BPELVT BPEL Verification TOOL	 ReDCAD Laboratory, University of Sfax , Tunisia

Rewrite rules

In the following, we present rewrite rules of BPEL grammar.

system: ^(RACRUL proc+);

proc : ^(PROCDEF name targetnamsp supjoinFail (xmlns)+ (importact)+ partnerlinks vars (msgexchangs)? (correlationsset)? (actbpel)+);

targetnamsp : ^(TARGNAMSPDEF Ponct ID Ponct ID Ponct);

supjoinFail : ^(SUPJOINFAILDEF Ponct ID Ponct);

xmlns : ^(TARGNAMSPPDEF Ponct ID Ponct ID Ponct ID Ponct);

importact : ^(IMPODEF namesplocat importType);

namesplocat : ^(NAMSPLOCATIONDEF location namespace) | ^(NAMSPLOCATIONDEF namespace location) ;

location :^(LOCDEF Ponct ID Ponct) ;

namespace : ^(NAMESPDEF Ponct ID Ponct ID Ponct);

importType : ^(IMPOTYPDEF Ponct ID Ponct ID Ponct);

msgexchangs : ^(MSGECHANGSDEF (msgexchang)+);

msgexchang : ^(MSGECHANGDEF name);

correlations : ^(CORRS (correlation)+);

correlation : ^(CORREF name properties initiate);

initiate : ^(INITIATDEF Ponct ID Ponct);

correlationsset : ^(CORRSSET (correlationset)+);

correlationset : ^(CORSETDEF name);

partnerlinks : ^(PARTNERLINKS (partnerLink)+);

partnerLink : ^(PARTNERLINKDEF name partlinktyp (myrole)? (partnerRole)?);

partlinktyp : ^(PLTDEF Ponct ID Ponct ID Ponct);

myrole : ^(MYROLEDEF Ponct ID Ponct);

partnerRole : ^(PARTROLEEDEF Ponct ID Ponct);

vars : ^(VARSDEF (var)+);

var : ^(VARIABLEDEF name messagetype (initialization)?);

messagetype : ^(MSGTYPDEF Ponct ID Ponct ID Ponct);

initialization : ^(INITIALIZATDEF partlink endpointref);

partlink : ^(PLDEF Ponct ID Ponct);

endpointref : ^(ENDPOINTREFDEF Ponct ID Ponct);

actbpel : ^(ACTBPELDEF invoke) | ^(ACTBPELDEF receive) | ^(ACTBPELDEF reply) |
^(ACTBPELDEF assign) | ^(ACTBPELDEF whileact) | ^(ACTBPELDEF repeatUntil) |
^(ACTBPELDEF ifcond) | ^(ACTBPELDEF flow) | ^(ACTBPELDEF pick) |
^(ACTBPELDEF sequence) | ^(ACTBPELDEF scopeact) | ^(ACTBPELDEF exitact) |
^(ACTBPELDEF throwact) | ^(ACTBPELDEF rethrow)| ^(ACTBPELDEF compensate) |
^(ACTBPELDEF foreach) | ^(ACTBPELDEF waitact) | ^(ACTBPELDEF emptyact) |
^(ACTBPELDEF opacact) | ^(ACTBPELDEF validateact);

Basic activities

receive: ^(RECEIVEDEF name partlink porttypoperat variable (createinstance)?
(messageexchange)? (correlations)? (documentation)?);

porttypoperat : ^(PORTOPDEF porttype operation) | ^(PORTOPDEF operation porttype) ;

createinstance : ^(CREATINSTDEF Ponct ID Ponct);

messageexchange : ^(MSGEXCHDEF Ponct ID Ponct);

porttype : ^(PORTTYPDEF Ponct ID Ponct ID Ponct);

operation : ^(OPERATIONDEF Ponct ID Ponct);

invoke : ^(INVOKDEF name partlink operation porttype inputvariable (outvariable)?
(documentation)?);

inputvariable : ^(INPUTVARDEF Ponct ID Ponct);

outvariable : ^(OUTVARDEF Ponct ID Ponct);

assign : ^(ASSIGNDEF validate name (copy)+);

copy : ^(COPYDEF from to);

from : ^(FROMDEF (varpartfr)? (valcontentto)?);

to : ^(TODEF (varpartto)? (valcontentto)?);

varpartfr : ^(VARPARTDEF variable part) | ^(VARPARTDEF part variable) | ^(VARPARTDEF expr) | ^(VARPARTDEF variable) | ^(VARPARTDEF partlink (endpointref)?);

varpartto : ^(VARPARTTODEF variable part) | ^(VARPARTTODEF part variable) | ^(VARPARTTODEF variable) | ^(VARPARTTODEF partlink (endpointref)?);

valcontentto : ^(VALCONTENTTODEF expr) | ^(VALCONTENTTODEF (idponct)+) | ^(VALCONTENTTODEF query);

query : ^(QUERY2DEF ID Ponct ID);

validate : ^(VALIDDEF Ponct ID Ponct);

variable : ^(VARIABLEPDEF Ponct ID Ponct);

part : ^(PARTASSDEF Ponct ID Ponct);

expr : ^(EXPRDEF (idponct)+);

