รายการอ้างอิง

[1] Roger S. Pressman. Software Engineering: a practitioner's approach. Fifth Edition, (n.p):McGraw-Hill, 2001: pp. 673-698.

[2] Martin D. Fraser ,Kuldeep Kumar, and Vijay K. Vaishanavi. Informal and Formal Requirements Specification Languages: Bridging the Gap. IEEE TSE Vol17 No. 5. MAY 1991: 454-466.

[3] J. Bowen, and M. Hunchey. Seven Myths of Formal Methods. IEEE Software. Jul. 1995:34-41.

[4] Wiwat Vatanawood. Formal Specification Synthesis Using Requirements Particle Networks. Doctoral's Thesis, Department of Computer Engineering, Graduate School, Chulalongkorn University, 2002.

[5] W. Vatanawood, W. Kamchornwate,and W. Rivepiboon. Requirement Particle Networks: An Approach to Formal Software Functional Requirements Modeling. Preceeding of the IASTED International Conference on Applied Simulation and Modelling, Canada, 2000.

[6] Rãzvan Diaconescu ,and Kokichi Futatsugi. CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification. Volume 6 of AMAST series in Computing. World Scientific. 1998.

[7] Rãzvan Diaconescu ,Kokichi Futatsugi ,and Shusaku Iida. Component-based Algrbraic Specification and Verification in CafeOBJ. OBJ/CafeOBJ/Maude at FM'99. Springer-Verlag Berlin Heidelberg. 1999.

[8] M. Rattichote, and W. Vatanawood. A Formal Spcification Method from State Diagram. Preceeding of the second International Conference on Intelligent Technologies (InTech'2001), pp. 402-408, Bangkok Thailand, November 2001.

[9] Michihiro Matsumoto, and Kokichi Futasugi. Object Composition and Refinement by using Non-Observable Projection Operators: A Case Study of the Automated Teller Machine system. OBJ/CafeOBJ/Maude at FM'99. Theta Bucharest/Romania. 1999.

[10] Andrew Harry. Formal Method Fact File VDM and Z. (n.p):John Wiley and Son. 1996.

[11] Kokichi Futatsugi, and Ataru Nakawa.An Overview of CafeOBJ Specification Environment: an algebraic approach for creating, verifying, and maintaining formal specifications over networks. Proceeding of 1st IEEE International Conference on Formal Engineering Methods. IEEE, 1997.

[12] Rãzvan Diaconescu ,and Kokichi Futatsugi. CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification. Volume 6 of AMAST series in Computing. World Scientific. 1998:99-102.

[13] Mark Allen Weiss. Data Structured and Algorithm Analysis in C. Second Edition. (n.p):Addision-wesley.1996: pp. 74-77.

[14] Kyle Loudon. Mastering Algorithms with C. First Edition. (n.p):O'Reilly.1999: pp.302-306.

ภาคผนวก

ภาคผนวก ก
วากยสัมพันธ์ของภาษาคาเฟโอบีเจ

เนื่องจากข้อมูลเข้าซึ่งเป็นข้อกำหนดของมอดูลต่างๆ นั้นเป็นข้อกำหนดของภาษาคาเฟโอบีเจ ดังนั้นมอดูลที่จะนำเข้าจะต้องมีความถูกต้องตามวากยสัมพันธ์ของภาษาคาเฟโอบีเจก่อน ซึ่งมีรูปแบบตาม BNF ดังนี้

CafeOBJ programs

program ::= { *module* | *view* | *eval* }

Module declaration

module ::= *module_type module_name* [*list_of_parameters*]
[*principal_sort*] "{" *module_element* * "}"

module_type ::= module | module! | module*

module_name ::= *ident*

parameters ::= "(" *parameter* +, ")"

parameter ::= [protecting | extending] *parameter_name* "::" *module_expr*

parameter_name ::= *ident*

principal_sort ::= principal_sort *sort_name*

module_elt ::= *sharing* | *import* | *sort* | *record* | *operation* | *variable* | *axiom*
| *comment*

sharing ::= share "(" *module_name* ")"

import ::= { protecting | extending | using }"(" *module_expr*, ... ")"

sort ::= *visible_sort* | *hidden_sort*

visible_sort ::= "[" *sort_decl* +, "]"

hidden_sort ::= "*[" *sort_decl* +, "]*"

sort_decl ::= *sort_name* + [*supersorts* +]

supersorts ::= "<" *sort_name* +

sort_name ::= *sort_symbol*[*qualifier*]

sort_symbol ::= *ident*

qualifier ::= "."*module_expression*[*qualifier*]

record ::= record *sort_name* [*super* +]"{" { *slot* | *comment* } + "}"

super ::= "[" *sort_name* ["(" *slot_rename* +, ")"] "]"

slot ::= *slot_name* ":" *sort_name*

| *slot_name* "=" "("*term*")" ":" *sort_name*

slot_name ::= *ident*

slot_rename ::= *slot_name* "->" *slot_name*

operation ::= { op | bop } *operation_symbol* ":" [*arity*] "->" *coarity*

[*op_attrs*]

arity ::= *sort_name* *

coarity ::= *sort_name*

op_attrs ::= "{" *op_attr* * "}"

op_attr ::= constr | associative | commutative | idempotent

| { id: | idr: } "(" *term* ")" | coherent

| strat: "(" *natural* ... ")"

| prec: *natural* | l-assoc | r-assoc

variable ::= var *var_name* ":" *sort_name* | vars *var_name* + ":" *sort_name*

var_name ::= *ident*

axiom ::= equation | cequation | transition | ctransition

equation ::= { eq | beq } [*label*] *term* "=" *term* "."

cequation ::= { ceq | bceq } [*label*] *term* "=" *term* if *term* "."

transition ::= { trans | btrans } [*label*] *term* "=>" *term* "."

ctransition ::= { ctrans | bctrans } [*label*] *term* "=>" *term* if *term* "."

label ::= "[" *ident* "]:"

comment ::= [-- | --> | ** | **>] ASCII character ...

View Declarations

view ::= view *view_name* from *module_expr* to *module_expr*

"{" *view_element* *, "}"

view_name ::= *ident*

view_element ::= *sort_map* | *op_view* | *variable*

op_view ::= *op_map* | *term* "->" *term*

Module Expressions

module_expression ::= module_name | sum | rename | instantiation
| "(" module_expr ")"

sum ::= *module_expr*ession { "+" *module_expr*ession } +

rename ::= *module_expr*ession "*" "{" *rename_map* +, "}"

instantiation ::= *module_expr* "(" { *ident*[*qualifier*] "<=" aview }+ ")"

rename_map ::= *sort-map* | *op_map*

sort_map ::= { sort | hsort } *sort_name* "->" *ident*

op_map ::= { op | bop } *op_name* "->" *operator_symbol*

op_name ::= *operation_symbol* | "(*operation_symbol*")"*qualifier*

aview ::= *view_name* | *module_expr*ession
| view to *module_expr*ession "{"*view_element* *, "}"

Evaluations

eval ::= *eval_interpreter* | *eval_tram*

eval_interpreter ::= { reduce | behavioral-reduce | execute } [*context*] *term* "."

context ::= in *module_expr*ession ":"

eval_tram ::= tram { reduce | execute } [*contect*] *term* "."

