



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Μορφότυπα κινητού περιεχομένου σε συστήματα DVB

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ελένη Α. Βασιλείου

Επιβλέπων: Μ.Ε.Θεολόγου
Καθηγητής Ε.Μ.Π

Αθήνα, Νοέμβριος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Μορφότυπα κινητού περιεχομένου σε συστήματα DVB

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ελένη Α. Βασιλείου

Επιβλέπων: Μ.Ε.Θεολόγου
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12/02/2008

.....
Μ.Ε.Θεολόγου
Καθηγητής

.....
Ε.Δ.Συκάς
Καθηγητής

.....
Γ.Ι.Στασινόπουλος
Καθηγητής

Αθήνα, Νοέμβριος 2008

.....
Ελένη Α. Βασιλείου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Βασιλείου Α. Ελένη, 2008

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΛΗΨΗ

Η PCF είναι ένα διαδραστικό σχήμα περιγραφής τηλεοπτικών (iTV) υπηρεσιών σχεδιασμένο να είναι ανεξάρτητο πλατφορμας. Μια υπηρεσία που περιγράφεται χρησιμοποιώντας την PCF μπορεί να μετατραπεί αυτόματα σε ένα κατάλληλο σχήμα για οποιαδήποτε πλατφόρμα στόχο.

Η PCF επιτρέπει μια υπηρεσία iTV να περιγραφεί χρησιμοποιώντας μια υψηλού επιπέδου δηλωτική σύνταξη. Το μορφότυπο περιγράφει την προοριζόμενη εμπειρία του θεατή χωρίς να ορίσει ακριβώς πώς πρέπει να δοθεί σε μια πλατφόρμα στόχο. Αυτό σημαίνει ότι οποιαδήποτε αυτόματη διαδικασία μετατροπής έχει επαρκή ελευθερία να μετασχηματίσει την περιγραφή στο πρότυπο εκτέλεσης που απαιτείται από την πλατφόρμα στόχο(target platform). Για να επιτραπεί σε μια υπηρεσία να περιγραφεί πιο αποτελεσματικά και με τρόπο που να μεταβιβάζει κάλλιστα την πρόθεση του συντάκτη, το μορφότυπο χρησιμοποιεί ένα μηχανισμό παραπομπής. Αυτό επιτρέπει την περιγραφή υπηρεσιών να είναι ελαστικά διαχωρισμένες και δίνει στον κωδικοποιητή ιδιαίτερη ελευθερία να αποφασίσει τον καλύτερο τρόπο πακετοποίησης της υπηρεσίας ώστε να ταιριάζει στην πλατφόρμα στόχο.

Η μορφή της PCF απεικονίζει το γεγονός ότι δεν προορίζεται ως πραγματικό σχήμα μετάδοσης, αλλά μάλλον ως ενδιάμεσο σχήμα που θα χρειαστεί μετατροπή σε μια συγκεκριμένη πλατφόρμα παρουσίασης πριν από τη μετάδοση.

Λέξεις Κλειδιά

PCF, τηλεόραση σε κινητά τηλέφωνα, Digital Video Broadcasting – Handheld, ψηφιακή τηλεόραση, DTV, DVB, DVB-T, DVB-H, MFN, SFN, ETSI, MPE-FEC, TPS, ευρυεκπομπή, μονοεκπομπή, Time slicing, κινητικότητα, MPE-FEC, 4K mode, DVB-H signaling, IP Datacast, In-depth Interleaving, απαιτήσεις του συστήματος, φορητότητα, μεταπομπή, διαδραστικότητα, φυσικό στρώμα, στρώμα ζευξης δεδομένων, προδιαγραφές, δομικά στοιχεία, IP Datacast, δίκτυα DVB- H

ABSTRACT

The PCF is an interactive television (iTV) service description format designed to be platform independent. A service described using the PCF can be automatically converted into a format suitable for any target platform.

The PCF accomplishes this by allowing the iTV service to be described using a high-level declarative syntax. The format describes the intended viewer experience without prescribing exactly how it should be rendered on a target platform. This means that any automatic conversion process has sufficient freedom to transform the description into the execution model required by the target platform. To enable a service to be described most efficiently, and in a way that best conveys the author's intent, the format uses a referencing mechanism. This allows the service description to be flexibly partitioned and gives the transcoder considerable freedom to decide the best way of packaging the service optimally to suit the target platform.

The form of the PCF reflects the fact that it is not intended as an actual transmission format, but rather as an intermediate format that will need conversion to a platform-specific representation prior to transmission.

Key Words

PCF, Mobile TV, Digital Video Broadcasting – Handheld, Digital Television, DTV, DVB , DVB-T, DVB-H, MFN, SFN, ETSI , MPE-FEC ,TPS , Unicast, Broadcast, Time slicing, Handover support, MPE-FEC, 4K mode, DVB-H signaling, IP Datacast, In-depth Interleaving, Requirements of the system, portable,handover,interactivity, physical layer, link layer, standard,component, IP Datacast, DVB- H network,

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Ε.Μ.Π. κ. Μιχαήλ Θεολόγου για τη δυνατότητα που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα καθώς και για την εμπιστοσύνη που μου έδειξε.

Επιπλέον, θα ήθελα να ευχαριστήσω το διδάκτορα κ. Δημήτρη Αξιώτη για την άψογη συνεργασία καθώς επίσης και για τη συμπαράστασή του κατά την εκπόνηση της παρούσας διπλωματικής. Είμαι ιδιαίτερα ευγνώμων για το χρόνο που διέθεσε αλλά και για τις πολύτιμες υποδείξεις και συμβουλές του που συνέβαλαν ουσιαστικά στην περάτωση της εργασίας.

Η διπλωματική αυτή αφιερώνεται στους γονείς μου και στον αδελφό μου, οι οποίοι με στήριξαν καθόλη τη διάρκεια των σπουδών μου.

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Εικόνα 1 . Αναλογική και Ψηφιακή μετάδοση.....	25
Εικόνα 2. Μονοεκπομπή (unicast).....	28
Εικόνα 3 . Ευρυεκπομπή (broadcast)	28
Εικόνα 4. Στοιβά πρωτοκόλλων στην μεταφορά δεδομένων.....	33
Εικόνα 5. Μετάδοση υπηρεσιών IP μέσω DVB-H.....	34
Εικόνα 6. Περιγραφή της δομής ενός DVB-H δέκτη	34
Εικόνα 7 . Περιοχή κάλυψης των 2K ,4K και 8K modes.....	38
Εικόνα 8 . Μέθοδος in-depth interleaving.....	39
Εικόνα 9. Πολυπλεξία και παράλληλη μετάδοση υπηρεσιών.....	41
Εικόνα 10 . Περιγραφή της τεχνικής time slicing.....	42
Εικόνα 11. Υπηρεσίες DVB-H και DVB-T	43
Εικόνα 12. Μέθοδος Delta-T.....	12
Εικόνα 13. Στοιβά πρωτοκόλλων του DVB-H.....	45
Εικόνα 14. Σταδια προστασίας λαθων κατά την μετάδοση δεδομένων στο DVB-H.....	46
Εικόνα 15. Δομή ενός πλαισίου MPE-FEC.....	47
Εικόνα 16. Τιμές κάποιων παραμέτρων που χρησιμοποιούνται στα προτυπα DVB-T & DVB-H.....	50
Εικόνα 17. Κατανάλωση ισχύος ανάμεσα σε δέκτες DVB-T και DVB-H.....	51
Εικόνα 18 .Κοινόχρηστο δίκτυο με το DVB-T.....	52
Εικόνα 19. Ιεραρχικό δίκτυο DVB-H.....	53

Εικόνα 20. Αποκλειστικό δίκτυο DVB-H.....	54
Εικόνα 21. Σύστημα IPDC.....	56
Εικόνα 22. Μετακινώντας το δομικό στοιχείο ELC.....	63
Εικόνα 23. Άξονες μιας περιγραφής PCF.....	78
Εικόνα 24. Μοντέλο των τύπων δεδομένων.....	79
Εικόνα 25. Μοντέλο αντικειμένων των δεδομένων octet.....	98
Εικόνα 26. Σύνθετοι τύποι δεδομένων.....	108
Εικόνα 27. Τα δομικά αντικείμενα της PCF.....	111
Εικόνα 28. Ένα παράδειγμα ιεραρχίας των δομικών στοιχείων της PCF.....	116
Εικόνα 29. Μοντέλο πλοήγησης και παραπομπής της PCF.....	116
Εικόνα 30. Μοντέλο προδιαγραφών των δομικών στοιχείων.....	127
Εικόνα 31 Μοντέλο προδιαγραφών δομικών στοιχείων στο οποίο παρουσιάζονται ομάδες.....	130
Εικόνα 32. Προδιαγραφές αντικειμένων απαρίθμησης.....	132
Εικόνα 33. Μοντέλο γεγονότων των δομικών στοιχείων.....	133
Εικόνα 34. Η σχέση μεταξύ των αντικειμένων που ανήκουν στο μοντέλο προδιαγραφών και στο μοντέλο υποστασιοποίησης.....	139
Εικόνα 35. Συμπεριφορά των οπτικών δομικών στοιχείων.....	148
Εικόνα 36. Μοντέλο αντικειμένων των δομικών στοιχείων.....	152
Εικόνα 37 . Ιεραρχία του δομικού στοιχείου explicit layout.....	157
Εικόνα 38. Σχετικός προσδιορισμός θέσης των θυγατρικών δομικών στοιχείων explicit layout.....	157
Εικόνα 39. Μετακίνηση θυγατρικών δομικών στοιχείων μέσα σε ένα ELC.....	158
Εικόνα 40 . Σωρός εμφάνισης μετά την τοποθέτηση των θυγατρικών δομικών στοιχείων explicit layout.....	159

Εικόνα 41. Επικαλυπτόμενα Explicit Layout Containers.....	160
Εικόνα 42. Ιεραρχία των δομικών στοιχείων διαγράμματα ροής.....	161
Εικόνα 43. Ιδιότητες του layout box	162
Εικόνα 44. Μοντέλο αντικειμένων διαγραμμάτων ροής.....	163
Εικόνα 45. Ένα σχεδιάγραμμα το οποίο αποτελείται block-level από στοιχεία.....	165
Εικόνα 46. Ένα σχεδιάγραμμα το οποίο αποτελείται από ανώνυμα block-boxes.....	166
Εικόνα 47. Ένα σχεδιάγραμμα το οποίο αποτελείται από στοιχεία inline-level.....	167
Εικόνα 48. Κάθετη ευθυγράμμιση κελιών.....	176
Εικόνα 49. Σχεδιαστικό διάγραμμα πλαισίων της PCF.....	177
Εικόνα 50. Η ιδιότητα referenceScreenMapping έχει τεθεί ως "display-anamorphic".....	180
Εικόνα 51. Η ιδιότητα referenceScreenMapping έχει τεθεί ως "pixel".....	180
Εικόνα 52. Ταξινόμηση στοιχείων συγκεκριμένου τύπου.....	182
Εικόνα 53. Μοντέλο επίδειξης επιπέδων σωρού.....	182
Εικόνα 54. Συμπεριφορά του σωρού εμφάνισης	183
Εικόνα 55. Το μοντέλο αντικειμένων παρουσιάζει πως συνδέονται τα γεγονότα με άλλες πτυχές της υπηρεσίας.....	188
Εικόνα 56. Αφηρημένη ιεραρχία των δομικών στοιχείων.....	189
Εικόνα 57. Επίπεδα δομικών στοιχείων όπου ένα γεγονός στοχεύει στο δομικό στοιχείο service και στα δομικά στοιχεία των ενεργών scene.....	190
Εικόνα 58. Ιεραρχία των δομικών στοιχείων μέσα σε ένα παράδειγμα scene.....	191
Εικόνα 59. Μετάδοση ενός γεγονότος από ένα focused component σε ένα Service component.....	192
Εικόνα 60. Μετάδοση ενός γεγονότος αφού έχει φθάσει στο component scene	193
Εικόνα 61. Μετάδοση ενός γεγονότος από ένα component που έχει δημιουργηθεί σε ένα component Service.....	195

Εικόνα 62. Το μοντέλο της γλώσσας δράσης	198
Εικόνα 63. Καταστάσεις του αντικειμένου action language.....	202
Εικόνα 64. Το μοντέλο αντικειμένων για την Statemachine.....	212
Εικόνα 65. Το μοντέλο των αντικειμένων transition και onevent	213
Εικόνα 66. Αναπαράσταση ενός αντικειμένου transition σε ένα statechart	216
Εικόνα 67. Αναπαράσταση ενός αντικειμένου top state σε ένα statechart	218
Εικόνα 68 .Το αντικείμενο top state είναι σχεδιασμένο.....	218
Εικόνα 69. Το αντικείμενο top state δεν έχει σχεδιαστεί αλλά υπονοείται η ύπαρξή του.....	219
Εικόνα 70. Αναπαράσταση ενός αντικειμένου initial state σε ένα statechart.....	220
Εικόνα 71. Αναπαράσταση ενός αντικειμένου final state σε ένα statechart.....	221
Εικόνα 72 . Αναπαράσταση ενός αντικειμένου history state σε ένα statechart.....	222
Εικόνα 73. Όλες οι επισκέψεις στο αντικείμενο enclosingstate εκμεταλλεύονται την συμπεριφορά του αντικειμένου history state.....	223
Εικόνα 74. Το αντικείμενο history state μπορεί να χρησιμοποιηθεί χωρίς το συνοδευτικό αντικείμενο initial state.....	223
Εικόνα 75. Τα αντικείμενα history και initial state έχουν και τα δύο μία εξωτερική μετάβαση.....	224
Εικόνα 76. Αναπαράσταση ενός αντικειμένου state σε ένα statechart.....	226
Εικόνα 77. Η περιγραφή του αντικειμένου «stateone» περιλαμβάνει μεταβάσεις και θυγατρικά αντικείμενα state.....	227
Εικόνα 78. Φωλιασμένα αντικείμενα states με καταστάσεις εισόδου και εξόδου.....	228
Εικόνα 79. Αναπαράσταση ενός αντικειμένου junction state σε ένα statechart.....	231
Εικόνα 80. Το αντικείμενο junction state χρησιμοποιείται για να απλοποιήσει τις πολλαπλές μεταβάσεις υπό συνθήκες	232
Εικόνα 81. Πολλαπλές μεταβάσεις υπό συνθήκες	233

Εικόνα 82. Αναπαράσταση ενός αντικειμένου junction state σε ένα statechart.....	234
Εικόνα 83 .Το αντικείμενο choice state	235
Εικόνα 84. Μία μοναδική μετάβαση δεν μπορεί να αντικαταστήσει τις μεταβάσεις διαμέσου του αντικειμένου choice state.....	235
Εικόνα 85.Η μετάδοση ενός γεγονότος τερματίζεται στην εγγενή συμπεριφορά στο οποίο έχουμε εστιάσει.	238
Εικόνα 86. Μετάδοση ενός γεγονότος όπου το δομικό στοιχείο στο οποίο έχουμε εστιάσει έχει εγγενή συμπεριφορά	239
Εικόνα 87. Μετάδοση ενός γεγονότος όπου το δομικό στοιχείο στο οποίο έχουμε εστιάσει δεν έχει εγγενή συμπεριφορά	240

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας 1. Πιθανές τιμές των των s48 και s49 bits.....	36
Πίνακας 2. Τιμές παραμέτρων των 2K, 4K και 8K mode	37
Πίνακας 3 . Πίνακας περιγραφής ενός τύπου δεδομένων.....	82
Πίνακας 4. Ο τύπος δεδομένων Boolean	84
Πίνακας 5 . Ο τύπος δεδομένων integer.....	84
Πίνακας 6 . Ο τύπος δεδομένων enumeration	85
Πίνακας 7 .Ο τύπος δεδομένων string	86
Πίνακας 8. Ο τύπος δεδομένων colour.....	87
Πίνακας 9. Ο τύπος δεδομένων currency.....	88
Πίνακας 10. Ο τύπος δεδομένων date	89
Πίνακας 11. Ο τύπος δεδομένων font family.....	90
Πίνακας 12. Ο τύπος δεδομένων fontSize.....	91
Πίνακας 13. Ο τύπος δεδομένων marked up text	92
Πίνακας 14. Ο τύπος δεδομένων name.....	92
Πίνακας 15 . Ο τύπος δεδομένων position.....	93
Πίνακας 16. Ο τύπος δεδομένων proportion	94
Πίνακας 17. Ο τύπος δεδομένων size.....	94
Πίνακας 18 . Ο τύπος δεδομένων time.....	95
Πίνακας 19. Ο τύπος δεδομένων timecode.....	96

Πίνακας 20. MIME τύποι της PCF.....	100
Πίνακας 21. Οπτικά δομικά στοιχεία.....	146
Πίνακας 22. Μη – οπτικά λειτουργικά δομικά στοιχεία	150
Πίνακας 23. Τα δομικά στοιχεία variable και cookie	151

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΑΡΚΤΙΚΟΛΕΞΑ

2.	ΨΗΦΙΑΚΗ ΤΗΛΕΟΡΑΗ ΣΤΗΝ ΕΥΡΩΠΗ	25
2.1	Τι είναι η ψηφιακή τηλεόραση (DTV)	25
2.2	Πλεονεκτήματα της DTV	26
2.3	Mobile TV	27
3.	DVB-H	30
3.1	Εισαγωγή- Ιστορική αναδρομή	30
3.2	Απαιτήσεις του συστήματος	31
3.3	Αρχιτεκτονική του συστήματος DVB-H.....	33
3.3.1	Γενική περιγραφή.....	33
3.3.2	Νέα στοιχεία.....	36
3.3.2.1	Φυσικό Επίπεδο (<i>Physical Layer</i>).....	36
3.3.2.1.1	TPS Signalling Bits.....	36
3.3.2.1.2	4K Mode	37
3.3.2.1.3	In-Depth Interleavers	39
3.3.2.1.4	5MHz Channel Bandwidth.....	40
3.3.2.2	Επίπεδο Ζεύξης Δεδομένων (<i>Link/MAC Layer</i>).....	40
3.3.2.2.1	Time-Slicing	40
3.3.2.2.1.1	Delta - T method.....	43
3.3.2.2.2	MPE-FEC	44
3.4	Θέματα Αποδοσης.	49
3.4.1	Doppler effect	49
3.5	Σύγκριση DVB-H και DVB-T.....	50
4.	DVB-H NETWORKS	
4.1	DVB-T Shared Networks.....	52
4.2	DVB-H Hierarchical Networks.....	52
4.3	Dedicated DVB-H Networks.....	54
4.4	IPDataCasting over DVB-H.....	55

5 . PCF.....	57
5.1 Εισαγωγή	58
5.2 Επισκόπηση μιας περιγραφής υπηρεσίας PCF	58
5.3 Δομικά Στοιχεία	60
5.3.1 Το Δομικό Στοιχείο Υπηρεσία	62
5.3.2 Το Δομικό Στοιχείο Σκηνή	62
5.3.3 Το Δομικό Στοιχείο Σχεδιάγραμμα	62
5.3.3.1 Αναλυτικό Σχεδιάγραμμα	62
5.3.3.2 Σχεδιάγραμμα ροής.....	63
5.3.4 Το Δομικό Στοιχείο Return Path	64
5.3.5 Προσαρμοσμένα δομικά στοιχεία	66
5.4 Περιεχόμενο.....	66
5.5 Συμπεριφορά.....	67
5.5.1 Γεγονότα.....	68
5.5.2 Γλώσσα Δράσης.....	68
5.6 Δομή μιας περιγραφής υπηρεσίας PCF	70
5.6.1 Παραπομπή.....	70
5.6.2 Αντιγραφή.....	71
5.7 Αντιμετώπιση των διαφορών μεταξύ των πλατφόρμων.	72
5.7.1 Βαθμοί	72
5.7.2 Προφίλ.....	73
6 . ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	74
6.1 Εισαγωγή	74
6.1.1 Γλώσσα ισχυρού ελέγχου τύπων (Strong typing)	75
6.1.2 Στατική και ενεργή περιγραφή	76
6.1.3 Παρουσίαση Μιας Υπηρεσίας	76
6.1.4 Μοντέλο παραπομπής	77
6.1.5 Διαμέριση και επαναχρησιμοποίηση δεδομένων	77
6.2 Τύποι Δεδομένων	80
6.2.1 Περιγραφή Τύπων Δεδομένων	80
6.2.1.1 Μοντέλο Τύπων Δεδομένων	80
6.2.1.2 Διάστημα Περιγραφής	81
6.2.1.3 Το αντικείμενο τιμή	82
6.2.2 Πρωταρχικοί τύποι δεδομένων	83
6.2.2.1 Boolean.....	83
6.2.2.2 Ακέραιοι Αριθμοί.....	84
6.2.2.3 Απαρίθμηση.....	84
6.2.2.4 Συμβολοσειρά.....	85
6.2.3 Οι τύποι δεδομένων Core	86
6.2.3.1 Χρώμα.....	86
6.2.3.2 Νόμισμα	87
6.2.3.3 Ημερομηνία	88
6.2.3.4 Ημερομηνία και ώρα	88

6.2.3.5	<i>Οικογένειες γραμματοσειρών</i>	89
6.2.3.6	<i>Μέγεθος γραμματοσειράς</i>	90
6.2.3.7	<i>Μαρκαρισμένο κείμενο</i>	91
6.2.3.8	<i>Όνομα</i>	92
6.2.3.9	<i>Θέση</i>	93
6.2.3.10	<i>Αναλογία</i>	94
6.2.3.11	<i>Μέγεθος</i>	94
6.2.3.12	<i>Ωρα</i>	95
6.2.3.13	<i>Χρονική Αναφορά</i>	94
6.2.3.14	<i>URI</i>	96
6.2.3.15	<i>Κλειδιά του χρήστη</i>	97
6.2.4	<i>Αντικείμενα δεδομένων octet</i>	97
6.2.4.1	<i>Εισαγωγή στα δεδομένα octet</i>	97
6.2.4.2	<i>Μοντέλο των δεδομένων octet</i>	98
6.2.4.3	<i>Στοιχεία περίληψης δεδομένων octet (Octet data containers)</i>	99
6.2.4.3.1	<i>Φορητοί τύποι MIME</i>	99
6.2.4.3.2	<i>Αντικείμενα Meta property</i>	101
6.2.4.3.3	<i>Ενσωματωμένο ακρυπτογραφητο κείμενο δεδομένων</i>	101
6.2.4.3.4	<i>Ενσωματωμένα δυαδικά δεδομένα</i>	101
6.2.4.3.5	<i>Ενσωματωμένα base64 δεδομένα</i>	102
6.2.4.3.6	<i>Ενσωματωμένα δεκαεξαδικά δεδομένα</i>	102
6.2.4.3.7	<i>Ενσωματωμένα εντός εισαγωγικό εκτυπώσιμα δεδομένα</i>	102
6.2.4.3.8	<i>Δεδομένα εξωτερικού σώματος του κώδικα</i>	103
6.2.4.3.9	<i>Αντικείμενα δεδομένων πολλαπλών τύπων</i>	102
6.2.4.4	<i>Τύποι αντικειμένων που αποτελούνται από δεδομένα octet</i>	105
6.2.4.4.1	<i>Αντικείμενα συμβολοσειρών που αποτελούνται από δεδομένα</i>	105
6.2.4.4.2	<i>Μαρκαρισμένο κείμενο και δεδομένα octet</i>	105
6.2.4.4.3	<i>Αντικείμενα εικόνων που αποτελούνται από δεδομένα octet</i>	106
6.2.4.4.4	<i>Αντικείμενα ροών δεδομένων octet</i>	107
6.2.5	<i>Σύνθετοι τύποι</i>	107
6.2.5.1	<i>Σύνθετοι τύποι δεδομένων</i>	107
6.2.5.2	<i>Ο τύπος δεδομένων χαρτογράφηση και το αντικείμενο χαρτογράφηση</i>	108
6.2.5.3	<i>Ο τύπος δεδομένων πίνακας και το αντικείμενο πίνακας</i>	109
6.3	<i>Δομή Περιγραφής Μιας Υπηρεσίας</i>	111
6.3.1	<i>Αντικείμενα Περιγραφής</i>	111
6.3.2	<i>Τα δομικά στοιχεία</i>	111
6.3.3	<i>Το Αντικείμενο Συλλογή</i>	113
6.3.4	<i>Στοιχείο περίληψης της PCF (PCF container)</i>	113
6.3.5	<i>Το Αντικείμενο Σκηνή</i>	113
6.3.6	<i>Το Αντικείμενο Υπηρεσία</i>	114
6.3.7	<i>Κανόνες Αντικειμενοθέτησης</i>	115

6.4	Παραπομπή Και Πλοήγηση	116
6.4.1	Μοντέλο Παραπομπής	116
6.4.2	Τυπεί Παραπομπή.....	117
6.4.2.1	Αντικείμενα παραπομπών της PCF	117
6.4.2.2	Μορφή και ανάλυση μιας διαδρομής παραπομπής	117
6.4.3	Ανάλυση κειμένου	120
6.4.4	Τα αντικείμενα παραπομπής Map	122
6.4.5	Τα αντικείμενα παράμετροι.....	123
6.4.6	Τα αντικείμενα παραπομπών πλοήγησης	125

7. ΓΕΝΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΩΝ ΔΟΜΙΚΩΝ ΣΤΟΙΧΕΙΩΝ

7.1	Επισκόπηση.....	126
7.2	Μοντέλο Προδιαγραφών των δομικών στοιχείων	127
7.2.1	Επισκόπηση.....	127
7.2.2	Καθορισμός Διεπαφών	127
7.2.2.1	Ομάδες	130
7.2.2.3	Προδιαγραφές Απαρίθμησης	132
7.2.2.4	Προδιαγραφές Αντιμετωπίσιμων γεγονότων	133
7.2.2.5	Προδιαγραφές παραγόμενων γεγονότων	134
7.2.2.6	Προδιαγραφές Αντιμετωπίσιμων Ενεργειών	135
7.2.2.7	Προδιαγραφές παραγόμενων λαθών	135
7.2.2.8	Προοριζόμενη εφαρμογή	136
7.2.2.9	Αντικείμενο επισκόπησης	136
7.2.3	Κειμενική Περιγραφή	136
7.2.4	Προδιαγραφές Συμπεριφοράς	137
7.3	Μοντέλο Υποστασιοποίησης δομικών στοιχείων	138
7.3.1	Δομικά στοιχεία	139
7.3.2	Ιδιότητες	140
7.3.3	Διαδοχικές Συνδεδεμένες Ιδιότητες	141
7.3.4	Το δομικό στοιχείο ανοχής υλοποίησης.....	141
7.4	Συμπεριφορά Δομικών Στοιχείων	142
7.4.1	Επισκόπηση συμπεριφοράς	142
7.4.2	Πρόσβαση στις ιδιότητες των δομικών στοιχείων	142
7.4.3	Αντιμετωπίσιμα γεγονότα <i>Handled events</i>	143
7.4.4	Αντιμετωπίσιμες ενέργειες	143
7.4.5	Παραγόμενα Γεγονότα	144
7.4.6	Παραγόμενα λάθη	144
7.4.7	Πεδίο των δομικών στοιχείων	145
7.5	Προσδιορισμός των κατηγοριών των δομικών στοιχείων της PCF	145
7.5.1	Επισκόπηση.....	145
7.5.2	Οπτικά δομικά στοιχεία	146
7.5.3	Μη οπτικά δομικά στοιχεία	150
7.5.3.1	Λειτουργικά δομικά στοιχεία	150
7.5.3.2	Τα δομικά στοιχεία μεταβλητή και <i>cookie</i>	151

7.5.4	Το δομικό στοιχείο <i>Container</i>	152
7.6	Συνήθη δομικά στοιχεία	152
7.7	Τα δομικά στοιχεία <i>Schema</i>	153

8. ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΧΕΔΙΑΓΡΑΜΜΑΤΩΝ

8.1	Εισαγωγή	155
8.2.2	Στοιχεία και χαρακτηριστικά του <i>Explicit layout container</i>	155
8.2	Αναλυτικά Σχεδιαγράμματα	156
8.2.1	Εισαγωγή	156
8.3	Διαγράμματα ροής	160
8.3.1	Εισαγωγή	160
8.3.2	Τα στοιχεία σχεδιαγραμμάτων ροής	160
8.3.3	Το μοντέλο πλαισίων διαγραμμάτων ροής	161
8.3.4	Τύποι πλαισίων των διαγραμμάτων ροής	163
8.3.4.1	Επισκόπηση	163
8.3.4.2	Περιεχόμενα μπλοκ	164
8.3.4.3	Τα στοιχεία <i>Block-level</i>	165
8.3.4.4	<i>Block formatting context</i>	166
8.3.4.5	Τα στοιχεία <i>Inline-level</i>	166
8.3.4.6	Ευθύγραμμο πλαίσιο μορφοποίησης	167
8.3.5	Ιδιότητες Σχεδιαγραμμάτων	169
8.3.5.1	Γενικές ιδιότητες	169
8.3.5.2	Συγκεκριμένες ιδιότητες πλευρών	170
8.4	Κείμενο ροής	171
8.5	Σχεδιάγραμμα πίνακα	171
8.5.1	Εισαγωγή	171
8.5.2	Αλγόριθμοι σχεδιαγραμμάτων πινάκων	171
8.5.2.1	Σταθερο σχεδιάγραμμα πίνακα	172
8.5.2.2	Αυτόματο σχεδιάγραμμα πίνακα	172
8.5.2.3	Αλγόριθμος ύψους πίνακα	174
8.5.2.4	Αλγόριθμος ύψους γραμμών	174
8.5.2.5	Αλγόριθμος ύψους κελιών	174
8.5.2.6	Ευθυγράμμιση περιεχομένου κελιού	176
8.5.2.6.1	Οριζόντια ευθυγράμμιση	176
8.5.3	Σύνορα	177
8.6	Flow layout container components	177
8.7	Μοντέλο οθόνης αναφοράς	179
8.7.1	Η οθόνη αναφοράς	179
8.7.2	Χαρτογράφηση της οθόνης αναφοράς σε μία συσκευή στόχο.	179
8.7.2.1	Η ανάλυση της οθόνης της συσκευής στόχου είναι ίδια με την ανάλυση της οθόνης αναφοράς.	179
8.7.2.2	Η ανάλυση της οθόνης της συσκευής στόχου είναι διαφορετική από την ανάλυση της οθόνης αναφοράς.	179
8.7.2.3	Κλιμάκωση της οθόνης αναφοράς.	180

8.8	Εγγραφή βίντεο και γραφικών.	181
8.9	Μοντέλο εμφάνισης σωρού	182
8.9.1	Αρχικοποίηση του σωρού εμφάνισης	182
8.9.2	Χειρισμός του σωρού εμφάνισης	183
8.10	Επιλογή γραμματοσειράς	184
9.	ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΥΜΠΕΡΙΦΟΡΑΣ	185
9.1	Εισαγωγή.....	185
9.1.1	Ενδογενής συμπεριφορά δομικών στοιχείων	185
9.1.2	Ανεξάρτητη συμπεριφορά	186
9.2	Γεγονότα	186
9.2.1	Run-time event model.....	186
9.3	Μοντέλο μετάδοσης γεγονότων	187
9.3.1	Εισαγωγή.....	187
9.3.1.1	Μοντέλο Αντικειμένων	187
9.3.1.2	Ιεραρχία περίεξης των δομικών στοιχείων	188
9.3.1.3	Μετάδοση Γεγονότων	189.
9.3.2	Γεγονότα του Συστήματος	189
9.3.2.1	Επισκόπηση	189
9.3.2.2	Κανόνες μετάδοσης των γεγονότων του συστήματος	190
9.3.3	Γεγονότα τα οποία εισάγονται από τον χρήστη	191
9.3.3.1	Επισκόπηση	191
9.3.3.2	Έλεγχος ε.τίασης	191
9.3.3.3	Κανόνες μετάδοσης γεγονότων που εισάγονται από τον χρήστη...192	
9.3.4	Γεγονότα δομικών στοιχείων	193
9.3.4.1	Επισκόπηση.....	193
9.3.4.2	Κανόνες μετάδοσης γεγονότων δομικών στοιχείων	194
9.3.5	Γεγονότα Λαθών	195
9.3.5.1	Επίπεδα λαθών κατά τον χρόνο εκτέλεσης και προκαθορισμένες αποκρίσεις.....	196
9.3.5.2	Τύποι λαθών	196
9.4	Γλώσσα Δράσης	197
9.4.1	Εισαγωγή.....	197
9.4.2	Αναπαράσταση και εκτέλεση	197
9.4.3	Έγκυρη γλώσσα δράσης	199
9.4.4	Τύποι δεδομένων και αντικείμενα της γλώσσας δράσης	200
9.4.5	Run-time data mapping.....	200
9.4.5.1	Execution context.....	200
9.4.6	Run-time execution model.....	202
9.4.6.1	Δηλώσεις	202
9.4.6.2	Δήλωση Ανάθεσης	202
9.4.6.3	Δήλωση ορισμού	203
9.4.6.4	Action call statement.....	204
9.4.6.5	Δήλωση Συνθήκης	205

9.4.6.6	Δήλωση βρόχου και έλεγχος βρόχου	206
9.4.6.7	Λάθη εκτέλεσης	206
9.4.7	Εκφράσεις και συνθήκες	207
9.4.7.1	Αποτίμηση	207
9.4.7.2	Αριθμητικοί τελεστές	208
9.4.7.3	Λογικοί τελεστές	208
9.4.7.4	Σχεσιακοί τελεστές	209.
9.5	Συντομεύσεις της γλώσσας δράσεις	210
9.6	Μηχανή καταστάσεων	210
9.6.1	Εισαγωγή.....	210
9.6.1.1	Ορισμός καταστάσεων	211
9.6.1.3	Μοντέλο αντικειμένων	212
9.6.1.4	Transition and onevent object model.....	212
9.6.2	Το αντικείμενο Statemachine.....	213
9.6.3	Το αντικείμενο Transition collection.....	214
9.6.3.1	Το αντικείμενο Transition.....	215
9.6.3.2	Το αντικείμενο Trigger.....	216
9.6.3.3	Το αντικείμενο Guard.....	217
9.6.3.4	Το αντικείμενο Action.....	217
9.6.4	Το αντικείμενο Top state.....	218
9.6.4.1	Το αντικείμενο αρχικής κατάστασης	219
9.6.4.2	Το αντικείμενο τελικής κατάστασης	220
9.6.4.3	Το αντικείμενο History state.....	221
9.6.5	Το αντικείμενο State collection.....	225
9.6.5.1	Το αντικείμενο state	226
9.6.5.1.1	Το αντικείμενο κατάστασης εισόδου	228
9.6.5.1.2	Το αντικείμενο κατάστασης εξόδου	229
9.6.5.1.3	Το αντικείμενο εσωτερικών καταστάσεων	229
9.6.5.2	Το αντικείμενο διακλάδωσης καταστάσεων	230
9.6.5.3	Το αντικείμενο επιλογής κατάστασης	233
9.7	Το αντικείμενο OnEvent – συντόμευση της statemachine	236
9.8	Συμπεριφορά που ορίζεται από τον χρήστη.	238
9.8.1	Κανόνες για την συμπεριφορά που ορίζεται από τον χρήστη.	238
9.8.2	Μετάδοση γεγονότος κατά την οποία εμπλέκεται συμπεριφορά που ορίζεται από τον χρήστη.	238
10.	ΒΙΒΛΙΟΓΡΑΦΙΑ	242

1. ΑΡΤΙΚΟΛΕΞΑ

ATSC	Advanced Television Systems Committee
CDS	Component Definition Syntax
CSS2	Cascading Style Sheets, level 2
CRC	Cyclic Redundancy Check
DTV	Digital Television
DVB	Digital Video Broadcasting
DVB-H	Digital Video Broadcasting for Handhelds
DVB-T	Digital Video Broadcasting – Terrestrial
ELC	Explicit Layout Container
ES	Elementary Stream
ETSI	European Telecommunications Standards Institute
GPRS	General Packet Radio Service
GSM	Global System for Mobile
HTML	HyperText Markup Language
IDFT	Inverse Discrete Fourier Transformation
IP	Internet Protocol
IPDC	IP Datacasting
ISBD	Integrated Services Digital Broadcasting
JPEG	Joint Photographic Experts Group
MFNs	Multi Frequency Networks
MHP	Multimedia Home Platform

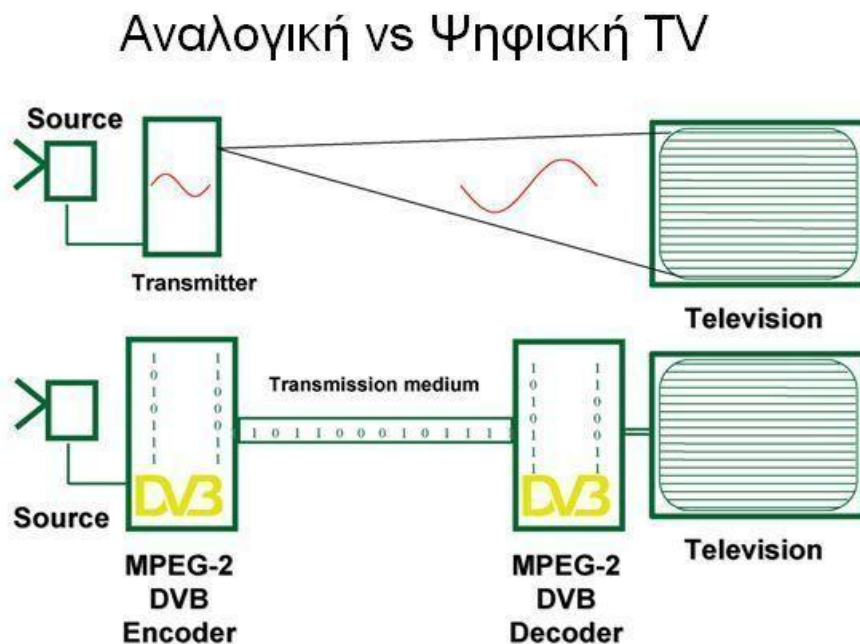
MIME	Multipurpose Internet Mail Extensions
MPE-FEC	Multi-Protocol Encapsulation – Forward Error Correction
MPEG	Moving Picture Experts Group
OFDM	Orthogonal Frequency Division Multiplexing
PCF	Portable Content Format
PDA s	Personal Digital Assistants
PFC	Page Flow Container
PNG	Portable Network Graphics
RS	Reed-Solomon
SFC	Scroll Flow Container
SFN s	Single-Frequency Networks
TDM	Time Division Multiplexing
TFC	Truncate Flow Container
TPS	Transmission Parameter Signalling
TS	Transport Stream
UHF	Ultra high frequency
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
UTF	Unicode Transformation Format
VHF	Very-High Frequency
XML	eXtensible Markup Language

2. ΨΗΦΙΑΚΗ ΤΗΛΕΟΡΑΣΗ ΣΤΗΝ ΕΥΡΩΠΗ

2.1 Τι είναι η ψηφιακή τηλεόραση (DTV)

Η ψηφιακή τηλεόραση (Digital Television - DTV) είναι ένας τρόπος μετάδοσης των κινούμενων εικόνων και του ήχου που συνθέτουν ένα τηλεοπτικό πρόγραμμα, μαζί με άλλες υπηρεσίες όπως κείμενο και διαδραστικότητα. Χρησιμοποιεί ψηφιακά διαμορφωμένα δεδομένα, τα οποία είναι συμπιεσμένα και απαιτεί αποκωδικοποίηση από ένα ειδικά σχεδιασμένο σετ τηλεόρασης. Τα πλεονεκτήματα που δίνει είναι πολλά έναντι του αναλογικού συστήματος μετάδοσης τηλεόρασης που χρησιμοποιούσαμε για σχεδόν εβδομήντα χρόνια.

Η αναλογική και η ψηφιακή τηλεόραση διαφέρουν στον τρόπο που η πληροφορία μεταφέρεται από την πηγή στο δέκτη. Με απλά λόγια, στην αναλογική ευρυεκπομπή το σήμα είναι στη μορφή ενός συνεχούς κύματος, ενώ στην ψηφιακή είναι στη μορφή διακριτών τμημάτων πληροφορίας. Η απαιτούμενη πληροφορία για ένα τηλεοπτικό πρόγραμμα κωδικοποιείται σε μια ψηφιακή ακολουθία από μηδέν και ένα, όπως δηλαδή λειτουργεί ένας υπολογιστής. Με τον τρόπο αυτό η τεχνική ποιότητα μπορεί να είναι πολύ καλύτερη και περισσότερο συνεκτική. Η ψηφιακή ροή δεδομένων καταλαμβάνει πολύ λιγότερο εύρος ζώνης μετάδοσης και επιτρέπει τη συνύπαρξη πέντε, έξι ή επτά διαφορετικών προγραμμάτων στο εύρος ζώνης που χρειαζόταν στο παρελθόν για ένα μόνο αναλογικό κανάλι.



Εικόνα 1. Αναλογική και Ψηφιακή μετάδοση

Αυτό σημαίνει περισσότερες επιλογές παρακολούθησης υπηρεσιών για τον τηλεθεατή. Επιπλέον, υπάρχει η δυνατότητα πραγματικού widescreen για τα ψηφιακά κανάλια (κάτι που η αναλογική τηλεόραση δε μπορεί να πετύχει πλήρως), ενώ αξιοσημείωτη είναι η δυνατότητα για εικόνα υψηλής ευκρίνειας.

Αυτό που ξεχωρίζει, όμως, είναι η εντελώς νέα εμπειρία της διαδραστικότητας. Ο τηλεθεατής χρησιμοποιώντας το τηλεχειριστήριο μπορεί να έχει πρόσβαση σε κάθε είδους πληροφορία: το ψηφιακό κείμενο είναι πιο ευδιάκριτο από το παλαιότερο σύστημα teletext και η «εμπλουτισμένη» τηλεόραση (enhanced TV) παρέχει σχετιζόμενες πληροφορίες που τρέχουν παράλληλα με το παρακολουθούμενο τηλεοπτικό πρόγραμμα. Μερικά ψηφιακά συστήματα επιτρέπουν την επιλογή της κάμερας μέσω της οποίας ο τηλεθεατής παρακολουθεί τα γεγονότα ή την αύξηση της επιλογής πληροφοριών είτε αυτά είναι ειδήσεις είτε αθλητικά είτε άλλο περιεχόμενο. Μερικές ψηφιακές υπηρεσίες παρέχουν ακόμη και πρόσβαση σε σελίδες του διαδικτύου μέσω του τηλεοπτικού σετ.

Τα πρότυπα ψηφιακής τηλεόρασης που έχουν αναπτυχθεί σήμερα είναι τα:

- Advanced Television Systems Committee (ATSC) – 8VSB (Αμερική)
- Digital Video Broadcasting (DVB) – COFDM (Ευρώπη)
- Integrated Services Digital Broadcasting (ISBD) – BST-OFDM (Ιαπωνία)

Το σύστημα της ψηφιακής τηλεόρασης σχεδιάστηκε για να αντικαταστήσει σταδιακά το υπάρχον αναλογικό. Φέρει αρκετές καινοτομίες, ενώ εισάγει σε αυτό που ονομάζουμε τηλεόραση σήμερα υπηρεσίες (πχ διαδραστικές) που ανήκουν παραδοσιακά σε άλλους χώρους, όπως οι ηλεκτρονικοί υπολογιστές και τα δίκτυα υπολογιστών.

Στην Ευρώπη είναι απόφαση της Ευρωπαϊκής Ένωσης να προωθήσει στα κράτη μέλη της την ψηφιακή τηλεόραση και μάλιστα να ολοκληρωθεί η μετάβαση από την αναλογική στην πλήρως ψηφιακή μετάδοση μέχρι το 2015. Είναι ένα φιλόδοξο έργο με αρκετές τεχνικές και οικονομικές δυσκολίες, που υπό προϋποθέσεις (προσεκτικός σχεδιασμός σε κάθε χώρα και κατάλληλες επενδύσεις) θα δώσει ώθηση στην ανάπτυξη του χώρου.

2.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΨΗΦΙΑΚΗΣ ΤΗΛΕΟΡΑΣΗΣ

Η τηλεοπτική τεχνολογία έχει διατηρήσει την αναλογική μέθοδο εκπομπής σε μια εποχή που έχει επικρατήσει για τις περισσότερες μορφές πληροφορίας η ψηφιοποιημένη αποθήκευση και μεταφορά. Μόνο την τελευταία δεκαετία και χάρη στα πρότυπα συμπίεσης και πολυπλεξίας εικόνας της οικογένειας MPEG και το ευρωπαϊκό σύνολο προδιαγραφών DVB για διαμόρφωση και μετάδοση έχει αποκτήσει η ψηφιακή τηλεόραση μια συγκεκριμένη προοπτική.

