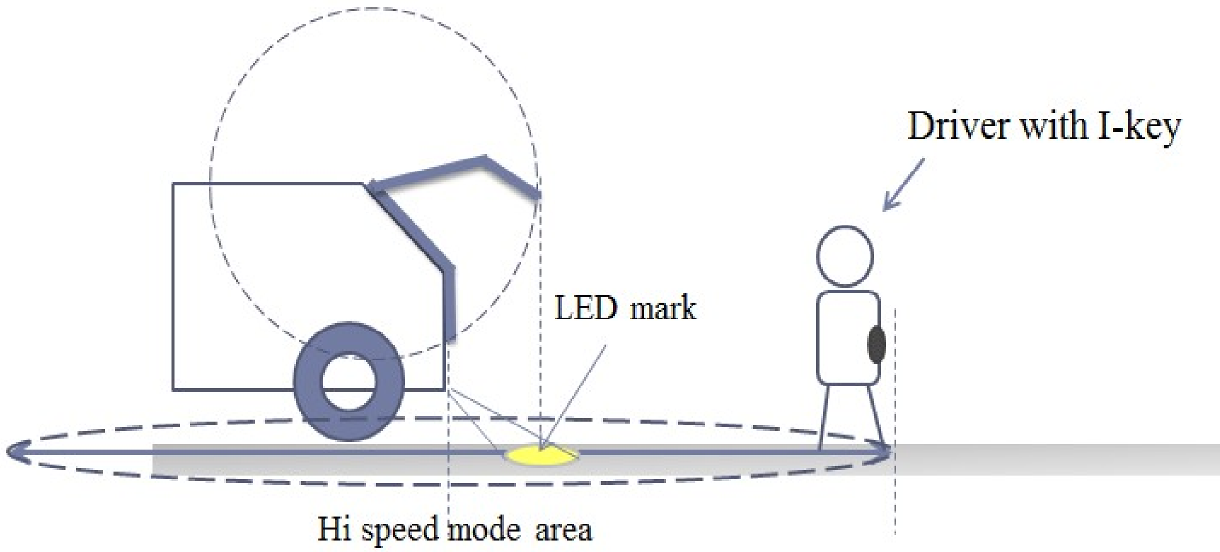


Fig 4.4 No.2 scenario graph of "One-Step Back Door System"

Once the driver with I-key entered in the "Hi-speed mode area", LED light will be turned on and the ground will reflect the LED light to show the welcome will to the driver, and at the same time, directly show the driver where to operate the back door. And the center point of the LED mark shall be on the outside of the opening and closing trajectory of the back door in order to meet the requirement of "UF3: Customer can open/close BD from the back of BD without stepping back from BD moving area".

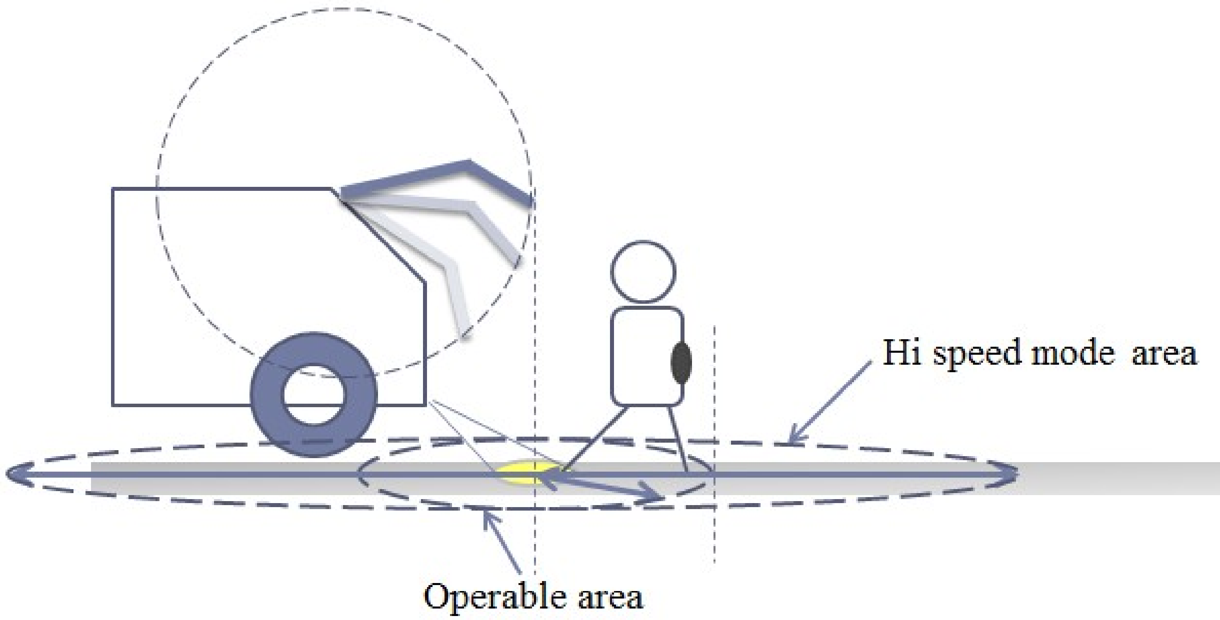


Fig 4.5 No.3 scenario graph of "One-Step Back Door System"

And the "One-Step Back Door System" has the function to judge the position of the I-key, it meets the requirement of "BD should not open/close unless customer intends" of the "Safety" shown in Fig 4.2. If driver with I-key does not has the willingness to open the back door, he or she just walks through the "Hi-speed mode area", then LED will not blink, only I-key is detected in the "Operable area", then the LED will blink to show deeper welcome will. And if driver step his or her foot on the LED mark, then sensor will be used to detect the foot of the driver, if foot is detected, then the back door will open automatically. In other words, the "One-Step Back Door System" will open the back door unless both the driver's foot and the I-key are detected in the "operable area". And through the discussion with the engineers, sonar is assumed to use as one of the methods to realize the function of the sensor.

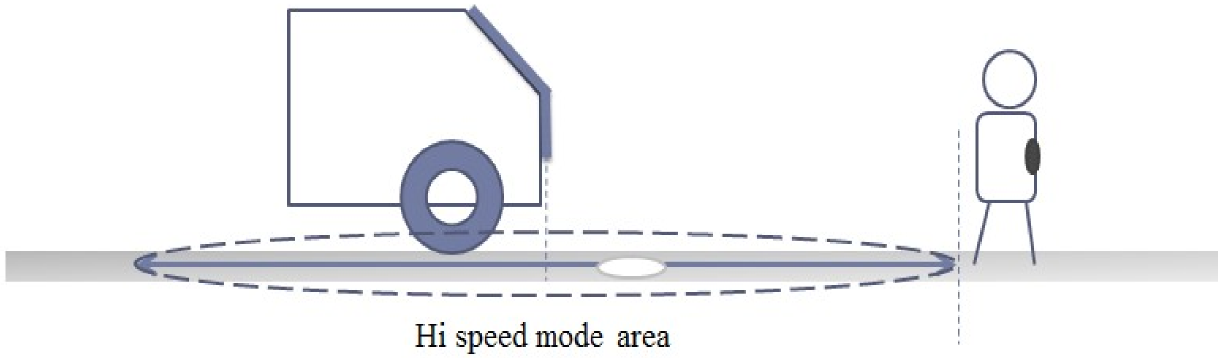


Fig 4.6 No.4 scenario graph of "One-Step Back Door System"

And as mentioned above, if driver with I-key entered in the "Hi-speed mode area", but he or she does not want to open the back door and leave the area. In this situation, if I-key is not detected in five seconds, the LED light will be turn off. And if the Driver with I-key is detected in the "Hi-speed mode area" for over one minute, but he or she does not step on the LED mark, then the LED light will also be turn off. These two scenes meet the requirement of save the energy of the system.

Beyond the analysis discussed above, use case is also recommended to clarify the functionality of a system in terms of how it is used to achieve the goals of its various users. Fig 4.7 shows the use case in the high level00. The system boundary is the "system of interest", in other words, it is the E/E architecture of the "One-Step Back Door System" which we want to develop this time. There are five actors outside the system boundary. The "Vehicle 2013" represents the existing vehicle which is discussed in Chapter Three. The other four actors are "Driver with I-key", "Occupants without I-key", "External Environment" and 'Ground". In this article, we will focus on the two most popular and normal scenarios which are the driver who with I-key to open or close the back door. And in "External Environment", we just consider about the "ground". These stakeholders are the main actors included in the scenario discussed above.

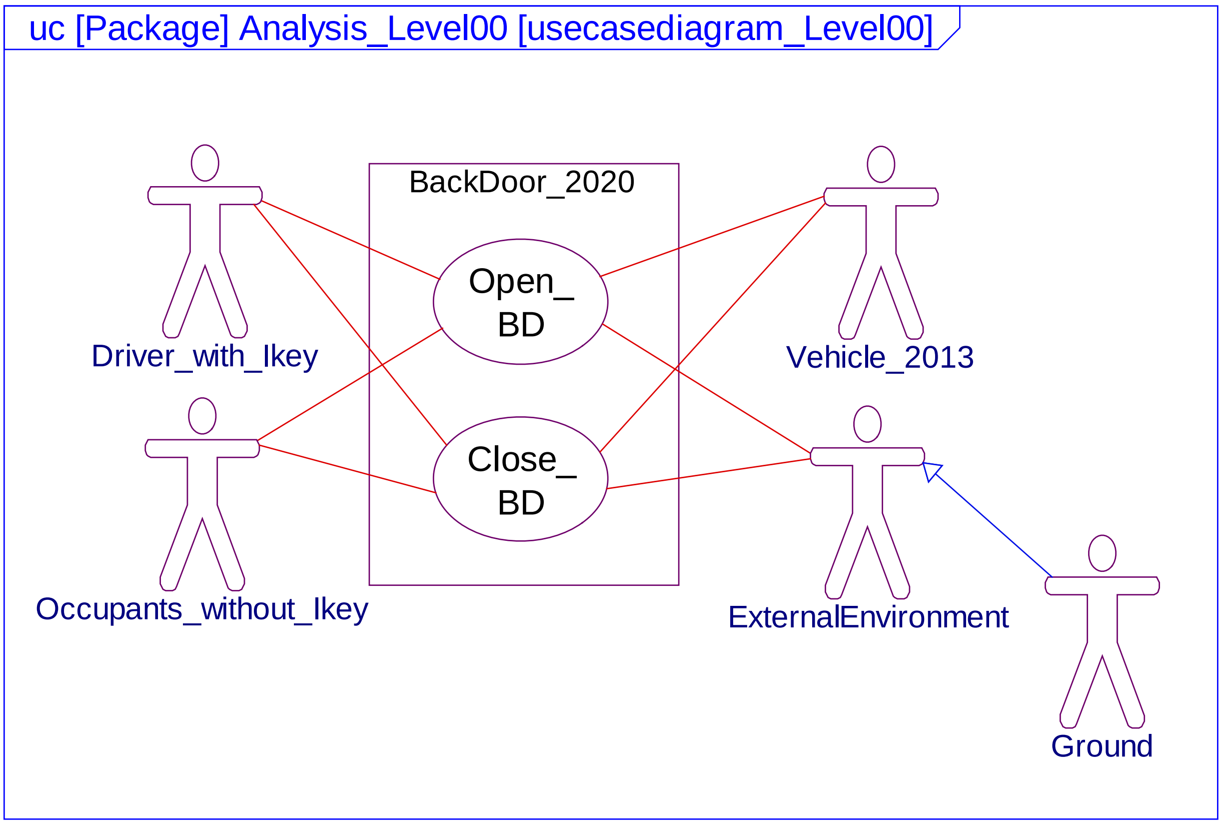


Fig 4.7 use case diagram of context level

Based on the requirement diagram, scenario diagrams and the use case diagram, the functions which should be performed by the "One-Step Back Door System" are gradually picked up. But as shown in Fig 4.7, at this time, the system is still a black box, what components are necessary and what functions should be realized by what components are still not clarified. In order to design the E/E architecture of the "One-Step Back Door System", we should flow down to the next level to solve the questions just mentioned.

4.2 Functionality Analysis and Synthesis

Based on the discussion of last section, we assume that “Sonar system” is necessary which functions as the sensor to detect the foot of the driver. And LED is also required to show the welcome wills to the driver. And a component which can detect the I-key position is also indispensable, we assume it named as “Movement Detector”. And the system we want to develop this time is focus on “One-Step”, so the function such as “Unlock back door”, “Open back door” are realized by the existing vehicle which named “Vehicle2013”, and it is already discussed in Chapter Three.

As a result, “Sonar System”, “Movement Detector” and “LED” are assumed as the components of the “One-Step Back Door System” at this stage, and some of the functional requirement will still be satisfied by the existing vehicle which shown as “Vehicle 2013” in the requirement diagram.

And a high level (level00) sequence diagram is shown in Fig 4.8 to show a top level behavioral of the “One-Step Back Door System”. There are several points what I want to mention here to show how the sequence diagram is completed by changing revising several times. It also shows the process of thinking of the system.

At first, in fact, the “Ground” is not considered in the sequence diagram since all the attention is paid to the “One-Step Back Door System” itself. It brings a bad impact that if without “Ground”, there is no objects can reflect the Sonar and LED light, and without the ground, the other factors which might have impact on the LED will also be ignored such as the reflection of the sunlight. It will influence the parametric design of the LED.

Second point, we correct the term of “receive driver approach BD” to “switch to Hi-speed mode”, during the discuss with engineers, it came clear that movement detector (MD) will “switch to Hi-speed mode” once “receive driver approach BD”, “receive driver approach BD” just represent the condition when MD will “switch to Hi-speed mode”. Therefore, the function must performed of MD is “switch to Hi-speed mode”. The term “switch to Hi-speed mode” make clear the function should be performed by the MD, and it also clarify in what condition, LED and Sonar should be turn on. It has a big influence on the activity diagram.

Third point, “detect I-key movement” is added, the reason it did not pick up at the beginning is that “stepping on light area” is naturally considered as the next step of “driver recognize LED light”, therefore, “detect I-key movement” which is one of the main functions of the “One-Step Back Door System” is ignored. And with adding “detect I-key movement”, it let us clarify the function of the system and it also refers to the security issue. Moreover, it let us have the chance to continue think about two scenarios which are whether the key is in the “Hi-speed mode area” or in the “operable area”.

Moreover, there is no message “detect driver foot” and “judge open or close back door”, the reason for it is we take it for granted that backdoor will be unlocked once driver foot is detected. After noticed this, “send request of unlocking BD” is added, but this is also a little bit ambiguous, how the request be sent? In fact, it just transfers the signal, then let vehicle 2013 to decide whether to open or close. Therefore, “send detected signal of driver foot” is the really function which “One-Step Back Door System” should perform.

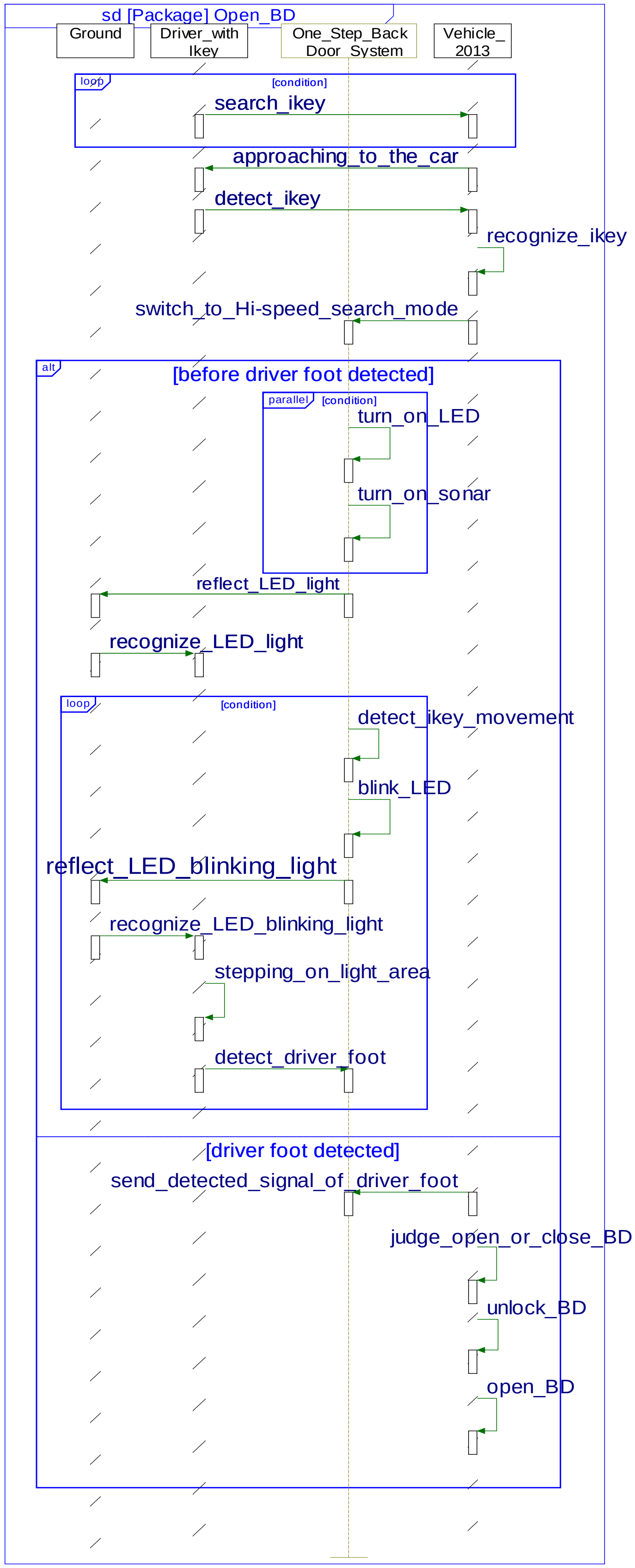


Fig 4.8

Sequence diagram of level01 of "One-Step Back Door System"

During the process of completing the sequence diagram, use case diagram, requirement diagram and scenario graphs are all considered at the same time. By thinking of the system from these different aspects, the functional requirements of the “One-Step Back Door System” are clarified as follows:

1) Switch to Hi-speed mode
2) Turn on LED
3) Turn on Sonar
4) Detect I-key movement
5) Blink LED
6) Detect driver foot
7) Send detected signal of driver foot

Then we will flow down to the next level (level01), 1)~7) can be seen as seven low level use cases, and what we should do is to clarify the interactions among the system components and how the functions are performed by the system components to realize this seven use cases to meet the requirements of user friendliness and “Omotenashi”. The sequence diagrams are shown in the Appendix. There are also many noticed points during defining the functions of the system components by using sequence diagrams. The notice points will be discussed in the Chapter Five. And the “Close BD” use case is also analyzed abide the process of analyzing the “Open BD” use case. The sequence diagrams are shown in Appendix.

