Modeling Crisis Management System With the Restricted Use Case Modeling Approach

Gong Zhang[1], Tao Yue[2], and Shaukat Ali[3]

[1]School of Computer Science and Engineering, Beihang University, Beijing, China
zhanggong@sei.buaa.edu.cn

[2]Simula Research Laboratory
P.O. Box 134, Lysaker, Norway
{tao, shaukat}@simula.no

Abstract. Use case modeling is commonly applied to document requirements. Use case specifications (UCSs) are usually structured, unrestricted textual documents complying with a certain template. However, because they remain essentially textual, ambiguities are inevitable. In our previous work, we proposed a new use case modeling approach, named as Restricted Use Case Modeling (RUCM), which is composed of a set of well-defined restriction rules and a new template. The goal was to reduce ambiguity and facilitate automated analysis. In our works, RUCM has been systematically and empirically evaluated through case studies to be easy to apply and leads to higher quality of UML analysis models. In this paper, we modeled the Crisis Management System (CMS) case study using RUCM and our experience proved that RUCM is easy to apply and sufficient to capture the requirements provided for the case study.

Keywords: Use Case; Use Case Modeling; Use Case Template; Restriction Rules.

1 Selection of the Approach

We selected a use case modeling approach called Restricted Use Case Modeling (RUCM) approach [7, 8]. The goal of RUCM approach is to reduce ambiguity of use case models and facilitate automated analysis and generation of other diagrams. RUCM has been systematically and empirically evaluated through case studies to be easy to apply and leads to higher quality of UML analysis models. RUCM is being extended to cover more aspects such as real-time, safety, variability and crosscutting aspects of use case modeling, which all together form the Zen-RUCM framework.

RUCM was initially developed for use case modeling in the object-oriented paradigm. However, in our recent works, the approach has been extended for aspect-orientation as well. In this paper, we only used object-oriented version of the RUCM approach for modeling Crisis Management System (CMS). The approach is for requirements specification and analysis and our tool called aToucan [7, 8] can automatically generate initial analysis models from RUCM models. In this paper, we modeled only a single system since currently we do not support capturing variability in RUCM specifications, which is one of our ongoing works.

2 Restricted Use Case Modeling (RUCM) Approach

In this section, we describe key constructs of the RUCM approach.

2.1 Use case template

Our use case template has eleven first-level fields (1st column in Table 1.). The last four fields are decomposed into second-level fields (2nd column in the last four rows). The last column of each row explains the corresponding field(s). There is no need to further discuss the first seven fields since they are straightforward and commonly encountered in many templates. Below we focus the discussion on the *Basic Flow* and *Alternative Flows* fields.

Table 1. Use case template

Use Case Name	The name of the use case. It usually starts with a verb.	
Brief Description	Summarizes the use case in a short paragraph.	
Precondition	What should be true before the use case is executed.	
Primary Actor	The actor which initiates the use case.	
Secondary Actors	Other actors the system relies on to accomplish the services of the use case.	
Dependency	Include and extend relationships to other use cases.	
Generalization	Generalization relationships to other use cases.	
Basic Flow	Specifies the main successful path, also called "happy path".	
	Steps (numbered)	Flow of events.
	Postcondition	What should be true after the basic flow executes.
Specific Alternative Flows	Applies to one specific step of the reference flow.	
	RFS	A reference flow step number where flow branches from.
	Steps (numbered)	Flow of events.
	Postcondition	What should be true after the alternative flow executes.
Global Alternative Flows	Applies to all the steps of the reference flow.	
	Steps (numbered)	Flow of events.
	Postcondition	What should be true after the alternative flow executes.
Bounded Alternative Flows	Applies to more than one step of the reference flow, but not all of them.	
	RFS	A list of reference flow steps where flow branches from.
	Steps (numbered)	Flow of events.
	Postcondition	What should be true after the alternative flow executes.