Τα πλεονεκτήματα τα οποία εισάγει η ψηφιακή τεχνολογία στην διαμόρφωση και διανομή του τηλεοπτικού σήματος είναι αρκετά. Μερικά από αυτά είναι:

- Σταθερή ποιότητα εικόνας, με μεγαλύτερη ανοχή στις ατέλειες του ασύρματου ή ενσύρματου διαύλου. Εξάλειψη φαινομένων όπως αυτά της θόλωσης, των πολλαπλών ειδώλων ή του θορύβου («χιονιού»).
- Μειωμένος λόγος σήματος προς θόρυβο που απαιτείται σε σύγκριση με την αναλογική μετάδοση. Αυτό επιτρέπει τη μείωση της εκπεμπόμενης ισχύος μέχρι και κατά 30 dB χωρίς να διαταράσσεται η ποιότητα της εικόνας.
- Καλύτερη εκμετάλλευση του φάσματος λόγω της συμπίεσης του σήματος βασικής ζώνης. Για παράδειγμα, ένα επίγειο κανάλι UHF αντιστοιχεί σε ένα και μόνο αναλογικό τηλεοπτικό πρόγραμμα, ενώ το ίδιο εύρος μπορεί να φιλοξενήσει μέχρι και 4 ψηφιακά προγράμματα υψηλής ποιότητας.
- Ευέλικτες τεχνικές πολυπλεξίας των ψηφιακών συστημάτων, που επιτρέπουν την συνύπαρξη πολλών προγραμμάτων και υπηρεσιών επιλεγόμενης ποιότητας και ευκρίνειας.
- Μεταβλητή ταχύτητα (bit rate) εκπομπής, ανάλογα με τις απαιτήσεις ποιότητας του προγράμματος, κάτι που μεταξύ άλλων αυξάνει το κέρδος πολυπλεξίας (multiplexing gain) στην περίπτωση της ταυτόχρονης μετάδοσης πολλών προγραμμάτων μεταβλητού ρυθμού.
- Δυνατότητα επεξεργασίας της εικόνας στο δέκτη μετά τη λήψη μέσω αλγορίθμων ψηφιακής επεξεργασίας (digital image post-processing), όπως χρωματική διόρθωση, αποκοπή ορίων, αυξομείωση του μεγέθους ή αφαίρεση θορύβου.
- Εύκολος εμπλουτισμός των τηλεοπτικών προγραμμάτων μέσω τυποποιημένων αρχιτεκτονικών (π.χ. MHP, OpenTV) με τοπικές εφαρμογές που εκμεταλλεύονται τις δυνατότητες των σύγχρονων «έξυπνων δεκτών».
- Ενσωμάτωση διαφόρων πολυμεσικών εφαρμογών και υπηρεσιών δεδομένων, όπως αμφίδρομων υπηρεσιών και διαδικτυακής πρόσβασης σε μια κοινή ψηφιακή πλατφόρμα, με προϋπόθεση ότι υπάρχει διαθέσιμο κανάλι επιστροφής (reverse path).[7]

2.3 Mobile TV

Η εκπομπή ψηφιακού τηλεοπτικού σήματος σε κινητές συσκευές (Mobile TV) συνδυάζει τα δύο καταναλωτικά προϊόντα, που παρουσίασαν τις μεγαλύτερες πωλήσεις στην ιστορία της ανθρωπότητας: τηλεοράσεις και κινητά τηλέφωνα. Ως αποτέλεσμα, οι Mobile TV εφαρμογές θεωρούνται υποψήφιος για την επόμενη ασύρματη “killer application”. Έρευνα αγοράς φανερώνει αυξημένη ζήτηση και ταχεία ανάπτυξη εφαρμογών, η οποία θα οδηγήσει σε σημαντικά οφέλη. Το τηλεοπτικό περιεχόμενο θα

αποτελέσει το συστατικό το οποίο θα καθορίσει τη ζήτηση στην επόμενη γενιά ασύρματων κινητών συσκευών, δεδομένου ότι οι καταναλωτές επιζητούν την επικοινωνία και τη διασκέδαση οπουδήποτε, χρησιμοποιώντας μόνο μια συσκευή. Στην αλυσίδα αξίας εμπλέκονται κατασκευαστές συσκευών, κάτοχοι υποδομών, πάροχοι περιεχομένου, φορείς ευρυεκπομπής, πάροχοι κινητών επικοινωνιών, κατασκευαστές κυκλωμάτων κλπ. Σύμφωνα με αναλυτές, η αντίστοιχη αγορά αναμένεται να αναπτυχθεί μαζικά έως το 2009.

Όπως συμβαίνει στις περισσότερες νέες τεχνολογίες, υπάρχουν πολλά διαφορετικά πρότυπα. Οι πάροχοι κινητών επικοινωνιών επιθυμούν τη χρήση των δικτύων τους προκειμένου να αποστείλουν τηλεοπτικές υπηρεσίες στους πελάτες τους. Παρόλο που η δυνατότητα αυτή υπάρχει και χρησιμοποιείται από κάποιους παρόχους, η μετάδοση πολυμεσικών ροών περιορίζεται μέσω μεθόδων μονοεκπομπής. Έτσι περιορίζεται το πλήθος των χρηστών που έχουν ταυτόχρονη πρόσβαση σε μια υπηρεσία και απαιτείται από τους παρόχους η προσεκτική εξισορρόπηση του δικτυακού φόρτου και του κόστους ανά megabyte δεδομένων. Συνεπώς, μαζικές προσφορές είναι δύσκολο να επιτευχθούν, χωρίς συμβιβασμούς στην ποιότητα.

Τα τηλεοπτικά προγράμματα μπορούν να μεταδοθούν στους κινητούς χρήστες με δύο τρόπους: ευρυεκπομπή ή μονοεκπομπή. Κατά την μετάδοση ευρυεκπομπής το ίδιο περιεχόμενο πληροφορίας γίνεται διαθέσιμο, ταυτόχρονα για ένα μεγάλο αριθμό κινητών χρηστών στα κινητά τους τηλέφωνα. Κατά την μετάδοση μονοεκπομπής το περιεχόμενο της πληροφορίας μεταδίδεται βάση της υπηρεσίας σημείου προς σημείο από μια πηγή σε ένα μοναδικό προορισμό. Η μετάδοση μπορεί επίσης να είναι πολυεκπομπή σε μία ομάδα κινητών χρηστών, όπου το περιεχόμενο της πληροφορίας γίνεται διαθέσιμο σε πολλαπλούς συνδρομητές.



Εικόνα 2. Μονοεκπομπή (unicast)



Εικόνα 3. Ευρυεκπομπή (broadcast)

Συγκρίνοντας την ευρυεκπομπή και την μονοεκπομπή, παρατηρείται ότι παρόλο που η μετάδοση μονοεκπομπής υποστηρίζει πληθώρα υπηρεσιών, δε μπορεί να εξυπηρετήσει μαζική ζήτηση, ειδικά σε περίπτωση απεριόριστου χρόνου θέασης που παρέχεται από τους παρόχους υπηρεσιών. Τα συστήματα ευρυεκπομπής εξυπηρετούν τη μαζική ζήτηση αλλά δεν προσαρμόζονται εύκολα στην υποστήριξη μεγάλου πλήθους καναλιών και υπηρεσιών που δεν μεταδίδονται μέσω ευρυεκπομπής (Pay-TV, Pay-per-view). Λύση στα θέματα χωρητικότητας δίνεται με το συνδυασμό των δύο προσεγγίσεων σε ένα ενοποιημένο υβριδικό σενάριο, το οποίο θα περιορίζει τα μειονεκτήματα του ενός συστήματος εκμεταλλευόμενο τα πλεονεκτήματα του άλλου. Τα δημοφιλέστερα προγράμματα εκπέμπονται και τα υπόλοιπα αποστέλλονται επιλεκτικά σε συγκεκριμένους χρήστες, μέσω συνδέσεων μονοεκπομπής. Το σενάριο εγγυάται εξοικονόμηση δικτυακών πόρων. Ένα παράγοντα επιτυχίας στη υβριδική διανομή κινητών τηλεοπτικών υπηρεσιών αποτελεί η συνεργασία και όχι ο ανταγωνισμός με τους πάροχους κινητών επικοινωνιών .

Η Ευρωπαϊκή Επιτροπή επέλεξε ως **πρότυπο** μετάδοσης της **mobile TV** για όλα τα ευρωπαϊκά δίκτυα το **DVB-H**. Το DVB-H [Digital Video Broadcasting for Handhelds] είναι ήδη το πιο διαδεδομένο format mobile TV στην Ευρώπη. Η θέσπιση του προτύπου DVB-H σε κλίμακα ΕΕ προσφέρει στους φορείς εκμετάλλευσης και στον κλάδο το απαραίτητο εύρος αγοράς για τη μαζική προώθηση υπηρεσιών κινητής τηλεόρασης σε ολόκληρη την Ευρώπη. Από την ύπαρξη ευρωπαϊκού κοινού προτύπου θα επωφεληθούν επίσης οι καταναλωτές, που θα είναι σε θέση, οποτεδήποτε και οπουδήποτε στην Ευρώπη, να παρακολουθούν τηλεόραση στα τηλέφωνα ή τις κινητές συσκευές τους. Το DVB-H βρίσκεται στη φάση μεταξύ δοκιμών και έναρξης εμπορικής εκμετάλλευσης σε 16 χώρες. Η εμπορική εκμετάλλευση υπηρεσιών DVB-H έχει ήδη αρχίσει στην Ιταλία, ενώ στη διάρκεια του τρέχοντος έτους αναμένεται να προστεθούν και η Φιλανδία, η Αυστρία, η Γαλλία, η Ελβετία και η Ισπανία.

3. DVB-H (Digital Video Broadcasting – Handheld)

3.1. Εισαγωγή – Ιστορική Αναδρομή

Η ψηφιοποίηση των παραδοσιακών συστημάτων εκπομπής (broadcast systems) παρουσιάζει σημαντική πρόοδο τα τελευταία χρόνια. Η ανάπτυξη αυτή μπορεί να σχετισθεί με την παρουσία του DVB-T (Digital Video Broadcasting – Terrestrial) που αποτελεί ένα πρότυπο ψηφιακής επίγειας τηλεόρασης και βρίσκεται ήδη σε λειτουργία σε 71 χώρες του πλανήτη. Η επιλογή του DVB-T βασίσθηκε σε σημαντικά χαρακτηριστικά του προτύπου, όπως η δυνατότητα λήψης υπηρεσιών εκπομπής σε φορητές συσκευές και σε αυτοκίνητα.

Εντωμεταξύ τα πλεονεκτήματα ενός ισχυρού επίγειου συστήματος εκπομπής, όπως το DVB-T, προσέλκυσε το ενδιαφέρον της βιομηχανίας κινητών τηλεπικοινωνιών. Συγκεκριμένα, η δυνατότητα πρόσβασης κινητών τερματικών μέσω ασύρματων συνδέσμων point-to-multipoint, η ευρεία γεωγραφική κάλυψη και η υψηλή χωρητικότητα μετάδοσης που προσφέρει το DVB-T, αποτελούν στοιχεία που κέντρισαν το ενδιαφέρον της βιομηχανίας.

Το διεθνές DVB Project ικανοποίησε αυτό το ενδιαφέρον αναπτύσσοντας ένα νέο πρότυπο, το DVB-H (Digital Video Broadcasting – Handheld) που αποτελεί ένα ψηφιακό πρότυπο εκπομπής (broadcast standard) για τη μετάδοση περιεχομένου σε μικρού μεγέθους φορητές συσκευές όπως πχ. κινητά τηλέφωνα, PDAs (Personal Digital Assistants) κλπ. Ο καθορισμός των τεχνικών προδιαγραφών ξεκίνησε το φθινόπωρο του 2002 και ολοκληρώθηκε το Φεβρουάριο του 2004. Τέλος, εκδόθηκε ως πρότυπο από το Ευρωπαϊκό Ινστιτούτο Τηλεπικοινωνιακών Προτύπων (ETSI - European Telecommunications Standards Institute), το Νοέμβριο του 2004 [3]. Η τεχνολογία DVB-H προέρχεται από το πρότυπο DVB-T και είναι συμβατή με αυτό. Επιπροσθέτως, λαμβάνει υπόψη ιδιότητες των τυπικών κινητών τερματικών, όπως μέγεθος, βάρος, φορητότητα και κυρίως εξοικονόμηση ενέργειας. Το DVB-H μπορεί να προσφέρει downstream κανάλι με υψηλό data-rate (Mbit/s), ως βελτίωση των δικτύων κινητής τηλεφωνίας, το οποίο είναι προσβάσιμο από τις περισσότερες τυπικές συσκευές. Το κανάλι αυτό μπορεί να χρησιμοποιηθεί σε εφαρμογές ροής ήχου και εικόνας, σε καταφόρτωση αρχείων και πολλές άλλες υπηρεσίες. Συνεπώς, το DVB-H γεφυρώνει τα παραδοσιακά συστήματα εκπομπής με τον κόσμο των κυψελωτών δικτύων, εισάγοντας ταυτόχρονα νέους τρόπους παροχής υπηρεσιών σε φορητά τερματικά και παρέχοντας νέες επιχειρηματικές δραστηριότητες στους παρόχους περιεχομένου (content providers) και στους διαχειριστές δικτύων (network operators).

3.2. Απαιτήσεις Συστήματος

Οι απαιτήσεις του συστήματος καθορίστηκαν από το DVB Project το 2002 [1],[2], [4]

➤ Μικρή κατανάλωση ενέργειας

Οι δέκτες που χρησιμοποιούνται σε αυτά τα συστήματα είναι πρωτίστως φορητοί (hand-held) και κινητοί. Ο όρος handheld terminal περιλαμβάνει πολυμεσικά κινητά τηλέφωνα με έγχρωμες οθόνες, PDAs και pocket PCs. Οι συσκευές αυτές έχουν κάποια κοινά χαρακτηριστικά: μικρές διαστάσεις και βάρος, καθώς και λειτουργία βάσει μπαταρίας. Αυτές οι ιδιότητες αποτελούν προϋποθέσεις για κινητή χρήση αλλά παράλληλα δημιουργούν περιορισμούς στο σύστημα μετάδοσης. Οι τερματικές συσκευές στερούνται εξωτερικής παροχής ενέργειας και αναγκάζονται να λειτουργούν με περιορισμένο απόθεμα. Η χαμηλή κατανάλωση ενέργειας είναι συνεπώς απαραίτητη.

➤ Κινητικότητα (handover)

Όπως αναφέρεται παραπάνω οι δέκτες που χρησιμοποιούν το πρότυπο DVB-H πρέπει να είναι κινητοί. Ως εγγύηση της κινητικότητας των χρηστών, αυτό το σύστημα πρέπει να είναι σε θέση να εξασφαλίσει ότι οι υπηρεσίες του θα συνεχιστούν να παρέχονται καθώς ένας χρήστης εγκαταλείπει μια κυψέλη και εισέρχεται σε μια καινούρια. Όμως τα ταχέως μεταβαλλόμενα κανάλια είναι επιρρεπή σε σφάλματα και η κατάσταση χειροτερεύει διότι οι ενσωματωμένες κεραιές των συσκευών, έχουν μικρές διαστάσεις και δε μπορούν να στοχεύσουν το μεταδότη, σε περίπτωση που η τερματική συσκευή βρίσκεται εν κινήσει. Επίσης παρατηρούνται παρεμβολές όταν μεταδίδονται και λαμβάνονται ραδιοσήματα GSM από την ίδια συσκευή. Επομένως η διαχείριση του downstreaming αρκετών Mbit/s από φορητές συσκευές αποτελεί απαιτητική υπόθεση.

➤ Προσαρμοστικότητα και εξέλιξη.

Το DVB-H είναι ένα πρότυπο το οποίο αναμένεται να χρησιμοποιηθεί σε εσωτερικούς, σε υπαίθριους χώρους, από πεζούς ανθρώπους και σε περιβάλλοντα κινούμενων οχημάτων. Επομένως το σύστημα μετάδοσης πρέπει να προσφέρει αρκετή ευελιξία και εξελισσιμότητα για να επιτρέψει την υποδοχή των υπηρεσιών DVB-H σε διάφορες ταχύτητες, βελτιστοποιώντας την μετάδοση του σήματος.

➤ ***Συμβατότητα με το DVB-T***

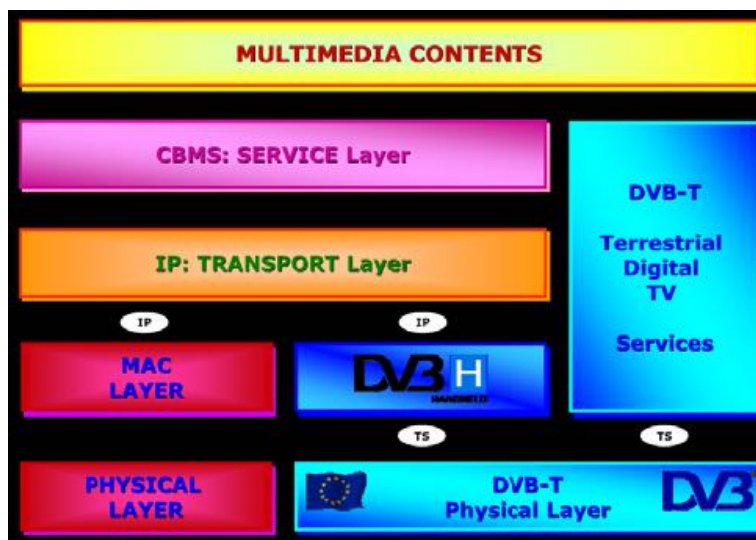
Τέλος, το νέο σύστημα πρέπει να είναι συμβατό και να χρησιμοποιεί την υπάρχουσα υποδομή του DVB-T, ώστε να επιτυγχάνεται η επαναχρησιμοποίηση του εξοπλισμού μετάδοσης.

Είναι ακόμη απαραίτητο να αναφέρουμε ότι το DVB-H πρέπει να έχει ρυθμό δεδομένων περίπου 10 Mbit/s ανά κανάλι. Τα κανάλια μετάδοσης θα επιμερισθούν στην ζώνη εκπομπής UHF. Είναι δυνατόν όμως να χρησιμοποιηθεί εναλλακτικά η VHF Band III αλλά και μη-παραδοσιακές τηλεοπτικές ζώνες εκπομπής.

3.3 Αρχιτεκτονική Συστήματος DVB-H

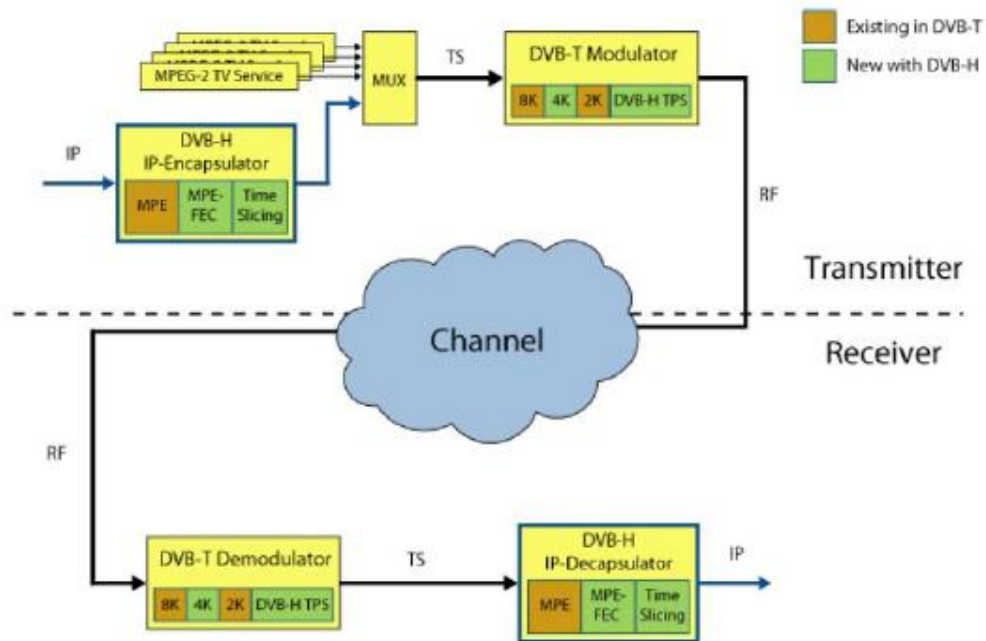
3.3.1. Γενική Περιγραφή

Οι παραδοσιακές υπηρεσίες εκπομπής του DVB-T χρησιμοποιούν τη στοίβα πρωτοκόλλων εκπομπής Transport Stream (TS). Το πρότυπο μετάδοσης DVB-H ορίζει στοιχεία στο φυσικό επίπεδο (physical layer) και στο επίπεδο ζεύξης δεδομένων (link/MAC layer) (εικόνα 4).



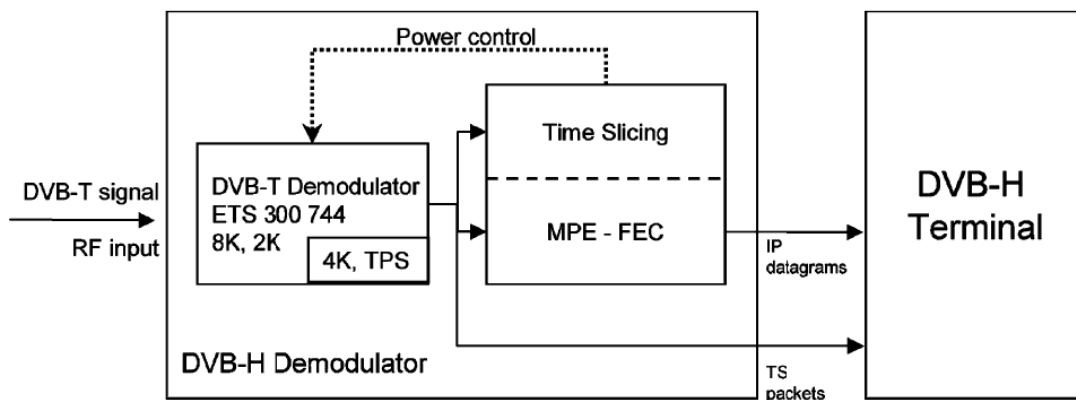
Εικόνα 4. Στοίβα πρωτοκόλλων στην μεταφορά δεδομένων.

Χρησιμοποιεί έναν αλγόριθμο εξοικονόμησης ενέργειας βασισμένο στη μετάδοση υπηρεσιών με διαίρεση χρόνου. Η τεχνική αυτή ονομάζεται time slicing και επιφέρει αυξημένη εξοικονόμηση ενέργειας και ήπιες διαπομπές όταν ο δέκτης μετακινείται σε γειτονική δικτυακή κυψέλη. Σε συνθήκες ανεπαρκούς σήματος, η αξιόπιστη μετάδοση επιτυγχάνεται με το σχήμα MPE-FEC (Multi-Protocol Encapsulation – Forward Error Correction). Επιπλέον, το πρότυπο ορίζει ένα πρόσθετο δικτυακό mode (4K mode), προσφέροντας πρόσθετη ευελιξία στο σχεδιασμό SFNs (Single-Frequency Networks), δεδομένου ότι τα δίκτυα αυτά ταιριάζουν στη λήψη σήματος από κινητά τερματικά και παρέχουν βελτιωμένο κανάλι σηματοδότησης και συνεπώς αξιόπιστες υπηρεσίες.



Εικόνα 5. Μετάδοση υπηρεσιών IP μέσω DVB-H.

Ένα παράδειγμα μετάδοσης υπηρεσιών IP μέσω του DVB-H απεικονίζεται στην εικόνα 5 [5]. Στην περίπτωση αυτή μεταδίδονται ταυτόχρονα παραδοσιακές τηλεοπτικές υπηρεσίες MPEG-2 και υπηρεσίες time-sliced DVB-H, μέσω της ίδιας πολύπλεξης (multiplex). Το φορητό τερματικό αποκωδικοποιεί (χρησιμοποιεί) μόνο τις υπηρεσίες IP.[4]



Εικόνα 6. Περιγραφή της δομής ενός DVB-H δέκτη

Η δομή του τερματικού δέκτη (user equipment) φαίνεται στην εικόνα 6. [5]. Περιλαμβάνει δέκτη και τερματικό DVB-H. Ο δέκτης αποτελείται από DVB-T αποδιαμορφωτή, time slicing module και ένα προαιρετικό MPE-FEC module. Ο DVB-T αποδιαμορφωτής ανακτά τα πακέτα των ροών MPEG-2 TSs (Transport Streams) από το ληφθέν σήμα DVB-T RF (Radio Frequency). Προσφέρονται τρία modes μετάδοσης (2K, 4K, 8K) με την κατάλληλη σηματοδότηση. Το time-slicing module ελέγχει το δέκτη ώστε να αποκωδικοποιήσει την κατάλληλη υπηρεσία. Τέλος, το MPE-FEC module παρέχει μια επιπλέον λειτουργία FEC, βοηθώντας το δέκτη να αντεπεξέρχεται σε δύσκολες καταστάσεις λήψης.

3.3.2. Νέα Στοιχεία

3.3.2.1. Φυσικό Επίπεδο (Physical Layer)

Στο φυσικό επίπεδο του DVB-T προστέθηκαν τέσσερα νέα χαρακτηριστικά τα οποία περιγράφονται στην συνέχεια. Η μετάδοση εξακολουθεί να γίνεται βάσει του προτύπου DVB-T, χρησιμοποιώντας Ορθογώνια Πολύπλεξη με Διαίρεση Συχνότητας (Orthogonal Frequency Division Multiplexing - OFDM). Η σηματοδότηση στο DVB-H υλοποιείται με τρόπο συμβατό προς το DVB-T. Επιπλέον οι ροές DVB-H είναι απολύτως συμβατές με τις DVB-T TS ροές, μεταφέροντας «κλασσικές» DVB-T πληροφορίες.

Ως αποτέλεσμα, μια ροή DVB-H μπορεί να εκπεμφθεί μέσω :

- Δικτύων αναμεταδοτών DVB-T αφιερωμένων (dedicated) σε DVB-H υπηρεσίες
- Δικτύων DVB-T τα οποία μεταφέρουν παράλληλα κλασσικές DVB-T και DVB-H υπηρεσίες.

Γι' αυτόν το λόγο, τα βασικά νέα χαρακτηριστικά του DVB-H (time-slicing και βελτιωμένο σχήμα FEC) τοποθετήθηκαν εσκεμμένα στο επίπεδο ζεύξης δεδομένων.[4]

3.3.2.1.1. TPS Signalling Bits

Η σηματοδότηση των παραμέτρων των ροών DVB-H, χρησιμοποιεί μια προέκταση του καναλιού TPS (Transmission Parameter Signalling) του DVB-T. Το κανάλι TPS αποτελεί ένα δεσμευμένο κανάλι πληροφοριών που παρέχει παραμέτρους ρύθμισης (tuning parameters) στο δέκτη. Τα bits της σηματοδότησης αναβαθμίστηκαν ώστε να περιλαμβάνουν δύο πρόσθετα bits τα οποία πληροφορούν για την παρουσία υπηρεσιών DVB-H και την πιθανή ύπαρξη προστασίας MPE-FEC σε τουλάχιστον μια ροή DVB-H. Τα bits τα οποία προστέθηκαν είναι τα s48,s49. Στο παρακάτω πίνακα (πίνακας 1) δείχνονται όλοι οι πιθανοί συνδυασμοί των s48, s49 bits και το αποτέλεσμα που δημιουργούν. [2],[4]

s48	s49	DVB-H signaling
0	X	Time Slicing not used
1	X	Time Slicing used = DVB-H *
X	0	MPE-FEC not used
X	1	MPE-FEC used *

Πίνακας 1. Πιθανές τιμές των των s48 και s49 bits

3.3.2.1.2. 4K Mode

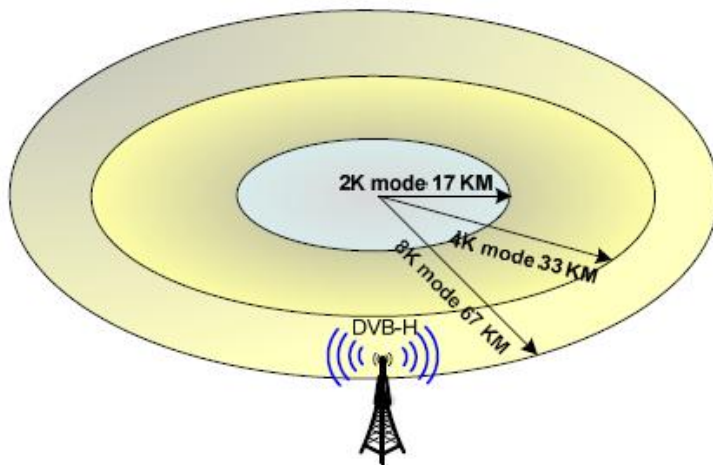
Οι ροές DVB-H μεταδίδονται με χρήση πολύπλεξης 4K OFDM , η οποία δεν αποτελεί μέρος του προτύπου DVB-T. Το DVB-T παρέχει δύο OFDM modes τα οποία ονομάζονται 2K και 8K προκειμένου να υποστηρίξει διαφορετικές δικτυακές τοπολογίες. Το DVB-H εισάγει επιπρόσθετα το 4K mode, το οποίο δημιουργείται από έναν IDFT (Inverse Discrete Fourier Transformation) μετασχηματισμό 4096 σημείων, στον OFDM διαμορφωτή. Το DVB-T δεν περιλαμβάνει το mode αυτό, άρα το 4K mode μπορεί να χρησιμοποιηθεί μόνο σε DVB-H δίκτυα. Ο πίνακας 2 παρουσιάζει τιμές παραμέτρων για τα τρία διαθέσιμα modes μετάδοσης

OFDM parameter	Mode		
	2K	4K	8K
Overall carriers (= FFT size)	2048	4096	8192
Modulated carriers	1705	3409	6817
Useful carriers	1512	3024	6048
OFDM symbol duration (μ s)	224	448	896
Guard interval duration (μ s)	7,14,28,56	14,28,56,112	28,56,112,224
Carrier spacing (kHz)	4.464	2.232	1.116
Maximum distance of transmitters (km)	17	33	67

Πίνακας 2. Τιμές παραμέτρων των 2K, 4K και 8K mode

Το 2K mode προσφέρει μετάδοση σε μικρού μεγέθους κυψέλες και μέγιστη ταχύτητα λήψης . Στο 4K mode η κινητικότητα αυξάνεται και πιο συγκεκριμένα διπλασιάζεται σε σχέση με το 8K mode και το μέγεθος SFN κυψέλης είναι διπλάσιο σε σχέση με το 2K mode. Η 4K αποσκοπεί στο να προσφέρει έναν επιπλέον βαθμό ευελιξίας στη σχεδίαση δικτύων DVB-H με πιο ελαστική ανταλλαγή ανάμεσα στις κυψέλες μετάδοσης και στις δυνατότητες λήψης εν κινήσει. Η επιλογή του διαστήματος φύλαξης αντιπροσωπεύει την ανταλλαγή μεταξύ χωρητικότητας και του μεγέθους της SFN κυψέλης.

Στην συνέχεια απεικονίζεται η περιοχή κάλυψης και η μέγιστη απόσταση μετάδοσης που μπορεί να επιτευχθεί από τους διάφορους τρόπους μετάδοσης.



Εικόνα 7 .Περιοχή κάλυψης των 2K ,4K και 8K modes.

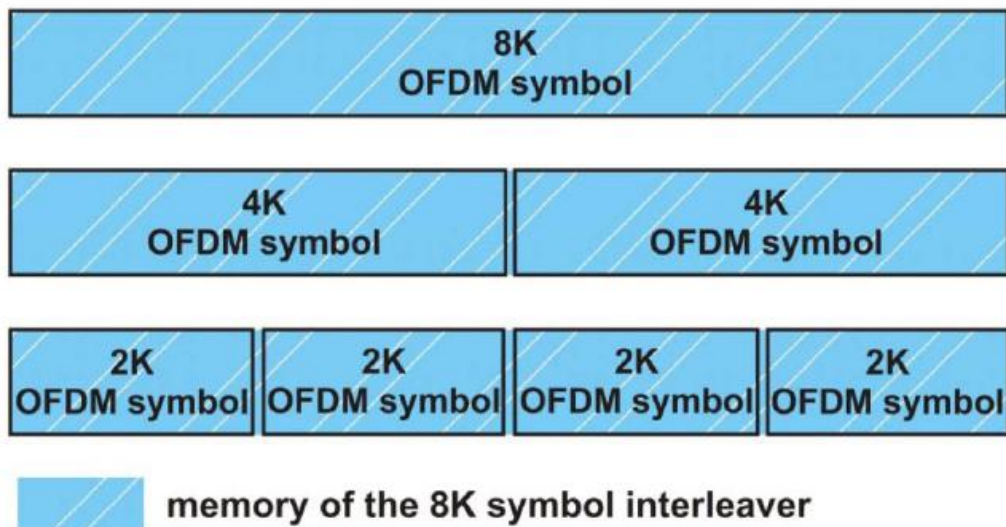
Ο σχεδιαστής του δικτύου έχει την δυνατότητα να επιλέξει ανάμεσα σε αυτά τα τρία mode.[1]

- 2K mode: κατάλληλο για κυσέλες μικρής ακτίνας (μέγιστη απόσταση 17 km ανάμεσα στους μεταδότες). Υποστηρίζει εξαιρετικά υψηλές ταχύτητες λήψης πχ. λήψη σε αυτοκινητόδρομους και τρένα ταχείας (high-speed) κυκλοφορίας.
- 4K mode: κατάλληλο για κυσέλες μικρής και μεσαίας ακτίνας (μέγιστη απόσταση 33 km ανάμεσα στους μεταδότες). Υποστηρίζει υψηλές ταχύτητες λήψης και κρίνεται κατάλληλο για τα περισσότερα σενάρια χρήσης του DVB-H πχ. λήψη σε αυτοκινητόδρομους κλπ.
- 8K mode: κατάλληλο για κυσέλες μεγάλης ακτίνας (μέγιστη απόσταση 67 km ανάμεσα στους μεταδότες). Μολονότι υποστηρίζει αρκετά υψηλές ταχύτητες λήψης, δεν αποτελεί καλή επιλογή, όταν είναι αναγκαία η υψηλή ταχύτητα. Είναι ευαίσθητο σε φαινόμενα Doppler.

Η χρήση της πολύπλεξης OFDM επιτρέπει το βέλτιστο σχεδιασμό του δικτύου, καθώς αυξάνεται το πλήθος των καναλιών που χρησιμοποιούν την ίδια συχνότητα [4]. Τα δίκτυα αυτά ονομάζονται Single Frequency Networks (SFNs). Πολλαπλοί μεταδότες χρησιμοποιούν την ίδια συχνότητα. Το πλεονέκτημα σε σχέση με τα Multi Frequency Networks (MFNs) είναι ότι υπάρχει επαναχρησιμοποίηση των συχνοτήτων. Επομένως στο DVB-H, είναι δυνατή η μετάδοση πολλών σημάτων πάνω από την ίδια συχνότητα.

3.3.2.1.3. In-Depth Interleavers

Σε σχέση με τα τρία διαθέσιμα modes, ορίζονται σχήματα διεμπλοκής συμβόλων (interleaving mode schemes). Αφού το mode 8K έχει τον μεγαλύτερο αριθμό φερόντων μεταξύ των τριών mode, απαιτεί και την περισσότερη μνήμη για την εκτέλεση της αναδιάταξης σε σχέση με τα mode 2K και 4K. Χοντρικά ο αριθμός των φερόντων στο mode 8K είναι τετραπλάσιος σε σχέση με το mode 2K και διπλάσιος σε σχέση με το mode 4K. Επομένως είναι λοιπόν φυσικό, η αυξημένη μνήμη του 8K symbol interleaver, να χρησιμοποιείται και στα 3 modes. Επομένως, ο symbol interleaver του τερματικού μπορεί να επεξεργαστεί δεδομένα που έχουν μεταδοθεί σε ένα 8K OFDM σύμβολο, σε δύο 4K OFDM σύμβολα ή σε τέσσερα 2K OFDM σύμβολα. Το αποτέλεσμα είναι αυξημένο «βάθος» διεμπλοκής στα 2K και 4K modes, άρα και βελτιωμένη απόδοση. Αν το πλήρες μέγεθος της μνήμης χρησιμοποιηθεί, η μέθοδος ονομάζεται in-depth interleaving.



Εικόνα 8 . Μέθοδος in-depth interleaving.

Το 4K mode και οι in-depth interleavers επηρεάζουν το φυσικό επίπεδο. Όμως οι υλοποίησή τους δεν απαιτεί αύξηση στην πολυπλοκότητα του εξοπλισμού (λογικές πύλες, μνήμη) των μεταδοτών και δεκτών. Ένας τυπικός κινητός αποδιαμορφωτής ήδη ενσωματώνει αρκετή μνήμη και «λογική» για τη διαχείριση σημάτων 8K, η οποία υπερκαλύπτει τις απαιτήσεις του 4K mode. Τέλος, το φάσμα εκπομπής είναι κοινό και για τα τρία modes, επομένως δεν απαιτούνται αλλαγές στα φίλτρα των μεταδοτών.

3.3.2.1.4. 5MHz Ευρος Ζώνης Καναλιού

Τέλος η τελευταία προσθήκη που έχει γίνει στο πρότυπο DVB-T στο φυσικό στρώμα είναι η χρήση ενός φάσματος 5 MHz. Στο DVB-T χρησιμοποιούνται τρεις ζώνες συχνοτήτων για την εκπομπή κλασσικών τηλεοπτικών σημάτων (VHF band III 174..230 MHz, UHF band IV 470..598 MHz, UHF band V 598..862 MHz) και οι οποίες διαθέτουν φάσμα 6, 7, 8 MHz σε κάθε κανάλι.

3.3.2.2. Επίπεδο Ζεύξης Δεδομένων (Link/MAC Layer)

Οπώς αναφέρεται παραπάνω το DVB-H χρησιμοποιεί το ίδιο φυσικό επίπεδο με το DVB-T με ορισμένες μικρές επεκτάσεις. Αυτός είναι ο λόγος για τον οποίο το DVB-H είναι συμβατό με το DVB-T. Εντούτοις, υπάρχουν ακόμα μερικές σημαντικές απαιτήσεις για να εκπληρωθεί η δυνατότητα της κινητής τηλεοπτικής μετάδοσης .

Αυτές οι απαιτήσεις εξετάζονται στο στρώμα ζεύξης δεδομένων (*Link/MAC Layer*). Σε αυτό το επίπεδο γίνονται οι σημαντικότερες αλλαγές σε σχέση με το πρότυπο DVB-T. Λαμβάνοντας υπόψη την απαίτηση μικρής κατανάλωσης ισχύος το DVB-H εισάγει μια ειδική τεχνική η οποία ονομάζεται time slicing.

Το time slicing οδηγεί στην αξιοσημείωτη μικρή κατανάλωση ισχύος και πρόσθετα επιτρέπει ομαλή μετάβαση από μία κυψέλη σε μία άλλη. Για την λύση του ζητήματος της ευρωστίας και των εξασθενημένων λαμβανόμενων σημάτων εισάγεται μια νέα τεχνική προστασίας αποκαλούμενη MPE-FEC (Multi-Protocol Encapsulation Error Correction). Αυτή η τεχνική υιοθετεί την κωδικοποίηση Reed- Solomon.

3.3.2.2.1. Time-Slicing

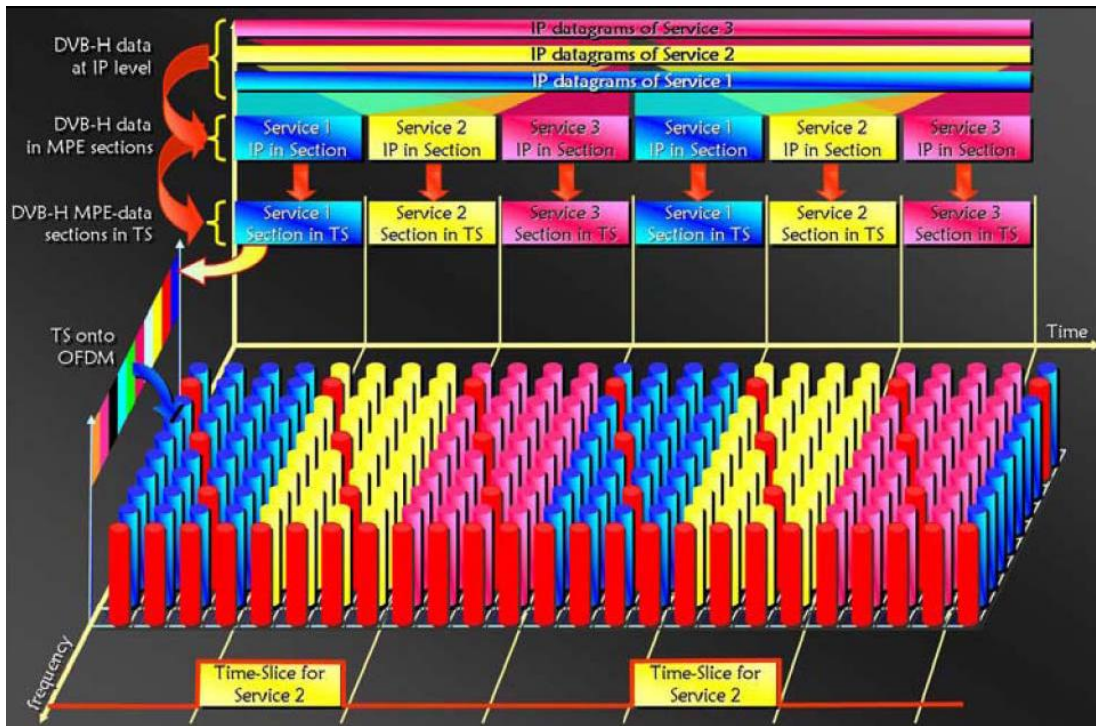
Στην παραδοσιακή συνεχή ροή δεδομένων (όπως γίνεται στην μετάδοση δεδομένων DVB-T MPEG-2) διάφορες υπηρεσίες (π.χ. τηλεοπτικά προγράμματα) μεταδίδονται σε μία ροή δεδομένων. Οι υπηρεσίες πολυπλέκονται και μεταδίδονται παράλληλα. Επομένως, δεν είναι δυνατό να μεταδοθεί μόνο μια ροή δεδομένων(υπηρεσία) αλλά ο δέκτης πρέπει να επεξεργαστεί όλα τα στοιχεία και κατόπιν επιλέξει την επιθυμητή ροή δεδομένων (υπηρεσία). Αυτό οδηγεί προφανώς στη μεγάλη κατανάλωση ισχύος (εικόνα 8).



Εικόνα 9. Πολυπλεξία και παράλληλη μετάδοση υπηρεσιών

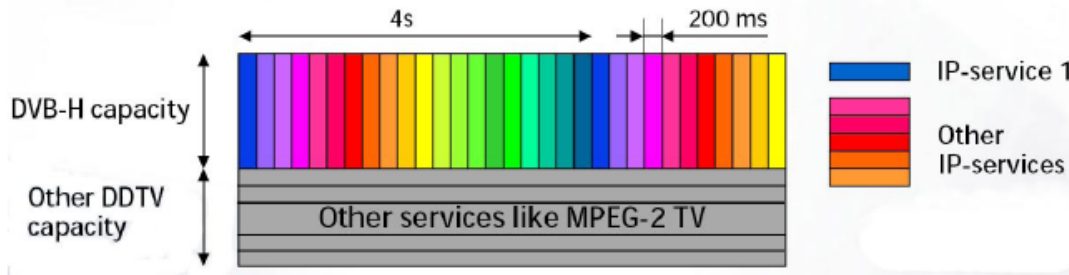
Ο σκοπός του time slicing είναι η μείωση της μέσης κατανάλωσης ισχύος του τερματικού και η ομαλή μετάβαση και χωρίς διακοπή των υπηρεσιών από μία κυψέλη σε μια άλλη. Η τεχνική αυτή βασίζεται στην τεχνική TDM (Time Division Multiplexing), η οποία χρησιμοποιείται στα συστήματα επικοινωνιών για την παροχή υπηρεσιών σε διαφορετικούς χρήστες σε διαφορετικές χρονοθυρίδες (slots). Οι υπηρεσίες δεν μεταδίδονται πλέον παράλληλα αλλά σε περιοδικά slot τα οποία ονομάζονται ριπές. Αυτές οι ριπές δεν έχουν απαραίτητα την ίδια διάρκεια.

Το DVB-H χρησιμοποιεί την τεχνική Multi-Protocol Encapsulation (MPE) για να εισάγει αυτοδύναμα πακέτα IP (IP datagrams) σε ροές MPEG TS. Δηλαδή, τα αυτοδύναμα πακέτα IP ενθυλακώνονται σε τμήματα MPE (MPE sections) και τα τελευταία κατακερματίζονται και εισάγονται σε πακέτα ροών MPEG TS. Τα τμήματα αυτά δεν μεταδίδονται μόλις είναι διαθέσιμα. Αντίθετα, συσσωρεύονται σε μια εγγραφή (record) μέχρι 191 kbytes και κατόπιν η εγγραφή αυτή μεταδίδεται το ταχύτερο δυνατόν, χρησιμοποιώντας όλους τους διαθέσιμους πόρους που παρέχει το φυσικό επίπεδο. Έτσι τα δεδομένα μιας DVB-H υπηρεσίας μεταδίδονται στον ατμοσφαιρικό διάυλο με τη μορφή περιοδικών ριπών (bursts) .(εικόνα 9)



Εικόνα 10 .Περιγραφή της τεχνικής time slicing

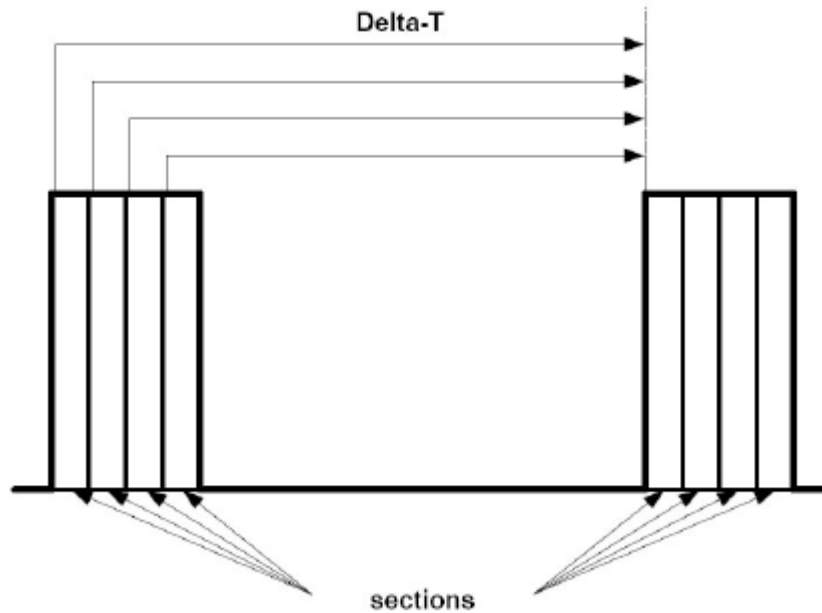
Κάθε υπηρεσία που μεταδίδεται χρησιμοποιεί όλους τους διαθέσιμους πόρους για ένα συγκεκριμένο χρονικό διάστημα . Μετά το τέλος αυτού του χρονικού διαστήματος , μια άλλη υπηρεσία μπορεί να μεταδοθεί και έπειτα μια άλλη κ.ο.κ. μετά από κάποιες υπηρεσίες που μεταδόθηκαν η πρώτη υπηρεσία είναι έτοιμη να μεταδοθεί ξανά . Αυτό παρουσιάζεται στην εικόνα 10 όπου η περίοδος μεταξύ της αποστολής της ίδιας υπηρεσίας για αυτό το παράδειγμα είναι 4s. Αφού όλα τα δεδομένα μιας υπηρεσίας έχουν ληφθεί μέσα σε μια χρονοθυρίδα των 200 ms, ο δέκτης μπορεί να αποδιαμορφώσει το σήμα (στα 200 ms) και να διακόψει (power-off) τη διαδικασία αποδιαμόρφωσης για 3800 ms, περιμένοντας για την επόμενη ριπή (της ίδιας υπηρεσίας). Κατά τη διάρκεια της διακοπής, ο αποκωδικοποιητής χρησιμοποιεί τα ενταμιευμένα δεδομένα τα οποία υπάρχουν στην μνήμη της συσκευής .Έτσι εξασφαλίζεται συνεχή εικόνα στο τερματικό. Ο δέκτης γνωρίζει πότε να εκκινήσει (power-on), προκειμένου να λάβει την επόμενη ριπή. Έτσι, η σηματοδότηση γίνεται πολύ ανθεκτική σε σφάλματα μετάδοσης. Χρησιμοποιώντας την μέθοδο time slicing , περισσότερο από 90% της ισχύος μπορεί να σωθεί .