Since the functions what should be performed of each system are defined by using the sequence diagrams. In order to have a whole view of how the “One-Step Back Door System” is performed, the functions are allocated to each component composing the system by using the activity diagram shown in Fig 4.9

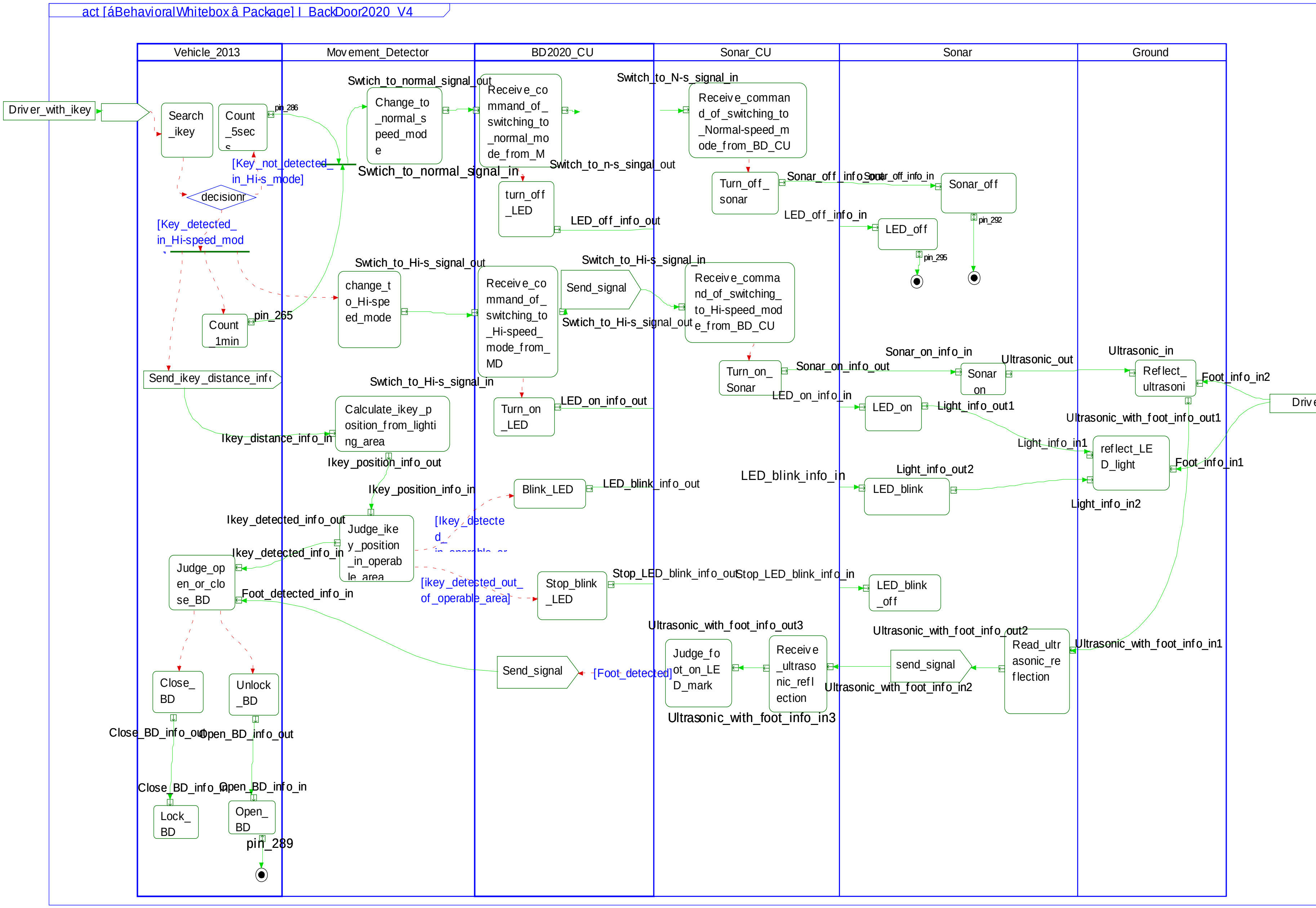


Fig 4.9 Activity diagram of "One-Step Back Door System"

Fig 4.9 shows how the "One-Step Back Door System" is performed to realize both the "Open BD" use case and the "Close BD" use case.

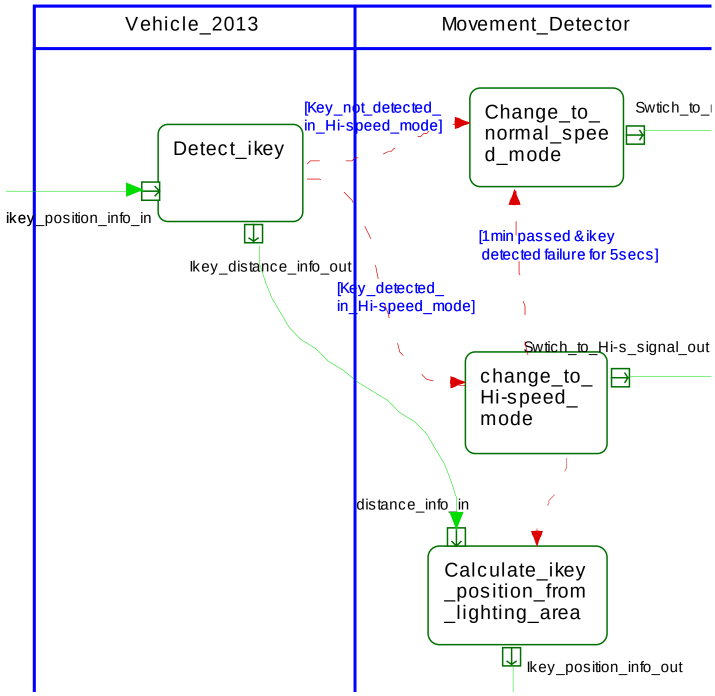


Fig 4.10 part of the earlier vision of the activity diagram of "One-Step Back Door System"

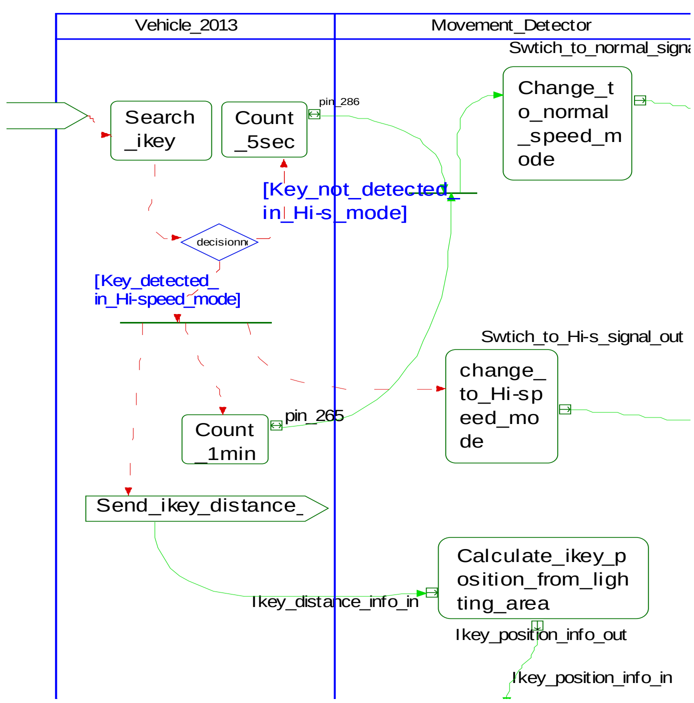


Fig 4.11 part of the final vision of the activity diagram of "One-Step Back Door System" (1)

What I want to mention is during the process, we should ensure consistency between models. One of the most powerful aspects of the SysML is the ability to check the consistency between diagrams, which is often glossed over. In order to give a good level of confidence in the models, these consistency checks are essential.

Fig 4.10 and Fig 4.11 shows the part of in what situation, the “Movement Detector” will be changed to the “Hi-speed mode” and in what situation it will be changed to “Normal speed mode”. The corresponding part is analyzed in sequence diagram of “switch to Hi-speed mode” shown in appendix.

In the earlier vision, the switching of the state is not clear. With the discussion, it carried out a problem of what component should perform the function of “counting numbers”, this is also very important to the designers. The functions can be clarified by using different behavioral diagrams from various aspects. And from this example, we find that the problem of interaction between “Vehicle 2013” and “Movement Detector” is easily found by analyzing the system by using activity diagram, which might not easily found just by analyzing the sequence diagram. Therefore, analyzing the system by using different behavioral diagrams is very important and useful.

Moreover, we found that back door control unit (BD2020 CU) and the sonar control unit (Sonar CU) can be synthesized by checking the activity diagram. Fig 4.12 shows the interactions between “BD2020 CU” and “Sonar CU”.

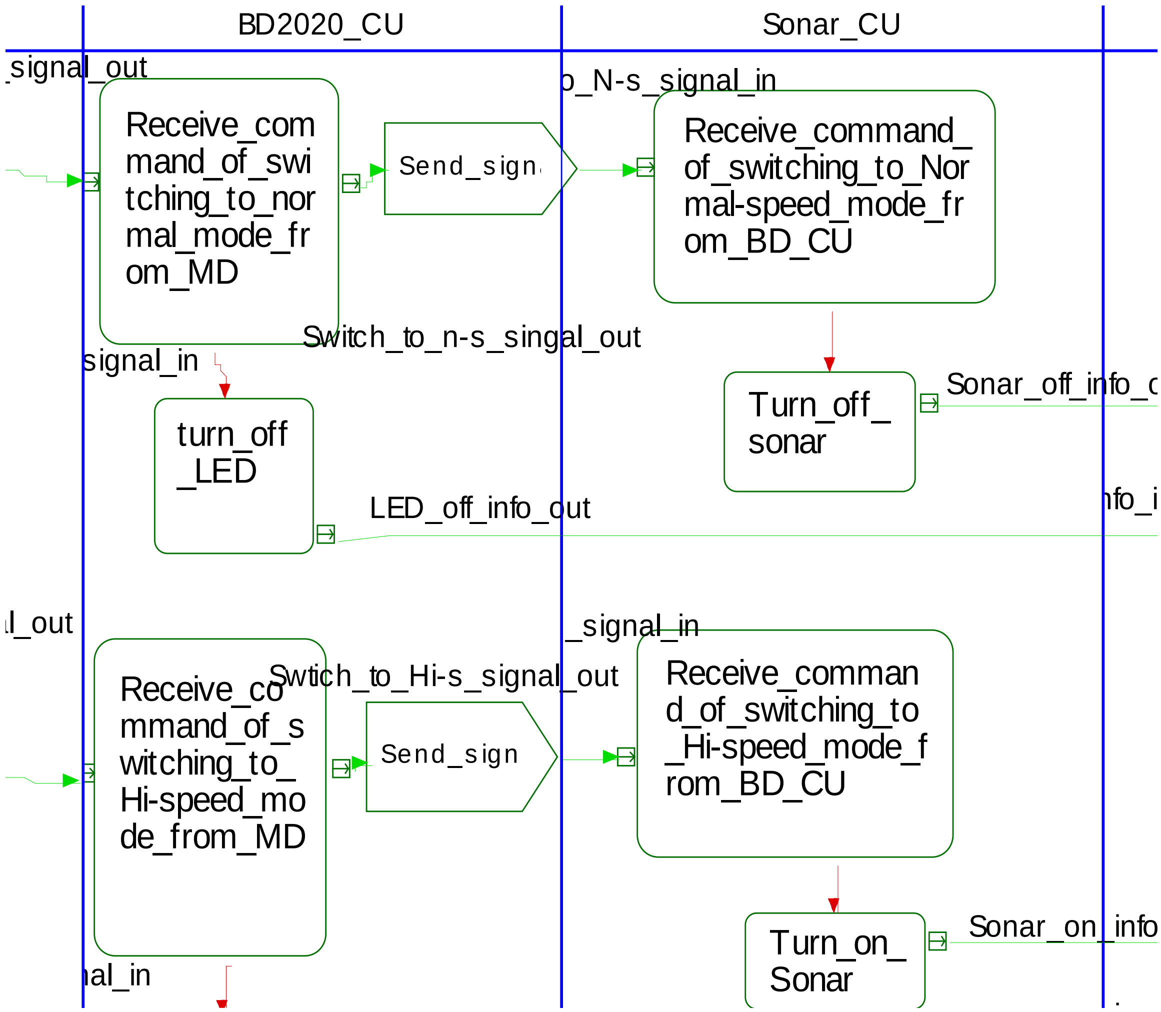


Fig 4.12 part of the final vision of the activity diagram of "One-Step Back Door System" (2)

From Fig 4.12, we can notice that the interactions between "BD2020 CU" and "Sonar CU" are "switch to Hi-speed mode signal" are "switch to Normal speed mode signal". And if we check the function allocated in corresponding swim lane, we find that two of the functions should be performed by "BD2020 CU" are "Receive command of switching to Normal mode from Movement Detector" and "Receive command of switching to H-speed mode from Movement Detector". And "Sonar CU" performs the two functions of "Receive command of switching to Normal mode from BD2020 CU and "Receive command of switching to H-speed mode fromBD2020 CU" once the signal is received. The signals are the same kind and the functions are also the same kind. Therefore, we see a possibility of synthesizing the "BD2020 CU" and the "Sonar CU".

Interfaces are organized in a much more clear way by using the block definition diagram. Fig 4.13 shows the block definition diagram of "One-Step Back Door System". The system of interest is included in the boundary which named "One-Step Back Door System". And the actors outside the boundary are also shown in the block definition diagrams to clarify the interfaces between user, and the "One-Step Back Door System", and that among the internal system. From the block definition, we noticed that the interfaces of "Movement Detector" are all with the "Vehicle 2013". In order words, all the functions realized by "Movement Detector" meet the request of "Vehicle 2013", and the only interface between "Vehicle 2013" and "BD2020 CU" is "receive signal of foot detected". And by checking the activity diagram shown in Fig 4.9, we can also notice that since the SysML provide the ability to check the consistency between diagrams. Based on the discussion mentioned above, there is a possibility of synthesizing the "Vehicle 2013" and the "Movement Detector". Though the feasibility should be evaluated by a deeper discussion between the components designer by using other ways, but at least, the block definition diagram of Fig 4.13 shows the possibility of synthesizing the system components in the early stage of the development process on the system level.

As a conclusion of this section, the two use case "Open BD" and "Close BD" are descripted by the behavioral diagrams such as sequence diagrams, activity diagrams in order to clarify the functional requirement which then can be refined the requirement. And the system structure of the "One-Step Back Door System" is descripted by the block definition diagram, and we find that some of the system components can be synthesized.

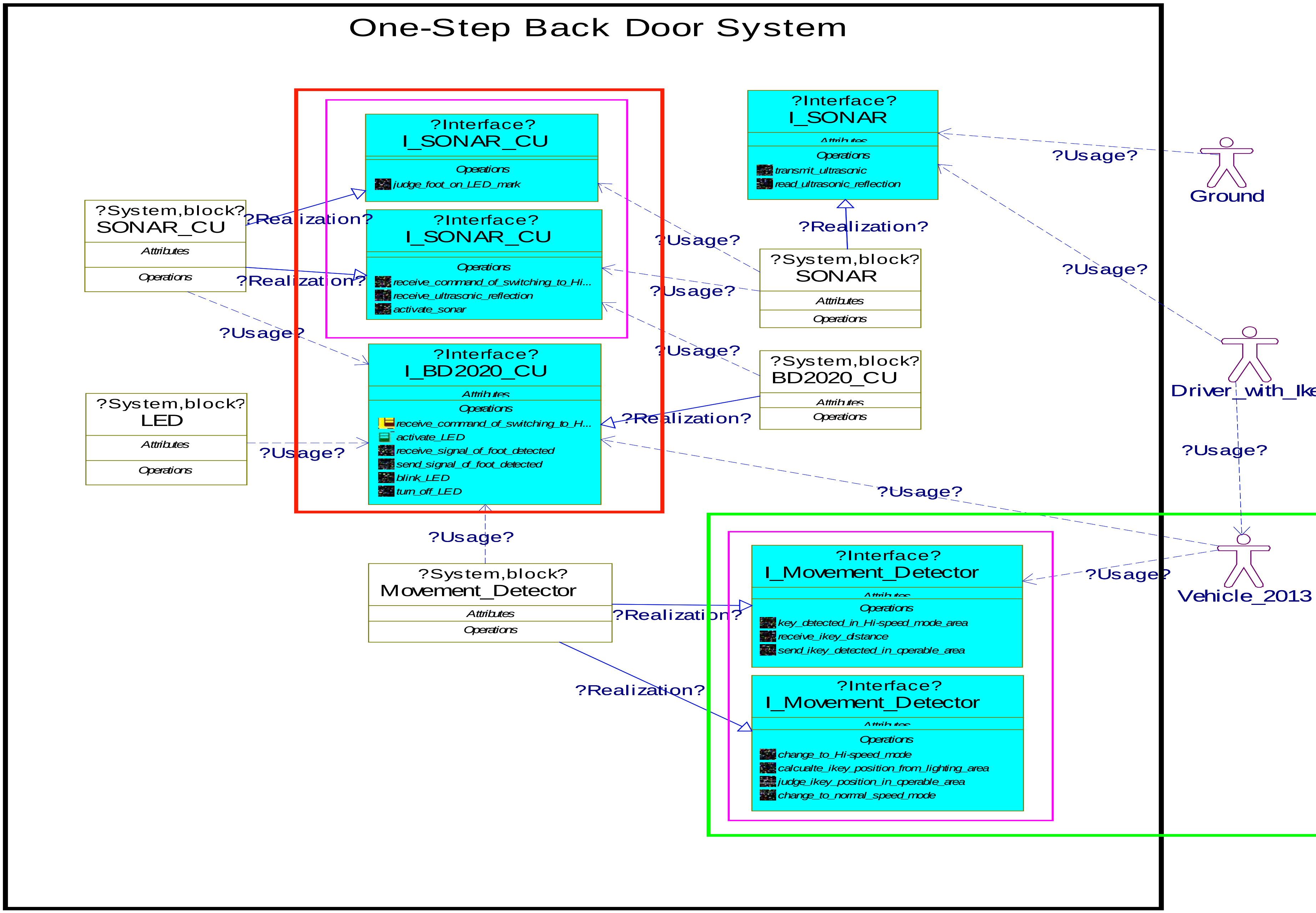


Fig 4.13 Block definition diagram of "One-Step Back Door System"

4.3 Refinement of requirement

In the last section, the functions of the each system component are clearly derived by using the behavioral diagrams to meet the requirement of the user needs of user friendliness and “Omotenashi”. In this section, the functional requirement defined in last section will be refined to the requirement diagram shown in section 4.1.

For example, for UF2 requirement “Customers can open and close the back door with natural motion and one action without dirtying their hands and their clothes” shown in Fig 4.2. Based on the functional requirement defined by the behavioral diagrams, we know that the “One-Step Back Door System” can detect the customers foot placed once the customer step on the LED mark area. Therefore, we can derive the functional requirement which is “Senor can detect the customer’s foot placed between the sensor and the ground” from UF2 requirement. In last section, sonar system which includes “Sonar” and “Sonar CU” is assumed to be the component of “One-Step Back Door System” since it will function as the sensor to detect the foot of the driver. However, in fact, engineers can provide several methods to meet this functional requirement such as by using “Laser radar”, “Capacity”, “Infrared ray” and “Sonar”. The methods might be deleted or added after due to whether they can satisfy the other requirements, but the block definition diagram is recommend here to record the several methods as an evident that a lot of methods are considered and also shows the traceability of selecting the method. Fig 4.14 shows the alternative method to satisfy the functional requirement.

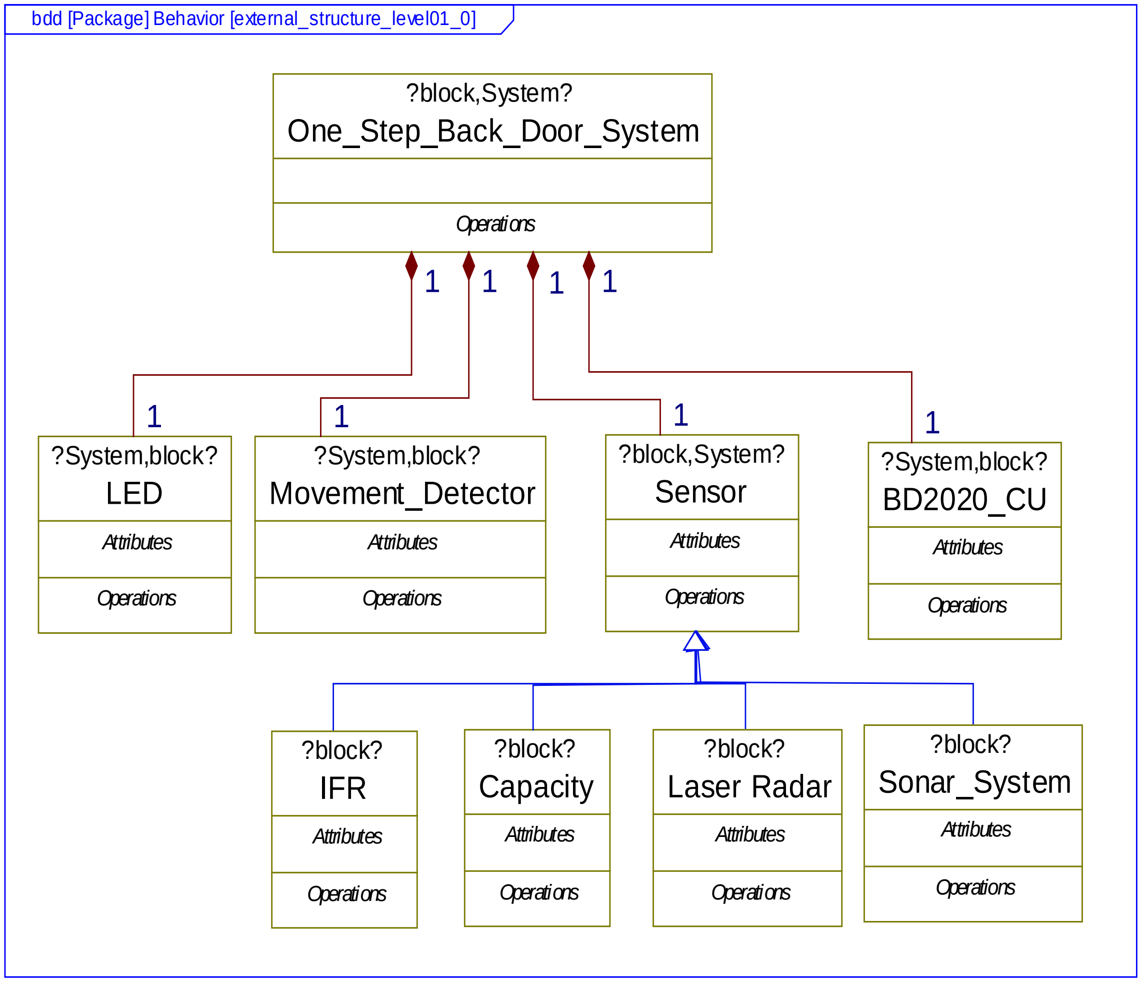


Fig 4.14 Alternative method to satisfy the functional requirement

However, with the analysis of the requirement, requirement that "Sensor shall function no matter that vehicle is covered by the splash or the mud" is partitioned to the "Environment" packages which contained in "Operation". Then both functional requirement and performance requirement can be derived from this requirement as follow:

1) Functional requirement: Sensor can detect the customer's foot placed between the sensor and the ground.