A basic flow describes a main successful path. It often does not include any condition or branching [4]. It is recommended to describe separately the conditions and branching in alternative flows. A basic flow is composed of a sequence of steps and a postcondition. Each UCS can only have one basic flow. *Alternative flows* describe all the other scenarios or branches, both success and failure. An alternative flow always depends on a condition occurring in a specific step in a flow of reference, referred to as *reference flow*, and that reference flow is either the basic flow or an alternative flow itself. The branching condition is specified in the reference flow by following restriction rules (R20 and R22—Section 2.2). We refer to steps specifying

such conditions as condition *steps* and the other steps as *action steps*. Similarly to the basic flow, an alternative flow is composed of a sequence of numbered steps. The action steps can be one of the following five interactions (which are reused from [3] except for the fifth):

1) Primary actor → system: the primary actor sends a request and data to the system;
2) System → system: the system validates a request and data;
3) System → system: the system alters its internal state (e.g., recording or modifying something);
4) System → primary actor: the system replies to the primary actor with a result; and
5) System → secondary actor: the system sends requests to a secondary actor.

All steps are numbered sequentially. This implies that each step is completed before the next one is started. If there is a need to express conditions, iterations, or concurrency, then specific keywords, specified as restriction rules should be applied.

We classify alternative flows into three types: specific, global, and bounded alternative flows. This classification is adapted from [1]. A *specific alternative flow* is an alternative flow that refers to a specific step in the reference flow. A *bounded alternative flow* is a flow that refers to more than one step in the reference flow–consecutive steps or not. A *global alternative flow* (called *general alternative flow* in [1]) is an alternative flow that refers to any step in the reference flow. Distinguishing different types of alternative flows makes interactions between the reference flow and its alternative flows much clearer. For specific and bounded alternative flows, a RFS (Reference Flow Step) section, specified as rule R19, is used to specify one or more (reference flow) step numbers. Whether and where the flow merges back to the reference flow or terminates the use case must be specified as the last step of the alternative flow. Similarly to the branching condition, merging and termination are specified by following restriction rules (R24 and R25—Section 2.2). By doing so, we can avoid potential ambiguity in UCSs caused by unclear specification of interactions between the basic flow and its corresponding alternative flows. Each alternative flow must have a postcondition (enforced by restriction rule R26—Section 2.2).

It is usual to provide a postcondition describing a constraint that must be true when a use case terminates. If the use case contains alternative flows, then the postcondition of the use case should describe not only what must be true when the basic flow terminates but also what must be true when each alternative flow terminates. The branching condition to each alternative flow is then necessarily part of the postcondition (to distinguish the different possible results). In such a case, the postcondition becomes complex and the branching condition for each alternative flow is redundantly described (both in the steps of flows and the postcondition), which therefore increases the risk of ambiguity in UCSs. Our template enforces that each flow of events (both basic flow and alternative flows) of a UCS contains its own postcondition and therefore avoids such ambiguity.

2.2 Restriction rules

The restriction rules are classified into two groups: restrictions on the use of natural language, and restrictions enforcing the use of specific keywords for specifying

control structures. The first group of restrictions is further divided into two categories according to their location of application (see below). Each restriction rule is assigned a unique number.

Table 2. Restrictions (R1-R16)

#	Description	Explanation
R1	The subject of a sentence in basic and alternative flows should be the system or an actor.	Enforce describing flows of events correctly. These rules conform to our use case template (the five interactions).
R2	Describe the flow of events sequentially.	
R3	Actor-to-actor interactions are not allowed.	
R4	Describe one action per sentence. (Avoid compound predicates.)	Otherwise it is hard to decide the sequence of multiple actions in a sentence.
R5	Use present tense only.	Enforce describing what the system does, rather than what it will do or what it has done.
R6	Use active voice rather than passive voice.	Enforce explicitly showing the subject and/or object(s) of a sentence.
R7	Clearly describe the interaction between the system and actors without omitting its sender and receiver.	
R8	Use declarative sentences only. "Is the system idle?" is a non-declarative sentence.	Commonly required for writing UCSs.
R9	Use words in a consistent way.	Keep one term to describe one thing.
R10	Don't use modal verbs (e.g., *might*)	Modal verbs and adverbs usually indicate uncertainty; therefore metrics should be used if possible.
R11	Avoid adverbs (e.g., *very*).	
R12	Use simple sentences only. A simple sentence must contain only one subject and one predicate.	Reduce ambiguity and facilitate automated NL parsing.
R13	Don't use negative adverb and adjective (e.g., *hardly*, *never*), but it is allowed to use *not* or *no*.	
R14	Don't use pronouns (e.g. *he*, *this*).	
R15	Don't use participle phrases as adverbial modifier. For example, the italic-font part of the sentence "ATM is idle, *displaying a Welcome message*", is a participle phrase.	
R16	Use "the system" to refer to the system under design consistently.	Keep one term to describe the system; therefore reduce ambiguity.

