รายการอ้างอิง

1. Janathan Bowen Formal Specification and Documentation Using Z: A Case Study Approach. International Thomson Publishing,1996.
2. J- R Abrial. The B – Book Assigning Programs to Meanings. Cambridge University Press, 1996.
3. Roger S. Pressman. Software Engineering: a practitioner's approach. Sixth Edition, (n.p) : McGraw-Hill, 2005
4. Grady Booch, Jame Rumbaugh and Ivar Jacobson. The Unified Modeling Language User Guide. Addison Wesley, 1998.
5. Alan Dennis, Barbara Haley Wixom and David Tegargen. System Analysis and Design An Object Oriented Approach with UML. John Wiley & Son, 2002.
6. Jim Arlow, Ila Neustadt. UML and The Unified Process Pratical Object – Oriented Analysis and Design. Addison Wesley, 2002
7. Bernd Oestereich. Developing Software with UML Object – Oriented Analysis and Design in Practice. Addison Wesley, 1998.
8. Eric Meyer, Jeanine Souquières, A Systematic approach to transform OMT Diagrams to a B Specification. World Congress on Formal Methods in the Development of Computing Systems, 1999
9. Hung Ledang and Jeanine Souquières, Integrating UML and B Specification Techniques. Workshop at Informatik, 2001.
10. Hung Ledang and Jeanine Souquières, Modeling Class Operations in B: a case study on the pump component. Technical Report A01-R-011. Laboratory Lorrian de Recherche en Informatique et ses Applications, 2001.
11. Colin Snook and Michael Butler, Tool – Supported Use of UML for Developing B Project. In Proceeding of Declarative System & Software Engineering (DSSE), 2002.
12. Rafael Marcano-Kamenoff, Formalizing Pattern Applicability – An Approach based on UML and B (Dissertation Abstract). In the 17 th IEEE International Conference on Automated Software Engineering (ASE), 2000.

13. J. Orawan, Consistency Analysis of Use Case and State Diagram Using B Method. A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science Department of Computer Science, Thammasat University, 2003

ภาคผนวก

ภาคผนวก ก

บทความเรื่อง "การตรวจสอบความสอดคล้องแผนภาพคลาสและแผนภาพซีเควนซ์ด้วยหลักการของภาษาบี" นำเสนอในงานประชุมวิชาการ The 1st KMITL International Conference on Science and Applied Science 2006 จัดที่โรงแรม สวิสโซเทล เลอ คองคอร์ด จังหวัดกรุงเทพมหานคร ระหว่างวันที่ 8 – 10 มีนาคม พ.ศ. 2549 บทคัดย่อนี้ตีพิมพ์ไว้ใน Proceedings of the KMITL International Conference on Science and Applied Science 2006

Consistency Check of Class Diagram and Sequence Diagrams using B-Method

Waitaya Sricharunrat[1] and Wiwat Vatanawood[2]

Department of Computer Engineering, Faculty of Engineering,

Chulalongkorn University, Pathumwan, Bangkok, Thailand.

Waitaya.S@Student.chula.ac.th[1], wiwat@chula.ac.th[2]

ABSTRACT

This paper proposes a systematic mean of consistency check for UML class diagram and its related sequence diagrams representing the critical scenarios using B-Method. The B-Method is a formal specification modeling which is used to describe the semantics of system in terms of mathematical notations – set theory and first-order predicate logic. In our approach, a class diagram and its related sequence diagrams are formally translated into B Abstract Machine (BAM) using a set of our translation rules. Our translation rules generate the semantics of both structural and behavioral properties of the UML class diagram and sequence diagrams.

This paper focuses on two parts. Firstly, the formalization of the UML class diagram - a collection of classes and their relations such as association, aggregation, composition, generalization or inheritance, is investigated and defined for the structural property. Secondly, the formalization of UML sequence diagrams – a collection of scenarios which illustrate the major interactions between related classes as to achieve a specific goal, is defined for the behavioral property and verified against their original structure in class diagram. Moreover, we formally define the complex operations within the critical sequence diagrams by exploiting the calling-called dependency between class operations from Hung Ledang's work. The formal specification in BAM is finally generated and verified by B-Toolkit.

Keywords: UML, Class Diagram, Sequence Diagram, B-Method, Formal Specifications Modeling, B Abstract Machine

1. INTRODUCTION

UML (Unified Modeling Language) is the language used to analyze and design software system. Both developers and users usually prepare a UML class diagram and its related sequence diagrams to describe the structural and behavioral properties of the target software system. Practically, they have to finish a large number of UML class diagrams and sequence diagrams and the verification of the UML diagrams must be tediously conducted to walkthrough the consistency among the diagrams. An alternative of the systematic approach to deal with these problems is to exploit the formal specifications modeling to ease the consistency checking. The formal specification modeling is a formal description of a software system in terms of mathematical notations as to help prove of syntactical and semantic correctness [1]. Therefore, both developers and users understand the software system model in the same way. [2], [3]

This paper proposes an approach to formally define class diagram and sequence diagrams into formal specifications called BAM (B Abstract Machine). Firstly, the approach will translates all attributes of classes from class diagram and relationships among those classes. Secondly, such approach will translate operations of class diagram from critical scenarios of sequence diagrams. Our approach applies Hung Ledang's Calling-Called Dependency concepts on how to build the hierarchical structure of the related class operations

[4], [5]. The result of the translation in BAM statements helps to check the consistency of software system which is represented by class diagram and sequence diagrams.

This paper is organized as follows. Section 2 presents overview of backgrounds. Section 3 explains overview of our purposed scheme, while a case study of cash money transfer bank will be described in section 4. Section 5 illustrates the formalization of UML class diagram and sequence diagrams. Finally, in section 6, the conclusion of this work is discussed.

2. Overview of Backgrounds

2.1 The Calling-Called Dependency between Class Operations [4], [5]

Hung Ledang proposed an approach to build the relationship among the class operations into hierarchical tree, called the Calling-Called dependency between class operations, to help construct a BAM. Hung Ledang divided the class operations regarding their calling behaviors into 2 groups; 1) Non-Basic operations are the class operations that typically call the other operations during their run-time activities and 2) Basic operations are the operations that typically not call the other operations.

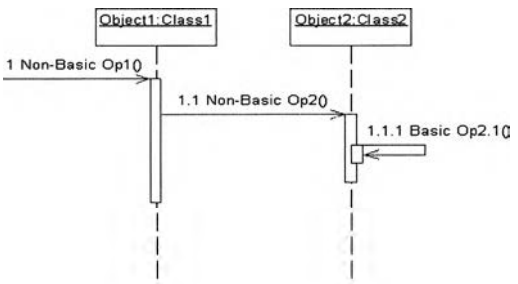


Figure 1.Non-Basic and Basic Operations from critical scenario of sequence diagram

In figure 1, both operation Op1( ) and Op2( ) are Non-Basic operations that invoke the other operations at least once. For example, the operation Op1( ) calls Op2( ) once while the operation Op2( ) also calls Op2.1( ). The operation Op2.1( ) is Basic operation that does not call the other operations at all. The Non-Basic operation Op1( ) is a calling-operation, and the Basic Op2.1( ) is called-operation. While the Non-Basic operation Op2( ) is both calling-operation and called-operation as shown in figure 2.

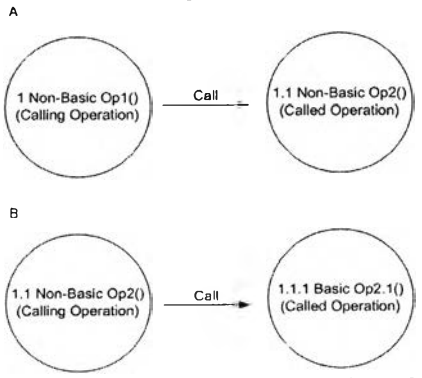
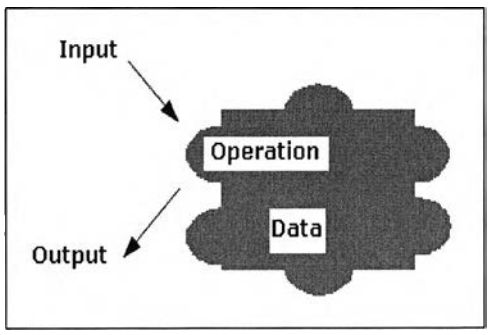


Figure 2.Calling-Called Dependency of Non-Basic and Basic Operations

2.2 B-Method [1]

The B-Method represents a formal specifications modeling that can be used in software development life cycle. The specifications method focuses on the concepts of modularity and information hiding. The BAM notations are used to specify a module to represent each class or object. Each module is defined to encapsulate structural and behavioral properties. The relationship between the BAMs can be defined to represent their collaborations. In fact, developer practically considers a BAM module to a class and utilizes them for developing many complex systems. The structure of BAM is graphically illustrated in figure 3A and the essential clause names of BAM syntax is listed in figure 3B.



MACHINE
USES
SEES
SETS
VARIABLES
INVARIANT
INITIALISATION
OPERATIONS
END

(A) (B)

Figure 3.The Structure of BAM
(A) Information hiding and (B) Importance Clauses in BAM

3. Our Purposed Scheme

We propose a scheme of translating both UML class diagram and related sequence diagrams into BAM specifications and eventually conduct the syntactical and semantic consistency checking using B-Toolkit program [6]. The overview of our proposed scheme is shown in figure 4. We begin to consider the given class diagram and map each class to a BAM module with attributes. A set of BAM skeleton modules is generated with the corresponding attributes. The relations among classes are considered as well to create the relations of BAM accordingly. The related interaction among classes in sequence diagrams will be considered especially on the class operations and the operations of the BAM modules are completely appended. In B-Methods, a BAM implementation module is expected to describe the details of called operations sequences. We provide a set of rules to generate the associated BAM implementation modules. The formal specifications in BAM – the BAM modules with relations and their implementation modules, will be finally gathered and refined. We provide a set of translation rules to cope with activities mentioned above. The B-Toolkit program is used to do the consistency checking. Both developers and users will be guided and provided with our systematic scheme to evaluate their software system model in the early stage.

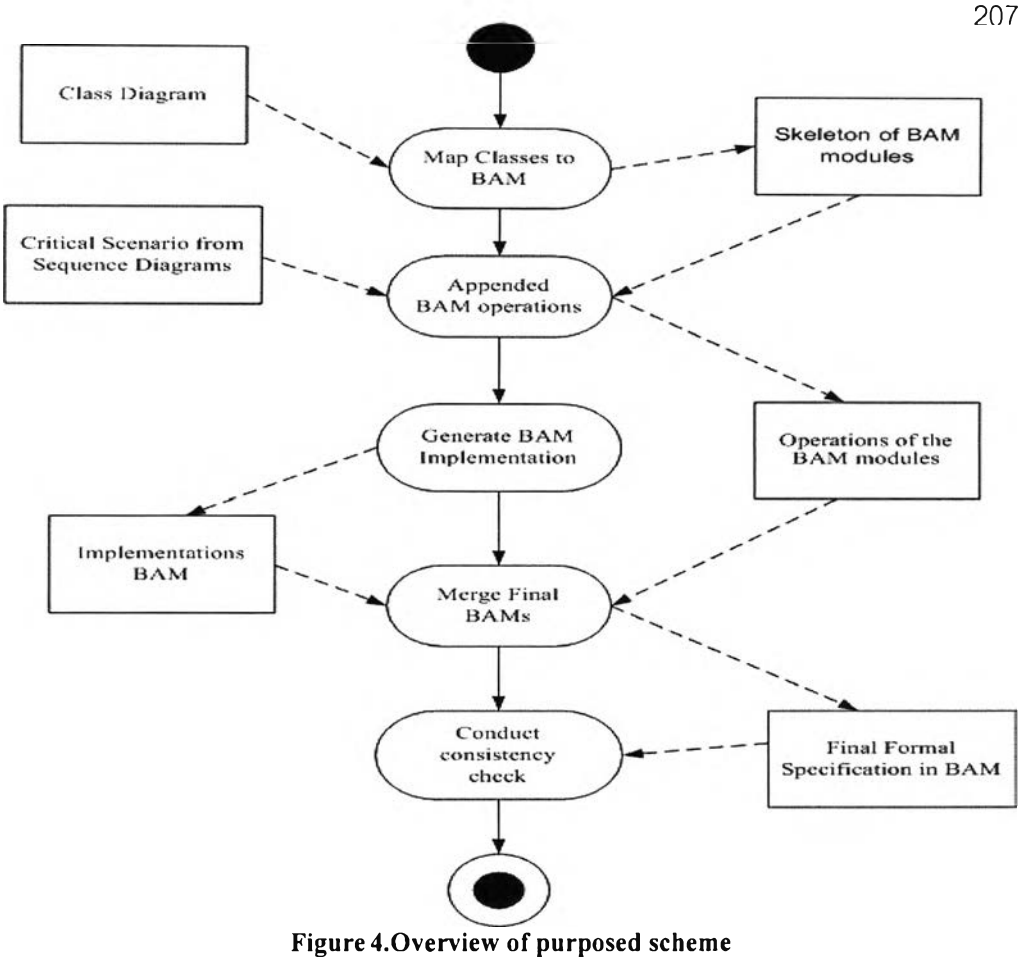


Figure 4.Overview of purposed scheme

4. Case Study

In this section, we introduce a case study of Tuition fee payment system. The class diagram, in figure 5, shows a set of classes named as class *Bank, Student, StudentAccount, UniversityAccount, UndergraduateStudent* with attributes, operations and their relations. The types of relations are drawn with multiplicity notations to describe the structural property of the Tuition fee payment system. To demonstrate one of the payment scenarios, a sequence diagram, in figure 6, shows the interaction between classes to conduct the transfer cash from *StudentAccount* to *UniversityAccount*. A student requests *Bank* to do the operation *transferCash()*. The *Bank* performs the requested operation by asking the *StudentAccount to* do the *withdrawMoney()* operation and asking the *BankAccount* to do the *depositMoney()*. To order to withdraw the money, the *StudentAccount* will perform the called operation *decreaseAmount()*.

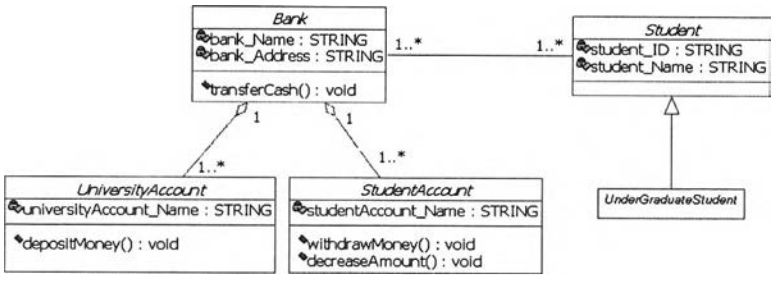


Figure 5.Class diagram of a Tuition fee payment system

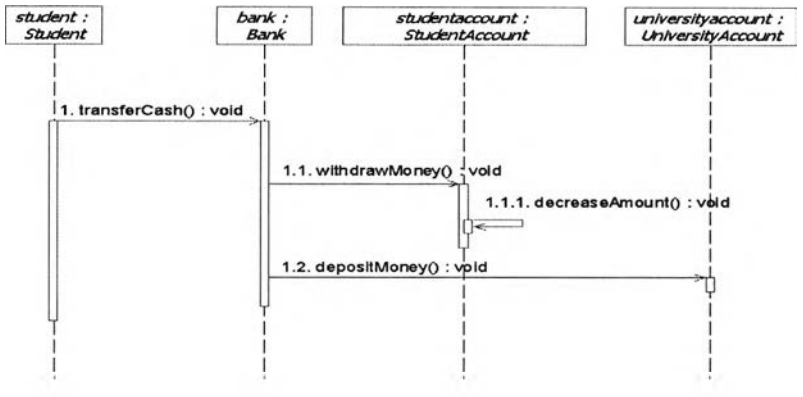


Figure 6.Sequence diagram of a cash payment scenario

5. Translation of UML Class Diagram and Sequence Diagrams into BAM

This section will distinguish between the rule for translation of class diagram and of sequence diagrams by using a sample in case study of tuition fee payment system as follow

1. Generate BAM of BasicClass and BAM of Class will be described in section 5.1.
2. Generate BAM of Relation between classes are association, aggregation and composition. These will be provided in section 5.2.
3. Generate BAM of sub class inherited all of attributes from super class will be explored in section 5.3
4. Generate BAM of Relation (or BAM implicit relation) between sub class inherited from super class and the other class will also be presented in section 5.3.
5. Specify all of operations in BAM of BasicClass and BAM of Class by consider from calling – called of operations of critical scenario of sequence diagrams will be provided in section 5.4.
6. Generate Implementation BAM of Class for describing the sequence order of the invoked operations in each scenario from sequence diagrams will be depicted in section 5.5.

5.1 Translating UML Class Diagram

In this section, we demonstrate the translating of UML class diagram into BAM gradually. We create several BAM modules to each class and named it accordingly in the MACHINE clause, for example, *BasicStudent* and *Student* to represent class *Student*. All of the attributes of each class are defined into VARIABLES clause. The types of attributes will

be considered as sets in SETS clause while INVARIANT clause define the domain set of each variable found in VARIABLES clause. The INITIALISATION clause contains the initial preconditions of each essential attributes in a BAM module. The USES clause will represent the relation between a BAM and the others. The BAM modules of *BasicStudent* and *Student* are shown in figure 7.

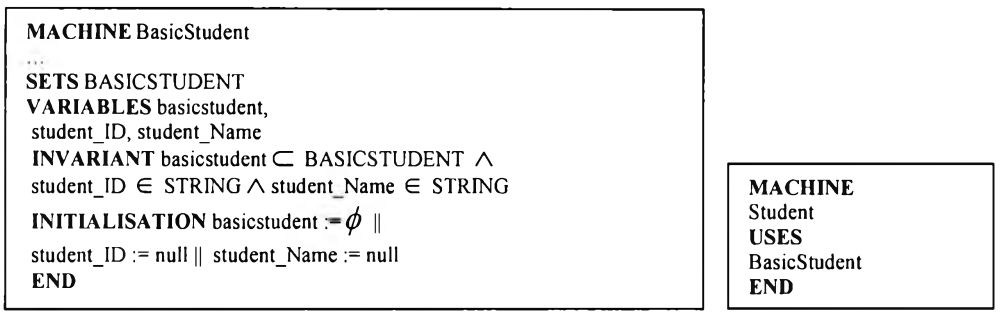


Figure 7.The BAM modules of class *Student*

5.2 Translating the Relations between Classes

An association between two classes in UML is formally defined as a BAM module with "Asso" prefix to its name. The association is considered as a set of order pair of Cartesian product of two relating classes. The multiplicity of the association will be defined as well to represent the number of instance of each class which has relationship. Table 1 shows the mapping between predicates for variety of multiplicity.

Table 1.The Multiplicity and its mapping predicates

Predicate	Multiplicity
RelName ⊂ BASICCLASS1 × BASICCLASS2 dom(RelName) = basicclass1 ∧ ran(RelName) = basicclass2 ∧ ∀ (xx,yy).(((xx ∈ dom(RelName)) ∧ (yy ∈ ran(RelName))) → card((RelName)[{xx}]) ≥ 0 ∧ card((RelName)$^{-1}$[{yy}] ≥ 0)	* Or 0...*
RelName ⊂ BASICCLASS1 × BASICCLASS2 dom(RelName) = basicclass1 ∧ ran(RelName) = basicclass2 ∧ ∀ (xx,yy).(((xx ∈ dom(RelName)) ∧ (yy ∈ ran(RelName))) → card((RelName)[{xx}]) ≥ 1 ∧ card((RelName)$^{-1}$[{yy}]) ≥ 1)	1...*
RelName ⊂ BASICCLASS1 × BASICCLASS2 dom(RelName) = basicclass1 ∧ ran(RelName) = basicclass2 ∧ ∀ (xx,yy).(((xx ∈ dom(RelName)) ∧ (yy ∈ ran(RelName))) → card((RelName)[{xx}]) = 1 ∧ card((RelName)$^{-1}$[{yy}]) = 1)	1
RelName ⊂ BASICCLASS1 × BASICCLASS2 dom(RelName) = basicclass1 ∧ ran(RelName) = basicclass2 ∧ ∀ (xx,yy).(((xx ∈ dom(RelName)) ∧ (yy ∈ ran(RelName))) → card((RelName)[{xx}]) ≥ 0 ∧ card((RelName)[{xx}]) ≤ 1 ∧ card((RelName)$^{-1}$[{yy}]) ≥ 0 ∧ card((RelName)$^{-1}$[{yy}]) ≤ 1)	0...1

The sample of a BAM module for the association between class *Bank* and *Student*, in figure 5, is shown in figure 8. The Aggregation and composition in UML are defined in the similar steps. We use "Aggr" and "Compo" as the prefix to theirs names respectively. Figure 9 shows the composition between *Bank* and *StudentAccount*. In order to implement the aggregation and composition between two classes, the container class has to carry the implicit references to all contents classes. Therefore, we automatically add a reference variable to the container class. As shown in figure 10, the class *Bank* contains a set of *StudentAccount* object so that the reference to *StudentAccount* object called *RefStudentAccountID* is added.

MACHINE Asso_Bank_Student
USES BasicBank, BasicStudent
VARIABLE asso_bank_student
INVARIANT asso_bank_student $\subset$ BASICBANK $\times$ BASICSTUDENT $\wedge$
dom(asso_bank_student) = basicbank $\wedge$ ran(asso_bank_student) = basicstudent $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(asso_bank_student)) $\wedge$ (yy $\in$ ran(asso_bank_student)))
$\rightarrow$ card((asso_bank_student)[{xx}]) $\geq$ 1 $\wedge$ card((asso_bank_student)$^{-1}$[{yy}]) $\geq$ 1)
...
END

Figure 8.A sample of the BAM for association

MACHINE Compo_Bank_StudentAccount
USES BasicBank, BasicStudentAccount
VARIABLE compo_bank_studentaccount
INVARIANT compo_bank_studentaccount $\subset$ BASICBANK $\times$ STUDENTACCOUNT $\wedge$
dom(compo_bank_studentaccount) = basicbank $\wedge$
ran(compo_bank_studentaccount) = basicstudentaccount $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(compo_bank_studentaccount)) $\wedge$
(yy $\in$ ran(compo_bank_studentaccount)))
$\rightarrow$ card((compo_bank_studentaccount)[{xx}]) $\geq$ 1 $\wedge$
card((compo_bank_studentaccount)$^{-1}$[{yy}]) = 1)
...
END

Figure 9.A sample of BAMs for composition

MACHINE BasicBank
...
VARIABLES
...
RefStudentAccountID
INVARIANT
...
RefStudentAccountID $\in$ STRING
INITIALISATION
...
RefStudentAccountID := null
END

Figure 10.A sample of implicit reference for container class

5.3 Translating Sub Class Inherited from a Super Class

When a sub class is inherited from a super class, the sub class will have almost the same properties as super class. All of the non-private attributes and operations will be implicitly copied from super class, as well as various relations. Technically, to generate a BAM for a sub class, we can copy the BAM of super class and paste into the BAM of sub class and do the refinement. In figure 11, a BAM for the sub class *UnderGraduateStudent* is shown, all of the attributes and operations are copied from the super class *Student* and the variable names are refined to avoid the name clashing.

MACHINE BasicUnderGraduateStudent
...
SETS BASICUNDERGRADUATESTUDENT
VARIABLES basicundergraduatestudent,
undergraduatestudent_ID, undergraduatestudent_Name
INVARIANT basicundergraduatestudent $\subset$ BASICUDERGRADUATESTUDENT $\wedge$
undergraduatestudent_ID $\in$ STRING $\wedge$ undergraduatestudent_Name $\in$ STRING
...
END

Figure 11.A sample of basic BAM for the sub class *UnderGraduateStudent*

Moreover, the association between the super class *Bank* and *Student* will be implicitly inherited to the sub class *UnderGraduateStudent* as well. We define the extra implicit association BAM between sub class *Bank* and *UnderGraduateStudent* as shown in figure 12.

MACHINE ImplicitAsso_Bank_UnderGraduateStudent
USES BasicUnderGraduateStudent, BasicBank
VARIABLES implicitasso_bank_undergraduatestudent
INVARIANT
implicitasso_bank_undergraduatestudent
$\subset$ BASICBANK $\times$ BASICUNDERGRADUATESTUDENT $\wedge$
dom(implicitasso_bank_undergraduatestudent) = basicbank $\wedge$
ran(implicitasso_bank_undergraduatestudent) = basicundergraduatestudent $\wedge$
$\forall$ (xx,yy).(((xx $\in$ dom(implicitasso_bank_undergraduatestudent)) $\wedge$
(yy $\in$ ran(implicitasso_bank_undergraduatestudent)))
$\rightarrow$ card((implicitasso_bank_undergraduatestudent)[{xx}]) $\geq$ 1 $\wedge$
card((implicitasso_bank_undergraduatestudent)$^{-1}$[{yy}]) $\geq$ 1)
...
END

Figure 12.A sample of implicit association BAM for the sub class *UnderGraduateStudent*

5.4 Appending the Operations to BAM from Sequence Diagrams

After a set of BAMs is generated from a class diagram, these BAMs must be appended with their operations described in the related sequence diagrams. Typically, the sequence diagram shows the operation names invoked and the correspondent operations between two classes. Using Hung Ledang's technique called Calling-Called Dependency Model [4], we can classify the operations into "Basic" and "Non-Basic" operation groups. The Basic operations will be appended into OPERATIONS clause of basic BAM modules and the Non-Basic operations will be appended into the related original BAM modules.

As shown in figure 13, every operation names in sequence diagrams will be appended into the BAMs completely.