ภาคผนวก ข
อาร์พีเอ็นเท็กซ์สำหรับกรณีศึกษา

อาร์พีเอ็นเท็กซ์สำหรับกรณีศึกษาทั้ง 3 กรณี ได้แก่ การแปลงจากนิพจน์อินฟิกไปเป็นนิพจน์โพสต์ฟิก การแปลงจากนิพจน์โพสต์ฟิกไปเป็นนิพจน์อินฟิก และการเรียงลำดับแบบแทรก เมื่อทำการแปลงจากแผนภาพเครือข่ายอนุภาคความต้องการมาเป็นข้อมูลนำเข้าแบบ อาร์พีเอ็นเท็กซ์แล้ว จะได้อยู่ในรูปของตารางเอ็กเซลดังนี้

การแปลงจากนิพจน์อินฟิกไปเป็นนิพจน์โพสต์ฟิก

ตารางที่ ข-1 อาร์พีเอ็นเท็กซ์สำหรับการแปลงนิพจน์อินฟิกเป็นนิพจน์โพสต์ฟิก

#id	name	what	where	precond	out	ack	nack
R1P1	infix2postfix	{R1P2}{R1P3}{R1P4}	-	-	M	-	-
R1P2	tail	-	I	-	{R1P1}	-	-
R1P3	S	-	-	-	{R1P1}	-	-
R1P4	append	{R1P5}	P	{R1P6}	{R1P1}	-	-
R1P5	head	-	I	{R1P7}	{R1P4}{R1P}	-	-
R1P6	isoperand	{R1P5}	-	-	-	{R1P4}	-
R1P7	empty	-	I	-	-	-	{R1P5}
R2P1	infix2postfix	{R2P2}{R2P3}{R2P4}	-	-	M	-	-
R2P2	tail	-	I	-	{R2P1}	-	-
R2P3	push	{R2P5}	S	{R2P6}	{R2P1}	-	-
R2P4	P	-	-	-	{R2P1}	-	-
R2P5	head	-	I	{R2P7}	{R2P3}{R2P6}	-	-
R2P6	isopenparenthesis	{R2P5}	-	-	-	{R2P3}	-
R2P7	empty	-	I	-	-	-	{R2P5}
R3P1	infix2postfix	{R3P2}{R3P3}{R3P4}	-	-	M	-	-
R3P2	tail	-	I	-	{R3P1}	-	-
R3P3	push	{R3P5}	S	{R3P6}{R3P7}	{R3P1}	-	-
R3P4	P	-	-	-	{R3P1}	-	-
R3P5	head	-	I	{R3P8}	{R3P3}{R3P6}	-	-
R3P6	isoperator	{R3P5}	-	-	-	{R3P3}	-
R3P7	isempty	-	S	-	-	{R3P3}	-
R3P8	empty	-	I	-	-	-	{R3P5}
R4P1	infix2postfix	{R4P2}{R4P3}{R4P4}	-	-	M	-	-
R4P2	tail	-	I	-	{R4P1}	-	-
R4P3	push	{R4P5}	S	{R4P6}{R4P7}{R4P8}	{R4P1}	-	-
R4P4	P	-	-	-	{R4P1}	-	-
R4P5	head	-	I	{R4P9}	{R4P3}{R4P6}{R4P7}	-	-
R4P6	isoperator	{R4P5}	-	-	-	{R4P3}	-
R4P7	lw	{R4P10}{R4P5}	-	-	-	{R4P3}	-
R4P8	isempty	-	S	-	-	-	{R4P3}
R4P9	empty	-	I	-	-	-	{R4P5}
R4P10	top	-	S	-	{R4P7}	-	-

ตารางที่ ข-1 อาร์พีเอ็นเท็กซ์สำหรับการแปลงนิพจน์อินฟิกเป็นนิพจน์โพสต์ฟิก (ต่อ)

#id	name	what	where	precond	out	ack	nack
R5P1	infix2postfix	{R5P2}{R5P3}{R5P4}	-	-	M	-	-
R5P2	I	-	-	-	{R5P1}	-	-
R5P3	pop	-	S	-	{R5P1}	-	-
R5P4	append	{R5P5}	P	{R5P6}	{R5P1}	-	-
R5P5	top	-	S	{R5P9}	{R5P4}{R5P6}	-	-
R5P6	gte	{R5P5}{R5P7}	-	{R5P8}	-	{R5P4}	-
R5P7	head	-	I	{R5P10}	{R5P6}{R5P8}	-	-
R5P8	isoperator	{R5P7}	-	-	-	{R5P6}	-
R5P9	isempty	-	S	-	-	-	{R5P5}
R5P10	empty	-	I	-	-	-	{R5P7}
R6P1	infix2postfix	{R6P2}{R6P3}{R6P4}	-	-	M	-	-
R6P2	tail	-	I	-	{R6P1}	-	-
R6P3	push	{R6P5}	S	{R6P6}{R6P7}	{R6P1}	-	-
R6P4	P	-	-	-	{R6P1}	-	-
R6P5	head	-	I	{R6P8}	{R6P3}{R6P6}	-	-
R6P6	isoperator	{R6P5}	-	-	-	{R6P3}	-
R6P7	isopenparenthesis	{R6P9}	-	-	-	{R6P3}	-
R6P8	empty	-	I	-	-	-	{R6P5}
R6P9	top	-	S	{R6P10}	{R6P7}	-	-
R6P10	isempty	-	S	-	-	-	{R6P9}
R7P1	infix2postfix	{R7P2}{R7P3}{R7P4}	-	-	M	-	-
R7P2	I	-	-	-	{R7P1}	-	-
R7P3	pop	-	S	-	{R7P1}	-	-
R7P4	append	{R7P5}	P	{R7P6}{R7P7}	{R7P1}	-	-
R7P5	top	-	S	{R7P8}	{R7P4}{R7P6}	-	-
R7P6	isopenparenthesis	{R7P5}	-	-	-	-	{R7P4}
R7P7	iscloseparenthesis	{R7P9}	-	-	-	{R7P4}	-
R7P8	isempty	-	S	-	-	-	{R7P5}
R7P9	head	-	I	{R7P10}	{R7P7}	-	-
R7P10	empty	-	I	-	-	-	{R7P9}
R8P1	infix2postfix	{R8P2}{R8P3}{R8P4}	-	-	M	-	-
R8P2	tail	-	I	-	{R8P1}	-	-
R8P3	pop	-	S	{R8P5}{R8P6}	{R8P1}	-	-
R8P4	P	-	-	-	{R8P1}	-	-
R8P5	isopenparenthesis	{R8P7}	-	-	-	{R8P3}	-
R8P6	iscloseparenthesis	{R8P9}	-	-	-	{R8P3}	-
R8P7	top	-	S	{R8P8}	{R8P5}	-	-
R8P8	isempty	-	S	-	-	-	{R8P7}
R8P9	head	-	I	{R8P10}	{R8P6}	-	-
R8P10	empty	-	I	-	-	-	{R8P9}
R9P1	infix2postfix	{R9P2}{R9P3}{R9P4}	-	-	M	-	-
R9P2	I	-	-	-	{R9P1}	-	-
R9P3	pop	-	S	-	{R9P1}	-	-
R9P4	append	{R9P5}	P	{R9P6}	{R9P1}	-	-
R9P5	top	-	S	{R9P7}	{R9P4}	-	-
R9P6	empty	-	I	-	-	{R9P4}	-
R9P7	isempty	-	S	-	-	-	{R9P5}