Restriction rules R1-R16 in Table 2. constrain the use of natural language: the table explains why they are needed to reduce ambiguity. Rules R1-R7 apply only to action steps; they do not apply to condition steps, preconditions or postconditions. Rules R8-R16 apply to all sentences in a UCS: action steps, condition steps, preconditions, postconditions, and sentences in the brief description. Rules R8-R11 and R16 aim to reduce ambiguity of UCSs; the remaining rules (R12-R15) can help reduce ambiguity and also facilitate automated generation of analysis models. Recall that facilitating automated derivation of initial analysis models from UCSs is also one of our goals, though this is not discussed in this paper. These two sets of restrictions are thought to be good practice for writing clear and concise UCSs (e.g., [1, 3, 5]) except for R13 and R15. We add these two rules because we observed that negative

adverbs, negative adjectives, and participle phrases are very difficult to parse by natural language parsers. R9 requires using words consistently to document UCSs. A common approach to do so is to use a domain model and glossary (e.g., [4], [2]) as a basis to write UCSs.

The remaining ten restriction rules (R17-R26) constrain the use of control structures, except R26 that specifies that each basic flow and alternative flow should have its own postcondition. R17 and R18 specify keywords to describe use case dependencies include and extend. Sentences containing the keywords INCLUDE USE CASE and EXTENDED BY USE CASE are referred to as dependency sentences. R19 specifies keyword RFS, which is used in a specific (or bounded) alternative flow to refer to a step number (or a set of step numbers) of a reference flow that this alternative flow branches from. Rules R20-R23 specify the keywords used to specify conditional logic sentences (IF-THEN-ELSE-ELSEIF-ENDIF), concurrency sentences (MEANWHILE), condition checking sentences (VALIDATES THAT), and iteration sentences (DO-UNTIL), respectively. Keyword VALIDATES THAT (R22) specifies that a condition is evaluated by the system and must be true to proceed to the next step. This rule also requires that an alternative flow describing what happens when the validation fails (the condition does not hold) be described. Rules R24 and R25 specify that an alternative flow ends with a step using either keyword ABORT or keyword RESUME STEP, thereby clearly specifying whether the flow returns back to the reference flow and where (using keyword RESUME STEP followed by a returning step number) or terminates (using keyword ABORT).

R17-R21 and R23 have been proposed in the literature and we reused them with some variation. R22, R24 and R25 are newly proposed in this work for the purpose of making the whole set of restrictions as complete as possible so that flows of events and interactions between the basic flow and the alternatives can be clearly and concisely specified. Applying these rules helps reducing ambiguity in UCSs, and also facilitates automated NL processing (e.g., correctly parse sentences with our specified keywords) and the generation of analysis models, especially sequence diagrams.

The detailed description of RUCM is provided in [7, 8]. For transforming RUCM models to UML class, sequence, activity and state machine diagrams, please refer to our previous work [6, 9, 10].

3 RUCM Model for the Crisis Management System

In the requirements specification document of CMS, there is only one use case in the provided case study and thus we don't show the use case diagram. The RUCM model for CMS is available in [11], and it is also presented in Appendix. In our model, we describe how we used the RUCM template and restrictions to specify the use case specification for use case "Communicate with Other Coordinator". In addition, we demonstrate how we use RUCM to specify non-functional requirements provided in the case study document.