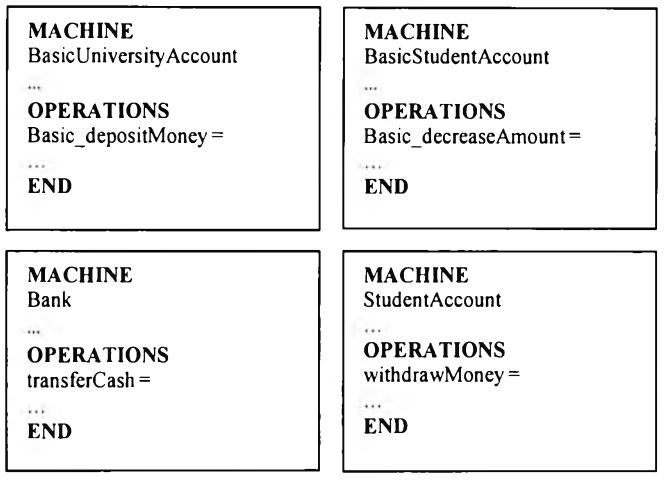


Figure 13.A sample of operation names appended into BAMs

5.5 Generating the BAM Implementation modules

In order to illustrate the sequence of the calling-called operations for each scenario described by a sequence diagram, we generate the extra BAM implementation modules to refine these sequences of invoked operations. As shown in figure 14, the BAM implementation module of *StudentAccount* shows that the *withdrawMoney* operation in BAM will call another operation named *Basic_decreaseAmount*. Another example is the BAM implementation module of *Bank*. The *transferCash* operation in BAM will call two other operations named *withdrawMoney* and *Basic_depositMoney* in sequential order.

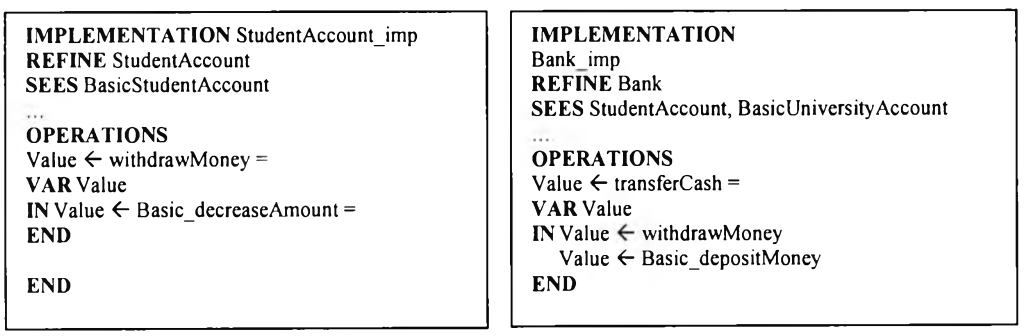


Figure 14.A sample of the BAM implementation modules

6. DISCUSSION

This paper has been further developed from Hung Ledang's research, which allow the user to specify multiplicity of each class appropriately in BAM of relation such as association, aggregation, composition and BAM of implicit relation too. Therefore these can

support "association class" of UML class diagram, and develop calling - called operations of critical scenario from sequence diagrams. These help to increase flexibility of UML design from developer.

7. CONCLUSION

We propose an alternative of the consistency checking for UML class diagram and its related sequence diagrams. Given a set of class diagram and sequence diagrams, we propose a set of translation rules to map a class, its attributes, operations and relations between classes into a formal specification notation called BAM. The translation rules guide to generate a set of BAMs systematically and do some automatic refinement of the specification as well. The varieties of relations are covered such as association, aggregation, composition and inheritance.

The BAMs will be refined using a set of BAM implementation modules by describing the sequence order of the invoked operations in each scenario of the software system.

A case study of Tuition fee payment system is briefly described and the examples of the BAMs are shown. The final BAM specification has been checked by B-Toolkit program and the result has no conflict and applicable.

REFERENCES

[1] J- R Abrial. 1996 *The B – Book Assigning Programs to Meanings*. Cambridge University Press.

[2] Grady Booch, Jame Rumbaugh and Ivar Jacobson. 1998 *The Unified Modeling Language User Guide*. Addison Wesley.

[3] Jim Arlow, Ila Neustadt. 2002 *UML and The Unified Process Pratical Object – Oriented Analysis and Design*. Addison Wesley.

[4] Hung Ledang and Jeanine Souquières. 2001 Integrating UML and B Specification Techniq ues. *Workshop at Informatik,2001* .

[5] Hung Ledang and Jeanine Souquières. 2001 Modeling Class Operations in B: a case study on the pump component. *Technical Report A01-R-011. Laboratory Lorrian de Recherche en Informatique et ses Applications, 2001.*

[6] Kevin Lano and Howard Haughton. 1996 *Specification in B: An Introduction using the B Toolkit*. Imperial College Press.

ภาคผนวก ข
แอ็บสแตร็คแมชชีนบีของระบบการยืมคืนหนังสือในห้องสมุด

- ไลบรารีแอ็บสแตร็คแมชชีนบีทั้งหมด มีดังนี้

- ไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน (BooleanType)

```
MACHINE
BooleanType

SETS
BOOLEAN = {TRUE,FALSE}

END
```

- ไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระ (StringType)

```
MACHINE
StringType

SETS
STRING

VARIABLES
null,
EmptyString

INVARIANT
null : STRING &
EmptyString : STRING

INITIALISATION
null := EmptyString

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสหนังสือที่ขาย (BAM BasicSuplierBook)

```
MACHINE
BasicSupplierBook

SEES
StringType,
BooleanType
```

```
SETS
BASICSUPPLIERBOOK

VARIABLES
basicsupplierbook,
supplierbook_Name,
supplierbook_ISBN

INVARIANT
basicsupplierbook <: BASIC_SUPPLIERBOOK &
supplierbook_Name : STRING &
supplierbook_ISBN : STRING

INITIALISATION
basicsupplierbook := {} ||
supplierbook_Name := null ||
supplierbook_ISBN := null

OPERATIONS
boolean <-- basic_checkListOfBook(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสผู้ขาย (BAM BasicSuplier)

```
MACHINE
BasicSupplier

SEES
StringType

SETS
BASICSUPPLIER

VARIABLES
basicsupplier,
supplier_Name,
supplier_Address

INVARIANT
basicsupplier <: BASICSUPPLIER &
supplier_Name : STRING &
supplier_Address : STRING

INITIALISATION
basicsupplier := {} ||
supplier_Name := null ||
```

```
supplier_Address := null

OPERATIONS
basicsupplier1 <-- Basic_Contact(a1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
basicsupplier1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสห้องสมุด (BAM BasicLibrary)

```
MACHINE
BasicLibrary

SEES
StringType

SETS
BASICLIBRARY

VARIABLES
basiclibrary,
library_Name,
library_Address,
library_NumberOfBook

INVARIANT
basiclibrary <: BASICLIBRARY &
library_Name : STRING &
library_Address : STRING &
library_NumberOfBook : NAT

INITIALISATION
basiclibrary := {} ||
library_Name := null ||
library_Address := null ||
library_NumberOfBook := 0

OPERATIONS
voidbasiclibrary1 <-- Basic_addBook(a1,x1) =
PRE
a1 : STRING &
x1 : NAT
THEN
voidbasiclibrary1 := null
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสบรรณารักษ์ (BAM BasicLibrarian)

```
MACHINE
BasicLibrarian

SEES
StringType

SETS
BASICLIBRARIAN

VARIABLES
basiclibrarian,
librarian_ID,
librarian_Name,
librarian_Surname

INVARIANT
basiclibrarian <: BASICLIBRARIAN &
librarian_ID : STRING &
librarian_Name : STRING &
librarian_Surname : STRING

INITIALISATION
basiclibrarian := {} ||
librarian_ID := null ||
librarian_Name := null ||
librarian_Surname := null

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายการการยืมคืน (BAM BasicBorrowReturnTransaction)

```
MACHINE
BasicBorrowReturnTransaction

SEES
StringType,
BooleanType

SETS
BASICBORROWRETURNTRANSACTION

VARIABLES
basicborrowreturntransaction,
borrowreturntransaction_ID,
borrowreturntransaction_Name,
borrowreturntransaction_DateBorrow,
borrowreturntransaction_DateReturn,
borrowreturntransaction_NumberOfBookToBorrow,
borrowreturntransaction_NumberOfBookToReturn

INVARIANT
basicborrowreturntransaction <: BASICBORROWRETURNTRANSACTION &
borrowreturntransaction_ID : STRING &
```

```
borrowreturntransaction_Name : STRING &
borrowreturntransaction_DateBorrow : STRING &
borrowreturntransaction_DateReturn : STRING &
borrowreturntransaction_NumberOfBookToBorrow : NAT &
borrowreturntransaction_NumberOfBookToReturn : NAT

INITIALISATION
basicborrowreturntransaction := {} ||
borrowreturntransaction_ID := null ||
borrowreturntransaction_Name := null ||
borrowreturntransaction_DateBorrow := null ||
borrowreturntransaction_DateReturn := null ||
borrowreturntransaction_NumberOfBookToBorrow := 0 ||
borrowreturntransaction_NumberOfBookToReturn := 0

OPERATIONS
boolean <-- Basic_checkLimitMaximumBook(x1,x2) =
PRE
x1 : NAT &
x2 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_confirmToBorrow(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_compareDateBorrowAndReturn(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_confirmToReturn(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;
```

```
basicborrowreturntransaction1 <-- Basic_getBorrowReturnInfo(a1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
basicborrowreturntransaction1 := null
 /* User can modify output value or post condition of operation here*/
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายการการจอง (BAM BasicReservationTransaction)

```
MACHINE
BasicReservationTransaction

SEES
StringType,
BooleanType

SETS
BASICRESERVATIONTRANSACTION

VARIABLES
basicreservationtransaction,
reservationtransaction_Name,
reservationtransaction_DateReservation,
reservationtransaction_NumberBook,
RefReservationBookID

INVARIANT
basicreservationtransaction <: BASICRESERVATIONTRANSACTION &
reservationtransaction_Name : STRING &
reservationtransaction_DateReservation : STRING &
reservationtransaction_NumberBook : NAT &
RefReservationBookID : STRING

INITIALISATION
basicreservationtransaction := {} ||
reservationtransaction_Name := null ||
reservationtransaction_DateReservation := null ||
reservationtransaction_NumberBook := 0 ||
RefReservationBookID := null

OPERATIONS
boolean <-- Basic_confirmToReservation(a1,x1) =
PRE
a1 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END
```

```
END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสใบสั่งซื้อ (BAM BasicPurchaseOrder)

```
MACHINE
BasicPurchaseOrder

SEES
StringType,
BooleanType

SETS
BASICPURCHASEORDER

VARIABLES
basicpurchaseorder,
purchaseorder_Date,
RefBookToOrderID

INVARIANT
basicpurchaseorder <: BASICPURCHASEORDER &
purchaseorder_Date : STRING &
RefBookToOrderID : STRING

INITIALISATION
basicpurchaseorder := {} ||
purchaseorder_Date := null ||
RefBookToOrderID := null

OPERATIONS
boolean <-- Basic_confirmToBuy(a1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคซูเปอร์คลาสสมาชิกห้องสมุด (BAM BasicMember)

```
MACHINE
BasicMember

SEES
StringType,
BooleanType

SETS
BASICMEMBER

VARIABLES
basicmember,
```

```
member_ID,
member_Name,
member_Surname,
member_Address,
member_DateBorrow,
member_DateReturn,
member_DateDeadLineReturn

INVARIANT
basicmember <: BASICMEMBER &
member_ID : STRING &
member_Name : STRING &
member_Surname : STRING &
member_Address : STRING &
member_DateBorrow : STRING &
member_DateReturn : STRING &
member_DateDeadLineReturn : STRING

INITIALISATION
basicmember := {} ||
member_ID  := null ||
member_Name := null ||
member_Surname := null ||
member_Address := null ||
member_DateBorrow := null ||
member_DateReturn := null ||
member_DateDeadLineReturn := null

OPERATIONS
boolean <-- Basic_checkMember(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

voidbasicmember1 <-- Basic_aviableToBorrow =
BEGIN
voidbasicmember1 := null
END;

boolean <-- Basic_chargeFee(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคซับคลาสนิสิต (BAM BasicStudent)

```
MACHINE
BasicStudent

SEES
StringType,
BooleanType

SETS
BASICSTUDENT

VARIABLES
basicstudent,
student_Year,
student_ID,
student_Name,
student_Surname,
student_Address,
student_DateBorrow,
student_DateReturn,
student_DateDeadLineReturn

INVARIANT
basicstudent <: BASICSTUDENT &
student_Year : NAT &
student_ID : STRING &
student_Name : STRING &
student_Surname : STRING &
student_Address : STRING &
student_DateBorrow : STRING &
student_DateReturn : STRING &
student_DateDeadLineReturn : STRING

INITIALISATION
basicstudent := {} ||
student_Year := 0 ||
student_ID  := null ||
student_Name := null ||
student_Surname := null ||
student_Address := null ||
student_DateBorrow := null ||
student_DateReturn := null ||
student_DateDeadLineReturn := null

OPERATIONS
boolean <-- Basic_checkMember(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;
```

```
voidbasicstudent1 <-- Basic_aviableToBorrow =
BEGIN
voidbasicstudent1 := null
END;

boolean <-- Basic_chargeFee(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสผู้สอน (BAM BasicInstructor)

```
MACHINE
BasicInstructor

SEES
StringType,
BooleanType

SETS
BASICINSTRUCTOR

VARIABLES
basicinstructor,
instructor_Position,
instructor_ID,
instructor_Name,
instructor_Surname,
instructor_Address,
instructor_DateBorrow,
instructor_DateReturn,
instructor_DateDeadLineReturn

INVARIANT
basicinstructor <: BASICINSTRUCTOR &
instructor_Position : STRING &
instructor_ID : STRING &
instructor_Name : STRING &
instructor_Surname : STRING &
instructor_Address : STRING &
instructor_DateBorrow : STRING &
instructor_DateReturn : STRING &
instructor_DateDeadLineReturn : STRING

INITIALISATION
basicinstructor := {} ||
instructor_Position := null ||
instructor_ID  := null ||
```

```
instructor_Name := null ||
instructor_Surname := null ||
instructor_Address := null ||
instructor_DateBorrow := null ||
instructor_DateReturn := null ||
instructor_DateDeadLineReturn := null

OPERATIONS
boolean <-- Basic_checkMember(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

voidbasicinstructor1 <-- Basic_aviableToBorrow =
BEGIN
voidbasicinstructor1 := null
END;

boolean <-- Basic_chargeFee(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสหนังสือ (BAM BasicBook)

```
MACHINE
BasicBook

SEES
StringType,
BooleanType

SETS
BASICBOOK

VARIABLES
basicbook,
book_ID,
book_Name,
book_Description,
book_Edition,
book_DateBorrow,
book_DateReturn,
```

```
book_DateReservation,
book_BorrowFlag,
book_ReservationFlag

INVARIANT
basicbook <: BASICBOOK &
book_ID : STRING &
book_Name : STRING &
book_Description : STRING &
book_Edition : STRING &
book_DateBorrow : STRING &
book_DateReturn : STRING &
book_DateReservation : STRING &
book_BorrowFlag : BOOLEAN &
book_ReservationFlag : BOOLEAN

INITIALISATION
basicbook := {} ||
book_ID := null ||
book_Name := null ||
book_Description := null ||
book_Edition := null ||
book_DateBorrow := null ||
book_DateReturn := null ||
book_DateReservation := null ||
book_BorrowFlag := FALSE ||
book_ReservationFlag := FALSE

OPERATIONS
boolean <-- Basic_setBorrowStatus(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_setReturnStatus(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_checkBookInfo(a1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
```

```
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_setReservationStatus(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_resetBorrowStatus(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายชื่อหนังสือที่ถูกจอง (BAM BasicReservationBookList)

```
MACHINE
BasicReservationBookList

SEES
StringType

SETS
BASICRESERVATIONBOOKLIST

VARIABLES
basicreservationbooklist,
reservationbooklist_Name

INVARIANT
basicreservationbooklist <: BASICRESERVATIONBOOKLIST &
reservationbooklist_Name : STRING

INITIALISATION
basicreservationbooklist := {} ||
reservationbooklist_Name := null

OPERATIONS
basicreservationbooklist1 <-- Basic_getReservationBook =
BEGIN
basicreservationbooklist1 := null
/* User can modify output value or post condition of operation here */
END
```

```
END
```

- แอ็บสแตร็คแมชชีนบีเบชิคคลาสหนังสือที่จะสั่งซื้อ (BAM BasicBookToOrder)

```
MACHINE
BasicBookToOrder

SEES
StringType

SETS
BASICBOOKTOORDER

VARIABLES
basicbooktoorder,
booktoorder_ISBN,
booktoorder_Name

INVARIANT
basicbooktoorder <: BASICBOOKTOORDER &
booktoorder_ISBN : STRING &
booktoorder_Name : STRING

INITIALISATION
basicbooktoorder := {} ||
booktoorder_ISBN := null ||
booktoorder_Name := null

OPERATIONS
basicbooktoorder1 <-- Basic_displayListOfBook(a1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
basicbooktoorder1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีคลาสหนังสือที่ขาย (BAM SuplierBook)

```
MACHINE
SupplierBook

USES
BasicSupplierBook,
Asso_SupplierBook_Supplier

END
```

- แอ็บสแตร็คแมชชีนบีคลาสผู้ขาย (BAM Suplier)

MACHINE
Supplier

USES
BasicSupplier,
AssoDealWith_Supplier_Librarian,
Asso_SupplierBook_Supplier

END

- แอ็บสแตร็คแมชชีนบีคลาสห้องสมุด (BAM Library)

MACHINE
Library

USES
BasicLibrary,
Asso_Librarian_Library

END

- แอ็บสแตร็คแมชชีนบีคลาสบรรณารักษ์ (BAM Librarian)

MACHINE
Librarian

USES
BasicLibrarian,
AssoDealWith_Supplier_Librarian,
Asso_BorrowReturnTransaction_Librarian,
Asso_ReservationTransaction_Librarian,
Asso_Librarian_PurchaseOrder,
Asso_Librarian_Library

END

- แอ็บสแตร็คแมชชีนบีคลาสรายการการยืมคืน (BAM BorrowReturnTransaction)

MACHINE
BorrowReturnTransaction

SEES
StringType

USES
BasicBorrowReturnTransaction,
Asso_BorrowReturnTransaction_Book,
Asso_BorrowReturnTransaction_Librarian,
Asso_Member_BorrowReturnTransaction
ImplicitAsso_Student_BorrowReturnTransaction,
ImplicitAsso_Instructor_BorrowReturnTransaction

```
OPERATIONS
borrowreturntransaction1 <-- borrowBook(a1,a2,a3,x1,x2) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
x1 : NAT &
x2 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
borrowreturntransaction1 := null
/* User can modify output value or post condition of operation here */
END;

borrowreturntransaction2 <-- returnBook(a1,a2,a3,x1,x2) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
x1 : NAT &
x2 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
borrowreturntransaction2 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสรายการการจอง (BAM ReservationTransaction)

```
MACHINE
ReservationTransaction

SEES
StringType

USES
BasicReservationTransaction,
Asso_ReservationTransaction_Book,
Asso_ReservationTransaction_Librarian,
Compo_ReservationTransaction_ReservationBook

OPERATIONS
reservationtransaction1 <-- reservationBook(a1,a2,a3,x1) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
reservationtransaction1 := null
/* User can modify output value or post condition of operation here */
```

```
END;

reservationtransaction2 <-- insertList(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
reservationtransaction2 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสใบสั่งซื้อ (BAM PurchaseOrder)

```
MACHINE
PurchaseOrder

SEES
StringType,
BooleanType

USES
BasicPurchaseOrder,
Asso_Librarian_PurchaseOrder,
Aggr_PurchaseOrder_BookToOrder

OPERATIONS
boolean <-- createTransaction(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีซูเปอร์คลาสสมาชิกห้องสมุด (BAM Member)

```
MACHINE
Member

SEES
StringType

USES
BasicMember,
Asso_Member_BorrowReturnTransaction
```

```
OPERATIONS
member1 <-- getMember(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
member1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีซับคลาสนิสิต (BAM Student)

```
MACHINE
Student

SEES
StringType

USES
BasicStudent,
ImplicitAsso_Student_BorrowReturnTransaction

OPERATIONS
student1 <-- getMember(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
student1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสผู้สอน (BAM Instructor)

```
MACHINE
Instructor

SEES
StringType

USES
BasicInstructor,
ImplicitAsso_Instructor_BorrowReturnTransaction

OPERATIONS
instructor1 <-- getMember(a1,a2) =
PRE
a1 : STRING &
```

```
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
instructor1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสหนังสือ (BAM Book)

```
MACHINE
Book

SEES
StringType

USES
BasicBook,
Asso_BorrowReturnTransaction_Book,
Asso_ReservationTransaction_Book

OPERATIONS
book1 <-- getBook(a1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
book1 := null
/* User can modify output value or post condition of operation here */
END;

book2 <-- getBookInfo(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
book2 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสหนังสือที่ถูกจอง (BAM ReservationBookList)

```
MACHINE
ReservationBookList

USES
BasicReservationBookList,
Compo_ReservationTransaction_ReservationBookList

END
```

- แอ็บสแตร็คแมชชีนบีคลาสหนังสือที่จะสั่งซื้อ (BAM BookToOrder)

```
MACHINE
BookToOrder

USES
BasicBookToOrder,
Aggr_PurchaseOrder_BookToOrder

END
```

- แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสหนังสือ (BAM IntermediateBook)

```
MACHINE
IntermediateBook

SEES
StringType

USES
BasicBook,
Asso_BorrowReturnTransaction_Book,
Asso_ReservationTransaction_Book

OPERATIONS
boolean <-- Intermediate_checkBook(a1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์ทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสรายการการยืมคืนกับคลาสหนังสือ (Asso_BorrowReturnTransaction_Book)

```
MACHINE
Asso_BorrowReturnTransaction_Book

USES
BasicBorrowReturnTransaction,
BasicBook
```

```
VARIABLES
asso_borrowreturntransaction_book

INVARIANT
asso_borrowreturntransaction_book <: BASICBORROWRETURNTRANSACTION * BASICBOOK &
dom(asso_borrowreturntransaction_book) = basicborrowreturntransaction &
ran(asso_borrowreturntransaction_book) = basicbook &
!(xx,yy).(((xx : dom(asso_borrowreturntransaction_book)) &
            (yy : ran(asso_borrowreturntransaction_book)))
         => card((asso_borrowreturntransaction_book)[{xx}]) >= 1 &
             card((asso_borrowreturntransaction_book)~[{yy}])= 1)

INITIALISATION
asso_borrowreturntransaction_book := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสรายการการยืมคืนกับคลาสบรรณารักษ์

(Asso_BorrowReturnTransaction_Librarian)

```
MACHINE
Asso_BorrowReturnTransaction_Librarian

USES
BasicBorrowReturnTransaction,
BasicLibrarian

VARIABLES
asso_borrowreturntransaction_librarian

INVARIANT
asso_borrowreturntransaction_librarian <: BASICBORROWRETURNTRANSACTION *
BASICLIBRARIAN &
dom(asso_borrowreturntransaction_librarian) = basicborrowreturntransaction &
ran(asso_borrowreturntransaction_librarian) = basiclibrarian &
!(xx,yy).(((xx : dom(asso_borrowreturntransaction_librarian)) &
            (yy : ran(asso_borrowreturntransaction_librarian)))
         => card((asso_borrowreturntransaction_librarian)[{xx}]) = 1 &
             card((asso_borrowreturntransaction_librarian)~[{yy}]) >= 0)

INITIALISATION
asso_borrowreturntransaction_librarian := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสบรรณารักษ์กับคลาสห้องสมุด

(Asso_Librarian_Library)

```
MACHINE
Asso_Librarian_Library

USES
BasicLibrarian,
```

```
BasicLibrary

VARIABLES
asso_librarian_library

INVARIANT
asso_librarian_library <: BASICLIBRARIAN * BASICLIBRARY &
dom(asso_librarian_library) = basiclibrarian &
ran(asso_librarian_library) = basiclibrary &
!(xx,yy).(((xx : dom(asso_librarian_library)) &
            (yy : ran(asso_librarian_library)))
         => card((asso_librarian_library)[{xx}]) = 1 &
             card((asso_librarian_library)~[{yy}]) >= 1)

INITIALISATION
asso_librarian_library := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสบรรณารักษ์กับคลาสใบสั่งซื้อสินค้า

(Asso_Librarian_PurchaseOrder)

```
MACHINE
Asso_Librarian_PurchaseOrder

USES
BasicLibrarian,
BasicPurchaseOrder

VARIABLES
asso_librarian_purchaseorder

INVARIANT
asso_librarian_purchaseorder <: BASICLIBRARIAN * BASICPURCHASEORDER &
dom(asso_librarian_purchaseorder) = basiclibrarian &
ran(asso_librarian_purchaseorder) = basicpurchaseorder &
!(xx,yy).(((xx : dom(asso_librarian_purchaseorder)) &
            (yy : ran(asso_librarian_purchaseorder)))
         => card((asso_librarian_purchaseorder)[{xx}]) >= 0 &
             card((asso_librarian_purchaseorder)~[{yy}]) = 1)

INITIALISATION
asso_librarian_purchaseorder := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสสมาชิกห้องสมุดกับคลาสรายการการยืม คืน (Asso_Member_BorrowReturnTransaction)

```
MACHINE
Asso_Member_BorrowReturnTransaction

USES
BasicMember,
BasicBorrowReturnTransaction

VARIABLES
asso_member_borrowreturntransaction

INVARIANT
asso_member_borrowreturntransaction <: BASICMEMBER * BASICBORROWRETURNTRANSACTION
&
dom(asso_member_borrowreturntransaction) = basicmember &
ran(asso_member_borrowreturntransaction) = basicborrowreturntransaction &
!(xx,yy).(((xx : dom(asso_member_borrowreturntransaction)) &
            (yy : ran(asso_member_borrowreturntransaction)))
         => card((asso_member_borrowreturntransaction)[{xx}]) >= 0 &
             card((asso_member_borrowreturntransaction)~[{yy}])= 1)