การแปลงจากนิพจน์โพสต์ฟิกไปเป็นนิพจน์อินฟิก

ตารางที่ ข-2 อาร์พีเอ็นเท็กซ์สำหรับการแปลงนิพจน์โพสต์ฟิกเป็นนิพจน์อินฟิก

#id	name	what	where	precond	out	ack	nack
R1P1	postfix2infix	{R1P2}{R1P3}{R1P4}	-	-	M	-	-
R1P2	tail	-	P	-	{R1P1}	-	-
R1P3	S	-	-	-	{R1P1}	-	-
R1P4	push	{R1P5}	I	{R1P8}	{R1P1}	-	-
R1P5	list	{R1P7}{R1P6}	-	-	{R1P4}	-	-
R1P6	init-list	-	-	-	{R1P5}	-	-
R1P7	head	-	P	{R1P9}	{R1P5}{R1P8}	-	-
R1P8	isoperand	{R1P7}	-	-	-	{R1P4}	-
R1P9	empty	-	P	-	-	-	{R1P7}
R2P1	postfix2infix	{R2P2}{R2P3}{R2P4}	-	-	M	-	-
R2P2	P	-	-	-	{R2P1}	-	-
R2P3	push	{R2P5}	S	{R2P6}{R2P7}	{R2P1}	-	-
R2P4	pop	-	I	-	{R2P1}	-	-
R2P5	top	-	I	-	{R2P3}	-	-
R2P6	isempty	-	S	-	-	{R2P3}	-
R2P7	isoperator	{R2P8}	-	-	-	{R2P3}	-
R2P8	head	-	P	{R2P9}	{R2P7}	-	-
R2P9	empty	-	P	-	-	-	{R2P8}
R3P1	postfix2infix	{R3P2}{R3P3}{R3P4}	-	-	M	-	-
R3P2	tail	-	P	{R3P13}	{R3P1}	-	-
R3P3	pop	-	S	-	{R3P1}	-	-
R3P4	push	{R3P6}{R3P5}	-	-	{R3P1}	-	-
R3P5	pop	-	I	-	{R3P4}	-	-
R3P6	merge	{R3P7}{R3P8}	-	-	{R3P4}	-	-
R3P7	top	-	I	-	{R3P6}	-	-
R3P8	merge	{R3P9}{R3P14}	-	-	{R3P6}	-	-
R3P9	list	{R3P10}{R3P11}	-	-	{R3P8}	-	-
R3P10	head	-	P	{R3P12}	{R3P9}{R3P13}	-	-
R3P11	init-list	-	-	-	{R3P9}	-	-
R3P12	empty	-	P	-	-	-	{R3P10}
R3P13	isoperator	{R3P10}	-	-	-	{R3P2}	-
R3P14	top	-	S	{R3P15}	{R3P8}	-	-
R3P15	isempty	-	S	-	-	-	{R3P14}

การเรียงลำดับแบบแทรก

ตารางที่ ข-3 อาร์พีเอ็นเท็กซ์สำหรับการเรียงลำดับแบบแทรก

#id	name	what	where	precond	out	ack	nack
R1P1	issort	{R1P2}{R1P3}{R1P4}	-	-	M	-	-
R1P2	tail	-	Q	-	{R1P1}	-	-
R1P3	push	{R1P5}	S	{R1P6}	{R1P1}	-	-
R1P4	T	-	-	-	{R1P1}	-	-
R1P5	head	-	Q	{R1P7}	{R1P3}	-	-
R1P6	isempty	-	S	-	-	{R1P3}	-
R1P7	empty	-	Q	-	-	-	{R1P5}
R2P1	issort	{R2P2}{R2P3}{R2P4}	-	-	M	-	-
R2P2	tail	-	Q	-	{R2P1}	-	-
R2P3	push	{R2P5}	S	{R2P6}{R2P7}	{R2P1}	-	-
R2P4	T	-	-	-	{R2P1}	-	-
R2P5	head	-	Q	{R2P8}	{R2P3}{R2P6}	-	-
R2P6	gte	{R2P5}{R2P9}	-	-	-	-	{R2P3}
R2P7	isempty	-	T	-	-	{R2P3}	-
R2P8	empty	-	Q	-	-	-	{R2P5}
R2P9	top	-	S	-	{R2P6}	-	-
R3P1	issort	{R3P2}{R3P3}{R3P4}	-	-	M	-	-
R3P2	Q	-	-	-	{R3P1}	-	-
R3P3	pop	-	S	-	{R3P1}	-	-
R3P4	push	{R3P5}	T	{R3P6}	{R3P1}	-	-
R3P5	top	-	S	-	{R3P4}{R3P6}	-	-
R3P6	gte	{R3P8}{R3P5}	-	-	-	{R3P4}	-
R3P7	empty	-	Q	-	-	-	{R3P8}
R3P8	head	-	Q	{R3P7}	{R3P6}	-	-
R4P1	issort	{R4P2}{R4P3}{R4P4}	-	-	M	-	-
R4P2	tail	-	Q	-	{R4P1}	-	-
R4P3	push	{R4P5}	S	{R4P6}{R4P10}	{R4P1}	-	-
R4P4	T	-	-	-	{R4P1}	-	-
R4P5	head	-	Q	{R4P7}	{R4P3}{R4P6}{R4P10}	-	-
R4P6	gte	{R4P5}{R4P8}	-	-	-	{R4P3}	-
R4P7	empty	-	Q	-	-	-	{R4P5}
R4P8	top	-	T	{R4P9}	{R4P6}	-	-
R4P9	isempty	-	T	-	-	-	{R4P8}
R4P10	gte	{R4P5}{R4P11}	-	-	-	-	{R4P3}
R4P11	top	-	S	-	{R4P10}	-	-
R5P1	issort	{R5P2}{R5P3}{R5P4}	-	-	M	-	-
R5P2	Q	-	-	-	{R5P1}	-	-
R5P3	push	{R5P5}	S	{R5P6}{R5P7}	{R5P1}	-	-
R5P4	pop	-	T	-	{R5P1}	-	-
R5P5	top	-	T	{R5P9}	{R5P3}{R5P6}	-	-
R5P6	gte	{R5P8}{R5P5}	-	-	-	-	{R5P3}
R5P7	gte	{R5P8}{R5P11}	-	-	-	-	{R5P3}
R5P8	head	-	Q	{R5P10}	{R5P6}{R5P7}	-	-
R5P9	isempty	-	T	-	-	-	{R5P5}
R5P10	empty	-	Q	-	-	-	{R5P8}
R5P11	top	-	S	-	{R5P7}	-	-
R6P1	issort	{R6P2}{R6P3}{R6P4}	-	-	M	-	-
R6P2	Q	-	-	-	{R6P1}	-	-
R6P3	push	{R6P5}	S	{R6P6}	{R6P1}	-	-
R6P4	pop	-	T	-	{R6P1}	-	-
R6P5	top	-	T	{R6P7}	{R6P3}	-	-
R6P6	empty	-	Q	-	-	{R6P3}	-
R6P7	isempty	-	T	-	-	-	{R6P5}