2) Performance requirement: Sensor can detect customer's foot farther than 20cm.

The functional requirement is as the same as the one derived from the No.2 requirement of the "User friendliness" (UF2). What should be noticed is that the performance requirement is the constraint to the methods which mentioned above, since the sensor should be performed no matter vehicle is covered by the splash or the mud, so the choice of using "Infrared ray" and "Laser radar" as the sensor should be deleted since both of them are vulnerable to dirty. And the choice of using "Capacity" is also given up since it is not able to meet the performance requirement of detecting things farther than 20cm. Fig 4.16 shows the part which we discussed above of the requirement diagram. It shows how the systems component is decided to satisfy the functional requirement which derived from the source requirement. As a result, the internal structure is shown in Fig 4.15.

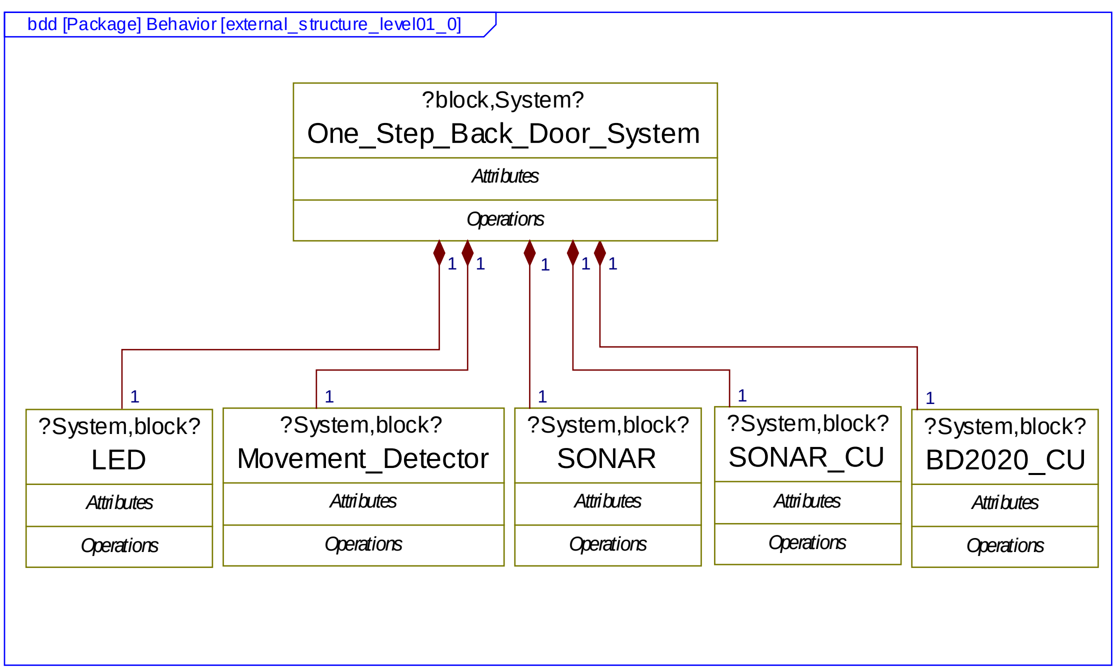


Fig 4.15 Block diagram of external structure of "One-Step Back Door System"

In terms of the process discussed above, as a result, we can get a completed requirement diagram. Fig 4.17 shows the hierarchies of specifications and requirements. It also depicts lager numbers of relationships to a single requirement. And it is useful for representing the traceability of a single requirement to examine how that requirement is satisfied, and to examine its derived relationships with other requirements.

The discussion mentioned above shows how the E/E architecture of the "One-Step Back Door System", which can be operated by natural motion, is designed based on the model-based systems engineering (MBSE). And the functions of the new back door system can be derived by analyzing the requirements.

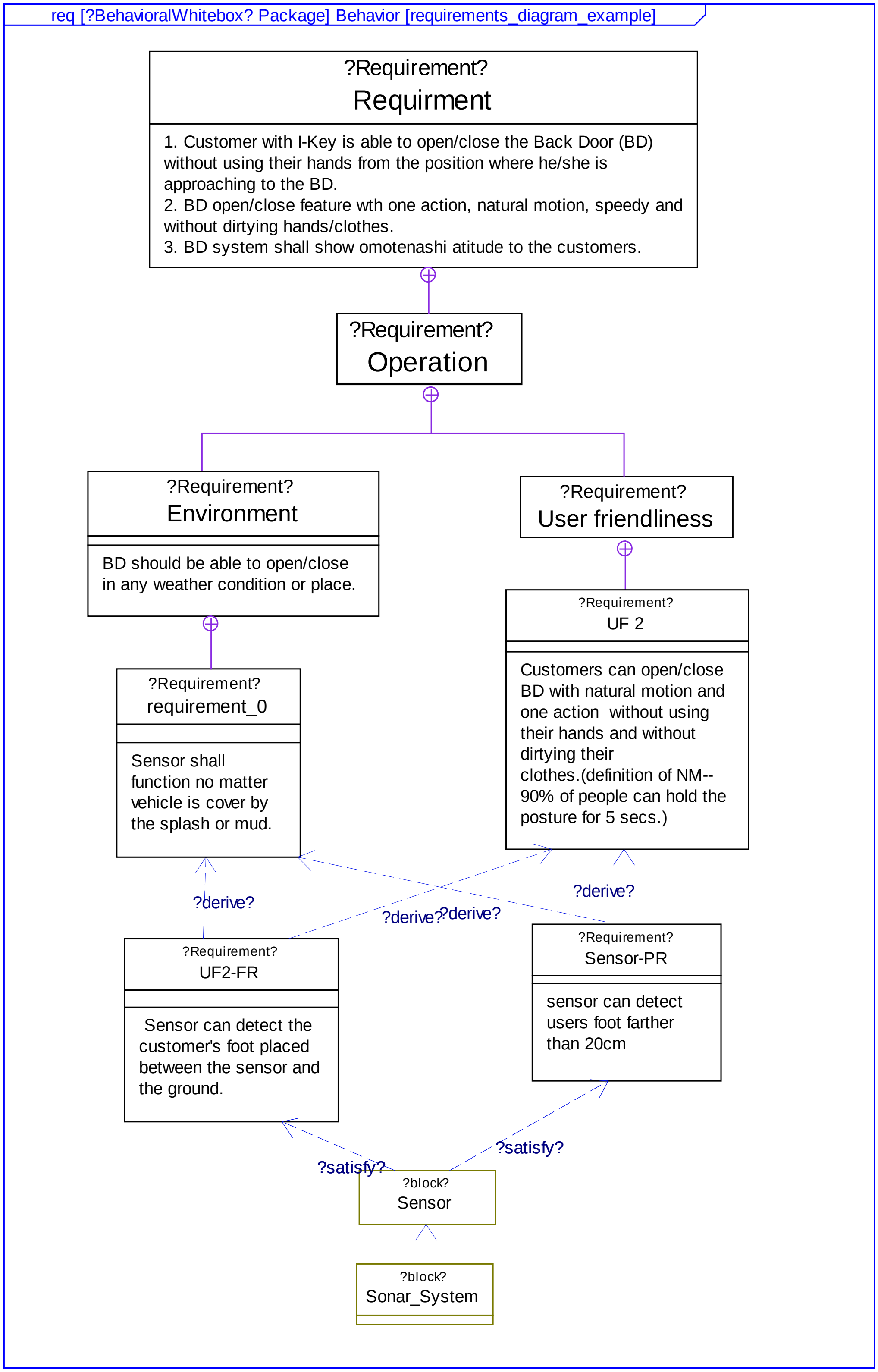


Fig 4.16 Part of the requirement diagram of determining "Sonar system"

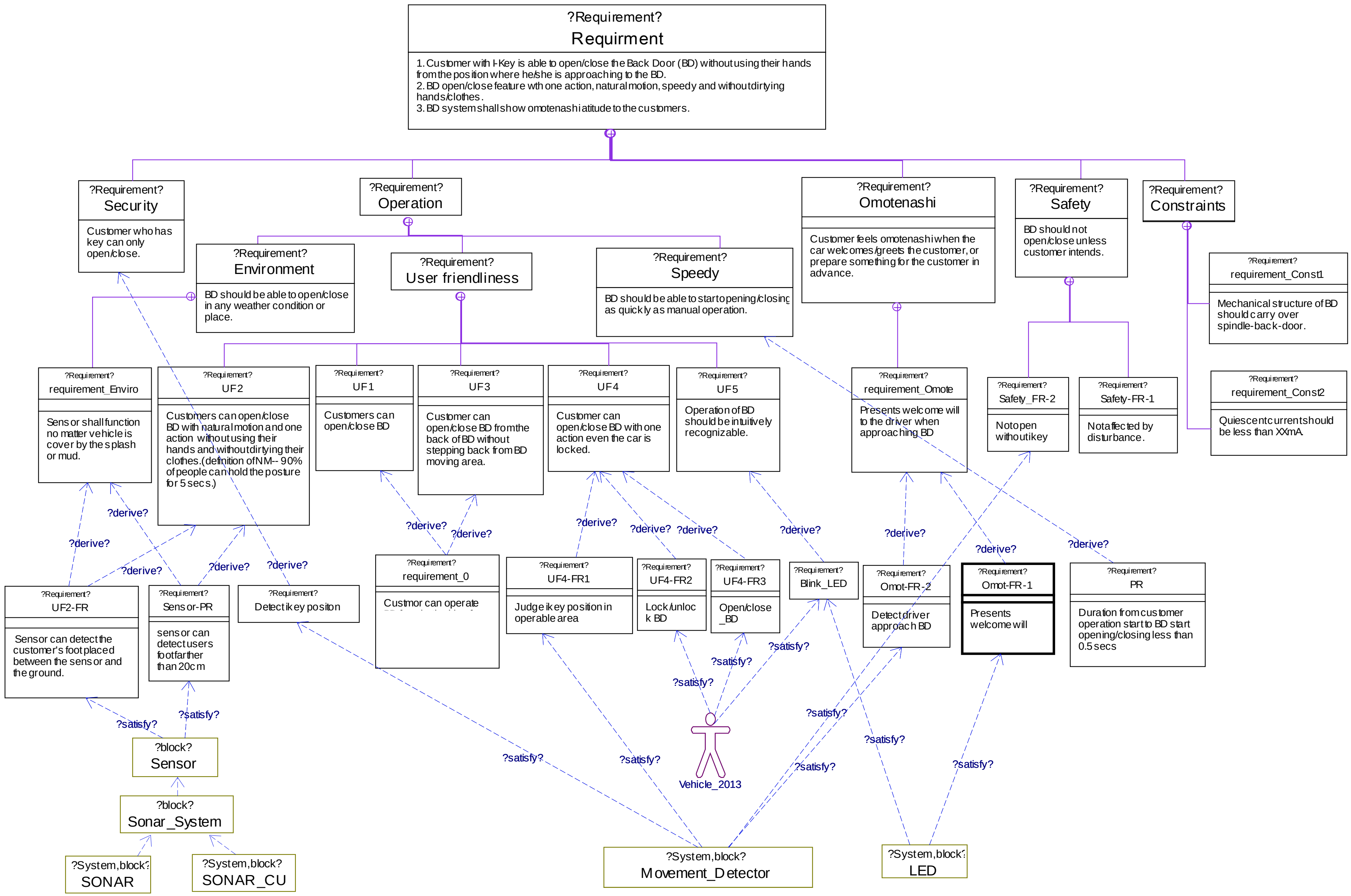


Fig 4.17 requirement diagram

5. Discussion and Recommendation

5.1 Noticed points

During the design process of the E/E architecture of the "One-Step Back Door System" by using model-based system engineering, a lot of points are discovered by using the behavioral diagrams such as sequence diagrams and the activity diagrams and the structure diagrams. By analyzing the requirements, functions of the "One-Step Back Door System" are derived without exception, which are easily be ignored by the current development process. For example, in sequence diagram of "switch to Hi-speed mode" shown in appendix, the option part shows what interaction is occurred in the condition of "1min passed in Hi-speed mode== TRUE & detect I-key==FALSE for 5sec". At first, the option part is considered in the sequence diagram of "detect I-key movement" for the reason that the condition written in the option part is defined to show what condition the movement detector should be in Hi-speed mode and return back to the normal speed mode. But the problem is found in activity diagram. In fact, it is a triggering condition for movement detector in Hi-speed mode or out of it, so the function of making decision of whether the "Movement Detector" should be switched to Hi-speed mode or not is made by the "Vehicle 2013".

Moreover, the function of realizing the "Open BD" use case is analyzed at first by the behavioral diagrams, and the system structure is also defined by using the structure diagram. And the position of placing the sonar is supposed on the back door handle, but problem is found when analyzing the function of "Close BD" use case. Since the start point of the "Close BD" is the back door is opened. Therefore, the antenna set on the back door handle is not able to detect the driver's I-key whether it is in the operable area or not. However the requirement of the "One-Step Back Door System" is user can close the back door with one action and natural motion which derived from the requirement "Back door open and close features with one action, natural motion, speedy and without dirtying the customers' hands and clothes". So the place of the antenna is discussed again in order to meet this requirement to show the "user friendliness" and "Omotenashi". From this example, we can see that the problem won't be found without analyzing the use case of "Close BD". And this kind of miss happen quite often, since functions should be performed of the "Open BD" use case and the "Close BD" use case are almost the same, so engineers just think of one of them. Therefore, use case is very important to the engineers to analyze the system on the system level.

5.2 Interview

In Chapter four, systems engineering process is shown to obtain the new back door system architecture by using SysML to solve the complication issue of the E/E architecture. In order to know whether this process is helpful to the E/E architecture design, I interviewed the engineers from the NISSAN MOTOR CO., LTD. They are Miyashita-San and Nakamoto-San from the electrical and electric department. From the interview, we know that they are very happy with this design process. They checked the system models I provided, and they think that use case diagrams and the sequence diagrams are really helpful and useful for defining the functional requirement without exception. And all the system models and information can be traceable back to its original requirement in the requirement diagram, which is very useful for the engineers to have a whole view of the system and the system models can saved as documentation which is very important for re-development such as adding new ECU to realize new function. Moreover, activity diagrams show the how the functions are allocated to the system components. It is not only useful to the system engineers, but also helpful for the components designers to know how his or her components are performed in the system. However, though design the E/E architecture by using SysML is very helpful, but the problem have to solve is that the cost of importing the SysML to the company, and the engineers should have to learn how to use it in order to take part in the design process.

5.3 Recommendation

First of all, with MBSE, the functions of E/E architecture can be fully considered without ignored. For instance, the engineers introduced me that in the company they also analyzed the system by using some diagrams like sequence diagram, but the question is a lot of key message and functions are ignored since they take it for granted, but in the sequence diagram of SysML, the function which should be performed, and which component to realize this function and even the statements of the messages are all provided carefully.

Secondly, nowadays, in many cases, systems engineers design the product with their experience, they are not quite clear about the details of designing components, so after they give the quotation to the suppliers or the subsystem engineers, the suppliers and subsystem engineers might point out the design can’t be realized, then the system engineers have to redesign the E/E architecture. This situation leads to the waste of time. However, time will be saved if by using MBSE approach, since it provides a chance for the engineers from different departments and the suppliers from different companies to discus with each other based on using the several system models, again these system models can help the engineers to find a lot of things at the early stage.

In addition, with the system models, engineers can be stimulated to think about the system layout. For example, in this research, scene of open back door is considered at the beginning, for ordinary people, we consider the scene of close back door might be almost the same with open back door, since the function is also defined. But with the discuss based the systems model, we find that the antenna which currently located on the back door should be moved to the bumper in order to realize the function of “detect I-key position”. So with MBSE approach, to some extent, not only the functions can be defined clearly, but also the allocation of the components can also be considered in the quite early step.

Thirdly, from activity diagram and block diagram, engineers can have a whole view of the system of which functions should be performed in which components. Currently, if some other new function added, the usual way is to plus a component to realize this function by pulling a harness. So the E/E architecture might be very complicated, but with activity diagram, functions are all displayed in the corresponding swim-lane, it is easy for the engineers to arrange the architecture.

Moreover, since engineers have a clear view of the system, so they can search for the suppliers based on the system models. In other words, engineers exactly know the functions so they can outsource the subsystem or the components to the suppliers who have the adequate ability to fulfill the tasks. It can be help to change the current situation, which suppliers might change the request for quotation for adding new functions.

Finally, return back to the development cycle, right after the stage of "engineering completion" which mentioned in the chapter one, things related to the money are all should be finished e.g. contract, profit and how much should be invested into the product, etc. But as a matter of fact, the amount of money invested and the profits often differ significantly compared with the expectation, largely because of the change of functions, the request for quotation from suppliers. It also leads to waste the time and money on the employees from financial department. Therefore, with the feedback from the engineers and the merit by using MBSE mentioned above. I believe that it is possible to prevent these kinds of matters by using systems engineering at the beginning of the development cycle.

6. Conclusion

In this thesis, the systems engineering process was shown to obtain the architecture of a back door system by using Systems Modeling Language to solve the complication issue of the E/E architecture design due to adding ECUs to realize new functions. I firstly analyzed the E/E architecture of an existing back door system which is consisted of a body control module and a contactless sensor. By using the behavioral diagrams such as sequence diagrams and activity diagrams, both the interactions between user and the current back door system, and the interactions among the system components are clarified. Through the analyzing of the existing back door system, how it performed was shown by using the system models, which was effective to design the E/E architecture for a new back door system which can be easily operated with one step. Then the E/E architecture of a new back door system is designed by model-base systems engineering. The requirements for the new back door are analyzed and stakeholders' needs are well defined by using the use case diagrams and the sequence diagrams. The functions of the new back door system were derived by analyzing the requirement and the traceability among the requirements is held in the system models. And the logical architecture is shown in the activity diagram by allocating the functions to each component which compose the system. And from the interview with the engineers of automotive company, I found that the activity diagram is very useful to design the E/E architecture since other methods such as topology should be used and the constraints such as the length of the wiring and the weight should also be considered refers to the activity diagram. As a result, design process of E/E architecture was evaluated by the engineers and they believe it must be effective compared with the current process. Therefore, MBSE approach is recommended since the early stage of development process on the system level is very important.