INITIALISATION
asso_member_borrowreturntransaction := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสรายการการจองกับคลาสหนังสือ (Asso_ReservationTransaction_Book)

```
MACHINE
Asso_ReservationTransaction_Book

USES
BasicReservationTransaction,
BasicBook

VARIABLES
asso_reservationtransaction_book

INVARIANT
asso_reservationtransaction_book <: BASICRESERVATIONTRANSACTION * BASICBOOK &
dom(asso_reservationtransaction_book) = basicreservationtransaction &
ran(asso_reservationtransaction_book) = basicbook &
!(xx,yy).(((xx : dom(asso_reservationtransaction_book)) &
            (yy : ran(asso_reservationtransaction_book)))
         => card((asso_reservationtransaction_book)[{xx}]) >= 1 &
             card((asso_reservationtransaction_book)~[{yy}]) = 1)

INITIALISATION
asso_reservationtransaction_book := {}
```

```
END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสรายการการจองกับคลาสบรรณารักษ์

(Asso_ReservationTransaction_Librarian)

```
MACHINE
Asso_ReservationTransaction_Librarian

USES
BasicReservationTransaction,
BasicLibrarian

VARIABLES
asso_reservationtransaction_librarian

INVARIANT
asso_reservationtransaction_librarian <: BASICRESERVATIONTRANSACTION * BASICLIBRARIAN &
dom(asso_reservationtransaction_librarian) = basicreservationtransaction &
ran(asso_reservationtransaction_librarian) = basiclibrarian &
!(xx,yy).(((xx : dom(asso_reservationtransaction_librarian)) &
          (yy : ran(asso_reservationtransaction_librarian)))
        => card((asso_reservationtransaction_librarian)[{xx}]) = 1 &
           card((asso_reservationtransaction_librarian)~[{yy}]) >= 0)

INITIALISATION
asso_reservationtransaction_librarian := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสหนังสือที่ขายกับคลาสผู้ขาย

(Asso_SupplierBook_Supplier)

```
MACHINE
Asso_SupplierBook_Supplier

USES
BasicSupplierBook,
BasicSupplier

VARIABLES
asso_supplierbook_supplier
INVARIANT
asso_supplierbook_supplier <: BASICSUPPLIERBOOK * BASICSUPPLIER &
dom(asso_supplierbook_supplier) = basicsupplierbook &
ran(asso_supplierbook_supplier) = basicsupplier &
!(xx,yy).(((xx : dom(asso_supplierbook_supplier)) &
          (yy : ran(asso_supplierbook_supplier)))
        => card((asso_supplierbook_supplier)[{xx}]) = 1 &
           card((asso_supplierbook_supplier)~[{yy}]) >= 0)

INITIALISATION
asso_supplierbook_supplier := {}
```

```
END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันการติดต่อสั่งซื้อระหว่างคลาสผู้ขายกับคลาสบรรณารักษ์ (AssoDealWith_Supplier_Librarian)

```
MACHINE
AssoDealWith_Supplier_Librarian

USES
BasicSupplier,
BasicLibrarian

VARIABLES
assodealwith_supplier_librarian

INVARIANT
assodealwith_supplier_librarian <: BASICSUPPLIER * BASICLIBRARIAN &
dom(assodealwith_supplier_librarian) = basicsupplier &
ran(assodealwith_supplier_librarian) = basiclibrarian &
!(xx,yy).(((xx : dom(assodealwith_supplier_librarian)) &
          (yy : ran(assodealwith_supplier_librarian)))
        => card((assodealwith_supplier_librarian)[{xx}]) = 1 &
           card((assodealwith_supplier_librarian)~[{yy}]) >= 0)

INITIALISATION
assodealwith_supplier_librarian := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสใบสั่งซื้อกับคลาสหนังสือที่จะสั่งซื้อ (Aggr_PurchaseOrder_BookToOrder)

```
MACHINE
Aggr_PurchaseOrder_BookToOrder

USES
BasicPurchaseOrder,
BasicBookToOrder

VARIABLES
aggr_purchaseorder_booktoorder
INVARIANT
aggr_purchaseorder_booktoorder <: BASICPURCHASEORDER * BASICBOOKTOORDER &
dom(aggr_purchaseorder_booktoorder) = basicpurchaseorder &
ran(aggr_purchaseorder_booktoorder) = basicbooktoorder &
!(xx,yy).(((xx : dom(aggr_purchaseorder_booktoorder)) &
          (yy : ran(aggr_purchaseorder_booktoorder)))
        => card((aggr_purchaseorder_booktoorder)[{xx}]) >= 1 &
           card((aggr_purchaseorder_booktoorder)~[{yy}]) >= 0 &
           card((aggr_purchaseorder_booktoorder)~[{yy}]) <= 1)

INITIALISATION
aggr_purchaseorder_booktoorder := {}
```

END

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสรายการการจองกับคลาสรายชื่อหนังสือที่ถูกจอง (Compo_ReservationTransaction_ReservationBookList)

```
MACHINE
Compo_ReservationTransaction_ReservationBookList

USES
BasicReservationTransactionList,
BasicReservationBook

VARIABLES
compo_reservationtransaction_reservationbooklist

INVARIANT
compo_reservationtransaction_reservationbooklist <: BASICRESERVATIONTRANSACTION
* BASICRESERVATIONBOOKLIST &
dom(compo_reservationtransaction_reservationbooklist) = basicreservationtransaction &
ran(compo_reservationtransaction_reservationbooklist) = basicreservationbooklist &
!(xx,yy).(((xx : dom(compo_reservationtransaction_reservationbooklist)) &
            (yy : ran(compo_reservationtransaction_reservationbooklist)))
        => card((compo_reservationtransaction_reservationbooklist)[{xx}]) >= 1 &
            card((compo_reservationtransaction_reservationbooklist)~[{yy}]) = 1)

INITIALISATION
compo_reservationtransaction_reservationbooklist := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาสผู้สอนกับคลาสรายการการยืมคืน (ImplicitAsso_Instructor_BorrowReturnTransaction)

```
MACHINE
ImplicitAsso_Instructor_BorrowReturnTransaction

USES
BasicInstructor,
BasicBorrowReturnTransaction
VARIABLES
implicitasso_instructor_borrowreturntransaction

INVARIANT
implicitasso_instructor_borrowreturntransaction <: BASICINSTRUCTOR *
BASICBORROWRETURNTRANSACTION &
dom(implicitasso_instructor_borrowreturntransaction) = basicinstructor &
ran(implicitasso_instructor_borrowreturntransaction) = basicborrowreturntransaction &
!(xx,yy).(((xx : dom(implicitasso_instructor_borrowreturntransaction)) &
            (yy : ran(implicitasso_instructor_borrowreturntransaction)))
        => card((implicitasso_instructor_borrowreturntransaction)[{xx}]) >= 0 &
            card((implicitasso_instructor_borrowreturntransaction)~[{yy}]) = 1)
INITIALISATION
```

```
implicitasso_instructor_borrowreturntransaction := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาสนิสิตกับคลาสรายการการยืมคืน (ImplicitAsso_Student_BorrowReturnTransaction)

```
MACHINE
ImplicitAsso_Student_BorrowReturnTransaction

USES
BasicStudent,
BasicBorrowReturnTransaction

VARIABLES
implicitasso_student_borrowreturntransaction

INVARIANT
implicitasso_student_borrowreturntransaction <: BASICSTUDENT *
BASICBORROWRETURNTRANSACTION &
dom(implicitasso_student_borrowreturntransaction) = basicstudent &
ran(implicitasso_student_borrowreturntransaction) = basicborrowreturntransaction &
!(xx,yy).(((xx : dom(implicitasso_student_borrowreturntransaction)) &
            (yy : ran(implicitasso_student_borrowreturntransaction)))
        => card((implicitasso_student_borrowreturntransaction)[{xx}]) >= 0 &
           card((implicitasso_student_borrowreturntransaction)~[{yy}])= 1)

INITIALISATION
implicitasso_student_borrowreturntransaction := {}

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีทั้งหมด มีดังนี้

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสรายการการยืมคืน (BorrowReturnTransaction_imp)

```
IMPLEMENTATION
BorrowReturnTransaction_imp

REFINES
BorrowReturnTransaction

SEES
StringType,
BooleanType,
BasicBorrowReturnTransaction,
Member,
Book,
BasicBook,
BasicMember
```

```
OPERATIONS
borrowreturntransaction1 <-- borrowBook(a1,a2,a3,x1,x2) =
VAR
member1,
book1,
boolean
IN
member1 <-- getMember(a1,a2);
boolean <-- Basic_checkLimitMaximumBook(x1,x2);
        IF
        boolean = TRUE
        THEN
        book1 <-- getBook(a1)
        ELSE
        skip;
boolean <-- Basic_setBorrowStatus(a1,a2);
boolean <-- Basic_confirmToBorrow(a1,a2)
END
END;

borrowreturntransaction2 <-- returnBook(a1,a2,a3,x1,x2) =
VAR
boolean
IN
boolean <-- Basic_setReturnStatus(a1,a2);
boolean <-- Basic_compareDateBorrowAndReturn(a1,a2);
boolean <-- Basic_chargeFee(a1,a2);
boolean <-- Basic_confirmToReturn(a1,a2)
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสรายการการจอง (ReservationTransaction_ imp)

```
IMPLEMENTATION
ReservationTransaction_imp

REFINES
ReservationTransaction

SEES
StringType,
BooleanType,
Book,
BasicBook,
BasicReservationBook,
BasicReservationTransaction

OPERATIONS
reservationtransaction1 <-- reservationBook(a1,a2,a3,x1) =
VAR
basicreservationbook1,
book2,
boolean
```

```
IN
basicreservationbook1 <-- Basic_getReservationBook;
book2 <-- getBookInfo(a1,a2);
boolean <-- Basic_setReservationStatus(a1,a2);
boolean <-- Basic_confirmToReservation(a1,x1)
END;

reservationtransaction2 <-- insertList(a1,a2) =
VAR
boolean
IN
boolean <-- Basic_resetBorrowStatus(a1,a2)
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร์คแมชชีนบีคลาสหนังสือ (Book_imp)

```
IMPLEMENTATION
Book_imp

REFINES
Book

SEES
StringType,
BooleanType,
IntermediateBook

OPERATIONS
book1 <-- getBook(a1) =
VAR
boolean
IN
boolean <-- Intermedaite_checkBook(a1)
END;

book2 <-- getBookInfo(a1,a2) =
VAR
boolean
IN
boolean <-- Basic_checkBookInfo(a1)
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร์คแมชชีนบีอินเทอร์มิเดียทคลาสหนังสือ (IntermediateBook_imp)

```
IMPLEMENTATION
IntermediateBook_imp

REFINES
IntermediateBook
```

```
SEES
StringType,
BooleanType,
BasicMember

OPERATIONS
boolean <-- Intermediate_checkBook(a1) =
VAR
voidbasicmember1
IN
        IF
        boolean = TRUE
        THEN
         voidbasicmember1 <-- Basic_aviableToBorrow
        ELSE
        skip
        END
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสใบสั่งซื้อ (PurchaseOrder_imp)

```
IMPLEMENTATION
PurchaseOrder_imp

REFINES
PurchaseOrder

SEES
StringType,
BooleanType,
BasicBookToOrder
OPERATIONS
boolean <-- createTransaction(a1,a2) =
VAR
basiccooktoorder1
IN
basiccooktoorder1 <-- Basic_displayListOfBook(a1)
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีซูเปอร์คลาสสมาชิกของห้องสมุด (Member_imp)

```
IMPLEMENTATION
Member_imp

REFINES
Member

SEES
StringType,
BooleanType,
```

```
BasicMember

OPERATIONS
member1 <-- getMember(a1,a2) =
VAR
boolean
IN
boolean <-- Basic_checkMember(a1,a2)
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีซับคลาสผู้สอน (Instructor_imp)

```
IMPLEMENTATION
Instructor_imp

REFINES
Instructor

SEES
StringType,
BooleanType,
BasicInstructor

OPERATIONS
instructor1 <-- getMember(a1,a2) =
VAR
boolean
IN
boolean <-- Basic_checkMember(a1,a2)
END

END
```

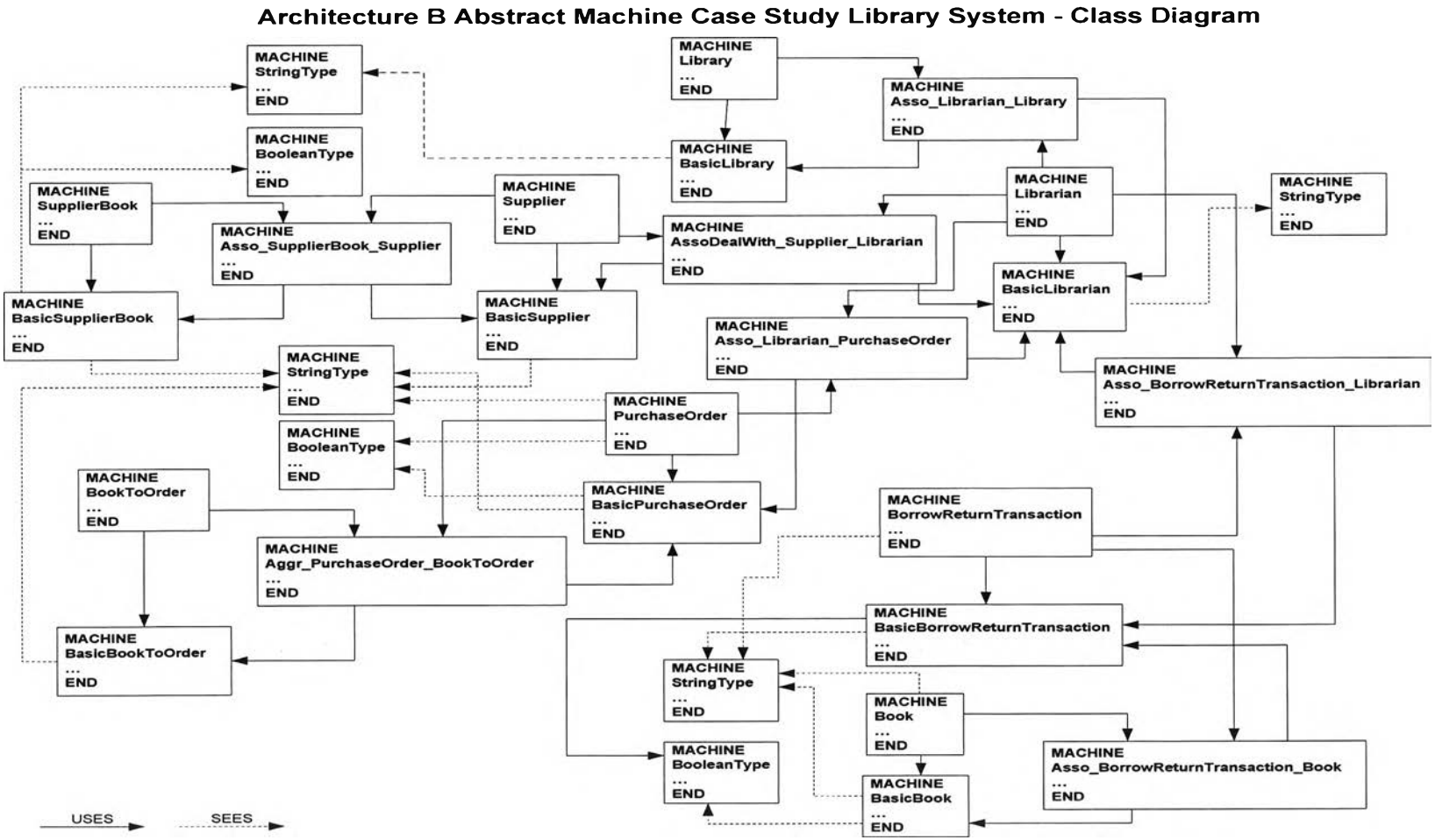
- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีซับคลาสนิสิต (Student_imp)

