

SPECIFICATION DES TESTS FONCTIONNELS DE L'EDITEUR DE VSL

(VSL editor for Papyrus - Functional testing)

DEPARTEMENT / SERVICE	NUMERO DOCUMENT / DOCUMENT NUMBER	PAGE
LES / STI	61565273 305 3	1/11
		- REVISION

SUIVI DES EVOLUTIONS

	- le : 30/10/07	A le :	B le :	C le :	D le :
Etabli par Signature	F. Nizou				
Approuvé par Signature					

Indices de révision	Modifications
A	
B	
C	
D	

TABLE DES MATIERES

Pages

1. OBJECT.....	3
2. FONCTIONNELLE.....	4
2.1. GUIDE DE LECTURE	4
2.2. LITERAL VALUE.....	4
2.3. COMPOSITE VALUE.....	7
2.4. EXPRESSION.....	8
2.5. TIME EXPRESSION	9
2.6. REGLES NON RECONNUES	11
2.7. RESULTATS	11
3. OPERATIONNELLE	11

1. OBJECT

Ce document est la spécification des tests fonctionnels pour l'éditeur de VSL. Il se présente sous la forme d'un tableau dont la première colonne est le test à jouer. Le remplissage de ce tableau fournit une évaluation quantifiée de l'état de l'éditeur et assure au moins la non régression.

La première partie spécifie les tests fonctionnels (pour chaque construction du langage) la deuxième partie spécifie les tests opérations.

	date	24/10/2007
Outil	Papyrus V7	
Version	V1	

2. FONCTIONNELLE

2.1. Guide de lecture

Cette partie est un ensemble d'expression à soumettre à l'éditeur. Il est découpé en tableau pour chaque règle ou construction du langage.

La colonne VSL contient l'expression à tester, la colonne Err/Val spécifie si c'est un test de validation ou d'erreur (i.e que le parseur devrait valider l'expression ou signaler une erreur). La partie commentaire ajoute des informations pour jouer le test ou sur ce que le test évalue. La partie OK/KO est à remplir à l'issu du test et la partie Descr doit spécifier la raison de l'échec d'un test.

Règle					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)

2.2. Literal value

Numbers					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
1	10	Val	Integer	ok	
2	-10	Val	Signed Integer	ok	
3	10.1	Err	Integer	ok	
4	Ox1A2F	Val	Integer Hexa	ok	

5	Ob10101	Val	Integer Binaire	ok	
6	*	Val	Unlimited Natural	ok	
7	10	Val	Unlimited Natural	ok	
8	-10	Err	Unlimited Natural	ok	
9	1.1	Val	Real	ok	
10	-1.1	Val	Signed Real	ok	
11	12.1E2	Val	Scientific Real	ok	
12	12.1E-1	Val	Scientific Real	ok	
13	12.1E+1	Val	Scientific Real	ok	
14	12	Err	Real	ok	

Enumeration Specification					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
15	s	Val	TimeUnitKind	ok	
16	Rouge	Val	FeuEnum (Rouge, Orange, Vert) définit dans le modèle de l'utilisateur	ok	
17	Rouge	Err	TimeUnitKind	ok	
18	Luminous	Err	TimeUnitKind	ok	

Boolean					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
19	true	Val		ok	
20	false	Val		ok	
21	plop	Err		ok	

String					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
22	'luminou'	Val	String	ok	
23	'lumi\nou'	Val	String avec caractères protégés	ok	
24	'lumin\nou'	Val	String avec caractères protégés	ko	Règle non envisagée dans le parseur
25	1.2	Err	Mauvais typage	ok	

<i>Null</i>					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
26	null	Val	String	ok	
27	null	Val	Integer	ok	

<i>Default</i>					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
28	-	Val	Integer	ok	
29	-	Val	String	ok	