Εικόνα 11. Υπηρεσίες DVB-H και DVB-T

3.3.2.2.1.1. Μέθοδος Delta - T

Προφανώς, πρέπει να υπάρχει κάποιος μηχανισμός που να εξασφαλίζει ότι ο δέκτης θα είναι ενεργός στο σωστό χρόνο, καθώς αναμένεται η επιθυμητή ριπή. Για αυτόν το λόγο χρησιμοποιείται η μέθοδος Delta-T. Ο σκοπός αυτής της μεθόδου είναι να επισημαίνει την έναρξη της επόμενης ριπής. Κάθε ριπή περιέχει πληροφορίες σχετικά με τον χρόνο άφιξης της επόμενης ριπής ριπής (που ανήκει στην ίδια υπηρεσία). Οι πληροφορίες συγχρονισμού της μεθόδου delta-t είναι σχετικές, δηλ. αναφέρουν μόνο τον χρόνο από την έναρξη της λήψης του συγκεκριμένου τμήματος MPE έως την έναρξη του επομένου τμήματος MPE. Με αυτό τον τρόπο η σηματοδότηση γίνεται ανθεκτική σε σφάλματα.

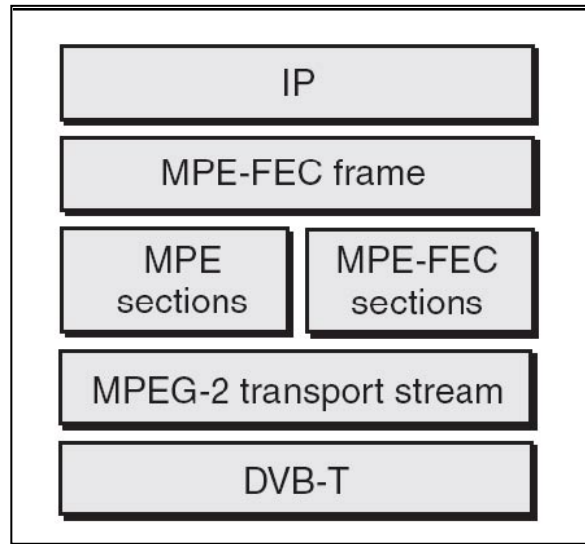


Εικόνα 12. Μέθοδος Delta-T

Ένα άλλο πλεονέκτημα της χρήσης του σχετικού χρόνου delta -t αντί του απόλυτου είναι ότι δεν υπάρχει καμία ανάγκη για ακριβή συγχρονισμό πομπού και δεκτή .

3.3.2.2 MPE - FEC

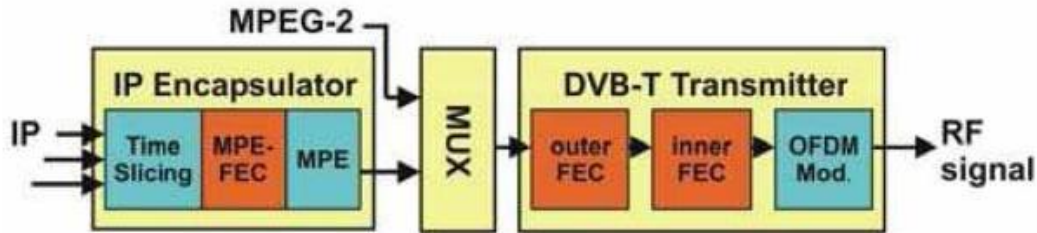
Η Multi-Protocol Encapsulation (MPE) είναι μία μέθοδος που χρησιμοποιείται για την παροχή δεδομένων πάνω σε ροές μετάδοσης MPEG2-TS σε δίκτυα DVB. Τα πακέτα IP ενθυλακώνονται σε τμήματα MPE. Πολλαπλά τμήματα MPE δημιουργούν μία στοιχειώδη ροή (Elementary Stream-ES) η οποία είναι μια ροή πακέτων MPEG2-TS. Κάθε τμήμα MPE αποτελείται από μια επικεφαλίδα, κυκλικό έλεγχο πλεονασμού (CRC) και ένα ωφέλιμο φορτίο, το οποίο είναι πραγματικά το πακέτο IP που έχει ενθυλακωθεί σε ένα τμήμα MPE.



Εικόνα 13 .Στοιβή πρωτοκόλλων του DVB-H

Το DVB-T σχεδιάστηκε για να λαμβάνει υπηρεσίες TV συνήθως από σταθερούς δέκτες, με μεγάλες κεραιές. Εντούτοις, είναι δυνατό να παραληφθούν οι υπηρεσίες DVB-T κατά την κίνηση των οχημάτων, αλλά σε αυτή την περίπτωση, η ανάγκη για το κατάλληλο μέγεθος κεραιών είναι ακόμα μεγαλύτερη. Υπάρχουν δύο στάδια για την προστασία λαθών που χρησιμοποιούνται σε αυτά τα πρότυπα. Είναι ένας εσωτερικός συνελκτικός κώδικας και ένας εξωτερικός κώδικας που ονομάζεται Reed-Solomon Av και αυτοί παρέχουν μια ισχυρή προστασία δεν είναι ικανοποιητικοί στην περίπτωση του προτύπου DVB-H.

Σε αντίθεση με το DVB-T οι δέκτες στα δίκτυα DVB-H είναι γενικά κινητοί και χαρακτηρίζονται από μικρές κεραιές. Εκτός από αυτό, η μετάδοση σημάτων έχει να αντιμετωπίσει φαινόμενα πολλαπλών διαδρομών όπως επιλεκτική εξασθένιση συχνότητας και το φαινόμενο Doppler. Μαζί με το θόρυβο όλα αυτά δημιουργούν πολύ άσχημες συνθήκες μετάδοσης. Για αυτούς τους λόγους το σύστημα προστασίας λαθών που χρησιμοποιείται στο DVB-T δεν είναι επαρκές. Προκειμένου λοιπόν να διατηρηθεί η χρησιμότητα στο σύστημα DVB-H, ορίστηκε (στο MAC επίπεδο) ένας πρόσθετος μηχανισμός ασφαλείας, ο οποίος εφαρμόζεται πάνω στα δεδομένα των ριπών. Ο μηχανισμός αυτός ονομάζεται MPE-FEC (Multi-Protocol Encapsulation Forward Error Correction). Το σύστημα MPE-FEC αποσκοπεί στη μείωση των απαιτήσεων του ποσοστού Signal/Noise (S/N ratio) κατά την διάρκεια της λήψης από τη φορητή συσκευή. Πειραματικοί έλεγχοι του DVB-H (φθινόπωρο 2004) απέδειξαν ότι η χρήση του MPE-FEC επιφέρει ένα κέρδος περίπου 7 dB, σε σχέση με το DVB-T. Όλα τα στάδια της προστασίας λαθών στο DVB-H παρουσιάζονται στην εικόνα 13.

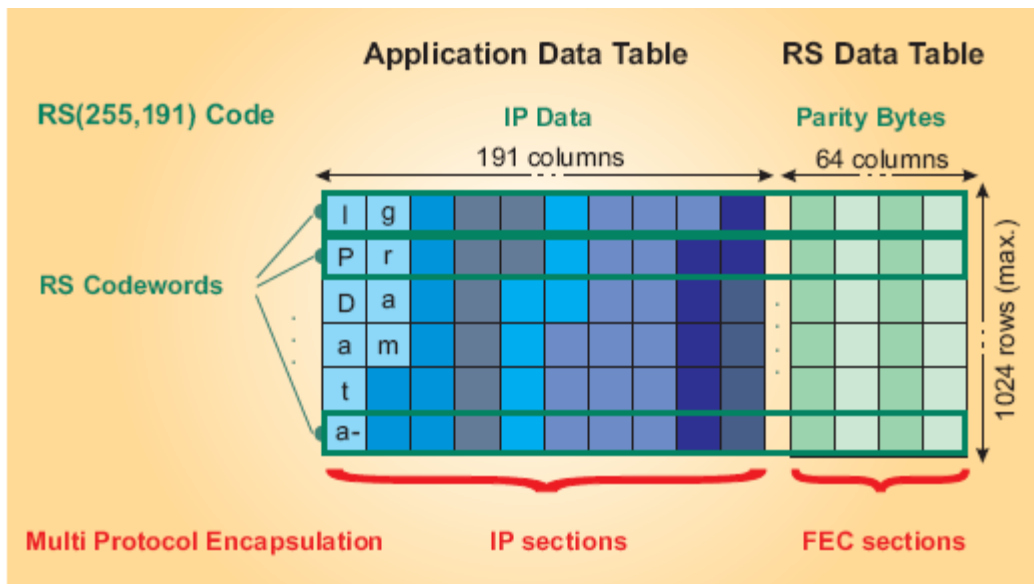


Εικόνα 14 . Σταδια προστασίας λαθων κατά την μετάδοση δεδομένων στο DVB-H

Το MPE-FEC μαζί με το Time slicing είναι οι δύο κύριες βελτιώσεις στο πρότυπο DVB-H. Είναι επίσης στενά συνδεδεμένα καθώς όταν μία ριπή χρησιμοποιείται στο time slicing αντιστοιχεί σε ένα πακέτο MPE-FEC. Σε αντίθεση με το time slicing το οποίο χρησιμοποιείται υποχρεωτικά στο πρότυπο DVB-H η χρήση των πακέτων MPE-FEC είναι προαιρετική. Οι τεχνικές MPE-FEC, MPE και time slicing αποτελούν από κοινού τον κωδικοποιητή DVB-H (DVB-H codec), ο οποίος παρέχει τη βασική λειτουργικότητα DVB-H.

Το πλαίσιο MPE-FEC αποτελείται από ένα πίνακα με σταθερό πλήθος 255 στηλών και μεταβλητό αριθμό γραμμών. Οι πληροφορίες για τον αριθμό σειρών βρίσκονται στο SI και οι τιμές που μπορεί να πάρει είναι 256, 512, 768 και 1024. Εφόσον κάθε κελί του πλαισίου αντιστοιχεί σε ένα byte ένα πλαίσιο MPE-FEC μπορεί να έχει μέγιστο μέγεθος 2 Mbits.

Ένα πλαίσιο MPE-FEC αποτελείται από δύο μέρη : τον Πίνακα Δεδομένων Εφαρμογής (Application Data Table) και τον Πίνακα Δεδομένων RS (Reed-Solomon Data Table). Ο πρώτος πίνακας περιλαμβάνει 191 στήλες, ενώ ο δεύτερος 64. Στον Πίνακα Δεδομένων Εφαρμογής καταχωρούνται τα πακέτα IP της υπηρεσίας που πρόκειται να προστατευθούν. Κατόπιν εφαρμόζεται ο κώδικας RS(255,191) σε κάθε γραμμή του πρώτου πίνακα και τα bytes ισοτιμίας (parity bytes) που προκύπτουν, καταχωρούνται στο δεύτερο πίνακα. Ύστερα από την κωδικοποίηση, τα πακέτα IP ενθυλακώνονται σε τμήματα IP (IP sections) με τρόπο που ορίζει η τεχνική MPE. Τα δεδομένα εφαρμογής ακολουθούνται από τα δεδομένα ισοτιμίας τα οποία διαβάζονται από το δεύτερο πίνακα ανά στήλη. Κατόπιν ενθυλακώνονται σε διαφορετικά τμήματα FEC. Η δομή του πλαισίου FEC περιέχει ένα εικονικό (virtual) block interleaving, επιπροσθέτως της κωδικοποίησης. Η εγγραφή προς, και το διάβασμα από το FEC πλαίσιο γίνεται ανά στήλη, ενώ η κωδικοποίηση ανά γραμμή.



Εικόνα 15 .Δομή ενός πλαισίου MPE-FEC

- **RS coding**

Τα πακέτα IP βρίσκονται μέσα στο πλαίσιο τοποθετημένα το ένα μετά το άλλο ξεκινώντας από την πάνω αριστερή γωνία του πίνακα και συνεχίζοντας διαγωνίως .Δεν υπάρχει κενός χώρος μεταξύ των πακέτων IP αν και μπορεί τα πακέτα IP να έχουν διαφορετικό μέγεθος. Εάν ένα πακέτο δεδομένων IP δεν τελειώνει μέχρι το τέλος μιας στήλης τότε καταλαμβάνει και την επόμενη στήλη. Όταν όλα τα IP datagrams εισαχθούν στον πίνακα κάθε θέση byte που δεν συμπληρώθηκε γεμίζεται με μηδενικά bytes.

Ο Πίνακας Δεδομένων RS γεμίζει με τα byte ισοτιμίας τα οποία υπολογίζονται από τα πακέτα IP και πιθανώς από μηδενικά bytes. Ο κώδικας Reed-Solomon RS (255,191) χρησιμοποιεί field generator polynomial (πολυώνυμο παραγωγής πεδίου) και code generator polynomial (πολυώνυμο παραγωγής κώδικα) . Κάθε γραμμή περιέχει μια κωδικό-λέξη RS. Κάποιες από τις δεξιότερες στήλες του πίνακα δεδομένων RS μπορεί να απορρίπτονται και ως εκ τούτου να μην μεταδίδονται, κάνοντας τον πίνακα διάτρητο. Ο ακριβής αριθμός των διάτρητων στηλών RS δεν είναι ανάγκη να σηματοδοτείται επακριβώς και μπορεί να αλλάζει δυναμικά μεταξύ πλαισίων.

Χρησιμοποιώντας αυτή την τεχνική, ο πίνακας στοιχείων RS είναι εντελώς γεμάτος και έτσι το πλαίσιο MPE-FEC ολοκληρώνεται

- **Block interleaving**

Αυτό επιτυγχάνεται από τη διαφορετική κατεύθυνση των διαδικασιών εγγραφής - ανάγνωσης και της κωδικοποίησης. Συγκεκριμένα η εγγραφή και η ανάγνωση του πλαισίου δεδομένων MPE-FEC γίνονται κάθετα, στην κατεύθυνση στηλών, ενώ η κωδικοποίηση εκτελείται οριζόντια, στην κατεύθυνση σειρών. Ως αποτέλεσμα έχουμε μια ενδογενή παρεμβολή σε όλα τα δεδομένα.

- **Decoding**

Ένα σημαντικό γνώρισμα της διαδικασίας της αποκωδικοποίησης είναι ο κώδικας προστασίας CRC-32 όποιος χρησιμοποιείται αντί για το checksum. Ο κώδικας CRC-32 αναφέρει στο δέκτη εάν ένα μήνυμα αλλοιώθηκε ή όχι. Τα τμήματα MPE μπορεί να χρησιμοποιούν είτε checksum (άθροισμα ελέγχου) είτε CRC-32, αν και συνιστάται η χρήση του CRC-32. Ωστόσο, όταν χρησιμοποιείται MPE-FEC είναι υποχρεωτική η χρήση του CRC-32. Στην πράξη όλα τα τερματικά πρέπει να υποστηρίζουν και τα δύο, το checksum και το CRC-32.

3.4 ΘΕΜΑΤΑ ΑΠΟΔΟΣΗΣ

Τα επίγεια συστήματα εκπομπής παρέχουν τρόπους αντιμετώπισης των πολλαπλών σημάτων που λαμβάνονται ως αντίγραφα από τους δέκτες. Στο DVB-T, η ανθεκτικότητα απέναντι στην ηχώ (echo) επιτυγχάνεται με τη χρήση ενός guard interval ανάμεσα σε κάθε διαμορφωμένο σύμβολο. Ορίζεται δηλαδή μια περίοδος μετάβασης μεταξύ των συμβόλων, κατά την οποία το κανάλι αγνοείται από τους δέκτες.

Η ενόχληση από την ηχώ αντιμετωπίζεται με ευκολία όταν οι σταθεροί δέκτες εκπομπής έχουν μια ποιοτική κατευθυντική (selective) κεραία οροφής, η οποία στρέφεται προς τον μεταδότη. Το πρόβλημα όμως γίνεται πιο σύνθετο, καθώς απαιτείται η στόχευση κινητών ή φορητών δεκτών, οι οποίοι διαθέτουν μη-κατευθυντικές (omni-directional) κεραίες και βρίσκονται σε ύψος 1m από το επίπεδο της γης.

3.4.1 Doppler Effect

Το φαινόμενο **Doppler** που πήρε το όνομά του από τον Κρίστιαν Ντόπλερ (*Christian Doppler*), είναι η παρατηρούμενη αλλαγή στη συχνότητα και το μήκος κύματος από παρατηρητή που βρίσκεται σε σχετική κίνηση με την πηγή των κυμάτων. Για κύματα όπως τα ηχητικά κύματα, που διαδίδονται μέσα σε κάποιο υλικό μέσο, η ταχύτητα τόσο του παρατηρητή όσο και της πηγής, πρέπει να προσδιορίζεται σε σχέση με το μέσο διάδοσης. Το τελικό φαινόμενο Doppler μπορεί επομένως να προκύψει είτε από την κίνηση του παρατηρητή, είτε από την κίνηση της πηγής, είτε και των δύο, ως προς το μέσο διάδοσης. Καθεμιά από αυτές τις δύο επιδράσεις αναλύεται ξεχωριστά. Για κύματα που δεν χρειάζονται υλικό μέσο για τη διάδοσή τους, όπως τα ηλεκτρομαγνητικά (φως) ή τα βαρυτικά κύματα στην ειδική θεωρία της σχετικότητας, μόνο η σχετική ταχύτητα του παρατηρητή ως προς την πηγή παίζει ρόλο.

Η φαινομενική ολίσθηση συχνότητας ή ολίσθηση Doppler ορίζεται από την σχέση

$$\Delta f_D = V * \frac{f_{rf}}{C} * \cos(\Phi)$$

Η παραπάνω εξίσωση συσχετίζει την ολίσθηση Doppler με την ταχύτητα του κινητού δέκτη και την γωνία μεταξύ της κατεύθυνσης της κίνησης του κινητού δέκτη και της κατεύθυνσης άφιξης του σήματος. Αν ο κινητός δέκτης κινείται προς την περιοχή από την οποία έρχεται το σήμα η ολίσθηση είναι θετική και αν ο δέκτης

απομακρύνεται από την περιοχή προέλευσης του σήματος η ολίσθηση συχνότητας είναι αρνητική.

Ο μηχανισμός των guard intervals αποτελεί μέθοδο αντιμετώπισης ανωμαλιών στο πεδίο του χρόνου. Αντίθετα, τα αποτελέσματα του Doppler effect αντιμετωπίζονται μόνο από αφιερωμένες μεθόδους επεξεργασίας σήματος που λειτουργούν στους δέκτες. Επομένως, ένας ποιοτικός δέκτης «Mobile TV» πρέπει πρωτίστως να είναι ποιοτικός «Mobile» δέκτης και να υλοποιεί όχι μόνο αλγορίθμους αντιμετώπισης χρονικών ανωμαλιών (time-delayed echoes) αλλά και συχνοτήτων (frequency-shifted echoes). Είναι λοιπόν απίθανο, οι αποδιαμορφωτές σταθερής λήψης (set-top boxes) να χρησιμοποιηθούν σε περιβάλλοντα κινητής λήψης.

3.5 Συγκριση DVB-H & DVB-T

Υπαρχουν πολλοί λόγοι για τους οποίους το DVB-T είναι διαφορετικό από το DVB-H. Το DVB-T για την ψηφιακή τηλεοπτική μετάδοση σημάτων σε σταθερούς τηλεοπτικούς δέκτες. Τα τερματικά σε αυτή την περίπτωση βρίσκονται υπό τάση 220 Volt και έχουν σταθερή παροχή ηλεκτρικού ρεύματος. Ο πίνακας παρακάτω αναφέρει τις διαφορές κάποιων παραμέτρων που χρησιμοποιούνται στα δύο προτυπα.

	Digital TV over DVB-T	IP Datacasting over DVB-H
Display	Large TV screen	Smaller, mobile phone screen
Antenna	Large, roof-top	Internal
Power Supply	Fixed, continuous	Battery powered limited
Reception mode	Fixed, continuous	Mobile

Εικόνα16. Τιμές κάποιων παραμέτρων που χρησιμοποιούνται στα προτυπα DVB-T & DVB-H.

Στο DVB-H το σήμα της ψηφιακής τηλεόρασης στέλνεται με πακέτα IP πάνω σε δίκτυα IP ενώ στο DVB-T το σήμα της ψηφιακής τηλεόρασης στέλνεται ως μία συνεχής ροή MPEG-2. Μόνο 128 έως 384 Kbps απαιτούνται για την μετάδοση κάθε τηλεοπτικού προγράμματος υψηλής ποιότητας βίντεο σε κινητά τηλέφωνα. Αυτό σημαίνει ότι 50 έως 80 τηλεοπτικά προγράμματα TV μπορούν να μεταδοθούν από το δίκτυο. Σε αντίθεση, η τεχνολογία DVB-T που σχεδιάστηκε για τερματικά με μεγάλες οθόνες, μπορεί να μεταφέρει μόνο 3 έως 5 προγράμματα TV πάνω στο ίδιο δίκτυο.

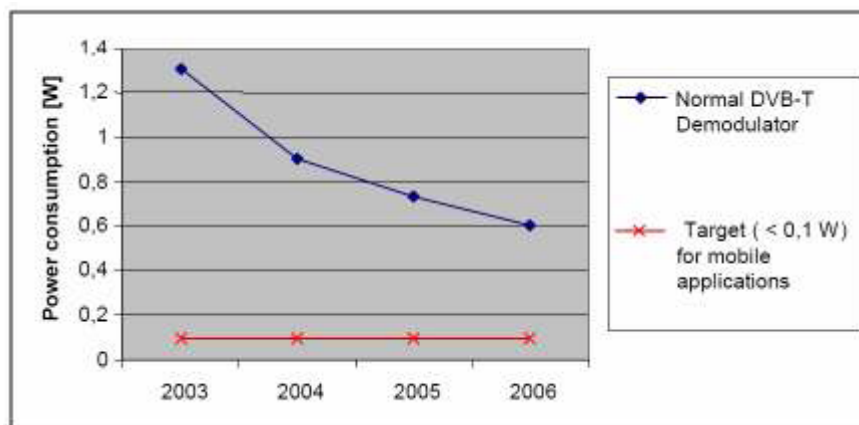
Επιπλέον, το DVB-H έχει σχεδιαστεί με σκοπό να υποστηρίξει την εσωτερική καθώς επίσης και υπαίθρια κάλυψη για κινητές συσκευές με μικρές ενσωματωμένες

κεραίες. Το DVB-T την υπαίθρια κάλυψη με μεγάλες κεραίες των τερματικών που είναι τοποθετημένες στις στέγες των σπιτιών .

Επιπλέον το πρότυπο DVB-T δεν είναι κατάλληλο για τα κινητά περιβάλλοντα λόγω του φαινομένου Doppler . Επομένως μια ειδική τεχνική διαμόρφωσης, η COFDM με 4K φέροντα, έχει υιοθετηθεί από DVB-H για να αντιμετωπίσει αυτόν τον περιορισμό. Σε ένα κινητό περιβάλλον, το σήμα εκτίθεται σε διάφορα φαινόμενα ραδιοδιάδοσης όπως παρεμβολές, εξασθένιση και το υψηλό επίπεδο θορύβου προκαλούμενου από τον άνθρωπο. Εντούτοις, αυτό έχει αντιμετωπιστεί στο DVB-H χρησιμοποιώντας την τεχνική διόρθωσης λαθών MPE-FEC για να εξασφαλιστεί η ευρωστία.

Μια άλλη διαφορά μεταξύ των δύο δικτύων είναι ότι η μετάδοση τηλεοπτικών σημάτων στο DVB-T είναι μία συνεχή ροή ενώ στο DVB-H, οι δέκτες μπορούν να μην είναι ενεργοποιημένοι όλη την ώρα και με αυτό τον τρόπο να εξοικονομούν έως 90 τοις εκατό ενέργεια. Η λύση είναι η τεχνική time slicing που χρησιμοποιείται στο DVB-H.

Το παρακάτω σχήμα περιγράφει της κατανάλωσης ισχύος των δεκτών στο DVB-T έναντι των δεκτών στο DVB-H. Είναι προφανές ότι ανεξάρτητα από την εξέλιξη του DVB-T , δεν έχει ακόμα φθάσει ακόμα στα ίδια επίπεδα κατανάλωσης ισχύος όπως το DVB-H.



Εικόνα 17 .Κατανάλωση ισχύος ανάμεσα σε δέκτες DVB-T και DVB-H.

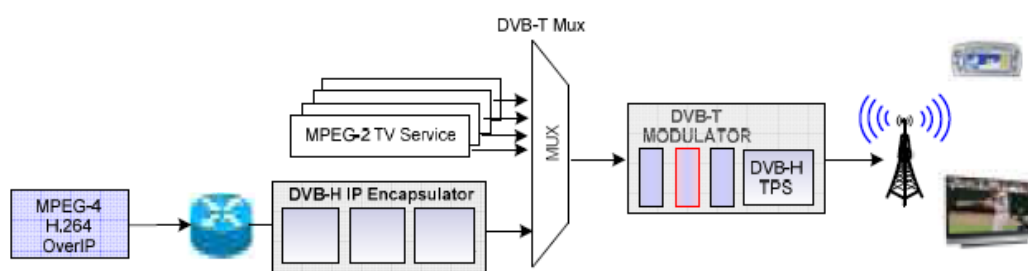
Αν και υπάρχουν κάποιες διαφορές ανάμεσα στο DVB-T και στο DVB-H τα δύο αυτά πρότυπα είναι συμβατά δεδομένου ότι μοιράζονται το ίδιο φυσικό στρώμα ,τους ίδιους διαμορφωτές και τους ίδιους μεταδότες. Αυτό είναι ένα βασικό πλεονέκτημα για το DVB-H επειδή μπορεί να χτιστεί πάνω από την υπάρχουσα υποδομή του DVB-T .

4. DVB-H ΔΙΚΤΥΑ

Έχουμε αναφέρει σε προηγούμενη παράγραφο ότι το DVB-H έχει σχεδιαστεί για να χρησιμοποιεί την υπάρχουσα υποδομή ενός δικτύου DVB-T. Σε αυτή την παράγραφο θα δούμε όλες τις πιθανές δομές που μπορούν εφαρμοσθούν σε ένα δίκτυο DVB-H. Τρεις διαφορετικές δομές δικτύου για το πρότυπο DVB-H περιγράφονται στη συνέχεια.

4.1 Κοινόχρηστο Δίκτυο με το DVB-T

Σε αυτήν την προσέγγιση το σύνολο των μεταδοτών που χρησιμοποιούνται για DVB-T χρησιμοποιείται επίσης για το DVB-H. Η μόνη αλλαγή που πρέπει να γίνει μέσα στις συσκευές αποστολής σημάτων (μεταδότες), είναι η ενσωμάτωση των bits DVB-H σηματοδότησης στις παραμέτρους του καναλιού TPS (Transmission Parameter Signalling channel). Τα κινητά τηλεοπτικά κανάλια που προέρχονται από την ενθυλάκωση των τμημάτων IP μοιράζονται τους ίδιους DVB-T πολυπλέκτες με άλλα επίγεια τηλεοπτικά κανάλια. Τα τηλεοπτικά κανάλια που προορίζονται για τα στάσιμα τηλεοπτικά σύνολα μπορούν να κωδικοποιηθούν σε MPEG-2, ενώ τα τηλεοπτικά κανάλια που προορίζονται για κινητές μικροσυσκευές μπορούν να κωδικοποιηθούν, για παράδειγμα σε MPEG-4. Αυτά τα κανάλια συνδυάζονται με την διαδικασία της πολυπλεξίας και στην συνέχεια στέλνονται ως μία ενιαία ροή η οποία μπορεί να διαβιβαστεί αφού πρώτα διαμορφωθεί.



Εικόνα 18 .Κοινόχρηστο δίκτυο με το DVB-T

Το βασικό στοιχείο σε αυτή την δομή είναι η ενθυλάκωση των πακέτων IP. Από τα εισερχόμενα πακέτα IP δημιουργούνται τμήματα MPE καθώς και τμήματα MPE-FEC εάν αυτά χρησιμοποιούνται. Επίσης εφαρμόζεται η τεχνική time slicing.

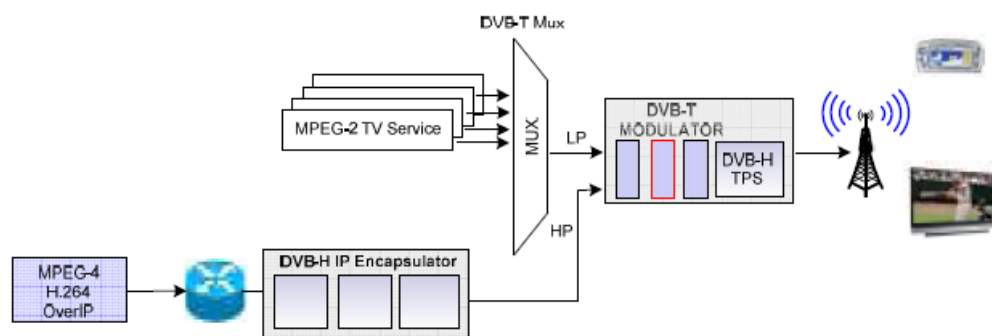
Αν και αυτή η δομή δικτύου επιτρέπει την επαναχρησιμοποίηση της υποδομής ενός δικτύου DVB-T το βασικό μειονέκτημα της είναι ότι χρησιμοποιείται ένας μοναδικός τρόπος μετάδοσης ,και αυτό γιατί οι δέκτες σε ένα δίκτυο DVB-T δεν υποστηρίζουν τον τρόπο μετάδοσης 4k. Κατά συνέπεια ,η διαθεσιμότητα των υπηρεσιών DVB-H

καθώς και η ποιότητα των υπηρεσιών θα υποβιβαστούν ειδικά για εσωτερικούς και κινούμενος σε υψηλές ταχύτητες δέκτες.

Επιπλέον, όπως αναφέραμε νωρίτερα η μετάδοση DVB-T προορίζεται για στάσιμα τερματικά με μια μεγάλη τοποθετημένη κεραία στις στέγες των κτιρίων που μπορούμε να τη κατευθύνουμε προς τη συσκευή αποστολής σημάτων(μεταδότη). Σε αντίθεση, το σήμα DVB-H πρέπει να φθάσει σε κινητές μικροσυσκευές με μικρές ενσωματωμένες κεραίες σε ένα ασταθές κινητό περιβάλλον. Μια άλλη απαίτηση είναι ότι τα κινητά τερματικά μπορεί να βρίσκονται σε εσωτερικούς χώρους. Λόγω αυτών των παραγόντων η απαραίτητη διαβιβασθείσα ισχύς στο DVB-H πρέπει να είναι πολύ υψηλότερη απ' ό,τι στο DVB-T. Το τελευταίο εξαρτάται επίσης από το ύψος κεραίων.

4.2 Ιεραρχικό δίκτυο DVB-H

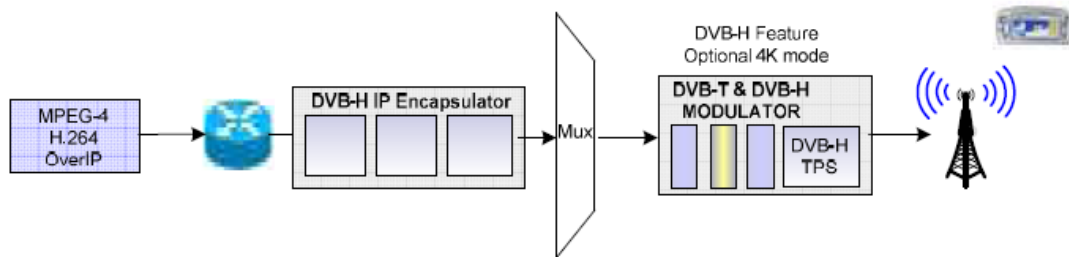
Το DVB-H μπορεί να συνυπάρχει με το DVB-T ιεραρχικά, σε αντίθεση με ένα shared DVB-T δίκτυο όπου οι ροές δεδομένων DVB-T και DVB-H συνδυάζονται και μεταδίδονται ως μία ροή, σε ένα hierarchical δίκτυο οι ροές DVB-T και DVB-H μεταδίδονται ξεχωριστά. Η διαμόρφωση στις δύο ροές γίνεται ιεραρχικά όπου μια ροή DVB-T διαμορφώνεται με χαμηλή προτεραιότητα και μία ροή DVB-H διαμορφώνεται με υψηλή προτεραιότητα.



Εικόνα 19 . Ιεραρχικό δίκτυο DVB-H

4.3 Αποκλειστικό δίκτυο DVB-H

Ένα αποκλειστικό δίκτυο DVB-H είναι αυτό στο οποίο οι υπηρεσίες DVB-H δεν μοιράζονται την πολυπλεξία TS με υπάρχοντες υπηρεσίες DVB-T. Επομένως η πλήρης πολυπλεξία είναι αφιερωμένη στην μεταφορά μόνο υπηρεσιών DVB-H.



Εικόνα 20 . Αποκλειστικό δίκτυο DVB-H

Ένα χαρακτηριστικό αποκλειστικό δίκτυο αποτελείται από διάφορες SFN (Single Frequency Network) περιοχές, οι οποίες δημιουργούν έπειτα ένα δίκτυο MFN (Multi Frequency Network) Το μέγιστο μέγεθος ενός τέτοιου SFN είναι τυπικά της τάξεως των δεκάδων χιλιομέτρων και εξαρτάται από διάφορες παραμέτρους, όπως το μέγεθος FFT και το guard interval και τα γεωγραφικά χαρακτηριστικά της περιοχής.

Σε αντίθεση με τις προηγούμενες δομές ένα αποκλειστικό δίκτυο DVB-H είναι το καταλληλότερο δεδομένου ότι μπορεί να προσαρμοστεί σε διαφορετικές συνθήκες. Μπορεί να είναι βέλτιστο για οποιαδήποτε επιθυμητή χωρητικότητα και κάλυψη χωρίς οποιονδήποτε περιορισμό που επιβάλλεται από το DVB-T. Επίσης ένα αποκλειστικό δίκτυο μπορεί να εκμεταλλευθεί τη χρησιμοποίηση του τρόπου μετάδοσης 4K και του in-depth interleaving που δίνει μια πρόσθετη ευελιξία για τη βελτιστοποίηση της απόδοσης δικτύων. Σύμφωνα με μία έρευνα διαπιστώθηκε ότι χρησιμοποιώντας την ίδια μεταδοθείσα ισχύ η περιοχή κάλυψης είναι μεγαλύτερη σε ένα dedicated DVB-H δίκτυο σε σύγκριση με ένα shared DVB-H δίκτυο[8].

Το πρακτικό μειονέκτημα αυτής της προσέγγισης είναι ο μικρός αριθμός χρηστών, ο οποίος έχει να επιβαρυνθεί τις δαπάνες. Αλλά αυτό το πρόβλημα μπορεί να λυθεί από την αύξηση της διαθεσιμότητας και της δημοτικότητας του DVB-H .

4.4 IP datacasting

Το IP Datacasting (IPDC) είναι τεχνολογία εκπομπής η οποία επιτρέπει τη διανομή ψηφιακού περιεχομένου σε ευρύ κοινό. Χρησιμοποιεί τα δίκτυα DVB-T για να εκπέμψει δεδομένα IP-based.

Ενώ οι πρώτες υπηρεσίες θα αφορούν τηλεοπτικό περιεχόμενο, μια ποικιλία τύπων περιεχομένου μπορούν να μεταδοθούν με τη βοήθεια του IPDC. Τυπική υπηρεσία είναι η μετάδοση πληροφοριών, όπου τα δεδομένα πρέπει να μεταδοθούν ταυτόχρονα σε πολλαπλούς αποδέκτες πχ. παιχνίδια, ήχος, εικόνα κλπ.

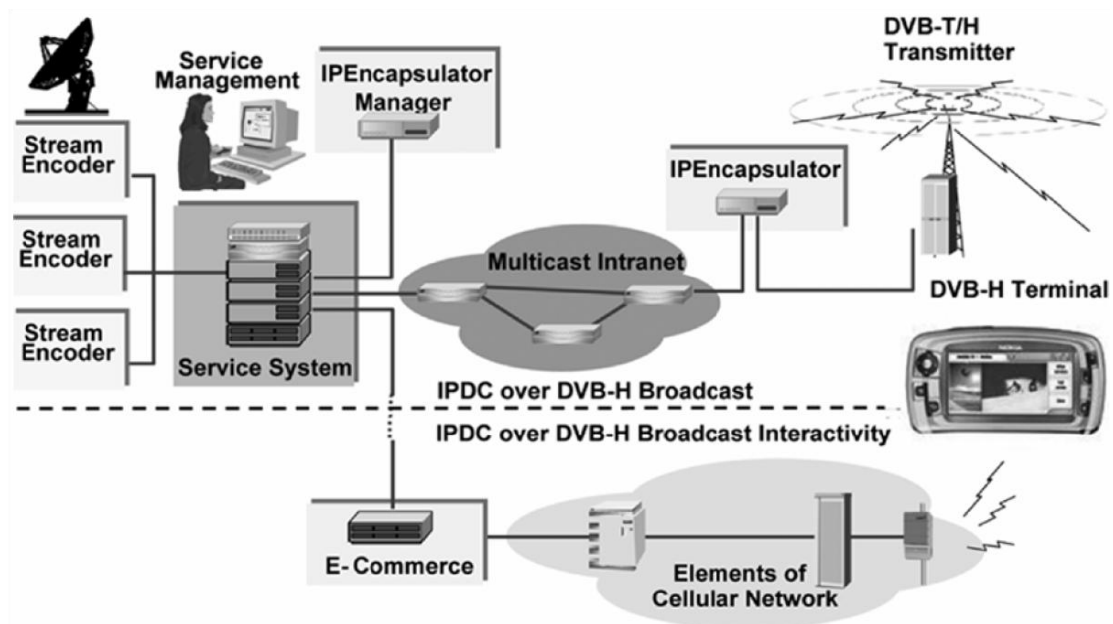
Το IPDC αποτελεί τεχνολογία εκπομπής (broadcast) ή πολυεκπομπής (one-to-many/multicast) και συνεπώς αποτελεί οικονομικό και αποδοτικό μέσο για πρόσβαση στο ευρύ κοινό. Για τους καταναλωτές τα οφέλη περιλαμβάνουν ένα νέο τρόπο πρόσβασης ψηφιακού περιεχομένου και υπηρεσιών, μέσω σταθερών ή κινητών τερματικών συσκευών.

Ο κύριος στόχος της προτυποποίησης του IPDC είναι η δημιουργία μιας καθολικής αγοράς, που θα υποστηρίζει υπηρεσίες, τερματικές συσκευές, εξοπλισμό και λογισμικό. Απαιτείται λοιπόν η διαλειτουργικότητα (interoperability) που θα οδηγήσει σε μια λύση end-to-end, η οποία θα βασίζεται σε ανοιχτά πρότυπα. Ένα σημαντικό χαρακτηριστικό του “κινητού” IPDC είναι η διαθεσιμότητα του αλληλεπιδραστικού καναλιού επιστροφής (return channel). Το πολυμεσικό περιεχόμενο εκπέμπεται μέσω διαύλου εκπομπής και παράλληλα η αλληλεπίδραση λαμβάνει χώρα μέσω του καναλιού επιστροφής του κυψελωτού δικτύου πχ. Αγορά περιεχομένου, ψηφοφορία κλπ. Η συνεργασία των κυψελωτών τεχνολογιών και των αντιστοίχων εκπομπής, επιτρέπει τη δημιουργία νέων υπηρεσιών πχ. mobile TV.

Το IP DataCasting over DVB-H αποτελεί λοιπόν, το τεχνολογικό κλειδί για τη ψηφιακή σύγκλιση μεταξύ παραδοσιακού τηλεοπτικού και τηλεπικοινωνιακού κόσμου. Η σύγκλιση επιτυγχάνεται διότι :

- Η τεχνολογία εκπομπής έχει ψηφιακό μέλλον
- Η ψηφιακή εκπομπή ανοίγει νέους διαύλους διανομής περιεχομένων
- Η τεχνολογία IPDC παρέχει τους πόρους που απαιτούνται για να καταστούν επιτυχείς οι διάφορες υπηρεσίες εκπομπής πολυμεσικών περιεχομένων
- Το ψηφιακό περιεχόμενο είναι ήδη διαθέσιμο και μπορεί να χρεωθεί σύμφωνα με την κατανάλωση.

Στην εικόνα 21 παρουσιάζεται ένα πλήρες σύστημα IPDC



Εικόνα 21 . Σύστημα IPDC

Αρχικά το σύστημα υπηρεσιών (service system) παράγει τις διάφορες ροές IP στο δίκτυο πχ. ροές εικόνας. Αυτές κατόπιν διανέμονται μέσω του multicast intranet στους IP encapsulators, οι οποίοι δίνουν ως έξοδο τις ροές DVB-H TS. Οι TS ροές προωθούνται στους μεταδότες DVB-T/H. Ο κόμβος e-Commerce μπορεί να χρησιμοποιηθεί για την τιμολόγηση των χρηστών. Το σύστημα IPDC μπορεί να περιλαμβάνει λειτουργίες, δανεισμένες από το GPRS (General Packet Radio Service) ή το UMTS (Universal Mobile Telecommunications System).

5. PCF

Το DVB Portable Content Format (PCF) είναι ένας τυποποιημένος τρόπος περιγραφής των διαδραστικών ψηφιακών τηλεοπτικών (iTV) υπηρεσιών. Παρέχει μία πλατφόρμα ανεξάρτητης οργάνωσης για την ενδο-επιχειρησιακή ανταλλαγή διαδραστικού περιεχομένου και συνεπώς είναι ένα μέσο για να αυξηθεί η διαλειτουργικότητα των εργαλείων δημιουργίας, των κεφαλών (head-end) συστημάτων και των δικτύων εκπομπής. Η PCF επιτρέπει την iTV περιγραφή υπηρεσίας να δημιουργηθεί ανεξάρτητα από μια συγκεκριμένη πλατφόρμα. Μια περιγραφή που συλλαμβάνεται χρησιμοποιώντας την PCF δεν προορίζεται για την πραγματική μετάδοση σε ένα ψηφιακό τηλεοπτικό δίκτυο και η μορφή της PCF το απεικονίζει αυτό. Μάλλον, είναι μια ενδιάμεση μορφή που πρέπει να μετατραπεί με κάποια ειδική πλατφόρμα απεικόνισης πριν από την μετάδοση. Η PCF θα επιτρέψει μια μεγάλη ποικιλία διαδραστικών υπηρεσιών να χρησιμοποιηθούν αποτελεσματικά σε πολλαπλές πλατφόρμες (target platform) συμπεριλαμβάνοντας το MHP και νόμιμες πλατφόρμες με το ελάχιστο κόστος επαναδημιουργίας. Μία περιγραφή χρησιμοποιώντας την PCF θα είναι πρακτική να μετατρέπεται από μια μηχανή παρέχοντας αναπτυσσόμενες εφαρμογές με ένα βαθμό ανεξαρτησίας από την συγκεκριμένη πλατφόρμα.

Η PCF είναι ένα μορφότυπο δεδομένων για την περιγραφή μιας διαδραστικής υπηρεσίας. Δεν είναι ένα εργαλείο δημιουργίας, εν τούτοις εργαλεία δημιουργίας μπορούν να αναπτυχθούν που θα γράφουν τα δεδομένα εξόδου σύμφωνα με τη PCF. Δεν είναι προδιαγραφή μιας τεχνολογίας που επεκτείνεται σε έναν ψηφιακό τηλεοπτικό δέκτη. Το DVB έχει ήδη μια τέτοια τεχνολογία, την εγχώρια πλατφόρμα πολυμέσων (MHP, Multimedia Home Platform). Η ανάπτυξη μιας πρότασης που παραδίδεται σε πολλαπλές πλατφόρμες στόχους μπορεί να αμφισβητηθεί λόγω παραλλαγών που υπάρχουν μεταξύ των πλατφορμών.

Αυτό μπορεί να οφείλεται σε:

- Φυσικούς περιορισμούς, όπως μια πλατφόρμα που δεν έχει διαδρομή επιστροφής.
- Εμπορικούς περιορισμούς, όπως οι εφαρμογές από έναν συγκεκριμένο πάροχο που δεν επιτρέπεται να έχουν πρόσβαση συγκεκριμένοι πόροι πλατφορμών.
- Λειτουργικούς περιορισμούς, όπως μέρος της οθόνης μένει δεσμευμένο από μία πλατφόρμα διαχείρισης για ειδική πλοήγηση προγραμμάτων ή μαρκάρισμα
- Έμφυτες διαφορές που υπάρχουν μεταξύ των πλατφορμών, όπως είναι οι διαθέσιμες γραμματοσειρές και οι κανόνες που αποδίδουν ένα κείμενο.

Αυτό έχει σαν επακόλουθο μερικές διαδραστικές υπηρεσίες να είναι προσωρινά δύσκολο να παραδώσουν μια πανομοιότυπη αίσθηση στον κάθε θεατή της πλατφόρμας. Εντούτοις, είναι γενικά δυνατό να προσφέρει, όπως καθορίζεται από τον πάροχο της διαδραστικής υπηρεσίας, μια αρκετά παρόμοια εμπειρία στον κάθε θεατή της πλατφόρμας.

Η PCF εφοδιάζει την βιομηχανία με ένα εργαλείο που διευκολύνει τη συνύπαρξη και την μετάβαση στο MHP επιτρέποντας την ανταλλαγή διαδραστικών τηλεοπτικών υπηρεσιών σε πολλαπλές πλατφόρμες.

Αυτό θα ελαχιστοποιήσει επίσης το συνολικό κόστος της επέκτασης των διαδραστικών ψηφιακών τηλεοπτικών υπηρεσιών σε πολλαπλές πλατφόρμες, αυξάνοντας την προσιτότητα μιας διαδραστικής τηλεοπτικής υπηρεσίας .