Acknowledgements

First of all I would like to thank Professor Nishimura Hidekatsu, my supervisor. He has provided me with an incredible amount of knowledge, advice and encouragement over my 2 years in Keio University. Without his guidance and assistance, this research won't be able to be finished. Professor Nishimura guided me step by step through a process of how doing research, and provided me with countless suggestions regarding the direction and contents of my research. Professor Nishimura spent a lot of time on his students, this help me a lot when I face problems and when my method and direction was wrong. I am really grateful for his patience and understanding. Moreover, since I was raised up in Shanghai and I am planning to work in Japan, Professor Nishimura also gives me a huge support during my job hunting and he also taught me a lot about the Japanese company culture.

I would also like to thank engineers Miyashita-San, Nakamoto-San and Uchida-San from the automotive company. They are the specialists who really know a lot of the E/E architecture. We spent around three hours every week to make a discussion, and teach me much knowledge not only just about my thesis, but also the thinking way of Japanese engineers. And I really impressed of their attitude to the work. I believe that Japanese automotive company will be the top with such excellent engineers.

And I also want to thank my fellow classmates, for their assistance.Yon-San and Nakakura-San, we usually discussed the research with each other, and they really gave me a lot ideas and great courage.

Finally, I would also like to thank my family for their huge supporting for the two years even though they are in Shanghai. They often came to Tokyo to help me refresh myself .

Reference

[1] INCOSE systems engineering handbook, 3.6 of 20, Version 3.1, August 2007

[2] ISO/IEC/IEEE 42010 International standard, p5-8, first edition, 2011-12-01

[3] ISO/IEC/26702, IEEE std 1220-2005 International standard, p11-13, first edition, 2007-07-15

[4] Sanford Friedenthal, Alan Moore, Rick Steiner, A Practical Guide to SysML, The System Modeling Language. Second Edition, The MK/OMG Press

[5] Jon Holt and Simon Perry, *SysMlL for Systems Engineering.*

[6] G. Leen and D. Heffernan. Expanding automotive electronic system. In *IEEE Computer*, volume 35, pages 88–93, 2002.

[7] O. Larses. *Architecting and modeling of automotive embedded systems*. PhD thesis, Royal Institute of Technology, 2005.

[8] Peter Wallin., Jakob Axelsson., *A Case Study of Issues Related to Automotive E/E System Architecture Development.*

[9] INCOSE International Workshop MBSE Workshop January 26th-28th, 2013

[10] Harald Heinecke, Klaus-Peter Schnelle, Helmut Fennel, Jürgen Bortolazzi, Lennart Lundh, Jean Leflour, Jean-Luc Maté, Kenji Nishikawa, Thomas Scharnhorst,. *AUTomotive Open System ARchitecture - An Industry-Wide Initiative to Manage the Complexity of Emerging Automotive E/E-Architectures.* 2004

[11] Thomas Blote, *複雑なアーキテクチャーのための開発ツール*. 2010

[12] Thomas Liebetrau, Ursula Kelling, Tobias Otter, Magnus Hell. *Energy Saving in Automotive E/E Architectures*.2012

[13] Frickenstein, Wilhelmi, Hudi, Michael, Tanneberger, Milke,. *New Automotive Standards* 15th Int. Conference 'Advances in Automotive Electronic' .Ludwigsburg .2011

[14] Purnendu Sinha, Vinod Agrawal*, Evaluation of Electric-Vehicle Architecture Alternatives. 2012

[15] Jürgen Leohold, *Communication Requirements for Automotive Systems* WFCS 2004, 5thIEEE Workshop on Factory Communication Systems, Wien, Sep. 22-24, 2004

[16] Hannu Jaakkola and Bernhard Thalheim. (2011) "Architecture-driven modelling methodologies." In:*Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII*. Anneli Heimbürger et al. (eds). IOS Press. p. 98

[17] Kabsu Han, Jeonghun Cho, *A Distributed Service Architecture for Networked Automotive E/E System.* 2007

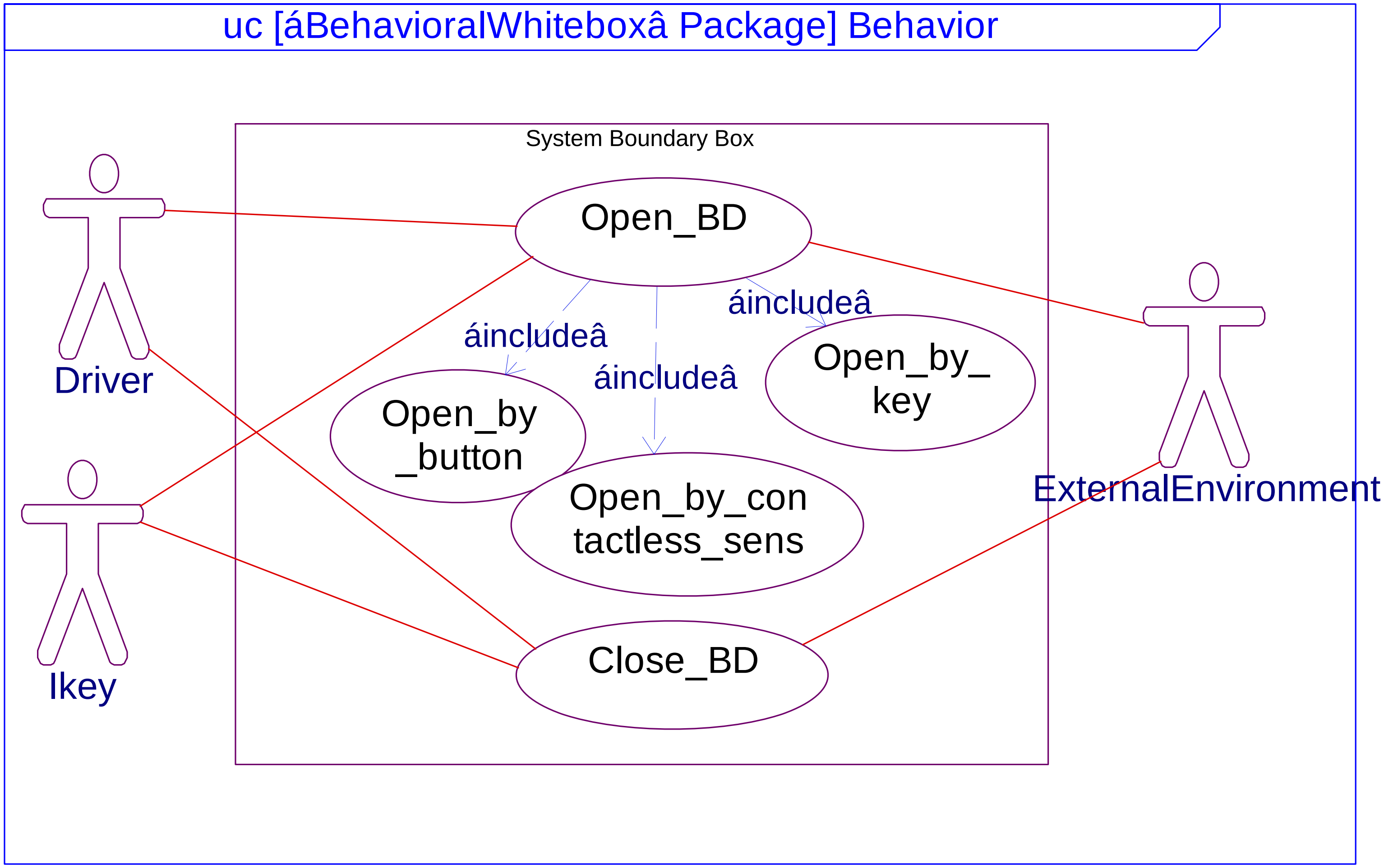
[18] Chris Quigley, Peter Jones, Ross McMurran, *Design approaches for integrating CAN with emerging time-triggered protocols.*2006

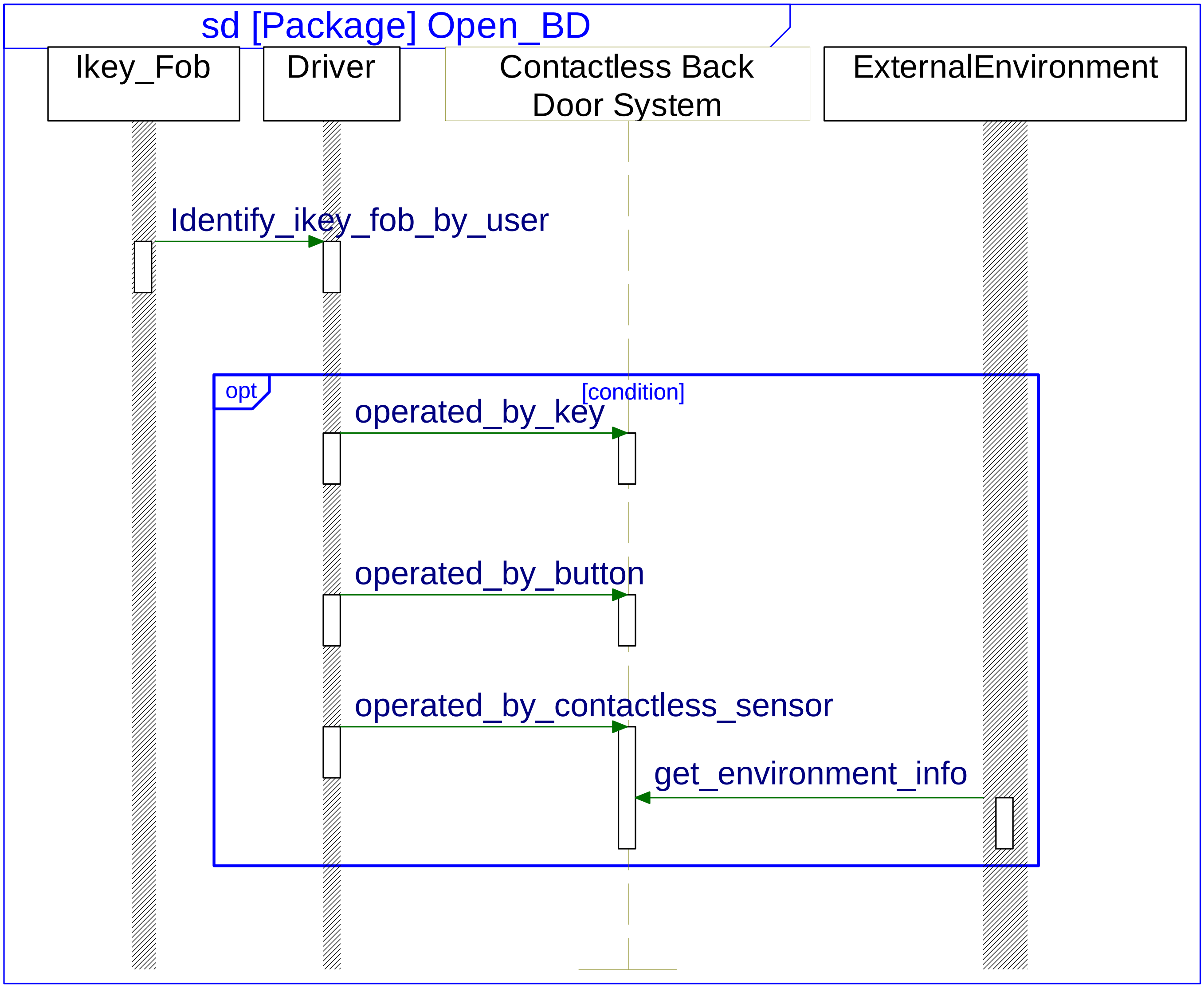
[19] *広がりを見せるアーキテクチャの標準化*.J2005.2 RT

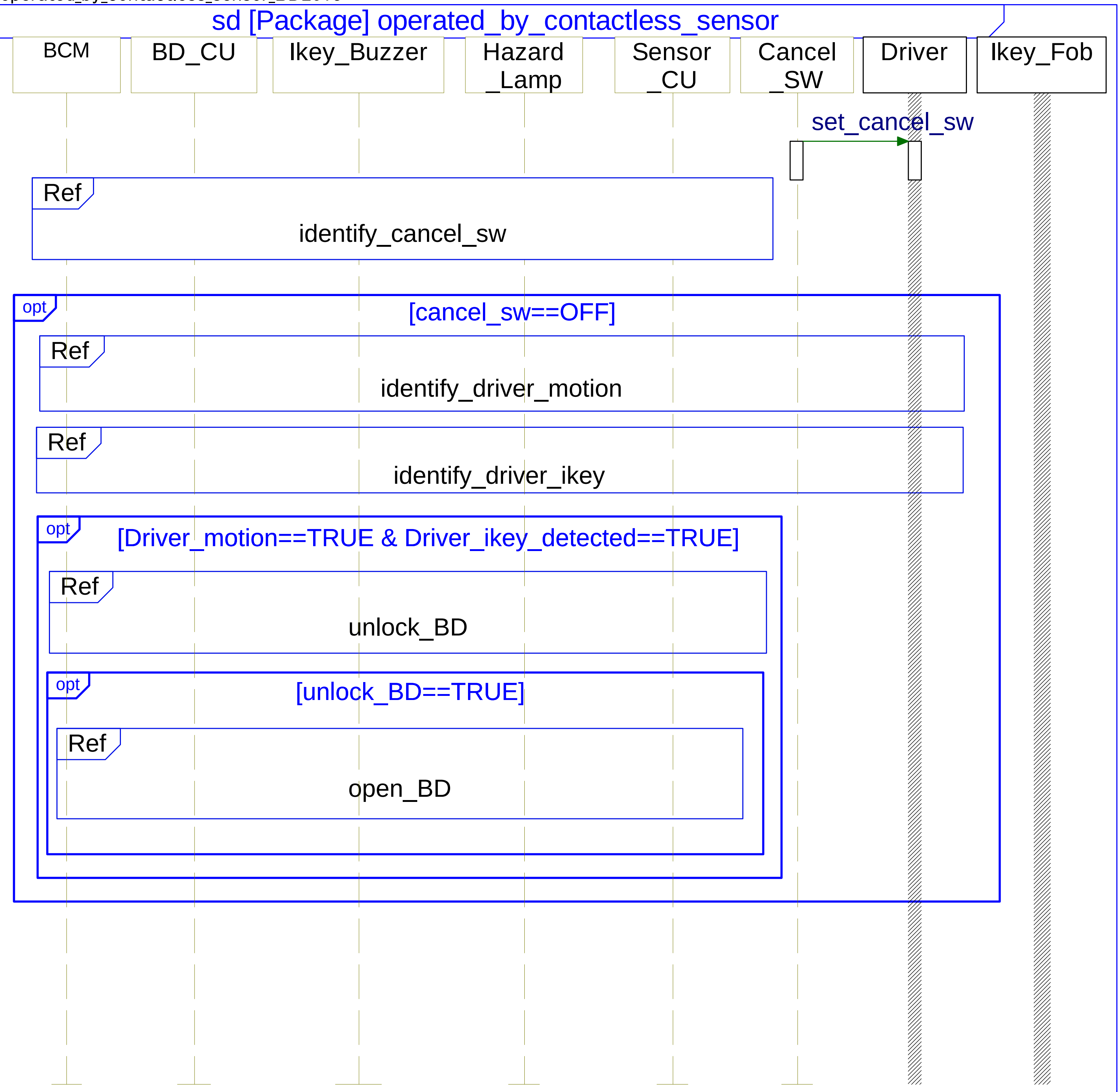
[20] 阿部睦, *今後の車載電子システムとアーキテクチャ指向・形式手法の適用*.2011

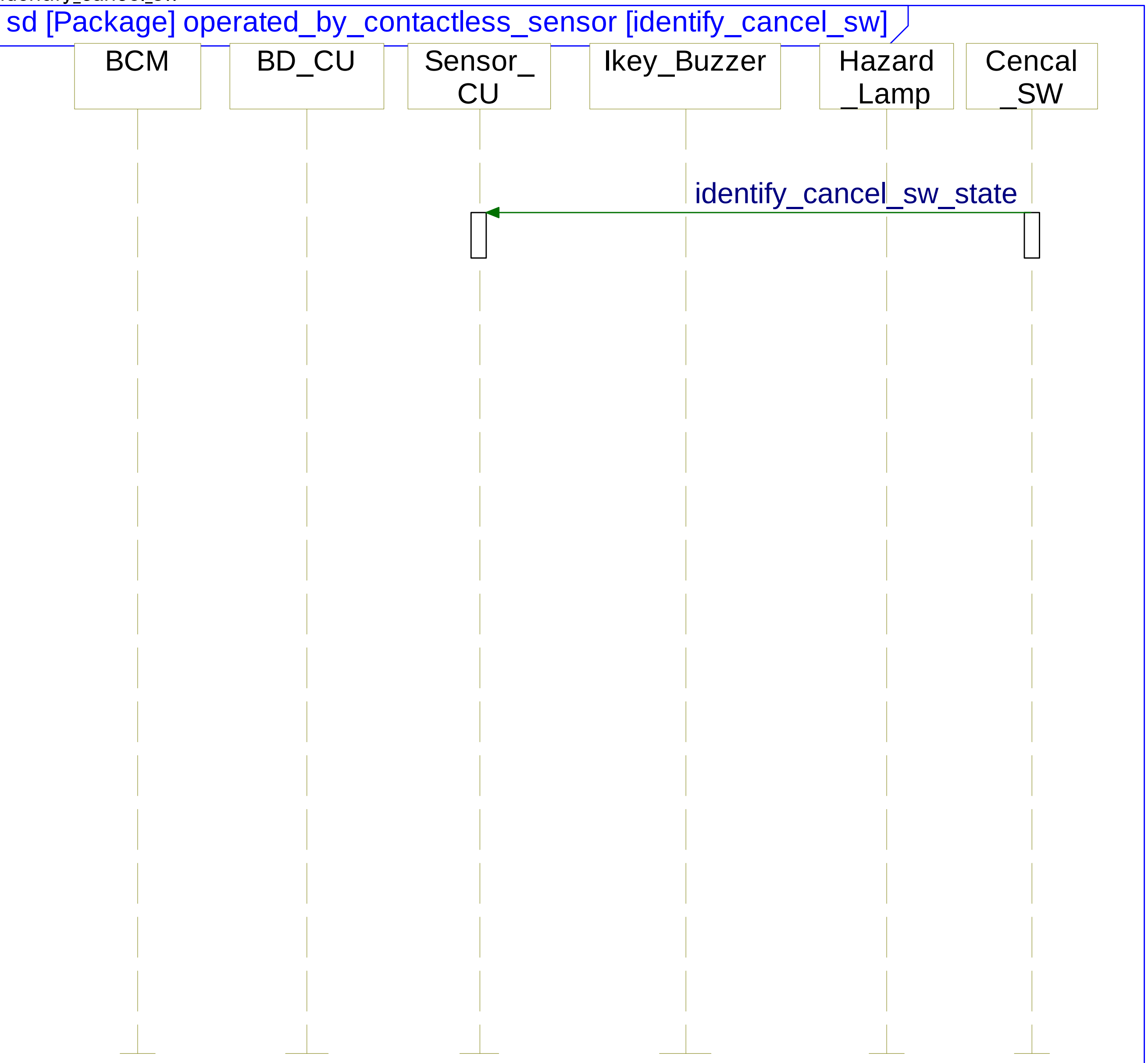
Appendix

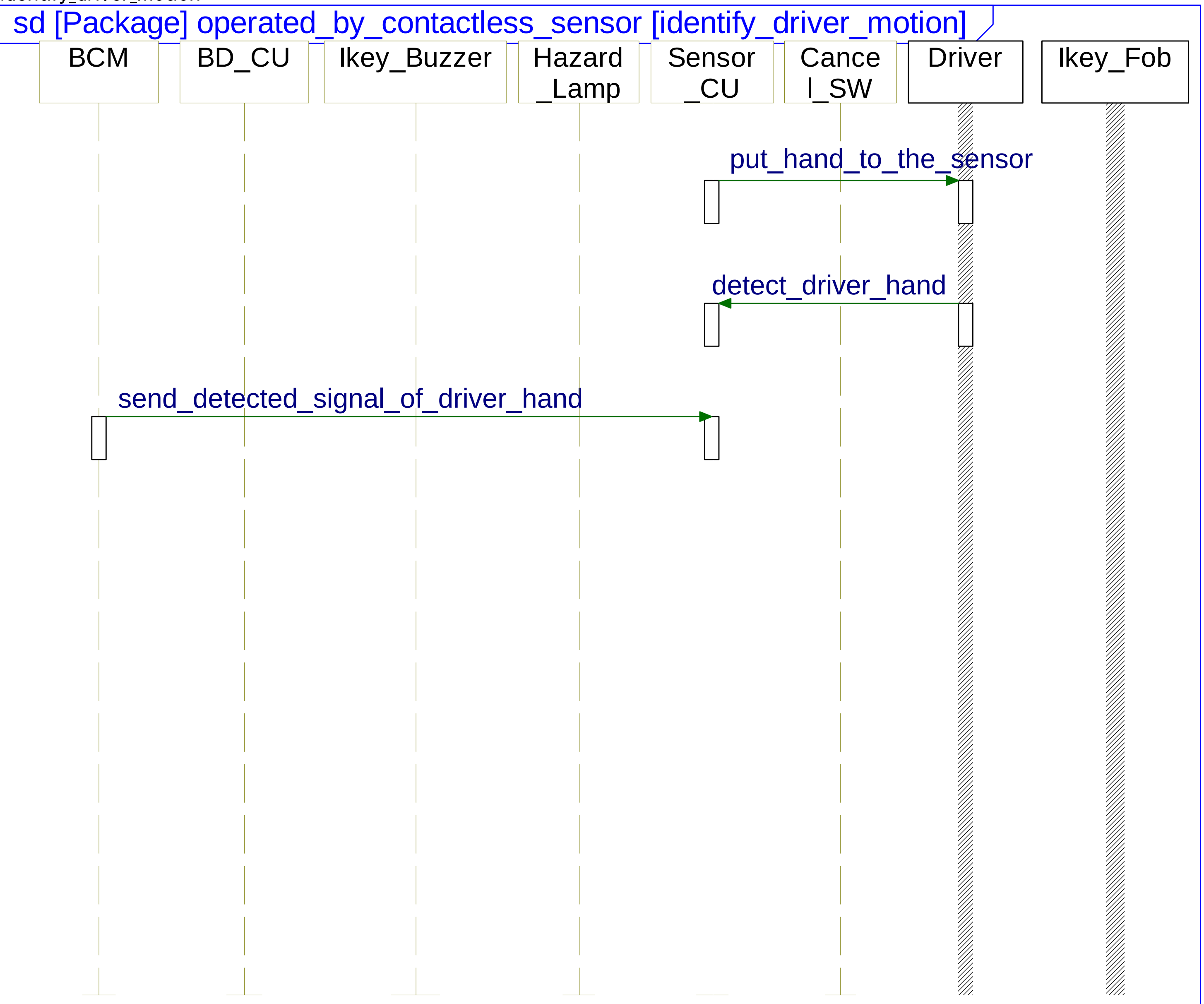
Appendix of “Contactless Back Door System”


uc [áBehavioralWhiteboxâ Package] Behavior
System Boundary Box
Open_BD
áincludeâ
áincludeâ
áincludeâ
Open_by_
key
Open_by
_button
Open_by_con
tactless_sens
Close_BD
Driver
Ikey
ExternalEnvironment