```
IMPLEMENTATION
Student_imp

REFINES
Student

SEES
StringType,
BooleanType,
BasicStudent

OPERATIONS
student1 <-- getMember(a1,a2) =
VAR
boolean
IN
boolean <-- Basic_checkMember(a1,a2)
```

ต้นฉบับ หน้าขาดหาย

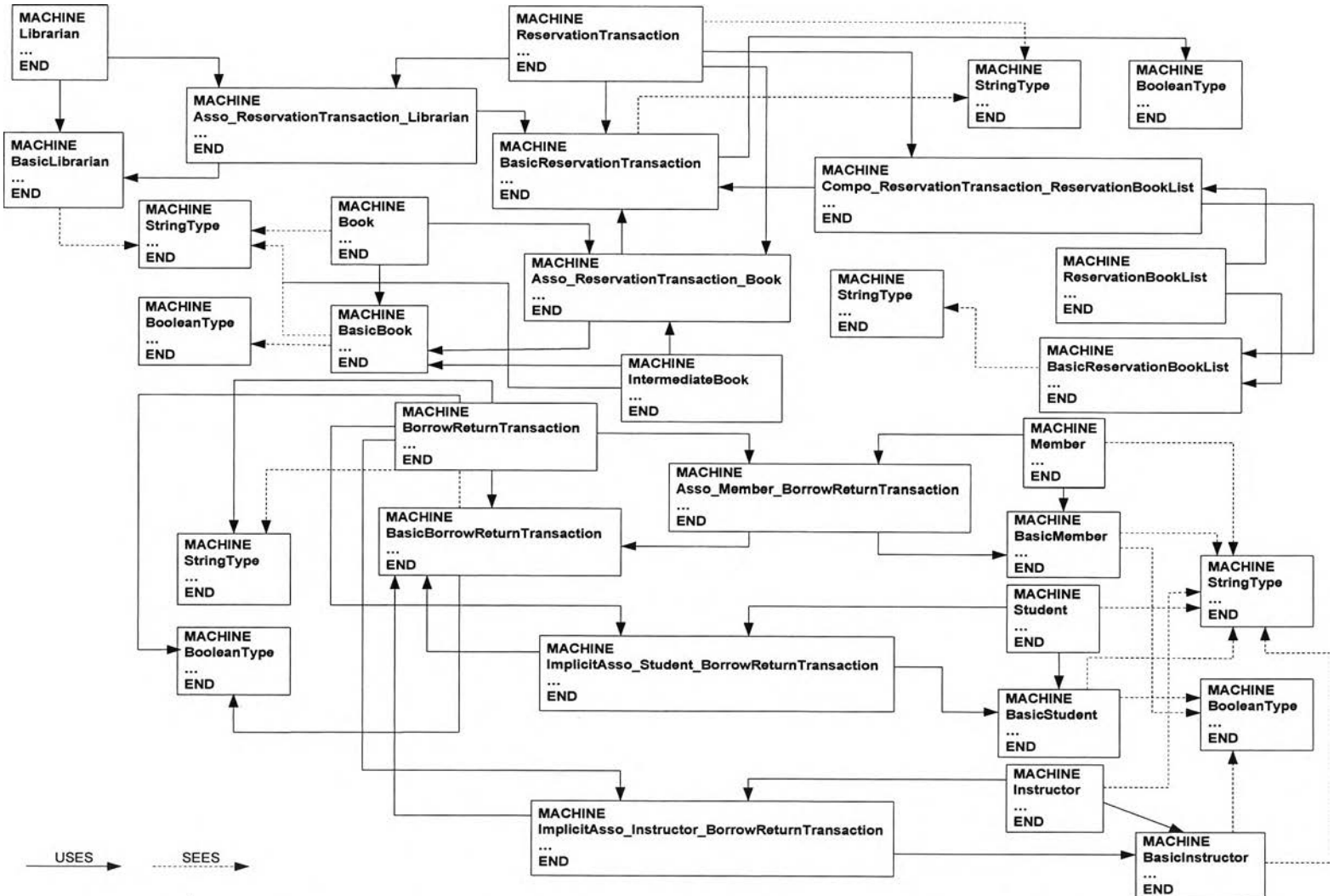
ภาคผนวก ค

สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของระบบการยืมคืนหนังสือในห้องสมุด



รูปที่ ค-1 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพคลาสในระบบการยืมคืนหนังสือในห้องสมุด

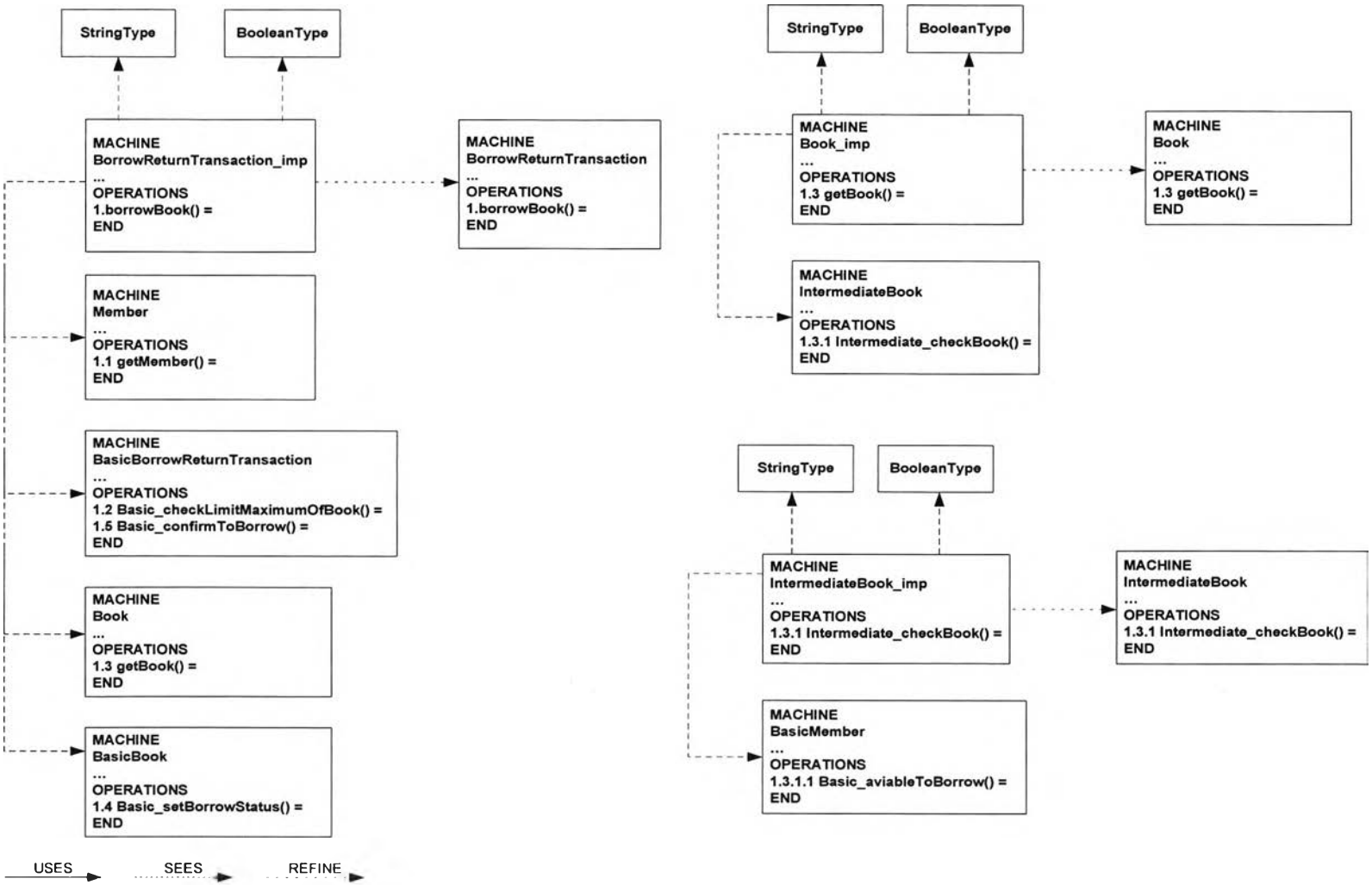
Architecture B Abstract Machine Case Study Library System - Class Diagram (Cont)



รูปที่ ค-1 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพคลาสในระบบการยืมคืนหนังสือในห้องสมุด (ต่อ)

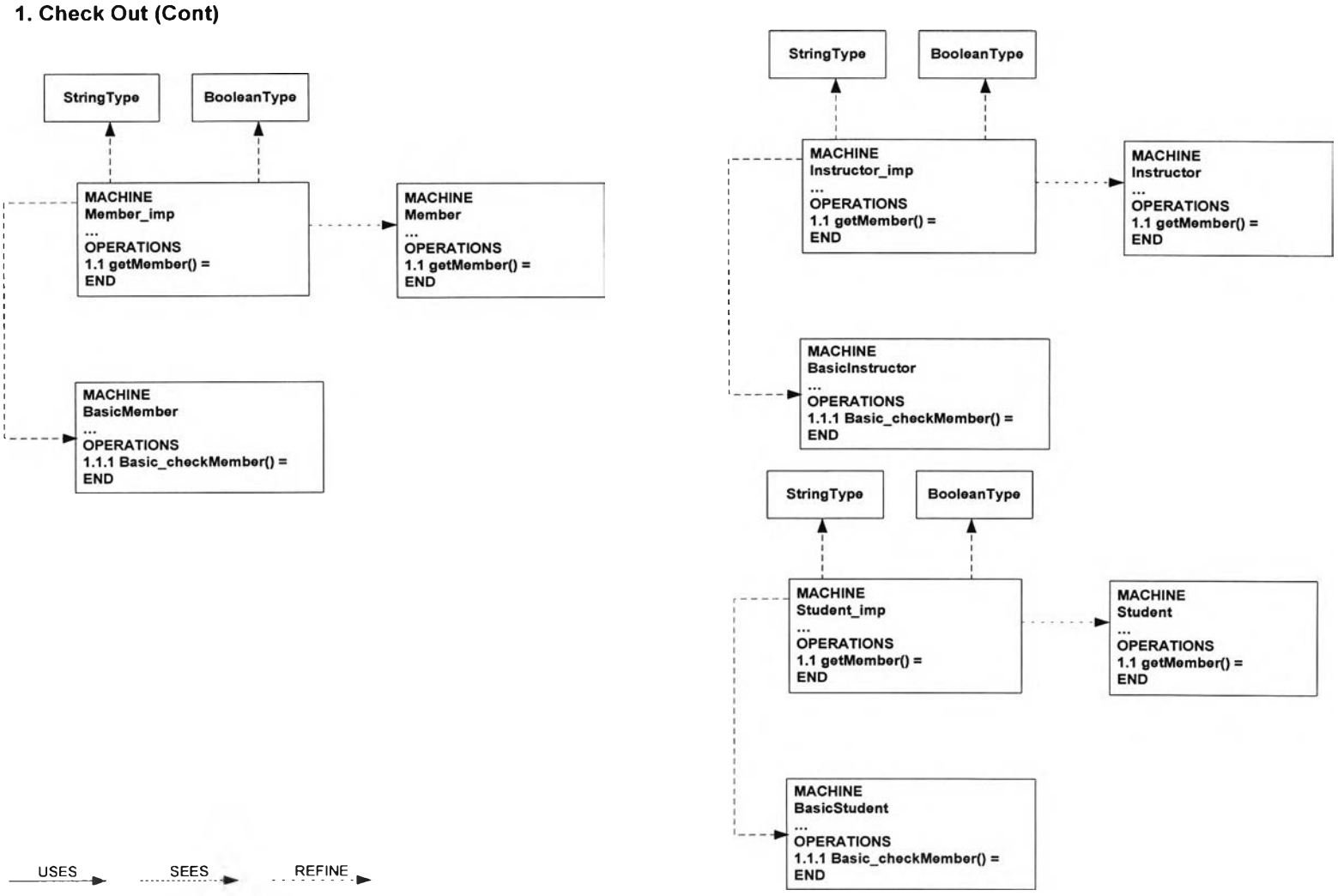
Architecture B Abstract Machine Case Study Library System - Sequence Diagram

1. Check Out



รูปที่ ค-2 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การยืมหนังสือในระบบการยืมคืนหนังสือในห้องสมุด

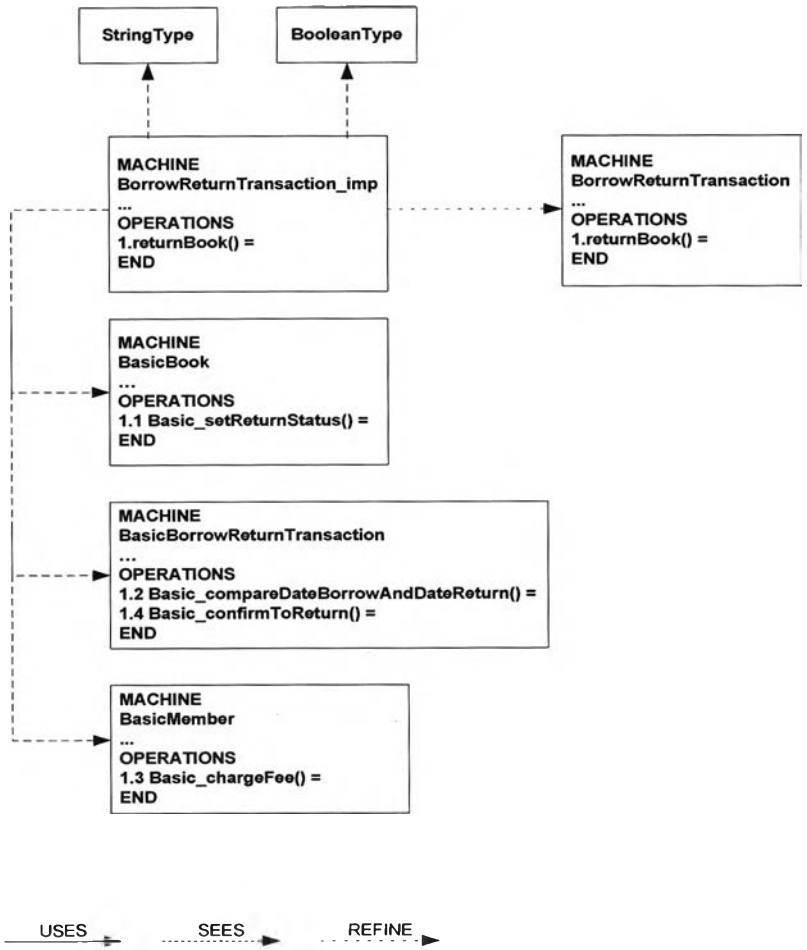
Architecture B Abstract Machine Case Study Library System - Sequence Diagram (Cont)



รูปที่ ค-2 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การยืมหนังสือในระบบการยืมคืนหนังสือในห้องสมุด (ต่อ)

Architecture B Abstract Machine Case Study Library System - Sequence Diagram (Cont)

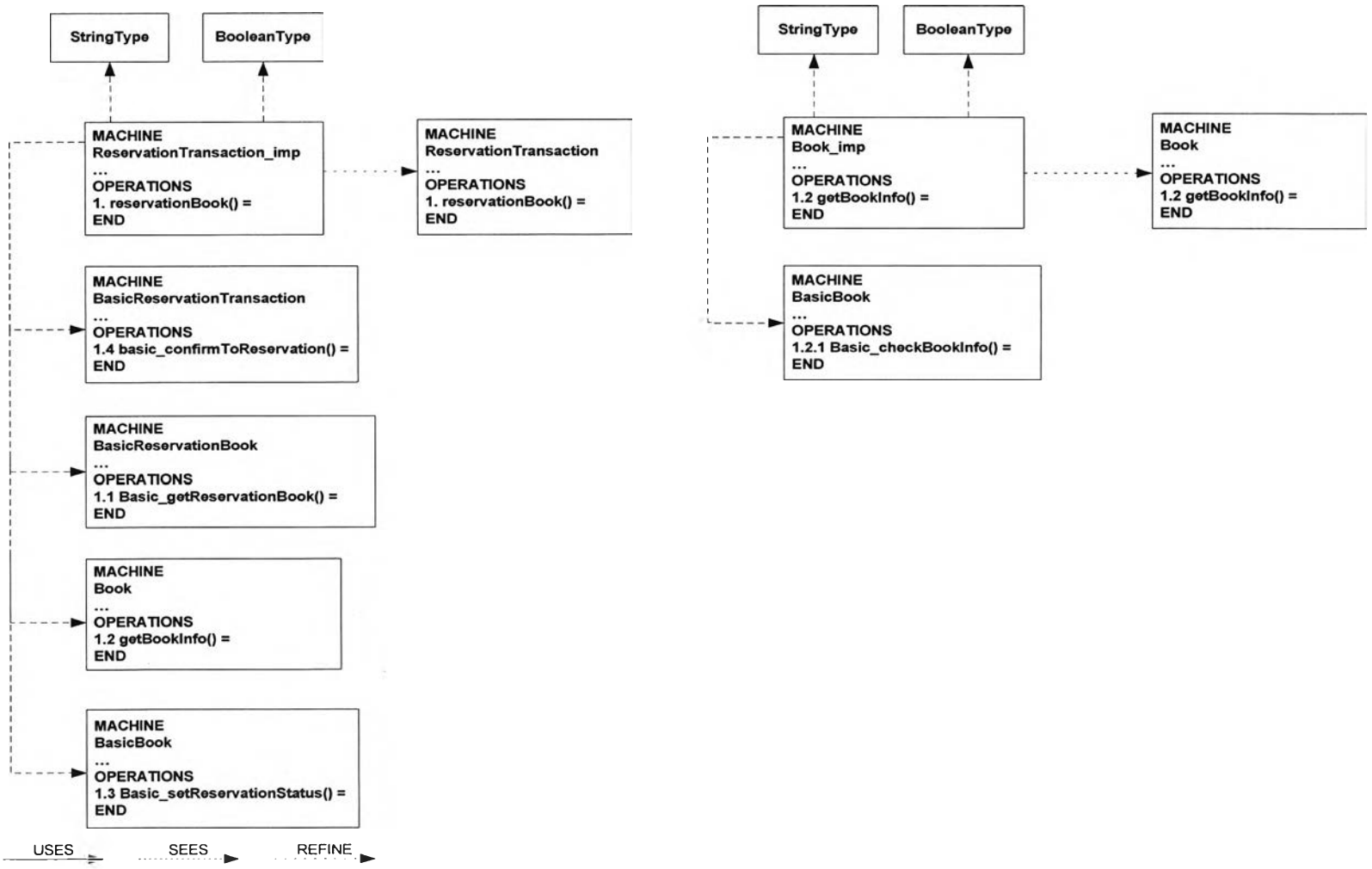
2. Return Book



รูปที่ ค-3 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การคืนหนังสือในระบบการยืมคืนหนังสือในห้องสมุด

Architecture B Abstract Machine Case Study Library System - Sequence Diagram (Cont)

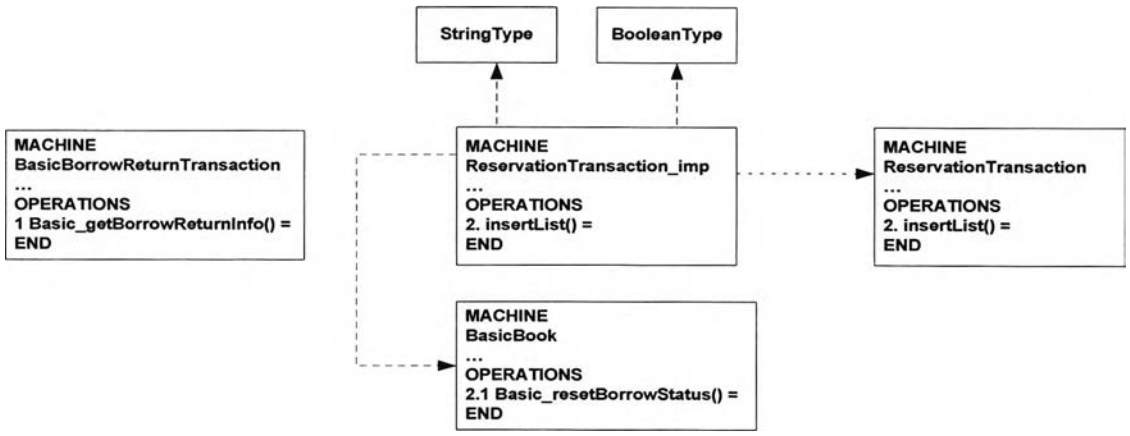
3. Make Reservation Book



รูปที่ ค-4 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การจองหนังสือในระบบการยืมคืนหนังสือในห้องสมุด

Architecture B Abstract Machine Case Study Library System - Sequence Diagram (Cont)

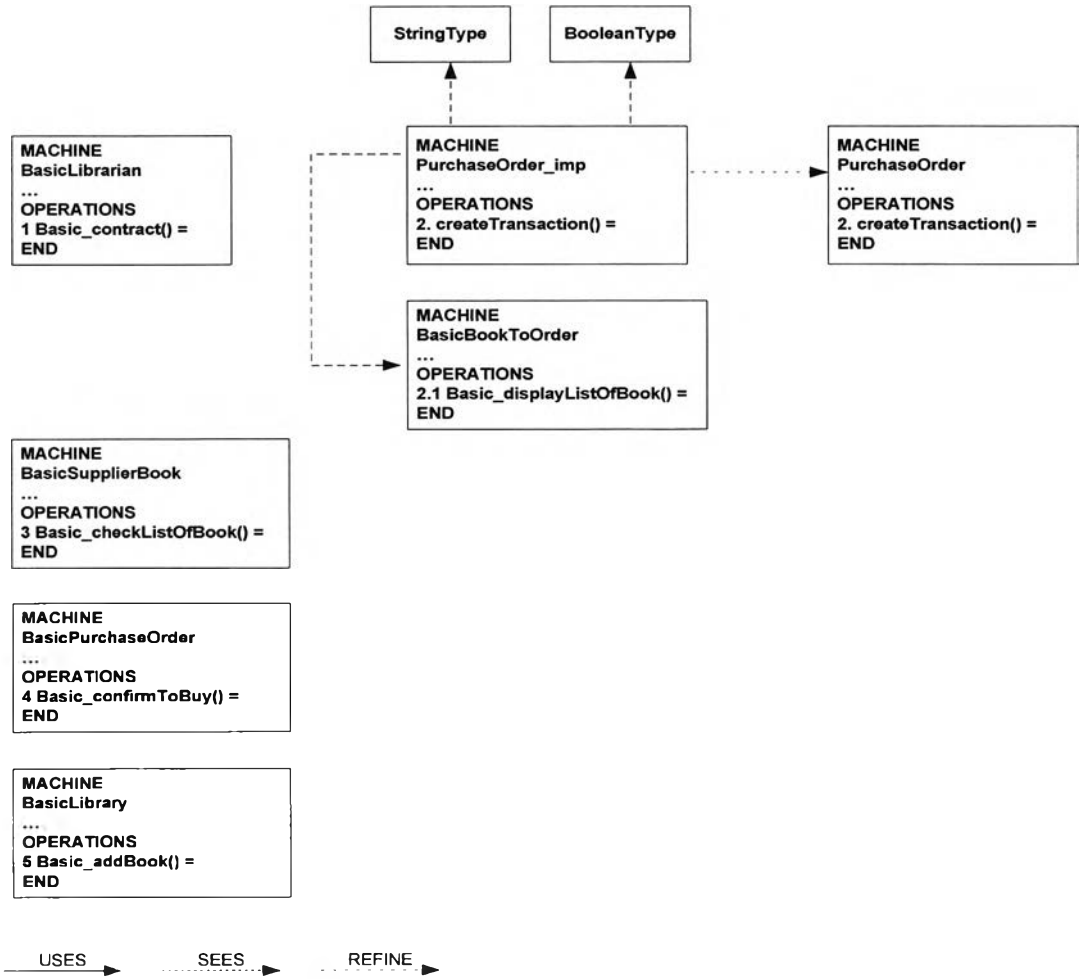
4. Borrow Book Reservation



รูปที่ ค-5 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การยืมหนังสือที่จองในระบบการยืมคืนหนังสือในห้องสมุด

Architecture B Abstract Machine Case Study Library System - Sequence Diagram (Cont)

5. Buy New Book



รูปที่ ค-6 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การสั่งซื้อหนังสือในระบบการยืมคืนหนังสือในห้องสมุด

ภาคผนวก ง
แอ็บสแตร็คแมชชีนบีของระบบการฝากและถอนเงินในธนาคาร

- ไลบรารีแอ็บสแตร็คแมชชีนบีทั้งหมด มีดังนี้

- ไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน (BooleanType)

```
MACHINE
BooleanType

SETS
BOOLEAN = {TRUE,FALSE}

END
```

- ไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระ (StringType)

```
MACHINE
StringType

SETS
STRING

VARIABLES
null,
EmptyString

INVARIANT
null : STRING &
EmptyString : STRING

INITIALISATION
null := EmptyString

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสผู้ฝาก (BAM BasicDepositor)

```
MACHINE
BasicDepositor

SEES
StringType

SETS
BASICDEPOSITOR
```

```
VARIABLES
basicdepositor,
depositor_ID,
depositor_Name,
depositor_Surname,
depositor_Address

INVARIANT
basicdepositor <: BASICDEPOSITOR &
depositor_ID : STRING &
depositor_Name : STRING &
depositor_Surname : STRING &
depositor_Address : STRING

INITIALISATION
basicdepositor := {} ||
depositor_ID := null ||
depositor_Name := null ||
depositor_Surname := null ||
depositor_Address := null

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสผู้ถอน (BAM BasicWithdrawer)

```
MACHINE
BasicWithdrawer

SEES
StringType

SETS
BASICWITHDRAWER

VARIABLES
basicwithdrawer,
withdrawer_ID,
withdrawer_Name,
withdrawer_Surname,
withdrawer_Address

INVARIANT
basicwithdrawer <: BASICWITHDRAWER &
withdrawer_ID : STRING &
withdrawer_Name : STRING &
withdrawer_Surname : STRING &
withdrawer_Address : STRING

INITIALISATION
basicwithdrawer := {} ||
withdrawer_ID := null ||
withdrawer_Name := null ||
withdrawer_Surname := null ||
withdrawer_Address := null
```

END

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายการการฝากเงิน (BAM BasicDepositTransaction)

```
MACHINE
BasicDepositTransaction

SEES
StringType,
BooleanType

SETS
BASICDEPOSITTRANSACTION

VARIABLES
basicdeposittransaction,
deposittransaction_Date,
deposittransaction_ID,
deposittransaction_DepositAmount

INVARIANT
basicdeposittransaction <: BASICDEPOSITTRANSACTION &
deposittransaction_Date : STRING &
deposittransaction_ID : STRING &
deposittransaction_DepositAmount : NAT

INITIALISATION
basicdeposittransaction := {} ||
deposittransaction_Date := null ||
deposittransaction_ID := null ||
deposittransaction_DepositAmount := 0

OPERATIONS
boolean <-- Basic_createDepositTransaction(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

boolean <-- Basic_approveToDeposit(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;
```

```
basicdeposittransaction1 <-- Basic_listDepositStatement(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
basicdeposittransaction1 := 0
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายการการถอนเงิน (BAM BasicWithdrawTransaction)

```
MACHINE
BasicWithdrawTransaction

SEES
StringType,
BooleanType

SETS
BASICWITHDRAWTRANSACTION

VARIABLES
basicwithdrawtransaction,
withdrawtransaction_Date,
withdrawtransaction_ID,
withdrawtransaction_WithdrawAmount

INVARIANT
basicwithdrawtransaction <: BASICWITHDRAWTRANSACTION &
withdrawtransaction_Date : STRING &
withdrawtransaction_ID : STRING &
withdrawtransaction_WithdrawAmount : NAT

INITIALISATION
basicwithdrawtransaction := {} ||
withdrawtransaction_Date := null ||
withdrawtransaction_ID := null ||
withdrawtransaction_WithdrawAmount := 0

OPERATIONS
boolean <-- Basic_createWithdrawTransaction(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;
```

```
boolean <-- Basic_approveToWithdraw(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

basicwithdrawtransaction1 <-- Basic_listWithdrawStatement(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
basicwithdrawtransaction1 := 0
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคซูเปอร์คลาสเจ้าหน้าที่ (BAM BasicStaff)

```
MACHINE
BasicStaff

SEES
StringType

SETS
BASICSTAFF

VARIABLES
basicstaff,
staff_ID,
staff_Name,
staff_Position

INVARIANT
basicstaff <: BASICSTAFF &
staff_ID : STRING &
staff_Name : STRING &
staff_Position : STRING

INITIALISATION
basicstaff := {} ||
staff_ID := null ||
staff_Name := null ||
staff_Position := null

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคซับคลาสพนักงานฝากเงิน (BAM BasicDepositOfficer)

```
MACHINE
BasicDepositOfficer

SEES
StringType

SETS
BASICDEPOSITOFFICER

VARIABLES
basicdepositofficer,
depositofficer_ID,
depositofficer_Name,
depositofficer_Position

INVARIANT
basicdepositofficer <: BASICDEPOSITOFFICER &
depositofficer_ID : STRING &
depositofficer_Name : STRING &
depositofficer_Position : STRING

INITIALISATION
basicdepositofficer := {} ||
depositofficer_ID := null ||
depositofficer_Name := null ||
depositofficer_Position := null

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคซับคลาสพนักงานถอนเงิน (BAM BasicWithdrawOfficer)

```
MACHINE
BasicWithdrawOfficer

SEES
StringType

SETS
BASICWITHDRAWOFFICER

VARIABLES
basicwithdrawofficer,
withdrawofficer_ID,
withdrawofficer_Name,
withdrawofficer_Position

INVARIANT
basicwithdrawofficer <: BASICWITHDRAWOFFICER &
withdrawofficer_ID : STRING &
withdrawofficer_Name : STRING &
withdrawofficer_Position : STRING
```

INITIALISATION
basicwithdrawofficer := {} ||
withdrawofficer_ID := null ||
withdrawofficer_Name := null ||
withdrawofficer_Position := null

END

- แอ็บสแตร็คแมชชีนบีเบซิคซับคลาสผู้จัดการสาขา (BAM BasicBranchManager)

MACHINE
BasicBranchManager

SEES
StringType

SETS
BASICBRANCHMANAGER

VARIABLES
basicbranchmanager,
branchmanager_ID,
branchmanager_Name,
branchmanager_Position

INVARIANT
basicbranchmanager <: BASICBRANCHMANAGER &
branchmanager_ID : STRING &
branchmanager_Name : STRING &
branchmanager_Position : STRING

INITIALISATION
basicbranchmanager := {} ||
branchmanager_ID := null ||
branchmanager_Name := null ||
branchmanager_Position := null

END

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสสถานภาพทางการเงิน (BAM BasicMoneyStatus)

MACHINE
BasicMoneyStatus

SEES
StringType

SETS
BASICMONEYSTATUS

VARIABLES
basicmoneystatus,
moneystatus_LastedDate,
moneystatus_Amount,
RefDayBalanceID

INVARIANT
basicmoneystatus <: BASICMONEYSTATUS &
moneystatus_LastedDate : STRING &
moneystatus_Amount : NAT &
RefDayBalanceID : STRING

INITIALISATION
basicmoneystatus := {} ||
moneystatus_LastedDate := null ||
moneystatus_Amount := 0 ||
RefDayBalanceID := null

END

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสบัญชีเงินฝาก (BAM BasicSavingAccount)

MACHINE
BasicSavingAccount

SEES
StringType

SETS
BASICSAVINGACCOUNT

VARIABLES
basicsavingaccount,
savingaccount_Date,
savingaccount_ID,
savingaccount_Name,
savingaccount_Type,
savingaccount_Balance

INVARIANT
basicsavingaccount <: BASICSAVINGACCOUNT &
savingaccount_Date : STRING &
savingaccount_ID : STRING &
savingaccount_Name : STRING &
savingaccount_Type : STRING &
savingaccount_Balance : NAT

INITIALISATION
basicsavingaccount := {} ||
savingaccount_Date := null ||
savingaccount_ID := null ||
savingaccount_Name :=null ||
savingaccount_Type := null ||
savingaccount_Balance := 0

OPERATIONS
basicsavingaccount1 <-- Basic_increaseBalance(a1,x1) =
PRE
a1 : STRING &
x1 : NAT
THEN

```
/* User can insert or not insert condition's operation by using IF here */
basicsavingaccount1 := 0
/* User can modify output value or post condition of operation here */
END;

basicsavingaccount2 <-- Basic_getCurrentAmount =
BEGIN
basicsavingaccount2 := 0
/* User can modify output value or post condition of operation here */
END;

basicsavingaccount3 <-- Basic_decreaseBalance(a1,x1) =
PRE
a1 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
basicsavingaccount3 := 0
/* User can modify output value or post condition of operation here */
END;

basicsavingaccount4 <-- Basic_checkBalance(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
basicsavingaccount4 := 0
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสยอดเงินคงค้าง (BAM BasicDayBalance)

```
MACHINE
BasicDayBalance

SEES
StringType

SETS
BASICDAYBALANCE

VARIABLES
basicdaybalance,
daybalance_Date,
daybalance_IncreaseAmount,
daybalance_DecreaseAmount,
daybalance_TotalAmount

INVARIANT
basicdaybalance <: BASICDAYBALANCE &
daybalance_Date : STRING &
daybalance_IncreaseAmount : NAT &
```

```
daybalance_DecreaseAmount : NAT &
daybalance_TotalAmount : NAT

INITIALISATION
basicdaybalance := {} ||
daybalance_Date := null ||
daybalance_IncreaseAmount := 0 ||
daybalance_DecreaseAmount := 0 ||
daybalance_TotalAmount := 0

OPERATIONS
basicdaybalance1 <-- Basic_increaseAmount(x1) =
PRE
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
basicdaybalance1 := 0
/* User can modify output value or post condition of operation here */
END;

basicdaybalance2 <-- Basic_decreaseAmount(x1) =
PRE
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
basicdaybalance2 := 0
/* User can modify output value or post condition of operation here */
END

basicdaybalance3 <-- Basic_getDialyBalance =
BEGIN
basicdaybalance3 := 0
/* User can modify output value or post condition of operation here */
END;

END
```

- แอ็บสแตร็คแมชชีนบีคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีคลาสผู้ฝาก (BAM Depositor)

```
MACHINE
Depositor

USES
BasicDepositor,
AssoRequest_DepositTransaction_Depositor

END
```

- แอ็บสแตร็คแมชชีนบีคลาสผู้ถอน (BAM Withdrawer)

```
MACHINE
Withdrawer

USES
BasicWithdrawer,
AssoRequest_WithdrawTransaction_Withdrawer

END
```

- แอ็บสแตร็คแมชชีนบีคลาสรายการการฝากเงิน (BAM DepositTransaction)

```
MACHINE
DepositTransaction

SEES
StringType

USES
BasicDepositTransaction,
AssoDeposit_SavingAccount_DepositTransaction,
AssoAffect_DayBalance_DepositTransaction,
AssoRequest_DepositTransaction_Depositor,
Compo_DepositTransaction_DepositOfficer

OPERATIONS
deposittransaction1 <-- setDepositAmount(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
deposittransaction1 := 0
/* User can modify output value or post condition of operation here */
END;

deposittransaction2 <-- postDepositTransaction(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
deposittransaction2 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสรายการการถอนเงิน (BAM WithdrawTransaction)

```
MACHINE
WithdrawTransaction

SEES
StringType

USES
BasicWithdrawTransaction,
AssoWithdraw_SavingAccount_WithdrawTransaction,
AssoAffect_DayBalance_WithdrawTransaction,
AssoRequest_WithdrawTransaction_Withdrawer,
Compo_WithdrawTransaction_WithdrawOfficer