ภาคผนวก ค
คลังจัดเก็บตัวดำเนินการสำหรับกรณีศึกษา

เนื่องจากการแปลงข้อกำหนดคาเฟโอบีเจจากแผนภาพเครือข่ายอนุภาคความต้องการ จะต้องใช้โหนดการดำเนินการที่ตรงกับที่ระบุไว้ในคลังจัดเก็บตัวดำเนินการเท่านั้น ซึ่งคลังจัดเก็บตัวดำเนินการนี้ สามารถเขียนเพิ่มได้ซึ่งจะอยู่ในรูปของมอดูลคาเฟโอบีเจ ในที่นี้ได้ทำการกำหนดมอดูลไว้ในคลังจัดเก็บตัวดำเนินการเริ่มต้น 5 มอดูลคือ มอดูลการตรวจสอบความเป็นสมาชิก (ismember) มอดูลโครงสร้างข้อมูลแบบรายการ (list) มอดูลโครงสร้างข้อมูลแบบกองซ้อน (stack) มอดูลโครงสร้างข้อมูล แบบกองซ้อนโดยใช้ข้อมูลแบบรายการ (stack of list) และมอดูลการตรวจสอบความสำคัญ (precedence) โดยมีรายละเอียดดังต่อไปนี้

มอดูลการตรวจสอบความเป็นสมาชิก

มอดูลการตรวจสอบความเป็นสมาชิก จะดำเนินการตรวจสอบดังนี้

- การดำเนินการตรวจสอบตัวดำเนินการ (isoperator)

  จะตรวจสอบข้อมูลที่เข้ามาว่าเป็นตัวดำเนินการหรือไม่ ถ้าเป็นจะให้ค่าตรรกะเป็นจริง แต่ถ้าไม่เป็นตัวดำเนินการจะให้ค่าเท็จ

- การดำเนินการตรวจสอบตัวถูกดำเนินการ (isoperand)

  จะตรวจสอบข้อมูลที่เข้ามาว่าเป็นตัวถูกดำเนินการหรือไม่ ซึ่งจะให้ผลลัพธ์เป็นค่าทางตรรกะ

- การดำเนินการตรวจสอบวงเล็บเปิด (isopenparenthesis)

  จะตรวจสอบข้อมูลที่เข้ามาว่าเป็นเครื่องหมายวงเล็บเปิดหรือไม่

- การดำเนินการตรวจสอบวงเล็บปิด (iscloseparenthesis)

  จะตรวจสอบข้อมูลที่เข้ามาว่าเป็นเครื่องหมายวงเล็บปิดหรือไม่

มอดูลการตรวจสอบความเป็นสมาชิก เมื่ออยู่ในรูปของข้อกำหนดคาเฟโอบีเจ จะเป็นดังรูปที่ ค-1

มอดูลโครงสร้างข้อมูลแบบรายการ

มอดูลโครงสร้างแบบรายการ จะมีการดำเนินการดังต่อไปนี้

- การดำเนินการตรวจสอบว่าไม่มีข้อมูลอยู่ในโครงสร้างข้อมูลแบบรายการ (empty)

  ถ้ามีข้อมูลอยู่จะให้ค่าตรรกะเป็นเท็จ แต่ถ้าไม่มีข้อมูลอยู่จะให้ค่าเป็นจริงออกมา

```
mod* ISMEMBER {
        pr(STRING)
        [ Nat Elt ]

        op init : -> String
        op isnil : String -> Bool
        op isoperator : String -> Bool
        op isoperand : String -> Bool
        op isopenparenthesis : String -> Bool
        op iscloseparenthesis : String -> Bool
        var A : Elt
        var S : String
        eq isnil(init) = true .
        eq isoperator(init) = false .
        eq isoperand(init) = false .
        eq isopenparenthesis(init) = false .
        eq iscloseparenthesis(init) = false .
        ceq isnil(S) = true if S == "" .
        ceq isnil(S) = false if S =/= "" .
        ceq isoperator(S) = true if S == "+" .
        ceq isoperator(S) = true if S == "-" .
        ceq isoperator(S) = true if S == "*" .
        ceq isoperator(S) = true if S == "/" .
        ceq isoperator(S) = false if S =/= "*" and S =/= "-" and S =/= "+" and S =/= "/" .
        ceq isopenparenthesis(S) = true if S == "(" .
        ceq isopenparenthesis(S) = false if S =/= "(" .
        ceq iscloseparenthesis(S) = true if S == ")" .
        ceq iscloseparenthesis(S) = false if S =/= ")" .
        ceq isoperand(S) = true if S =/= "*" and S =/= "-" and S =/= "+" and S =/= "/"
                and S =/= "(" and S =/= ")" and S =/= "" .
        ceq isoperand(S) = false if S == "*" or S == "-" or S == "+" or S == "/" or S == "("
                or S =/= ")" or S == "" .
}
```