5.1 ΕΙΣΑΓΩΓΗ

Αυτό το κεφάλαιο παρέχει μια υψηλού επιπέδου επισκόπηση των διάφορων διαθέσιμων στοιχείων που χρησιμοποιούνται μέσα στην PCF, και πώς αυτά συνδυάζονται για να διαμορφώσουν μια πλήρη περιγραφή διαδραστικών τηλεοπτικών υπηρεσιών.

Η PCF είναι ένα διαδραστικό σχήμα περιγραφής τηλεοπτικών (iTV) υπηρεσιών σχεδιασμένο να είναι ανεξάρτητο πλατφορμας. Μια υπηρεσία που περιγράφεται χρησιμοποιώντας την PCF μπορεί να μετατραπεί αυτόματα σε ένα κατάλληλο σχήμα για οποιαδήποτε πλατφόρμα στόχο.

Η PCF επιτρέπει μια υπηρεσία iTV να περιγραφεί χρησιμοποιώντας μια υψηλού επιπέδου δηλωτική σύνταξη. Το μορφότυπο περιγράφει την προοριζόμενη εμπειρία του θεατή χωρίς να ορίσει ακριβώς πώς πρέπει να δοθεί σε μια πλατφόρμα στόχο. Αυτό σημαίνει ότι οποιαδήποτε αυτόματη διαδικασία μετατροπής έχει επαρκή ελευθερία να μετασχηματίσει την περιγραφή στο πρότυπο εκτέλεσης που απαιτείται από την πλατφόρμα στόχο(target platform). Για να επιτραπεί σε μια υπηρεσία να περιγραφεί πιο αποτελεσματικά και με τρόπο που να μεταβιβάζει κάλλιστα την πρόθεση του συντάκτη, το μορφότυπο χρησιμοποιεί ένα μηχανισμό παραπομπής. Αυτό επιτρέπει την περιγραφή υπηρεσιών να είναι ελαστικά διαχωρισμένες και δίνει στον κωδικοποιητή ιδιαίτερη ελευθερία να αποφασίσει τον καλύτερο τρόπο πακετοποίησης της υπηρεσίας ώστε να ταιριάζει στην πλατφόρμα στόχο.

Η μορφή της PCF απεικονίζει το γεγονός ότι δεν προορίζεται ως πραγματικό σχήμα μετάδοσης, αλλά μάλλον ως ενδιάμεσο σχήμα που θα χρειαστεί μετατροπή σε μια συγκεκριμένη πλατφόρμα παρουσίασης πριν από τη μετάδοση.

5.2 Επισκόπηση μιας περιγραφής υπηρεσίας PCF

Το παρακάτω είναι ένα παράδειγμα μιας απλής υπηρεσίας PCF:

```
<PCF xmlns="http://www.dvb.org/pcf/pcf">
  <Service name="hello_world">
    <String name="pcfSpecVersion" value="1.0"/>
    <URI name="firstScene" value="#hello_scene"/>
    <Scene name="hello_scene">
      <Component class="TextBox" name="hello_text">
        <Size name="size" value="100 40"/>
        <String name="content" value="Hello, World!"/>
      </Component>
      <OnEvent name="exit">
        <Trigger eventtype="KeyEvent">
          <UserKey name="key" value="VK_PREV"/>
        </Trigger>
        <ExitAction/>
      </OnEvent>
    </Scene>
  </Service>
</PCF>
```

Ενώ αυτή η υπηρεσία είναι πολύ μικρή, χρησιμοποιεί πολλές από τις βασικές δομές της PCF. Το στοιχείο **Service** παρέχει το σημείο εισόδου στην υπηρεσία, και η υπόλοιπη περιγραφή της υπηρεσίας είτε συμπεριλαμβάνεται είτε παραπέμπεται, μέσα από αυτό το στοιχείο.

Αμέσως μετά μέσα στο στοιχείο **Service** υπάρχει το στοιχείο **Scene**. Όλες οι υπηρεσίες PCF υποδιαιρούνται σε ένα ή περισσότερα **Scenes**, και κάθε **Scene** αντιπροσωπεύει έναν προορισμό πλοήγησης μέσα στην υπηρεσία. Η οπτική εμφάνιση ενός **Scene** καθορίζεται από το σύνολο των **Components** που το **Scene** περιέχει. Σε αυτήν την περίπτωση υπάρχει ένα μοναδικό **Component**, το **TextBox**, το οποίο χρησιμοποιείται για να επιδείξει το μήνυμα "Hello, World!". Η PCF ορίζει ένα σύνολο εργαλείων των τυποποιημένων **components** που μπορεί να περιλαμβάνονται μέσα σε μια περιγραφή ενός **Scene** για την περιγραφή της επιθυμητής εμφάνισης και συμπεριφοράς.

Όλοι οι τύποι **component** έχουν ένα αριθμό από **Properties**(ιδιότητες). Για το **component TextBox**, έχουν οριστεί μόνο το **μέγεθός** και το **περιεχόμενό** του. Ένα **TextBox** έχει διάφορες άλλες ιδιότητες που δεν έχουν οριστεί και σε αυτήν την περίπτωση θα χρησιμοποιηθούν οι προκαθορισμένες τιμές τους.

Το τελευταίο στοιχείο μέσα στο στοιχείο **Scene** περιγράφει έναν χειριστή γεγονότος, **OnEvent**. Για αυτό το στοιχείο **Scene** ο χειριστής του γεγονότος αποκρίνεται στο θεατή πιέζοντας το εικονικό κλειδί "VK_PREV" το οποίο θα χαρτογραφηθεί σε ένα

συγκεκριμένο κλειδί τηλεχειρισμού πλατφορμών, π.χ. "Backup", "Back", "Previous", "Cancel". Ο χειριστής γεγονόςτος περιέχει μια πράξη , **ExitAction**, η οποία καθοδηγεί την υπηρεσία στην έξοδο.

Ενώ αυτή είναι μια πολύ απλή υπηρεσία επεξηγεί τις βασικές πτυχές μιας περιγραφής υπηρεσίας PCF :

- *Components*- οι δομικές μονάδες μιας περιγραφής υπηρεσιών PCF.
- *Content*- ελέγχεται και παρουσιάζεται χρησιμοποιώντας PCF components.
- *Behaviour* - η απάντηση στα γεγονότα που παράγονται στο χρόνο εκτέλεσης.

5.3 ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ

Είναι τα δομικά στοιχεία της περιγραφής μιας υπηρεσίας PCF. Πολλά components είναι οπτικά "widgets" και χρησιμοποιούνται για να περιγράψουν την οπτική εμφάνιση της υπηρεσίας σε ένα συγκεκριμένο σημείο, παραδείγματος χάριν TextBox, Video και Menu. Υπάρχουν επίσης και μη οπτικά **component**, τα οποία μπορούν να χρησιμοποιηθούν για την παρουσίαση μη οπτικού περιεχομένου ή της παροχής άλλων πτυχών της λειτουργίας μιας υπηρεσίας, παραδείγματος χάριν Audio, Stream και Random.

Όλοι οι τύποι **component** μπορούν να δηλωθούν χρησιμοποιώντας ένα γενικό **component item**. Αυτό έχει μια ιδιότητα *τάξης* (class property) για να διευκρινίσει τον τύπο του **component** και μια ιδιότητα παρόχου (provider property) για να προσδιορίσει το σώμα που έχει το συγκεκριμένο **component**. Τα component που περιγράφονται έχουν τον πάροχο "dvn.org" Ωστόσο για τέτοια **component** η ιδιότητα του παρόχου μπορεί να παραλειφθεί στην δήλωση τους, δεδομένου ότι "dvn.org" είναι η default τιμή για αυτήν την ιδιότητα

Κάθε κατηγορία component έχει ένα σύνολο σχετικών ιδιοτήτων, και ο καθορισμός αυτών των ιδιοτήτων καθορίζει πως θα εμφανίζεται μια συγκεκριμένη υπόσταση ενός component.. Το παρακάτω είναι ένα παράδειγμα μιας δήλωσης ενός component DVB TextBox.

```
<Component class="Textbox" name="MainText">  
  <Size name="size" value="100 40"/>  
  <Color name="fillcolor" value="#0000AA"/>  
  <Color name="textcolor" value="#FFFFFF"/>  
  <String name="content" value="Hello, World!"/>  
</Component>
```

Πρέπει να σημειωθεί ότι η ιδιότητα του παρόχου δεν έχει δηλωθεί ρητά στη δήλωση, δεδομένου ότι ο "dwb.org" είναι η προκαθορισμένη τιμή για αυτή την ιδιότητα. Σε αυτήν την περίπτωση ορίζονται τέσσερις ιδιότητες του στοιχείου Textbox: το μέγεθος, το χρώμα παρασκηνίου(background colour), το χρώμα του κειμένου και η περιεχόμενη συμβολοσειρά. Πολλές ιδιότητες των στοιχείων μπορούν να αλλάξουν δυναμικά κατά τον χρόνο εκτέλεσης. Στο παραπάνω παράδειγμα το περιεχόμενο του textbox μπορεί να ανανεωθεί και να εμφανίσει ένα διαφορετικό κείμενο ή μπορεί να αλλάξει το χρώμα του κειμένου για παρουσιαστεί με διαφορετικό τρόπο.

Η χρησιμοποίηση του γενικού *component* παρέχει διάφορα πλεονεκτήματα σχετικά με την επεκτασιμότητα της PCF, και την δυνατότητα να δηλωθούν συνήθη *component* που χρησιμοποιούν την ίδια σύνταξη με τα τυποποιημένα *component* της PCF. Ωστόσο ένας αριθμός τυποποιημένων component που υπάρχουν μπορούν να δηλωθούν σε μία συγκεκριμένη κατηγορία. Αυτά τα components είναι γνωστά ως *schema components* και έχουν πάντα τον πάροχο "dwb.org". Παραδείγματος χάριν, το ακόλουθο παράδειγμα περιγράφει ένα TextBox που είναι ισότιμο με το *components* της κατηγορίας TextBox:

```
<TextBox name="MainText">  
  <Size name="size" value="100 40"/>  
  <Color name="fillcolor" value="#0000AA"/>  
  <Color name="textcolor" value="#FFFFFF"/>  
  <String name="content" value="Hello, World"/>  
</TextBox>
```

Οι διάφοροι τύποι component μπορούν να οριστούν ώστε να έχουν ενδογενή συμπεριφορά (intrinsic behaviour) ώστε να μπορούν να μεταβάλλονται σε περίπτωση που συμβαίνουν εξωτερικά γεγονότα. Για παράδειγμα το στοιχείο Menu θα μεταβληθεί όταν πατηθεί ένα συγκεκριμένο κουμπί αλλάζοντας την επισήμανση του στοιχείου menu στη οθόνη.

Τα components μπορούν επίσης να οριστούν ώστε να παράγουν γεγονότα για να επισημάνουν κάποια εσωτερική αλλαγή. Αυτό παρέχει έναν μηχανισμό που επιτρέπει στην περιγραφή υπηρεσίας να περιέχει κάποια συγκεκριμένη συμπεριφορά πέρα από αυτήν που καθορίζεται στις προδιαγραφές του component.

5.3.1 Το Δομικό Στοιχείο Υπηρεσία

Το *Service* component είναι μια μορφή *explicit layout container* μέσα στην οποία τα δομικά στοιχεία *scene* και άλλα στοιχεία που απαιτούνται να υπάρχουν μεταξύ διαφορετικών *Scenes* μπορούν να περιγραφούν, παραδείγματος χάριν, ένα στοιχείο *IntegerVar* που χρησιμοποιείται για την αποθήκευση της υψηλότερης βαθμολογίας. Εκτός από αυτό, το στοιχείο *Service* παρέχει το σημείο εισόδου στην υπηρεσία διευκρινίζοντας το αρχικό *Scene* που παρουσιάζεται.

5.3.2 Το Δομικό Στοιχείο Σκηυή

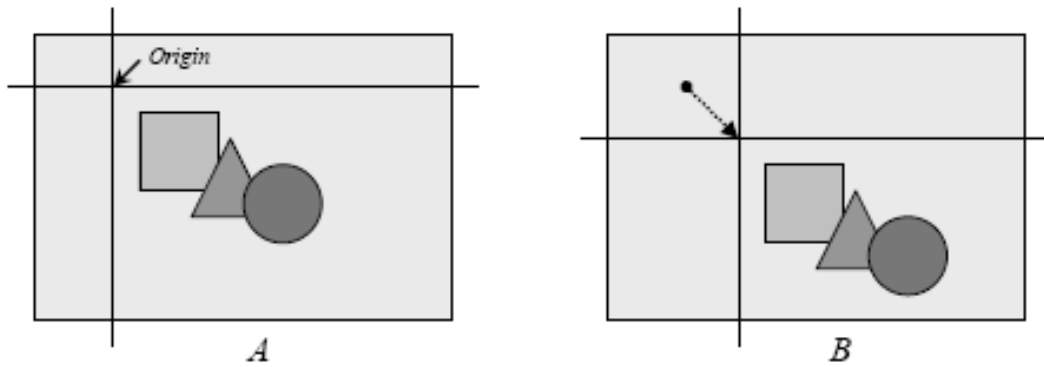
Το *scene* component είναι επίσης μια μορφή *explicit layout container* μέσα στην οποία μπορούν να περιγραφούν άλλα στοιχεία. Κανονικά, η διάρκεια ζωής όλων των στοιχείων μέσα στο *Scene* είναι η ίδια με τη διάρκεια ζωής του ίδιου του στοιχείου *Scene*. Μια μετάβαση από το στοιχείο *Scene* θα προκαλέσει την εξαφάνιση όλων των θυγατρικών στοιχείων και της συνοδευτικής κατάστασης. Για αυτόν τον λόγο, οποιαδήποτε στοιχεία που χρειάζονται να υπάρχουν μεταξύ των στοιχείων *Scenes* (για παράδειγμα μια εικόνα υποβάθρου ή μια μεταβλητή για την αποθήκευση υψηλής βαθμολογίας) πρέπει να περιληφθούν στο στοιχείο *Service*.

5.3.3 Το Δομικό Στοιχείο Σχεδιάγραμμα

5.3.3.1 Αναλυτικό Σχεδιάγραμμα

Ένα αναλυτικό σχεδιάγραμμα (*explicit layout*) περιγράφεται χρησιμοποιώντας το στοιχείο *ELC* (*Explicit Layout Container*). Αυτό το στοιχείο δεν έχει δικιά του οπτική εμφάνιση. Είναι αντ' αυτού ένα σχεδιάγραμμα για άλλα στοιχεία τα οποία πρέπει να τοποθετηθούν σε μια ακριβή θέση.

Ένα στοιχείο *ELC* έχει μια ιδιότητα προέλευσης (*origin property*) αλλά δεν έχει μέγεθος. Τα στοιχεία που δηλώνονται μέσα σε ένα στοιχείο *ELC* έχουν θέσεις που έχουν ορισθεί σε σχέση με την προέλευση. Συνεπώς η μετακίνηση του γονικού στοιχείου *ELC* από την θέση προέλευσης θα προκαλέσει την μετακίνηση όλων των θυγατρικών στοιχείων στην ίδια σχετική απόσταση. Αυτό παρουσιάζεται στην εικόνα 22. Το διάγραμμα Α παρουσιάζει τρία στοιχεία τα οποία περιέχονται μέσα σε ένα στοιχείο *ELC* και παρουσιάζεται και η θέση προέλευσης. Στο διάγραμμα Β το στοιχείο *ELC* έχει μετακινηθεί από τη θέση προέλευσης το οποίο έχει σαν αποτέλεσμα να μετακινηθούν και όλα τα θυγατρικά στοιχεία στη ίδια απόσταση .



Εικόνα 22. Μετακινώντας το δομικό στοιχείο ELC

Όταν η ιδιότητα ορατότητας (visible property) ενός στοιχείου ELC είναι “true” τότε όλα τα θυγατρικά του στοιχεία είναι ορατά. Παρόμοια όταν η ιδιότητα της ορατότητας ενός στοιχείου ELC είναι “false” κανένα από τα θυγατρικά του στοιχεία δεν είναι ορατό. Υπάρχει μια παραλλαγή του στοιχείου ELC το οποίο ονομάζεται *StaticELC* και δεν υποστηρίζει την τροποποίηση θέσης κατά την διάρκεια του χρόνου. Ενώ το στοιχείο *StaticELC* είναι λιγότερο ικανό από το ELC προσφέρει μεγαλύτερη φορητότητα.

5.3.3.2 Σχεδιάγραμμα ροής.

Ένα διάγραμμα ροής (Flow layout) περιγράφεται χρησιμοποιώντας έναν συνδυασμό από *Flow components* και *flow layout container components*.

Τα *Flow components* χρησιμοποιούνται για να διαχειρίζονται το περιεχόμενο που εμφανίζεται, το οποίο μπορεί να είναι ένα μίγμα κειμένου και άλλων στοιχείων της PCF.

Τα *flow layout container components* χρησιμοποιούνται για να τοποθετήσουν το περιεχόμενο που περιγράφεται από ένα *Flow component*. Αυτό επιτυγχάνεται χρησιμοποιώντας κανόνες “ροής”, έτσι ώστε το περιεχόμενο που περιγράφεται από ένα *Flow component* να παρουσιάζεται χωρίς να απαιτείται από τον συντάκτη υπηρεσίας να διευκρινίσει τις ακριβείς θέσεις. Αυτός ο μηχανισμός είναι παρόμοιος με τον τρόπο που το περιεχόμενο μέσα σε μια σελίδα HTML «ρέει» σε ένα HTML browser.

Υπάρχουν τρεις τύποι των flow container component:

- Ένα **SFC** (*Scroll Flow Container*) component εμφανίζει το περιεχόμενο ροής σε ένα ορθογώνιο σταθερού πλάτους, ύψους και προέλευσης. Το περιεχόμενο ροής που υπερβαίνει τα όρια του SFC μπορεί να εμφανιστεί με κύλιση του περιεχόμενου επάνω και κάτω.
- Ένα **PFC** (*Page Flow Container*) component εμφανίζει το περιεχόμενο ροής σε ένα ορθογώνιο σταθερού πλάτους, ύψους και προέλευσης. Το περιεχόμενο ροής που υπερβαίνει τα όρια του PFC μπορεί να φανεί με «σελιδοποίηση» του περιεχομένου μπροστά και πίσω.
- Ένα **TFC** (*Truncate Flow Container*) component εμφανίζει το περιεχόμενο ροής σε ένα ορθογώνιο σταθερού πλάτους και προέλευσης, αλλά μεταβλητού ύψους. Το TFC μπορεί να επεκταθεί, μέσα στα όρια που ορίζει ο συντάκτης, μόνο στην προς τα κάτω κατεύθυνση που απαιτείται για να προσαρμοστεί το περιεχόμενο ροής. Το περιεχόμενο ροής που υπερβαίνει το μέγιστο μέγεθος του επεκτάσιμου δια-γράμματος θα πρέπει να περικοπεί.

Το περιεχόμενο που παρουσιάζεται σε οποιοδήποτε *flow container* component περιγράφεται χρησιμοποιώντας κάποιο από τα ακόλουθα flow components.

- Ένα **Flow** component μπορεί να περιέχει ένα μίγμα κειμένου και άλλων οπτικών στοιχείων PCF.
- Ένα **Table** component είναι μια ειδική περίπτωση ενός **Flow** όπου το περιεχόμενο ροής οργανώνεται σε μια συνοπτική δομή.
- Ένα **TextFlow** component είναι μια περιορισμένη μορφή ενός **Flow** που περιορίζεται στο να περιέχει απλό μαρκαρισμένο κείμενο.

5.3.4 Το Δομικό Στοιχείο Return Path

Υπάρχουν πολλές περιπτώσεις όπου μια υπηρεσία πρέπει να στείλει μερικά δεδομένα πίσω στο head-end σύστημα ή στον πάροχο περιεχομένου. Αυτό επιτυγχάνεται χρησιμοποιώντας όπου είναι δυνατόν κάποιο είδος ενός μονοπατιού επιστροφής (return path.). Αυτά εμπίπτουν σε δύο κατηγορίες:

- Always On Return Path (π.χ. xDSL / Cable Modem συστήματα).
- Intermittent / On Request / Dial Up

Σαφώς υπάρχουν και πλατφόρμες όπου δεν έχουν *return path components* και σε τέτοιες περιπτώσεις υπάρχουν εναλλακτικές μέθοδοι με τις οποίες μπορεί να επιτευχθεί η λειτουργικότητα με ένα περιορισμένο σύνολο διαδρομών επιστροφής. Για παράδειγμα ένας θεατής που χρησιμοποιεί μια συγκεκριμένη υπηρεσία θα μπορούσε να του ζητηθεί να τηλεφωνήσει σε έναν αριθμό, ή να στείλει ένα γραπτό μήνυμα χρησιμοποιώντας το κινητό του τηλέφωνο, το οποίο θα μπορούσε να χρησιμοποιηθεί με σκοπό την συγκέντρωση στοιχείων από μία διαδραστική υπηρεσία.

Ένα τέτοιο παράδειγμα μπορεί να είναι μια διαδραστική υπηρεσία ψηφοφορίας. Η ιδανική και επιθυμητή συμπεριφορά θα ήταν η υπηρεσία να επέστρεφε την ψήφο του χρήστη πάντα μέσω ενός dial up return path. Εντούτοις, σε πλατφόρμες όπου δεν υπάρχουν τέτοια return paths, ή δεν είναι διαθέσιμα, ο χρήστης θα μπορούσε να προτραπεί να στείλει ένα γραπτό μήνυμα SMS δηλώνοντας το "A", "B" ή "C" σε έναν δεδομένο αριθμό χρησιμοποιώντας το κινητό του τηλέφωνο.

Η τυποποιημένη εφαρμογή ενός return path στηρίζεται πάνω σε τρία αντικείμενα: σε ένα ReturnPath component, σε ένα Transaction component και σε ένα *transfer collection*. Ένα τέταρτο component, το component Indicate, είναι μια περικομμένη έκδοση του στοιχείου ReturnPath, του οποίου σκοπός είναι να επιτρέπει την λειτουργία return path όταν μια σύνδεση υπάρχει αλλά δεν υπάρχει κάποια συναλλαγή όπως χρησιμοποιείται σε μια απλή εφαρμογή ψηφοφορίας. Ένα πέμπτο στοιχείο, το *SecureReturnPath* επιτρέπει την πραγματοποίηση ασφαλούς μεταφοράς δεδομένων.

Τα δεδομένα transfer collection καθορίζουν μια ακολουθία πληροφοριών που μεταφέρονται πάνω σε ένα return path. Μόνο δεδομένα τα οποία βρίσκονται σε αλληλουχία μπορούν να μεταφερθούν πάνω σε ένα return path. Δεδομένα τα οποία βρίσκονται σε αλληλουχία είναι ένα περιορισμένο σύνολο τύπων δεδομένων κατάλληλα για μεταφορά μέσω ενός return path, για παράδειγμα ακέραιοι αριθμοί, συμβολοσειρές, μεταβλητές, ημερομηνίες αλλά **όχι** ορθογώνια πλαίσια, κουμπιά, επιλογές κ.λ.π. Οι προδιαγραφές της PCF καθορίζουν εάν ένας τύπος component είναι ή όχι αλληλουχία δεδομένων. Στην αρχική έκδοση της PCF δεν απαιτείται μεταφορά των γραφικών μέσω ενός return path και επομένως οι εικόνες δεν ορίζονται ως αλληλουχία δεδομένων.

Το ReturnPath component ενσωματώνει το ίδιο το return path και διαχειρίζεται την πραγματική διαδικασία μεταφοράς δεδομένων. Αυτό το component δεν έχει καμία οπτική εμφάνιση και εφαρμόζει τη λειτουργία return path και για τις δύο πλατφόρμες του που είναι οι always-on (πάντα ενεργή) και dial-up.

Οι ιδιότητές του ReturnPath component περιέχουν πληροφορίες για τον καθορισμό του εξυπηρετητή εφαρμογής και της παρούσας κατάστασης της return path σύνδεσης (π.χ. "closed", "open", "opening", "closing")

Κατά την διάρκεια της εκτέλεσης το ReturnPath component επιτυγχάνει μια αναφορά στο component Transaction το οποίο περιέχει την πηγή και/ή τα transfer collection δεδομένα προορισμού.

Το component Transaction ενσωματώνει την κατάσταση μιας πραγματικής διαδικασίας μεταφοράς δεδομένων και παράγει ένα γεγονός έκβασης που καθορίζει την επιτυχία ή την αποτυχία της συναλλαγής δεδομένων.

5.3.5 Προσαρμοσμένα Δομικά Στοιχεία

Το σύνολο των τυποποιημένων component κατηγοριών από το DVB μπορεί να αυξηθεί με τη δημιουργία *custom components*. Ένα προσαρμοσμένο δομικό στοιχείο μπορεί να δημιουργηθεί για μια πλατφόρμα στόχο ώστε εκμεταλλευτεί τις ικανότητες της πλατφόρμας που δεν είναι διαθέσιμες μέσω της χρήση των τυποποιημένων components της PCF. Όταν χρησιμοποιείται ένα προσαρμοσμένο δομικό στοιχείο θα πρέπει πάντα να δηλώνεται ο πάροχος της ιδιότητας.

Οι περιγραφές υπηρεσιών PCF που περιέχουν προσαρμοσμένα δομικά στοιχεία *custom components* θα λειτουργήσουν μόνο στις πλατφόρμες στόχους που εφαρμόζουν αυτά τα components, και έτσι είναι πιθανό να είναι λιγότερο φορητά από τις υπηρεσίες PCF που χρησιμοποιούν μόνο τα τυποποιημένα στοιχεία .

5.4 Περιεχόμενο

Το περιεχόμενο ρυθμίζεται μέσα σε μια περιγραφή υπηρεσίας PCF χρησιμοποιώντας ένα σύνολο τυποποιημένων τύπων δεδομένων της PCF ως εξής:

- **Primitive types** - βασικός τύπος δεδομένων όπως ακέραιοι αριθμοί και σταθερές Boolean
- **Core types**- τύποι δεδομένων όπως το μέγεθος, το χρώμα το MarkedUpText, οι οποίοι ενσωματώνουν μία ή περισσότερες **Primitive τιμές δεδομένων** .
- **Octet data**- αυτά είναι κωδικοποιημένα δυαδικά στοιχεία. Για παράδειγμα octet δεδομένα μπορούν να χρησιμοποιηθούν για να χειριστούν δεδομένα εικόνας, ροές δεδομένων και δεδομένα χαρακτήρων σε διάφορες κωδικοποιήσεις.
- **Compound Types**- αυτοί οι τύποι περιέχουν διάφορες άλλες τιμές, που είναι διαταγμένες και είναι προσβάσιμες με κάποιο συγκεκριμένο τρόπο. Για παράδειγμα ζεύγη από τιμές-κλειδιά όπου η τιμή κάποιου στοιχείου μπορεί να φανεί χρησιμοποιώντας το αντίστοιχο κλειδί.

Η PCF παρέχει τα μέσα ώστε τα αντικείμενα να βρίσκονται συγκεντρωμένα σε μορφή δομών και πινάκων.

Δομές όπως οι τύποι δεδομένων Collection παρέχουν ένα μηχανισμό ώστε τα συναφή δεδομένα να είναι προσβάσιμα ως μία ενιαία λογική μονάδα. Οι πίνακες επιτρέπουν δεδομένα του ίδιου τύπου να είναι ομαδοποιημένα ώστε να μπορούν εύκολα να αριθμηθούν και να επαναληφθούν.

5.5 Συμπεριφορά

Η συμπεριφορά κατά τον χρόνο εκτέλεσης που συμβάλλει στην εμπειρία του θεατή μιας υπηρεσίας PCF είναι ένα καθοδηγούμενο γεγονός. Όταν συμβαίνει ένα γεγονός μπορεί να «καταναλωθεί» από ένα δομικό στοιχείο με συνέπεια την εκτέλεση της συμπεριφοράς που καθορίζεται στο τμήμα των προδιαγραφών του δομικού στοιχείου.

Παραδείγματος χάριν, το δομικό στοιχείο Menu ορίζεται έτσι ώστε να ανταποκρίνεται στο πάτημα των πλήκτρων UP/DOWN με την κίνηση του φωτισμένου πλαισίου πάνω-κάτω στα menuItemems που περιέχει. Ο συντάκτης υπηρεσιών δεν χρειάζεται να περιγράψει αυτή την συμπεριφορά.

Εναλλακτικά, ένα γεγονός μπορεί να «καταναλωθεί» από ένα συνήθη χειριστή που δηλώνεται μέσα στην περιγραφή υπηρεσιών. Αυτό μπορεί να χρησιμοποιηθεί για την δημιουργία μίας συνήθους συμπεριφοράς που δεν είναι ενδογενές τμήμα της λειτουργίας ενός διαθέσιμου δομικού στοιχείου ,

Παραδείγματος χάριν, όταν εστιάσουμε σε ένα menuItemem που περιέχεται σε ένα δομικό στοιχείο Menu αυτό θα παράγει ένα γεγονός "OnFocus", το οποίο θα μπορούσε να χρησιμοποιηθεί για να οδηγήσει κάποια συνήθη "roll-over" συμπεριφορά, έτσι ώστε οι πληροφορίες να παρουσιάζονται μέσα σε ένα TextBox.

Πολλά από τα χαρακτηριστικά γνωρίσματα τα οποία είναι διαθέσιμα για τον καθορισμό κάποιας συμπεριφοράς μέσα στην PCF, προέρχονται από τα χαρακτηριστικά μιας statemachine τα οποία είναι διαθέσιμα μέσα στα UML statecharts.

Ένα σύνολο καταστάσεων μπορεί να οριστεί παράλληλα με ένα σύνολο κανόνων για να καθοριστούν οι συνθήκες για τις μεταβάσεις μεταξύ αυτών των καταστάσεων. Ένα σύνολο ενεργειών μπορεί έπειτα να καθοριστεί για την εκτέλεση ως απόκριση σε συγκεκριμένα γεγονότα, ή ως απόκριση σε μια συγκεκριμένη μετάβαση σε κάποια κατάσταση .

5.5.1 Γεγονότα

Υπάρχουν τρεις κατηγορίες γεγονότων:

- **System events** - γεγονότα τα οποία εμφανίζονται χωρίς την παρέμβαση του θεατή.
- **Component events** - δημιουργούνται από ένα στοιχείο για να υποδηλώσουν μια αλλαγή σε κάποια εσωτερική τους κατάσταση .
- **User input events** - δημιουργούνται ως απόκριση στον θεατή που κάνει κάποια ενέργεια. Για παράδειγμα το πάτημα ενός κουμπιού στο τηλεχειριστήριο

Ανάλογα με τον τύπο του γεγονότος και της κατάστασης της υπηρεσίας, ένα γεγονός που παράγεται θα είναι διαθέσιμο για κατανάλωση σε ορισμένα στοιχεία μέσα στην περιγραφή υπηρεσίας σύμφωνα με το καθορισμένο σύνολο κανόνων διάδοσης του γεγονότος.

5.5.2 Γλώσσα Δράσης

Η PCF περιλαμβάνει μια γλώσσα δράσης (action language) για να διευκρινίσει ακολουθίες ενεργειών που εκτελούνται είτε ως άμεση απόκριση σε ένα γεγονός που εμφανίζεται ,είτε ως αποτέλεσμα μιας μεταβατικής κατάστασης που προκαλείται από ένα γεγονός. Η Action Language της PCF αποτελείται από *actions* *μεταβλητές*, *συνθήκες* και *έλεγχους*. Αυτά μπορούν να συνδυαστούν για να διαμορφώσουν απλά scripts.

Τα actions είναι ουσιαστικά εντολές που απευθύνονται σε συγκεκριμένα στοιχεία που περιλαμβάνουν ενδεχομένως τα ενεργά στοιχεία *scene* και *service*. Για παράδειγμα, μια ενέργεια μπορεί να θέσει κάποιο ερώτημα σε κάποια ιδιότητα ενός στοιχείου ή να θέσει κάποια συγκεκριμένη τιμή σε αυτή την ιδιότητα .

Εκτός από την ανάγνωση και τη τροποποίηση των ιδιοτήτων κάποιου στοιχείου η *action language* της PCF υποστηρίζει ενέργειες για την παραγωγή των *στοιχείων scene transitions* εκτελώντας απλές μαθηματικές και λογικές διαδικασίες, διαχειρίζοντας σειρές και εκτελώντας συγκεκριμένες εντολές .

- *Variables* : χρησιμοποιούνται για να αποθηκεύσουν και να ανακτήσουν δεδομένα που απαιτούνται κατά την διάρκεια της εκτέλεσης μιας σειράς ενεργειών.
- *Conditions* : χρησιμοποιούνται για να καταστήσουν την εκτέλεση μιας συγκεκριμένης ακολουθίας εξαρτώμενη από κάποια Boolean συνθήκη που

ικανοποιείται, επιτρέποντας μια επιλογή εκτελούμενων διαδρομών να προσδιορισθούν

- *Controls* : χρησιμοποιούνται από κοινού με τις *Conditions* για να επηρεαστεί περαιτέρω η ροή της εκτέλεσης μιας ακολουθίας ενεργειών διευκρινίζοντας βρόχους και άλματα. Για παράδειγμα, η μετάβαση στην ένδειξη "Game Over" εμφανίζεται μόνο όταν η τιμή το στοιχείου IntegerVar που περιέχει τον αριθμό των εναπομεινάντων ζωών ίσο με μηδέν.

Εξ ορισμού, ένα *action* εκτελείται στο πλαίσιο του γονικού component και μπορεί «να δει» (δηλ. να έχει πρόσβαση και να τροποποιήσει) μόνο τα στοιχεία που είναι επίσης στο ίδιο πλαίσιο. Υπάρχουν μερικές καταστάσεις όπου είναι απαραίτητο να υπάρχει πρόσβαση σε στοιχεία που είναι έξω από αυτό το κανονικό πλαίσιο. Παραδείγματος χάριν, ένα *action* μπορεί να χρειαστεί να έχει πρόσβαση σε ένα στοιχείο integerVar το οποίο έχει καθοριστεί μέσα στο service component για να ενημερώσει ένα αποτέλεσμα παιχνιδιού που πρέπει να υπάρχει μεταξύ των στοιχείων scenes. Η PCF παρέχει ένα μηχανισμό υπό μορφή *στοιχείων παραμέτρων (parameter items)*, ώστε να επιτύχει αυτό.

Το ακόλουθο παράδειγμα δείχνει την χρήση της action language:

```
<Scene name="scene">
  <Collection name="variables">
    <IntegerVar name="counter">
      <Integer name="value" value="1"/>
    </IntegerVar>
  </Collection>
  <OnEvent>
    <Trigger eventtype="KeyEvent">
      <UserKey name="key" value="VK_ENTER"/>
    </Trigger>
    <ActionLanguage name="jumpCounter">
      variables.counter.value += 10;
    </ActionLanguage>
  </OnEvent>
</Scene>
```

5.6 Δομή μιας περιγραφής υπηρεσίας PCF

Για μια απλή υπηρεσία μπορεί να είναι σκόπιμο να είναι όλες οι πτυχές της περιγραφής της υπηρεσίας σε ένα έγγραφο PCF, και όλες οι σχέσεις μεταξύ των στοιχείων να περιγράφονται ρητά μέσω της τοποθέτησης των στοιχείων ιεραρχικά στην περιγραφή .

Για μεγαλύτερες υπηρεσίες, εντούτοις, μπορεί να μην είναι εφικτό να τοποθετηθεί ολόκληρη η περιγραφή υπηρεσίας σε ένα ενιαίο κείμενο PCF για λόγους αποδοτικής περιγραφής. Η PCF παρέχει ελαστικούς μηχανισμούς παραπομπής για την προσαρμογή ακραίων δομών για την περιγραφή μιας υπηρεσίας.

5.6.1 Παραπομπή

Με σκοπό την παροχή ευελιξίας σε κάποιο τμήμα η PCF χρησιμοποιεί ένα σχήμα παραπομπής για να επιτραπεί σε ένα στοιχείο να συμπεριλάβει άλλα «μακρινά» στοιχεία της PCF, ή ομάδες στοιχείων PCF. Μια τέτοια αναφορά μπορεί να περιγραφεί χρησιμοποιώντας την ιδιότητα *href*. Η τιμή αυτής της ιδιότητας είναι μία περιγραφή της διαδρομής στο στοιχείο που παραπέμπεται.

Στο ακόλουθο παράδειγμα, το TextBox δεν συμπεριλαμβάνει το περιεχόμενο κείμενο άμεσα, αλλά αντ' αυτού περιέχει μια παραπομπή. Αυτή η παραπομπή δείχνει έπειτα το πραγματικό περιεχόμενο κειμένου, το οποίο δηλώνεται χωριστά:

```
<Component type="TextBox" name="hello_text">  
  <String name="content" href="#../mycontent/msg"/>  
</Component>
```

```
<Collection name="mycontent">  
  <String name="msg" value="Hello, World!"/>  
</Collection>
```

Τα πλεονεκτήματα μιας τέτοιας παραπομπής είναι :

- Το ίδιο σύνολο στοιχείων μπορεί να δημιουργηθεί με πολυάριθμα σύνολα περιεχομένου. Ένα σύνολο στοιχείων μπορεί να χρησιμοποιηθεί σαν παρουσίαση "πρότυπου", η οποία μπορεί να συνδυαστεί με διαφορετικά σύνολα περιεχομένου σε διαφορετικά scenes.

- Ένα σύνολο στοιχείων που δηλώνεται μία φορά μπορεί να συμπεριληφθεί μέσα σε πολλαπλά scenes. Αυτό επιτρέπει στο δομικό στοιχείο scene το οποίο περιέχει «σταθερό κείμενο» να ξαναχρησιμοποιηθεί πολλές φορές, και επίσης επιτρέπει να δημιουργηθούν νέα σύνθετα στοιχεία και κάποια πρόσθετη συμπεριφορά να δηλωθεί σε μία θέση και έπειτα να συμπεριληφθεί οπουδήποτε απαιτείται .

5.6.2 Αντιγραφή

Μια δεύτερη μέθοδος παραπομπής χρησιμοποιεί το *στοιχείο copy*. Αυτό επιτρέπει την παραπομπή των στοιχείων που περιλαμβάνονται μέσα σε κάποιο άλλο στοιχείο. Το στοιχείο *copy* αντικαθίσταται αποτελεσματικά με τα στοιχεία που περιλαμβάνονται μέσα στο στοιχείο που αναφέρεται για παράδειγμα, η συλλογή "MyContent" που αναφέρεται παρακάτω περιέχει δύο *στοιχεία string*. Η δήλωση Scene περιλαμβάνει ένα στοιχείο Copy με σκοπό "να τραβήξει" αυτά τα *στοιχεία string* έτσι ώστε περιγραφούν μέσα στο στοιχείο scene . Τα στοιχεία TextBox περιέχουν στοιχεία *String* που χρησιμοποιούν τη λειτουργία *href* για να προσδιορίσουν το μεμονωμένο κομμάτι του κειμένου ώστε να το χρησιμοποιεί ως περιεχόμενό του.

```
<Collection name="MyContent">
  <String name="Item1" value="Mozart"/>
  <String name="Item2" value="Beethoven"/>
</Collection>

<Scene name="Composers">
  <!-- Include referenced content within collection -->
  <Copy href="#../MyContent"/>

  <TextBox name="Box1">
    <!-- Refer to included content item -->
    <String name="content" href="#../Item1"/>
    <Size name="size" value="100 40"/>
  </TextBox>

  <TextBox name="Box2">
    <!-- Refer to included content item -->
    <String name="content" href="#../Item2"/>
    <Size name="size" value="100 40"/>
  </TextBox>
</Scene>
```

5.7 Αντιμετώπιση των διαφορών μεταξύ των πλατφόρμων.

Η PCF παρέχει πολλά εργαλεία λειτουργιών για να επιτρέψει την περιγραφή διαδραστικών τηλεοπτικών υπηρεσιών σε μία πλατφόρμα ανεξάρτητου τρόπου. Η PCF αφαιρεί την απαίτηση για έναν συντάκτη υπηρεσιών να έχει μια λεπτομερή γνώση γλωσσών προγραμματισμού που ενσωματώνονται στα διάφορα διαδραστικά λογισμικά που υπάρχουν. Αυτό δεν σημαίνει ότι οι έμφυτες διαφορές μεταξύ των πλατφορμών απλά εξαφανίζονται: εάν μια συγκεκριμένη πλατφόρμα δεν έχει, για παράδειγμα, ενσωματωμένη την λειτουργία `return path`, δεν υπάρχει κανένας τρόπος η PCF να μπορεί να επιτρέψει την ασφαλή αμφίδρομη επικοινωνία. Για να δώσει στο συντάκτη υπηρεσιών έναν τρόπο να αντιμετωπίσει αυτό, η PCF ενσωματώνει δύο έννοιες, *βαθμοί*, *ελευθερίας* και *προφίλ*.

5.7.1 Βαθμοί Ελευθερίας

Παρά την διαφορετική φύση των διάφορων αναπτυσσόμενων διαδραστικών τεχνολογιών είναι συχνά δυνατό διάφορες πλατφόρμες, να αποδώσουν την ίδια εμπειρία στον θεατή. Αυτό συμβαίνει κυρίως εξαιτίας του γεγονότος ότι κάτω από τα διαφορετικά διακριτικά λογισμικά η αρχιτεκτονική του υλικού είναι παρόμοια.

Γενικά το πρόβλημα δεν είναι η υποστασιοποίηση ενός συγκεκριμένου στοιχείου, όπως ένα `TextBox`, σε διάφορες πλατφόρμες. Μάλλον, η πρόκληση για την PCF είναι να δημιουργήσει ορισμούς στοιχείων που κλείνουν ισοζύγιο μεταξύ τους.

- σύλληψη της συντακτικής πρόθεσης με ικανοποιητική ακρίβεια ώστε να δοθεί μια ικανοποιητική εμπειρία στο θεατή
- εξασφάλιση της φορητότητας κάθε στοιχείου σε ένα ευρύ φάσμα στόχων
- ελαχιστοποίηση της ειδίκευσης των στοιχείων ως ένας τρόπος μεγιστοποίησης της εκμετάλλευσης κάθε πλατφόρμας, για την αποφυγή πολλαπλασιασμού παρόμοιων στοιχείων

Η PCF αποδέχεται αυτό επιτρέποντας ένα *βαθμό ελευθερίας* στον προσδιορισμό κάθε στοιχείου PCF. Οποδήποτε είναι δυνατόν ο βαθμός ελευθερίας ελαχιστοποιείται, ιδανικά στο μηδέν. Εντούτοις, όπου ένας βαθμός ελευθερίας ορίζεται μπορεί να φανεί εάν υπάρχει μια ιδανική και ελάχιστη αποδεκτή εφαρμογή ή εάν οποιαδήποτε εφαρμογή μέσα στο βαθμό ελευθερίας είναι τόσο καλή όσο κάποια άλλη.

Για παράδειγμα, μπορούμε να εξετάσουμε ένα στοιχείο `TextBox`. Αυτό μπορεί να οριστεί έτσι ώστε μια γραμμή του περιεχόμενου κειμένου να αποδίδεται πάντα μέσα στην ορατή περιοχή του στοιχείου. Ωστόσο μπορεί να οριστεί με ένα βαθμό ελευθερίας έτσι ώστε η ακριβής τοποθέτηση κάθε χαρακτήρα δεν είναι υποχρεωτική εφ' όσον η αποδιδόμενη γραμμή κειμένου είναι απολύτως ορατή. Σε αυτήν την περίπτωση καμία ιδιαίτερη εφαρμογή δεν μπορεί να θεωρηθεί περισσότερο "ιδανική"

από οποιαδήποτε άλλη. Αντ' αυτού, αυτός ο βαθμός ελευθερίας είναι απαραίτητος επειδή δεν υπάρχει κανένα πρότυπο για την απόδοση κειμένου σε διαδραστικά λογισμικά .

5.7.2 Προφίλ

Για να εξασφαλιστεί το γεγονός ότι η PCF είναι σχετική με όσο το δυνατόν περισσότερες πλατφόρμες σχεδιάζεται σκόπιμα να είναι απεριόριστη από τις ικανότητες της λιγότερης ισχυρής συσκευής στόχου που εξετάζεται.

Αυτό το χαρακτηριστικό της PCF είναι απαραίτητο εάν οι αυξανόμενες ικανότητες και η ποικιλομορφία των συσκευών στόχων, οι οποίες μπορούν να αναμένονται ως πλατφόρμες εξελιχθούν κατά τη διάρκεια του χρόνου και προσαρμοστούν. Εντούτοις, αυτό αναπόφευκτα σημαίνει ότι δεν θα μπορούν όλες οι λειτουργίες που εκφράζονται μέσω της PCF να κωδικοποιηθούν ώστε να δουλεύουν σε όλες τις πλατφόρμες

Η έννοια του προφίλ της PCF έχει καθοριστεί, ώστε κάθε προφίλ να ενσωματώνει ένα σύνολο χαρακτηριστικών γνωρισμάτων της PCF έτσι ώστε να χαρτογραφεί καλά έναν συγκεκριμένο τομέα εφαρμογής .

6 Αρχιτεκτονική

Η αρχιτεκτονική ενός σχήματος είναι το πλαίσιο και η δομή που το σχήμα παρέχει σε έναν συντάκτη ώστε να του δοθεί η δυνατότητα να περιγράψει μία φορητή διαδραστική υπηρεσία. Τα στοιχεία που συμπεριλαμβάνονται στην αρχιτεκτονική ενός σχήματος είναι: *τύποι δεδομένων, δομές δεδομένων, προσδιορισμοί στοιχείων και μοντέλα αναφοράς*. Η αρχιτεκτονική περιγράφει τα θεμελιώδη στοιχεία της PCF από τα οποία δημιουργούνται άλλες περιοχές της PCF.

6.1 Εισαγωγή

Ο στόχος του καθορισμού της αρχιτεκτονικής της PCF, ανεξάρτητα από όλες τις άλλες πτυχές της PCF είναι να εξασφαλιστεί η ύπαρξη μιας συνεπούς και διαλειτουργικής περιγραφής πλαισίου για όλα τα μέρη μιας υπηρεσίας PCF.

Η αρχιτεκτονική της PCF:

- περιγράφει πλήρεις διαδραστικές υπηρεσίες
- επιτρέπει σε πτυχές μιας περιγραφής υπηρεσίας να δημιουργηθούν και να τροποποιηθούν ανεξάρτητα
- έχει ένα εκτεταμένο συντακτικό για την υποστήριξη της δημιουργίας καινούριων χαρακτηριστικών.
- εκφράζει χαρακτηριστικά γνωρίσματα σε διαφορετικά επίπεδα κλιμάκωσης.
- είναι μια απεριόριστη περιγραφή υπηρεσίας που εκμεταλλεύεται τα χαρακτηριστικά γνωρίσματα όλων των συστημάτων
- ενθυλακώνει περιγραφές πρόσθετων ιδιοτήτων
- υποστηρίζει επαρκή εγκυρότητα και επαλήθευση μιας υπηρεσίας.