OPERATIONS
withdrawtransaction1 <-- setWithdrawAmount(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
withdrawtransaction1 := 0
/* User can modify output value or post condition of operation here */
END;

withdrawtransaction2 <-- postWithdrawTransaction(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
withdrawtransaction2 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีซูเปอร์คลาสเจ้าหน้าที่ (BAM Staff)

```
MACHINE
Staff

USES
BasicStaff,
AssoDo_SavingAccount_Staff

END
```

- แอ็บสแตร็คแมชชีนบีซับคลาสพนักงานฝากเงิน (BAM DepositOfficer)

```
MACHINE
DepositOfficer

USES
BasicDepositOfficer,
Compo_DepositTransaction_DepositOfficer,
ImplicitAssoDo_SavingAccount_DepositOfficer

END
```

- แอ็บสแตร็คแมชชีนบีซับคลาสพนักงานถอนเงิน (BAM WithdrawOfficer)

```
MACHINE
WithdrawOfficer

USES
BasicWithdrawOfficer,
Compo_WithdrawTransaction_WithdrawOfficer,
ImplicitAssoDo_SavingAccount_WithdrawOfficer

END
```

- แอ็บสแตร็คแมชชีนบีซับคลาสผู้จัดการสาขา (BAM BranchManager)

```
MACHINE
BranchManager

USES
BasicBranchManager,
AssoDo_MoneyStatus_BranchManager,
ImplicitAssoDo_SavingAccount_BranchManager

END
```

- แอ็บสแตร็คแมชชีนบีคลาสสถานภาพทางการเงิน (BAM MoneyStatus)

```
MACHINE
MoneyStatus

SEES
StringType

USES
BasicMoneyStatus,
Aggr_MoneyStatus_DayBalance,
AssoDo_MoneyStatus_BranchManager

OPERATIONS
moneystatus1 <-- listDialyStatus(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
```

```
THEN
/* User can insert or not insert condition's operation by using IF here */
moneystatus1 := 0
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสบัญชีเงินฝาก (BAM SavingAccount)

```
MACHINE
SavingAccount

SEES
StringType

USES
BasicSavingAccount,
AssoDeposit_SavingAccount_DepositTransaction,
AssoWithdraw_SavingAccount_WithdrawTransaction,
AssoDo_SavingAccount_Staff,
ImplicitAssoDo_SavingAccount_BranchManager,
ImplicitAssoDo_SavingAccount_DepositOfficer,
ImplicitAssoDo_SavingAccount_WithdrawOfficer

OPERATIONS
savingaccount1 <-- listBalance(a1,a2,a3) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
savingaccount1 := 0
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสยอดเงินคงค้าง (BAM DayBalance)

```
MACHINE
DayBalance

USES
BasicDayBalance,
AssoAffect_DayBalance_DepositTransaction,
AssoAffect_DayBalance_WithdrawTransaction,
Aggr_MoneyStatus_DayBalance

END
```

- แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสรายการฝากเงิน (BAM IntermediateDepositTransaction)

```
MACHINE
IntermediateDepositTransaction

SEES
StringType,
BooleanType

USES
BasicDepositTransaction,
AssoDeposit_SavingAccount_DepositTransaction,
AssoAffect_DayBalance_DepositTransaction,
AssoRequest_DepositTransaction_Depositor,
Compo_DepositTransaction_DepositOfficer

OPERATIONS
boolean <-- Intermediate_approveToDeposit(a1,a2,x1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสรายการถอนเงิน (BAM IntermediateWithdrawTransaction)

```
MACHINE
IntermediateWithdrawTransaction

SEES
StringType,
BooleanType

USES
BasicWithdrawTransaction,
AssoWithdraw_SavingAccount_WithdrawTransaction,
AssoAffect_DayBalance_WithdrawTransaction,
AssoRequest_WithdrawTransaction_Withdrawer,
Compo_WithdrawTransaction_WithdrawOfficer

OPERATIONS
boolean <-- Intermediate_approveToWithdraw(a1,a2,x1) =
PRE
a1 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
```

END

END

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์ทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันทำระหว่างคลาสบัญชีเงินฝากกับคลาสเจ้าหน้าที่ (AssoDo_SavingAccount_Staff)

```
MACHINE
AssoDo_SavingAccount_Staff

USES
BasicSavingAccount,
BasicStaff

VARIABLES
assodo_savingaccount_staff

INVARIANT
assodo_savingaccount_staff <: BASICSAVINGACCOUNT * BASICSTAFF &
dom(assodo_savingaccount_staff) = basicsavingaccount &
ran(assodo_savingaccount_staff) = basicstaff &
!(xx,yy).(((xx : dom(assodo_savingaccount_staff)) &
            (yy : ran(assodo_savingaccount_staff)))
        => card((assodo_savingaccount_staff)[{xx}]) = 1 &
           card((assodo_savingaccount_staff)~[{yy}]) >= 1)

INITIALISATION
assodo_savingaccount_staff := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันฝากเงินระหว่างคลาสบัญชีเงินฝากกับคลาสรายการการฝากเงิน (AssoDeposit_SavingAccount_DepositTransaction)

```
MACHINE
AssoDeposit_SavingAccount_DepositTransaction

USES
BasicSavingAccount,
BasicDepositTransaction

VARIABLES
assodeposit_savingaccount_deposittransaction

INVARIANT
assodeposit_savingaccount_deposittransaction <: BASICSAVINGACCOUNT *
BASICDEPOSITTRANSACTION &
dom(assodeposit_savingaccount_deposittransaction) = basicsavingaccount &
ran(assodeposit_savingaccount_deposittransaction) = basicdeposittransaction &
```

```
!(xx,yy).(((xx : dom(assodeposit_savingaccount_deposittransaction)) &
          (yy : ran(assodeposit_savingaccount_deposittransaction)))
        => card((assodeposit_savingaccount_deposittransaction)[{xx}]) >= 0 &
           card((assodeposit_savingaccount_deposittransaction)~[{yy}]) = 1)

INITIALISATION
assodeposit_savingaccount_deposittransaction := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันถอนเงินระหว่างคลาสบัญชีเงินฝากกับคลาสรายการการถอนเงิน (AssoWithdraw_SavingAccount_WithdrawTransaction)

```
MACHINE
AssoWithdraw_SavingAccount_WithdrawTransaction

USES
BasicSavingAccount,
BasicWithdrawTransaction

VARIABLES
assowithdraw_savingaccount_withdrawtransaction

INVARIANT
assowithdraw_savingaccount_withdrawtransaction <: BASICSAVINGACCOUNT *
BASICWITHDRAWTRANSACTION &
dom(assowithdraw_savingaccount_withdrawtransaction) = basicsavingaccount &
ran(assowithdraw_savingaccount_withdrawtransaction) = basicwithdrawtransaction &
!(xx,yy).(((xx : dom(assowithdraw_savingaccount_withdrawtransaction)) &
          (yy : ran(assowithdraw_savingaccount_withdrawtransaction)))
        => card((assowithdraw_savingaccount_withdrawtransaction)[{xx}]) >= 0 &
           card((assowithdraw_savingaccount_withdrawtransaction)~[{yy}]) = 1)

INITIALISATION
assowithdraw_savingaccount_withdrawtransaction := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันแจ้งความจำนงฝากเงินระหว่างคลาสรายการการฝากเงินกับคลาสผู้ฝาก (AssoRequest_DepositTransaction_Depositor)

```
MACHINE
AssoRequest_DepositTransaction_Depositor

USES
BasicDepositTransaction,
BasicDepositor

VARIABLES
assorequest_deposittransaction_depositor

INVARIANT
assorequest_deposittransaction_depositor <: BASICDEPOSITTRANSACTION * BASICDEPOSITOR &
```

```
dom(assorequest_deposittransaction_depositor) = basicdeposittransaction &
ran(assorequest_deposittransaction_depositor) = basicdepositor &
!(xx,yy).(((xx : dom(assorequest_deposittransaction_depositor)) &
            (yy : ran(assorequest_deposittransaction_depositor)))
        => card((assorequest_deposittransaction_depositor)[{xx}]) = 1 &
            card((assorequest_deposittransaction_depositor)~[{yy}]) >= 1)

INITIALISATION
assorequest_deposittransaction_depositor := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันแจ้งความจำนงถอนเงินระหว่างคลาสรายการการถอนเงินกับคลาสผู้ถอน (AssoRequest_WithdrawTransaction_Withdrawer)

```
MACHINE
AssoRequest_WithdrawTransaction_Withdrawer

USES
BasicWithdrawTransaction,
BasicWithdrawer

VARIABLES
assorequest_withdrawtransaction_withdrawer

INVARIANT
assorequest_withdrawtransaction_withdrawer <: BASICWITHDRAWTRANSACTION *
BASICWITHDRAWER &
dom(assorequest_withdrawtransaction_withdrawer) = basicwithdrawtransaction &
ran(assorequest_withdrawtransaction_withdrawer) = basicwithdrawer &
!(xx,yy).(((xx : dom(assorequest_withdrawtransaction_withdrawer)) &
            (yy : ran(assorequest_withdrawtransaction_withdrawer)))
        => card((assorequest_withdrawtransaction_withdrawer)[{xx}]) = 1 &
            card((assorequest_withdrawtransaction_withdrawer)~[{yy}]) >= 1)

INITIALISATION
assorequest_withdrawtransaction_withdrawer:= {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันทำระหว่างคลาสสถานภาพทางการเงินกับคลาสผู้จัดการสาขา (AssoDo_MoneyStatus_BranchManager)

```
MACHINE
AssoDo_MoneyStatus_BranchManager

USES
BasicMoneyStatus,
BasicBranchManager

VARIABLES
assodo_moneystatus_branchmanager
```

```
INVARIANT
assodo_moneystatus_branchmanager <: BASICMONEYSTATUS * BASICBRANCHMANAGER &
dom(assodo_moneystatus_branchmanager) = basicmoneystatus &
ran(assodo_moneystatus_branchmanager) = basicbranchmanager &
!(xx,yy).(((xx : dom(assodo_moneystatus_branchmanager)) &
            (yy : ran(assodo_moneystatus_branchmanager)))
        => card((assodo_moneystatus_branchmanager)[{xx}]) = 1 &
            card((assodo_moneystatus_branchmanager)~[{yy}]) >= 1)

INITIALISATION
assodo_moneystatus_branchmanager := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันมีผลกระทบระหว่างคลาสยอดเงินคงค้างกับคลาส

  รายการการฝากเงิน (AssoAffect_DayBalance_DepositTransaction)

```
MACHINE
AssoAffect_DayBalance_DepositTransaction

USES
BasicDayBalance,
BasicDepositTransaction

VARIABLES
assoaffect_daybalance_deposittransaction

INVARIANT
assoaffect_daybalance_deposittransaction <: BASICDAYBALANCE * BASICDEPOSITTRANSACTION
&
dom(assoaffect_daybalance_deposittransaction) = basicdaybalance &
ran(assoaffect_daybalance_deposittransaction) = basicdeposittransaction &
!(xx,yy).(((xx : dom(assoaffect_daybalance_deposittransaction)) &
            (yy : ran(assoaffect_daybalance_deposittransaction)))
        => card((assoaffect_daybalance_deposittransaction)[{xx}]) >= 0 &
            card((assoaffect_daybalance_deposittransaction)~[{yy}]) = 1)

INITIALISATION
assoaffect_daybalance_deposittransaction := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันมีผลกระทบระหว่างคลาสยอดเงินคงค้างกับคลาส

  รายการการถอนเงิน (AssoAffect_DayBalance_WithdrawTransaction)

```
MACHINE
AssoAffect_DayBalance_WithdrawTransaction

USES
BasicDayBalance,
BasicWithdrawTransaction
```

```
VARIABLES
assoaffect_daybalance_withdrawtransaction

INVARIANT
assoaffect_daybalance_withdrawtransaction <: BASICDAYBALANCE *
BASICWITHDRAWTRANSACTION &
dom(assoaffect_daybalance_withdrawtransaction) = basicdaybalance &
ran(assoaffect_daybalance_withdrawtransaction) = basicwithdrawtransaction &
!(xx,yy).(((xx : dom(assoaffect_daybalance_withdrawtransaction)) &
            (yy : ran(assoaffect_daybalance_withdrawtransaction)))
        => card((assoaffect_daybalance_withdrawtransaction)[{xx}]) >= 0 &
            card((assoaffect_daybalance_withdrawtransaction)~[{yy}]) = 1)

INITIALISATION
assoaffect_daybalance_withdrawtransaction := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสสถานภาพทางการเงินกับคลาสยอดเงินคงค้าง (Aggr_MoneyStatus_DayBalance)

```
MACHINE
Aggr_MoneyStatus_DayBalance

USES
BasicMoneyStatus,
BasicDayBalance

VARIABLES
aggr_moneystatus_daybalance

INVARIANT
aggr_moneystatus_daybalance <: BASICMONEYSTATUS * BASICDAYBALANCE &
dom(aggr_moneystatus_daybalance) = basicmoneystatus &
ran(aggr_moneystatus_daybalance) = basicdaybalance &
!(xx,yy).(((xx : dom(aggr_moneystatus_daybalance)) &
            (yy : ran(aggr_moneystatus_daybalance)))
        => card((aggr_moneystatus_daybalance)[{xx}]) >= 1 &
           (card((aggr_moneystatus_daybalance)~[{yy}]) >= 0 &
            card((aggr_moneystatus_daybalance)~[{yy}]) <= 1))

INITIALISATION
aggr_moneystatus_daybalance := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสรายการการฝากกับคลาสพนักงานฝากเงิน (Compo_DepositTransaction_DepositOfficer)

```
MACHINE
Compo_DepositTransaction_DepositOfficer

USES
BasicDepositTransaction,
BasicDepositOfficer

VARIABLES
compo_deposittransaction_depositofficer

INVARIANT
compo_deposittransaction_depositofficer <: BASICDEPOSITTRANSACTION *
BASICDEPOSITOFFICER &
dom(compo_deposittransaction_depositofficer) = basicdeposittransaction &
ran(compo_deposittransaction_depositofficer) = basicdepositofficer &
!(xx,yy).(((xx : dom(compo_deposittransaction_depositofficer)) &
            (yy : ran(compo_deposittransaction_depositofficer)))
        => card((compo_deposittransaction_depositofficer)[{xx}]) = 1 &
            card((compo_deposittransaction_depositofficer)~[{yy}]) >= 1)

INITIALISATION
compo_deposittransaction_depositofficer := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสรายการการถอนกับคลาสพนักงานถอนเงิน (Compo_WithdrawTransaction_WithdrawOfficer)

```
MACHINE
Compo_WithdrawTransaction_WithdrawOfficer

USES
BasicWithdrawTransaction,
BasicWithdrawOfficer

VARIABLES
compo_withdrawtransaction_withdrawofficer

INVARIANT
compo_withdrawtransaction_withdrawofficer <: BASICWITHDRAWTRANSACTION *
BASICWITHDRAWOFFICER &
dom(compo_withdrawtransaction_withdrawofficer) = basicwithdrawtransaction &
ran(compo_withdrawtransaction_withdrawofficer) = basicwithdrawofficer &
!(xx,yy).(((xx : dom(compo_withdrawtransaction_withdrawofficer)) &
            (yy : ran(compo_withdrawtransaction_withdrawofficer)))
        => card((compo_withdrawtransaction_withdrawofficer)[{xx}]) = 1 &
            card((compo_withdrawtransaction_withdrawofficer)~[{yy}]) >= 1)

INITIALISATION
compo_withdrawtransaction_withdrawofficer := {}
```

END

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันทำโดยปริยายระหว่างซับคลาสบัญชีเงินฝากกับคลาสพนักงานฝากเงิน (ImplicitAssoDo_SavingAccount_DepositOfficer)

```
MACHINE
ImplicitAssoDo_SavingAccount_DepositOfficer

USES
BasicSavingAccount,
BasicDepositOfficer

VARIABLES
implicitassodo_savingaccount_depositofficer

INVARIANT
implicitassodo_savingaccount_depositofficer <: BASICSAVINGACCOUNT * BASICDEPOSITOFFICER
&
dom(implicitassodo_savingaccount_depositofficer) = basicsavingaccount &
ran(implicitassodo_savingaccount_depositofficer) = basicdepositofficer &
!(xx,yy).(((xx : dom(implicitassodo_savingaccount_depositofficer)) &
            (yy : ran(implicitassodo_savingaccount_depositofficer)))
        => card((implicitassodo_savingaccount_depositofficer)[{xx}]) >= 1 &
            card((implicitassodo_savingaccount_depositofficer)~[{yy}])>= 1)

INITIALISATION
implicitassodo_savingaccount_depositofficer:= {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันทำโดยปริยายระหว่างซับคลาสบัญชีเงินฝากกับคลาสพนักงานถอนเงิน (ImplicitAssoDo_SavingAccount_WithdrawOfficer)

```
MACHINE
ImplicitAssoDo_SavingAccount_WithdrawOfficer

USES
BasicSavingAccount,
BasicWithdrawOfficer

VARIABLES
implicitassodo_savingaccount_withdrawofficer

INVARIANT
implicitassodo_savingaccount_withdrawofficer <: BASICSAVINGACCOUNT *
BASICWITHDRAWOFFICER &
dom(implicitassodo_savingaccount_withdrawofficer) = basicsavingaccount &
ran(implicitassodo_savingaccount_withdrawofficer) = basicwithdrawofficer &
!(xx,yy).(((xx : dom(implicitassodo_savingaccount_withdrawofficer)) &
            (yy : ran(implicitassodo_savingaccount_withdrawofficer)))
        => card((implicitassodo_savingaccount_withdrawofficer)[{xx}])>= 1 &
            card((implicitassodo_savingaccount_withdrawofficer)~[{yy}])>= 1)
```

```
INITIALISATION
implicitassodo_savingaccount_withdrawofficer:= {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันทำโดยปริยายระหว่างซับคลาสบัญชีเงินฝากกับคลาสผู้จัดการสาขา (ImplicitAssoDo_SavingAccount_BranchManager)

```
MACHINE
ImplicitAssoDo_SavingAccount_BranchManager

USES
BasicSavingAccount,
BasicBranchManager

VARIABLES
implicitassodo_savingaccount_branchmanager

INVARIANT
implicitassodo_savingaccount_branchmanager <: BASICSAVINGACCOUNT * BASICBRANCHMANAGER &
dom(implicitassodo_savingaccount_branchmanager) = basicsavingaccount &
ran(implicitassodo_savingaccount_branchmanager) = basicbranchmanager &
!(xx,yy).(((xx : dom(implicitassodo_savingaccount_branchmanager)) &
            (yy : ran(implicitassodo_savingaccount_branchmanager)))
        => card((implicitassodo_savingaccount_branchmanager)[{xx}]) >= 1 &
            card((implicitassodo_savingaccount_branchmanager)~[{yy}]) >= 1)

INITIALISATION
implicitassodo_savingaccount_branchmanager := {}

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีทั้งหมด มีดังนี้

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสรายการการฝากเงิน (DepositTransaction _imp)

```
IMPLEMENTATION
DepositTransaction_imp

REFINES
DepositTransaction

SEES
StringType,
BooleanType,
IntermediateDepositTransaction,
BasicSavingAccount

OPERATIONS
deposittransaction1 <-- setDepositAmount(a1,a2,x1) =
VAR
boolean
IN
boolean <-- Intermediate_approveToDeposit(a1,a2,x1)
```

```
END;

deposittransaction2 <-- postDepositTransaction(a1,a2,x1) =
VAR
basicsavingaccount1,
basicdaybalance1
IN
basicsavingaccount1 <-- Basic_increaseBalance(a1,x1);
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสรายการการฝากเงิน

  (IntermediateDepositTransaction _imp)

```
IMPLEMENTATION
IntermediateDepositTransaction_imp

REFINES
IntermediateDepositTransaction

SEES
StringType,
BooleanType,
BasicDayBalance

OPERATIONS
boolean <-- Intermediate_approveToDeposit(a1,a2,x1) =
VAR
basicdaybalance1
IN
IF
        boolean = TRUE
        THEN
        basicdaybalance1 <-- Basic_increaseAmount(x1)
        ELSE
        skip
        END
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสรายการการถอนเงิน (WithdrawTransaction_imp)

```
IMPLEMENTATION
WithdrawTransaction_imp

REFINES
WithdrawTransaction

SEES
StringType,
BooleanType,
IntermediateWithdrawTransaction,
```

```
BasicSavingAccount,
BasicDayBalance

OPERATIONS
withdrawtransaction1 <-- setWithdrawAmount(a1,a2,x1) =
VAR
basicsavingaccount2,
boolean
IN
basicsavingaccount2 <-- Basic_getCurrentAmount;
boolean <-- Intermediate_approveToWithdraw(a1,a2,x1)
END;

withdrawtransaction2 <-- postWithdrawTransaction(a1,a2,x1) =
VAR
basicsavingaccount3,
basicdaybalance2
IN
basicsavingaccount3 <-- Basic_decreaseBalance(a1,x1);
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสรายการการฝากเงิน

  (IntermediateWithdrawTransaction _imp)

```
IMPLEMENTATION
IntermediateWithdrawTransaction_imp

REFINES
IntermediateWithdrawTransaction

SEES
StringType,
BooleanType,
BasicDayBalance

OPERATIONS
boolean <-- Intermediate_approveToWithdraw(a1,a2,x1) =
VAR
basicdaybalance2
IN
IF
        boolean = TRUE
        THEN
        basicdaybalance2 <-- Basic_decreaseAmount(x1)
        ELSE
        skip
        END
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสสถานภาพทางการเงิน (MoneyStatus_imp)

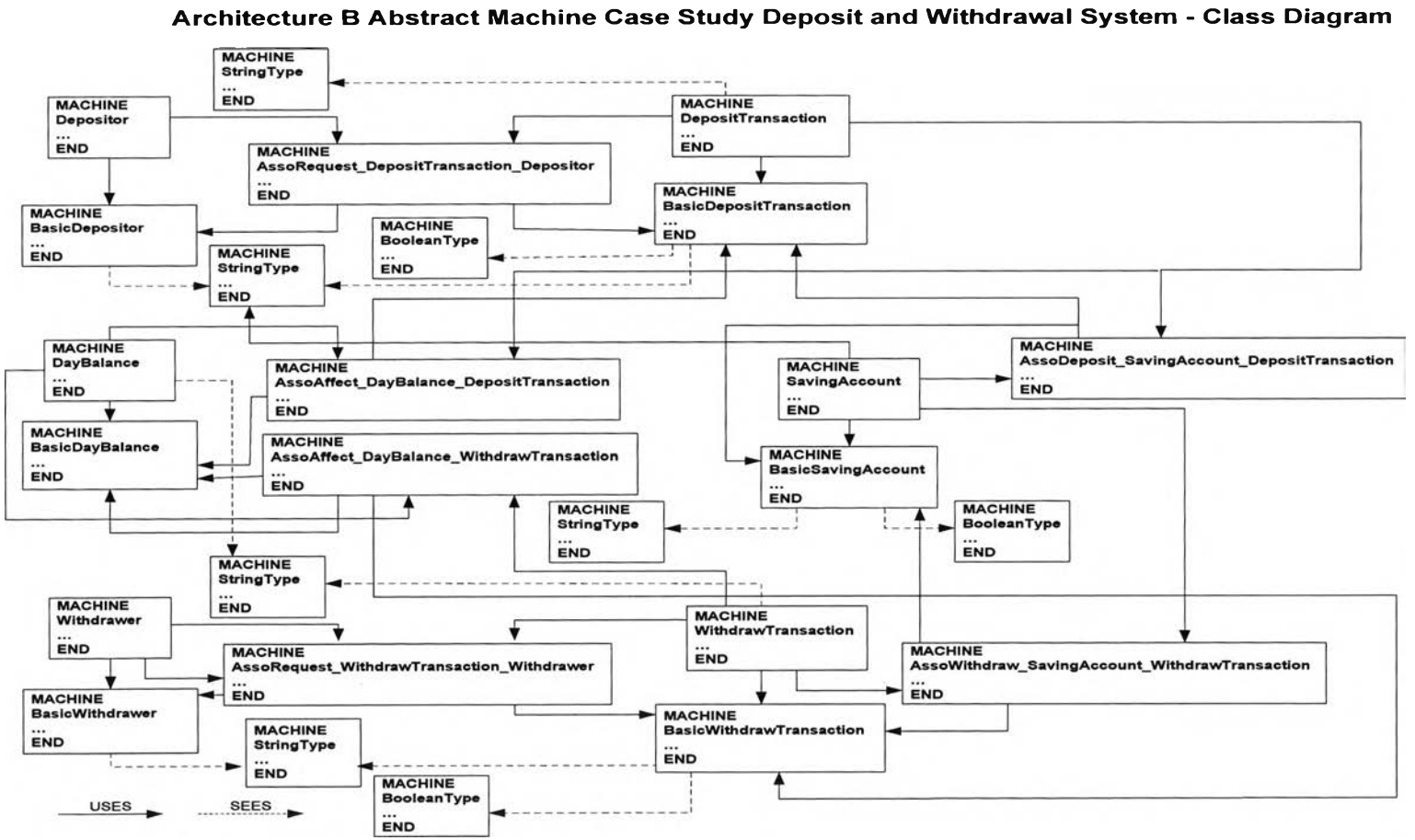
```
IMPLEMENTATION
MoneyStatus_imp

REFINES
MoneyStatus
SEES
StringType,
BasicDayBalance

OPERATIONS
moneystatus1 <-- listDailyStatus(a1,a2) =
VAR
basicdaybalance3
IN
basicdaybalance3 <-- Basic_getDailyBalance
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสบัญชีเงินฝาก (SavingAccount_imp)

```
IMPLEMENTATION
SavingAccount_imp

REFINES
SavingAccount

SEES
StringType,
BasicSavingAccount,
BasicDepositTransaction,
BasicWithdrawTransaction

