THE *NAVIGATION PLAN DEFINITION LANGUAGE* (NPDL) SYNTAX

TECHNICAL REPORT

Kelly Rosa Braghetto[1], Osvaldo K. Takai[1], João Eduardo Ferreira[1], Calton Pu[2]

[1] Department of Computer Science,
University of São Paulo
Rua do Matão, 1010 - Cidade Universitária
05508-090 São Paulo - SP – Brasil
e-mail: {kellyrb, takai, jef}@ime.usp.br

[2]College of Computing,
Georgia Institute of Technology
801 Atlantic Drive, Atlanta GA 30332-0280 – USA
e-mail: calton@cc.gatech.edu

1. Introduction

The *Navigation Plan Definition Language* uses the main concepts of *RiverFish* Architecture [2][3] and Process Algebra [4] to represent processes. According to Process Algebra, the behavior of a system can be modeled as algebraic expressions. These expressions are formed by atomic actions composed by operators that indicate the execution order of these actions. *RiverFish* Architecture proposes a relational structure to maintain processes, processes instances and execution log data.

The NPDL language defines commands to create, delete and update processes in a relational database. As in Process Algebra, in NPDL processes are defined by algebraic expressions involving mainly actions and operators. NPDL implements the most important operators of Process Algebra. NPDL defines also other operators that model frequent behaviors in workflow processes that are difficult do map only with Process Algebra operators.

2. NPDL Syntax

Consider that the grammar of NPDL language is represented by *(N, T, P, S)*, where *N* is the set of variable (non terminal) symbols; *T* is the set of terminal symbols; *P* is the set of grammar production rules and *S* is the starting variable from *N*.

The NPDL obeys the following properties:

1) $S \in N$

2) *N* = { action-description, add-action-execution-call, add-function-execution-call, add-process-service-description, add-rule-execution-call, choice-operator-sign, command, condition, deadlock-symbol, delete-action, delete-function, delete-process, delete-rule, discriminator-operator-sign, execution-call, factor, function-description, function-limited-repetition, interleaved-parallel-operator-sign, left-parentheses, limited-repetition, list-actions, list-functions, list-navigation-plan, list-processes, list-rules, multi-merge-sign, new-action, new-function, new-process, new-rule, parallel-operator-sign, process, process-description, process-expression, right-parentheses, S, service-description, sequential-operator-sign, term, rule-description, unlimited-repetition}

3) *T* = {ACTION, ACTIONS, ADD, CALL, CREATE, DESCRIPTION, DROP, EXECUTION, FROM, FUNCTION, FUNCTIONS, NAVIGATION, PLAN, PROCESS, PROCESSES, RULE, RULES, SELECT, SERVICE, SET, \n**,** (**,**) **,** = **,** . , +**,** ^, || , |*, #, *, ?, &, %, %!}

4) The set *P* is generated through the production rules described in BNF (*Backus-Naur Form*) format in Table 1.