Κάθε ένα από τα παραπάνω επηρεάζει όλες τις πτυχές μιας περιγραφής υπηρεσίας της PCF και έτσι η υποκείμενη αρχιτεκτονική του σχήματος παρέχει συγκεντρωμένες και συνεπείς λύσεις. Οι κοινές αρχιτεκτονικές δομές απλοποιούν τη διαδικασία δημιουργίας και παρέχουν ένα πλαίσιο για τις επεκτάσεις της PCF.

Η PCF πρέπει να είναι εύκολο να υιοθετηθεί από τη βιομηχανία και, όπου είναι δυνατόν, να βασίζεται στα υπάρχοντα δημοσιευμένα πρότυπα. Η αρχιτεκτονική της PCF είναι βασισμένη σε ένα σύνολο προτύπων τα οποία είναι σε κοινή χρήση και προέρχονται από οργανώσεις όπως η DVB και η W3C. Αυτά τα πρότυπα έχουν επιλεγεί ως κατάλληλα για την περιγραφή διαδραστικών τηλεοπτικών υπηρεσιών. Κατ' αυτό τον τρόπο, ένας συντάκτης μπορεί να χρησιμοποιήσει ένα σχήμα περιγραφής που να είναι συνεπές και στην περιγραφή υπηρεσιών PCF και σε άλλα σχήματα που περιέχονται.

6.1. Γλώσσα ισχυρού ελέγχου τύπων (Strong typing)

Η αρχιτεκτονική της PCF υποστηρίζει ένα ισχυρό μοντέλο που επιτρέπει την φορητότητα και την επεκτασιμότητα. Αυτό παρέχει σε ένα αποκωδικοποιητή PCF τη δυνατότητα μετατροπής δεδομένων στις περισσότερες πλατφόρμες, κατάλληλου τύπου, για την απόδοση, είτε σε ένα head-end σύστημα είτε σε έναν δέκτη.

Παρέχει επίσης επεκτασιμότητα, επιτρέποντας τη δημιουργία νέων χαρακτηριστικών γνωρισμάτων στα πλαίσια ενός ήδη δομημένου συνόλου τύπων δεδομένων.

Ένας συντάκτης πρέπει να δηλώσει ρητά τον τύπο δεδομένων κάθε στοιχείου. Κατά συνέπεια δίνεται σε ένα αποκωδικοποιητή η επιλογή ενός μηχανισμού που χρησιμοποιείται για μετατροπή μιας τιμής στον κατάλληλο τύπο.

Ο τύπος δεδομένων για μία τιμή δεν θα καθοριστεί σιωπηρά, για παράδειγμα η θέση όπου μια τιμή καθορίζεται σε μια περιγραφή υπηρεσίας. Η επικύρωση του τύπου μιας τιμής από τον αποκωδικοποιητή είναι απαραίτητη, δεδομένου ότι ένας συντάκτης δεν μπορεί να υποθέσει πότε μία συγκεκριμένη τιμή θα μετατραπεί στην εφικτή μορφή της σε ένα head-end σύστημα είτε σε έναν δέκτη.

Ως συνεπεία είναι απαραίτητη η προδιαγραφή ενός μεγάλου αριθμού τύπων δεδομένων.

Υπάρχουν οι ακόλουθες κατηγορίες τύπων δεδομένων .

- **primitive types** - τα «άτομα» των τύπων δεδομένων, συμπεριλαμβάνονται ακέραιοι αριθμοί και συμβολοσειρές.
- **core types** - τιμές με μια σταθερή δομή που κατασκευάζονται από ένα μίγμα τύπων. Παραδείγματος χάριν, μια θέση συντεταγμένων κατασκευάζεται από δύο τιμές ακεραίων αριθμών.
- **octet data items**- Ένα μπλοκ που αποτελείται από δεδομένα octet, με παραμέτρους οι οποίες χρησιμοποιούνται από ένα αποκωδικοποιητή για να αποκωδικοποιήσει και να ερμηνεύσει αυτά τα δεδομένα. Το μοντέλο βασίζεται στο πρότυπο MIME (Multipurpose Internet Mail Extensions). Τα δεδομένα octet χρησιμοποιούνται για δεδομένα όπως εικόνες, ήχοι και βίντεο.
- **compound types** - δομές δεδομένων που καθιερώνονται από την συγκέντρωση τιμών οποιουδήποτε τύπου σε μια περιγραφή υπηρεσίας. Οι τύποι **compound** έχουν περιορισμένες καταστάσεις όσον αφορά το τρόπο που τα στοιχεία που ανήκουν μέσα σε μια συλλογή προσδιορίζονται και αναφέρονται. Για παράδειγμα σε έναν **compound** τύπο πίνακα, τα στοιχεία προσδιορίζονται και αναφέρονται από το δείκτη που έχουν.

Ένας από τους τύπους core είναι ένας τύπος δεδομένων για την αναπαράσταση μαρκα-ρισμένου κειμένου. Αυτό το περιεχόμενο περιέχει πληροφορίες που αφορούν το ύψος και τη δομή και βρίσκονται ενσωματωμένες στη ροή του κειμένου.

6.1.2 Στατική και ενεργή περιγραφή

Μια περιγραφή υπηρεσίας που γράφεται με την βοήθεια της PCF μπορεί να χρησιμοποιηθεί για την ενδο-επιχειρησιακή ανταλλαγή διαδραστικών υπηρεσιών που περιγράφουν την εμπειρία του χρήστη συμπεριλαμβανομένης της αλληλεπίδρασης.

Επομένως, τα στοιχεία μιας περιγραφής υπηρεσίας PCF θα πρέπει να περιλαμβάνουν

- **στατική περιγραφή** - στοιχεία *scene* και το περιεχόμενό τους που περιγράφεται όπως παρουσιάζονται αρχικά σε έναν χρήστη
- **ενεργή περιγραφή** - η φορητή περιγραφή μιας δραστηριότητας που μπορεί να εμφανιστεί κατά της διάρκειας ζωής μιας υπηρεσίας.

Τα δύο είδη περιγραφής θα πρέπει να είναι συμβατά έτσι ώστε οι τροποποιήσεις που συμβαίνουν κατά τον χρόνο εκτέλεσης να μπορούν να γίνουν σε ένα καθορισμένο υποσύνολο χαρακτηριστικών γνωρισμάτων της στατικής περιγραφής. Για να επιτύχει αυτό, η γλώσσα δράσης της PCF θα πρέπει να είναι *strongly typed* έτσι ώστε μια τιμή κάποιου συγκεκριμένου τύπου δεδομένων σε μία στατική περιγραφή να μπορεί να ελεγχθεί κατά την διάρκεια του χρόνου εκτέλεσης από την ενεργό περιγραφή.

6.1.3 Παρουσίαση Μιας Υπηρεσίας

Η υψηλού επιπέδου ανατομία μιας περιγραφής υπηρεσίας PCF είναι σύμφωνη με τις απαραίτητες πληροφορίες που παρουσιάζουν μία διαδραστική υπηρεσία. Η PCF επιτρέπει σε έναν συντάκτη να εκφράζει την προοριζόμενη εμπειρία του χρήστη από την άποψη μιας διαδραστικής υπηρεσίας, που αποτελείται από υψηλού επιπέδου δομικά στοιχεία και αντικείμενα *scenes* τα οποία παρέχουν μια μονάδα παρουσίασης και διαδραστικότητας σε έναν χρήστη.

Η αρχιτεκτονική ορίζεται ως ένα **σύνολο πληροφοριών PCF**, το οποίο είναι ανεξάρτητο από την μορφή της παρουσίασης.

Οι περιγραφές υπηρεσιών PCF αποτελούνται από *στοιχεία πληροφοριών*, πολλά από τα οποία παρουσιάζονται ως δομικά στοιχεία της PCF και περιγράφονται αναλυτικά στο κεφάλαιο 7. Το στοιχείο ρίζα (*root item*) μιας περιγραφής υπηρεσίας PCF είναι ένα *αντικείμενο υπηρεσίας (service item)*. Κάθε αντικείμενο *service* περιέχει ένα ή περισσότερα αντικείμενα *scene*, και μπορεί να καθορίσει ποιο από αυτά τα στοιχεία *scene* θα παρουσιαστεί αρχικά σε έναν χρήστη.

Τα αντικείμενα *Service* και τα αντικείμενα *scene* προέρχονται από τον τύπο δεδομένων *map* της PCF. Κάθε ένας από αυτά τα στοιχεία μπορεί να περιέχει άλλα PCF αντικείμενα τα οποία είναι ομαδοποιημένα, με τον περιορισμό ότι κάθε στοιχείο

θα έχει ένα μοναδικό όνομα. Τα αντικείμενα *Collection* προέρχονται επίσης από τον τύπο δεδομένων *map* της PCF και μπορεί να περιέχουν ομαδοποιημένα και καθορισμένα από τον συντάκτη άλλα αντικείμενα PCF. Επομένως, μία περιγραφή υπηρεσίας PCF θα είναι μία ιεραρχική δομή των αντικειμένων PCF.

6.1.4 Μοντέλο παραπομπής

Το μοντέλο παραπομπής μέσα στην PCF επιτρέπει να γίνονται συνδέσεις μεταξύ οποιουδήποτε:

- Αναγνωρίσιμου στοιχείου μέσα σε μια περιγραφή υπηρεσίας PCF
- Στοιχείου PCF που καθορίζεται σε οποιοδήποτε τοπικό αρχείο ή ισοδύναμο data container;
- Στοιχείου PCF που καθορίζεται σε μια θέση που δίνεται από ένα URI (Uniform Resource Identifier).
- Πόρου που συνδέεται με την περιγραφή υπηρεσίας.

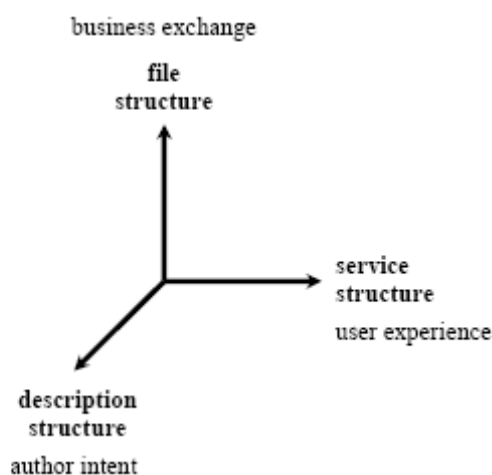
6.1.5 Διαμερισμός και επαναχρησιμοποίηση δεδομένων

Πολλοί εξωτερικοί παράγοντες επηρεάζουν στον τρόπο με τον οποίο τα στοιχεία μιας περιγραφής υπηρεσίας PCF χωρίζονται σε assets

Αυτοί περιλαμβάνουν:

- τον τρόπο με τον οποίο μια υπηρεσία μπορεί να παραδοθεί, π.χ. με δυναμικές αναπροσαρμογές.
- τους διαφορετικούς ρόλους που περιγράφουν μια υπηρεσία, π.χ. ένας γραφικός σχεδιαστής ή ένας δημοσιογράφος .
- τα συστήματα που επεκτείνονται για να παραγάγουν μια υπηρεσία, π.χ. συνεργαζόμενα συστήματα διαχείρισης.

Για να δοθεί υπεύθυνα στα άτομα που ασχολούνται με το γράψιμο και την ανάπτυξη των συστημάτων PCF η ελευθερία να περιγράψουν μια υπηρεσία με το πιο κατάλληλο τρόπο στην επιχείρησή τους, η αρχιτεκτονική του σχήματος σχεδιάζεται συγκεκριμένα για να υποστηρίξει τον ελαστικό διαχωρισμό περιγραφών. Αυτό επιτρέπει την επαναχρησιμο-ποίηση των στοιχείων με παραπομπές και την αυθαίρετη διανομή των στοιχείων σε αρχεία. Η ελάχιστη μονάδα που καθορίζει την κλιμάκωση αυτού του διαχωρισμού θα είναι ένα *στοιχείο πληροφοριών PCF*.



Εικόνα 23 . Άξονες μιας περιγραφής PCF

Η εικόνα 23 παρουσιάζει τρεις βαθμούς ελευθερίας για την διατύπωση μιας περιγραφής υπηρεσίας που παρέχει η αρχιτεκτονική PCF.

Κάθε βαθμός ελευθερίας επιτρέπει σε έναν συντάκτη να εκφράσει μια υπηρεσία με διαφορετικό τρόπο και η πιο αποδοτική είναι πιθανόν ο συνδυασμός και των τριών.

Οι βαθμοί ελευθερίας χαρακτηρίζονται ως εξής :

- **δομή υπηρεσίας**- η υπηρεσία περιγράφεται εξ ολοκλήρου με την ιεραρχία των components που καθορίζουν την παροχή της υπηρεσίας. Αυτό επιτρέπει σε έναν συντάκτη να ελέγχει ρητά κάθε πτυχή και παράμετρο του ορισμού κάθε μέρους μιας υπηρεσίας όπως δίνεται από τον χρήστη. Μια υπηρεσία εκφρασμένη με την χρήση μόνο της service structure είναι:
 - μία πλήρης περιγραφή της εμπειρίας του χρήστη από μια υπηρεσία
 - παρουσιάζεται σε μόνο ένα αρχείο
 - δεν χρησιμοποιεί PCF στοιχεία παραπομπών

- **δομή περιγραφής**- η υπηρεσία εκφράζεται με έναν modular τρόπο, επιτρέποντας στα στοιχεία να ορισθούν μία μόνο φορά και να επαναχρησιμοποιηθούν από παραπομπές. Αυτό επιτρέπει σε έναν συντάκτη να δείξει το προοριζόμενο πλαίσιο υπηρεσιών, όπως να δηλώσει ότι η ίδια παρουσίαση σχεδιοτύπου(template) πρέπει να οριστεί μία φορά σε μια θέση και να επαναληφθεί σε διάφορα scenes. Μια υπηρεσία που εκφράζεται χρησιμοποιώντας κυρίως την δομή περιγραφής είναι:

- καθορισμένη δηλώνοντας όλα τα μη-δομικά στοιχεία μία φορά σε ένα υψηλό επίπεδο, μέσα στην ιεραρχία των στοιχείων που **κάνουν** την υπηρεσία να είναι ορισμένη ξεχωριστά και κυρίως με ενδο-αρχειακή αναφορά.
 - παρουσιάζεται σε μόνο ένα αρχείο.
 - χρησιμοποιεί *τα στοιχεία παραπομπής PCF* για τη διευκρίνιση των intra-file συνδέσεων .
- **δομή αρχείου**– Η περιγραφή μιας υπηρεσίας μπορεί να μοιραστεί σε πολλά διαφορετικά αρχεία. Τα αρχεία είναι χρήσιμες συλλογές PCF information αντικειμένων που «συμμορφώνουν» τη μεταφορά των PCF δεδομένων για ενδοεπι-χειρησιακή ανταλλαγή ή ανάπτυξη κωδικοποίησης. Αυτό μπορεί να χρησιμοποιηθεί για την ελαχιστοποίηση του αντίκτυπου των αναπροσαρμογών μιας υπηρεσίας και επιτρέπει σε έναν συντάκτη να διανείμει τα δεδομένα στις πιο κατάλληλες συλλογές για τη διαχείριση των στοιχείων όσον αφορά ένα συγκεκριμένο σύστημα.

Μια υπηρεσία που εκφράζεται εξ ολοκλήρου σε διαφορετικά αρχεία :

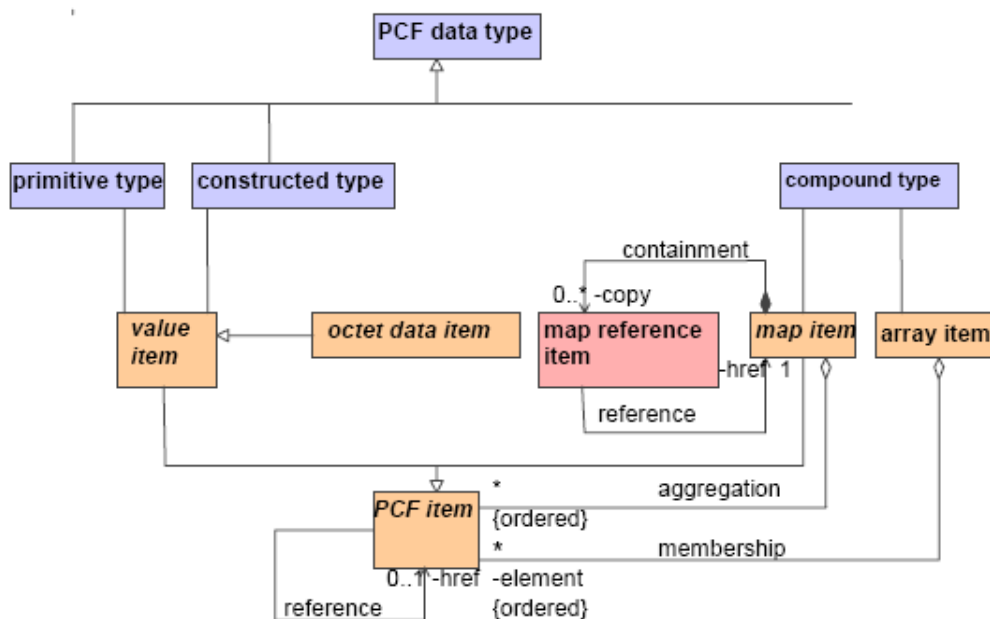
- περιγράφεται με ένα *αντικείμενο information* με αναφορές μεταξύ αρχείων που χρησιμοποιούνται για να καθορίσουν την ιεραρχία των component της υπηρεσίας
- παρουσιάζεται σε πολλά αρχεία
- χρησιμοποιεί *τα αντικείμενα reference* για τον καθορισμό των συνδέσεων μεταξύ αρχείων .

6.2 Τύποι Δεδομένων

6.2.1 Περιγραφή Τύπων Δεδομένων

6.2.1.1 Μοντέλο Τύπων Δεδομένων

Το μοντέλο τύπων δεδομένων PCF είναι ένα μετα-πρότυπο που περιγράφει τα σύνολα τιμών για διάφορες περιπτώσεις των στοιχείων της PCF. Το υψηλό επίπεδο ταξινόμησης των τύπων δεδομένων PCF και τα στοιχεία PCF που είναι μέλη αυτών των τύπων δεδομένων παρουσιάζονται στην εικόνα 24.



Εικόνα 24. Μοντέλο των τύπων δεδομένων.

Ένα στοιχείο PCF θα αντιπροσωπεύει μια τιμή οποιουδήποτε τύπου δεδομένων PCF. Ένα στοιχείο PCF μπορεί να έχει τις ακόλουθες τρεις ιδιότητες:

- **Name** - ένα προαιρετικό όνομα για την τιμή, το οποίο μπορεί να χρησιμοποιηθεί για να την προσδιορίσει μέσα σε μια περιγραφή υπηρεσίας PCF. Το όνομα θα ανήκει στον τύπο δεδομένων "NCName".
- **href** - μια διαδρομή που καθορίζει ότι η τιμή ενός στοιχείου PCF ορίζεται από την ανάλυση μιας παραπομπής. Όταν η ιδιότητα href δεν είναι κενή η τιμή του στοιχείου PCF θα καθοριστεί επιλύοντας μια παραπομπή και αντικαθιστώντας το στοιχείο, διαφορετικά η τιμή θα καθοριστεί ως τμήμα του στοιχείου PCF.

- **context** - η τιμή είναι μια απαρίθμηση η οποία μπορεί να τεθεί είτε ως «αρχική» είτε ως «παραγόμενη», καθορίζοντας το πλαίσιο εφαρμογής με το οποίο μια μη κενή αναφορά href συνδέεται με το αντικείμενο που θα αναλυθεί. Η προκαθο-ρισμένη τιμή είναι η «αρχική».

Ένα στοιχείο PCF είναι είτε μία τιμή που είναι μέλος κάποιου συνόλου που ορίζεται από έναν τύπο δεδομένων PCF ή μια παραπομπή σε μία τιμή. Μια περιγραφή υπηρεσίας PCF θα αποτελείται από τέσσερις διαφορετικές ειδικεύσεις ενός στοιχείου PCF, όπως παρου-σιάζονται παρακάτω

- **value item** - μια τιμή οποιουδήποτε primitive τύπου δεδομένων, ή μια τιμή οποιουδήποτε core τύπου δεδομένων της PCF.
- **octet data item** - ένα στοιχείο που καθορίζεται από έναν μπλοκ octet δεδομένων που διευκρινίζουν είτε το περιεχόμενο του είτε τη θέση του, τον τύπο και την κωδικοποίηση του.
- **map item** - ένα στοιχείο που περιέχει ένα σύνολο άλλων στοιχείων PCF, όπου κάθε στοιχείο μέσα στο *στοιχείο Map* θα προσδιορίζεται από το μοναδικό **όνομά** του μέσα στο σύνολο.
- **array item** - ένα στοιχείο που περιέχει ένα διατεταγμένο σύνολο άλλων στοιχείων PCF, όπου κάθε στοιχείο μέσα στην ακολουθία θα προσδιοριστεί από το **δείκτη** του μέσα στο σύνολο.

6.2.1.2 Διάστημα Περιγραφής

Κάθε τύπος δεδομένων μέσα στην PCF περιγράφεται από τα στοιχεία *value space*, *lexical space* και *set of constraining facets*, με έναν παρόμοιο τρόπο που περιγράφονται οι τύποι δεδομένων σε ένα σχήμα XML. Αυτά ορίζονται ως εξής :

- **value space** - το σύνολο των πιθανών τιμών για ένα δοσμένο τύπο δεδομένων. Κάθε τιμή υποδηλώνεται από ένα ή περισσότερα literals μέσα στο *lexical space* Ένα εργαλείο PCF, όπως ένα εργαλείο δημιουργίας ή μετατροπής, θα είναι ικανό να αναπαραστήσει όλες τις τιμές οποιωνδήποτε τύπων δεδομένων PCF .
- **lexical space** - το σύνολο των έγκυρων literals για ένα τύπο δεδομένων Όλα τα literals μέσα στο *lexical space* θα έχουν την αντίστοιχη τιμή στο **value space**. Οι τύποι δεδομένων έχουν επίσης ένα *canonical lexical representation*

που είναι ένα υποσύνολο του **lexical space**. Κάθε τιμή στο lexical space θα έχει ακριβώς μια αντίστοιχη αναπαράσταση στο canonical lexical space.

- **set of constraining facets** -ένα facet είναι μία πτυχή καθορισμού του πεδίου value space. Γενικά, κάθε ένα facet χαρακτηρίζει ένα value space κατά μήκος ενός ανεξάρτητου άξονα ή μιας διάστασης. Τα facets περιλαμβάνουν την ελάχιστη και την μέγιστη τιμή ενός υπολογίσιμου στοιχείου. Ένα συμβατό εργαλείο θα αναπα-ραστήσει μόνο τις τιμές που ικανοποιούν έναν τύπο δεδομένων μέσα στα όρια των δηλωμένων constraining facets.

Κάθε κανονικός τύπος δεδομένων περιγράφεται σύμφωνα με αυτούς τους κανόνες. Στο παρόν έγγραφο, αυτή η περιγραφή αναπαρίσταται μέσα από μια συνοπτική μορφή, όπως διευκρινίζεται και περιγράφεται στον πίνακα 3.

Name	Όνομα
Value space	Επιτρεπτές τιμές
Lexical space	Έγκυρα literals
Canonical lexical rep.	Κανονικά literals
Constraining facets	Facets

Πίνακας 3 . Πίνακας περιγραφής ενός τύπου δεδομένων

6.2.1.3 Το αντικείμενο τιμή

Κάθε τύπος δεδομένων που είναι primitive ή core έχει ένα αντίστοιχο στοιχείο value που επιτρέπει σε έναν συντάκτη να αναπαραστήσει κάποια τιμή ενός τύπου δεδομένων μέσα σε μια δομή PCF. Για κάθε τύπο δεδομένων, κάθε στοιχείο value θα αποτελείται από τις ακόλουθες επιπλέον ιδιότητες

- **type**- το όνομα του τύπου δεδομένων PCF που αναπαριστάται από το στοιχείο value.
- **value** - η τιμή που αναπαρίσταται σύμφωνα με τον τύπο δεδομένων .
- **exact** - μια Boolean τιμή, παίρνει την τιμή true για να δείξει ότι ένας χρήστης πρέπει να παρουσιάσει μια ακριβή αναπαράσταση της τιμής και παίρνει τη τιμή false εάν μια πλατφόρμα επιτρέπει να παρουσιάσει μια προσέγγιση της τιμής. Η προκαθορισμένη τιμή είναι false.

- **nil** - δείχνει ότι ένα στοιχείο PCF δεν έχει καμία τιμή. Όταν μια τιμή παρέχεται και το στοιχείο nil είναι true η τιμή θα πρέπει να αγνοηθεί. Η προκαθορισμένη τιμή για αυτήν την παράμετρο είναι false.

Παράδειγμα 1:

Μια τιμή '10' σύμφωνα με τον τύπο δεδομένων των ακέραιων αριθμών αναπαρίσταται ως ένα στοιχείο value που χρησιμοποιεί το στοιχείο του ακεραίου αριθμού ως εξής:

```
<Integer name="number_ten" value="10"/>
```

Παράδειγμα 2 :

Οποιοδήποτε στοιχείο που περιλαμβάνει την ακόλουθη προδιαγραφή χρώματος όταν κωδικοποιείται, θα πρέπει να αποδίδεται με μια 24-bit αναπαράσταση. Οποιαδήποτε πλατφόρμα που δεν μπορεί να επιτύχει μια ακριβή παρουσίαση θα πρέπει να το αναφέρει ως λάθος

```
<Color name="fillColor" value="#F9EDC5" exact="true"/>
```

6.2.2 Πρωταρχικοί τύποι δεδομένων

Οι primitive τύποι δεδομένων είναι οι δομικές μονάδες για όλους τους άλλους τύπους του σχήματος. Οι τιμές είναι ατομικές και δεν εκφράζονται σε αντιστοιχία με άλλες δομές μέσα στο σχήμα.

6.2.2.1 Boolean

Ο τύπος δεδομένων *Boolean* θα χρησιμοποιηθεί για τις τιμές που μπορούν να είναι σε μία από τις δύο καταστάσεις true ή false. Ο τύπος δεδομένων Boolean ορίζεται σύμφωνα με τον πίνακα 4.

Στο πεδίο lexical οι τιμές "1" και "true" αντιπροσωπεύουν αληθινή τιμή και οι τιμές "0" και "false" αντιπροσωπεύουν ψευδή τιμή .

Name Value space	Boolean
Value space	{ true, false }
Lexical space	< true false 0 1 >
Canonical lexical rep.	< true false >
Constraining facets	Δεν επιτρέπονται κενά διαστήματα

Πίνακας 4. Ο τύπος δεδομένων Boolean

6.2.2.2 Ακέραιοι Αριθμοί

Ο τύπος δεδομένων integer ορίζεται στον πίνακα 5.

Name	integer
Value space	{-2147483648, ..., -1, 0, 1, ..., 2147483647}
Lexical space	< (\+ -)? [0-9]+ >
Canonical lexical rep.	< 0 (-? [1-9] [0-9]*) >
Constraining facets	Δεν επιτρέπονται κενά διαστήματα Η ελάχιστη τιμή που περιλαμβάνεται είναι η -2147483648 Η μέγιστη τιμή που περιλαμβάνεται είναι η 2147483647.

Πίνακας 5. Ο τύπος δεδομένων integer

6.2.2.3 Απαρίθμηση

Ο τύπος δεδομένων *enumeration* παρέχει έναν αφηρημένο μηχανισμό για τον ορισμό των ιδιοτήτων χρησιμοποιώντας κατάλληλα σύνολα ετικετών. Με αυτό τον τρόπο, είναι δυνατό να οριστεί ένα σύνολο τύπων δεδομένων που μπορούν να επεκταθούν σε μελλοντικές εκδόσεις της PCF.

Μία **bijjective** χαρτογράφηση που περιέχει x απαριθμημένα στοιχεία θα καθορίσει μια απαρίθμηση η οποία αποκαλείται e . Κάθε απαριθμημένο στοιχείο αποτελείται από έναν ακέραιο αριθμό εύρους $0 \dots (X - 1)$ και από ένα literal από την ακολουθία l_0 έως $l_{(x-1)}$.

Λαμβάνοντας υπόψη αυτήν την χαρτογράφηση, ο τύπος δεδομένων *enumeration* ορίζεται σύμφωνα με τον πίνακα 6.

Name	enumeration
Value space	$\{ 0, \dots, (x - 1) \}$
Lexical space	$\{ l_0, \dots, l_{(x-1)} \}$.
Canonical lexical rep.	$\{ l_0, \dots, l_{(x-1)} \}$.
Constraining facets	Δεν επιτρέπονται τα κενά διαστήματα

Πίνακας 6. Ο τύπος δεδομένων enumeration

ΠΑΡΑΔΕΙΓΜΑ: Ένα text component έχει μια ιδιότητα για την αναπαράσταση οριζόντιας ευθυγράμμισης η οποία ονομάζεται "h-align". Η παρακάτω ένα προς ένα χαρτογράφηση ορίζεται για τον τύπο δεδομένων *enumeration* $\{ 0 _ "left", 1 _ "center", 2 _ "right", 3 _ "justify" \}$. Μόνο τα literals που αναπαριστούνται σε αυτό το σύνολο μπορούν να χρησιμοποιηθούν για την διεκκρίνηση των τιμών αυτής της ιδιότητας.

6.2.2.4 Συμβολοσειρά

Ο τύπος δεδομένων *string* θα αναπαραστήσει μια ακολουθία χαρακτήρων Unicode. Ο τύπος δεδομένων *string* ορίζεται στον πίνακα 7.

Name A sequence of any Unicode characters. Length defined by number of characters.	string
Value space	Μία ακολουθία Unicode χαρακτήρων
Lexical space	< (. \n \r)* >
Canonical lexical rep	< (. \n \r)* >
Constraining facets	Μήκος που ορίζεται από αριθμο'θα η γράμματα

Πίνακας 7. Ο τύπος δεδομένων string

Παράδειγμα

Χρησιμοποιώντας το σχήμα PCF XML ένα στοιχείο *string value* μπορεί να παρουσιαστεί με δύο τρόπους. Τα ακόλουθα δύο στοιχεία *string value* είναι ισοδύναμα.

```
<String name="eg" value="Example string value."/>
```

```
<String name="eg">  
  <PlainTextData>Example string value.</PlainTextData>  
</String>
```

6.2.3 Οι τύποι δεδομένων Core

6.2.3.1 Χρώμα

Ο τύπος δεδομένων *colour* παρέχει τα μέσα για έναν συντάκτη να περιγράψει ένα χρώμα με μία πλατφόρμα-ανεξάρτητου τρόπου. Η λεξικολογική αναπαράσταση βασίζεται στις αποδεκτές τιμές των χρωμάτων από την HTML, εκτεταμένες ώστε να υποστηρίξουν τιμές για την διαφάνεια των χρωμάτων όπως είναι διαθέσιμες μέσα στην MHP Java class "org.dvb.ui.DVBColor". Ο τύπος δεδομένων *colour* ορίζεται στον πίνακα 8 με μια πρόσθετη τιμή που αναπαριστά το επίπεδο διαφάνειας για το χρώμα.

Η ελάχιστη τιμή για το κόκκινο, το πράσινο ή το μπλε component ενός χρώματος είναι το 0, η οποία αναπαριστά την περίπτωση όπου δεν εμφανίζεται κανένα colour component. Αυτή η τιμή αυξάνεται γραμμικά και φθάνει στην μέγιστη τιμή 255 που αναπαριστά την μέγιστη ένταση του colour component. Για τη διαφάνεια ενός χρώματος, το 0 αναπαριστά την πλήρη διαφάνεια και αυξάνοντας γραμμικά φθάνει στην μέγιστη τιμή 255 που δείχνει ότι το χρώμα είναι πλήρως αδιαφανές.

Στη λεξικολογική μορφή, όλες οι τιμές για τα colour components και η διαφάνεια θα πρέπει να αναπαριστούνται ως διψήφιοι δεκαεξαδικοί αριθμοί.

Στους δεκαεξαδικούς αριθμούς το '0' αναπαρίσταται ως '00'.

Name	color
Value space	Μία ακολουθία 4ακεραίων τιμών μεταξύ 0..255
Lexical space	< # [0-9A-Fa-f]{6} ([0-9A-Fa-f]{2})? >.
Canonical lexical rep.	< # [0-9A-F]{8} >
Constraining facets	Δεν επιτρέπονται κενά διαστήματα Η ελάχιστη αποδεκτή τιμή είναι η - "#00000000". Η μέγιστη αποδεκτή τιμή είναι η "#FFFFFFFF".

Πίνακας 8. Ο τύπος δεδομένων colour

Παράδειγμα

Το πλήρες αδιαφανές λευκό είναι "# FFFFFFFF" και το αδιαφανές μαύρο είναι "# 000000FF".

6.2.3.2 Νόμισμα

Ο τύπος δεδομένων *currency* αναπαριστά αριθμητικές τιμές, όπως ποσά χρημάτων, τα οποία εκφράζονται κανονικά, χρησιμοποιώντας το δεκαδικό σύστημα και δύο δεκαδικά ψηφία. Η PCF δεν έχει τύπους δεδομένων float. Ο τύπος δεδομένων *currency* παρέχει τον τρόπο για την αναπαράσταση ψηφίων που χρησιμοποιούνται συχνά. Ο τύπος δεδομένων *currency* ορίζεται στον πίνακα 9.

Name	currency
Value space	{ -21474836.48, ..., -0.01, 0.00, 0.01, ..., 21474836.47 }
Lexical space	< (\+ -)? [0-9]+ \. [0-9] [0-9] >
Canonical lexical rep.	< (0 -? [1-9] [0-9]*) \. [0-9] [0-9] >
Constraining facets	<p>Δεν επιτρέπονται κενά διαστήματα Η ελάχιστη τιμή που περιλαμβάνεται είναι η -21474836.48.</p> <p>Η μέγιστη τιμή που περιλαμβάνεται είναι η 21474836.47.</p> <p>Η ακρίβεια είναι πάντα 2δεκαδικά ψηφία</p>

Πίνακας 9. Ο τύπος δεδομένων currency

Παράδειγμα

Το 0 αναπαριστάται ως "0.00", και το 10 ως "10.00".

6.2.3.3 Ημερομηνία

Ο τύπος δεδομένων *date* αναπαριστά μια ημερομηνία. Ο τύπος δεδομένων *date* περιγράφεται στον πίνακα 10.

Name	date
Value space	Όλες οι έγκυρες ημερομηνίες σύμφωνα με το Γρηγοριανό ημερολόγιο από 1 _η Ιανουαρίου 0000 μέχρι 31 _η Δεκεμβρίου 9999
Lexical space	< [0-9]{4} - [0-1] [0-9] - [0-3] [0-9] ((\+ -) [0-1] [0-9] : [0-5] [0-9]) Z)? >
Canonical lexical rep.	< [0-9]{4} - [0-1] [0-9] - [0-3] [0-9] ((\+ -) [0-1] [0-9] : [0-5] [0-9]) Z)? >
Constraining facets	Δεν επιτρέπονται τα κενά διαστήματα. Η ημερομηνία("0000") είναι έγκυρη . Η ελάχιστη ημερομηνία που περιλαμβάνεται είναι η "0000-01-01+12:00". Η μέγιστη ημερομηνία που περιλαμβάνεται είναι η "9999-12-31-11:59".

Πίνακας 10. Ο τύπος δεδομένων date

6.2.3.4 Ημερομηνία και ώρα

Ο τύπος δεδομένων *date and time* αναπαριστάνει μία στιγμή στον χρόνο συνδυάζοντας την αναπαράσταση μιας ημερομηνίας με το χρόνο κατά την διάρκεια της ημέρας εκείνης της ημερομηνίας εκφρασμένης σε όρους ωρών, λεπτών, δευτερολέπτων και χιλιοστών του δευτερολέπτου από την έναρξη εκείνης της ημερομηνίας.

6.2.3.5 Οικογένειες γραμματοσειρών

Ο τύπος δεδομένων *font family* αναπαριστά τις περιγραφές των στοιχείων *font family* που είναι συμβατές με τις προδιαγραφές των στοιχείων *font family* που περιγράφονται στην πρόταση 15.2.2 της CSS2 .Η τιμή ενός στοιχείου *font family* θα

καθορίσει μια λίστα αναλυτικών και γενικών ονομάτων των στοιχείων που ανήκουν στον τύπο δεδομένων font family.

Ο τύπος δεδομένων font family ορίζεται στον πίνακα 11.

Name	fontFamily
Value space	λίστα με μία ή περισσότερες οικογένειες γραμματοσειρών που προσδιορίζονται αναλυτικά ή γενικά με τα ονόματα των γραμματοσειρών
Lexical space	μία λίστα που περιέχει ένα ή περισσότερα αναλυτικά ή γενικά ονόματα των γραμματοσειρών που διαχωρίζονται με κόμμα
Canonical lexical rep.	μία λίστα που περιέχει ένα ή περισσότερα αναλυτικά ή γενικά ονόματα των γραμματοσειρών που διαχωρίζονται με κενά διαστήματα
Constraining facets	οι οικογένειες γραμματοσειρών δεν πρέπει να περιέχουν κενά διαστήματα

Πίνακας 11. Ο τύπος δεδομένων font family

Παράδειγμα:

Οι παρακάτω τιμές των στοιχείων δείχνουν μια περιγραφή του στοιχείου *font family* όπως παρουσιάζεται σύμφωνα με το σχήμα PCF XML

```
<font-family value="Tiresias, Helvetica, sans-serif"/>
```

6.2.3.6 Μέγεθος γραμματοσειράς

Ο τύπος δεδομένων **font size** αναπαριστά τις περιγραφές των στοιχείων που είναι συμβατές με τις προδιαγραφές των στοιχείων font family που περιγράφονται στην πρόταση 15.2.2 της CSS2. Η τιμή ενός στοιχείου **font size** θα καθορίσει το απόλυτο μέγεθος, το σχετικό μέγεθος, το μήκος ή το ποσοστό όπως ορίζονται στις προδιαγραφές της πρότασης 15.2.4 της CSS2. Ο τύπος δεδομένων font size ορίζεται στον πίνακα 11.

Name	fontSize
Value space	Πρόταση 15.2.4 του CSS2
Lexical space	Πρόταση 15.2.4 του CSS2
Canonical lexical rep.	Πρόταση 15.2.4 του CSS2
Constraining facets	όχι αρνητικοί ακέραιοι αριθμοί

*Πίνακας 12. Ο τύπος δεδομένων **fontSize***

Οι απόλυτες τιμές του στοιχείου **font size** που ορίζονται στις προδιαγραφές της CSS2.

- **xx-small**
- **x-small**
- **small**
- **medium**
- **large**
- **x-large**
- **xx-large**

Οι σχετικές τιμές του στοιχείου **font size** που ορίζονται στις προδιαγραφές της CSS2.

- **Larger**
- **Smaller**

6.2.3.7 Μαρκιαρισμένο κείμενο

Ο τύπος δεδομένων *marked up text* περιέχει τιμές που είναι έγγραφα XML σύμφωνα με την παρουσίαση του τύπου δεδομένων *marked up text* στην παράγραφο 6.6. Η διαμόρφωση του *marked up text* επιτρέπει σε strings που περιέχουν κείμενο να δομηθούν σε παραγράφους και πίνακες δίνοντας την επιλογή στον συντάκτη να εφαρμόσει παραμέτρους που αφορούν το στυλ. Ο τύπος δεδομένων *marked up text* ορίζεται στον πίνακα 13.

Name	markedUpText
Value space	περιπτώσεις αντικειμένων XML εγγράφων τα οποία είναι έγκυρα σύμφωνα με το σχήμα XML "x-dvb-pcf.xsd".
Lexical space	καλά διαμορφωμένο έγγραφο XML σύμφωνα με το σχήμα XML "x-dvb-pcf.xsd"
Canonical lexical rep.	Το μαρκαρισμένο κείμενο θα παρουσιαστεί σε μία Canonical XML μορφή
Constraining facets	σύμφωνα με τα πρότυπα της XML

Πίνακας 13. Ο τύπος δεδομένων marked up text

Παράδειγμα

```
<MarkedUpText name="markedUp">
  <body xmlns="http://www.dvb.org/pcf/x-dvb-pcf">
    <p>Example text with <em>emphasis</em>.</p>
  </body>
</MarkedUpText>
```

6.2.3.8 Όνομα

Ο τύπος δεδομένων *name* καθορίζει τιμές για όλα τα πιθανά ονόματα που μπορούν να χρησιμοποιηθούν μέσα στην PCF. Ο τύπος δεδομένων *name* ορίζεται στον πίνακα 14.

Name	name
Value space	< [\p{L}_] [\p{L}\p{N}_\.\-]* >
Lexical space	< [\p{L}_] [\p{L}\p{N}_\.\-]* >
Canonical lexical rep.	< [\p{L}_] [\p{L}\p{N}_\.\-]* >
Constraining facets	Δεν επιτρέπονται κενά διαστήματα .

Πίνακας 14. Ο τύπος δεδομένων name

6.2.3.9 Θέση

Ο τύπος δεδομένων *position* θα χρησιμοποιηθεί για να αναπαραστήσει τιμές που εντοπίζουν ένα οπτικό component στην οθόνη. Οι τιμές δημιουργούνται από ζεύγη ακεραίων τιμών όπως ορίζεται στον πίνακα 15. Για τα οπτικά component μία τιμή του τύπου δεδομένων *position* αναπαριστά το pixel το οποίο βρίσκεται στο αριστερό άνω άκρο τις οθόνης .

Name	<i>position</i>
Value space	{ (integer × integer) }
Lexical space	< (\+ -)? [0-9]+ \p{Zs} (\+ -)? [0-9]+ >
Canonical lexical rep.	< (0 -? [1-9] [0-9]*) \p{Zs} (0 -? [1-9] [0-9]*) >
Constraining facets	Ακολουθίες ακεραίων μήκους 2. Δεν επιτρέπονται κενά διαστήματα

Πίνακας 15 . Ο τύπος δεδομένων position

Η πρώτη τιμή ενός ζεύγους τιμών θα είναι η συντεταγμένη ενός pixel κατά μήκος του οριζόντιου άξονα, που μετριέται από αριστερά προς στα δεξιά. Η δεύτερη τιμή του ζεύγους τιμών θα είναι η συντεταγμένη pixel κατά μήκος του κάθετου άξονα, που μετριέται από πάνω προς τα κάτω. Η τιμή "0.0" θα βρίσκεται στο πάνω και αριστερό μέρος της οθόνης και αναφέρεται στο πρώτο pixel που βρίσκεται μέσα στην οθόνη.

Παράδειγμα:

Το pixel που βρίσκεται στην κάτω δεξιά γωνία της οθόνης στη προκαθορισμένη οθόνη αναφοράς είναι στην θέση "719.575".

6.2.3.10 Αναλογία

Ο τύπος δεδομένων *proportion* αναπαριστά παράγοντες αναλογίας. Αυτοί επιτρέπουν σε μια τιμή να περιγραφεί ως ποσοστό κάποιας άλλης. Ο τύπος δεδομένων *proportion* ορίζεται στον πίνακα 16.

Name	proportion
Value space	{ (integer × integer) }
Lexical space	< (\+ -)? [0-9]+ \p{Zs} (\+ -)? [0-9]+ >
Canonical lexical rep	< (0 -? [1-9] [0-9]*) \p{Zs} [0-9]+ >
Constraining facets	Ακολουθίες ακεραιών με μήκος 2 Δεν επιτρέπονται τα κενά διαστήματα. Ο δεύτερος ακέραιος δεν πρέπει να είναι ίσος με το 0.

Πίνακας 16. Ο τύπος δεδομένων proportion

Παράδειγμα: το ένα τρίτο μπορεί να διατυπωθεί ως "1 3", τα τρία όγδοα ως "3 8" και η φράση τρεις φορές μεγαλύτερο ως "3 1".

6.2.3.11 Μέγεθος

Οι τιμές του τύπου δεδομένων *size* αναπαριστούν το μέγεθος των ορθογώνιων αξόνων ενός οπτικού component. Οι τιμές αυτές αποτελούνται από ένα ζευγος ακεραιών όπως ορίζεται στον πίνακα 17.

Name	Size
Value space	{ (integer × integer) }
Lexical space	< (\+ -)? [0-9]+ \p{Zs} (\+ -)? [0-9]+ >
Canonical lexical rep.	< (0 -? [1-9] [0-9]*) \p{Zs} (0 -? [1-9] [0-9]*) >
Constraining facets	Ακολουθίες ακεραιών μήκους 2 . Δεν επιτρέπονται κενά διαστήματα μέσα στις ακέραιες τιμές .

Πίνακας 17. Ο τύπος δεδομένων size

Ο πρώτος αριθμός στο ζεύγος τιμών αναπαριστάει το μέγεθος ενός οπτικού component σε pixel όταν μετριέται κατά μήκος του οριζόντιου άξονα. Ο δεύτερος αριθμός αναπαριστάει το μέγεθος ενός οπτικού ενός component σε pixel όταν μετριέται κατά μήκος του κάθετου άξονα.

Το μηδέν και οι αρνητικές τιμές για τον οριζόντιο και για τον κάθετο άξονα μέγεθος ενός οπτικού ενός component επιτρέπονται. Οποιαδήποτε παρουσίαση που οδηγεί σε ένα οπτικό component δεν φανερώνεται στον χρήστη. Το μηδέν ή ένα αρνητικό μέγεθος δεν θα χρησιμοποιηθούν για να αναπαραστήσουν το γραμμικό μετασχηματισμό ενός οπτικού component.

Παράδειγμα: Το προκαθορισμένο μέγεθος της οθόνης αναφοράς γραφεται ως εξής "720 576".

6.2.3.12 Ώρα

Οι τιμές του τύπου δεδομένων *time* αναπαριστούν μια στιγμή του χρόνου που επανα-λαμβάνεται κάθε ημέρα.