<i>DateTime</i>					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
30	#Mon#	Val	Day String	ok	
31	#Sun#	Val	Day String	ok	
32	#Lum#	Err	Day String	ok	
33	#10 :52 :12#	Val	Time String	ok	
34	#10 :52 :12:29#	Val	Time String avec centisec	ok	
35	#25 :52 :12:29#	Err	Time String avec heure incorrect	ok	
36	#10 :60:12:29#	Err	Time String avec min incorrect	ok	
37	#10 :52 :60:29#	Err	Time String avec sec incorrect	ok	
38	#10 :52 :12:100#	Err	Time String avec centisec incorrect	ok	
39	#2000/11/19#	Val	Date String	ok	
40	#10000/11/19#	Err	Date String avec année incorrect	ok	
41	#2000/13/19#	Err	Date String avec mois incorrect	ok	
42	#2000/11/32#	Err	Date String avec jour du moi incorrect	ok	
43	#10 :52 :12 Mon#	Val	Time + day	ok	
44	#10 :52 :12 2000/11/19#	Val	Date + time	ok	
45	#2000/11/19 Mon#	Val	Date + day	ok	
46	#10 :52 :12 2000/11/19 Mon#	Val	Date + time + day	ok	

2.3. Composite value

<i>Interval</i>					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
47	[1..2]	Val	IntegerInterval	ok	
48	[1..2[Val	IntegerInterval	ok	
49]1..2]	Val	IntegerInterval	ok	
50]1..2[Val	IntegerInterval	ok	
51	[1.1..1.2]	Val	RealInterval	ok	
52	[1..1.2]	Err	IntegerInterval (homogénéité)	ok	
53	[1.1..3]	Err	IntegerInterval (homogénéité)	ok	
54	[1+1..3]	Val	IntegerInterval (homogénéité)	ko	Le parseur essaye de résoudre 1..3 comme opérande de +. Des parenthèses devraient résoudre le problème.
55	[#Mon#..#Sun#]	Val	DateTimeInterval (User Datatype)	ok	
56	[1..2]	Err	RealInterval (type check)	ok	

<i>Collection</i>					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
57	{1, 2, 3}	Val	Integer Vector	ok	
58	{}	Val	Integer Vector (Empty)	ok	
59	{1.1, 1, 2}	Err	Integer Vector (homogénéité)	ok	
60	{{1,2}, {1, 2}}	Val	Matrix	ok	
61	{1.1, 1.2}	Err	Integer Vector (Type check)	ok	

<i>Tuple</i>					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
62	(val=1, statQ=min, dir=incr)	Val	NFP_Integer	ok	
63	(val=1.1, statQ=min, dir=est)	Err	NFP_Real	ok	
64	(val=1.1, todo=2)	Err	NFP_Integer	ok	
65	(1.2, s)	Val	NFP_Duration	ok	

66	(min=1, max=2, avg=1.5)	Val	User type	ok	
67	(period=(1.1, s), jitter=(2.1, s))	Val	PeriodicPattern (tuple dans tuple)	ok	

Choice					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
68	a(1)	Val	User choice (a(Int) b(boolean))	ok	
69	a(true)	Err	Erreur de type	ok	
70	luminou(1)	Err	Pas de choix correspondant	ok	
71	closed(population=(value=1))	Val	ArrivalPattern	ok	

2.4. Expression

Variable					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
72	In\$a :Integer			ok	
73	out\$a :Integer			ok	
74	Inout\$a :Integer			ok	
75	In\$a :Integer = 1			ok	
76	In\$a :Integer = 1.1	Err		ok	
77	In\$a :PLOP	Err		ok	
78	a		Variable globale dans une VSL	ok	
79	p1.a		Variable dans le contexte p1	ok	
80	a	Val	Variable dans le contexte p1 lorsque l'on est dans ce contexte	ok	
81	a	Err	Variable dans le contexte p1 lorsque l'on est pas dans ce contexte	ok	
82	MyVar	err	Variable inexistante	ok	
83	b	Val	Property b taggée <<Var>>	ok	
84	p1.b	Val	Property b taggée <<Var>> dans le contexte p1	ok	
85	a	Err	Quand type attendu est Boolean	ok	

Operation					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
86	1 + 1			ok	
87	1 + a			ok	
88	a + 1			ok	
89	1 + 1.1	Err	+ non définie avec un Real	ok	
90	'aaa'.concat('b')			ok	
91	1 + 1	Err	Quand le type attendu est Real	ok	
92	1.plop(2)	Err	Plop n'existe pas	ok	

Property					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
93	p1.A.p			ok	
94	p1.A.p	Err	Quand le type attendu n'est pas celui de p	ok	
95	p	Val	Quand on est dans le context de A	ok	