รูปที่ ค-1 มอดูลข้อกำหนดคาเฟโอบีเจแสดงการตรวจสอบการเป็นสมาชิก

```
mod* LIST {
        pr(STRING)
        [ List ]

        op init-list : -> List
        op empty : List -> Bool
        op list : String List -> List
        op head : List -> String
        op tail : List -> List
        op append : List String -> List
        op last : List -> String
        op merge : List List -> List

        vars E E' : String
        vars L L' : List

        eq empty(init-list) = true .
        eq empty(list(E,L)) = false .
        eq head(list(E,L)) = E .
        ceq head(list(E,L)) = E if L == init-list .
        eq tail(list(E,L)) = L .
        ceq tail(list(E,L)) = init-list if L == init-list .
        ceq tail(list(E,L)) = list(E,tail(L)) if L =/= init-list .
        eq append(init-list, E) = list(E,init-list) .
        eq append(list(E,L), E') = list(E, append(L,E')) .
        ceq last(list(E,L)) = E if L == init-list .
        ceq last(L) = last(tail(L)) if tail(L) =/= init-list .
        ceq last(L) = head(L) if tail(L) == init-list .
        eq merge(init-list, L) = L .
        eq merge(list(E, L), L') = list(E, merge(L , L')) .
}
```

รูปที่ ค-2 มอดูลข้อกำหนดคาเฟโอบีเจแสดงโครงสร้างข้อมูลแบบรายการ

- การดำเนินการหาส่วนหัวของโครงสร้างข้อมูลแบบรายการ (head)
- การดำเนินการหาส่วนหางของโครงสร้างข้อมูลแบบรายการ (tail)

- การดำเนินการเพิ่มข้อมูลเข้าไปยังโครงสร้างข้อมูลแบบรายการ (append)
- การดำเนินการหาส่วนท้ายสุดของโครงสร้างข้อมูลแบบรายการ (last)
- การดำเนินการรวมข้อมูลแบบรายการสองรายการเข้าด้วยกัน (merge)

มอดูลโครงสร้างข้อมูลแบบรายการ เมื่ออยู่ในรูปของข้อกำหนดคาเฟโอบีเจ จะเป็นดังรูปที่ ค-2

มอดูลโครงสร้างข้อมูลแบบกองซ้อน

มอดูลโครงสร้างแบบกองซ้อนจะมีการดำเนินการดังต่อไปนี้

- การดำเนินการตรวจสอบว่าไม่มีข้อมูลอยู่ในกองซ้อน (isempty)

  ถ้ามีข้อมูลอยู่จะให้ค่าตรรกะเป็นเท็จ แต่ถ้าไม่มีข้อมูลอยู่หรือเป็นกองซ้อนว่างจะให้ค่าจริง

```
mod* STACK {
        pr(STRING)
        *[ Stack ]*

        op init : -> Stack
        bop isempty : Stack -> Bool
        bop top_ : Stack -> String
        bop pop_ : Stack -> Stack
        op push : String Stack -> Stack

        var A : String
        var B : Stack

        eq isempty(init) = true .
        eq isempty(push(A,B)) = false .
        eq pop push(A,B) = B .
        eq top push(A,B) = A .
}
```

รูปที่ ค-3 มอดูลข้อกำหนดคาเฟโอบีเจแสดงโครงสร้างข้อมูลแบบกองซ้อน

- การดำเนินการหาค่าที่อยู่บนสุดของกองซ้อน (top)
- การดำเนินการอ่านค่าที่อยู่บนสุดของกองซ้อน (pop)
- การดำเนินการเพิ่มค่าลงไปยังบนสุดของกองซ้อน (push)

มอดูลโครงสร้างข้อมูลแบบกองซ้อน เมื่ออยู่ในรูปของข้อกำหนดคาเฟโอบีเจ จะเป็นดังรูปที่ ค-3

มอดูลโครงสร้างข้อมูลแบบกองซ้อนโดยใช้ข้อมูลแบบรายการ

มอดูลโครงสร้างข้อมูลแบบกองซ้อนโดยใช้ข้อมูลแบรายการ จะมีการดำเนินการเหมือนกับโครงสร้างข้อมูลแบบกองซ้อนทุกประการ จะต่างกันเพียงชนิดของข้อมูลที่ใช้สำหรับมอดูลนี้จะเป็นชนิดของข้อมูลสำหรับโครงสร้างข้อมูลแบบรายการ

มอดูลโครงสร้างข้อมูลแบบกองซ้อนโดยใช้ข้อมูลแบบรายการ เมื่ออยู่ในรูปของข้อกำหนดคาเฟโอบีเจ จะเป็นดังรูปที่ ค-4

```
mod* STACK-OF-LIST {
        pr(LIST)
        *[ Stack ]*

        op init : -> Stack
        bop isempty : Stack -> Bool
        bop top_ : Stack -> List
        bop pop_ : Stack -> Stack
        op push : List Stack -> Stack

        var A : List
        var B : Stack

        eq isempty(init) = true .
        eq isempty(push(A,B)) = false .
        eq pop push(A,B) = B .
        eq top push(A,B) = A .
}
```

รูปที่ ค-4มอดูลข้อกำหนดคาเฟโอบีเจแสดงโครงสร้างข้อมูลแบบกองซ้อนโดยใช้ข้อมูลแบบรายการ

มอดูลการตรวจสอบความสำคัญ

มอดูลการตรวจสอบความสำคัญจะทำการเปรียบเทียบค่าของคู่ที่เข้า โดยมีการดำเนินการดังนี้

- การดำเนินการเปรียบเทียบค่ามากกว่าหรือเท่ากับ (gte)

จะทำการเปรียบเทียบค่าสองค่าที่เข้ามา โดยจะให้ค่าตรรกะเป็นจริงก็ต่อเมื่อค่าที่อยู่ส่วนหน้ามีค่ามากกว่าหรือเท่ากับค่าที่อยู่ส่วนหลัง

- การดำเนินการเปรียบเทียบค่าน้อยกว่า (lw)

จะทำการเปรียบเทียบค่าสองค่าที่เข้ามา โดยจะให้ค่าตรรกะจริงก็ต่อเมื่อค่าที่อยู่ส่วนหน้ามีค่าน้อยกว่าค่าที่อยู่ส่วนหลัง

มอดูลโครงสร้างข้อมูลแบบรายการ เมื่ออยู่ในรูปของข้อกำหนดคาเฟโอบีเจ จะเป็นดังรูปที่ ค-5

```
mod* PRECEDENCE {
        pr(STRING)

        op gte : String String -> Bool
        op lw : String String -> Bool
        op precedence : String -> Nat

        vars S S' : String

        ceq precedence(S) = 2 if S == "*" .
        ceq precedence(S) = 2 if S == "/" .
        ceq precedence(S) = 1 if S == "+" .
        ceq precedence(S) = 1 if S == "-" .

        ceq precedence(S) = 97 if S == "a" .
        ceq precedence(S) = 98 if S == "b" .
        ceq precedence(S) = 99 if S == "c" .
        ceq precedence(S) = 100 if S == "d" .
        ceq precedence(S) = 101 if S == "e" .
        ceq precedence(S) = 102 if S == "f" .
```