Table 1: NPDL Production Rules

```
<S> ::= (<command>? \n )+
<command> ::= <new-action>
    | <new-function>
    | <new-process>
    | <new-rule>
    | <add-action-execution-call>
    | <add-function-execution-call>
```

```
    | <add-process-service-description>
    | <add-rule-execution-call>
    | <delete-action>
    | <delete-function>
    | <delete-process>
    | <delete-rule>
    | <process-expression>
    | <list-actions>
    | <list-functions>
    | <list-rules>
    | <list-processes>
    | <list-navigation-plan>
<new-process> ::= CREATE PROCESS <process-description>
                            [<service-description>]
<new-action>  ::= CREATE ACTION <action-description>
                            [<execution-call>]
<new-function> ::= CREATE FUNCTION <function-description>
                            [<execution-call>]
<new-rule> ::= CREATE RULE <rule-description>
                            [<execution-call>]
<add-process-service-description> ::= ADD <process-description>
                        SERVICE DESCRIPTION <service-description>
<add-action-execution-call> ::= ADD <action-description>
                        EXECUTION CALL <execution-call>
<add-function-execution-call> ::= ADD <function-description>
                         EXECUTION CALL <execution-call>
<add-rule-execution-call> ::= ADD <rule-description>
                         EXECUTION CALL <execution-call>
<process-expression> ::= SET <process-description> = <process>
<delete-process> ::= DROP PROCESS <process-description>
<delete-action> ::= DROP ACTION <action-description>
<delete-function> ::= DROP FUNCTION <function-description>
<delete-rule> ::= DROP RULE <rule-description>
<list-processes> ::= SELECT PROCESSES
<list-actions> ::= SELECT ACTIONS
<list-functions> ::= SELECT FUNCTIONS
<list-rules> ::= SELECT RULES
<list-navigation-plan> ::= SELECT NAVIGATION PLAN FROM PROCESS
```

```
  <process-description>
<left-parentheses> ::= (
<right-parentheses> ::= )
<choice-operator-sign> ::= +
<sequential-operator-sign> ::= .
<parallel-operator-sign> ::= ||
<interleaved-parallel-operator-sign> ::= |*
<multi-merge-operator-sign> ::= &
<discriminator-operator-sign> ::= ^
<deadlock-symbol> ::= #
<unlimited-repetition> := ?*
<limited-repetition> := ? <unsigned_integer>[1]
<function-limited-repetition> := ? <function>
<condition> := % <rule>
<not-condition> := %! <rule>
<process> ::= <term>
           | <process> <choice-operator-sign> <term>
<term> ::= <subterm>
           | <term> <interleaved-parallel-operator-sign> <subterm>
<subterm> ::= <prefactor>
           | <subterm> <parallel-operator-sign> <prefactor>
<prefactor> ::= <factor>
           | <prefactor> <sequential-operator-sign> <factor>
<factor> ::= <subfactor>
           | <factor> <multi-merge-operator-sign> <subfactor>
           | <factor> <discriminator-operator-sign> <subfactor>
<subfactor> ::= <action>
           | <process-description>
           | <process>
           | <deadlock-symbol>
           | <left-parentheses> <process> <right-parentheses>
           | <subfactor><unlimited-repetition>
           | <subfactor><limited-repetition>
           | <subfactor><function-limited-repetition>
           | <condition><subfactor>
```

[1] <unsigned_integer> is defined in ANSI SQL92 BNF specification.

```
          | <not-condition><subfactor>
<action> ::= <action-description>
<rule> ::= <rule-description>
<function> ::= <function-description>
<action-description> ::= <identifier>[2]
<rule-description> ::= <identifier>
<process-description> ::= <identifier>
<service-description> ::= <character_string_literal>[3]
<execution-call> ::= <character_string_literal>
```

The draft version of NPDL was presented in [1] and had only three composition operators: *sequential composition* (•), *alternative composition* (+) and *parallel composition* (||). With these operators it is possible specify only basic control-flow structures. The BNF presented in this document corresponds to final version of NPDL, which was extended with additional operators that enable NPDL to specify frequent behaviors of control-flows.

Examples of NPDL commands:

```
CREATE RULE R1 'VerifyPurchasingForm';
CREATE ACTION A1 'ProcessPurchasing';
CREATE ACTION A2 'ApplyDiscount';
CREATE ACTION A3 'PrintReceipt';
CREATE PROCESS P1 'Purchasing Control Process';
SET P1 = %R1 A1 . ( A2.A3 + A2);
```

[2] <identifier> is defined in ANSI SQL92 BNF specification.

[3] <character_string_literal> is defined in ANSI SQL92 BNF specification.

References

[1] K. R. Braghetto, O. K. Takai, J.E, Ferreira, C.PU. NavigationPlanTool: Uma Ferramenta para o Controle de Processos no Modelo de Dados Relacional. Accepted for DEMO section in SBBD2005.

http://www.sbbd-sbes2005.ufu.br/arquivos/NavPlanTool.pdf

[2] J. E. Ferreira, O. K. Takai, C. Pu. Integration of Business Processes with Autonomous Information Systems: A Case Study in Government Services. In: 7th International IEEE

[3] J.E. Ferreira; O.K. Takai; C. Pu. Integration of Collaborative Information System in Internet Applications using RiverFish Architecture. Accepted for International Conference on Collaborative Computing: Networking, Applications and Work sharing, 2005, San Jose. USA.

[4] W.J. Fokkink, Introduction o Process Algebra. Texts in Theoretical Computer Science. Springer-Verlag, Berlin, 2000.