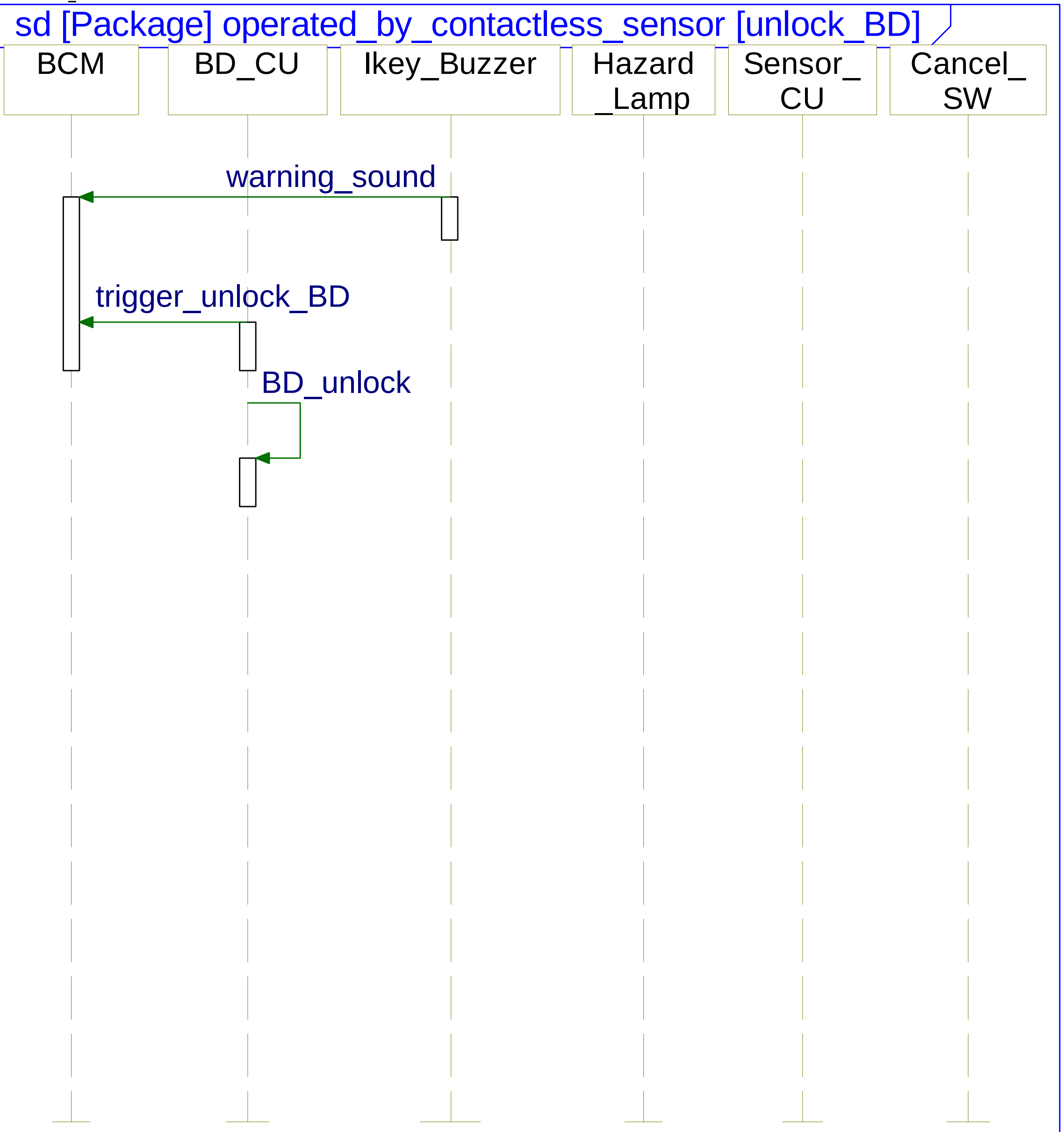

sd [Package] Open_BD
Ikey_Fob
Driver
Contactless Back Door System
ExternalEnvironment
Identify_ikey_fob_by_user
opt
[condition]
operated_by_key
operated_by_button
operated_by_contactless_sensor
get_environment_info

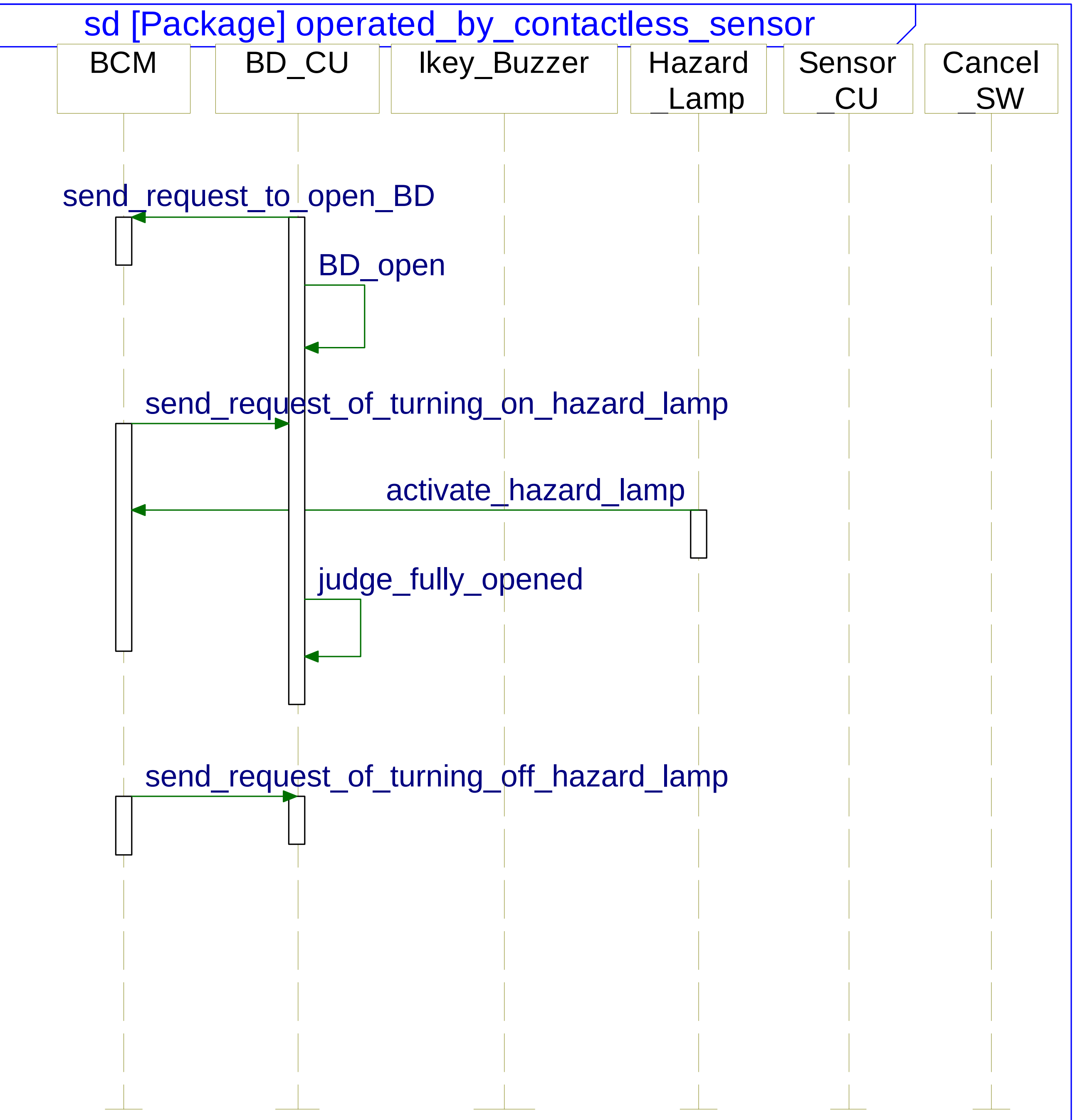

sd [Package] operated_by_contactless_sensor
BCM
BD_CU
Ikey_Buzzer
Hazard _Lamp
Sensor _CU
Cancel _SW
Driver
Ikey_Fob
set_cancel_sw
Ref
identify_cancel_sw
opt
[cancel_sw==OFF]
Ref
identify_driver_motion
Ref
identify_driver_ikey
opt
[Driver_motion==TRUE & Driver_ikey_detected==TRUE]
Ref
unlock_BD
opt
[unlock_BD==TRUE]
Ref
open_BD


sd [Package] operated_by_contactless_sensor [identify_cancel_sw]
BCM
BD_CU
Sensor_ CU
Ikey_Buzzer
Hazard _Lamp
Cencal _SW
identify_cancel_sw_state

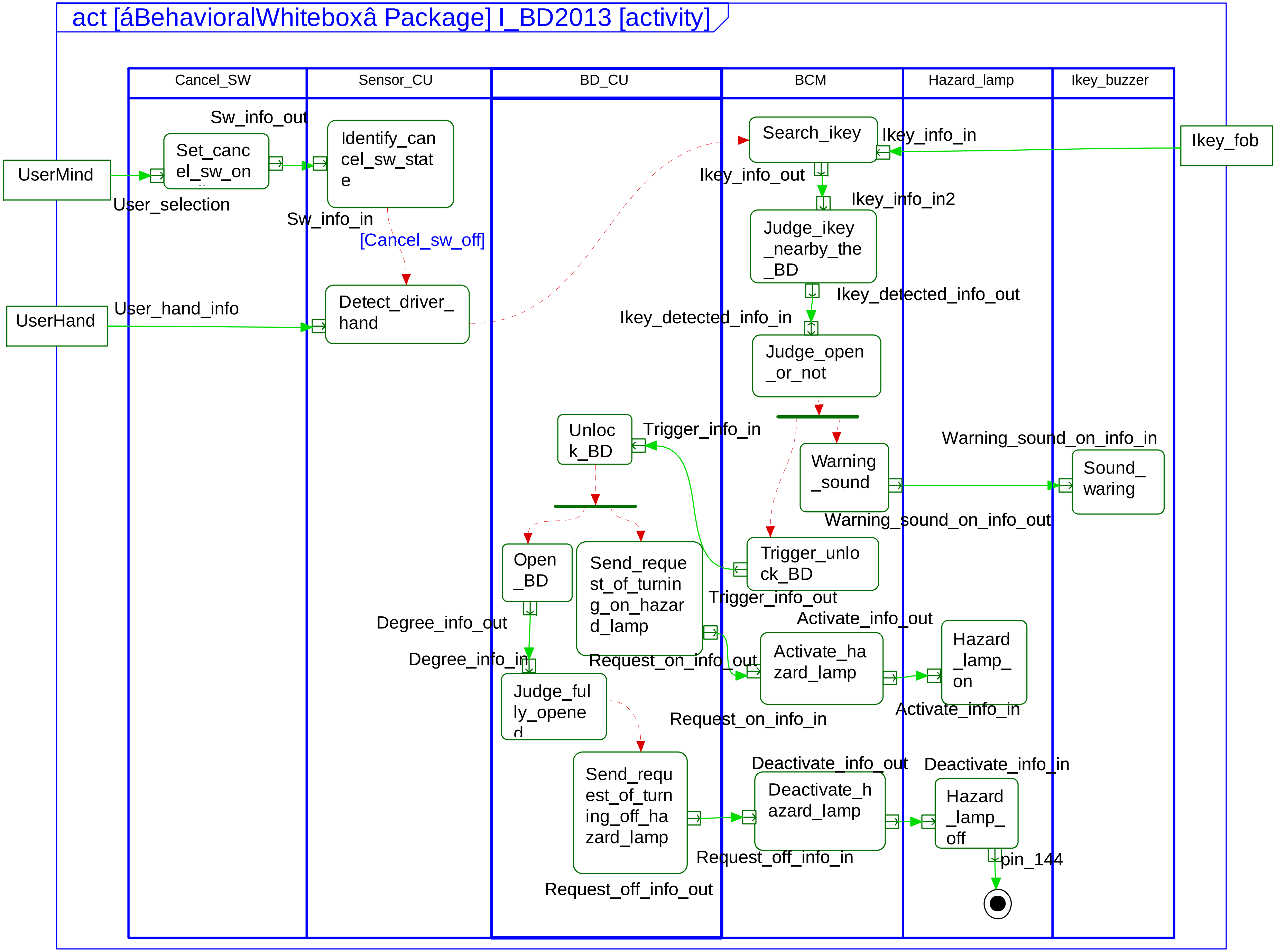

sd [Package] operated_by_contactless_sensor [identify_driver_motion]
BCM
BD_CU
Ikey_Buzzer
Hazard_Lamp
Sensor_CU
Cancel_SW
Driver
Ikey_Fob
put_hand_to_the_sensor
detect_driver_hand
send_detected_signal_of_driver_hand

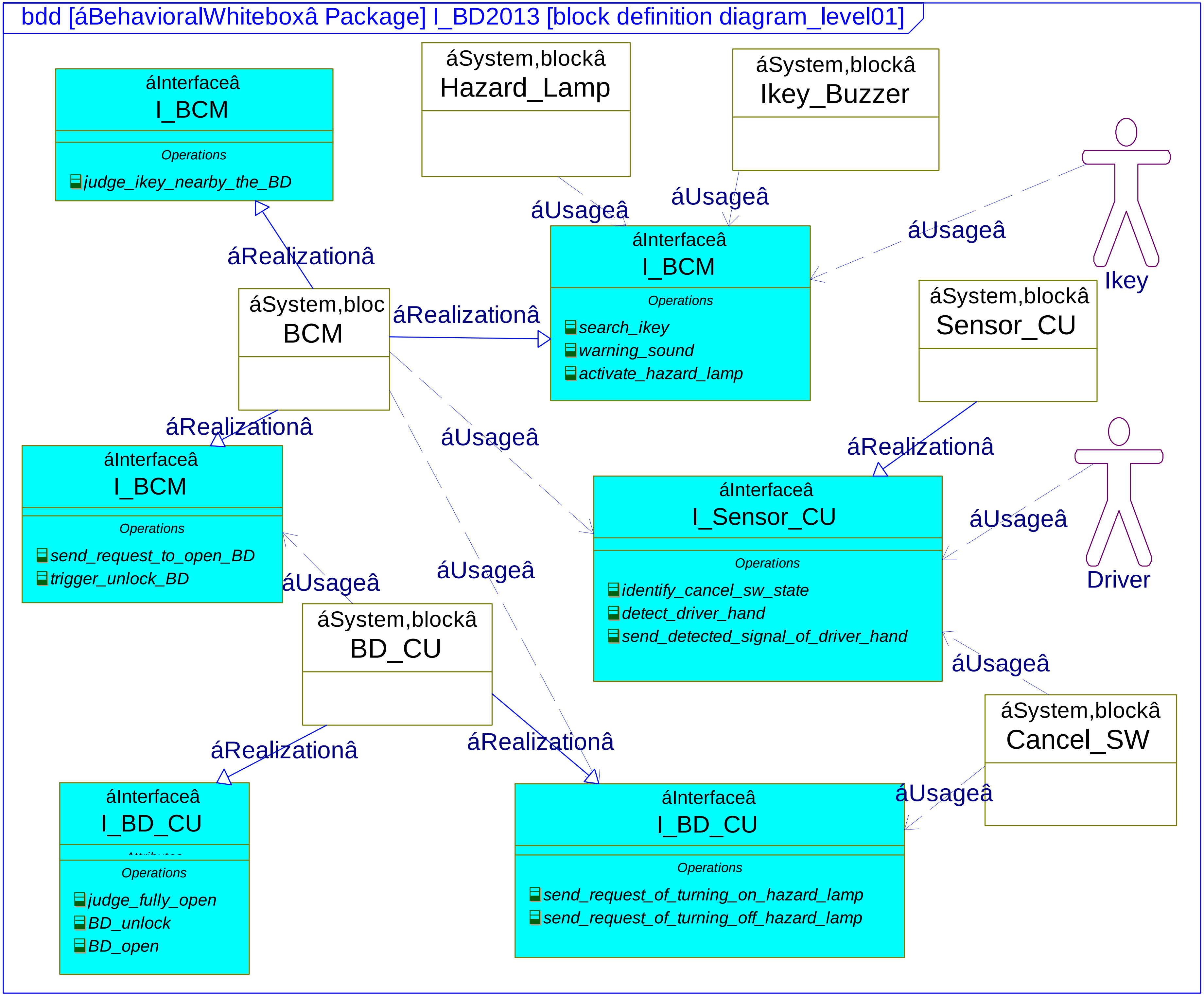

sd [Package] operated_by_contactless_sensor [identify_driver_ikey]
BCM
BD_CU
Ikey_Buzzer
Hazard _Lamp
Sensor _CU
Cancel _SW
Ikey_Fob
Driver
search_ikey
send_signal_to_BCM
judge_ikey_nearby_the_BD


sd [Package] operated_by_contactless_sensor [unlock_BD]
BCM
BD_CU
Ikey_Buzzer
Hazard_Lamp
Sensor_CU
Cancel_SW
warning_sound
trigger_unlock_BD
BD_unlock