References

1. Bittner K. and Spence I.: Use Case Modeling. Addison-Wesley Boston (2002)
2. Bruegge B. and Dutoit A. H.: Object-Oriented Software Engineering Using UML, Patterns, and Java. Prentice Hall, 2nd Edition (2004)
3. Cockburn A.: Writing effective use cases. Addison-Wesley Boston (2001)
4. Larman C.: Applying UML and Patterns. Prentice-Hall, 3rd Edition (2004)
5. Schneider G. and Winters J. P.: Applying use cases: a practical guide. Object Technology, Addison-Wesley (1998)
6. Yue T., Ali S. and Briand L.: Automated Transition from Use Cases to UML State Machines to Support State-based Testing. Proc. The Seventh European Conference on Modeling Foundations and Applications (ECMFA) (2011)
7. Yue T., Briand L. and Labiche Y.: Facilitating the Transition from Use Case Models to Analysis Models: Approach and Experiments. ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 22 (1). (2013)
8. Yue T., Briand L. C. and Labiche Y.: A Use Case Modeling Approach to Facilitate the Transition Towards Analysis Models: Concepts and Empirical Evaluation. Proc. MODELS2009, 5795/2009, pp. 484-498 (2009)
9. Yue T., Briand L. C. and Labiche Y.: An Automated Approach to Transform Use Cases into Activity Diagrams. Proc. 6th ECMFA, LNCS 6138, pp. 337-353 (2010)
10. Yue T., Briand L. C. and Labiche Y.: Automatically Deriving a UML Analysis Model from a Use Case Model. Simula Research Laboratory, Technical Report 2010-15 (2010)
11. Gong Z., Tao Y. and Shaukat A.: Modeling Crisis Management System with the Restricted Use Case Modeling Approach. Repository for Model-Driven Development (ReMoDD), October 21. Retrieved October 28, 2013, `http://www.cs.colostate.edu/remodd/v1/content/modeling-crisis-management-system-restricted-use-case-modeling-approach`

Appendix

Use Case: Communicate with Other Coordinator

The RUCM template and restriction rules were applied to specify the use case specification for this use case as shown below. The use case specification is composed of one basic flow, six specific alternative flows and two global alternative flows. The control branches from the basic flow to all the alternative flows are clearly specified using RUCM. Restriction rules on natural language were applied to make English sentences more precise. Our own interpretation on some of the use case specification provided in the workshop case study is reflected as part of the model. Notice that we specified postconditions for each of the flow of events.

Use Case Name	Communicate with Other Coordinator
Brief Description	To resolve a crisis situation as quickly and cost-effectively as possible in cooperation with other coordinators. Coordination is required as PoliceOfficer may enable Fireman to reach the crisis location faster. This can be done by for example escorting the fire trucks or by creating roadblocks.
Precondition	PoliceStationCoordinator and FireStationCoordinator are aware of the crisis. PoliceStationCoordinator and FireStationCoordinator have not established communication.
Primary Actor	PoliceStationCoordinator
Secondary Actors	FireStationCoordinator
Dependency	None
Generalization	None

Basic Flow (Untitled) ▼	Steps	
	1	PoliceStationCoordinator communicates with FireStationCoordinator to establish communication.
	2	PoliceStationCoordinator exchanges crisis details with FireStationCoordinator.
	3	PoliceStationCoordinator communicates with FireStationCoordinator to define the number of fire trucks and police vehicles to deploy.
	4	**IF** the negotiation duration is within a predefined limit **THEN**
	5	**DO**
	6	PoliceStationCoordinator proposes one route for a fire truck and one route for a police vehicle to reach crisis location.
	7	**UNTIL** All the fire trucks and police vehicles have routes defined.
	8	**IF** FireStationCoordinator agrees to the routes proposed by PoliceStationCoordinator **THEN**
	9	PoliceStationCoordinator communicates with FireStationCoordinator about dispatched vehicles according to the route plan.
	10	**ENDIF**
	11	**ENDIF**
	12	**DO**
	13	PoliceStationCoordinator communicates with FireStationCoordinator to know the arrivals of a police vehicle and fire truck at a targeted location.
	14	**IF** the police vehicle or the fire truck has arrived the targeted location **THEN**
	15	PoliceStationCoordinator communicates with FireStationCoordinator to know the completion of the objective of the police vehicle or fire truck.
	16	**ENDIF**
	17	**UNTIL** All the fire trucks and police vehicles are proceed.
	18	PoliceStationCoordinator and FireStationCoordinator agree to close the crisis.
	Postcondition	The crisis has been closed.