รูปที่ ค-5 มอดูลข้อกำหนดคาเฟโอบีเจแสดงการตรวจสอบความสำคัญ

```
        ceq precedence(S) = 103 if S == "g" .
        ceq precedence(S) = 104 if S == "h" .
        ceq precedence(S) = 105 if S == "i" .
        ceq precedence(S) = 106 if S == "j" .
        ceq precedence(S) = 107 if S == "k" .
        ceq precedence(S) = 108 if S == "l" .
        ceq precedence(S) = 109 if S == "m" .
        ceq precedence(S) = 110 if S == "n" .
        ceq precedence(S) = 111 if S == "o" .
        ceq precedence(S) = 112 if S == "p" .
        ceq precedence(S) = 113 if S == "q" .
        ceq precedence(S) = 114 if S == "r" .
        ceq precedence(S) = 115 if S == "s" .
        ceq precedence(S) = 116 if S == "t" .
        ceq precedence(S) = 117 if S == "u" .
        ceq precedence(S) = 118 if S == "v" .
        ceq precedence(S) = 119 if S == "w" .
        ceq precedence(S) = 120 if S == "x" .
        ceq precedence(S) = 121 if S == "y" .
        ceq precedence(S) = 122 if S == "z" .

        ceq  gte(S, S') = true if (precedence(S) >= precedence(S')) .
        ceq  gte(S, S') = false if precedence(S) < precedence(S') .
        ceq  lw(S, S') = true if precedence(S) < precedence(S') .
        ceq  lw(S, S') = false if precedence(S) >= precedence(S') .
}
```

รูปที่ ค-5 มอดูลข้อกำหนดคาเฟโอบีเจแสดงการตรวจสอบความสำคัญ (ต่อ)

ภาคผนวก ง
คู่มือการใช้งาน

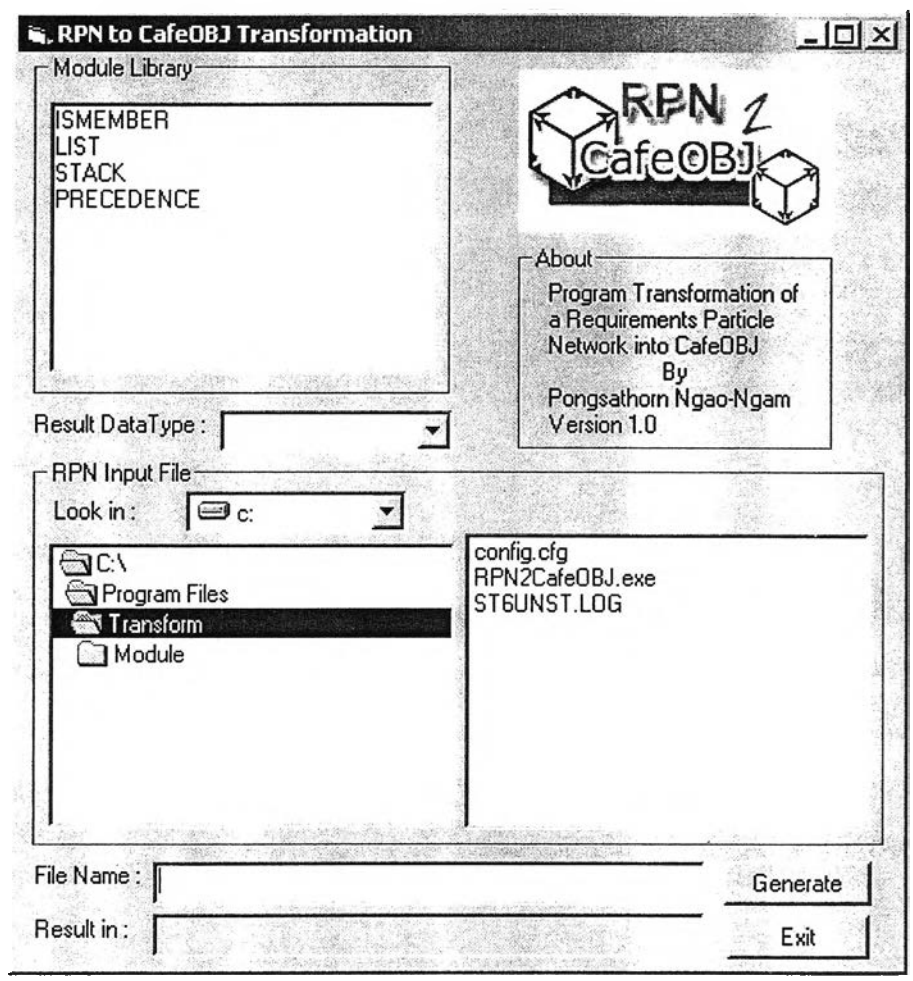
การใช้โปรแกรม CafeOBJ2RPN สำหรับการแปลงเครือข่ายอนุภาคความต้องการเป็นข้อกำหนดรูปนัยคาเฟโอบีเจ มีวิธีการดังนี้

เริ่มต้นการใช้

เมื่อเริ่มต้นใช้งานครั้งแรก เรียกโปรแกรม RPN2CafeOBJ.exe หรือจะเรียกจากรายการเลือก