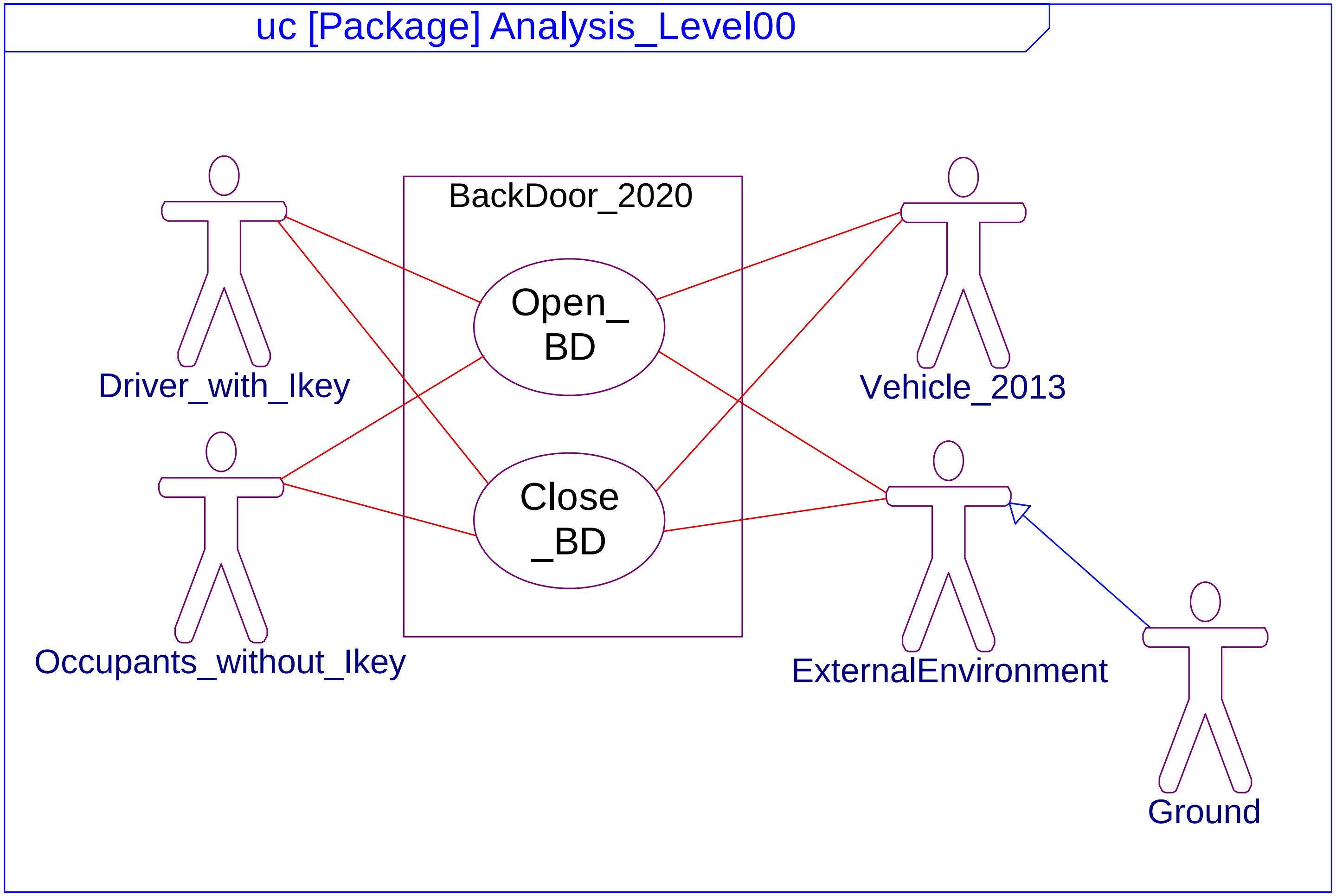


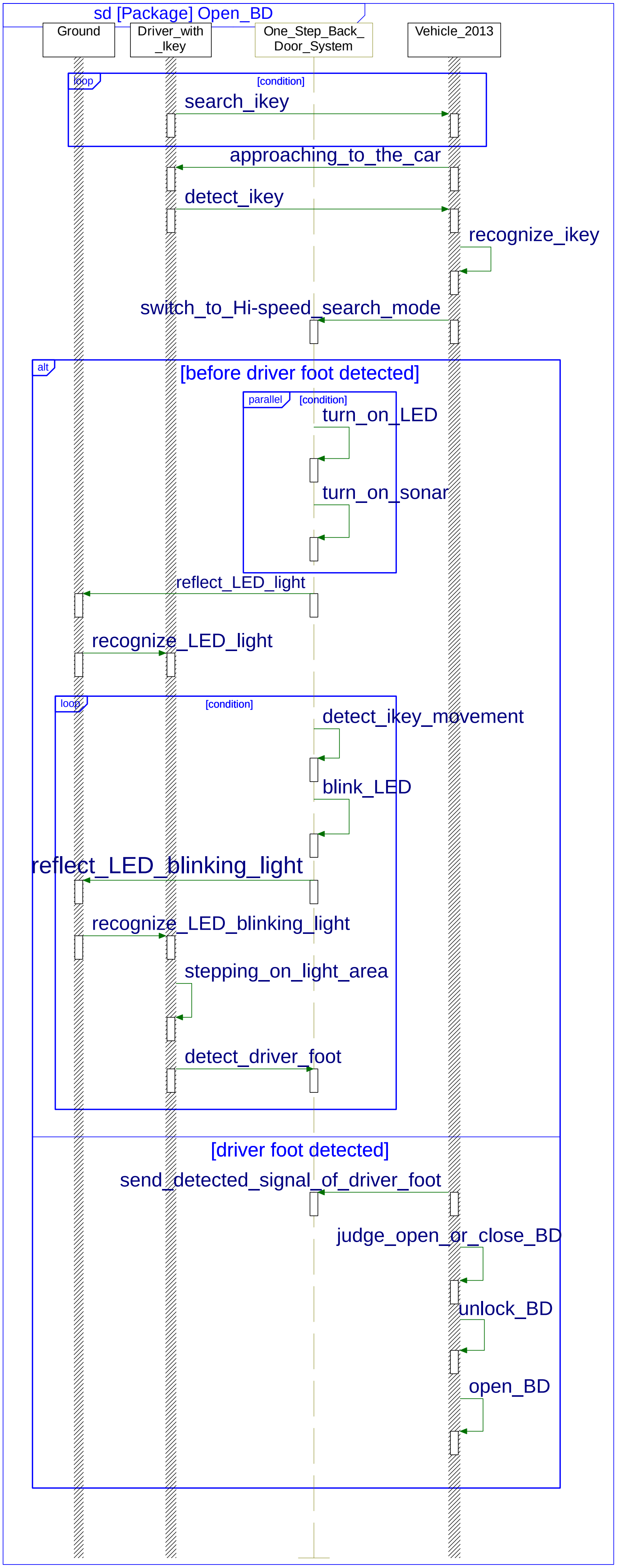
sd [Package] operated_by_contactless_sensor
BCM
BD_CU
Ikey_Buzzer
Hazard _Lamp
Sensor _CU
Cancel _SW
send_request_to_open_BD
BD_open
send_request_of_turning_on_hazard_lamp
activate_hazard_lamp
judge_fully_opened
send_request_of_turning_off_hazard_lamp

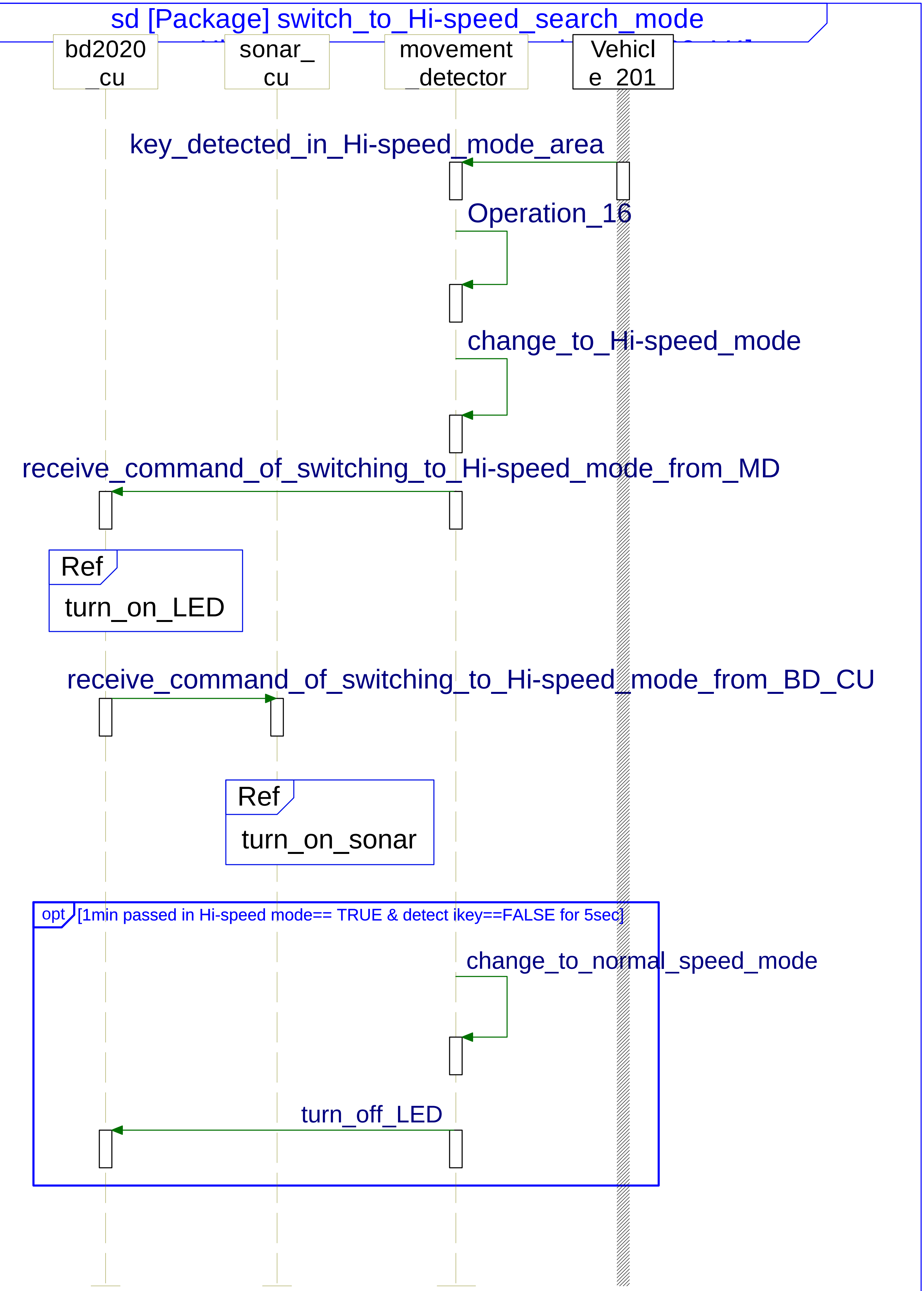

act [áBehavioralWhiteboxâ Package] I_BD2013 [activity]
Cancel_SW
Sensor_CU
BD_CU
BCM
Hazard_lamp
Ikey_buzzer
UserMind
User_selection
Set_cancel_sw_on
Sw_info_out
Sw_info_in
Identify_cancel_sw_state
[Cancel_sw_off]
UserHand
User_hand_info
Detect_driver_hand
Search_ikey
Ikey_info_in
Ikey_fob
Ikey_info_out
Ikey_info_in2
Judge_ikey_nearby_the_BD
Ikey_detected_info_out
Ikey_detected_info_in
Judge_open_or_not
Warning_sound_on_info_in
Warning_sound
Sound_waring
Warning_sound_on_info_out
Unlock_BD
Trigger_info_in
Trigger_unlock_BD
Trigger_info_out
Open_BD
Send_request_of_turning_on_hazard_lamp
Degree_info_out
Degree_info_in
Request_on_info_out
Activate_info_out
Activate_hazard_lamp
Hazard_lamp_on
Activate_info_in
Request_on_info_in
Judge_fully_opened
Deactivate_info_out
Deactivate_info_in
Send_request_of_turning_off_hazard_lamp
Deactivate_hazard_lamp
Hazard_lamp_off
Request_off_info_in
pin_144
Request_off_info_out

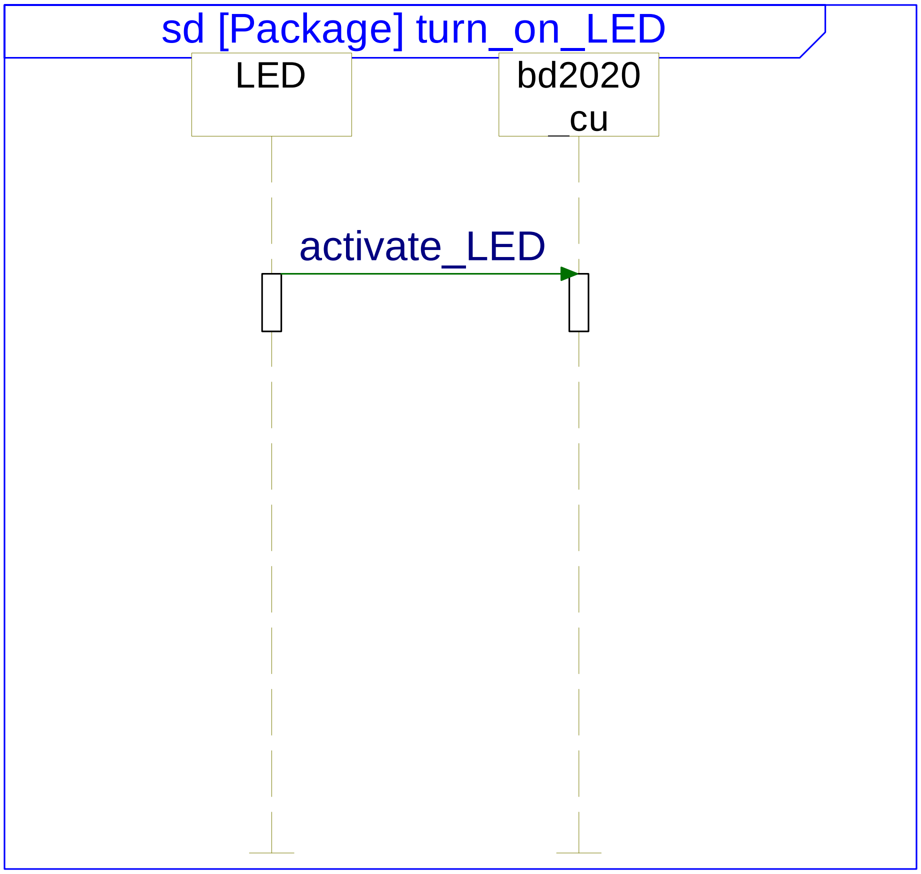

bdd [áBehavioralWhiteboxâ Package] I_BD2013 [block definition diagram_level01]
áInterfaceâ
I_BCM
Operations
judge_ikey_nearby_the_BD
áSystem,blockâ
Hazard_Lamp
áSystem,blockâ
Ikey_Buzzer
áUsageâ
áUsageâ
áUsageâ
Ikey
áRealizationâ
áInterfaceâ
I_BCM
Operations
search_ikey
warning_sound
activate_hazard_lamp
áSystem,bloc
BCM
áRealizationâ
áSystem,blockâ
Sensor_CU
áRealizationâ
áUsageâ
áRealizationâ
áInterfaceâ
I_BCM
Operations
send_request_to_open_BD
trigger_unlock_BD
áInterfaceâ
I_Sensor_CU
Operations
identify_cancel_sw_state
detect_driver_hand
send_detected_signal_of_driver_hand
áUsageâ
Driver
áUsageâ
áUsageâ
áSystem,blockâ
BD_CU
áUsageâ
áSystem,blockâ
Cancel_SW
áRealizationâ
áRealizationâ
áUsageâ
áInterfaceâ
I_BD_CU
Operations
judge_fully_open
BD_unlock
BD_open
áInterfaceâ
I_BD_CU
Operations
send_request_of_turning_on_hazard_lamp
send_request_of_turning_off_hazard_lamp

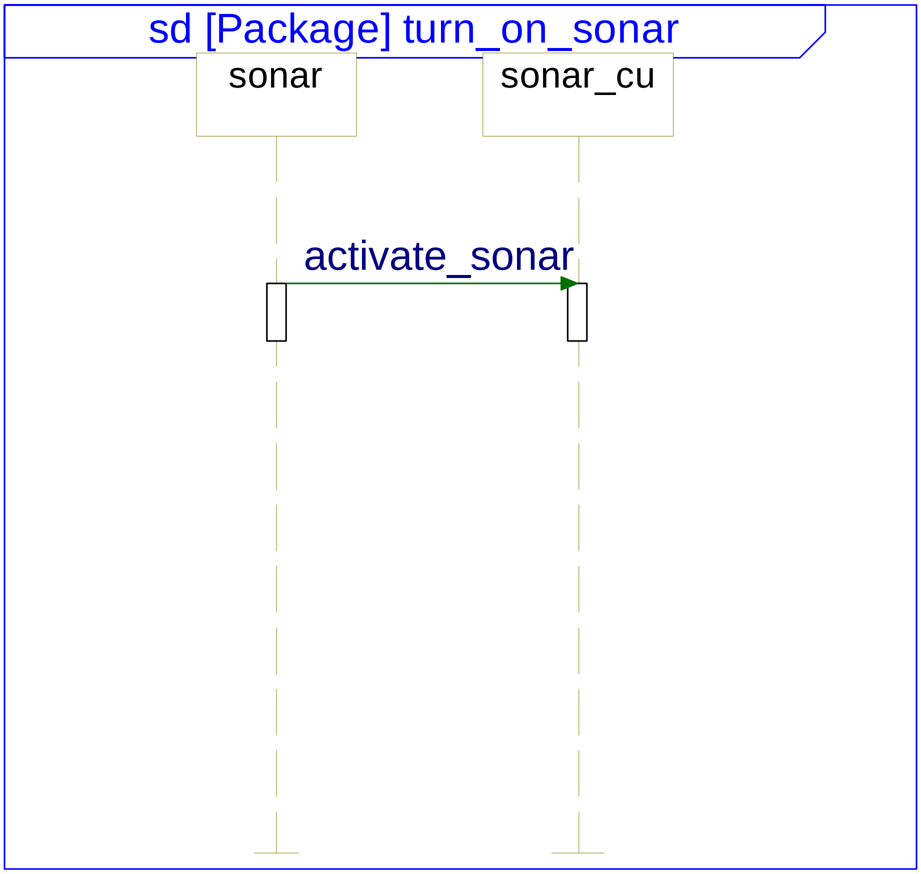
Appendix of “One-Step Back Door System”


uc [Package] Analysis_Level00
BackDoor_2020
Open_ BD
Close _BD
Driver_with_Ikey
Occupants_without_Ikey
Vehicle_2013
ExternalEnvironment
Ground