reply : ^(REPDEF name partlink porttypoperat variable (faultname)? (messageexchange)? (documentation)? (correlations)?);

faultname : ^(FAULTNAMEDEF Ponct ID Ponct ID Ponct);

condition : ^(CONDDEF (expressionLanguage)? valconds);

valcond : ^(VALCONDDEF ID Ponct ID (opeponct)+ ID Ponct ID) | ^(VALCONDDEF ID Ponct ID (opeponct)+ ID) | ^(VALCONDDEF ID (opeponct)+ ID) | ^(VALCONDDEF ID (opeponct)+ ID Ponct ID) | ^(VALCONDDEF ID Ponct Ponct ID Ponct) | ^(VALCONDDEF ID Ponct ID Ponct Ponct ID Ponct) ;

idponct : ^(IDPONCTDEF ID (opeponct (ID)?)*) ;

opeponct : ^(OPPONCTDEF OPE) | ^(OPPONCTDEF Ponct) ;

valconds: ^(VALCONDSDEF valcond (valcond)*) ;

expressionLanguage : ^(EXPRESSLANGDEF Ponct ID Ponct);

Structured activities

repeatUntil : ^(REPEATDEF name (actbpel)* condition (documentation)?);

whileact : ^(WHILEACTDEF name condition (actbpel)*);

ifcond : ^(IFDEF name condition (actbpel)+ (elseifcond)* (elsecond)?);

elsecond : ^(ELSEDEF (actbpel)+);

elseifcond : ^(ELSEIFDEF condition (actbpel)+);

sequence : ^(SEQDEF (name)? (actbpel)*);

flow : ^(FLOWDEF name OPE (actbpelfl)+);

waitact: ^(WAITACTDEF name foralarm);

emptyact: ^(EMPTYACTDEF name);

opacact: ^(OPACACTDEF name);

validateact: ^(VALIDACTDEF name variablesact);

variablesact : ^(VARACTVALIDDEF Ponct (ID)+ Ponct);

foreach : ^(FOREACHDEF parallel countername name (xmlns)? startcountervalue finalcountervalue (complcond)? scopeact (documentation)?);

complcond : ^(COMPLCONDDEF branch);

branch : ^(BRANCHDEF succbranchonly (idponct)+);

parallel: ^(PARALLELDEF Ponct ID Ponct);

countername : ^(COUNTERNAMEDEF Ponct ID Ponct);

expridcount : ^(EXPRIDDEF ID);

startcountervalue : ^(SATRTCOUNTERVALUEDEF endstartcountvalue);

endstartcountvalue : ^(ENDSTARTCOUNTDEF ID) | ^(ENDSTARTCOUNTDEF expridcount);

finalcountervalue : (FINALCOUNTERVALUEDEF endfinalcountvalue);

endfinalcountvalue : ^(ENDFINALCOUNTDEF ID) | ^(ENDFINALCOUNTDEF expridcount);

succbranchonly : ^(SCUCBRANCHDEF Ponct ID Ponct);

pick : ^(PICKDEF name createinstance (onmsg)+ (onalarm)? (documentation)?);

onmsg : ^(ONMSGDEF partlink operation porttype variable (actbpel)+);

onalarm : ^(ONALARMDEF scopeact foralarm);

foralarm : ^(FORALARMDEF (ID)+);

scopeact : ^(SCOPEDEF (name)? (isolated)? (partnerlinks)? (vars)? (correlationsset)?
(msgexchangs)? (actbpel)+ (faulthandler)? (compensatehandl)? (terminhandl)?
(eventhandler)? (documentation)?);

scopeactforeach : ^(SCOPEFOREACHDEF (name)? (isolated)? (partnerlinks)? (vars)?
(correlationsset)? (msgexchangs)? (actbpelfl)+ (faulthandler)?
(compensatehandl)? (terminhandl)? (eventhandler)? (documentation)?);

isolated : ^(ISOLATEDDEF Ponct ID Ponct);

Faults activities

faulthandler : ^(FAULTHANDLDEF (catchact)+ (catchactall)? (documentation)?);

catchact : ^(CATCHDEF (faultname)? (faultvariable)? (actbpel)* (documentation)?);

catchactall : ^(CATCHALLDEF (actbpel)* (documentation)?);

faultvariable : ^(FAULTVARDEF Ponct ID Ponct);

compensatehandl : ^(COMPENSHANDLDEF (actbpel)* (documentation)?);

terminhandl : ^(TERMINHANDLDEF (actbpel)* (documentation)?);

eventhandler : ^(EVENTHANDLDEF (onevent)+ (onalarm)? (documentation)?);

onevent : ^(ONEVENTDEF ID partlink operation (messageexchange)? (correlationsset)?
scopeact (documentation)?);

exitact : ^(EXITDEF name (documentation)?);

throwact: ^(THROWDEF name faultname faultvariable (documentation)?);

rethrow : ^(RETHROWDEF name (documentation)?);

compensate : ^(COMPENSATEDEF name (documentation)?);

compensatescope : ^(COMPENSATEDEF name (documentation)?);

documentation : ^(DOCDEF (source)? (language)? (expr)?);

source : ^(SRCDEF Ponct ID Ponct);

language : ^(LANGDEF ID Ponct);

name : ^(NAMEDEF Ponct ID Ponct);