Start → Programs → Transform → RPN2CafeOBJ

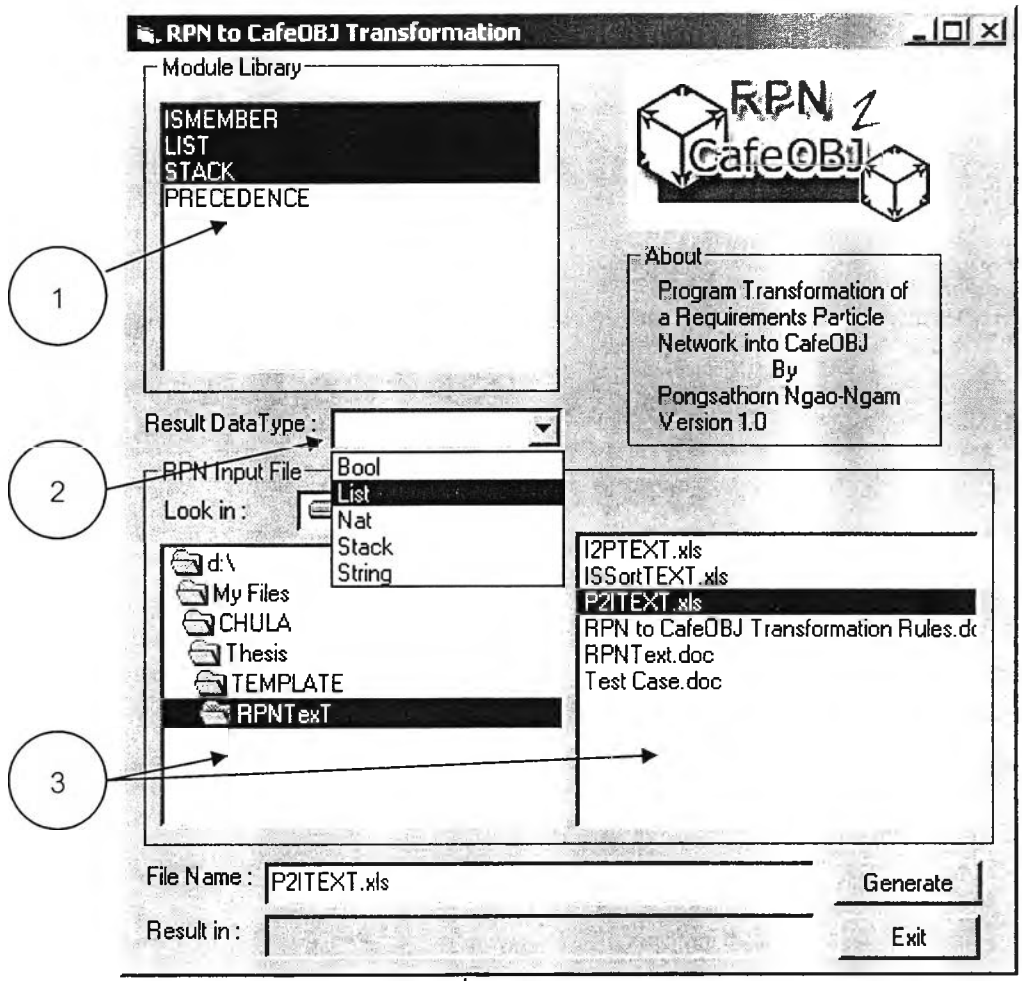
จากนั้นจะปรากฏหน้าจอดังรูปที่ ง-1



รูปที่ ง-1 หน้าจอเริ่มต้นโปรแกรม

การใส่ข้อมูลนำเข้า

หลังจากเตรียมข้อมูลนำเข้าเรียบร้อยแล้ว ทำการกำหนดข้อมูลนำเข้าสำหรับส่วนต่างๆ ดังรูปที่ ง-2



รูปที่ ง-2 การใส่ข้อมูลนำเข้า

1) ส่วนกำหนดมอดูล (Module)

เลือกมอดูลที่ต้องนำมาใช้ (Import) ในข้อกำหนดที่จะสร้างขึ้น โดยสามารถเลือกได้มากกว่า 1 มอดูล โดยการกดปุ่ม Ctrl หรือ Shift ค้างพร้อมกับการเลือก

2) ส่วนกำหนดชนิดข้อมูลของผลลัพธ์ (Result Data Type)

ผู้ใช้จะต้องทราบชนิดของการดำเนินการที่จะสร้างข้อกำหนดขึ้นมา และทำการเลือกชนิดของมูล (Sort) จากรายการที่แสดงให้ถ้าไม่มีในตัวเลือก ผู้ใช้สามารถพิมพ์เพื่อกำหนดใหม่เพิ่มขึ้นมาได้

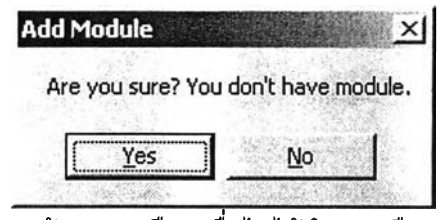
3) ส่วนนำเข้าอาร์พีเอ็นเท็กซ์ (RPN input file)

จากนั้นผู้ใช้ต้องเลือกแฟ้มข้อมูลอาร์พีเอ็นเท็กซ์ที่ได้เตรียมไว้ โดยชื่อที่ถูกเลือกจะปรากฏอยู่ในส่วนที่ระบุชื่อของแฟ้มข้อมูล (File Name)

การสร้างมอดูลข้อกำหนด

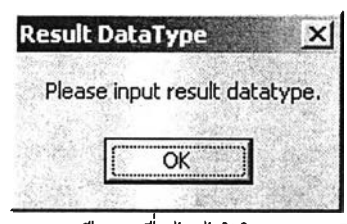
เมื่อป้อนข้อมูลนำเข้าเรียบร้อยแล้ว ให้กดปุ่มสร้าง (Generate) หากข้อมูลนำเข้าไม่ครบจะมีข้อความเตือนต่างๆดังนี้

- เมื่อไม่ได้ทำการเลือกมอดูลจากคลังจัดเก็บอนุภาค ระบบจะทำการถามเพื่อความ แน่ใจต้องการใช้หรือไม่ ดังรูปที่ ง-3



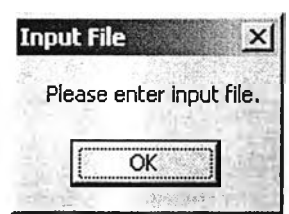
รูปที่ ง-3 ข้อความเตือนเมื่อไม่ได้ทำการเลือกมอดูล

- เมื่อไม่ได้ทำการกำหนดชนิดของข้อมูล ระบบจะเตือนให้ทราบว่าจำเป็นจะต้องใส่ข้อมูลในส่วนนี้ดังรูปที่ ง-4



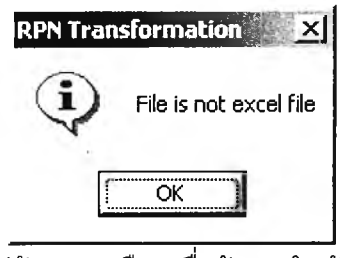
รูปที่ ง-4 ข้อความเตือนเมื่อไม่ได้กำหนดชนิดของข้อมูล

- เมื่อไม่ได้ทำการเลือกข้อมูลนำเข้าอาร์พีเอ็นท์ ระบบจะเตือนให้ใส่ข้อมูลนำเข้าดังรูปที่ ง-5