Specific Alternative Flow "alt1" ▼	**RFS** 4	
	1	**ELSE**
	2	The system records a Timeout.
	3	The system alterts PoliceStationCoordinator and FireStationCoordinator about the timeout.
	4	**RESUME STEP** 5 **MEANWHILE** PoliceStationCoordinator and FireStationCoordinator report the reason for timeout.
	5	**ENDIF**
	Postcondition	A timeout is recorded in the system.

Specific Alternative Flow "alt2" ▼	**RFS** 8	
	1	**ELSE**
	2	PoliceStationCoordinator removes the disagreed routes from the proposed routes.
	3	PoliceStationCoordinator **VALIDATES THAT** there are routes available to be proposed.
	4	**RESUME STEP** 6
	5	**ENDIF**
	Postcondition	A list of agreed routes is updated.

Specific Alternative Flow "alt3" ▼	**RFS** alt2 3	
	1	PoliceStationCoordinator informs FireStationCoordinator not to coordinate routes but to exchange information on vehicle locations and crisis details.
	2	**RESUME STEP** 9
	Postcondition	Routes for some police vehicles and fire stations are not coordinated.

Specific Alternative Flow "alt4" ▼	**RFS** 14	
	1	**ELSEIF** a police vehicle or fire truck breaks down on the way to the destination **THEN**
	2	PoliceStationCoordinator or FireStationCooridinator informs each other of the new ETA of the police vehicle or the fire truck **MEANWHILE** PoliceStationCoordinator and FireStationCoordinator assign a route to the replacement police vehicle or fire truck and dispatch it.
	3	**RESUME STEP** 13
	4	**ENDIF**
	Postcondition	The status of the location of the police vehicle or fire truck is updated. A new police vehicle or fire truck is dispatched.

Specific Alternative Flow "alt5" ▼	**RFS** 14	
	1	**ELSEIF** there is traffic of blocked routes **THEN**
	2	**RESUME STEP** 4
	3	**ENDIF**
	Postcondition	The status of the location of the police vehicle or fire truck is updated.

Specific Alternative Flow "alt6" ▼	**RFS** 14	
	1	**ELSEIF** the crisis is less severe than expected **THEN**
	2	PoliceStationCoordinator and FireStationCoordinator inform each other to recall one or more police vehicles or fire trucks.
	3	**RESUME STEP** 13
	4	**ENDIF**
	Postcondition	The status of the crisis is updated. One or more police vehicles or fire trucks are recalled.

Global Alternative Flow (Untitled) ▼	Communication is not available.
1	PoliceStationCoordinator and FireStationCoordinator continue to address the crisis individually.
2	**ABORT**
Postcondition	No communication is going on between PoliceStationCoordinator and FireStationCoordinator.

Global Alternative Flow (Untitled) ▼	Communication has been restored after a period of unavailable communication.
1	**IF** the crisis has been resolved **THEN**
2	**RESUME STEP** 18
3	**ENDIF**
4	**IF** the communication between PoliceStationCoordinator and FireStationCoordinator has not yet been established **THEN**
5	**RESUME STEP** 1
6	**ENDIF**
7	**IF** the route agreement has been reached **THEN**
8	PoliceStationCoordinator and FireStationCoordinator exchange information on routes established for police vehicles and fire trucks, locations of police vehicles and fire trucks, and the status of the crisis.
9	**DO**
10	**IF** a police vehicle or fire truck has been dispatched but not arrived **THEN**
11	**RESUME STEP** 9
12	**ENDIF**
13	**IF** a police vehicle or fire truck has arrived at the targeted location **THEN**
14	**RESUME STEP** 13
15	**ENDIF**
16	**IF** a police vehicle or fire truck has completed its objective **THEN**
17	**RESUME STEP** 15
18	**ENDIF**
19	**UNTIL** the status of all the police vehicles and fire trucks are updated.
20	**ENDIF**
Postcondition	The status of the crisis is updated. The status of all the police vehicles and fire trucks dispatched is updated.