OPERATIONS
savingaccount1 <-- listBalance(a1,a2,a3) =
VAR
basicsavingaccount4,
basicdeposittransaction1,
basicwithdrawtransaction1
IN
basicsavingaccount4 <-- Basic_checkBalance(a1,a2);
basicdeposittransaction1 <-- Basic_listDepositStatement(a1,a2);
basicwithdrawtransaction1 <-- Basic_listWithdrawStatement(a1,a2)
END

END
```

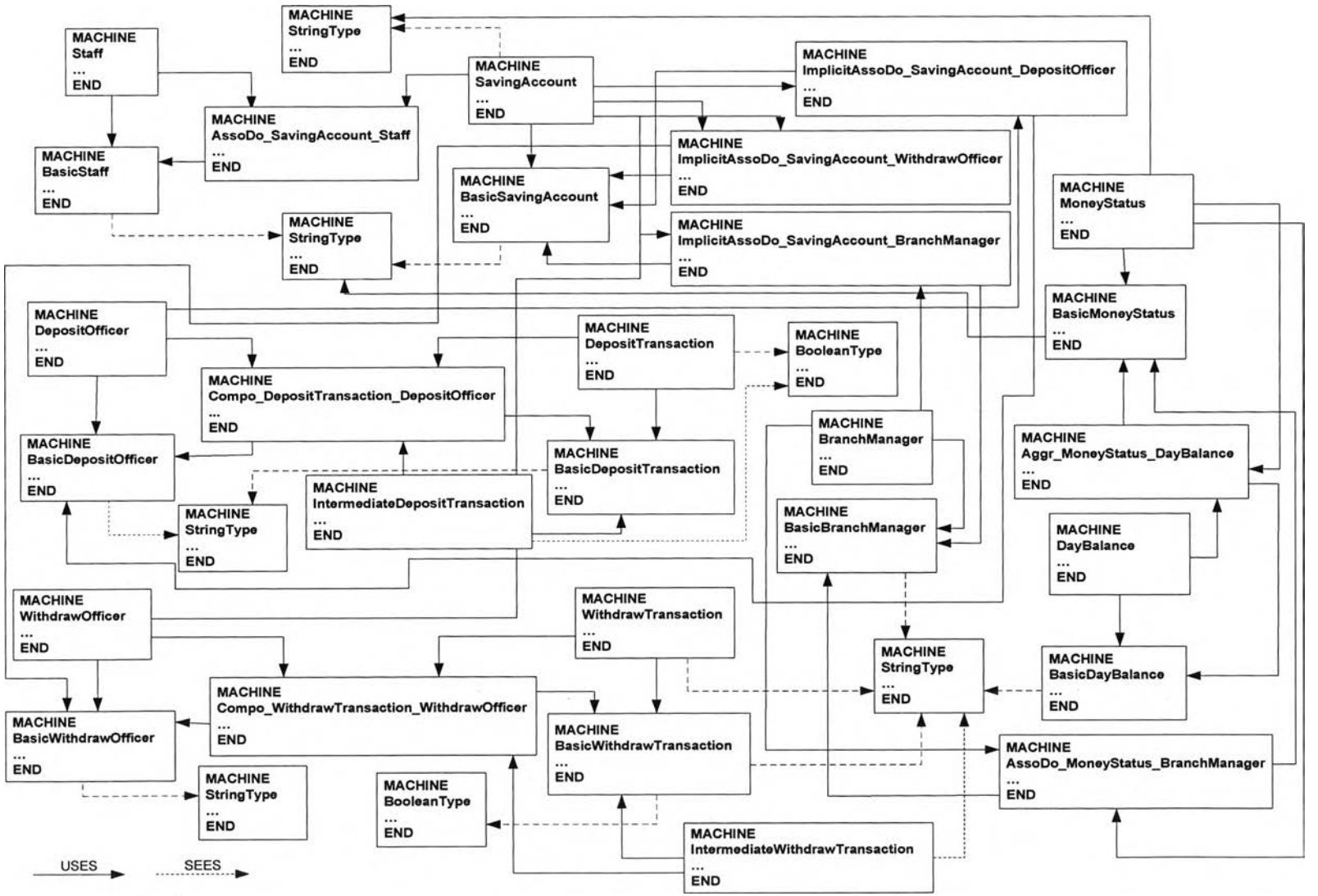
ภาคผนวก จ

สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของระบบการฝากและถอนเงินในธนาคาร



รูปที่ จ-1 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพคลาสในระบบการฝากและถอนเงินในธนาคาร

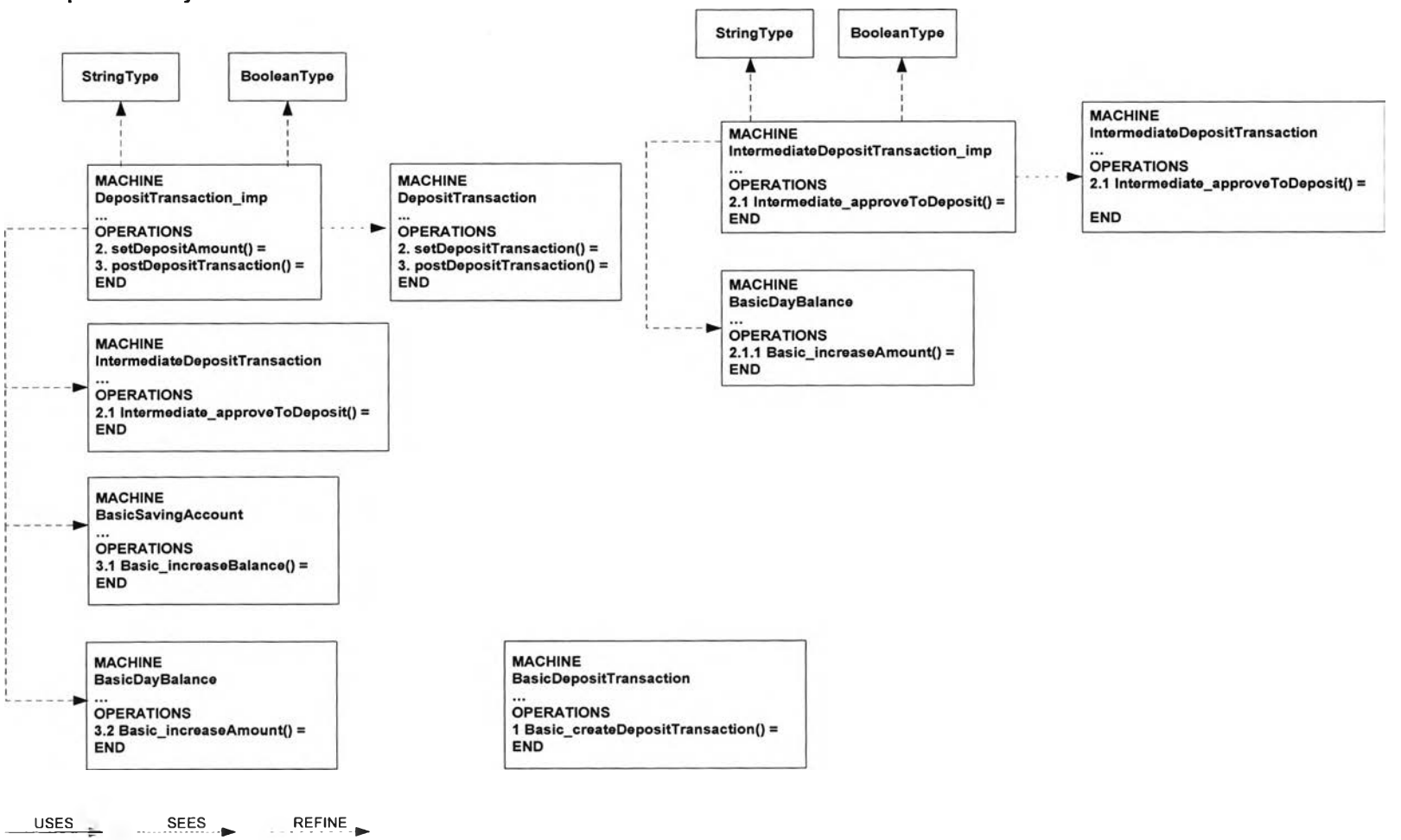
Architecture B Abstract Machine Case Study Deposit and Withdraw System - Class Diagram (Cont)



รูปที่ จ-1 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพคลาสในระบบการฝากและถอนเงินในธนาคาร (ต่อ)

Architecture B Abstract Machine Case Study Deposit and Withdrawal System - Sequence Diagram

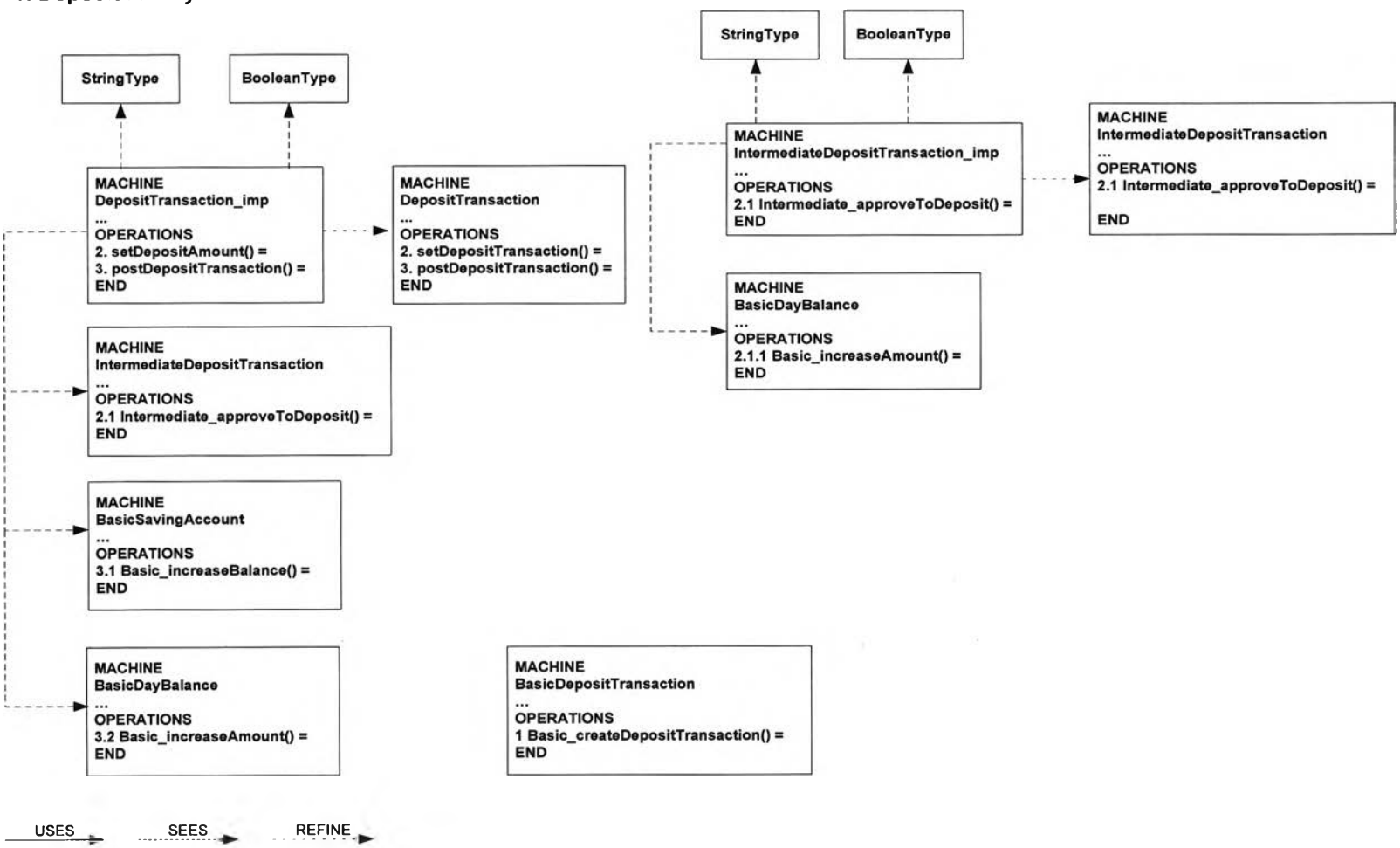
1. Deposit Money



รูปที่ จ-2 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การฝากเงินในระบบการฝากและถอนเงินในธนาคาร

Architecture B Abstract Machine Case Study Deposit and Withdrawal System - Sequence Diagram

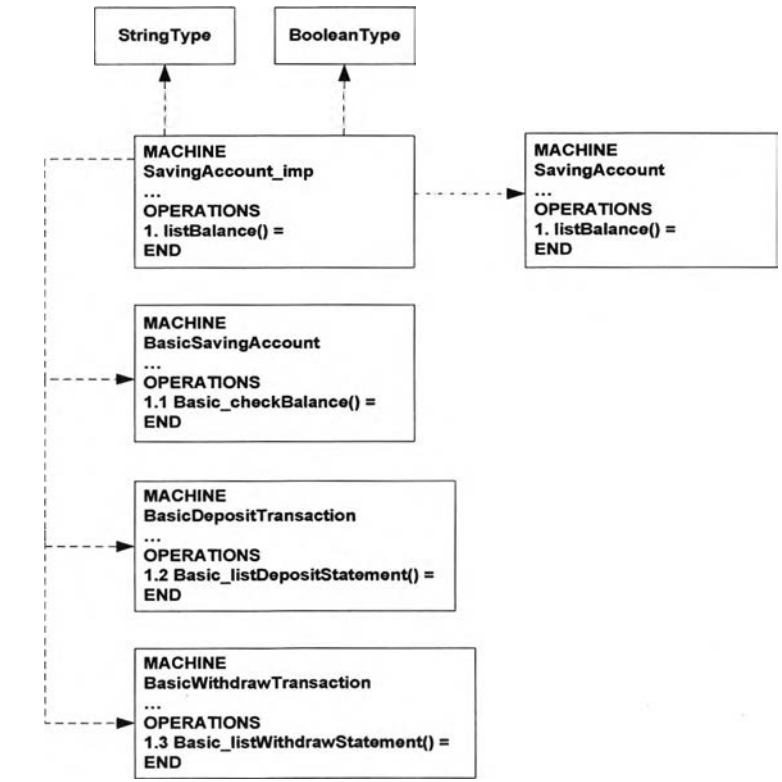
1. Deposit Money



รูปที่ จ-2 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การฝากเงินในระบบการฝากและถอนเงินในธนาคาร

Architecture B Abstract Machine Case Study Deposit and Withdrawal System - Sequence Diagram (Cont)

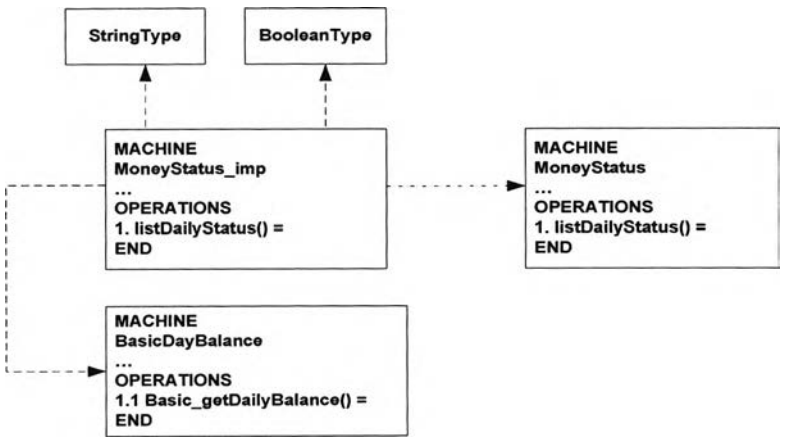
3. List Account Statement



รูปที่ จ-4 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การตรวจสอบยอดเงินคงค้างในระบบการฝากและถอนเงินในธนาคาร

Architecture B Abstract Machine Case Study Deposit and Withdrawal System - Sequence Diagram (Cont)

4. List Money Status



USES SEES REFINE

รูปที่ จ-5 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การตรวจสอบสถานภาพทางการเงินในระบบการฝากและถอนเงินในธนาคาร

ภาคผนวก ฉ
แอ็บสแตร็คแมชชีนบีของระบบระบบการลงทะเบียนนักศึกษา

- ไลบรารีแอ็บสแตร็คแมชชีนบีทั้งหมด มีดังนี้

- ไลบรารีแอ็บสแตร็คแมชชีนบีบูลีน (BooleanType)

```
MACHINE
BooleanType

SETS
BOOLEAN = {TRUE,FALSE}

END
```

- ไลบรารีแอ็บสแตร็คแมชชีนบีสายอักขระ (StringType)

```
MACHINE
StringType

SETS
STRING

VARIABLES
null,
EmptyString

INVARIANT
null : STRING &
EmptyString : STRING

INITIALISATION
null := EmptyString

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสธนาคาร (BAM BasicBank)

```
MACHINE
BasicBank

SEES
StringType

SETS
BASICBANK
```

```
VARIABLES
basicbank,
bank_Branch,
bank_Name,
bank_TotalAmount,
RefUniversityAccountID,
RefStudentAccountID

INVARIANT
basicbank <: BASICBANK &
bank_Branch : STRING &
bank_Name : STRING &
bank_TotalAmount : NAT &
RefUniversityAccountID : STRING &
RefStudentAccountID : STRING

INITIALISATION
basicbank := {} ||
bank_Branch := null ||
bank_Name := null ||
bank_TotalAmount := 0 ||
RefUniversityAccountID := null ||
RefStudentAccountID := null

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสบัญชีเงินฝากของมหาวิทยาลัย (BAM BasicUniversityAccount)

```
MACHINE
BasicUniversityAccount

SEES
StringType

SETS
BASICUNIVERSITYACCOUNT

VARIABLES
basicuniversityaccount,
universityaccount_Name,
universityaccount_Amount

INVARIANT
basicuniversityaccount <: BASIC_UNIVERSITYACCOUNT &
universityaccount_Name : STRING &
universityaccount_Amount : NAT

INITIALISATION
basicuniversityaccount := {} ||
universityaccount_Name :=null ||
universityaccount_Amount := 0
```

```
OPERATIONS
basicuniversityaccount1 <-- Basic_increaseAmount(x1) =
PRE
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
basicuniversityaccount1 := 0
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร์คแมชชีนบีเบซิคคลาสบัญชีเงินฝากของนิสิต (BAM BasicStudentAccount)

```
MACHINE
BasicStudentAccount

SEES
StringType

SETS
BASICSTUDENTACCOUNT

VARIABLES
basicstudentaccount,
studentaccount_Name,
studentaccount_Amount

INVARIANT
basicstudentaccount <: BASICSTUDENTACCOUNT &
studentaccount_Name : STRING &
studentaccount_Amount : NAT

INITIALISATION
basicstudentaccount := {} ||
studentaccount_Name :=null ||
studentaccount_Amount := 0

OPERATIONS
basicstudentaccount1 <-- Basic_decreaseAmount(x1) =
PRE
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
basicstudentaccount1 := 0
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสนิสิต (BAM BasicStudent)

```
MACHINE
BasicStudent

SEES
StringType,
BooleanType

SETS
BASICSTUDENT

VARIABLES
basicstudent,
student_ID,
student_Password,
student_Name,
student_Surname,
student_Faculty,
student_Address,
student_Age

INVARIANT
basicstudent <: BASICSTUDENT &
student_ID : STRING &
student_Password : STRING &
student_Name : STRING &
student_Surname : STRING &
student_Faculty : STRING &
student_Address : STRING &
student_Age : NAT

INITIALISATION
basicstudent := {} ||
student_ID := null ||
student_Password := null ||
student_Name := null ||
student_Surname := null ||
student_Faculty := null ||
student_Address := null ||
student_Age := 0

OPERATIONS
boolean <-- Basic_sendDocumentStudent(a1,a2,a3,a4,x1) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
a4 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END
```

END

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสนิสิตระดับปริญญาตรี (BAM BasicUnderGraduateStudent)

```
MACHINE
BasicUnderGraduateStudent

SEES
StringType,
BooleanType

SETS
BASICUNDERGRADUATESTUDENT

VARIABLES
basicundergraduatestudent,
undergraduatestudent_ID,
undergraduatestudent_Password,
undergraduatestudent_Name,
undergraduatestudent_Surname,
undergraduatestudent_Faculty,
undergraduatestudent_Address,
undergraduatestudent_Age

INVARIANT
basicundergraduatestudent <: BASICSTUDENT &
undergraduatestudent_ID : STRING &
undergraduatestudent_Password : STRING &
undergraduatestudent_Name : STRING &
undergraduatestudent_Surname : STRING &
undergraduatestudent_Faculty : STRING &
undergraduatestudent_Address : STRING &
undergraduatestudent_Age : NAT

INITIALISATION
basicundergraduatestudent := {} ||
undergraduatestudent_ID := null ||
undergraduatestudent_Password := null ||
undergraduatestudent_Name := null ||
undergraduatestudent_Surname := null ||
undergraduatestudent_Faculty := null ||
undergraduatestudent_Address := null ||
undergraduatestudent_Age := 0

OPERATIONS
boolean <-- Basic_sendDocumentStudent(a1,a2,a3,a4,x1) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
a4 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
```

```
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสนิสิตระดับบัณฑิตศึกษา (BAM BasicGraduateStudent)

```
MACHINE
BasicGraduateStudent

SEES
StringType,
BooleanType

SETS
BASICGRADUATESTUDENT

VARIABLES
basicgraduatestudent,
graduatestudent_ID,
graduatestudent_Password,
graduatestudent_Name,
graduatestudent_Surname,
graduatestudent_Faculty,
graduatestudent_Address,
graduatestudent_Age

INVARIANT
basicgraduatestudent <: BASICSTUDENT &
graduatestudent_ID : STRING &
graduatestudent_Password : STRING &
graduatestudent_Name : STRING &
graduatestudent_Surname : STRING &
graduatestudent_Faculty : STRING &
graduatestudent_Address : STRING &
graduatestudent_Age : NAT

INITIALISATION
basicgraduatestudent := {} ||
graduatestudent_ID := null ||
graduatestudent_Password := null ||
graduatestudent_Name := null ||
graduatestudent_Surname := null ||
graduatestudent_Faculty := null ||
graduatestudent_Address := null ||
graduatestudent_Age := 0

OPERATIONS
boolean <-- Basic_sendDocumentStudent(a1,a2,a3,a4,x1) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
a4 : STRING &
x1 : NAT
```

```
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสข้อมูลนิสิต (BAM BasicStudentData)

```
MACHINE
BasicStudentData

SEES
StringType,
BooleanType

SETS
BASICSTUDENTDATA

VARIABLES
basicstudentdata,
studentdata_ID,
studentdata_Name,
studentdata_Surname,
studentdata_Faculty,
studentdata_SubjectApprove,
studentdata_CreditApprove,
studentdata_Grade

INVARIANT
basicstudentdata <: BASICSTUDENTDATA &
studentdata_ID : STRING &
studentdata_Name : STRING &
studentdata_Surname : STRING &
studentdata_Faculty : STRING &
studentdata_SubjectApprove : STRING &
studentdata_CreditApprove : NAT &
studentdata_Grade : NAT

INITIALISATION
basicstudentdata := {} ||
studentdata_ID := null ||
studentdata_Name :=null ||
studentdata_Surname := null ||
studentdata_Faculty := null ||
studentdata_SubjectApprove := null ||
studentdata_CreditApprove := 0 ||
studentdata_Grade := 0

OPERATIONS
boolean <-- Basic_checkStudentRecord(a1,a2,a3,a4,a5,x1,x2) =
PRE
a1 : STRING &
a2 : STRING &
```

```
a3 : STRING &
a4 : STRING &
a5 : STRING &
x1 : NAT &
x2 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

voidbasicstudentdata1 <-- Basic_setStudentdata(a1,a2,a3,a4,a5,x1,x2) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
a4 : STRING &
a5 : STRING &
x1 : NAT &
x2 : NAT
THEN
voidbasicstudentdata1 := null
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสใบเสร็จรับเงิน (BAM BasicReceipt)

```
MACHINE
BasicReceipt

SEES
StringType,
BooleanType

SETS
BASICRECEIPT

VARIABLES
basicreceipt,
receipt_DateRegist,
receipt_SemesterToRegist,
receipt_SubjectOfStudentFromRegist,
receipt_NumberOfSubjectFromRegist,
receipt_NumberOfCreditFromRegist

INVARIANT
basicreceipt <: BASICRECEIPT &
receipt_DateRegist : STRING &
receipt_SemesterToRegist : NAT &
receipt_SubjectOfStudentFromRegist : STRING &
receipt_NumberOfSubjectFromRegist : NAT &
receipt_NumberOfCreditFromRegist : NAT
```

```
INITIALISATION
basicreceipt := {} ||
receipt_DateRegist := null ||
receipt_SemesterToRegist  := 0 ||
receipt_SubjectOfStudentFromRegist := null ||
receipt_NumberOfSubjectFromRegist := 0 ||
receipt_NumberOfCreditFromRegist := 0

OPERATIONS
boolean <-- Basic_confirmToPrintReceipt(a1,a2,x1,x2,x3) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT &
x2 : NAT &
x3 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสนายทะเบียน (BAM BasicRegistra)

```
MACHINE
BasicRegistra

SEES
StringType,
BooleanType

SETS
BASICREGISTRA

VARIABLES
basicregistra,
registra_ID,
registra_Name,
registra_StudentRegistStatus

INVARIANT
basicregistra <: BASICREGISTRA &
registra_ID : STRING &
registra_Name : STRING &
registra_StudentRegistStatus : STRING

INITIALISATION
basicregistra := {} ||
registra_ID := null ||
registra_Name := null ||
registra_StudentRegistStatus := null
```