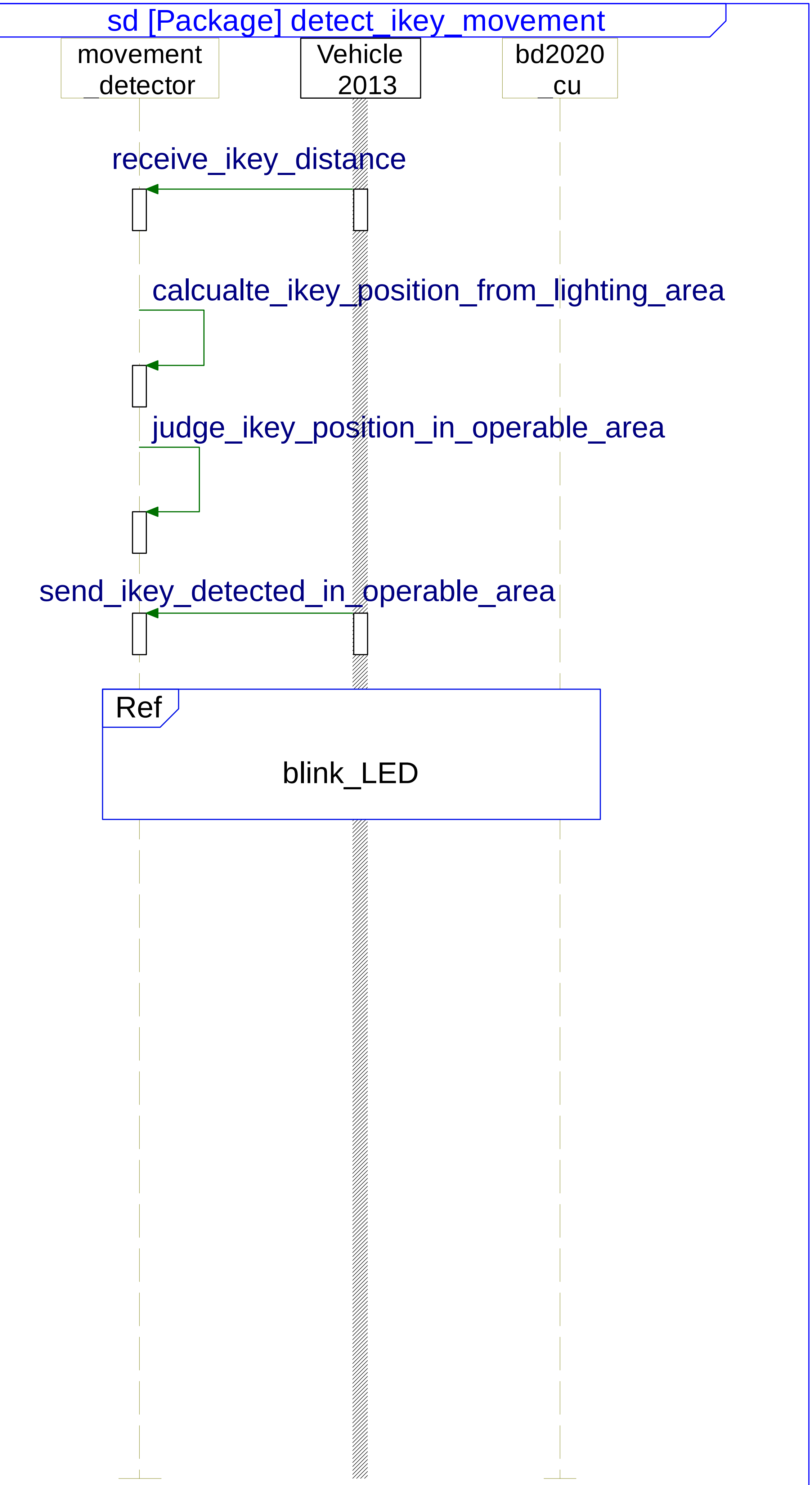

sd [Package] Open_BD
Ground
Driver_with _Ikey
One_Step_Back_ Door_System
Vehicle_2013
loop
[condition]
search_ikey
approaching_to_the_car
detect_ikey
recognize_ikey
switch_to_Hi-speed_search_mode
alt
[before driver foot detected]
parallel
[condition]
turn_on_LED
turn_on_sonar
reflect_LED_light
recognize_LED_light
loop
[condition]
detect_ikey_movement
blink_LED
reflect_LED_blinking_light
recognize_LED_blinking_light
stepping_on_light_area
detect_driver_foot
[driver foot detected]
send_detected_signal_of_driver_foot
judge_open_or_close_BD
unlock_BD
open_BD

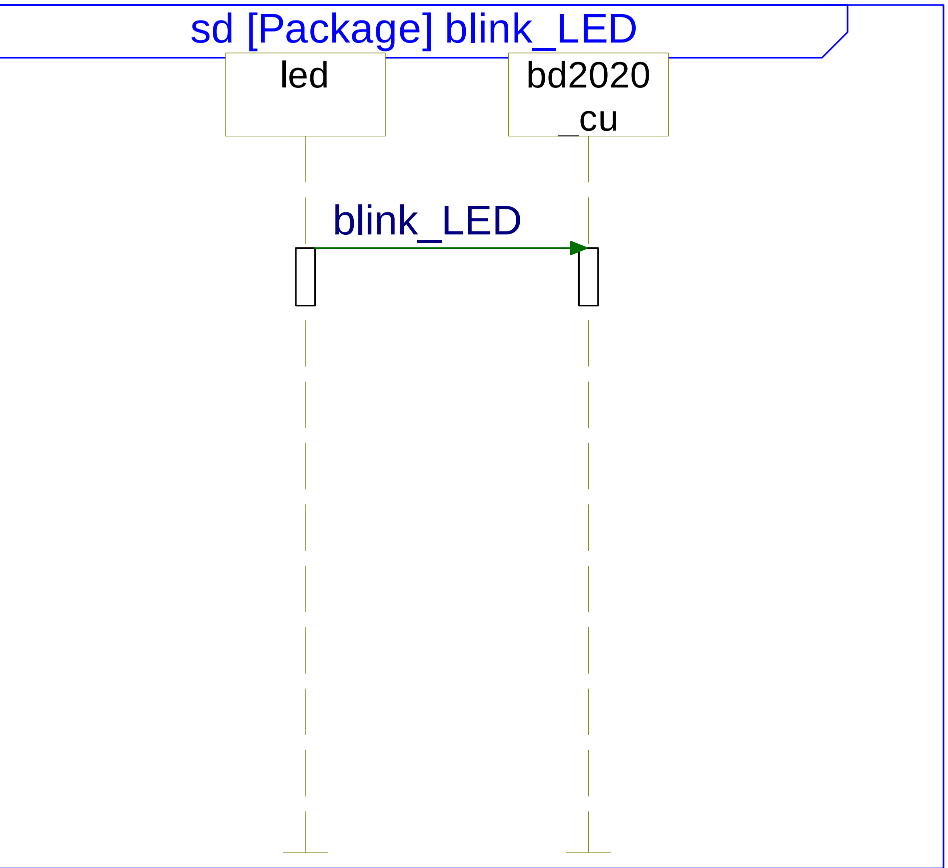

sd [Package] switch_to_Hi-speed_search_mode
bd2020 _cu
sonar_ cu
movement _detector
Vehicl e_201
key_detected_in_Hi-speed_mode_area
Operation_16
change_to_Hi-speed_mode
receive_command_of_switching_to_Hi-speed_mode_from_MD
Ref
turn_on_LED
receive_command_of_switching_to_Hi-speed_mode_from_BD_CU
Ref
turn_on_sonar
opt
[1min passed in Hi-speed mode== TRUE & detect ikey==FALSE for 5sec]
change_to_normal_speed_mode
turn_off_LED