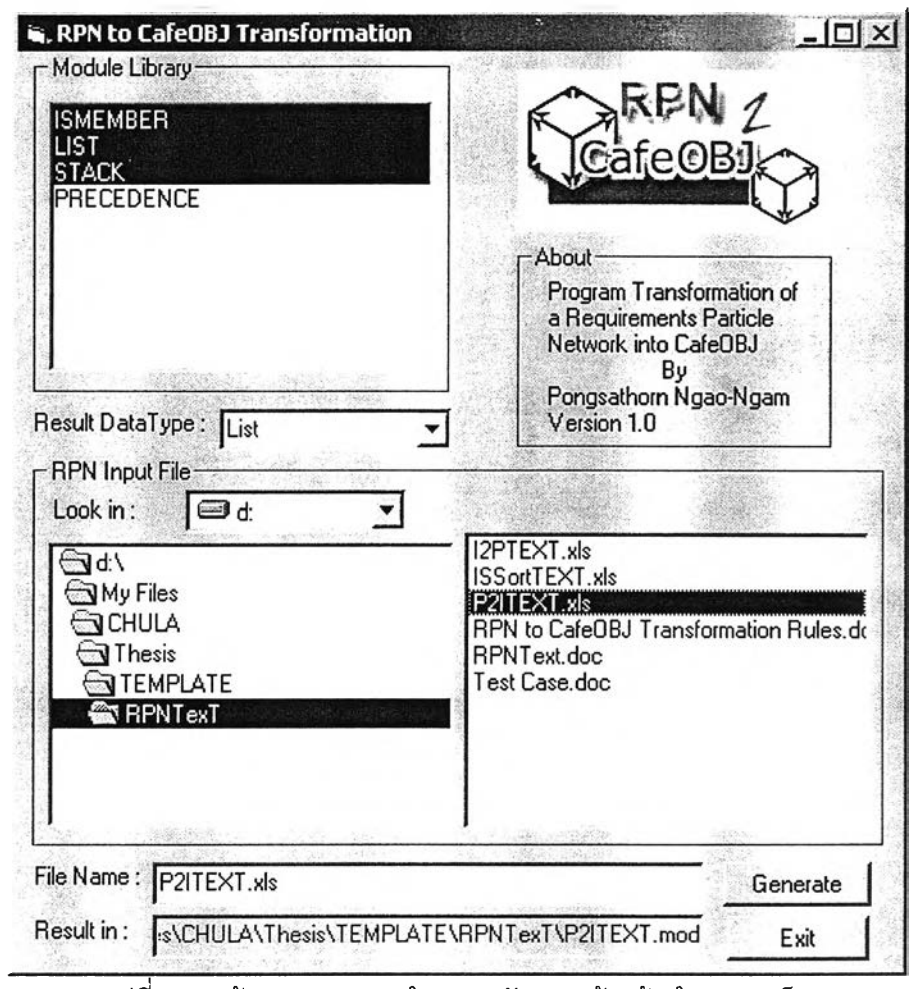
รูปที่ ง-5 รูปข้อความเตือนเมื่อไม่ได้เลือกข้อมูลนำเข้าอาร์พีเอ็นเท็กซ์

- เมื่อใส่ข้อมูลนำเข้าผิด โดยที่แฟ้มข้อมูลนำเข้าไม่ใช่แฟ้มเอ็กเซล ระบบจะเตือนบอกว่าชนิดของแฟ้มข้อมูลไม่ถูกต้องดังรูปที่ ง-6



รูปที่ ง-6 รูปข้อความเตือนเมื่อข้อมูลนำเข้าไม่ถูกต้อง

เมื่อการใส่ข้อมูลนำเข้าครบถ้วนและถูกต้องแล้วจึงกดปุ่ม Generate จะมีแท่งแสดงสถานะการทำงานขึ้นข้างล่างจนกระทั่งเสร็จ และจะปรากฏชื่อแฟ้มข้อมูลของผลลัพธ์ที่ช่องแสดงผลลัพธ์ (Result in) ดังรูปที่ ง-7

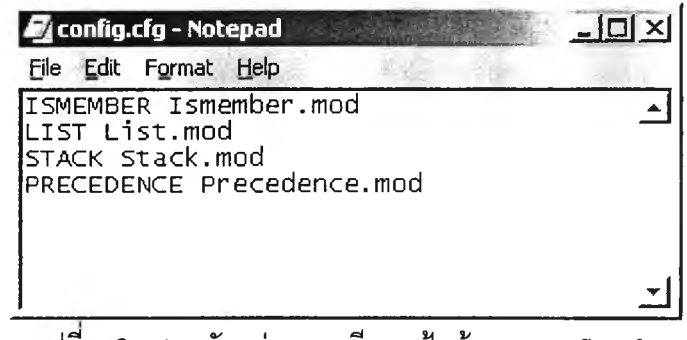


รูปที่ ง-7 หน้าจอแสดงการทำงานหลังจากสร้างข้อกำหนดเสร็จ

การเพิ่มเติมมอดูลในคลังจัดเก็บอนุภาค

ผู้ใช้สามารถกำหนดมอดูลเพิ่มในคลังจัดเก็บอนุภาคได้ โดยการนำแฟ้มข้อกำหนดคาเฟโอบีเจที่เตรียมไว้ และถูกต้องตามวายะสัมพันธ์ของภาษาคาเฟโอบีเจ สำเนาใส่ไว้ภายในโฟลเดอร์ชื่อ Module ภายใต้ตำแหน่งที่ทำการติดตั้งโปรแกรม (โดยปรกติจะอยู่ที่ C:\Program Files\Transform) และแก้ไขแฟ้มข้อมูลชื่อ config.cfg เพื่อเพิ่มชื่อของมอดูลเข้าไปให้โปรแกรมแสดงขึ้นมา โดยให้อยู่ในรูปแบบดังตัวอย่างในรูปที่ ง-8 ซึ่งมีรูปแบบดังนี้

(ชื่อที่ต้องการให้ปรากฏในส่วนของ Module ในโปรแกรม) (ชื่อแฟ้มนามสกุล .mod)



รูปที่ ง-8 แสดงตัวอย่างการเขียนแฟ้มข้อมูล config.cfg

ประวัติผู้เขียนวิทยานิพนธ์

นายพงศธร เงางาม เกิดเมื่อวันที่ 3 มีนาคม พ.ศ. 2518 ที่จังหวัดสุรินทร์ สำเร็จการศึกษาหลักสูตรวิศวกรรมศาสตร์บัณฑิต (วศ.บ.) สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ เมื่อปีการศึกษา 2538 จากนั้นได้เข้าทำงานที่ ที.เอ็น. อินฟอร์เมชั่น ซิสเท็ม จำกัด ต่อมาในปีการศึกษา 2543 ได้เข้าศึกษาต่อหลักสูตรวิทยาศาสตร์มหาบัณฑิต (วท.ม.) สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปัจจุบันทำงานที่ บริษัท เยเนอรัล อิเล็คทริค อินเตอร์เนชันเนล โอเปอเรชั่นส์ คอมพานี อิงค์ (GE Thailand)