2.5. Time Expression

Instant					
N°	VSL	Err/ val	commentaires	OK/ KO	Descr erreur (si erreur)
96	instantObs	Val	Quand le type attendu = NFP_DateTime	ok	
97	instantObs	Err	Quand le type attendu != NFP_DateTime	ok	
98	InstantObs[1]	Val	Avec Occur Index	ok	
99	InstantObs[1.1]	Err	Avec Occur Index avec mauvais typage	ok	
100	InstantObs[1 + 1]	Val	Avec Occur Index en expression	ok	
101	InstantObs when true	Val	Avec condition	ok	
102	InstantObs when 1 == 1	Val	Avec condition en expression	ok	
103	InstantObs when 1	Err	Avec condition mal typée	ok	
104	InstantObs[1] when true	Val	Avec Occur Index et expression	ok	
105	(InstantObs + durationObs)	Val	Sous forme duration + instant	ok	
106	(InstantObs + InstantObs)	Err	Mauvaise forme	ok	

107	(durationObs[1] + InstantObs[1])	Val	Sous forme duration + instant avec occur Index	ok	
108	(durationObs[1] when true + InstantObs[1] when true)	Val	Sous forme duration + instant avec occur Index et condition	ko	Le parseur considère true – InstantObs .. comme la condition. Des parenthèses devraient résoudre le problème.

Duration					
N°	VSL	Err/val	commentaires	OK/KO	Descr erreur (si erreur)
109	durationObs	Val	Quand le type attendu = NFP_Duration	ok	
110	durationObs	Val	Quand le type attendu != NFP_Duration	ok	
111	durationObs [1]	Val	Avec Occur Index	ok	
112	durationObs [1.1]	Err	Avec Occur Index avec mauvais typage	ok	
113	durationObs [1 + 1]	Val	Avec Occur Index en expression	ok	
114	durationObs when true	Val	Avec condition	ok	
115	durationObs when 1 == 1	Val	Avec condition en expression	ok	
116	durationObs when 1	Err	Avec condition mal typée	ok	
117	durationObs [1] when true	Val	Avec Occur Index et expression	ok	
118	(InstantObs - InstantObs)	Val	Sous forme duration + instant	ok	
119	(InstantObs [1] - InstantObs[1])	Val	Sous forme duration + instant avec occur Index	ok	
120	(InstantObs [1] when true - InstantObs[1] when true)	Val	Sous forme duration + instant avec occur Index et condition	ko	Le parseur considère true – InstantObs .. comme la condition. Des parenthèses devraient résoudre le problème.

Jitter					
N°	VSL	Err/val	commentaires	OK/KO	Descr erreur (si erreur)
121	Jitter(instantObs)	Val	Jitter	ok	
122	Jitter(durationObs)	Err		ok	
123	Jitter(instantObs)	Err	Quand attendu != NFP_Duration	ok	
124	Jitter(instantObs, instantObs)	Val	Avec instant, instant	ok	
125	Jitter(instantObs[1])	Val	Avec instant et occur index	ok	
126	Jitter(instantObs when true)	Val	Avec instant et cond	ok	

2.6. Règles non reconnues

- TimeInterval
- Choice avec choix non mentionné
- Variable avec direction non mentionnée
- Parenthèses optionnelles sur les operationCallExpression sans paramètre
- Notation normale pour les opérateurs infixe (1.+(2))

2.7. Résultats

122/ 126
97%

3. OPERATIONNELLE

Cette partie test l'opérationnalisation de l'éditeur dans l'outil. On doit y retrouver tout les éléments UML ou l'on doit pouvoir appeler l'éditeur. Dans le cas ou c'est éléments apparaissent plusieurs fois de le GUI de l'outil, on renseigne un test pour chaque.

Le test est « OK » si l'on peut ouvrir l'éditeur de VSL à partir de l'élément dans le GUI, que le type attendu est correct, que l'édition fonctionne normalement et que l'enregistrement est effectué à la validation. Si une expression VSL était déjà renseignée, l'éditeur contient déjà cette expression.

La liste suivante est la liste attendue et ne constitue en rien la liste exhaustive des éléments dont la valeur peut être spécifiée par une expression VSL.

Object	OK/KO
Contrainte TreeView	Non implémenté
Contrainte graphique	OK
Slot TreeView	Non implémenté
Slot Graphique	OK
Propriété Treeview (default value)	Non implémenté
Propriété Graphique (default value)	OK
Propriété de stéréotype	OK