Non-functional requirements

This use case is used for capture non-functional requirements of the system. These non- functional requirements are global for all the use cases in the use case model, though currently there is only one use case in the use case model. Therefore, we used Global Alternative Flows to capture each non-functional requirement provided in the case study description. Notice that for each non- functional requirement, RUCM naturally provides a mechanism to define an exception handler alternative flow, from which one can describe the scenario when the non-functional requirement is not met. As for specifying functional requirements, RUCM also enforces each flow of events to their own postconditions. By modeling this case study, we are confident that RUCM can also be used to either directly or with some adaption to capture non-functional properties.

Use Case Name	Non-functional requirements
Brief Description	This use case is used for capture non-functional requirements of the system. These non-functional requirements is global for all the use cases in the use case model.
Precondition	None
Primary Actor	None
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow "basicFlow" ▼	Steps
1	**RFS** CarCrashManagementSystems
Postcondition	None

Global Alternative Flow "IntegrityReq1" ▼	The system is active in basicFlow CURRENTSTEP.
1	The system **VALIDATES THAT** that the integrity of the communication regarding the locations of the crisis, police vehicles and fire trucks between PoliceStationCoordinator and FireStationCoordinator is preserved 99.99% of the time.
Postcondition	The integrity of the communication is validated. The integrity of the communication is preserved.

Specific Alternative Flow	RFS IntegrityReq1 1
"Exception Handler Of IntegrityReq1"▼	1 RESUME STEP basicFlow CURRENTSTEP.
	Postcondition The system is active in basicFlow CURRENTSTEP.

Global Alternative Flow	The system is active in basicFLOW CURRENTSTEP.
"IntegrityReq2"▼	1 The system VALIDATES THAT the integrity of all other data transmitted between PoliceStationCoordinator and FireStationCoordinator is preserved 95% of the time.
	Postcondition The integrity of the communication is validated. The integrity of the communication is preserved.

Specific Alternative Flow	RFS IntegrityReq2 1
"Exception Handler Of IntegrityReq2"▼	1 RESUME STEP basicFlow CURRENTSTEP.
	Postcondition The system is active in basicFlow CURRENTSTEP.

Global Alternative Flow	Guard Condition
"AvailabilityReq1"▼	1 IF There is at least one crisis active THEN
	2 The system VALIDATES THAT the crisis details, the route plan of the fire station and the police station, and the information related to the identification of PoliceStationCoordinator and FireStationCoordinator is available within 5 minutes.
	3 ELSE
	4 The system VALIDATES THAT the crisis details and the route plan of the fire station and the police station is available within 30 mins for every 48 hours.
	5 ENDIF
	Postcondition The availability of the system is validated.

Specific Alternative Flow	RFS AvailabilityReq1 1
" Exception Handler Of AvailabilityReq1 1"▼	1 The system informs PoliceStationCoordinator and FireStationCoordinator about the situation.
	2 RESUME STEP AvailabilityReq1 2.
	Postcondition PoliceStationCoordinator and FireStationCoordinator are informed of the status of the current availability of the system.

Specific Alternative Flow	RFS AvailabilityReq1 4
"Exception Handler Of AvailabilityReq1 2"▼	1 The system displays a warning MEANWHILE the system records the problem in the database.
	2 RESUME STEP AvailabilityReq1 4.
	Postcondition The problem is recorded in the database.

Global Alternative Flow	Guard Condition
"PerformanceReq1"▼	1 The system VALIDATES THAT the system shall respond to user requests within 5 seconds 95% of the time.
	2 The system VALIDATES THAT the system shall respond to user requests within 30 seconds 99.99% of the time.
	Postcondition The performance of the system is validated.

Specific Alternative Flow	RFS PerformanceReq1 1
"Exception Handler Of PerformanceReq1"▼	1 The system displays a warning.
	2 The system initiates the recovery procedure.
	Postcondition The status of the system performance is updated. Recovery procedure is initiated.