Name	time
Value space	(Όλες οι περιπτώσεις που έχουν μηδενική καθημερινή διάρκεια)
Lexical space	< [0-2] [0-9] : [0-5] [0-9] : [0-5] [0-9] (\. [0-9] [0-9]? [0-9]?)? (((\+ -) [0-1] [0-9] : [0-5] [0-9]) Z)? >
Canonical lexical rep.	< [0-2] [0-9] : [0-5] [0-9] : [0-5] [0-9] (\. [0-9] [0-9]? [0-9]?)? (((\+ -) [0-1] [0-9] : [0-5] [0-9]) Z)? >
Constraining facets	Δεν επιτρέπονται τα κενά διαστήματα. Η ελάχιστη ώρα που περιλαμβάνεται είναι η "00:00:00+12:00". Η μέγιστη ώρα που περιλαμβάνεται είναι η "23:59:59.999-11:59".

Πίνακας 18 . Ο τύπος δεδομένων time

6.2.3.13 Χρονική Αναφορά

Οι τιμές του τύπου δεδομένων *timecode* αναπαριστούν μια χρονική αναφορά σε ένα σημείο σε μία γραμμική ροή πολυμέσων, π.χ. ο προσδιορισμός ενός *frame* σε μια ακολουθία βίντεο. Οι τιμές του τύπου δεδομένων *timecode* ορίζονται στον πίνακα 19.

Name	Timecode
Value space	Αναφορές σε πλαίσια μιας ακολουθίας βίντεο ή ταινίας που το μήκος τους δεν ξεπερνάει τις 24 ώρες.
Lexical space	< [0-2] [0-9] : [0-5] [0-9] : [0-5] [0-9] \. [0-5] [0-9] >
Canonical lexical rep.	< [0-2] [0-9] : [0-5] [0-9] : [0-5] [0-9] \. [0-5] [0-9] >
Constraining facets	Δεν επιτρέπονται τα κενά διαστήματα. Η ελάχιστη τιμή που περιλαμβάνεται είναι η "00:00:00.00". Η μέγιστη τιμή που περιλαμβάνεται είναι η "'23:59:59.59".

Πίνακας 19. Ο τύπος δεδομένων timecode

6.2.3.14 URI

Ο τύπος δεδομένων στοιχείων *URI* αναπαριστάνει τιμές που είναι έγκυρα Uniform Resource Identifiers σύμφωνα με τον τύπο δεδομένων "anyURI" που υπάρχει στην πρόταση 3.2.17 στις προδιαγραφές του XML σχήματος.

Παράδειγμα

Τα ακόλουθα είναι έγκυρα URIs (Uniform Resource Identifier) σύμφωνα με τον τύπο δεδομένων της PCF:

<http://www.etsi.org/>
<ftp://ftp.sourceforge.net/pub/sourceforge/>
<urn:x-dvb-pcf:bbc.co.uk:bbc-one>

6.2.3.15 Κλειδιά του χρήστη

Ο τύπος δεδομένων *user key* αναπαριστά τα εικονικά κλειδιά που είναι διαθέσιμα σε έναν χρήστη. Ένας συντάκτης μιας υπηρεσίας PCF μπορεί να χρησιμοποιήσει οποιοδήποτε εικονικό κλειδί και να υποθέσει ότι ένα εργαλείο PCF θα ταιριάζει με ένα κατάλληλο κλειδί ή με ένα χαρακτηριστικό που διαθέσιμο στον χρήστη.

Ωστόσο όλες οι πλατφόρμες δεν παρέχουν μια χαρτογράφηση από ένα εικονικό κλειδί σε ένα συγκεκριμένο κλειδί πλατφόρμας ή σε ένα χαρακτηριστικό, και όταν δεν υπάρχει κάποια κατάλληλη χαρτογράφηση για ένα κλειδί τότε ένα εργαλείο της PCF δεν πρέπει να παρέχει την υπηρεσία και θα πρέπει να αναφέρει ένα λάθος.

Παράδειγμα

Ένα μετατροπέας για ένα δίκτυο χαρτογραφεί το εικονικό κλειδί "VK_CANCEL" σε μια περιγραφή υπηρεσίας PCF σε ένα πραγματικό κλειδί τηλεχειρισμού το οποίο ονομάζεται "backup" σε ένα δίκτυο. Ένα άλλος μετατροπέας για ένα διαφορετικό δίκτυο χαρτογραφεί το ίδιο εικονικό κλειδί σε ένα πραγματικό κλειδί τηλεχειρισμού που ονομάζεται "cancel".

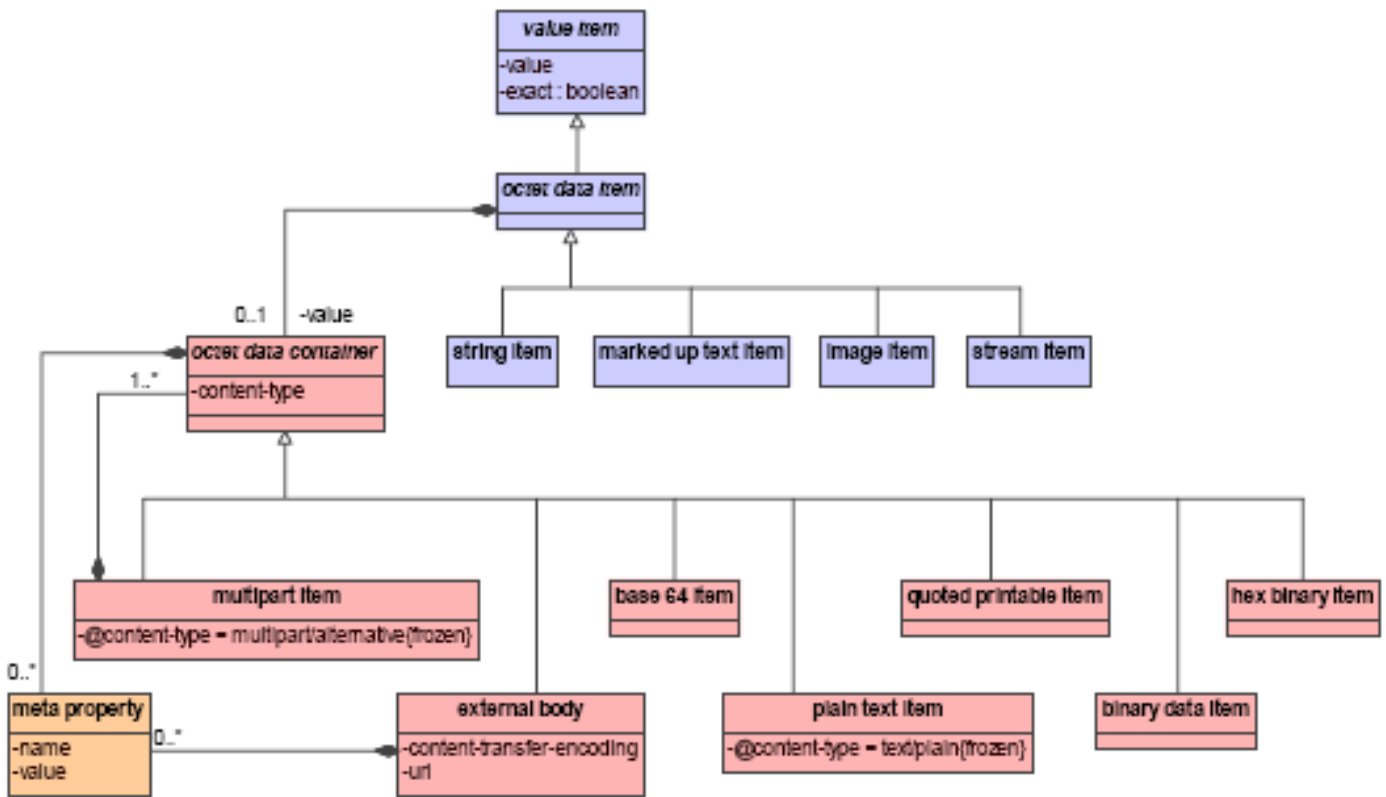
6.2.4 Αντικείμενα δεδομένων octet

6.2.4.1 Εισαγωγή στα δεδομένα octet

Πολλά στοιχεία δεδομένων για την PCF παρέχονται ως ακολουθίες ή ροές *octet στοιχείων*, για παράδειγμα ήχοι και δεδομένα εικόνας. Παρά τη δημιουργία των συγκεκριμένων τύπων δεδομένων της PCF για κάθε τύπο στοιχείων, η PCF παρέχει ένα πλαίσιο για τον χειρισμό όλων μορφών των octet δεδομένων με έναν συνεπή τρόπο. Ο τύπος των δεδομένων είναι μια ιδιότητα που συνδέεται με τα δεδομένα. Αυτός ο μηχανισμός βασίζεται στην Multipurpose Internet Mail Extensions (MIME) και δίνει τη δυνατότητα στα δεδομένα να είναι:

- ενσωματωμένα μέσα σε μια περιγραφή υπηρεσίας PCF.
- σε ένα ξεχωριστό τοπικό αρχείο.
- σε αναφορά με μία εξωτερική πηγή όπως ένα αρχείο ή μία ροή δεδομένων.

6.2.4.2 Μοντέλο των δεδομένων octet



Εικόνα 25. Μοντέλο αντικειμένων των δεδομένων octet.

Το παραπάνω σχήμα παρουσιάζει το μοντέλο των στοιχείων octet. Η PCF έχει επτά διαφορετικά είδη *octet data container* που μπορούν να αναπαραστήσουν την τιμή ενός *octet data* αντικειμένου και αυτοί περιγράφονται στην παράγραφο 6.2.4.3. Η PCF έχει τέσσερις τύπους δεδομένων που μπορούν να αναπαρασταθούν από *octet data items* και αυτοί περιγράφονται στην παράγραφο 6.2.4.4.

Το στοιχείο octet data της PCF παρέχει μια διεπαφή μεταξύ των στοιχείων value και των στοιχείων octet data. Τα στοιχεία Octet data είναι υποσύνολο των στοιχείων value όπου η ιδιότητα της τιμής ενός value στοιχείου παρέχεται από μια ακολουθία ή μια ροή στοιχείων octet data. Αυτή η διεπαφή επιτρέπει την περιγραφή των συνοδευτικών δεδομένων που σχετίζονται με το στοιχείο octet data, συμπεριλαμβανομένου του τύπου δεδομένων που παρέχει το στοιχείο octet data και του ίδιου του στοιχείου, είτε σε μια ενσωματωμένη μορφή είτε σε παραπομπή.

Παράδειγμα

```
<String name="whichOne" value="I am ignored.">
  <PlainTextData>I am preferred.</PlainTextData>
</String>
```

6.2.4.3 Αντικείμενα περίληψης δεδομένων octet (Octet data containers)

6.2.4.3.1 Φορητοί τύποι MIME

Οι φορητοί MIME τύποι της PCF είναι ένα σύνολο MIME τύπων όπου όλα τα συστήματα της PCF υποστηρίζουν. Μια περιγραφή υπηρεσίας που περιέχει στοιχεία *octet data* με συμβατό *content-type* με τους φορητούς τύπους MIME αναφέρεται επίσης και ως φορητή περιγραφή υπηρεσίας.

Τα στοιχεία *Octet data containers* έχουν μία "content-type" ιδιότητα. Η τιμή που ορίζεται σε αυτήν την ιδιότητα θα πρέπει :

- να είναι ακριβής όσον αφορά την κωδικοποίηση των στοιχείων octet.
- να είναι ένας από τους τύπους MIME που υποστηρίζονται από την PCF, όπως διευκρινίζεται στον πίνακα 21, για να εξασφαλιστεί η φορητότητα. Εάν η κωδικοποίηση των στοιχείων octet δεν είναι γνωστή, όπως στην περίπτωση μιας εξωτερικής πηγής τότε η ιδιότητα "content-type" μπορεί είτε να παραλειφθεί είτε μπορεί να οριστεί ένας γενικός τύπος MIME.
- θα ήταν αρμόζον για ένα *στοιχείο octet data* να περιέχει το *στοιχείο octet data container*. Στις περισσότερες περιπτώσεις αυτό είναι προφανές και υπάρχουν διάφοροι "όμοιοι" τύποι MIME που είναι εξίσου έγκυρες αναπαραστάσεις στοιχείων για έναν συγκεκριμένο τύπο των στοιχείων octet data. Για παράδειγμα η εικόνα/PNG και η εικόνα/jpeg για το αντικείμενο image. Εντούτοις, σε μερικές περιπτώσεις οι πιθανοί MIME τύποι δεν είναι ομότυποι και θα πρέπει να διευκρινιστεί κάποια χαρτογράφηση. Για παράδειγμα, text/x-dvb-pcf και text/plain για το αντικείμενο *marked up text item*.

MIME type	Description
text/plain	Σαφή δεδομένα κειμένου χωρίς τη μορφοποίηση εντολών ή οδηγιών.
text/x-dvb-pcf	Δεδομένα κειμένου σύμφωνα με τις σημειώσεις της PCF.
image/png	Octet δεδομένα σε μορφή εικονας PNG.
image/jpeg	Octet δεδομένα σε μορφή JPEG.
image/x-mpeg-iframe	Μία στοιχειώδης ροή βίντεο σε μορφή MPEG
audio/mpeg	Μία στοιχειώδης ροή βίντεο σε μορφή. MPEG.
video/mpeg	Μία στοιχειώδης ροή βίντεο σε μορφή. MPEG.
audio/basic	Δεδομένα octet στην βασική μορφή ήχου.
multipart/alternative	Αναλύεται στην παράγραφο 6.2.4.3.9.
application/octet-stream	Μία ροή octet δεδομένων που αποτελείται από αυθαίρετα δυαδικά στοιχεία..
video/x-MP2PS	Μία ροή octet δεδομένων που αναπαριστάει μία ροή προγράμματος MPEG-2.
video/x-MP2TS	Μία ροή octet δεδομένων που αναπαριστάει μία ροή μετάδοσης MPEG-2.
video/x-MP2TS-P	Μία ροή octet δεδομένων που αναπαριστάει μία ροή προγράμματος MPEG-2 μέσα σε μία ροή μετάδοσης MPEG-2.
video/x-MP2ES	Μία ροή octet δεδομένων που αναπαριστάει μία στοιχειώδης ροή MPEG-2 από απροσδιόριστους τύπους.

Πίνακας 20. MIME τύποι της PCF.

6.2.4.3.2 Αντικείμενα Meta property

Τα στοιχεία *Meta property* χρησιμοποιούνται από ένα συντάκτη για την παροχή πρόσθετων πληροφοριών στο σύστημα PCF για την ερμηνεία των στοιχείων *octet data*. Τα στοιχεία *Meta property* θα πρέπει να έχουν τις ακόλουθες ιδιότητες:

- **name** - το όνομα ενός στοιχείου *Meta property* που ανήκει στον τύπο δεδομένων *name* ορίζεται στην παράγραφο 6.2.3.8.
- **value** - η τιμή αυτή θα είναι μία συμβολοσειρά

Τα ονόματα όλων των αντικειμένων *meta-property* που περιέχονται μέσα σε ένα *octet data item* θα πρέπει να είναι μοναδικά

6.2.4.3.3 Ενσωματωμένο ακρυπτογράφητο κείμενο δεδομένων

Ένα αντικείμενο *embedded plain text data* μπορεί να περιέχει δεδομένα *octet* που κωδικοποιείται ως στοιχειώδες κείμενο σύμφωνα με τον MIME τύπο "text/plain". Τα *Plain text data* θα πρέπει να έχουν ένα χαρακτήρα κωδικοποίησης που θα ταιριάζουν με το PCF έγγραφο μέσα στο οποίο έχουν δηλωθεί.

ΠΑΡΑΔΕΙΓΜΑ

Τα ακόλουθα δύο πλαίσια XML που γράφονται σύμφωνα με το σχήμα PCF XML είναι ισοδύναμα.

```
<String name="headline1" value="Tsunami Early Warning"/>
```

```
<StringData name="headline1">  
  <PlainTextData>Tsunami Early Warning</PlainTextData>  
</StringData>
```

6.2.4.3.4 Ενσωματωμένα δυαδικά δεδομένα

Ένα αντικείμενο *embedded binary data* μπορεί να περιέχει δεδομένα τύπου *octet* τα οποία δεν είναι κωδικοποιημένα και έχουν μέγεθος 8-bit.

6.2.4.3.5 Ενσωματωμένα base64 δεδομένα

Ένα στοιχείο embedded base64 μπορεί να περιέχει octet δεδομένα τα οποία είναι κωδικοποιημένα σύμφωνα με την μέθοδο που ορίζεται στην παράγραφο 6.8 στις προδιαγραφές MIME.

ΠΑΡΑΔΕΙΓΜΑ

Τα δεδομένα εικόνας για ένα βέλος με κατεύθυνση προς τα πάνω μπορούν να κωδικοποιηθούν μέσα σε μια περιγραφή υπηρεσίας PCF χρησιμοποιώντας την κωδικοποίηση base64 ως εξής:

```
<ImageData name="up_arrow">
<Base64Data content-type="image/png">
iVBORw0KGgoAAAANSUhEUgAAAAoAAAACCAIAAAACUFjqAAAABmJLR
0QA/wD/AP+gvaeTAAAANUIE
QVR4nGP8//8/A27AhMxhZGRkZGTEKY1PN1wfsGHE6UazEs4loJsBj8f+///PSIK/
MQEANb4MF+Di
rHMAAAAASUVORK5CYII=
</Base64Data>
</ImageData>
```

6.2.4.3.6 5 Ενσωματωμένα δεκαεξαδικά δεδομένα

Ένα ενσωματωμένο αντικείμενο *hexadecimal binary data* μπορεί να περιέχει octet δεδομένα τα οποία είναι κωδικοποιημένα σύμφωνα με τις προδιαγραφές MIME.

Παράδειγμα

Η συμβολοσειρά "PCF" στην κωδικοποίηση χαρακτήρων UTF-8 παρουσιάζεται ως "514346" κατά την δεκαεξαδική μετατροπή των δεδομένων, όπου κάθε χαρακτήρας από την αρχική συμβολοσειρά αντιστοιχεί σε δύο δεκαεξαδικούς χαρακτήρες στην κωδικοποιημένη συμβολοσειρά.

6.2.4.3.7 Ενσωματωμένα εντός εισαγωγικών εκτυπώσιμα δεδομένα

Ένα embedded quoted printable data αντικείμενο μπορεί να περιέχει δεδομένα τα οποία είναι κωδικοποιημένα σύμφωνα με τις MIME προδιαγραφές.

Παράδειγμα

<QuotedPrintableData>

This is a UK currency =C2=A3 symbol, and for =3D measure ...

</QuotedPrintableData>

6.2.4.3.8 Δεδομένα εξωτερικού σώματος του κώδικα

Ένα αντικείμενο external body μπορεί να χρησιμοποιηθεί όταν δεδομένα octet ορίζονται σε μία εξωτερική πηγή στην περιγραφή υπηρεσίας PCF, παρά όταν βρίσκονται ενσωματωμένα στην περιγραφή υπηρεσίας.

Τα αντικείμενα external body πρέπει να έχουν τις ακόλουθες ιδιότητες:

- **content-transfer-encoding** -ορίζει την κωδικοποίηση που χρησιμοποιείται μέσα στα αντικείμενα external body που περιέχουν octet δεδομένα. Η προκαθορισμένη τιμή για αυτήν την ιδιότητα τίθεται ως "binary", και η τιμή "binary" μπορεί να χρησιμοποιηθεί ως προκαθορισμένη τιμή όταν καμία άλλη κωδικοποίηση δεν ορίζεται.
- **content-type** - τύπος περιεχομένου που συμπεριλαμβάνεται μέσα στο αρχείο. Αυτό είναι οποιοσδήποτε τύπος MIME που καταχωρείται με Internet Assigned Numbers Authority (IANA). Αυτή η ιδιότητα πρέπει να ορισθεί αλλά μπορεί να αφεθεί κενή για τα συστήματα PCF για να καθορισθεί αυτόματα ο τύπος content των στοιχείων octet.
- **uri** - ταυτότητα του πόρου που περιέχει τα στοιχεία octet για τα αντικείμενα external body .

Ένα external body αντικείμενο μπορεί να περιέχει ένα ή περισσότερα meta property αντικείμενα που καθορίζουν την θέση της πηγής που περιέχει τα external octet δεδομένα.

Παράδειγμα Το παρακάτω XML πλαίσιο δείχνει ένα αντικείμενο octet data που περιέχει μία παραπομπή ενός αντικειμένου external body στο λογότυπο του ETSI

```

<ImageData name="logo">
  <ExternalBody content-type="image/png"
    content-transfer-encoding="binary"
    uri="http://www.etsi.org/images/logos/etsi-white-small.png">
    <MetaProperty name="comment" value="small ETSI logo"/>
  </ExternalBody>
</ImageData>

```

6.2.4.3.9 Αντικείμενο δεδομένων πολλαπλών τύπων

Τα αντικείμενα *Multipart data* μπορεί να περιέχουν ένα ή περισσότερα *octet data* το κάθε ένα από τα οποία μπορεί να θεωρηθεί ως μέρος ενός αντικειμένου *octet data* που περιέχεται μέσα στο αντικείμενο *multipart data*. Η ιδιότητα *content-type* για ένα *Multipart data* αντικείμενο μπορεί να είναι "multipart/alternative".

Παράδειγμα

Μια υπηρεσία πρόκειται να έχει ως στόχο τρεις πλατφόρμες με διαφορετικά πρότυπα χρώματος. Η πρώτη πλατφόρμα έχει μία σταθερή παλέτα, η δεύτερη έχει μία δυναμικά ορισμένη παλέτα και η τρίτη υποστηρίζει πλήρως 24-bit παλέτες χρώματος. Ένας συντάκτης υπηρεσιών παρέχει δύο εικόνες: μια εικόνα θα λειτουργήσει με την πρώτη και την δεύτερη παλέτα που είναι ορισμένες στις πλατφόρμες. Η άλλη εικόνα είναι μια αληθινή εικόνα χρώματος που είναι κατάλληλη για στην τρίτη πλατφόρμα. Αυτό παρουσιάζεται στο ακόλουθο πλαίσιο XML:

```

<ImageData name="logo">
  <MultipartData content-type="multipart/alternative">
    <ExternalBody content-type="image/png"
      content-transfer-encoding="binary"
      uri="file:///c:/images/logo_palette.png">
      <MetaProperty name="palette" value="30"/>
    </ExternalBody>
    <ExternalBody content-type="image/png"
      content-transfer-encoding="binary"
      uri="file:///c:/images/logo_original.png">
      <MetaProperty name="colours" value="full"/>
    </ExternalBody>
  </MultipartData>
</ImageData>

```


6.2.4.4 Τύποι αντικειμένων που αποτελούνται από δεδομένα octet

6.2.4.4.1 Αντικείμενα συμβολοσειρών που αποτελούνται από δεδομένα octet

Τα αντικείμενα *value* του τύπου δεδομένων *string* της PCF, μπορούν να αναπαραστήσουν την ιδιότητα *value* με στοιχεία *octet data container*. Τα δεδομένα που παρέχονται στο στοιχείο *octet data container* μπορεί να έχουν *content-type* «κείμενο».

Ο μόνος φορητός *content* τύπος για αντικείμενα *string octet data* μπορεί να είναι ο "text/plain" MIME τύπος με το ίδιο σύνολο χαρακτήρων κωδικοποίησης που υπάρχουν στο έγγραφο PCF στο οποίο δηλώνεται το αντικείμενο *string value*. Ένα σύστημα PCF μπορεί να μετασχηματίσει οποιονδήποτε σύνολο χαρακτήρων κωδικοποίησης μιας συμβολοσειράς σε μία αναπαράσταση PCF συμβολοσειράς σε πλατφόρμα-εξαρτώμενου τρόπου.

Παράδειγμα

Το παρακάτω αντικείμενο *string value* το οποίο αναπαριστάται σύμφωνα με το σχήμα PCF XML που συνδέεται με το παρόν έγγραφο, είναι ένα αντικείμενο *string octet data* που ορίζεται από το περιεχόμενο σε ένα εξωτερικό αρχείο.

```
<String name="outside">
  <ExternalBody content-type="text/plain; charset=utf-8"
    content-transfer-encoding="binary"
    uri="file:external_text.txt"/>
</String>
```

6.2.4.4.2 Μαρκιαρισμένο κείμενο και δεδομένα octet

Τα αντικείμενα *Value* που υπάρχουν στον τύπο δεδομένων *marked up text* μπορούν να αναπαραστήσουν την ιδιότητα *value* με στοιχεία *octet data container*. Τα δεδομένα που παρέχονται σε ένα στοιχείο *octet data container* μπορεί να έχουν τύπο *content* "text/x-dvb-pcf" ή "text/plain".

ΠΑΡΑΔΕΙΓΜΑ:

Το παρακάτω *marked up text value* αντικείμενο το οποίο αναπαριστάται σύμφωνα με το σχήμα PCF XML κωδικοποιείται χρησιμοποιώντας τη κωδικοποίηση base 64. Η τιμή αυτού αντικειμένου *octet* είναι ισοδύναμη με αυτή που παρουσιάζεται στο παράδειγμα στην παράγραφο 6.2.3.7.

```
<MarkedUpText name="markedUp">  
  <Base64Data content-type="text/x-dvb-pcf">
```

```
PGJvZHkgeG1sbnM9XFxcXFxcXCJodHRwOi8vd3d3LmR2Yi5vcmevcGNmL3gtZ  
HZiL
```

```
XBjZlxcXFxcXFwiPg0KICA8cD5FeGFtcGxIIHRleHQgd2l0aCA8ZW0+ZW1waG  
FzaXM8L2VtPi48L3A+DQogPC9ib2R5Pg0K</Base64Data>  
</MarkedUpText>
```

6.2.4.4.3 Αντικείμενα εικόνων που αποτελούνται από δεδομένα octet

Ο τύπος δεδομένων image στην PCF θα αναπαριστά image media δεδομένα σύμφωνα με τον τύπο «εικόνας» MIME όπως ορίζεται στις MIME προδιαγραφές. Τα δεδομένα εικόνας θα αναπαρασταθούν από octet data container και δεν θα έχουν οποιαδήποτε άλλη μορφή λεξικολογικής αναπαράστασης στην PCF. Όπως καθορίζεται στην παράγραφο 6.2.4.3.1, οι δύο μόνο φορητοί τύποι MIME για δεδομένα octet είναι οι "image/png" και "image/jpeg". Ένα σύστημα PCF μπορεί να είναι ικανό να μετασχηματίσει ή να χρησιμοποιήσει άλλους τύπους δεδομένων, όπως περιγράφεται στην παράγραφο 6.2.4.3.1.

Το παρακάτω image octet data αντικείμενο το οποίο αναπαριστάνεται σύμφωνα με το σχήμα PCF XML παρουσιάζει μία μικρή εικόνα βέλους κωδικοποιημένη ως δεδομένα base 64.

```
<ImageData name="upArrow">  
  <Base64Data content-type="image/png">
```

```
IVBORw0KGgoAAAANSUgAAAAoAAAKCAIAAAACUFjqAAAABmJLR  
0QA/wD/AP+gv
```

```
aeTAAANUIEQVR4nGP8//8/A27AhMxhZGRkZGTEKY1PN1wfsGHE6UazEs4lo  
JsBj8
```

```
  f+///PSIK/MQEANb4MF+DirHMAAAAASUVORK5CYII=  
  </Base64Data>  
</ImageData>
```

6.2.4.4 Αντικείμενα ροών δεδομένων octet

Ο τύπος δεδομένων *stream* αναπαριστά μια σύνθεση ενός ή περισσότερων στοιχειωδών ροών *media*, όπως βίντεο, ήχος, ή άλλα στοιχεία μέσω με ένα σχήμα ροής. Τα δεδομένα *stream* μπορούν να αναπαρασταθούν από στοιχεία *octet data container*.

Όπως καθορίζεται στην παράγραφο 6.2.4.3.1, οι τιμές του στοιχείου *content-type* για *stream* δεδομένα μπορεί να είναι "video/mpeg", "audio/mpeg", "audio/basic", "video/x-MP2PS", "video/x-MP2TS", "video/x-MP2TS-P", "video/x-MP2ES" και "application/octet-stream".

Παράδειγμα

Το παρακάτω *stream octet data* αντικείμενο το οποίο αναπαριστάνεται σύμφωνα με το σχήμα PCF XML, παρουσιάζει μια υπηρεσία τηλεοπτικής μετάδοσης ("BBC One") η οποία παραδίδεται ως μέρος μιας ροής MPEG-2

```
<StreamData name="tv">
  <ExternalBody content-type="video/MP2TS-P"
    content-transfer-encoding="binary"
    uri="urn:x-dvb-pcf:bbc.co.uk:bbc-one"/>
</StreamData>
```

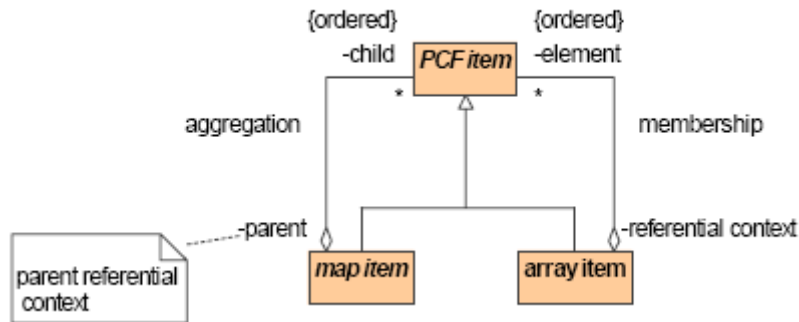
6.2.5 Σύυθετοι τύποι

6.2.5.1 Σύυθετοι τύποι δεδομένων

Οι τύποι δεδομένων *compound* είναι το σύνολο όλων των *containers* των PCF δεδομένων. Ο γενικός τύπος δεδομένων *compound* ειδικεύεται σε άλλους *compound* τύπους δεδομένων ως εξής

- παρέχοντας περιορισμούς που περιγράφουν τους τύπους δεδομένων που μπορούν να περιέχονται σε ένα *container*
- ορίζοντας δομές μέσα σε *container* που επιτρέπουν την αναφορά σε τιμές σε αυτά τα *containers*

Δύο υψηλού επιπέδου *compound* τύποι δεδομένων που συμπεριλαμβάνονται στην PCF, είναι οι *map* και *array* όπως φαίνεται στην εικόνα 26 .



Εικόνα 26. Σύνθετοι τύποι δεδομένων.

Τα δομικά στοιχεία μιας περιγραφής υπηρεσίας PCF προέρχονται από τον τύπο δεδομένων compound.

Κάθε αντικείμενο της PCF που είναι μια περίπτωση του τύπου δεδομένων compound θα έχει:

- Όλες τις ιδιότητες ενός στοιχείου PCF όπως καθορίζονται στην παράγραφο 6.2.1.1.
- Μια ιδιότητα **content** που περιέχει μηδέν ή περισσότερα PCF αντικείμενα
- Μια ιδιότητα **length** που είναι ο αριθμός των αντικειμένων στην ιδιότητα content του στοιχείου map container. Αυτή η ιδιότητα μπορεί να είναι μόνο ιδιότητα ανάγνωσης.

6.2.5.2 Ο τύπος δεδομένων χαρτογράφηση και το αντικείμενο χαρτογράφηση

Ο τύπος δεδομένων **map** μπορεί να είναι ένας σύνθετος (compound) τύπος δεδομένων που αναπαριστά όλες τις πιθανές ακολουθίες που περιέχουν ζευγη ονομάτων και PCF αντικειμένων. Ένα PCF αντικείμενο **map** είναι μια ακολουθία μηδενικών ή περισσότερων ζεύγων ονομάτων PCF αντικειμένων και είναι μια περίπτωση του τύπου δεδομένων **map**.

Κάθε στοιχείο PCF μέσα σε ένα αντικείμενο **map** θα έχει ένα όνομα που συνδέεται με αυτό και ορίζεται στην ιδιότητα name του αντικειμένου. Όλα τα ονόματα μέσα σε ένα αντικείμενο **map** θα πρέπει να είναι μοναδικά. Με τη βοήθεια της ενεργού περιγραφής PCF, θα είναι δυνατό να παρεμβληθεί ένα στοιχείο PCF μέσα σε ένα αντικείμενο map: στην έναρξη της ακολουθίας, στο τέλος της ακολουθίας, πριν από ένα άλλο στοιχείο στην ακολουθία ή μετά από ένα άλλο στοιχείο ακολουθίας. Τα αντικείμενα θα διαταχθούν αρχικά μέσα στην ακολουθία ενός αντικειμένου map σύμφωνα με τους κανόνες της δήλωσή τους.

Ένα αντικείμενο `map` θα έχει μόνο μία αρχική ιδιότητα ανάγνωσης που αναπαριστά το πρώτο αντικείμενο PCF στην ακολουθία μέσα στο αντικείμενο `map`.

Εάν το αντικείμενο `map` δεν περιέχει άλλα αντικείμενα οι αρχικές ιδιότητες που ορίζονται σε ένα αντικείμενο PCF θα είναι : η ιδιότητα `type` ορίζεται ως **boolean**, η ιδιότητα `value` ορίζεται ως "false" και η ιδιότητα `nil` ορίζεται ως "true". Ένα αντικείμενο `map` θα έχει μια ιδιότητα **parameter-list** η τιμή της οποίας θα είναι μία λίστα τιμών παραμέτρων. Θα είναι δυνατό να παραπεμφθεί ένα αντικείμενο PCF μέσα σε ένα αντικείμενο `map` χρησιμοποιώντας το μοναδικό όνομά του. Καθώς τα αντικείμενα `map` μπορεί να περιέχονται σε άλλα αντικείμενα `map`, ένας καθορισμός διαδρομής μπορεί να χρησιμοποιηθεί για την αναφορά των αντικειμένων σε περιγραφές που περιέχονται πολλά αντικείμενα `map`.

6.2.5.3 Ο τύπος δεδομένων πίνακας και το αντικείμενο πίνακας.

Ο τύπος δεδομένων **array**, είναι ένα σύνολο όλων των πιθανών ακολουθιών που περιέχουν αντικείμενα PCF όπου το καθένα είναι από τον ίδιο τύπο. Οι τιμές του τύπου δεδομένων `array` ονομάζονται αντικείμενα **array**. Ένα αντικείμενο `array` μπορεί να περιέχει μια καθορισμένη μήκους ακολουθία μηδενικών ή περισσότερων αντικειμένων PCF. Όλα τα PCF αντικείμενα μέσα σε ένα αντικείμενο `array` πρέπει να έχουν τον ίδιο τύπο δεδομένων. Η ιδιότητα `length` ενός αντικειμένου **array** είναι μια τελική παράμετρος, η οποία είναι επομένως μόνο για ανάγνωση μέσα στην ενεργή περιγραφή. Η παράμετρος αντιμε-τωπίζεται σύμφωνα με τους ακόλουθους κανόνες:

- Εάν η παράμετρος `length` είναι ορισμένη ,η δήλωση `array` δεν μπορεί να περιέχει περισσότερα αντικείμενα από αυτή την τιμή διαφορετικά θα αναφερθεί ένα λάθος.
- Εάν η παράμετρος `length` είναι ορισμένη και ο αριθμός των αντικειμένων είναι μικρότερος από την τιμή της παραμέτρου `length`, τα PCF αντικείμενα που έχουν δηλωθεί τοποθετούνται από την αρχή του πίνακα και οι υπόλοιπες κενές θέσεις θα γεμίσουν με αντικείμενα που έχουν μηδενική τιμή.
- Εάν η παράμετρος `length` δεν έχει ορισθεί, το μήκος του πίνακα θα είναι το ίδιο με τον αριθμό των PCF αντικειμένων που δηλώνονται μέσα στον πίνακα.

Παράδειγμα 1 1

Ένα αντικείμενο array που περιέχει αντικείμενα PCF τύπου string θα μπορούσε να έχει το όνομα "stringarray". Ένα αντικείμενο array που περιέχει string arrays θα μπορούσε να έχει το όνομα «stringarrayarray».

Τα PCF αντικείμενα που βρίσκονται μέσα σε ένα αντικείμενο array θα διαταχθούν και θα ταξινομηθούν σύμφωνα με τους κανόνες της δήλωσής τους. Οι τιμές των δεικτών θα πρέπει να ξεκινούν από το 0 για το πρώτο στοιχείο και η τιμή του δείκτη του τελευταίου στοιχείου θα πρέπει να είναι length-1.

Με τη βοήθεια της ενεργού περιγραφής της PCF, είναι εφικτό :

- να διαβαστεί η τιμή οποιουδήποτε στοιχείου ενός πίνακα από τον δείκτη του κάθε στοιχείου μέσα στον πίνακα .
- να επαναληφθούν όλα τα στοιχεία μέσα στον πίνακα.
- να αλλάξει η τιμή οποιουδήποτε στοιχείου μέσα στον πίνακα με μια άλλη τιμή του ίδιου τύπου.

Από την ενεργή περιγραφή, δεν θα είναι δυνατό να αλλάξει ο τύπος και το μήκος του πίνακα.

Παράδειγμα 2

Το ακόλουθο πλαίσιο XML παρουσιάζει δύο ισοδύναμες δηλώσεις αντικειμένων array τύπου string και είναι σύμφωνες με το σχήμα PCF XML.

```
<StringArray name="no_nil_values" length="4">
  <String value="First"/>
  <String value="Second"/>
</String>
```

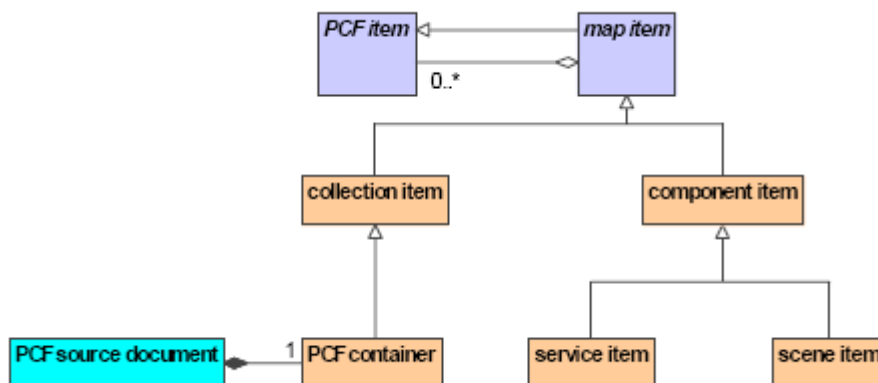
```
<StringArray name="nil_values_included" length="4">
  <String value="First"/>
  <String value="Second"/>
  <String nil="true"/>
  <String nil="true"/>
</String>
```

6.3 Δομή Περιγραφής Μιας Υπηρεσίας

Οι υπηρεσίες PCF περιγράφονται με την βοήθεια των αντικειμένων PCF, με δομικές περιγραφές υψηλού επιπέδου που θα πρέπει πάντα να παρουσιάζονται. Τα αντικείμενα περιγραφής που δημιουργούν μια υπηρεσία PCF παρουσιάζονται στην παράγραφο 6.3.1 και περιγράφονται λεπτομερέστερα στις παραγράφους 6.3.2 έως 6.3.6.

6.3.1 Αντικείμενα Περιγραφής

Τα αντικείμενα περιγραφής μια υπηρεσίας PCF είναι αντικείμενα service ,scene άλλα component και συλλογές αντικειμένων. Η σχέση μεταξύ αυτών των αντικειμένων παρουσιάζεται στο παρακάτω διάγραμμα .



Εικόνα 27. Τα δομικά αντικείμενα της PCF

Όλες οι περιγραφές PCF υπηρεσιών θα πρέπει να περιλαμβάνουν ένα PCF container. Ένα PCF container θα πρέπει να είναι κορυφαίο στοιχείο σε ένα έγγραφο PCF. Κάθε έγγραφο PCF θα πρέπει να έχει ακριβώς ένα μόνο PCF container. Όλα τα δομικά στοιχεία μιας περιγραφής υπηρεσίας PCF όπως το PCF container, component, collection Scene και service αντικείμενα θα είναι είδη του αντικειμένου map όπως περιγράφεται στην παράγραφο 6.2.5.2.

Παράδειγμα

Ένα έγγραφο PCF πρέπει να έχει ως στοιχείο ρίζας ένα PCF container οποίο αναπαριστάνεται ως ένα στοιχείο με το όνομα "PCF". Αυτό διευκρινίζεται στο ακόλουθο παράδειγμα:

```
<?xml version="1.0" encoding="UTF-8"?>
<PCF xmlns="http://www.dvb.org/pcf/pcf"
      xsi:schemaLocation="http://www.dvb.org/pcf/pcf pcf.xsd">

  <String name="example"
        value="Only PCF item in this PCF source document."/>
</PCF>
```

Τα χαμηλού επιπέδου containers μιας PCF περιγραφής υπηρεσίας είναι αντικείμενα τύπου collection. Αυτά επιτρέπουν σε έναν συντάκτη να ομαδοποιήσει αντικείμενα μέσα σε δομές της επιλογής του με έναν τρόπο που να είναι χρήσιμα για την συγκεκριμένη περιγραφή υπηρεσίας. Τα αντικείμενα Service και scene ανήκουν στο υψηλότερο επίπεδο μιας δομη-μένης περιγραφής.

6.3.2 Τα δομικά στοιχεία

Τα αντικείμενα Component παρέχουν μια δομημένη περιγραφή των διαφόρων περιπτώσεων των PCF components τύπων. Τα PCF components περιγράφονται στο κεφάλαιο 7.

Τα συστήματα PCF πρέπει να ερμηνεύσουν τα δομικά στοιχεία κατά την απόδοση μιας περιγραφής υπηρεσίας PCF. Υπό αυτήν τη μορφή, τα δομικά στοιχεία αντικείμενα είναι γνωστά ως **renderable** αντικείμενα. Ένα renderable αντικείμενο μπορεί να είναι ενεργό ή ανενεργό και μπορεί να ενεργοποιηθεί και να απενεργοποιηθεί από την ενεργή περιγραφή υπηρεσίας. Όταν το renderable αντικείμενο είναι ενεργό, πρέπει να παρουσιαστεί σε έναν θεατή και να είναι διαθέσιμο μέσα στην ενεργό περιγραφή PCF σύμφωνα με τους κανόνες που καθορίζονται για το renderable αντικείμενο. Τα δομικά στοιχεία θα πρέπει να ανήκουν στον τύπο δεδομένων map που περιγράφεται στην παράγραφο 6.2.5.2.

6.3.3 Το Αντικείμενο Συλλογή

Τα αντικείμενα Collection είναι ομάδες αντικειμένων PCF. Τα αντικείμενα Collection μπορούν να καθοριστούν από τους συντάκτες υπηρεσιών για να επιτρέψουν τη modular περιγραφή υπηρεσιών που παρέχει ιεραρχική αναφορά στα πλαίσιο και διευκολύνει την παραπομπή, όπως περιγράφεται στην παράγραφο 6.4. Τα αντικείμενα Collection είναι μέλη του τύπου δεδομένων map.

Παράδειγμα:

Ένα παράδειγμα ενός αντικειμένου collection σύμφωνα με το σχήμα PCF παρουσιάζεται στην συνέχεια. Το παράδειγμα παρουσιάζει μια συλλογή των ιδιοτήτων προεπιλογής στην εφαρμογή τους στα αντικείμενα news style σε μια υπηρεσία.

```
<Collection name="news_style">
  <Color name="textcolor" value="#DFDFDF"/>
  <String name="content" value="More news follows shortly ..."/>
  <Integer name="border-width" value="3"/>
</Collection>
```

6.3.4 Στοιχείο περίληψης της PCF (PCF container)

Ένα PCF container είναι ένα collection αντικείμενο το οποίο είναι το αντικείμενο ρίζα κάθε εγγράφου PCF. Ένα έγγραφο PCF μπορεί να προσδιοριστεί ως πόρος περιγραφής μια διαδραστικής υπηρεσίας που πρόκειται να προσφερθεί. Σε αυτή την περίπτωση, το PCF container στο έγγραφο PCF θα πρέπει να περιέχει μέσα σε αυτό ένα μοναδικό αντικείμενο υπηρεσίας το οποίο παρέχει το **root rendering context** για την υπηρεσία.

6.3.5 Το Αντικείμενο Σκηνή

Ο σκοπός του αντικειμένου scene είναι να περιγράψει μια χωρική και χρονικά συντονισμένη μονάδα περιγραφής της εμπειρίας του θεατή. Τα αντικείμενα scene είναι component αντικείμενα και επομένως, είναι renderable αντικείμενα .

Μόνο ένα αντικείμενο scene μέσα σε μία διαδραστική περιγραφή υπηρεσίας μπορεί να είναι ενεργό οποιαδήποτε στιγμή. Όταν ένα scene αντικείμενο είναι ενεργό θέτει την κατάσταση του ενεργή και ενεργοποιεί όλα τα renderable αντικείμενα που

υπάρχουν μέσα στο rendering περιεχόμενο. Όταν ένα scene αντικείμενο είναι απενεργοποιημένο θα απενεργοποιήσει όλα τα renderable αντικείμενα που υπάρχουν μέσα στο rendering περιεχόμενο του .

Τα PCF αντικείμενα που περιλαμβάνονται μέσα σε ένα αντικείμενο scene θα συμβάλουν στην εμπειρία του θεατή μόνο όταν το αντικείμενο scene που περιλαμβάνουν είναι ενεργό. Η κατάσταση τέτοιων PCF αντικειμένων θα διατηρηθεί μόνο όταν το αντικείμενο scene είναι ενεργό. Η κατάσταση τους θα αρχικοποιείται σε κάθε ενεργοποίηση του αντικειμένου scene.

6.3.6 Το Αντικείμενο Υπηρεσία

Ένα αντικείμενο service παρουσιάζει μία ολοκληρωμένη περιγραφή μιας διαδραστικής υπηρεσίας . Ένα αντικείμενο service είναι ένα component αντικείμενο και επομένως είναι ένα renderable αντικείμενο.

Ένα αντικείμενο service θα περιέχει, είτε άμεσα είτε από παραπομπή :

- όλα τα αντικείμενα scene που είναι μέρος της εμπειρίας του θεατή της υπηρεσίας που περιγράφεται
- οποιαδήποτε άλλα αντικείμενα PCF που είναι κοινά σε όλα τα αντικείμενα scene .

Η κατάσταση μιας περιόδου της εμπειρίας του θεατή σε μία χρονική στιγμή θα καθοριστεί από το συνδυασμό όλων των αντικειμένων που μπορούν να αποδοθούν με συγκεκριμένο τρόπο μέσα στο περιεχόμενο απόδοσης του αντικειμένου service και όλων των προανα-φερθέντων αντικειμένων μέσα στο ενεργό αντικείμενο scene.