```
OPERATIONS
voidbasicregistra1 <-- Basic_requestToFirstRegistration =
BEGIN
voidbasicregistra1 := null
END;

boolean <-- Basic_approveFirstRegistration(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสการลงทะเบียน (BAM BasicRegistration)

```
MACHINE
BasicRegistration

SEES
StringType,
BooleanType

SETS
BASICREGISTRATION

VARIABLES
basicregistration,
registration_StudentID,
registration_StudentPassword,
registration_OfferingCourseID,
registration_OfferingCourseName,
registration_OfferingCourseCredit,
registration_OfferingCourseInstructor,
registration_OfferingCourseSection,
registration_OfferingCourseRoom,
registration_TotalAttendedCourseName,
registration_TotalAttendedCourseCredit,
registration_TotalOfferingCourseName,
registration_TotalOfferingCourseCredit

INVARIANT
basicregistration <: BASICREGISTRATION &
registration_StudentID : STRING &
registration_StudentPassword : STRING &
registration_OfferingCourseID : STRING &
registration_OfferingCourseName : STRING &
registration_OfferingCourseCredit : NAT &
registration_OfferingCourseInstructor : STRING &
registration_OfferingCourseSection : STRING &
registration_OfferingCourseRoom : STRING &
```

```
registration_TotalAttendedCourseName : STRING &
registration_TotalAttendedCourseCredit : NAT &
registration_TotalOfferingCourseName : STRING &
registration_TotalOfferingCourseCredit : NAT

INITIALISATION
basicRegistration := {} ||
registration_StudentID := null ||
registration_StudentPassword := null ||
registration_OfferingCourseID := null ||
registration_OfferingCourseName := null ||
registration_OfferingCourseCredit := 0 ||
registration_OfferingCourseInstructor := null ||
registration_OfferingCourseSection := null ||
registration_OfferingCourseRoom := null ||
registration_TotalAttendedCourseName := null ||
registration_TotalAttendedCourseCredit := 0 ||
registration_TotalOfferingCourseName := null ||
registration_TotalOfferingCourseCredit := 0

OPERATIONS
boolean <-- Basic_validateIDAndPassword(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END;

basicregistration1 <-- Basic_obtainNewCourse =
BEGIN
basicregistration1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายวิชาทั้งหมดที่ได้รับการประเมิน (BAM BasicTotalAttendedCourse)

```
MACHINE
BasicTotalAttendedCourse

SEES
StringType

SETS
BASICTOTALATTENDEDCOURSE

VARIABLES
basictotalattendedcourse,
totalattendedcourse_TotalAttendedCourseName,
totalattendedcourse_TotalAttendedCourseCredit,
totalattendedcourse_Grade
```

```
INVARIANT
basictotalattendedcourse <: BASICTOTALATTENDEDCOURSE &
totalattendedcourse_TotalAttendedCourseName : STRING &
totalattendedcourse_TotalAttendedCourseCredit : NAT &
totalattendedcourse_Grade : NAT

INITIALISATION
basictotalattendedcourse := {} ||
totalattendedcourse_TotalAttendedCourseName := null ||
totalattendedcourse_TotalAttendedCourseCredit := 0 ||
totalattendedcourse_Grade := 0

OPERATIONS
basictotalattendedcourse1 <-- Basic_getTotalAttendedCourse =
BEGIN
basictotalattendedcourse1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายวิชาทั้งหมดที่เปิดสอนในภาคการศึกษา

(BAM BasicTotalOfferingCourse)

```
MACHINE
BasicTotalOfferingCourse

SEES
StringType

SETS
BASICTOTALOFFERINGCOURSE

VARIABLES
basictotalofferingcourse,
totalofferingcourse_TotalCourseName,
totalofferingcourse_TotalCourseInformation,
totalofferingcourse_TotalCredit,
totalofferingcourse_TotalFee,
RefOfferingCourseID

INVARIANT
basictotalofferingcourse <: BASICTOTALOFFERINGCOURSE &
totalofferingcourse_TotalCourseName : STRING &
totalofferingcourse_TotalCourseInformation : STRING &
totalofferingcourse_TotalCredit : NAT &
totalofferingcourse_TotalFee : NAT &
RefOfferingCourseID : STRING

INITIALISATION
basictotalofferingcourse := {} ||
totalofferingcourse_TotalCourseName := null ||
totalofferingcourse_TotalCourseInformation := null ||
totalofferingcourse_TotalCredit := 0 ||
```

```
totalofferingcourse_TotalFee := 0 ||
RefOfferingCourseID := null

OPERATIONS
basictotalofferingcourse1 <-- Basic_getTotalOfferingCourse =
BEGIN
basictotalofferingcourse1 := null
/* User can modify output value or post condition of operation here */
END;

basictotalofferingcourse2 <-- Basic_getTotalCredit =
BEGIN
basictotalofferingcourse2 := 0
/* User can modify output value or post condition of operation here */
END;

basictotalofferingcourse3 <-- Basic_getTotalFee =
BEGIN
basictotalofferingcourse3 := 0
/* User can modify output value or post condition of operation here */
END;

basictotalofferingcourse4 <-- Basic_setNewCourse(a1,x1) =
PRE
a1 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
basictotalofferingcourse4 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีเบซิคคลาสรายวิชาที่เปิดสอนในภาคการศึกษา (BAM BasicOfferingCourse)

```
MACHINE
BasicOfferingCourse

SEES
StringType

SETS
BASICOFFERINGCOURSE

VARIABLES
basicofferingcourse,
offeringcourse_Name,
offeringcourse_Section,
offeringcourse_Room,
offeringcourse_Instructor,
offeringcourse_Detail,
offeringcourse_Credit

INVARIANT
```

```
basicofferingcourse <: BASICOFFERINGCOURSE &
offeringcourse_Name : STRING &
offeringcourse_Section : STRING &
offeringcourse_Room : STRING &
offeringcourse_Instructor : STRING &
offeringcourse_Detail : STRING &
offeringcourse_Credit : NAT

INITIALISATION
basicofferingcourse := {} ||
offeringcourse_Name := null ||
offeringcourse_Section := null ||
offeringcourse_Room := null ||
offeringcourse_Instructor := null ||
offeringcourse_Detail := null ||
offeringcourse_Credit := 0

OPERATIONS
basicofferingcourse1 <-- Basic_getOfferingCourseDetail =
BEGIN
basicofferingcourse1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีคลาสธนาคาร (BAM Bank)

```
MACHINE
Bank

SEES
StringType

USES
BasicBank,
Asso_Student_Bank,
Compo_Bank_UniversityAccount,
Compo_Bank_StudentAccount,
ImplicitAsso_UnderGraduateStudent_Bank,
ImplicitAsso_GraduateStudent_Bank

OPERATIONS
bank1 <-- transferCash(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
bank1 := 0
```

```
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสบัญชีเงินฝากของมหาวิทยาลัย (BAM UniversityAccount)

```
MACHINE
UniversityAccount

SEES
StringType,
BooleanType

USES
BasicUniversityAccount,
Compo_Bank_UniversityAccount

OPERATIONS
boolean <-- depositMoney(a1,x1) =
PRE
a1 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสบัญชีเงินฝากของนิสิต (BAM StudentAccount)

```
MACHINE
StudentAccount

SEES
StringType,
BooleanType

USES
BasicStudentAccount,
Compo_Bank_StudentAccount

OPERATIONS
boolean <-- withdrawMoney(a1,x1) =
PRE
a1 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END
```

END

- แอ็บสแตร็คแมชชีนบีคลาสนิสิต (BAM Student)

MACHINE
Student

USES
BasicStudent,
Asso_Registration_Student,
Asso_Student_Registra,
Asso_Student_Bank

END

- แอ็บสแตร็คแมชชีนบีคลาสนิสิตระดับปริญญาตรี (BAM UnderGraduateStudent)

MACHINE
UnderGraduateStudent

USES
BasicUnderGraduateStudent,
ImplicitAsso_Registration_UnderGraduateStudent,
ImplicitAsso_UnderGraduateStudent_Registra,
ImplicitAsso_UnderGraduateStudent_Bank

END

- แอ็บสแตร็คแมชชีนบีคลาสนิสิตระดับบัณฑิตศึกษา (BAM GraduateStudent)

MACHINE
GraduateStudent

USES
BasicGraduateStudent,
ImplicitAsso_Registration_GraduateStudent,
ImplicitAsso_GraduateStudent_Registra,
ImplicitAsso_GraduateStudent_Bank

END

- แอ็บสแตร็คแมชชีนบีคลาสข้อมูลนิสิต (BAM StudentData)

MACHINE
StudentData

USES
BasicStudentData,
AssoRetrieve_Registra_StudentData

END

- แอ็บสแตร็คแมชชีนบีคลาสใบเสร็จรับเงิน (BAM Receipt)

```
MACHINE
Receipt

USES
BasicReceipt,
Asso_Receipt_Registration

END
```

- แอ็บสแตร็คแมชชีนบีคลาสนายทะเบียน (BAM Registra)

```
MACHINE
Registra

USES
BasicRegistra,
Asso_Student_Registra,
AssoRetrieve_Registra_StudentData,
ImplicitAsso_UnderGraduateStudent_Registra,
ImplicitAsso_GraduateStudent_Registra

END
```

- แอ็บสแตร็คแมชชีนบีคลาสการลงทะเบียน (BAM Registration)

```
MACHINE
Registration

SEES
StringType,
BooleanType

USES
BasicRegistration,
Asso_Registration_Student,
Asso_TotalOfferingCourse_Registration,
Asso_TotalAttendedCourse_Registration,
Asso_Receipt_Registration,
ImplicitAsso_Registration_UnderGraduateStudent,
ImplicitAsso_Registration_GraduateStudent

OPERATIONS
voidregistration1 <-- loginToRegistration(a1,a2) =
PRE
a1 : STRING &
a2 : STRING
THEN
voidregistration1 := null
END;

registration1 <-- getAttendedCourse =
BEGIN
registration1 := null
```

```
/* User can modify output value or post condition of operation here */
END;

registration2 <-- getOfferingCourse =
BEGIN
registration2 := null
/* User can modify output value or post condition of operation here */
END;

voidregistration2 <-- setSelectedCourse(a1,a2,a3,a4,a5,x1) =
PRE
a1 : STRING &
a2 : STRING &
a3 : STRING &
a4 : STRING &
a5 : STRING &
x1 : NAT
THEN
voidregistration2 := null
END;

boolean <-- confirmToRegist(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
boolean := FALSE
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสรายวิชาทั้งหมดที่ได้รับการประเมิน (BAM TotalAttendedCourse)

```
MACHINE
TotalAttendedCourse

USES
BasicTotalAttendedCourse,
Asso_TotalAttendedCourse_Registration

END
```

- แอ็บสแตร็คแมชชีนบีคลาสรายวิชาทั้งหมดที่เปิดสอนในภาคการศึกษา (BAM TotalOfferingCourse)

```
MACHINE
TotalOfferingCourse

SEES
StringType
```

```
USES
BasicTotalOfferingCourse,
Asso_TotalOfferingCourse_Registration,
Compo_TotalOfferingCourse_OfferingCourse

OPERATIONS
totalofferingcourse1 <-- getTotalOfferingCourseInformation =
BEGIN
totalofferingcourse1 := null
/* User can modify output value or post condition of operation here */
END

END
```

- แอ็บสแตร็คแมชชีนบีคลาสรายวิชาที่เปิดสอนในภาคการศึกษา (BAM OfferingCourse)

```
MACHINE
OfferingCourse

USES
BasicOfferingCourse,
Compo_TotalOfferingCourse_OfferingCourse

END
```

- แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีอินเทอร์มิเดียทคลาสการลงทะเบียน (BAM IntermediateRegistration)

```
MACHINE
IntermediateRegistration

SEES
StringType

USES
BasicRegistration,
Asso_Registration_Student,
Asso_TotalOfferingCourse_Registration,
Asso_TotalAttendedCourse_Registration,
Asso_Receipt_Registration,
ImplicitAsso_Registration_UnderGraduateStudent,
ImplicitAsso_Registration_GraduateStudent
OPERATIONS
voidintermediateregistration1 <-- Intermediate_setCourse(a1,a2,x1) =
PRE
a1 : STRING &
a2 : STRING &
x1 : NAT
THEN
/* User can insert or not insert condition's operation by using IF here */
voidintermediateregistration1 := null
/* User can modify output value or post condition of operation here */
```

END

END

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์ทั้งหมด มีดังนี้

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสใบเสร็จรับเงินกับคลาสการลงทะเบียน (Asso_Receipt_Registration)

```
MACHINE
Asso_Receipt_Registration

USES
BasicReceipt,
BasicRegistration

VARIABLES
asso_receipt_registration

INVARIANT
asso_receipt_registration <: BASICRECEIPT * BASICREGISTRATION &
dom(asso_receipt_registration) = basicreceipt &
ran(asso_receipt_registration) = basicregistration &
!(xx,yy).(((xx : dom(asso_receipt_registration)) &
            (yy : ran(asso_receipt_registration)))
        => card((asso_receipt_registration)[{xx}]) = 1 &
           card((asso_receipt_registration)~[{yy}]) >= 1)

INITIALISATION
asso_receipt_registration := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสนิสิตกับคลาสธนาคาร (Asso_Student_Bank)

```
MACHINE
Asso_Student_Bank

USES
BasicStudent,
BasicBank

VARIABLES
asso_student_bank

INVARIANT
asso_student_bank <: BASICSTUDENT * BASICBANK &
dom(asso_student_bank) = basicstudent &
ran(asso_student_bank) = basicbank &
!(xx,yy).(((xx : dom(asso_student_bank)) &
```

```
        (yy : ran(asso_student_bank)))
    => card((asso_student_bank)[{xx}]) >= 1 &
        card((asso_student_bank)~[{yy}]) >= 1)

INITIALISATION
asso_student_bank := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสรายวิชาทั้งหมดที่ได้รับการประเมินกับคลาสการลงทะเบียน (Asso_TotalAttendedCourse_Registration)

```
MACHINE
Asso_TotalAttendedCourse_Registration

USES
BasicTotalAttendedCourse,
BasicRegistration


VARIABLES
asso_totalattendedcourse_registration

INVARIANT
asso_totalattendedcourse_registration <:  BASICTOTALATTENDEDCOURSE * BASICREGISTRATION
&
dom(asso_totalattendedcourse_registration) = basictotalattendedcourse &
ran(asso_totalattendedcourse_registration) = basicregistration &
!(xx,yy).(((xx : dom(asso_totalattendedcourse_registration)) &
        (yy : ran(asso_totalattendedcourse_registration)))
    => card((asso_totalattendedcourse_registration)[{xx}]) = 1 &
        card((asso_totalattendedcourse_registration)~[{yy}]) >= 1)

INITIALISATION
asso_totalattendedcourse_registration := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสรายวิชาทั้งหมดที่เปิดสอนในภาคการศึกษากับคลาสการลงทะเบียน (Asso_TotalOfferingCourse_Registration)

```
MACHINE
Asso_TotalOfferingCourse_Registration

USES
BasicOfferingCourse,
BasicRegistration

VARIABLES
asso_offeringcourse_registration

INVARIANT
asso_offeringcourse_registration <: BASICOFFERINGCOURSE * BASICREGISTRATION &
```

```
dom(asso_offeringcourse_registration) = basicofferingcourse &
ran(asso_offeringcourse_registration) = basicregistration &
!(xx,yy).(((xx : dom(asso_offeringcourse_registration)) &
            (yy : ran(asso_offeringcourse_registration)))
        => card((asso_offeringcourse_registration)[{xx}]) = 1 &
            card((asso_offeringcourse_registration)~[{yy}]) >= 1)

INITIALISATION
asso_offeringcourse_registration := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันติดต่อระหว่างคลาสนิสิตกับคลาสนายทะเบียน

  (Asso_Student_Registra)

```
MACHINE
Asso_Student_Registra

USES
BasicStudent,
BasicRegistra

VARIABLES
asso_student_registra

INVARIANT
asso_student_registra <: BASICSTUDENT * BASICREGISTRA &
dom(asso_student_registra) = basicstudent &
ran(asso_student_registra) = basicregistra &
!(xx,yy).(((xx : dom(asso_student_registra)) &
            (yy : ran(asso_student_registra)))
        => card((asso_student_registra)[{xx}]) = 1 &
            card((asso_student_registra)~[{yy}]) >= 0)

INITIALISATION
asso_student_registra := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันระหว่างคลาสการลงทะเบียนกับคลาสนิสิต

  (Asso_Registration_Student)

```
MACHINE
Asso_Registration_Student

USES
BasicRegistration,
BasicStudent

VARIABLES
asso_registration_student
```

```
INVARIANT
asso_registration_student <: BASICREGISTRATION * BASICSTUDENT  &
dom(asso_registration_student) = basicregistration &
ran(asso_registration_student) = basicstudent &
!(xx,yy).(((xx : dom(asso_registration_student)) &
            (yy : ran(asso_registration_student)))
        => card((asso_registration_student)[{xx}]) >= 0 &
           card((asso_registration_student)~[{yy}]) >= 1)

INITIALISATION
asso_registration_student:= {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันเรียกดูระหว่างคลาสนายทะเบียนกับคลาสข้อมูลนิสิต (AssoRetrieve_Registra_StudentData)

```
MACHINE
AssoRetrieve_Registra_StudentData

USES
BasicRegistra,
BasicStudentData

VARIABLES
assoretrieve_registra_studentdata

INVARIANT
assoretrieve_registra_studentdata <: BASICREGISTRA * BASICSTUDENTDATA &
dom(assoretrieve_registra_studentdata) = basicregistra &
ran(assoretrieve_registra_studentdata) = basicstudentdata &
!(xx,yy).(((xx : dom(assoretrieve_registra_studentdata)) &
            (yy : ran(assoretrieve_registra_studentdata)))
        => card((assoretrieve_registra_studentdata)[{xx}]) >= 1 &
           card((assoretrieve_registra_studentdata)~[{yy}]) = 1)

INITIALISATION
assoretrieve_registra_studentdata := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอกริเกชันระหว่างคลาสรายวิชาทั้งหมดที่เปิดสอนในภาคการศึกษากับคลาสรายวิชาที่เปิดสอนในภาคการศึกษา (Aggr_ TotalOfferingCourse_OfferingCourse)

```
MACHINE
Aggr_TotalOfferingCourse_OfferingCourse

USES
BasicTotalOfferingCourse,
BasicOfferingCourse

VARIABLES
aggr_totalofferingcourse_offeringcourse
```

```
INVARIANT
aggr_totalofferingcourse_offeringcourse <: BASICTOTALOFFERINGCOURSE *
BASICOFFERINGCOURSE &
dom(aggr_totalofferingcourse_offeringcourse) = basictotalofferingcourse &
ran(aggr_totalofferingcourse_offeringcourse) = basicofferingcourse &
!(xx,yy).(((xx : dom(aggr_totalofferingcourse_offeringcourse)) &
            (yy : ran(aggr_totalofferingcourse_offeringcourse)))
        => card((aggr_totalofferingcourse_offeringcourse)[{xx}]) >= 1 &
           card((aggr_totalofferingcourse_offeringcourse)~[{yy}]) = 1)

INITIALISATION
aggr_totalofferingcourse_offeringcourse := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสธนาคารกับคลาสบัญชีเงินฝากของนิสิต (Compo_Bank_StudentAccount)

```
MACHINE
Compo_Bank_StudentAccount

USES
BasicBank,
BasicStudentAccount

VARIABLES
compo_bank_studentaccount

INVARIANT
compo_bank_studentaccount <: BASICBANK * BASICSTUDENTACCOUNT &
dom(compo_bank_studentaccount) = basicbank &
ran(compo_bank_studentaccount) = basicstudentaccount &
!(xx,yy).(((xx : dom(compo_bank_studentaccount)) &
            (yy : ran(compo_bank_studentaccount)))
          => card((compo_bank_studentaccount)[{xx}]) >= 1 &
             card((compo_bank_studentaccount)~[{yy}]) = 1)

INITIALISATION
compo_bank_studentaccount := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์คอมโพสิชันระหว่างคลาสธนาคารกับคลาสบัญชีเงินฝากของมหาวิทยาลัย (Compo_Bank_UniversityAccount)

```
MACHINE
Compo_Bank_UniversityAccount

USES
BasicBank,
BasicUniversityAccount
```

```
VARIABLES
compo_bank_universityaccount

INVARIANT
compo_bank_universityaccount <: BASICBANK * BASICUNIVERSITYACCOUNT &
dom(compo_bank_universityaccount) = basicbank &
ran(compo_bank_universityaccount) = basicuniversityaccount &
!(xx,yy).(((xx : dom(compo_bank_universityaccount)) &
            (yy : ran(compo_bank_universityaccount)))
         => card((compo_bank_universityaccount)[{xx}]) >= 1 &
             card((compo_bank_universityaccount)~[{yy}]) = 1)

INITIALISATION
compo_bank_universityaccount := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาสนิสิตระดับปริญญาตรีกับ คลาสธนาคาร (ImplicitAsso_UnderGraduateStudent_Bank)

```
MACHINE
ImplicitAsso_UnderGraduateStudent_Bank

USES
BasicUnderGraduateStudent,
BasicBank

VARIABLES
implicitasso_undergraduatestudent_bank

INVARIANT
implicitasso_undergraduatestudent_bank<: BASICUNDERGRADUATESTUDENT * BASICBANK &
dom(implicitasso_undergraduatestudent_bank) = basicundergraduatestudent &
ran(implicitasso_undergraduatestudent_bank) = basicbank &
!(xx,yy).(((xx : dom(implicitasso_undergraduatestudent_bank)) &
            (yy : ran(implicitasso_undergraduatestudent_bank)))
         => card((implicitasso_undergraduatestudent_bank)[{xx}]) >= 1 &
             card((implicitasso_undergraduatestudent_bank)~[{yy}]) >= 1)

INITIALISATION
implicitasso_undergraduatestudent_bank:= {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาสนิสิตระดับบัณฑิตศึกษากับ คลาสธนาคาร (ImplicitAsso_GraduateStudent_Bank)

```
MACHINE
ImplicitAsso_GraduateStudent_Bank

USES
BasicGraduateStudent,
BasicBank
```

```
VARIABLES
implicitasso_graduatestudent_bank

INVARIANT
implicitasso_graduatestudent_bank<: BASICGRADUATESTUDENT * BASICBANK &
dom(implicitasso_graduatestudent_bank) = basicgraduatestudent &
ran(implicitasso_graduatestudent_bank) = basicbank &
!(xx,yy).(((xx : dom(implicitasso_graduatestudent_bank)) &
            (yy : ran(implicitasso_graduatestudent_bank)))
        => card((implicitasso_graduatestudent_bank)[{xx}]) >= 1 &
            card((implicitasso_graduatestudent_bank)~[{yy}]) >= 1)

INITIALISATION
implicitasso_graduatestudent_bank:= {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาสนิสิตระดับปริญญาตรีกับคลาสนายทะเบียน (ImplicitAsso_UnderGraduateStudent_Registra)

```
MACHINE
ImplicitAsso_UnderGraduateStudent_Registra

USES
BasicUnderGraduateStudent,
BasicRegistra

VARIABLES
implicitasso_undergraduatestudent_registra

INVARIANT
implicitasso_undergraduatestudent_registra <: BASICUNDERGRADUATESTUDENT * BASICREGISTRA
&
dom(implicitasso_undergraduatestudent_registra) = basicundergraduatestudent &
ran(implicitasso_undergraduatestudent_registra) = basicregistra &
!(xx,yy).(((xx : dom(implicitasso_undergraduatestudent_registra)) &
            (yy : ran(implicitasso_undergraduatestudent_registra)))
        => card((implicitasso_undergraduatestudent_registra)[{xx}]) = 1 &
            card((implicitasso_undergraduatestudent_registra)~[{yy}]) >= 1)

INITIALISATION
implicitasso_undergraduatestudent_registra := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างซับคลาสนิสิตระดับบัณฑิตศึกษากับคลาสนายทะเบียน (ImplicitAsso_GraduateStudent_Registra)

```
MACHINE
ImplicitAsso_GraduateStudent_Registra

USES
BasicGraduateStudent,
```

```
BasicRegistra

VARIABLES
implicitasso_graduatestudent_registra

INVARIANT
implicitasso_graduatestudent_registra <: BASICGRADUATESTUDENT * BASICREGISTRA &
dom(implicitasso_graduatestudent_registra) = basicundergraduatestudent &
ran(implicitasso_graduatestudent_registra) = basicregistra &
!(xx,yy).(((xx : dom(implicitasso_graduatestudent_registra)) &
            (yy : ran(implicitasso_graduatestudent_registra)))
        => card((implicitasso_graduatestudent_registra)[{xx}]) = 1 &
            card((implicitasso_graduatestudent_registra)~[{yy}]) >= 1)

INITIALISATION
implicitasso_undergraduatestudent_registra := {}

END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างคลาสการลงทะเบียนกับซับคลาสนิสิตระดับปริญญาตรี (ImplicitAsso_Registration_UnderGraduateStudent)

```
MACHINE
ImplicitAsso_Registration_UnderGraduateStudent

USES
BasicRegistration,
BasicUnderGraduateStudent


VARIABLES
implicitasso_registration_undergraduatestudent

INVARIANT
implicitasso_registration_undergraduatestudent <: BASICREGISTRATION *
BASICUNDERGRADUATESTUDENT &
dom(implicitasso_registration_undergraduatestudent) = basicregistration &
ran(implicitasso_registration_undergraduatestudent) = basicundergraduatestudent &
!(xx,yy).(((xx : dom(implicitasso_registration_undergraduatestudent)) &
            (yy : ran(implicitasso_registration_undergraduatestudent)))
        => card((implicitasso_registration_undergraduatestudent)[{xx}]) = 1 &
            card((implicitasso_registration_undergraduatestudent)~[{yy}]) >= 1)

INITIALISATION
implicitasso_registration_undergraduatestudent := {}
END
```

- แอ็บสแตร็คแมชชีนบีของความสัมพันธ์แอสโซซิเอชันโดยปริยายระหว่างคลาสการลงทะเบียนกับซับคลาสนิสิตระดับบัณฑิตศึกษา (ImplicitAsso_Registration_GraduateStudent)