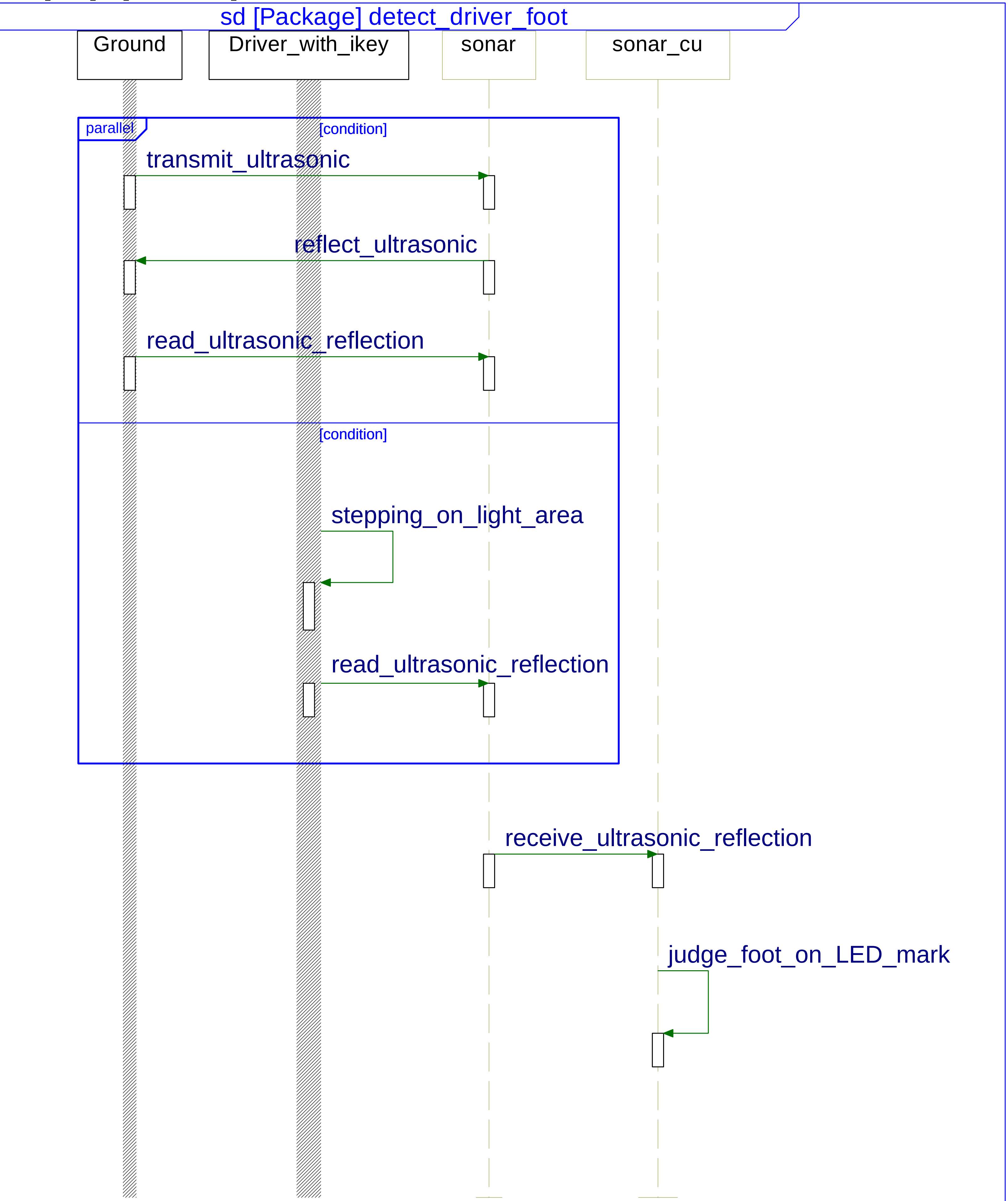

sd [Package] turn_on_LED
LED
bd2020
_cu
activate_LED

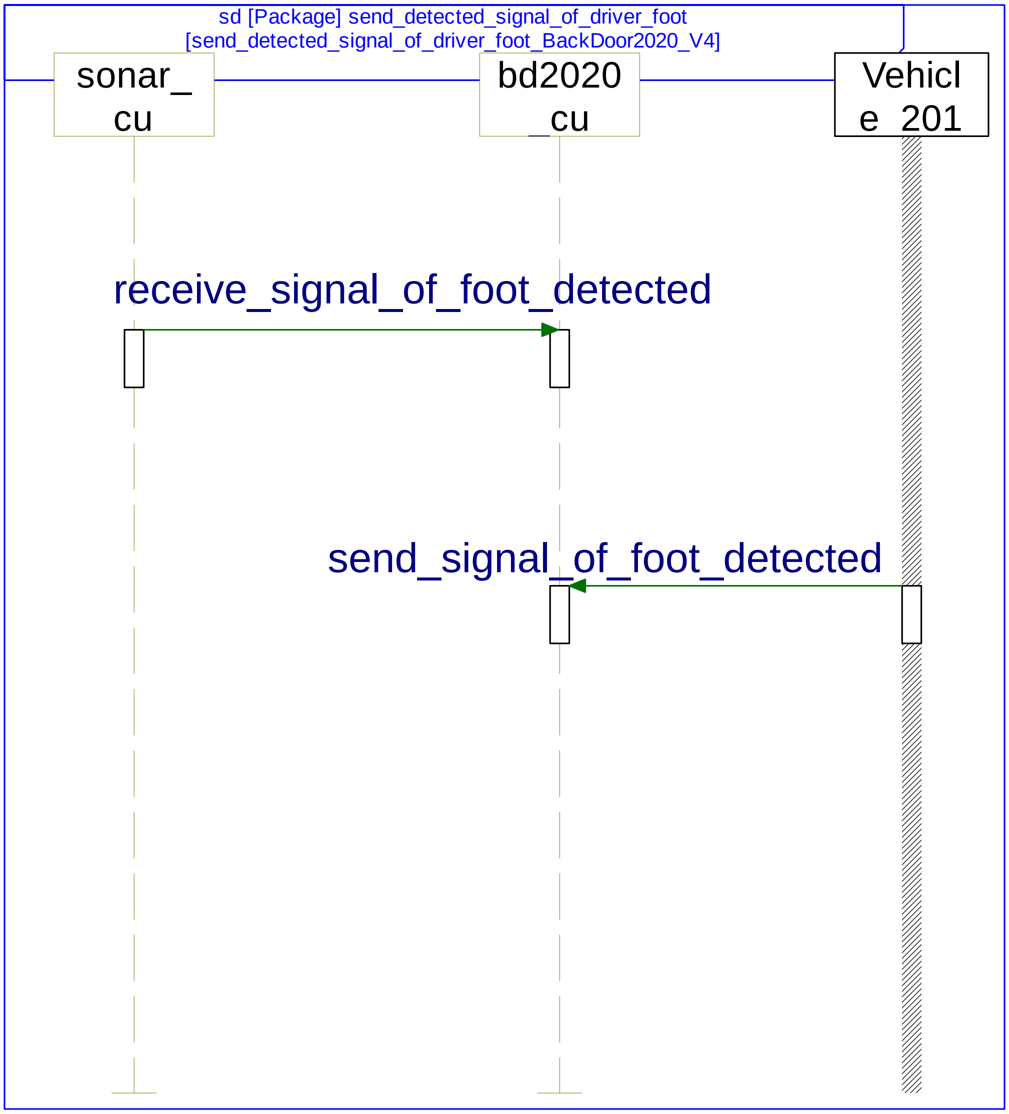

sd [Package] turn_on_sonar
sonar
sonar_cu
activate_sonar

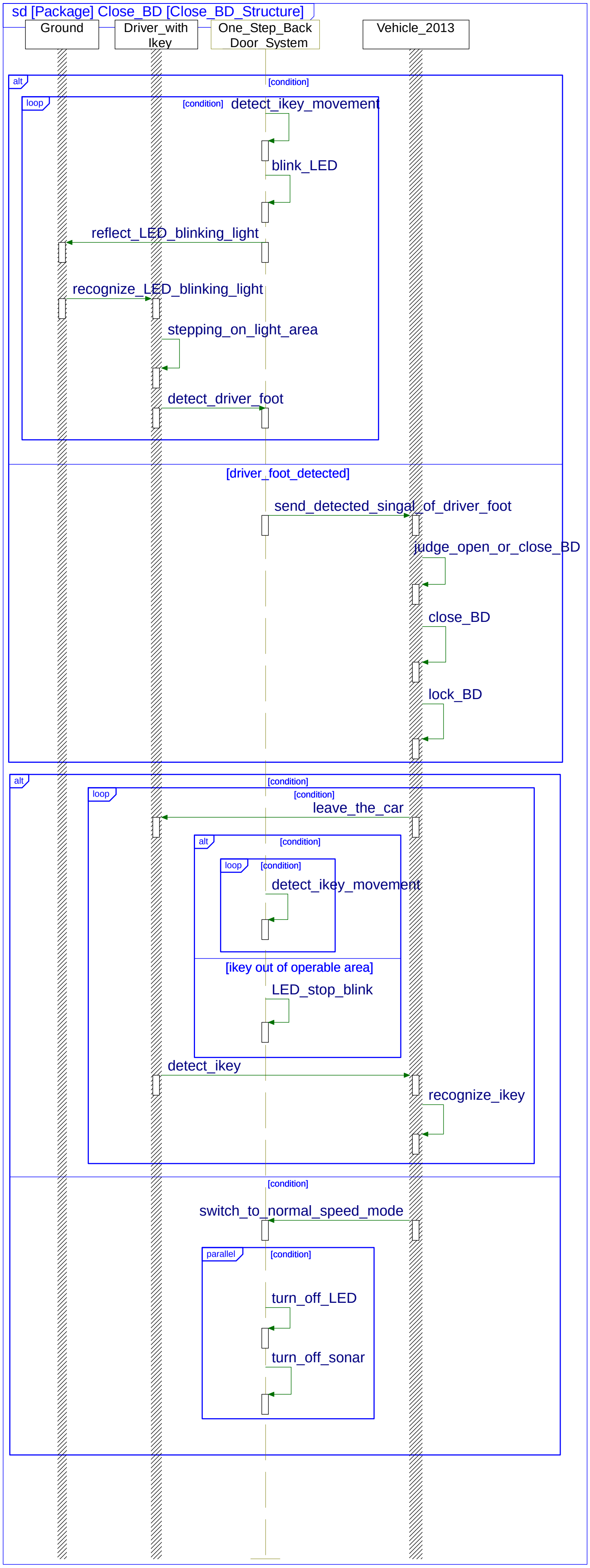
detect_ikey_movement_BackDoor2020_V4

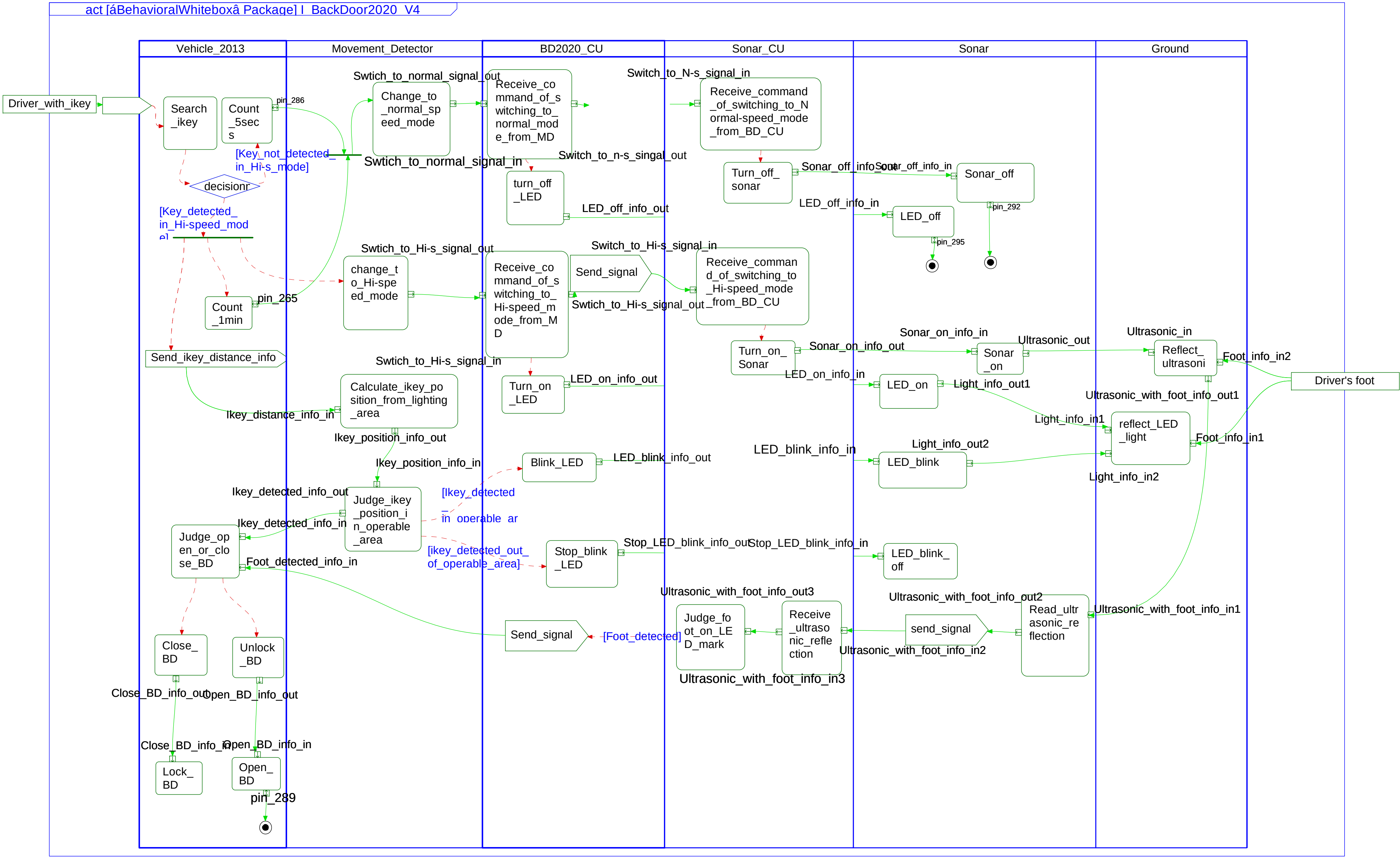


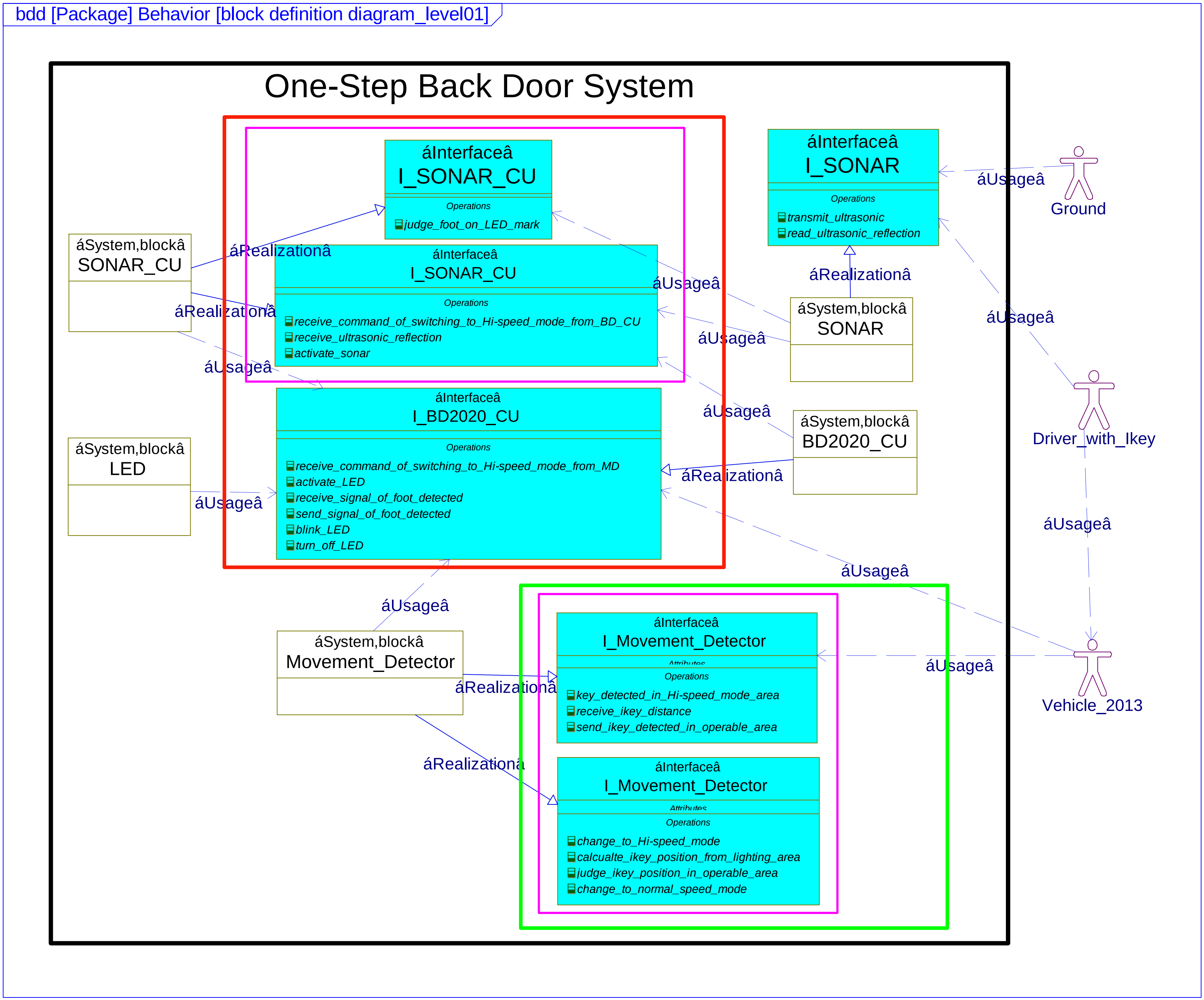

sd [Package] blink_LED
led
bd2020
_cu
blink_LED


sd [Package] detect_driver_foot
Ground
Driver_with_ikey
sonar
sonar_cu
parallel
[condition]
transmit_ultrasonic
reflect_ultrasonic
read_ultrasonic_reflection
[condition]
stepping_on_light_area
read_ultrasonic_reflection
receive_ultrasonic_reflection
judge_foot_on_LED_mark

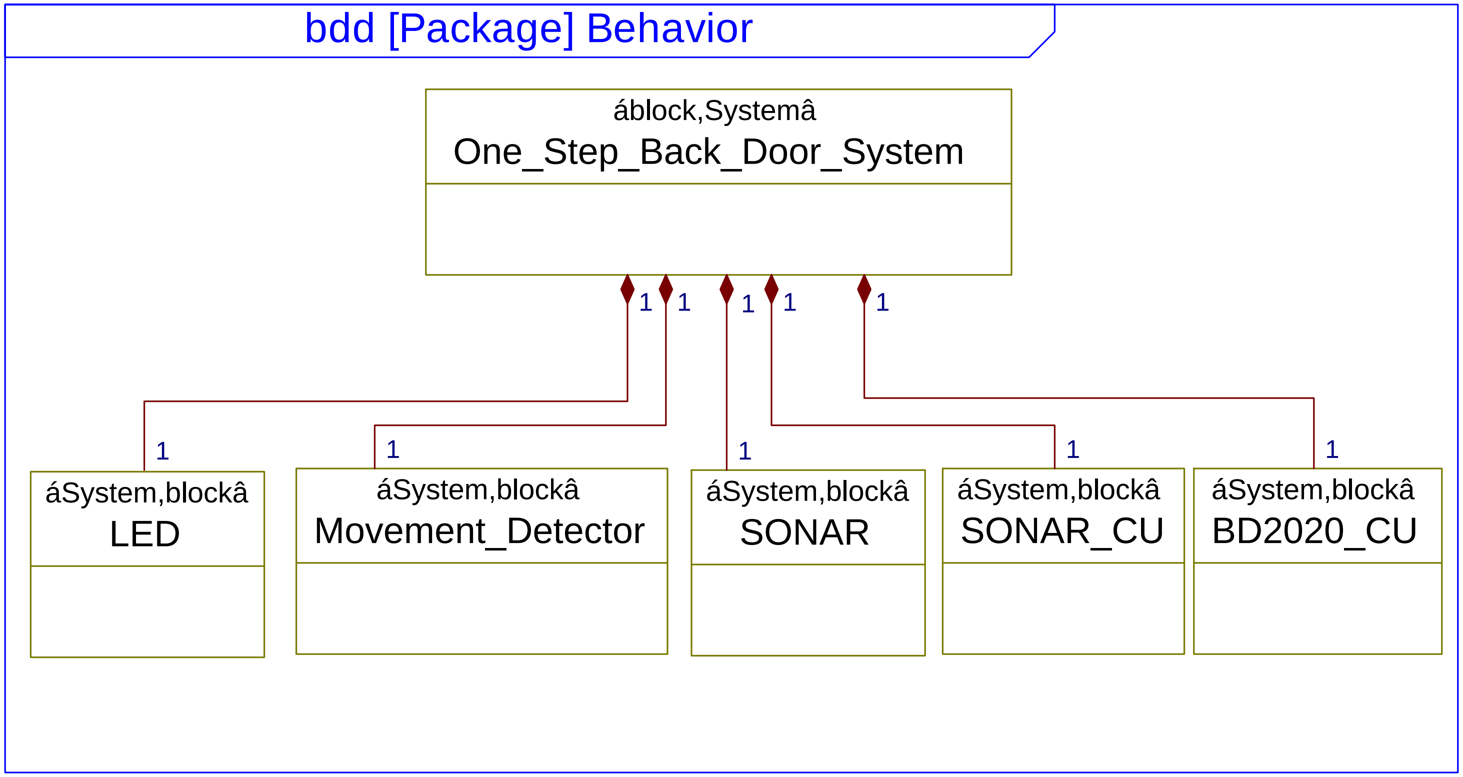

sd [Package] send_detected_signal_of_driver_foot
[send_detected_signal_of_driver_foot_BackDoor2020_V4]
sonar_
cu
bd2020
_cu
Vehicl
e_201
receive_signal_of_foot_detected
send_signal_of_foot_detected

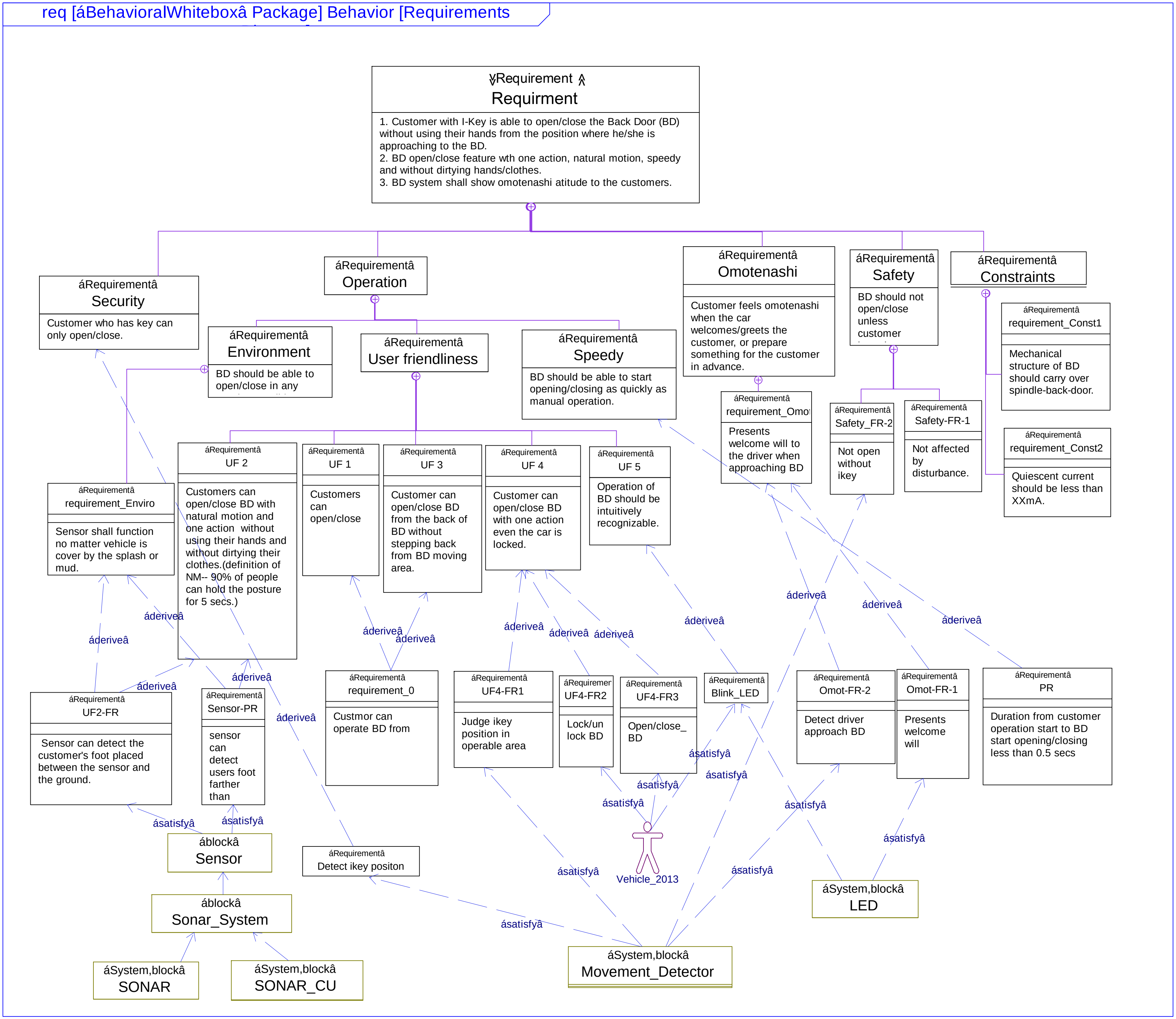

sd [Package] Close_BD [Close_BD_Structure]
Ground
Driver_with Ikey
One_Step_Back Door System
Vehicle_2013
alt
[condition]
loop
[condition]
detect_ikey_movement
blink_LED
reflect_LED_blinking_light
recognize_LED_blinking_light
stepping_on_light_area
detect_driver_foot
[driver_foot_detected]
send_detected_singal_of_driver_foot
judge_open_or_close_BD
close_BD
lock_BD
alt
[condition]
loop
[condition]
leave_the_car
alt
[condition]
loop
[condition]
detect_ikey_movement
[ikey out of operable area]
LED_stop_blink
detect_ikey
recognize_ikey
[condition]
switch_to_normal_speed_mode
parallel
[condition]
turn_off_LED
turn_off_sonar


act [áBehavioralWhiteboxâ Package] I BackDoor2020_V4
Vehicle_2013
Movement_Detector
BD2020_CU
Sonar_CU
Sonar
Ground
Driver_with_ikey
Search _ikey
Count _5sec s
pin_286
[Key_not_detected_ in_Hi-s_mode]
decisionr
[Key_detected_ in_Hi-speed_mod e]
Count _1min
pin_265
Send_ikey_distance_info
Swtich_to_normal_signal_out
Change_to _normal_sp eed_mode
Swtich_to_normal_signal_in
Swtich_to_Hi-s_signal_out
change_t o_Hi-spe ed_mode
Switch_to_Hi-s_signal_in
Calculate_ikey_po sition_from_lighting _area
Ikey_distance_info_in
Ikey_position_info_out
Ikey_position_info_in
Ikey_detected_info_out
Ikey_detected_info_in
Judge_ikey _position_i n_operable _area
Judge_op en_or_clo se_BD
Foot_detected_info_in
[Ikey_detected _ in_operable_ar
[ikey_detected_out_ of_operable_area]
Close_ BD
Unlock _BD
Close_BD_info_out
Open_BD_info_out
Close_BD_info_in
Open_BD_info_in
Lock_ BD
Open_ BD
pin_289
Receive_co mmand_of_s witching_to_ normal_mod e_from_MD
Switch_to_n-s_singal_out
turn_off _LED
LED_off_info_out
Receive_co mmand_of_s witching_to_ Hi-speed_m ode_from_M D
Send_signal
Swtich_to_Hi-s_signal_out
Turn_on _LED
LED_on_info_out
Blink_LED
LED_blink_info_out
Stop_blink _LED
Stop_LED_blink_info_out
Send_signal
[Foot_detected]
Switch_to_N-s_signal_in
Receive_command _of_switching_to_N ormal-speed_mode _from_BD_CU
Turn_off_ sonar
Sonar_off_info_out
Switch_to_Hi-s_signal_in
Receive_comman d_of_switching_to _Hi-speed_mode _from_BD_CU
Turn_on_ Sonar
Sonar_on_info_out
Ultrasonic_with_foot_info_out3
Judge_fo ot_on_LE D_mark
Receive _ultraso nic_refle ction
Ultrasonic_with_foot_info_in3
Sonar_off_info_in
Sonar_off
pin_292
LED_off_info_in
LED_off
pin_295
Sonar_on_info_in
Sonar _on
Ultrasonic_out
LED_on_info_in
LED_on
Light_info_out1
LED_blink_info_in
LED_blink
Light_info_out2
Stop_LED_blink_info_in
LED_blink_ off
Ultrasonic_with_foot_info_out2
send_signal
Ultrasonic_with_foot_info_in2
Read_ultr asonic_re flection
Ultrasonic_in
Reflect_ ultrasoni
Foot_info_in2
Driver's foot
Ultrasonic_with_foot_info_out1
Light_info_in1
reflect_LED _light
Foot_info_in1
Light_info_in2
Ultrasonic_with_foot_info_in1



bdd [Package] Behavior [block definition diagram_level01]
One-Step Back Door System
áInterfaceâ
I_SONAR_CU
Operations
judge_foot_on_LED_mark
áInterfaceâ
I_SONAR_CU
Operations
receive_command_of_switching_to_Hi-speed_mode_from_BD_CU
receive_ultrasonic_reflection
activate_sonar
áInterfaceâ
I_BD2020_CU
Operations
receive_command_of_switching_to_Hi-speed_mode_from_MD
activate_LED
receive_signal_of_foot_detected
send_signal_of_foot_detected
blink_LED
turn_off_LED
áInterfaceâ
I_SONAR
Operations
transmit_ultrasonic
read_ultrasonic_reflection
áSystem,blockâ
SONAR_CU
áSystem,blockâ
LED
áSystem,blockâ
SONAR
áSystem,blockâ
BD2020_CU
áSystem,blockâ
Movement_Detector
áInterfaceâ
I_Movement_Detector
Operations
key_detected_in_Hi-speed_mode_area
receive_ikey_distance
send_ikey_detected_in_operable_area
áInterfaceâ
I_Movement_Detector
Operations
change_to_Hi-speed_mode
calcualte_ikey_position_from_lighting_area
judge_ikey_position_in_operable_area
change_to_normal_speed_mode
áRealizationâ
áUsageâ
Ground
Driver_with_Ikey
Vehicle_2013


bdd [Package] Behavior
áblock,Systemâ
One_Step_Back_Door_System
1
1
1
1
1
1
áSystem,blockâ
LED
1
áSystem,blockâ
Movement_Detector
1
áSystem,blockâ
SONAR
1
áSystem,blockâ
SONAR_CU
1
áSystem,blockâ
BD2020_CU


req [áBehavioralWhiteboxâ Package] Behavior [Requirements
«Requirement»
Requirment
1. Customer with I-Key is able to open/close the Back Door (BD) without using their hands from the position where he/she is approaching to the BD.
2. BD open/close feature wth one action, natural motion, speedy and without dirtying hands/clothes.
3. BD system shall show omotenashi atitude to the customers.
áRequirementâ
Security
Customer who has key can only open/close.
áRequirementâ
Operation
áRequirementâ
Environment
BD should be able to open/close in any
áRequirementâ
User friendliness
áRequirementâ
Speedy
BD should be able to start opening/closing as quickly as manual operation.
áRequirementâ
Omotenashi
Customer feels omotenashi when the car welcomes/greets the customer, or prepare something for the customer in advance.
áRequirementâ
Safety
BD should not open/close unless customer
áRequirementâ
Constraints
áRequirementâ
requirement_Const1
Mechanical structure of BD should carry over spindle-back-door.
áRequirementâ
requirement_Const2
Quiescent current should be less than XXmA.
áRequirementâ
requirement_Omo
Presents welcome will to the driver when approaching BD
áRequirementâ
Safety_FR-2
Not open without ikey
áRequirementâ
Safety-FR-1
Not affected by disturbance.
áRequirementâ
requirement_Enviro
Sensor shall function no matter vehicle is cover by the splash or mud.
áRequirementâ
UF 2
Customers can open/close BD with natural motion and one action without using their hands and without dirtying their clothes.(definition of NM-- 90% of people can hold the posture for 5 secs.)
áRequirementâ
UF 1
Customers can open/close
áRequirementâ
UF 3
Customer can open/close BD from the back of BD without stepping back from BD moving area.
áRequirementâ
UF 4
Customer can open/close BD with one action even the car is locked.
áRequirementâ
UF 5
Operation of BD should be intuitively recognizable.
áderiveâ
áRequirementâ
UF2-FR
Sensor can detect the customer's foot placed between the sensor and the ground.
áRequirementâ
Sensor-PR
sensor can detect users foot farther than
áRequirementâ
requirement_0
Custmor can operate BD from
áRequirementâ
UF4-FR1
Judge ikey position in operable area
áRequiremer
UF4-FR2
Lock/un lock BD
áRequirementâ
UF4-FR3
Open/close_ BD
áRequirementâ
Blink_LED
áRequirementâ
Omot-FR-2
Detect driver approach BD
áRequirementâ
Omot-FR-1
Presents welcome will
áRequirementâ
PR
Duration from customer operation start to BD start opening/closing less than 0.5 secs
ásatisfyâ
áblockâ
Sensor
áRequirementâ
Detect ikey positon
Vehicle_2013
áSystem,blockâ
LED
áblockâ
Sonar_System
áSystem,blockâ
Movement_Detector
áSystem,blockâ
SONAR
áSystem,blockâ
SONAR_CU