Μόνο ένα scene αντικείμενο μπορεί να είναι ενεργό κάθε στιγμή. Η πλοήγηση σε ένα άλλο αντικείμενο scene θα προκαλέσει την απενεργοποίηση του αρχικού αντικειμένου scene και την ενεργοποίηση του στόχου αντικειμένου scene .

Ένα αντικείμενο service δεν πρέπει να περιέχει άλλα αντικείμενα service ή οποιουδήποτε άλλους «απογόνους» του. Όλα τα PCF αντικείμενα μέσα σε ένα αντικείμενο service θα πρέπει να διατηρήσουν την κατάσταση τους ανεξάρτητα από την ενεργοποίηση ενός αντικειμένου scene κατά την διάρκεια μιας περιόδου της υπηρεσίας .

Τα δομικά στοιχεία που συμβάλλουν στην εμπειρία του θεατή και περιγράφονται από όλα τα αντικείμενα scene θα πρέπει να περιλαμβάνονται μέσα σε ένα αντικείμενο service. Ένα αντικείμενο service μπορεί να παρέχει το root rendering content μιας περιγραφής υπηρεσίας .

Όταν μια υπηρεσία PCF αρχικοποιείται το σχετικό αντικείμενο service θα πρέπει να ενεργοποιηθεί καθώς και όλα τα αντικείμενα που αποδίδονται με συγκεκριμένο

τρόπο, εκτός από τα αντικείμενα scene μέσα στο περιεχόμενο απόδοσης του αντικείμενου service, θα πρέπει να ενεργοποιηθούν .

Αυτά τα αντικείμενα που αποδίδονται με συγκεκριμένο τρόπο παραμένουν ενεργοποιημένα κατά την διάρκεια μιας περιόδου κάποιας διαδραστικής υπηρεσίας μέχρι η περίοδος αυτή να τερματιστεί. Όταν η υπηρεσία τερματιστεί, όλα τα αντικείμενα που είναι απόγονοι των ανωτέρω αντικειμένων και το αντικείμενο service θα απενεργοποιηθούν.

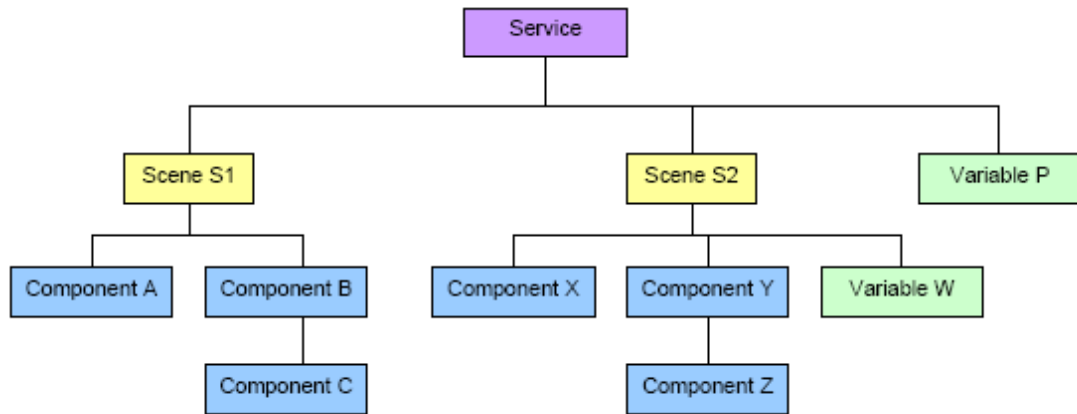
6.3.7 Κανόνες Αντικειμενοθέτησης

Οι κανόνες scoring καθορίζουν τον τρόπο με τον οποίο τα components μπορούν να έχουν πρόσβαση και να χειριστούν στις ιδιότητες μέσα σε άλλα components στην ιεραρχία των components χρησιμοποιώντας την ενεργή περιγραφή. Οι κανόνες scoring απεικονίζουν το πεδίο του χρόνου εκτέλεσης των μεταβλητών και τις ιδιότητες των components.

Η διάρκεια ζωής ενός component (αντικείμενου) που βρίσκεται μέσα σε ένα αντικείμενο scene θα πρέπει να είναι ίδια με τη διάρκεια ζωής του γονικού αντικείμενου scene. Ένα αντικείμενο scene και τα renderable αντικείμενα που είναι απόγονοι του, είναι ενεργο-ποιημένα μόνο όταν το ίδιο αντικείμενο scene είναι ενεργοποιημένο.

Οι κανόνες είναι οι εξής :

- ένα δομικό στοιχείο μπορεί να έχει πρόσβαση στις δικές του ιδιότητες και στις ιδιότητες όλων των δομικών στοιχείων που είναι απόγονοι του.
- στο επίπεδο service δεν υπάρχει πρόσβαση στις ιδιότητες των αντικειμένων scenes που περιέχονται στο αντικείμενο service ή σε οποιοδήποτε component υπάρχει μέσα στα αντικείμενα scene.



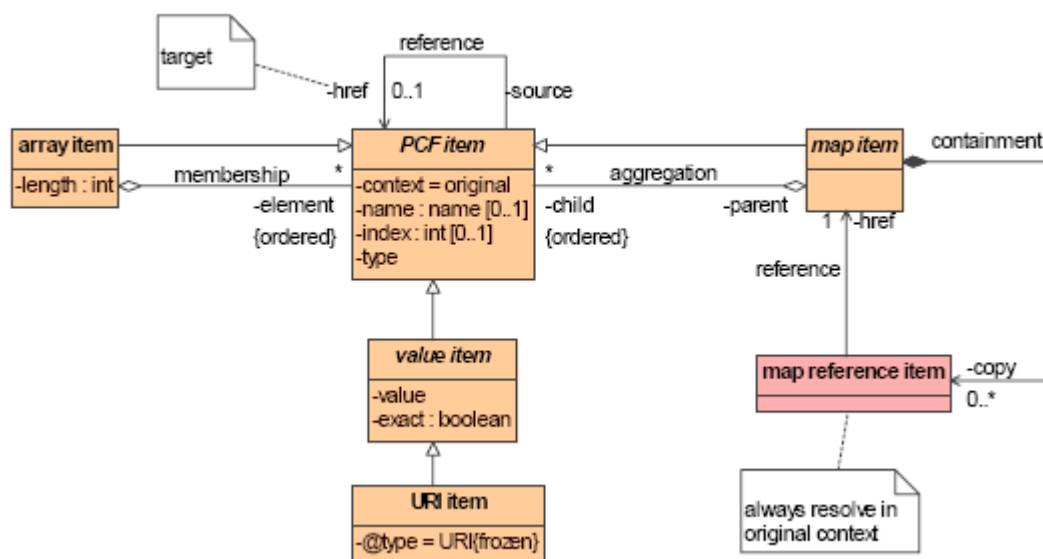
Εικόνα 28. Ένα παράδειγμα ιεραρχίας των δομικών στοιχείων της PCF.

Παράδειγμα

Ένα αντικείμενο Component C έχει μία ιδιότητα η οποία ονομάζεται fillcolor. Το αντικείμενο Component B μπορεί να έχει πρόσβαση στην τιμή C.fillcolor χρησιμοποιώντας την κατάλληλη action language.

6.4 Παραπομπή Και Πλοήγηση

6.4.1 Μοντέλο Παραπομπής



Εικόνα 29. Μοντέλο πλοήγησης και παραπομπής της PCF

6.4.2 Παραπομπή

6.4.2.1 Αντικείμενα παραπομπών της PCF

Τα αντικείμενα **references** επιτρέπουν σε οποιοδήποτε αντικείμενο της PCF να προσδιοριστεί από μία παραπομπή παρά από μια ρητά δηλωμένη τιμή. Αυτό θα επιτευχθεί με χρήση της ιδιότητας "href" που είναι διαθέσιμη για όλα τα αντικείμενα της PCF.

Οι παραπομπές γίνονται από ένα αντικείμενο PCF και αυτό το αντικείμενο είναι γνωστό ως **αντικείμενο source**. Οι παραπομπές γίνονται σε ένα PCF **αντικείμενο** και αυτό το αντικείμενο είναι γνωστό ως **target αντικείμενο**. Μια παραπομπή σε ένα αντικείμενο PCF μπορεί να είναι οποιοδήποτε αντικείμενο PCF με μια μη κενή ιδιότητα "href". Εάν η ιδιότητα "href" είναι μη κενή, η τιμή του PCF αντικειμένου θα καθοριστεί από την παραπομπή και οποιαδήποτε τοπική καθορισμένη τιμή θα αγνοηθεί. Η τιμή ενός αντικειμένου reference θα καθοριστεί με την ενσωμάτωση του περιεχομένου του αντικειμένου target της παραπομπής στη θέση του αντικειμένου source. Τα αντικείμενα source και target θα πρέπει να έχουν τον ίδιο τύπο. Εάν το όνομα του αντικειμένου source δεν έχει οριστεί και το όνομα του αντικειμένου target έχει οριστεί, τότε το όνομα του αντικειμένου source στην παραπομπή θα πρέπει να είναι το ίδιο με το αντικείμενο target.

6.4.2.2 Μορφή και ανάλυση μιας διαδρομής παραπομπής

Η μορφή της διαδρομής (path) μιας παραπομπής επιτρέπει σε ένα συντάκτη να προσδιορίσει μοναδικά οποιοδήποτε αντικείμενο σε μια περιγραφή υπηρεσίας PCF από το όνομα της διαδρομής. Δεδομένου ότι μια περιγραφή υπηρεσίας μπορεί να χωριστεί σε διάφορα έγγραφα πηγής PCF, η μορφή μιας PCF παραπομπής επιτρέπει την αναφορά μέσα σε ένα έγγραφο PCF αλλά και μεταξύ διαφορετικών εγγράφων PCF. Αυτό επιτυγχάνεται με την χρήση URIs (Uniform Resource Identifier).

Ο σκοπός της επίλυσης μια διαδρομής είναι για να καθορισθεί το αντικείμενο target όπου η διαδρομή αναφέρεται ή να φανεί αν η διαδρομή είναι άκυρη. Μία τιμή που παρουσιάζει μια διαδρομή ονομάζεται **PCF path**.

Τα αντικείμενα PCF που καθορίζονται από μία παραπομπή έχουν μία ιδιότητα **href** που προσδιορίζεται από μια "ref_path" παραγωγή. Το αντικείμενο **map** που είναι ο πιο κοντινός "πρόγονος" στο αντικείμενο source ονομάζεται **parent αντικείμενο** και όλα τα αντικείμενα που περιέχονται σε ένα αντικείμενο **map** είναι γνωστά ως **child items** ή **children**.

Το αντικείμενο **service** που είναι ο μακρινότερος πρόγονος από το στοιχείο source ονομάζεται **root service item**. Το PCF **container** που συνδέεται με το έγγραφο πηγής PCF ονομάζεται **root container item**.

Η μορφή μιας διαδρομής που ορίζεται χρησιμοποιώντας Extended Backhaus-Naur Form , μπορεί να είναι :

```
ref_path ::= (uri_part)? "#" ("/" | "~/")? path
uri_part ::= absoluteURI | relativeURI
path ::= path_item ("/")+ path | path_item
path_item ::= name (index)? | "." | ".."
index ::= "[" integer "]"
```

Μια PCF διαδρομή μπορεί να επιλυθεί σε μια σειρά βημάτων. Σε κάθε βήμα, η διαδρομή που θα επιλυθεί αφορά έναν συγκεκριμένο αντικείμενο PCF, γνωστό ως **relative item**, που καθορίζεται στο προηγούμενο βήμα. Επομένως, το πρώτο βήμα θα είναι να καθοριστεί το αρχικό *relative item*.

Όταν αναφέρεται το "uri_part" του "ref_path" τότε αυτό προσδιορίζει ρητά το PCF έγγραφο πηγής που περιέχει το αντικείμενο target (**target PCF source document**) και το PCF container θα είναι αρχική τιμή για το αντικείμενο relative .

Σε αυτήν την περίπτωση μπορεί να υπάρχει προαιρετικά το κυριολέκτημα "/" αλλά δεν επηρεάζει την επιλογή του αντικειμένου relative. Το προαιρετικό κυριολέκτημα "# ~" δεν επιτρέπεται.

Όταν το "uri_part" δεν υπάρχει τότε το **target PCF source document** θα είναι το PCF έγγραφο που περιέχεται στην ανάκληση της παραπομπής. Το αντικείμενο relative θα οριστεί έπειτα σύμφωνα με τους παρακάτω κανόνες

- Εάν υπάρχει το προαιρετικό κυριολέκτημα "/", η αρχική τιμή για το αντικείμενο relative θα είναι το αντικείμενο root container .
- Εάν υπάρχει το προαιρετικό κυριολέκτημα "~/", η αρχική τιμή για το αντικείμενο relative θα είναι το αντικείμενο root service.
- Εάν τα κυριολεκτήματα "/" ή "~/" δεν υπάρχουν, η αρχική τιμή για το αντικείμενο relative θα είναι το αντικείμενο parent του αντικειμένου source.

Παράδειγμα

Το παρακάτω πλαίσιο XML το οποίο είναι γραμμένο σύμφωνα με το σχήμα PCF XML δείχνει πως αναφέρονται τα στοιχεία σε ένα έγγραφο PCF το οποίο ονομάζεται "myservice.xml"

```

<!-- refer to PCF container as:
    - "myservice.xml#/" from outside or inside the source doc.
    - "#/" from inside the source document -->
<PCF>

<!-- refer to following integer "age" as:
    - "myservice.xml#/age" from outside or inside the source doc.
    - "#/age" from inside the source document -->
    <Integer name="age" value="32"/>
    <Service name="older">

<!-- refer to following integer "age" as:
    - "myservice.xml#/older/age" from out/inside the source doc.
    - "#/older/age" from inside the source document
    - "#age" locally
    - "#~/age" from in context derived from service "older" -->
    <Integer name="age" value="33"/>

<!-- transclude an age value from parent context - same as using
    "#/age" in original context, may differ in derived context -->
    <Integer name="oldage" href="#../age" context="original"/>
    </Service>
</PCF>

```

Εάν ένα "path " περιλαμβάνει ένα άλλο "path " τότε το αντικείμενο path θα επιλυθεί σε ένα νέο relative αντικείμενο και η διαδρομή που απομένει θα επιλυθεί στο επόμενο βήμα. Εάν ένα "path" αποτελείται από μόνο ένα αντικείμενο path τότε αυτό το αντικείμενο path θα επιλυθεί στο τελευταίο βήμα της ανάλυσης της διαδρομής και το αντικείμενο PCF που επιλύει θα είναι το αντικείμενο target στο οποίο αναφέρεται η διαδρομή .

Ο τρόπος με τον οποίο ένα "path_item" επιλύεται θα εξαρτηθεί από τον τρόπο με τον οποίο καθορίζεται, σύμφωνα με τους ακόλουθους κανόνες:

- ένα "όνομα" και ένας προαιρετικός "δείκτης", οι οποίοι θα οδηγήσουν σε ένα από τα "παιδιά" του αντικειμένου relative ως εξής:
 - όταν παρέχεται ένας "δείκτης" το όνομα θα αναφέρεται σε ένα αντικείμενο array η διαφορετικά η διαδρομή θα είναι άκυρη. Ο "δείκτης" είναι ένας μη αρνητικός αριθμός και θα αναφέρεται σε ένα αντικείμενο PCF ανάλογα με την σειρά που είναι διατεταγμένα και το "path_item" θα οδηγήσει σε αυτό το αντικείμενο. Εάν ο δείκτης είναι μεγαλύτερος από το μέγεθος του πίνακα η διαδρομή θα θεωρείται άκυρη.

- όταν δεν παρέχεται κανένας "δείκτης", η διαδρομή θα αναφέρεται σε ένα αντικείμενο PCF που είναι θα είναι "παιδί" του αντικειμένου relative.
- ένας χαρακτήρας τελειών ("."), όπου το "path_item" επιλύει το τρέχον relative αντικείμενο
- μια διπλή τελεία ("..") για να δείξει ότι το "path_item" θα επιλύσει το γονέα του relative αντικειμένου. Εάν δεν υπάρχει κάποιο αντικείμενο parent τότε η διαδρομή θα θεωρείται άκυρη.

6.4.3 Ανάλυση κειμένου

Οι διαδρομές που καθορίζονται από τις ιδιότητες "href" μιας παραπομπής ενός αντικειμένου PCF οδηγούν στα αρχικά δηλωμένα context ή σε ένα context το οποίο προέρχεται από εκεί όπου χρησιμοποιείται. Κάθε αντικείμενο PCF έχει μία ιδιότητα ως "derived". Η προκαθορισμένη τιμή για αυτήν την ιδιότητα θα είναι η "original".

Όταν η ιδιότητα "context" μιας παραπομπής ενός αντικειμένου PCF έχει τεθεί ως "original", τότε :

- Το αντικείμενο root container θα είναι το αντικείμενο container μέσα στο οποίο ορίζεται το reference
- Το αντικείμενο root service θα είναι ο ανώτατος πρόγονος του service component από το αντικείμενο της παραπομπής. Εάν δεν βρεθεί κάποιο αντικείμενο service το οποίο να είναι πρόγονος τότε η διαδρομή που καθορίζει το αντικείμενο reference είναι άκυρη.
- Η διαδρομή θα επιλυθεί σε ένα ιεραρχικό αναφερόμενο context που παρέχεται από το αντικείμενο root container και όλων των απογόνων του.

Εάν η ιδιότητα "context" ενός αντικειμένου reference τεθεί ως "derived" τότε θα ενεργοποιηθεί ένα αντικείμενο service και η παραπομπή θα είναι ενεργή. Στην συνέχεια θα πρέπει να χρησιμοποιηθούν οι ακόλουθοι κανόνες για την ανάλυση του περιεχομένου .

- Το αντικείμενο root container θα είναι το PCF container που είναι ο ανώτατος πρόγονος του ενεργοποιημένου αντικειμένου service.
- Το αντικείμενο root service θα είναι το ενεργοποιημένο αντικείμενο service
- Η διαδρομή θα επιλυθεί σε ένα ιεραρχικό αναφερόμενο context που παρέχεται από το αντικείμενο root container και όλων των απογόνων του.

Παράδειγμα

```
<PCF>
  <String name="title" value="this is not resolved here"/>
  <TextBox name="heading">
    <!-- ref cannot be resolved here: root service item unknown -->
    <String name="content" href="#../title" context="derived"/>
    <!-- other properties here -->
  </TextBox>

  <Service name="reuse">
    <!-- service properties omitted -->
    <Scene name="scene1">
      <String name="title" value="First scene"/>
      <TextBox href="#../heading" context="original"/>
    </Scene>

    <Scene name="scene2">
      <String name="title" value="Second scene"/>
      <TextBox href="#/heading" context="original"/>
    </Scene>
  </Service>
</PCF>
```

Όταν αναλυθούν όλες οι αναφορές θα είναι ισοδύναμο με το παρακάτω XML πλαίσιο:

```
<PCF>
  <Service name="reuse">

    <Scene name="scene1">
      <TextBox name="heading">
        <String name="content" value="First scene"/>
        <!-- other shared properties here -->
      </TextBox>
    </Scene>

    <Scene name="scene2">
      <TextBox name="heading">
        <String name="content" value="Second scene"/>
        <!-- other shared properties here -->
      </TextBox>
    </Scene>

  </Service>
</PCF>
```

6.4.4 Τα αντικείμενα παραπομπής Map

Ένα αντικείμενο **map reference** επιτρέπει σε όλο το περιεχόμενο ενός αντικειμένου map, του **αντικειμένου target map**, να αντιγραφεί στο περιεχόμενο ενός άλλου αντικειμένου, του **αντικειμένου source map**. Ένα αντικείμενο map μπορεί να μην περιέχει κανένα αντικείμενο reference ή να περιέχει πολλά αντικείμενα reference. Ένα αντικείμενο map έχει μια ιδιότητα **href** της οποίας η τιμή είναι μία διαδρομή η οποία οδηγεί στο αντικείμενο target map .

Όλα τα θυγατρικά αντικείμενα που υπάρχουν στο αντικείμενο target map θα αντιγραφούν μέσα στο αντικείμενο source map με την σειρά που είναι δηλωμένα έτσι ώστε αυτά να αντικαταστήσουν το αντικείμενο map reference. Εάν το αντικείμενο target map είναι κενό, η επίλυση του αντικειμένου map reference θα αφαιρέσει το αντικείμενο reference από το αντικείμενο source map.

Εάν, περισσότερα από ένα αντικείμενα PCF μέσα στο αντικείμενο source map έχουν το ίδιο όνομα θα εφαρμοστούν οι παρακάτω κανόνες πριν ολοκληρωθεί η ανάλυση της διαδρομής .

Για κάθε σύνολο αντικειμένων PCF με ίδιο επαναλαμβανόμενο όνομα μέσα στο αντικείμενο source map θα ισχύουν τα εξής :

- το στοιχείο με το επαναλαμβανόμενο όνομα με τον υψηλότερο δείκτη θα παραμείνει στο αντικείμενο source map στην ίδια σχετική θέση στην ακολουθία με τα άλλα αντικείμενα PCF.
- όλα τα άλλα αντικείμενα PCF με αυτό το όνομα θα αφαιρεθούν.

Παράδειγμα:

Το παρακάτω XML πλαίσιο , το οποίο είναι γραμμένο σύμφωνα με το PCF XML σχήμα, εμφανίζει ένα αντικείμενο map reference το οποίο εμφανίζεται ως το στοιχείο "Copy" και χρησιμοποιείται για να ενώσει δύο αντικείμενα collections μέσα στις ιδιότητες ενός component.

```
<Collection name="textinfo">
  <String name="content" value="Press red"/>
  <Color name="textcolor" value="#FF0000"/>
</Collection>
```

```
<Collection name="boxinfo">
  <Size name="size" value="100 30"/>
  <Position name="origin" value="150 210"/>
</Collection>
```

```
<TextBox name="instruction">
  <Copy href="#./textinfo">
  <Copy href="#./boxinfo">
  <!-- next property overrides that from "boxinfo" Copy -->
  <Position name="origin" value="600 60"/>
</TextBox>
```

6.4.5 Τα αντικείμενα παράμετροι

Η συμπεριφορά μέσα σε ένα δομικό στοιχείο μπορεί να έχει πρόσβαση μόνο στα δομικά στοιχεία τα οποία βρίσκονται στο ίδιο πλαίσιο με το δομικό στοιχείο μέσα στο οποίο περιλαμβάνονται. Τα αντικείμενα **parameter** παρέχουν ένα τρόπο ώστε να επιτρέπεται η συμπεριφορά να έχει πρόσβαση σε δομικά στοιχεία που βρίσκονται έξω από το περιλαμβανόμενο κείμενο

Ένα δομικό στοιχείο που δηλώνεται ως ομότιμο ενός δεύτερου δομικού στοιχείου μπορεί να περάσει μέσα στα συμφραζόμενα του δεύτερου δομικού στοιχείου, χρησιμοποιώντας τα αντικείμενα **parameter**

Τα δομικά στοιχεία **parameter** εμφανίζονται σαν να έχουν δηλωθεί τοπικά μέσα στο δομικό στοιχείο στο οποίο έχουν περάσει. Κατά συνέπεια μπορούν να προσεγγιστούν μέσα από τα συμφραζόμενα του δεύτερου δομικού στοιχείου. Σε κάθε αντικείμενο **parameter** θα δοθεί ένα νέο όνομα, ένα ψευδώνυμο, το οποίο πρέπει να είναι μοναδικό μέσα στο περιλαμβανόμενο δομικό στοιχείο και το οποίο θα χρησιμοποιηθεί για να αναφερθεί στο αντικείμενο το οποίο βρίσκεται μέσα στο περιεχόμενο του δομικού στοιχείου .

Παράδειγμα 1:

Το παρακάτω πλαίσιο XML παρουσιάζει πως μπορούν να χρησιμοποιηθούν τα αντικείμενα **parameter** ώστε να μοιράζονται μία κοινή μεταβλητή μεταξύ δύο **scene**. μεταβλητή μεταξύ δύο σκηνών. Η συμπεριφορά και στις δύο **scene** μπορεί να έχει πρόσβαση στη μεταβλητή, η οποία δεν χάνει της τιμή της κατά την μετάβαση στα **scene**.

```

<Service name="interactive_quiz">
  <IntegerVar name="score"/>
  <Scene name="level1" parameter-list="score:running_total">
    <Parameter name="running_total" class="IntegerVar"/>
    <!-- Behaviour here can now use running_total as if it was local -->
  </Scene>
  <Scene name="level2" parameter-list="score:running_total">
    <Parameter name="running_total" class="IntegerVar"/>
    <!-- Behaviour here has access to the same variable as in level1 -->
  </Scene>
</Service>

```

Παράδειγμα 2:

Το ακόλουθο πλαίσιο XML επεξηγεί τη μεταβαση μιας παραμέτρου μέσα στην ιεραρχία των δομικών στοιχείων της PCF ώστε να είναι προσβάσιμη από τα θυγατρικά δομικά στοιχεία.

```

<Service name="game">
  <StringVar name="userName"/>
  <Scene name="beginner" parameter-list="userName:name">
    <Parameter name="name" class="StringVar"/>
    <StaticELC name="score_banner" parameter-list="name:name">
      <Parameter name="name" class="StringVar"/>
      <!-- The userName variable in the service can be read and written
           here -->
    </StaticELC>
  </Scene>
</Service>

```

Παράδειγμα 3:

Το ακόλουθο πλαίσιο XML επεξηγεί τη χρήση μιας PCF διαδρομής για να περάσει μέσα σε ένα δομικό στοιχείο ένα αντικείμενο το οποίο περιέχεται σε ένα Collection.

```

<Service name="beginner">
  <Collection name="scoreboard">
    <StringVar name="name"/>
    <IntegerVar name="score"/>
  </Collection>
  <Scene name="scene1" parameter-list="scoreboard/name:name
scoreboard/score:score">
    <Parameter name="name" class="StringVar"/>
    <Parameter name="score" class="IntegerVar"/>
  </StaticELC>
</Scene>
</Service>

```

6.4.6 Τα αντικείμενα παραπομπών πλοήγησης

Τα αντικείμενα Navigation reference παρουσιάζουν συνδέσμους πλοήγησης μεταξύ αντικειμένων της PCF ή εξωτερικών αντικειμένων μιας περιγραφής υπηρεσίας PCF. Η τιμή ενός αντικειμένου navigation reference ονομάζεται στόχος πλοήγησης.

Παράδειγμα :

Το ακόλουθο πλαίσιο XML επεξηγεί πως μπορούν να συνδιαστούν τα αντικείμενα navigation reference με τα αντικείμενα παραπομπής της PCF.

```
<PCF>
  <ELC name="template1">
    <SceneNavigate>
      <URI name="target" href="#../content/next" context="derived"/>
    </SceneNavigate>
  </ELC>

  <Service>
    <Scene name="scene1">
      <ELC href="#/template1"/>
      <Collection name="content">
        <URI name="next" value="#~/scene2"/>
      </Collection>
    </Scene>
  </Service>
</PCF>
```

7 ΓΕΝΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΩΝ ΔΟΜΙΚΩΝ ΣΤΟΙΧΕΙΩΝ

7.1 Επισκόπηση

Τα components είναι οι λειτουργικές δομικές μονάδες μέσα σε μια περιγραφή υπηρεσίας PCF. Όλες οι τυποποιημένες και μη συνηθισμένες πτυχές μιας υπηρεσίας PCF περιγράφονται συνδυάζοντας διαφορετικούς τύπους component. Η PCF καθορίζει ένα τυποποιημένο εργαλειο-σύνολο από components για να εφαρμόσει τη συνήθως μια λειτουργία διαδραστικής υπηρεσίας που συνήθως χρησιμοποιείται .

Η PCF καθορίζει επίσης μια behaviour notation αρκετά σύνθετη για να επιτρέψει στα components που ανήκουν στο τυποποιημένο εργαλειοσύνολο να μπορούν να επεκταθούν και να συνδυαστούν.

Τα Components αρχικοποιούνται και διαχειρίζονται χρησιμοποιώντας ένα τυποποιημένο πρότυπο διεπαφών και ένα τυποποιημένο σύνολο κανόνων αντικειμενοθέτησης.

Ενώ το παρόν έγγραφο καθορίζει ποια είναι η προοριζόμενη εμφάνιση και η συμπεριφορά κάθε τύπου component, δεν διευκρινίζει πώς αυτή πρέπει να εφαρμοστεί. Η πραγματική εφαρμογή μπορεί να ποικίλει σημαντικά από πλατφόρμα σε πλατφόρμα και εξαρτάται από το υποκείμενο API κάθε πλατφόρμας και τους πόρους που χρησιμοποιεί.

Το μοντέλο αντικειμένων δομικών στοιχείων της PCF αποτελείται από δύο ξεχωριστά αλλά στενά συνδεδεμένα μοντέλα αντικειμένων

- Το **Component Specification Model**. Αυτό είναι ένα μετα-πρότυπο που χρησιμοποιείται για να καθορίσει ποιες κατηγορίες component είναι διαθέσιμες για χρήση μέσα στη PCF. Δεν είναι μέρος του προτύπου PCF για την περιγραφή διαδραστικών υπηρεσιών αλλά, περιγράφει τα χαρακτηριστικά των διεπαφών κάθε τύπου component .
- Το **Component Instantiation Model**. Περιλαμβάνει το σύνολο των αντικειμένων που χρησιμοποιούνται για την δηλώση και τον χειρισμό των component μέσα σε ένα έγγραφο PCF. Αυτό είναι μέρος του προτύπου PCF για την περιγραφή διαδραστικών υπηρεσιών και παρέχει τις βασικές δομικές μονάδες μιας περι-γραφής υπηρεσίας PCF.

Ο ορισμός μιας κατηγορίας component θα πρέπει να περιλαμβάνει.

- ένας πάροχο και ένα όνομα, τα οποία μαζί δημιουργούν ένα μοναδικό προσδιορισμό της κάθε κατηγορίας.
- τον ορισμό των διεπαφών του.
- ένα κείμενο που θα περιγράφει τις λεπτομέρειες αυτού του ορισμού.

7.2 Μοντέλο Προδιαγραφής δομικών στοιχείων

7.2.1 Επισκόπηση

Ο ορισμός των τύπων component αποτελείται από τα παρακάτω τρία τμήματα :

Interface Definition
Textual Description
Behaviour Specification.

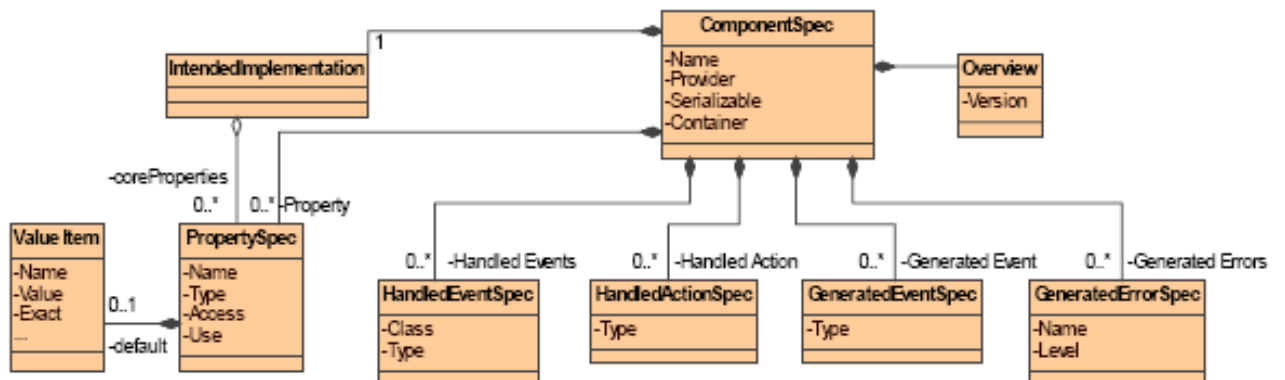
Οι πλατφόρμες στόχοι θα εφαρμόσουν τα τυποποιημένα PCF components σύμφωνα με τις προδιαγραφές τους. Οι νέοι τύποι component που θα προστεθούν στις μελλοντικές εκδόσεις των προδιαγραφών της PCF θα καθοριστούν επίσης χρησιμοποιώντας αυτό σχήμα προδιαγραφών.

Συνίσταται το σχήμα προδιαγραφών των component να χρησιμοποιείται για τον ορισμό οποιουδήποτε component που μπορεί να δημιουργηθεί. Ο καθορισμός των components με ένα τυποποιημένο σχήμα θα βοηθήσει στη συμβατότητα με άλλα εργαλεία της PCF, βοηθώντας τους συντάκτες να καταλάβουν τα components και διευκολύνουν την τυπο-ποίηση εάν κάποιο σύνηθες component υιοθετηθεί αργότερα ως τυποποιημένο PCF component.

7.2.2 Καθορισμός Διεπαφών

Το μοντέλο προδιαγραφών των component είναι το μετα-μοντέλο που χρησιμοποιείται για τον καθορισμό των χαρακτηριστικών της κάθε κατηγορίας **component**. Αυτό το μοντέλο αντικειμένων και οι σχετικοί συντακτικοί ορισμοί θα χρησιμοποιηθούν για τον προσδιορισμό των component κατηγοριών που είναι διαθέσιμες για χρήση μέσα σε μια περιγραφή υπηρεσίας PCF.

Αυτό το μοντέλο παρουσιάζεται στην εικόνα 33.



Εικόνα 33. Μοντέλο προδιαγραφών των δομικών στοιχείων

Αυτό το πρότυπο εφαρμόζεται με **Component Definition Syntax (CDS)**. Τα αντικείμενα μέσα στις προδιαγραφές του μοντέλου των component χρησιμοποιούνται για τον χαρακτηρισμό των χαρακτηριστικών διεπαφών των component κατηγοριών.

- Ένα στοιχείο **ComponentSpec** θα καθορίσει μια συγκεκριμένη **Component Class**. Ένα στοιχείο **ComponentSpec** θα πρέπει να έχει τις εξής ιδιότητες :
 - Ένα **provider(πάροχο)**. Αυτό θα προσδιορίσει το δημιουργό του στοιχείου component . Τα στοιχεία component που ορίζονται στο παρόν έγγραφο θα έχουν την προκαθορισμένη τιμή παρόχου "dvb.org".
 - Ένα αναγνωριστικό **όνομα (name)**. Αυτό θα είναι μοναδικό μεταξύ όλων των ονομάτων των component κατηγοριών από τον πάροχο.
 - Μια σημαία **serializable**. Αυτή θα καθορίσει εάν η κατηγορία component θα έχει μια σειριακή μορφή που μπορεί χρησιμοποιείται για την ανταλλαγή δεδομένων.
 - Μια σημαία **container**. Αυτή θα καθορίσει εάν υπάρχουν περιπτώσεις των component κατηγοριών που περιέχουν θυγατρικά component .

Ένα αντικείμενο **ComponentSpec** θα περιέχει:

- Ένα αντικείμενο **Properties** το οποίο μπορεί να περιέχει κανένα ή περισσότερα αντικείμενα **PropertySpec**. Αυτό καθορίζει το σύνολο ιδιοτήτων που μπορεί οριστεί ή να διαβαστεί σε κάποια περίπτωση κατηγορίας component . Αυτές οι ιδιότητες μπορούν είτε να οριστούν άμεσα χρησιμοποιώντας τα στοιχεία **PropertySpec**, ή συμπεριληφθούν ως μέρος μιας ομάδας ιδιοτήτων χρησιμοποιώντας τα στοιχεία **PropertyGroupRef**.
- Ένα προαιρετικό αντικείμενο **HandledEvents** το οποίο μπορεί να περιέχει κανένα ή περισσότερα αντικείμενα **HandledEventSpec**. Αυτό καθορίζει ένα σύνολο γεγονότων όπου ένα component αυτής της κατηγορίας μπορεί να ανταποκριθεί κατά τη διάρκεια της ζωής του. Αυτά τα γεγονότα μπορούν είτε να ορισθούν άμεσα χρησιμοποιώντας τα στοιχεία **HandledEventSpec**, ή να συμπεριληφθούν ως μέρος μιας ομάδας αντιμετωπίσιμων γεγονότων χρησιμοποιώντας τα στοιχεία **HandledEventGroupRef**.
- Ένα προαιρετικό αντικείμενο **HandledActions** που περιέχει **Properties** και το οποίο μπορεί να περιέχει κανένα ή περισσότερα αντικείμενα **HandledActionSpec**. Αυτό καθορίζει ένα σύνολο ενεργειών όπου ένα

component αυτής της κατηγορίας μπορεί να ανταποκριθεί κατά τη διάρκεια της ζωής του. Αυτά τα γεγονότα μπορούν είτε να ορισθούν άμεσα χρησιμοποιώντας τα στοιχεία **HandledActionSpec**, ή συμπεριληφθούν ως μέρος μιας ομάδας αντιμετώπισιμων ενεργειών χρησιμοποιώντας τα στοιχεία **HandledActionGroupRef**.

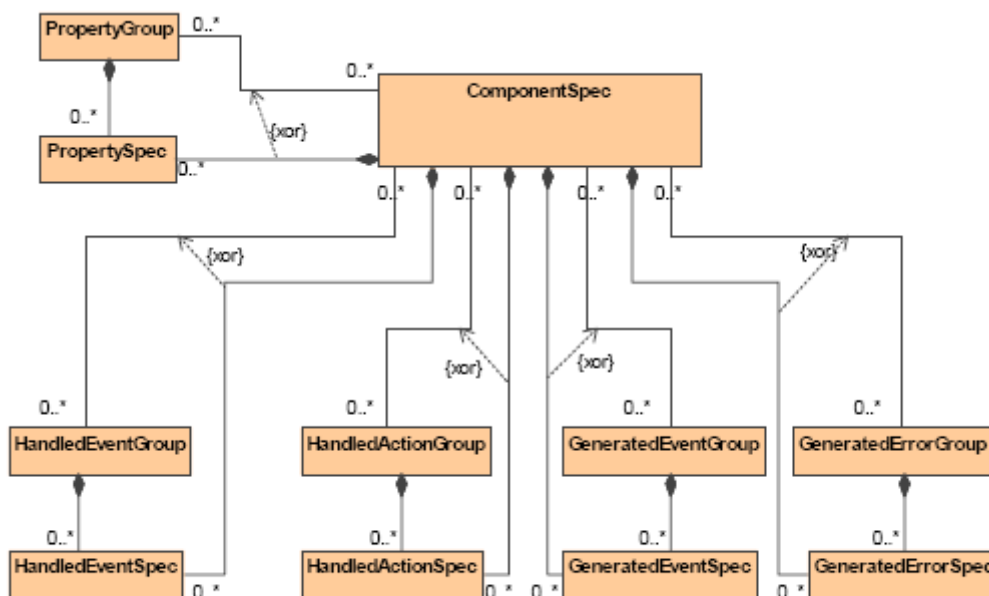
- Ένα προαιρετικό αντικείμενο **GeneratedEvents** που μπορεί να περιέχει κανένα ή περισσότερα αντικείμενα **GeneratedEventSpec**. Αυτό καθορίζει ένα σύνολο γεγονότων όπου ένα component αυτής της κατηγορίας μπορεί να δημιουργήσει κατά τη διάρκεια της ζωής του. Αυτά τα γεγονότα μπορούν είτε να ορισθούν άμεσα χρησιμοποιώντας τα στοιχεία **GeneratedEventSpec**, ή συμπεριληφθούν ως τμήμα μιας ομάδας παραγόμενων γεγονότων χρησιμοποιώντας τα στοιχεία **GeneratedEventGroupRef**.
- Ένα προαιρετικό αντικείμενο **GeneratedErrors** που μπορεί να περιέχει κανένα ή περισσότερα αντικείμενα **GeneratedErrorSpec**. Αυτό καθορίζει το σύνολο των λανθασμένων γεγονότων όπου ένα component αυτής της κατηγορίας μπορεί να δημιουργήσει κατά τη διάρκεια ζωής της. Αυτό το σύνολο μπορεί περιέχει όλα τα λάθη που παράγονται οποιασδήποτε έμφυτης συμπεριφοράς που καθορίζεται για αυτό το component. Αυτά τα λάθη μπορούν είτε να καθορισθούν άμεσα χρησιμοποιώντας τα στοιχεία **GeneratedErrorSpec**, είτε συμπεριληφθούν ως τμήμα μιας ομάδας παραγόμενων λαθών χρησιμοποιώντας τα στοιχεία **GeneratedErrorGroupRef**.
- Ένα αντικείμενο **Overview**. Το αντικείμενο overview περιέχει λεπτομερείς ιδιότητες που αφορούν την έκδοση της κατηγορίας component. Είναι [προβλέψιμο](#) ότι μερικοί τύποι component μπορούν να εξελεγχθούν κατά τη διάρκεια του χρόνου, έτσι ένας αριθμός έκδοσης επιτρέπει τον προσδιορισμό διαφορετικών εκδόσεων ενός component.
- Ένα αντικείμενο **IntendedImplementation**. Αυτό έχει μια μοναδική ιδιότητα **coreProperties**, και αυτή η ιδιότητα είναι μία λίστα ιδιοτήτων που **πρέπει** να εφαρμοστούν σε μια πλατφόρμα στόχο. Αυτές είναι οι ιδιότητες ενός component που απαιτούνται για να εξασφαλιστεί ότι όλες οι εφαρμογές παρέχουν μια ισοδύναμη εμπειρία στο χρήστη.

7.2.2.1 Ομάδες

Πολλοί τύποι component έχουν κοινά χαρακτηριστικά γνωρίσματα. Διαφορετικοί τύποι component μπορούν:

- να έχουν κοινές ιδιότητες, όπως το χρώμα ή τη θέση .
- να ανταποκριθούν στα ίδια γεγονότα, όπως ένα γεγονός που αφορά το πάτημα ενός κουμπιού.
- να ανταποκριθούν στις ίδιες ενέργειες, όπως ενέργειες παρουσιάσης /απόκρυψης.
- παράγουν τα ίδια γεγονότα, όπως OnSelect.
- παράγουν τα ίδια λάθη, όπως OnInvalidMediaType.

Όταν αυτό συμβαίνει, τα κοινά χαρακτηριστικά γνωρίσματα μπορούν να δηλωθούν ξεχωριστά από τις προδιαγραφές του component ως μέρος μιας ομάδας. Αυτή η ομάδα θα συμπεριληφθεί έπειτα με χρήση της αναφοράς μέσα στις προδιαγραφές του component.



Εικόνα 31 Μοντέλο προδιαγραφών δομικών στοιχείων στο οποίο παρουσιάζονται ομάδες

- Κάθε ιδιότητα που καθορίζεται μέσα σε μια ομάδα ιδιοτήτων θα πρέπει έχει ένα όνομα που θα είναι μοναδικό σε όλες τις ομάδες ιδιοτήτων .
- Κάθε γεγονός που ορίζεται μέσα σε μια ομάδα γεγονότων θα πρέπει να έχει ένα όνομα που θα είναι μοναδικό σε όλες τις ομάδες γεγονότων.
- Κάθε ενέργεια που ορίζεται μέσα σε μια ομάδα ενεργειών θα πρέπει να έχει ένα όνομα που θα είναι μοναδικό σε όλες τις ομάδες ενεργειών.
- Κάθε λάθος που καθορίζεται μέσα σε μια ομάδα λαθών θα πρέπει να έχει ένα όνομα που θα είναι μοναδικό σε όλες τις ομάδες λαθών.

Παράδειγμα:

Τα συνήθη component θα πρέπει να χρησιμοποιούν τις τυποποιημένες ομάδες όπου απαιτείται. Οι συνήθεις ομάδες μπορούν να οριστούν και μπορούν να χρησιμοποιηθούν μέσα στον ορισμό των συνήθων component. Όλα τα components που έχουν σύνορα, όπως το ορθογώνιο, το πολύγωνο, το TextBox περιλαμβάνουν μία ομάδα ιδιοτήτων **border_properties** μέσα στο σύνολο των ιδιοτήτων τους. Η ομάδα ιδιοτήτων **border_properties** περιέχει τις ιδιότητες όπως **linestyle** και **border-width**. Αυτό σημαίνει ότι ισοδύναμες ιδιότητες συνόρων μπορούν να αναφερθούν με συνέπεια σε όλους τους τύπους component. Απλοποιεί επίσης τις προδιαγραφές κάθε τύπου, αφού αντί να ορίζουμε κάθε ιδιότητα συνόρων για κάθε τύπο component, οι ιδιότητες συνόρων ορίζονται μία φορά στην ομάδα ιδιοτήτων και έπειτα η ομάδα αυτή συμπεριλαμβάνεται στις προδιαγραφές κάθε component.

Όταν ένα component δηλώνεται μέσα σε μια υπηρεσία PCF, οι ιδιότητες που μπορούν να δηλωθούν για αυτό component είναι η ένωση όλων των ιδιοτήτων μέσα στις ομάδες ιδιοτήτων που καθορίζονται για αυτό τον τύπο component, και οποιεσδήποτε άλλες μεμονωμένες ιδιότητες που καθορίζονται για αυτή την κατηγορία component.

Ένα ειδικό χαρακτηριστικό γνώρισμα των αντικείμενων property που ορίζεται ως μέρος της ομάδας ιδιοτήτων είναι ότι οι δηλώσεις των component μπορούν να κληρονομήσουν τιμές για αυτά τα αντικείμενα από ένα component ο οποίος είναι πρόγονος.

Οι ομάδες ιδιοτήτων χρησιμοποιούνται μόνο ως μέσο δήλωσης ιδιοτήτων μέσα στις προδιαγραφές των τύπων component και δεν χρησιμοποιούνται μονες τους μέσα σε μια περιγραφή υπηρεσίας PCF. Επομένως κατά την ρύθμιση των τιμών ιδιοτήτων σε κάποια περίπτωση ενός component η μόνη η διάκριση μεταξύ των ιδιοτήτων της ομάδας ιδιοτήτων και της συγκεκριμένης κατηγορίας ιδιοτήτων είναι η κληρονομικότητα που επιτρέπεται για τις ιδιότητες της ομάδας ιδιοτήτων.