```
MACHINE
ImplicitAsso_Registration_GraduateStudent

USES
BasicRegistration,
BasicGraduateStudent

VARIABLES
implicitasso_registration_graduatestudent

INVARIANT
implicitasso_registration_graduatestudent <: BASICREGISTRATION * BASICGRADUATESTUDENT &
dom(implicitasso_registration_graduatestudent) = basicregistration &
ran(implicitasso_registration_graduatestudent) = basicgraduatestudent &
!(xx,yy).(((xx : dom(implicitasso_registration_graduatestudent)) &
            (yy : ran(implicitasso_registration_graduatestudent)))
         => card((implicitasso_registration_graduatestudent)[{xx}]) = 1 &
             card((implicitasso_registration_graduatestudent)~[{yy}]) >= 1)

INITIALISATION
implicitasso_registration_graduatestudent := {}

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีทั้งหมด มีดังนี้

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสธนาคาร (Bank_imp)

```
IMPLEMENTATION
Bank_imp

REFINES
Bank

SEES
StringType,
BooleanType,
StudentAccount,
UniversityAccount
OPERATIONS
bank1 <-- transferCash(a1,a2,x1) =
VAR
boolean
IN
boolean <-- withdrawMoney(a1,x1);
boolean <-- depositMoney(a1,x1)
END
END
```

- อิมพลีเมนต์เทชันแอ็บสแตร็คแมชชีนบีคลาสการลงทะเบียน (Registration_imp)

```
IMPLEMENTATION
Registration_imp

REFINES
Registration

SEES
StringType,
BooleanType,
BasicRegistration,
BasicTotalAttendedCourse,
BasicTotalOfferingCourse,
IntermediateRegistration,
TotalOfferingCourse

OPERATIONS
voidregistration1 <-- loginToRegistration(a1,a2) =
VAR
boolean
IN
boolean <-- Basic_validateIDAndPassword(a1,a2)
END;

registration1 <-- getAttendedCourse =
VAR
basictotalattendedcourse1
IN
basictotalattendedcourse1 <-- Basic_getTotalAttendedCourse
END;

registration2 <-- getOfferingCourse =
VAR
basictotalofferingcourse1
IN
basictotalofferingcourse1 <-- Basic_getTotalOfferingCourse
END;

voidregistration2 <-- setSelectedCourse(a1,a2,a3,a4,a5,x1) =
VAR
voidintermediateregistration1
IN
voidintermediateregistration1 <-- Intermediate_setCourse(a1,a2,x1)
END;

boolean <-- confirmToRegist(a1,a2,x1) =
VAR
basictotalofferingcourse4,
totalofferingcourse1
IN
basictotalofferingcourse4 <-- Basic_setNewCourse(a1,x1);
totalofferingcourse1 <-- getTotalOfferingCourseInformation
END
```

```
END
```

- อิมพลีเมนต์เทชันแอ็บสแตร์คแมชชีนบีอินเทอร์มิเดียทคลาสการลงทะเบียน (IntermediateRegistration_imp)

```
IMPLEMENTATION
IntermediateRegistration_imp

REFINES
IntermediateRegistration

SEES
StringType,
BasicTotalOfferingCourse

OPERATIONS
voidintermediateregistration1 <-- Intermediate_setCourse(a1,a2,x1) =
VAR
basictotalofferingcourse2,
basictotalofferingcourse3
IN
basictotalofferingcourse2 <-- Basic_getTotalCredit;
basictotalofferingcourse3 <-- Basic_getTotalFee
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร์คแมชชีนบีคลาสบัญชีเงินฝากของนิสิต (StudentAccount_imp)

```
IMPLEMENTATION
StudentAccount_imp

REFINES
StudentAccount

SEES
StringType,
BooleanType,
BasicStudentAccount

OPERATIONS
boolean <-- withdrawMoney(a1,x1) =
VAR
basicstudentaccount1
IN
        IF
        boolean = TRUE
        THEN
        basicstudentaccount1 <-- Basic_decreaseAmount(x1)
        ELSE
        skip
        END
END
```

```
END
```

- อิมพลีเมนต์เทชันแอ็บสแตร์คแมชชีนบีคลาสบัญชีเงินฝากของมหาวิทยาลัย (UniversityAccount_imp)

```
IMPLEMENTATION
UniversityAccount_imp

REFINES
UniversityAccount

SEES
StringType,
BooleanType,
BasicUniversityAccount

OPERATIONS
boolean <-- depositMoney(a1,x1) =
VAR
basicuniversityaccount1
IN
        IF
        boolean = TRUE
        THEN
        basicuniversityaccount1 <-- Basic_increaseAmount(x1)
        ELSE
        skip
        END
END

END
```

- อิมพลีเมนต์เทชันแอ็บสแตร์คแมชชีนบีคลาสรายวิชาทั้งหมดที่เปิดสอนในภาคการศึกษา (TotalOfferingCourse_imp)

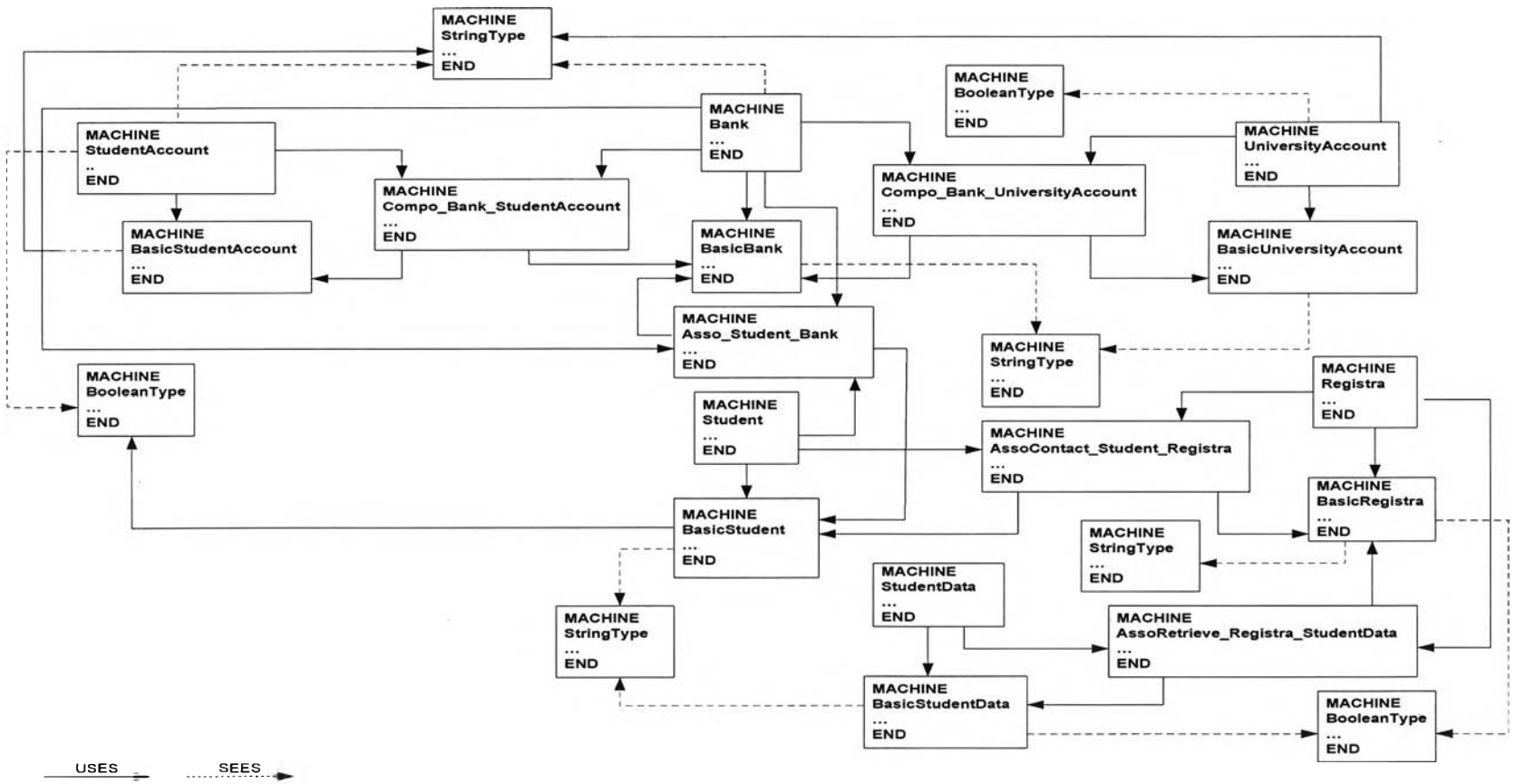
```
IMPLEMENTATION
TotalOfferingCourse_imp

REFINES
TotalOfferingCourse

SEES
StringType,
BasicOfferingCourse,
BasicRegistration

OPERATIONS
totalofferingcourse1 <-- getTotalOfferingCourseInformation =
VAR
basicofferingcourse1,
basicregistration1
IN
basicofferingcourse1 <-- Basic_getOfferingCourseDetail;
basicregistration1 <-- Basic_obtainNewCourse
```

ต้นฉบับ หน้าขาดหาย

ภาคผนวก ช

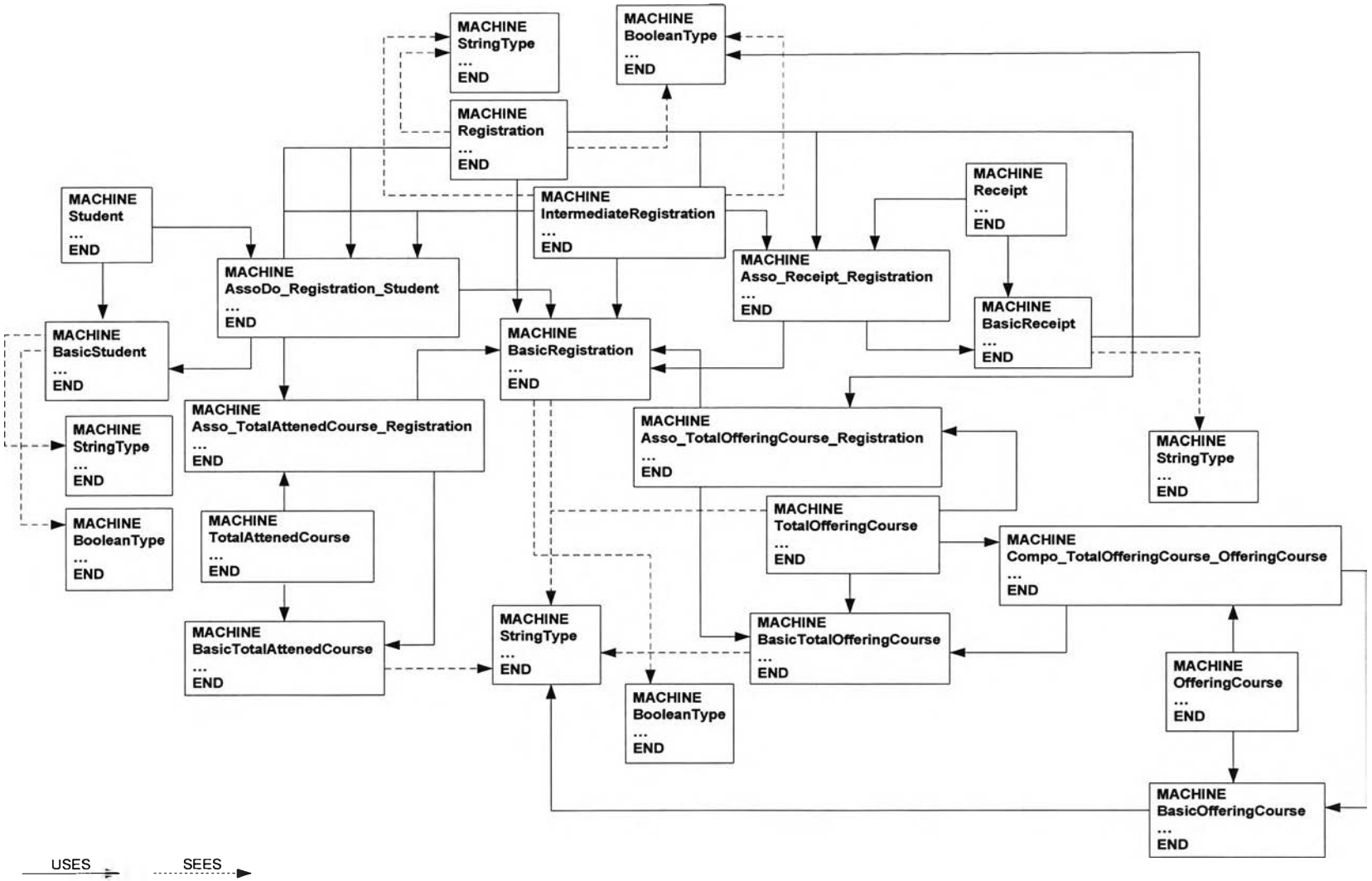
สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของระบบระบบการลงทะเบียนนักศึกษา

Architecture B Abstract Machine Case Study Register System - Class Diagram

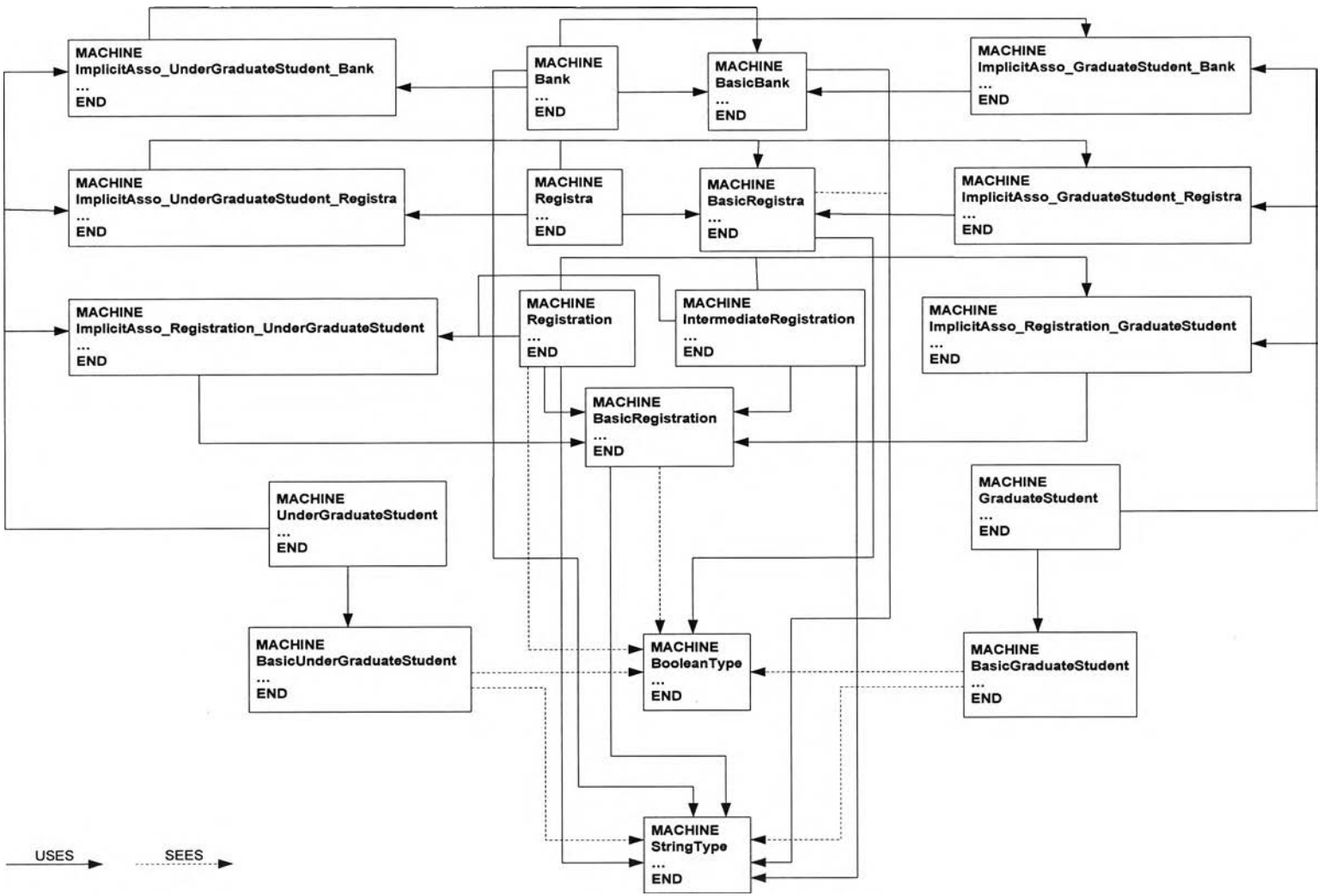


รูปที่ ช-1 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพคลาสในระบบการลงทะเบียนของนิสิต

Architecture B Abstract Machine Case Study Register System - Class Diagram (Cont)



รูปที่ ช-1 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพคลาสในระบบการลงทะเบียนของนิสิต (ต่อ)



รูปที่ ซ-1 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพคลาสในระบบการลงทะเบียนของนิสิต (ต่อ)

Architecture B Abstract Machine Case Study Registration System - Sequence Diagram

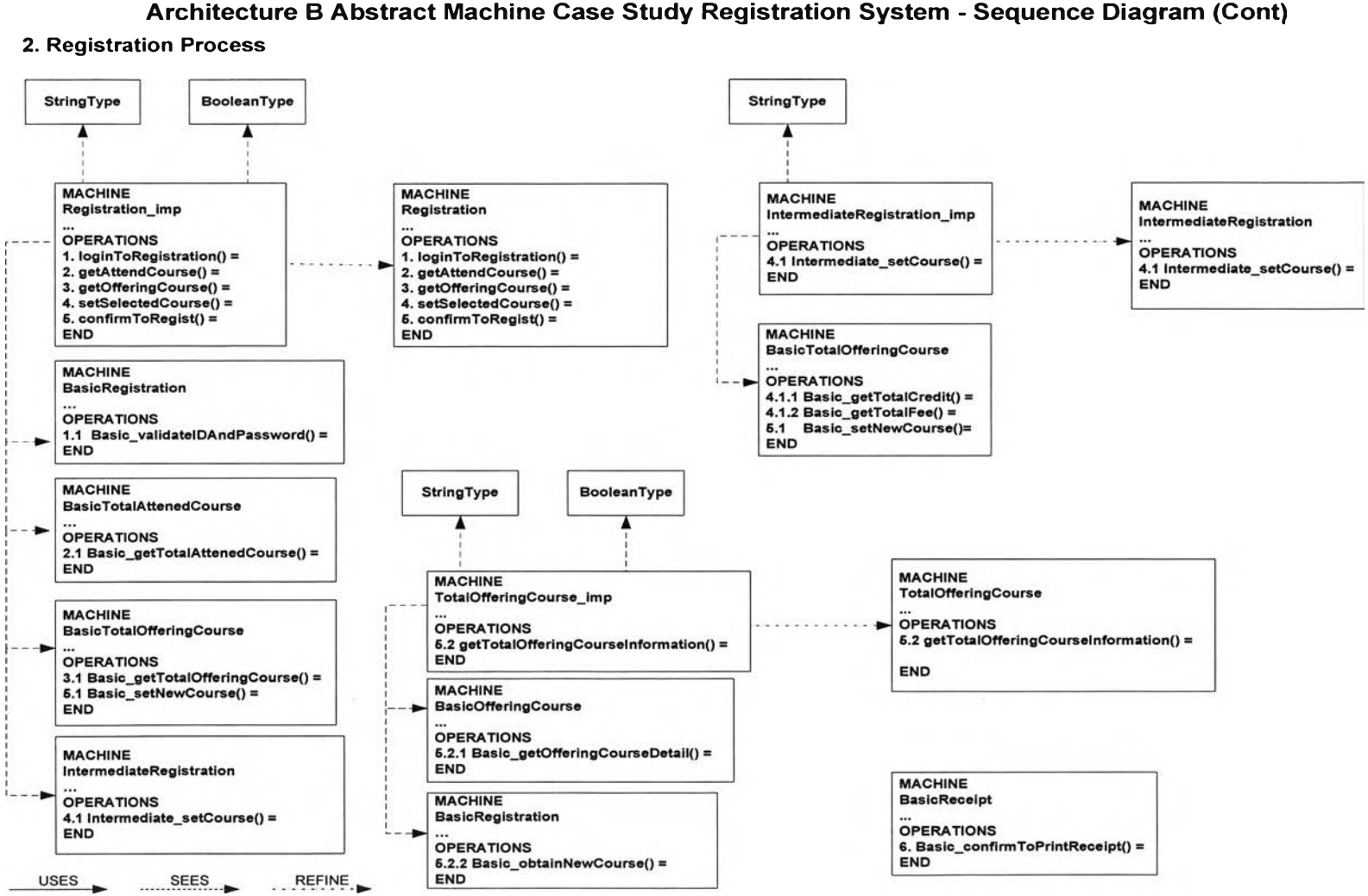
1. First Registration

```
MACHINE
BasicRegistra
...
OPERATIONS
1. Basic_requestToFirstRegistration() =
5. Basic_approveFirstRegistration() =
END
```

```
MACHINE
BasicStudent
...
OPERATIONS
2. Basic_sentDocumentStudent() =
END
```

```
MACHINE
BasicStudentData
...
OPERATIONS
3. Basic_checkStudentRecord() =
4. Basic_setStudentData() =
END
```

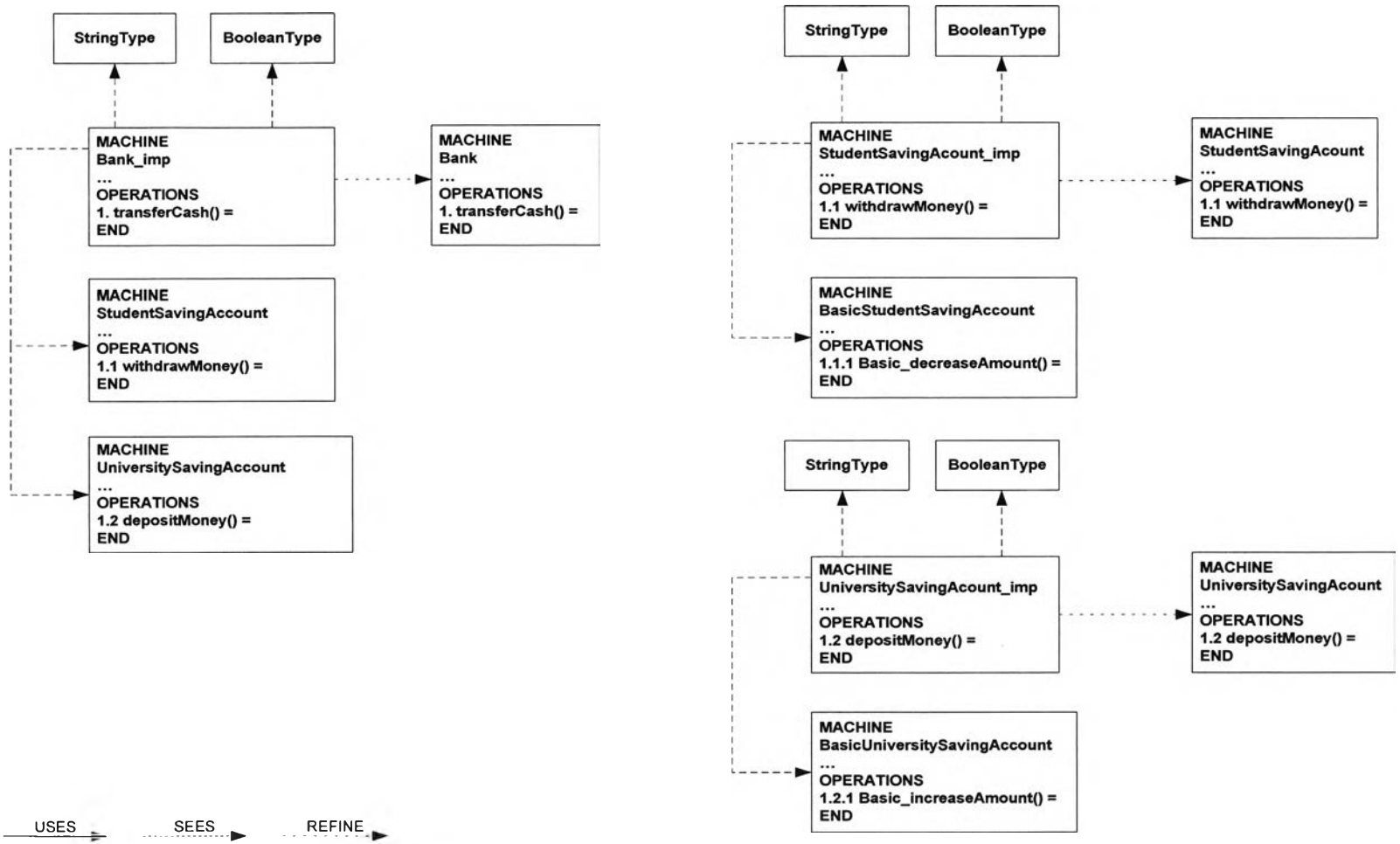
รูปที่ ซ-2 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การลงทะเบียนแรกเข้าในระบบการลงทะเบียนของนิสิต



รูปที่ ช-3 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์ขั้นตอนการลงทะเบียนในระบบการลงทะเบียนของนิสิต

Architecture B Abstract Machine Case Study Register System - Sequence Diagram (Cont)

3. Cash Money



รูปที่ ช-4 สถาปัตยกรรมแอ็บสแตร็คแมชชีนบีของแผนภาพซีเควนซ์ของเหตุการณ์การชำระเงินลงทะเบียนผ่านทางธนาคารในระบบการลงทะเบียนของนิสิต

ประวัติผู้เขียนวิทยานิพนธ์

นายไวทยะ ศรีจรุณรัตน์ เกิดที่กรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตรบัณฑิต คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และได้เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2546