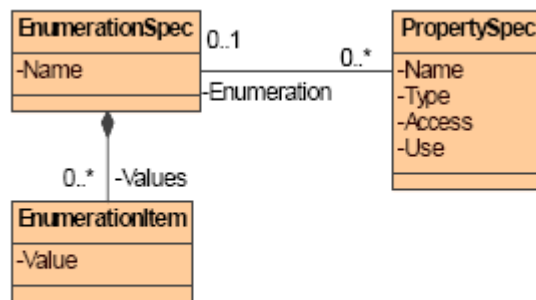
Το αντικείμενο **ComponentSpec** μπορεί να περιέχει ένα αντικείμενο **Properties**. Μέσα στο αντικείμενο **Properties** μπορεί να υπάρχουν αντικείμενα **PropertySpec** που ορίζουν συγκεκριμένες ιδιότητες σε μια κατηγορία component και **PropertyGroupRef** αντικείμενα, τα οποία παραπέμπουν σε ιδιότητες που περιλαμβάνονται στα αντικείμενα **PropertyGroup** και τα οποία δηλώνονται έξω από το component.

Τα αντικείμενα **PropertySpec** έχουν τις ακόλουθες ιδιότητες:

- Ένα **όνομα (name)**. Εάν το αντικείμενο **PropertySpec** είναι μέρος μιας ομάδας ιδιοτήτων, το όνομα θα πρέπει να είναι μοναδικό μεταξύ όλων των ιδιοτήτων που περιλαμβάνονται στην ομάδα ιδιοτήτων.
- Ένας **τύπος**. Αυτό θα διευκρινίσει τον τύπο **value item**, από το σύνολο των διαθέσιμων τύπων της PCF (integer, boolean, string, imageData κ.λπ.) που μπορούν να χρησιμοποιηθούν για να παρέχουν αυτήν την ιδιότητα μέσα σε μια δήλωση ενός component. Εάν ένα **PropertySpec** είναι τύπου **enumeration** τότε μπορεί να περιέχει ή να παραπέμπει σε ένα **EnumerationSpec** αντικείμενο που καθορίζει τις επιτρεπόμενες τιμές για την ιδιότητα.

7.2.2.3 Προδιαγραφές Απαρίθμησης

Ο τύπος δεδομένων enumeration ορίζεται λεπτομερώς στην παράγραφο 6.2.2.3. Τα αντικείμενα **EnumerationSpec** επιτρέπουν την δήλωση απαριθμήσεων. Μπορούν να δηλωθούν άμεσα μέσα στο αντικείμενο **PropertySpec** στο οποίο χρησιμοποιούνται, ή έξω από οποιαδήποτε αντικείμενο **ComponentSpec**, και να παραπέμπονται μέσα από τα αντικείμενα **PropertySpec** χρησιμοποιώντας ένα αντικείμενο **EnumerationRef**. Η σχέση μεταξύ των αντικειμένων **PropertySpec** και **EnumerationSpec** διευκρινίζεται στην εικόνα 32.



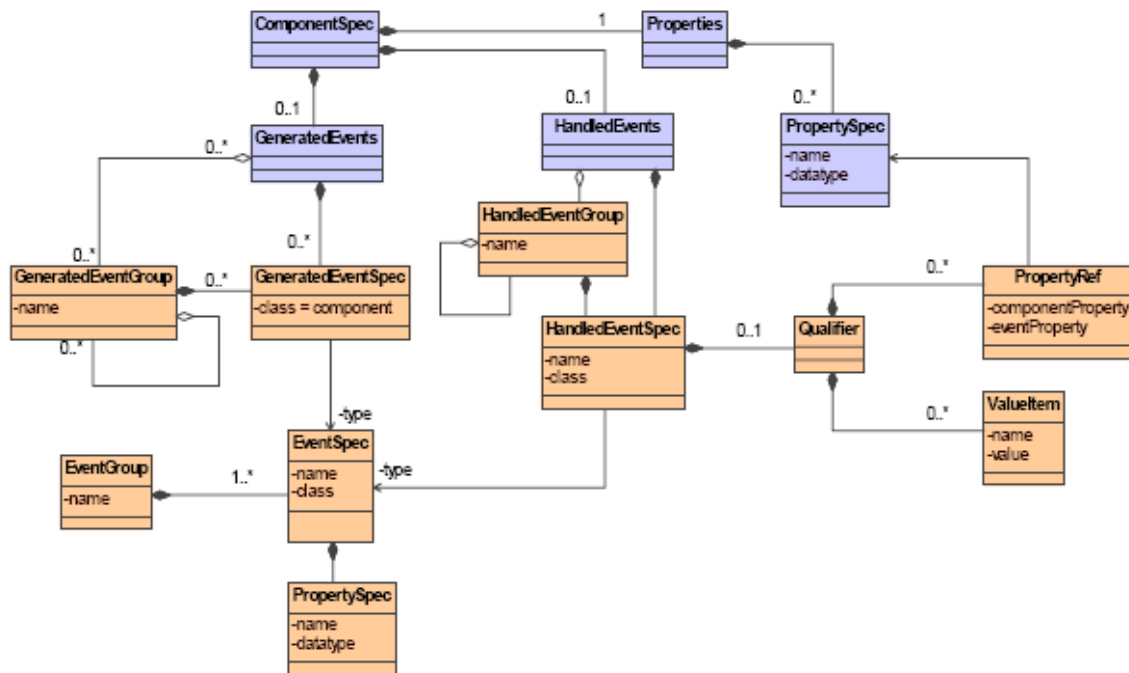
Εικόνα 32. Προδιαγραφές αντικειμένων απαρίθμησης

Ένα αντικείμενο EnumerationSpec μπορεί

- να έχει ένα όνομα
- να περιέχει ένα ή περισσότερα **EnumerationItems** καθένα από τα οποία ορίζουν μια πιθανή τιμή
- να έχει μια προκαθορισμένη ή οποιαδήποτε άλλη τιμή χρησιμοποιώντας ένα αντικείμενο String value

7.2.2.4 Προδιαγραφές Αντιμετωπίσιμων γεγονότων

Τα Components μπορούν να χειριστούν και να δημιουργήσουν γεγονότα. Η εικόνα 33 επεξηγεί πώς τα γεγονότα διαμορφώνονται στις προδιαγραφές της κατηγορίας component.



Εικόνα 33. Μοντέλο γεγονότων των δομικών στοιχείων

Το αντικείμενο **ComponentSpec** μπορεί να περιέχει ένα αντικείμενο **HandledEvents**. Μέσα στο αντικείμενο **HandledEvents** μπορεί να υπάρχουν αντικείμενα **HandledEventSpec**, τα οποία ορίζουν συγκεκριμένα αντιμετώπισιμα γεγονότα για την κατηγορία component, και **HandledEventGroupRef** αντικείμενα, τα οποία παραπέμπουν σε αντιμετώπισιμα γεγονότα που

περιλαμβάνονται στα αντικείμενα **HandledEventGroup** και είναι δηλωμένα έξω από το component .

Τα αντικείμενα **HandledEventSpec** έχουν τις ακόλουθες ιδιότητες:

- Ένα προσδιορισμό **class**. Αυτός ορίζει σε ποια κατηγορία γεγονός αναφέρεται το HandledEventSpec. Τα Component θα πρέπει να μπορούν να διαχειρίζονται γεγονότα που ανήκουν στις κατηγορίες **user** ή **system**. Τα Components δεν πρέπει να διαχειρίζονται γεγονότα που δημιουργούνται από άλλα Components .
- Ένας προσδιορισμός **type**. Αυτός διευκρινίζει τον τύπο του γεγονότος και συνδέει το HandledEventSpec με ένα γεγονός

Ένα αντικείμενο HandledEventSpec μπορεί να περιέχει έναν χαρακτηρισμό, ο οποίος περιορίζει τα γεγονότα που αντιμετωπίζονται βασισμένος στην ιδιότητα value του γεγονότος. Ο χαρακτηρισμός περιέχει:

- κανένα ή περισσότερα **PropertyRefs**. Αυτά δίνουν το όνομα της ιδιότητας στο γεγονός και το όνομα μιας ιδιότητα μέσα στο component. Η ιδιότητα μέσα στο component πρέπει να είναι του ίδιου τύπου με την ιδιότητα του γεγονότος που προσδιορίζεται.
- κανένα ή περισσότερα **ValueItems**. Αυτά δίνουν το όνομα της ιδιότητας στο γεγονός και μια τιμή. Η τιμή πρέπει να είναι ίδιου τύπου με την ιδιότητα του γεγονότος.

Ένα γεγονός θα αντιμετωπίζεται μόνο εάν οι τιμές των ιδιοτήτων του ταιριάζουν με όλες εκείνες που ορίζονται στην δήλωση του γεγονότος.

7.2.2.5 Προδιαγραφές παραγόμενων γεγονότων

Ένα αντικείμενο **ComponentSpec** μπορεί να περιέχει ένα αντικείμενο **GeneratedEvents**. Μέσα το αντικείμενο **GeneratedEvents** μπορεί να υπάρχουν **GeneratedEventSpec** αντικείμενα, τα οποία ορίζουν τα γεγονότα που παράγονται για την συγκεκριμένη κατηγορία component , και **GeneratedEventGroupRef** αντικείμενα τα οποία έχουν παραπομπές σε γεγονότα τα οποία έχουν παραχθεί και περιέχονται στα αντικείμενα **GeneratedEventGroup**. Τα αντικείμενα **GeneratedEventSpec** που συνδέονται με ένα **ComponentSpec** αντικείμενο καθορίζουν όλα τα component γεγονότα που παράγονται ως μέρος της συμπεριφοράς που ορίζεται από την έμφυτη συμπεριφορά για αυτήν κατηγορία component .

Τα αντικείμενα **GeneratedEventSpec** τα στοιχεία έχουν την παρακάτω ιδιότητα:

- Ένας προσδιορισμό **type**. Αυτό διευκρινίζει τον τύπο γεγονότος και συνδέει το **GeneratedEventSpec** με ένα γεγονός προδιαγραφών .

7.2.2.6 Προδιαγραφές Αντιμετωπίσιμων Ενεργειών

Τα αντικείμενα **ComponentSpec** μπορεί να περιέχουν ένα αντικείμενο **HandledActions**. Μέσα στο αντικείμενο **HandledActions** μπορεί να υπάρχουν **HandledActionSpec** αντικείμενα, τα οποία καθορίζουν τις αντιμετωπίσιμες ενέργειες σε μία συγκεκριμένη κατηγορία component, και **HandledActionGroupRef** αντικείμενα, τα οποία παραπέμπουν σε αντιμετωπίσιμες ενέργειες που περιέχονται στα αντικείμενα **HandledActionGroup**. Τα αντικείμενα **HandledActionSpec** που ανήκουν σε ένα **ComponentSpec** μπορούν να καθορίσουν όλες τις ενέργειες που προκαλούν την trigger behaviour μέσα στην inherent behaviour που ορίζεται για αυτή την κατηγορία component .

7.2.2.7 Προδιαγραφές παραγόμενων λαθών

Ένα αντικείμενο **ComponentSpec** μπορεί να περιέχει ένα **GeneratedErrors** αντικείμενο. Μέσα στο αντικείμενο **GeneratedErrors** μπορεί να υπάρχουν **GeneratedErrorSpec** αντικείμενα τα οποία καθορίζουν τα λάθη που δημιουργούνται για μια συγκεκριμένη κατηγορία component ,και **GeneratedErrorGroupRef** αντικείμενα, τα οποία παραπέμπουν σε λάθη που έχουν δημιουργηθεί και περιέχονται στα αντικείμενα **GeneratedErrorGroup**. Τα αντικείμενα **GeneratedErrorSpec** που ανήκουν σε ένα **ComponentSpec** μπορούν να καθορίσουν όλα τα component λάθη που έχουν δημιουργηθεί ως μέρος της συμπεριφοράς που ορίζεται από την έμφυτη συμπεριφορά για αυτή την κατηγορία component .

Τα αντικείμενα **GeneratedErrorSpec** έχουν την ακόλουθη ιδιότητα:

- Ένας προσδιορισμό **type**. Αυτός καθορίζει τον τύπο του λάθους και συνδέεται με το αντικείμενο **GeneratedErrorSpec** .

7.2.2.8 Προοριζόμενη εφαρμογή

Το αντικείμενο **ComponentSpec** μπορεί να περιέχει ένα αντικείμενο **IntendedImplementation**. Ένα αντικείμενο **IntendedImplementation** έχει:

- μια ιδιότητα **coreProperties**. Αυτή είναι μία λίστα με τα ονόματα όλων των component ιδιοτήτων που μπορούν να εφαρμοστούν στο ελάχιστο από μια εφαρμογή στόχο.

Η λίστα των core ιδιοτήτων καθορίζει τους βαθμούς ελευθερίας που μπορούν να εφαρμοστούν σε ένα component τύπο. Οι core ιδιότητες καθορίζουν τις εφαρμογές που ένα component μπορεί να εφαρμόσει. Όλες οι άλλες ιδιότητες καθορίζουν τι πρέπει ιδανικά να εφαρμόσει ένα component.

Αυτοί οι βαθμοί ελευθερίας επιτρέπουν στις πλατφόρμες χαμηλών-προδιαγραφών να εφαρμόσουν το ελάχιστο σύνολο λειτουργιών ενώ δεν περιορίζουν τις πλατφόρμες υψηλών-προδιαγραφών σε αυτό το ελάχιστο όριο.

7.2.2.9 Αντικείμενο επισκόπησης

Ένα αντικείμενο **ComponentSpec** μπορεί να περιέχει ένα αντικείμενο **Overview**. Ένα αντικείμενο **Overview** έχει:

- Μια ιδιότητα **version**. Αυτή δείχνει τον αριθμό έκδοσης της κατηγορίας component που δηλώνεται από **ComponentSpec** αντικείμενο.
- Προβλέπεται ότι μερικοί component τύποι μπορούν να εξελιχθούν κατά τη διάρκεια του χρόνου, έτσι ο αριθμός έκδοσης επιτρέπει τον προσδιορισμό διαφορετικών εκδόσεων του ίδιου component .

7.2.3 Κειμενική Περιγραφή

Το τμήμα κειμενικής περιγραφής μιας component κατηγορίας θα πρέπει να εκπληρεί δυο σκοπούς :

- να περιγράφει πώς συγκεκριμένες ιδιότητες μέσα στον ορισμό των διεπαφών επηρεάζουν την εμφάνιση ή/και την συμπεριφορά κάποιας component κατηγορίας.
- να παρέχει μια ανθρώπινη και ευανάγνωστη περιγραφή της component κατηγορίας.

Κάθε ιδιότητα που καθορίζεται μέσα στην περιγραφή των διεπαφών θα έχει κάποιο σχετικό κείμενο μέσα στην κειμενική περιγραφή για την εξήγηση της σημασίας της ιδιότητας για αυτόν τύπο component.

Η ερμηνεία θα πρέπει να γίνεται λεπτομερώς έτσι ώστε, να διευκρινίζεται πώς η αρχική τιμή, και επακόλουθες τροποποιήσεις στην τιμή της ιδιότητας επηρεάζουν τη εμφάνιση και την συμπεριφορά μιας περίπτωσης component για την οποία η ιδιότητα εφαρμόζεται.

Εκτός από την περιγραφή των μεμονωμένων τιμών θα πρέπει να υπάρχει και μια περιγραφή του γενικού component. Αυτή θα παρέχει μια επισκόπηση της εμφάνισης και της συμπεριφοράς του component και θα πρέπει να είναι ικανοποιητικά λεπτομερές έτσι ώστε, σε συνδυασμό με τις περιγραφές της ιδιότητας και την προδιαγραφή συμπεριφοράς, να παρέχει μια πλήρη περιγραφή της εμφάνισης και της συμπεριφορά μιας περίπτωσης component αυτής της κατηγορίας.

7.2.4 Προδιαγραφές Συμπεριφοράς

Οι προδιαγραφές συμπεριφοράς ενός component θα πρέπει καθορίζουν τη συμπεριφορά της component κατηγορίας, από την άποψη σχετικά πώς μια περίπτωση μιας component κατηγορίας αλληλεπιδρά με το υπόλοιπο της υπηρεσίας πέρα από τη διεπαφή αλλά και το πώς αυτή η αλληλεπίδραση απεικονίζεται στην εμφάνιση μιας περίπτωσης component και στη φύση οποιωνδήποτε επόμενων αλληλεπιδράσεων.

Όπου είναι εφικτό, ένα μοντέλο της συμπεριφοράς της component κατηγορίας πρέπει να περιγραφεί χρησιμοποιώντας ένα UML **statechart**. Η περιγραφή της συμπεριφοράς κατ' αυτό τον τρόπο δεν σημαίνει ότι μια εφαρμογή της component κατηγορίας σε μια πλατφόρμα στόχο πρέπει να περιέχει μία μηχανή καταστάσεων (statemachine) που ορίζεται από το statechart. Αυτό σημαίνει ότι το component θα πρέπει να συμπεριφέρεται σαν περιέχει την statemachine.

Το statechart θα καθορίσει:

- τα εσωτερικές καταστάσεις που μπορεί να υπάρχουν μέσα σε ένα component.
- την απόκριση ενός component σε εξωτερικά γεγονότα και ενέργειες.
- τις αλλαγές στην εμφάνιση ως απόκριση σε γεγονότα, ενέργειες και μεταβατικές καταστάσεις.
- τους όρους κάτω από τους οποίους το component θα δημιουργήσει **component events** και **component errors**.

Όπου είναι απαραίτητο το statechart πρέπει να συνοδευθεί από πρόσθετο περιγραφικό κείμενο.

Εάν η εσωτερική συμπεριφορά μιας κατηγορίας component είναι πάρα πολύ σύνθετη για την πλήρη περιγραφή, τότε είναι αποδεκτή μόνο για την παροχή μιας χαμηλού επιπέδου περιγραφής για το πώς το component συμπεριφέρεται στη διεπαφή του.

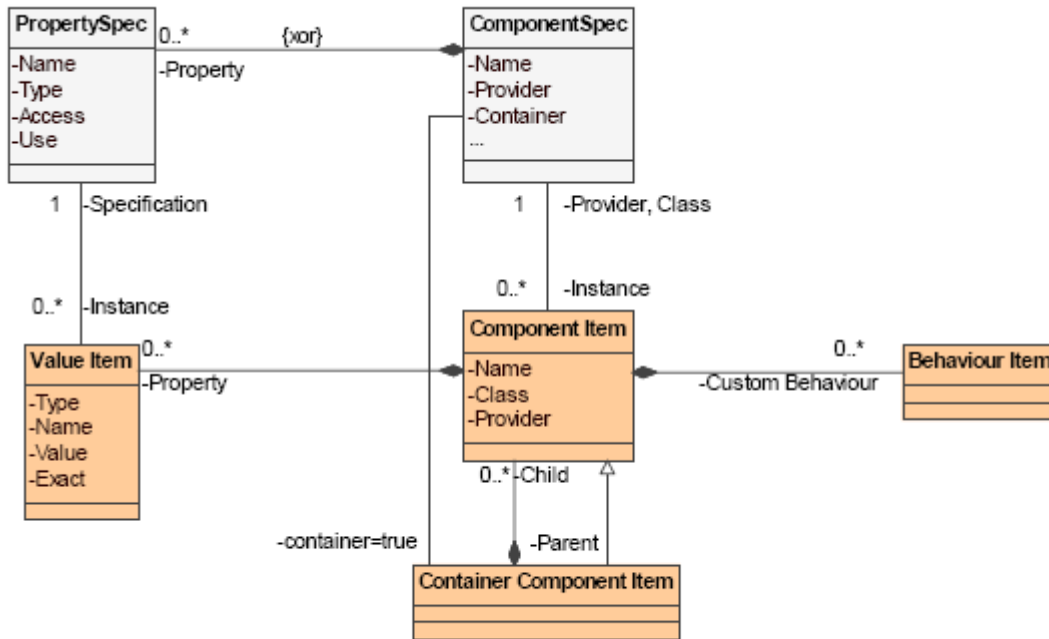
Αυτή θα περιλαμβάνει μια περιγραφή όλων των γεγονότων και των ενεργειών που θα ανταποκριθούν, και όλα τα component γεγονότα που θα δημιουργηθούν και κάτω από ποιες συνθήκες. Σε αυτή την περίπτωση η εσωτερική συμπεριφορά του component δεν πρέπει να διαμορφωθεί ως statechart, ωστόσο πρέπει να περιγραφεί με ικανοποιητικές λεπτομέρειες για να απεικονίσει την προοριζόμενη εμπειρία του χρήστη από την εσω-τερική συμπεριφορά του component.

Παράδειγμα:

Η εσωτερική συμπεριφορά ενός component για την εφαρμογή ενός παιχνιδιού Tetris μπορεί να είναι πάρα πολύ σύνθετη για να μοντελοποιηθεί ως statechart. Σε αυτή την περίπτωση η συμπεριφορά μπορεί να περιγραφεί με την επιλογή κάποιων screen-shots που συνοδεύονται από κάποιο κείμενο.

7.3 Μοντέλο Υποστασιοποίησης δομικών στοιχείων

Η δεύτερη πτυχή του PCF component μοντέλου, το **component instantiation model**, είναι μέρος της PCF. Αντικείμενα από το μοντέλο χρησιμοποιούνται για την δήλωση component περιπτώσεων μέσα σε μια περιγραφή υπηρεσίας PCF. Το μοντέλο component instantiation εφαρμόζεται μέσα στο σχήμα PCF XML. Το **component specification model** συσχετίζεται με το component instantiation model δεδομένου ότι το πρώτο καθορίζει τι είναι αναμενόμενο και έγκυρο μέσα στο τελευταίο. Οι σχέσεις μεταξύ των αντικειμένων μεταξύ των δύο μοντέλων διευκρινίζονται στην εικόνα 34.



Εικόνα 34. Η σχέση μεταξύ των αντικείμενων που ανήκουν στο μοντέλο προδιαγραφών και στο μοντέλο υποστασιοποίησης.

7.3.1 Δομικά στοιχεία

Όλες οι περιπτώσεις δομικών στοιχείων θα πρέπει να είναι διατεταγμένες σύμφωνα με τον τύπο δεδομένων map. Αυτό επιτρέπει στα αντικείμενα PCF να περιλαμβάνονται μέσα στο component. Κάθε περίπτωση δομικού στοιχείου θα πρέπει να έχει ένα **όνομα** που να την προσδιορίζει μοναδικά μέσα στο πεδίο της δήλωσης του.

Όλα components θα πρέπει να έχουν μια **κατηγορία (class)** και προαιρετικά έναν **παροχο (provider)**. Εάν ο πάροχος δεν ορίζεται η προκαθορισμένη τιμή του είναι "dvn.org". Ο **πάροχος** και η **κατηγορία** ενός component συνδέουν μια περίπτωση component με τη **component class προδιαγραφή του**, όπως καθορίζεται από ένα αντικείμενο ComponentSpec. Μια component περίπτωση θα ακολουθεί τους κανόνες που ορίζονται από το αντικείμενο ComponentSpec.

Ένα component μπορεί να περιέχει:

- κανένα ή περισσότερα αντικείμενα **value** για να καθορισμό των αρχικών τιμών των ιδιοτήτων του .
- κανένα ή περισσότερα αντικείμενα **behaviour** για τον καθορισμό των συνήθων επεκτάσεων της έμφυτης συμπεριφοράς του component.
- κανένα ή περισσότερα θυγατρικά αντικείμενα component .

7.3.2 Ιδιότητες

Κάθε ορισμός **ComponentSpec** μιας κατηγορίας component ορίζει ένα σύνολο ιδιοτήτων που επηρεάζουν την εμφάνιση και τη συμπεριφορά κάποιας περιπτώσης component αυτής της κατηγορίας. Μέσα σε μια δήλωση μιας component περίπτωσης οι ιδιότητες καθορίζονται από ένα σύνολο αντικειμένων value που διευκρινίζουν τις αρχικές τιμές της ιδιότητας και ως εκ τούτου την αρχική κατάσταση, αυτής της περίπτωσης component.

- Ένα αντικείμενο value έχει ένα **όνομα** που μπορεί να ταιριάζει με το όνομα ενός αντικειμένου PropertySpec που ορίζεται μέσα στις προδιαγραφές της component κατηγορίας.
- Ένα αντικείμενο value έχει έναν **τύπο δεδομένων**. Αυτό θα είναι ένας από τους τύπους δεδομένων που καθορίζονται από την αρχιτεκτονική PCF στην παράγραφο 6.2.
- Ένα αντικείμενο value έχει μία τιμή .
- Ο τύπος δεδομένων ενός αντικειμένου value με ένα συγκεκριμένο όνομα πρέπει να ταιριάζει με τον τύπο δεδομένων που ορίζεται από το αντικείμενο **PropertySpec** με το ίδιο όνομα στις προδιαγραφές της component κατηγορίας. Εάν ο τύπος δεδομένων **PropertySpec** είναι η απαρίθμηση, τότε το αντικείμενο value θα είναι τύπου **String** και η τιμή του θα πρέπει να είναι μία από αυτές που δίνονται στο αντικείμενο **EnumerationSpec**.

Οι ιδιότητες που χρειάζονται αρχικές τιμές μπορούν να τις λάβουν από μία εκ των τεσσάρων πηγών που υπάρχουν. Η τιμή μπορεί αποκτηθεί χρησιμοποιώντας τον ακόλουθο κανόνα προτεραιότητας. Εάν καμία δεν ορίζεται τότε χρησιμοποιείται η επόμενη επιλογή :

- 1) Μια τιμή δηλώνεται σε μία περίπτωση component.
- 2) Εάν η ιδιότητα ορίζεται σε ένα PropertyGroup αντικείμενο τότε εάν κάποια περίπτωση component στην προγονική ιεραρχία έχει μια δηλωμένη τιμή για την ιδιότητα, αυτή θα χρησιμοποιηθεί. Εάν περισσότεροι από έναν πρόγονο δηλώσουν μια τιμή, τότε θα χρησιμοποιηθεί η τιμή που είναι δηλωμένη στους περισσότερους προγόνους.
- 3) Εάν μια προκαθορισμένη τιμή έχει ορισθεί για την ιδιότητα στο PropertySpec τότε αυτή θα χρησιμοποιηθεί.
- 4) Εάν το **όνομα** της ιδιότητας έχει το πρόθεμα "-focus" ή "-disabled" "- τότε η τιμή της ιδιότητας παίρνει το ίδιο όνομα αλλά χωρίς να χρησιμοποιείται το πρόθεμα.

7.3.2 Διαδοχικές Συυδεδεμένες Ιδιότητες

Είναι δυνατό τα αντικείμενα property τα οποία ορίζονται ως μέλη ενός PropertyGroup να «κατρακυλήσουν» στην ιεραρχία των component μέσα σε μια περιγραφή υπηρεσιών, έτσι ώστε τα θυγατρικά component να μπορούν να κληρονομήσουν τις ιδιότητες που καθορίζονται μέσα στην ιεραρχία των προγόνων τους.

Εάν μια κατηγορία component περιέχει μια ιδιότητα η οποία ορίζεται μέσα σε ένα PropertyGroup, και κάποιο component δεν περιέχει ένα αντικείμενο value που να παρέχει μία τιμή για αυτή την ιδιότητα, τότε η ιδιότητα αυτή θα κληρονομηθεί από το κοντινότερο component στην ιεραρχία των προγόνων παρέχοντας μια τιμή. Αυτή η κληρονομιά πραγματοποιείται στα πλαίσια όπου το component χρησιμοποιείται.

ΠΑΡΑΔΕΙΓΜΑ:

Ένα component **Rectangle** δηλώνεται μέσα σε ένα **StaticELC**. Η περίπτωση **Rectangle** δεν έχει ορισμένες τιμές για τις ιδιότητες **fillcolor** ή **linestyle**. Το container στο οποίο η **Flow** παρουσιάζεται έχει ορισμένη την τιμή red (#FF0000) για την ιδιότητα **fillcolor** αλλά όχι για την ιδιότητα **linestyle**. Σε αυτήν την περίπτωση το **Rectangle** θα κληρονομήσει την ιδιότητα **fillcolor** του container γονέων, και έτσι θα έχει επίσης μία **fillcolor** ιδιότητα red. Δεδομένου ότι είναι αδύνατο να κληρονομήσει την ιδιότητα **linestyle** θα χρησιμοποιήσει την προκαθορισμένη τιμή για την ιδιότητα **linestyle**.

7.3.4 Το δομικό στοιχείο ανοχής υλοποίησης

Η PCF προορίζεται να είναι φορητή σε πολλές πλατφόρμες, κάθε μια με τα δικά της χαρακτηριστικά. Υπό αυτήν τη μορφή γίνεται αποδεκτό ότι σε διαφορετικές πλατφόρμες ορισμένα χαρακτηριστικά γνωρίσματα δεν μπορούν να αποδοθούν ακριβώς με τον ίδιο τρόπο, αλλά θα προσφέρουν πάντα μια ισοδύναμη εμπειρία στο χρήστη. Η επιτρεπόμενη ανοχή μέσα στην οποία τα χαρακτηριστικά γνωρίσματα μπορούν να αποδοθούν καθορίζεται από το στοιχείο **IntendedImplementation**.

Το στοιχείο **IntendedImplementation** καθορίζει την ελάχιστη εφαρμογή που μια πλατφόρμα στόχος πρέπει να παρέχει για αυτή την κατηγορία component. Έχει μια μοναδική ιδιότητα **coreProperties**. Αυτή η ιδιότητα απαριθμεί τις ιδιότητες του component που είναι απαραίτητες για την παροχή μιας ισοδύναμης εμπειρίας στον χρήστη. Οι ιδιότητες στην λίστα **coreProperties** ενός component θα εφαρμοστούν από μία πλατφόρμα στόχο. Όλες οι ιδιότητες που ορίζονται για μια κατηγορία component αλλά δεν περιλαμβάνονται στην λίστα **coreProperties** είναι μη-υποχρεωτικές. Αυτές μπορούν να εφαρμοστούν προαιρετικά σε μια πλατφόρμα στόχο.

7.4 Συμπεριφορά Δομικών Στοιχείων

7.4.1 Επισκόπηση συμπεριφοράς

Το γενικό μοντέλο των δομικών στοιχείων της PCF επιτρέπει την τροποποίηση μιας κατάστασης ενός component κατά την διάρκεια της ζωής του. Το μοντέλο επιτρέπει επίσης την επέκταση ενός καθορισμένου υποσύνολου έμφυτης συμπεριφοράς μιας περίπτωσης component.

Αυτές οι πτυχές της Συνήθους Συμπεριφοράς (Custom behaviour) περιγράφονται χρησιμοποιώντας ένα ή περισσότερα **behaviour items**.

Η Custom behaviour μπορεί να χρησιμοποιηθεί για να

- να επεκτείνει τη συμπεριφορά των επιμέρους components
- να ενώσει τη συμπεριφορά διαφορετικών component μέσα σε ένα container ή σε ένα αντικείμενο **Scene**.

Είναι δυνατό να τροποποιηθεί η κατάσταση μιας περίπτωσης component με δύο τρόπους:

- με την αλλαγή μιας ή περισσότερων τιμών της ιδιότητας.
- με την επίκληση μιας **ενέργειας** .

Και στις δύο περιπτώσεις αυτές οι αλλαγές θα καθορίζονταν χρησιμοποιώντας τη PCF **action language**.

7.4.2 Πρόσβαση στις ιδιότητες των δομικών στοιχείων

Είναι δυνατό να διαβάσουν οι τιμές των component ιδιοτήτων που ορίζονται μέσα στις προδιαγραφές ιδιοτήτων και έχουν τύπο **access** στις ιδιότητες **final**, **readOnly** και **readWrite**. Είναι δυνατό να τροποποιηθούν οι τιμές των component ιδιοτήτων που ορίζονται μέσα στις προδιαγραφές ιδιοτήτων και έχουν τύπο **access** στην ιδιότητα **readWrite**.

Όταν μια περίπτωση component έχει μια ή περισσότερες από τις τιμές των ιδιοτήτων της ενημερωμένες, η υποκείμενη εφαρμογή της component περίπτωσης θα ενημερωθεί για να απεικονίσει αυτήν την αλλαγή. Η ιδιότητες **read** and **write** θα πρέπει να είναι σύγχρονες: όταν μια component περίπτωση έχει τροποποιήσει μια τιμή κάποιας ιδιοτητας και αμέσως μετα διαβαστεί τότε η υποκείμενη component εφαρμογή θα εξασφαλίσει ότι η τιμη που επιστρέφεται από την διαδικασία **read** θα είναι η τιμή που έχει τροποποιηθεί .

Παράδειγμα:

Εάν η ιδιότητα `fillcolor` ενός `TextBox` τροποποιείται από πράσινο (# 00FF00) σε μπλε (# 0000FF) και αμέσως μετά διαβάζεται, η τιμή που επιστρέφεται από την διαδικασία `read` θα είναι επίσης μπλε (# 0000FF).

7.4.3 Αντιμετώπιση γεγονότα *Handled events*

Μερικοί τύποι `component` έχουν έμφυτη συμπεριφορά και μπορούν να ανταποκριθούν σε **γεγονότα**. Τα γεγονότα όπου ένα `component` κάποιου συγκεκριμένου τύπου μπορεί να ανταποκριθεί μπορούν να ορισθούν από το σύνολο των αντικειμένων `HandledEventSpec` μέσα στις προδιαγραφές της `component` κατηγορίας.

Παράδειγμα:

Ένα `component` `Button` θα ανταποκριθεί άμεσα στο πάτημα ενός πλήκτρου από ένα χρήστη χωρίς να χρειάζεται να διευκρινιστεί κάποια πρόσθετη συμπεριφορά. Θα αλλάξει την εμφάνισή της για να δείξει ότι είναι σε μία "ενεργή" κατάσταση.

7.4.4 Αντιμετώπισιμες ενέργειες

Οι ενέργειες των δομικών στοιχείων είναι εντολές που στέλνονται σε συγκεκριμένες περιπτώσεις `Component`, και που κατευθύνουν μια περίπτωση `Component` να εκτελέσει κάποια λειτουργία ή/και να αλλάξει κατάσταση.

Οι ενέργειες που μπορούν να σταλούν σε ένα `Component` ενός συγκεκριμένου τύπου καθορίζονται από το σύνολο αντικειμένων `HandledActionSpec` μέσα στις προδιαγραφές της `component` κατηγορίας.

Οι ενέργειες μπορούν να έχουν καμία ή περισσότερες παραμέτρους για να παρέχουν στο `Component` επιπλέον δεδομένα για την εκτέλεση αυτής της ενέργειας. Αυτές ορίζονται από τα αντικείμενα `parameter` τα οποία βρίσκονται μέσα στα αντικείμενα `HandledActionSpec`.

Η πλατφόρμα στόχος θα εφαρμόσει τα `components` έτσι ώστε να ανταποκριθούν κατάλληλα στις ενέργειες, όπως καθορίζεται από την `behaviour` προδιαγραφή μέσα στην προδιαγραφές της `component` κατηγορίας.

Παράδειγμα:

Το component `ReturnPath` εφαρμόζει ενέργεια `request` για να υποβάλλει μια συναλλαγή σε έναν απομακρυσμένο server. Αυτή η ενέργεια αναμένει ένα `Transaction` component ως παράμετρο.

7.4.5 Παραγόμενα Γεγονότα

Εκτός από την ανταπόκριση σε εξωτερικά ερεθίσματα από την υπηρεσία PCF όπως γεγονότα, αλλαγές ιδιοτήτων και ενεργειών, μερικά components θα είναι σε θέση να παράγουν **component events**. Αυτό τους επιτρέπει να παρέχουν μια ανακοίνωση των εσωτερικών μεταβατικών καταστάσεων σε μια ευρύτερη υπηρεσία PCF στην οποία περιλαμβάνονται.

Τα γεγονότα που ένα **component** κάποιου συγκεκριμένου τύπου μπορεί να παράγει καθορίζονται από το σύνολο των αντικειμένων **GeneratedEventSpec** μέσα στις προδιαγραφές της component κατηγορίας. Η πλατφόρμα στόχος θα εφαρμόσει τα components έτσι ώστε να παράγουν τα κατάλληλα γεγονότα, όπως καθορίζεται από την προδιαγραφή behaviour μέσα στις προδιαγραφές της component κατηγορίας

Παράδειγμα:

Εκτός από την αλλαγή της οπτικής εμφάνισής του, το component **Button**, όταν επιλέγεται, θα δημιουργήσει επίσης ένα γεγονός **OnSelect**. Αυτό το γεγονός μπορεί έπειτα να αντιμετωπιστεί από ένα custom behaviour αντικείμενο μέσα στο αντικείμενο **Scene** για να προσθέσει κάποια πρόσθετη συνήθη συμπεριφορά. Τα component γεγονότα μπορούν να έχουν καμία ή περισσότερες παραμέτρους για να παρέχουν πρόσθετα δεδομένα για ένα γεγονός.

7.4.6 Παραγόμενα Λάθη

Όπως και στην παραγωγή των γεγονότων κατά τη διάρκεια της κανονικής τους ζωής, μερικοί τύποι component μπορούν επίσης να παράγουν λάθη για την παροχή ανακοίνωσης ενός εσωτερικού προβλήματος. Τα λάθη που ένα component μιας συγκεκριμένης κατηγορίας μπορεί να παράγει καθορίζονται από το σύνολο των αντικειμένων **GeneratedErrorSpec** μέσα στην προδιαγραφή της component κατηγορίας.

Η πλατφόρμα στόχος θα εφαρμόσει τα components έτσι ώστε παράγουν τα κατάλληλα, λάθη όπως καθορίζεται από τη behaviour Specification μέσα στην προδιαγραφή της component κατηγορίας.

Παράδειγμα:

Ένα **Image** component θα παράγει ένα λάθος **OnInvalidMediaType** εάν ο τύπος δεν ταιριάζει με τον τύπο της εικόνας που υποστηρίζεται .

7.4.7 Πεδίο των δομικών στοιχείων

Το πεδίο μέσα στο οποίο τα components είναι τροποποιήσιμα από τα αντικείμενα behaviour καθορίζεται αυστηρά από την PCF.

7.5 Προσδιορισμός των κατηγοριών των δομικών στοιχείων της PCF

7.5.1 Επισκόπηση

Τα component στην PCF μπορούν ευρέως να ομαδοποιηθούν στις ακόλουθες κατηγορίες:

- **Visual components:** Αυτά τα component συμβάλλουν στις οπτικές πτυχές της εμπειρίας του θεατή .
- **Non-visual components:** Αυτά τα component δεν έχουν καμία οπτική εμφάνιση αλλά παρέχουν κάποια πτυχή της λειτουργικότητας της υπηρεσίας.
- **Container components:** Αυτά τα component μπορούν να περιέχουν διάφορα άλλα component.

7.5.2 Οπτικά δομικά στοιχεία

Ο παρακάτω πίνακας περιέχει όλα τα Visual components που είναι διαθέσιμα μέσα στην PCF

<i>Component Type</i>	<i>Description</i>
AxisLine	Σχεδιάζει μια οριζόντια ή κάθετη γραμμή.
Background	Σχεδιάζει μια εικόνα ή ένα στερεό χρώμα ως υπόβαθρο
Button	Εφαρμόζει την εμφάνιση και την συμπεριφορά ενός πλήκτρου
Clock	Εφαρμόζει την εμφάνιση και την συμπεριφορά ενός ρολογιού
ConnectStatusImage	Καθορίζει μια θέση αναφοράς μέσα στην οθόνη όπου συγκεκριμένα γραφικά της πλατφόρμας μπορεί να αποδοθούν για την εμφάνιση της κατάστασης σύνδεσης της διαδρομής επιστροφής του component
Ellipse	Σχεδιάζει μια έλλειψη.
HintTextBox	Σχεδιάζει ένα hint ακρυπογράφητο κείμενο μέσα σε μια καθορισμένη ορθογώνια περιοχή ως απόκριση σε focus γεγονότα πάνω σε visible components.
Image	Εμφανίζει μια εικόνα
ImageAnimated	Εμφανίζει ένα σύνολο εικόνων σε σειρά για την δημιουργία κάποιου animation
ImageScalable	Εμφανίζει μια εικόνα που μπορεί να μεταβληθεί η κλίμακα της ανάλογα με το διαθέσιμο μέγεθος
Line	Σχεδιάζει μια γραμμή μεταξύ δύο αυθαίρετων σημείων.
Menu	Εφαρμόζει την εμφάνιση και τη συμπεριφορά ενός menu.
NumericNavigator	Επιτρέπει σε έναν χρήστη να εισάγει έναν αριθμό και στην συνέχεια τον πλοηγεί έπειτα σε ένα αντικείμενο scene που συνδέεται με αυτό τον αριθμό.
PickList	Σχεδιάζει ένα αναδυόμενο παράθυρο το οποίο περιέχει μία λίστα textual αντικειμένων ,και επιτρέπει στο χρήστη να επιλέξει ένα αντικείμενο .
Pixel	Σχεδιάζει ένα pixel

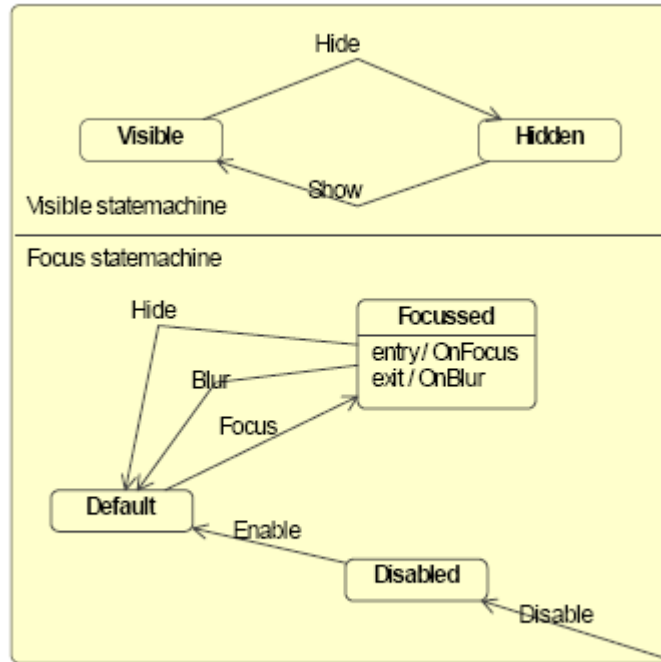
Polygon	Σχεδιάζει ένα πολύγωνο βασισμένο σε ένα σύνολο αυθαιρέτων σημείων.
RadioButtonGroup	Ομαδοποιεί ένα σύνολο πλήκτρων σε μια αλληλοαποκλειόμενη ομάδα.
Rectangle	Σχεδιάζει ένα ορθογώνιο.
SpinControl	Εφαρμόζει την εμφάνιση και τη συμπεριφορά ενός ελέγχου περιστροφής.
TextBox	Σχεδιάζει ένα text box.
TextInput	Εφαρμόζει την εμφάνιση και τη συμπεριφορά ενός πεδίου text input.
Ticker	Εφαρμόζει την εμφάνιση και τη συμπεριφορά ενός marquee ticker.
Subtitles	Ελέγχει την εμφάνιση των υποτίτλων που ορίζονται ως μια στοιχειώδης ροή ενός Stream component
Video	Ελέγχει την εμφάνιση ενός video που ορίζεται ως μια στοιχειώδης ροή ενός Stream component

Πίνακας 21. Οπτικά δομικά στοιχεία

Όλα τα visual components μπορούν αν είναι είτε σε κατάσταση "Visible" είτε σε κατάσταση "Hidden" .

Εξ ορισμού, όλα τα visual components μπορούν να εστιάσουν, εντούτοις αυτό μπορεί να αγνοηθεί είτε στην προδιαγραφή της component κατηγορίας είτε στη instance δήλωση θέτωντας την ιδιότητα focusable false

Τα Components που είναι focusable μπορούν να εφαρμόσουν το statechart που καθορίζεται στην εικόνα 35.



Εικόνα 35 . Συμπεριφορά των οπτικών δομικών στοιχείων

Για ένα component το οποίο δεν είναι focusable η Focus statemachine στην εικόνα 35 περιορίζεται μόνο στην default κατάσταση .

Όταν ένα focusable component βρίσκεται στην κατάσταση "Disabled" εξακολουθεί να έχει μία οπτική εμφάνιση , εάν ορίζεται κάποια, αλλά δεν έχει την δυνατότητα εστίασης .Όταν ένα focusable component βρίσκεται στην κατάσταση " Default" θα έχει να έχει οπτική εμφάνιση και θα είναι focusable .

Η μόνη εξάρτηση μεταξύ των δύο statemachines είναι ότι όταν ένα component βρίσκεται στην κατάσταση "hidden" δεν έχει την δυνατότητα εστίασης .

Ένα γεγονός "Show" θα συμβεί όταν η ιδιότητα **visible** τεθεί true . Ένα γεγονός "Hide" θα συμβεί όταν η ιδιότητα **visible** τεθεί false. Ένα γεγονός "Enable" θα συμβεί όταν η ιδιότητα **enabled** τεθεί true . Ένα γεγονός "Disable" θα συμβεί όταν η ιδιότητα **enabled** τεθεί false. Ένα γεγονός "Focus" θα συμβεί όταν δοθεί στο component εστίαση . Ένα γεγονός "Blur" θα συμβεί όταν το component χάσει την εστίαση .

Μόνο ένα visible component με μια ιδιότητα **focusable** της οποίας η τιμή έχει τεθεί true έχει την δυνατότητα εστίασης και το σημείο στο ποίο θα εστιάσει θα καθοριστεί από την ιδιότητα **tabindex** .

Μερικές κατηγορίες component μπορούν να οριστούν ώστε να αλλάζουν αυτόματα την εμφάνισή τους ανάλογα με την κατάσταση την οποία βρίσκονται χωρίς να χρειάζεται κάποια ενέργεια περιγραφής .

Προκειμένου να επιτρέπεται στον συντάκτη μιας περιγραφής υπηρεσίας να ελέγχει τέτοιες αλλαγές στην εμφάνιση, μια component προδιαγραφή μπορεί να συνδέσει τις ιδιότητες με κάθε component κατάσταση. Αυτή η ένωση επιτυγχάνεται με τη χρήση των τυποποιημένων προθεμάτων στο όνομα της κάθε ιδιότητας :

- Για να συνδεθεί μια ιδιότητα με την κατάσταση "Default" δεν θα χρησιμοποιηθεί κανένα πρόθεμα.
- Για να συνδεθεί μια ιδιότητα με την κατάσταση "Focused" θα χρησιμοποιηθεί το πρόθεμα "-focus".
- Για να συνδεθεί μια ιδιότητα με την κατάσταση "Disabled" θα χρησιμοποιηθεί το πρόθεμα "-disabled" .

7.5.3 Μη οπτικά δομικά στοιχεία

7.5.3.1 Λειτουργικά δομικά στοιχεία

Ο πίνακας 24 απαριθμεί τα λειτουργικά μη-οπτικά δομικά στοιχεία που είναι διαθέσιμα στην PCF.

Component Type	Description
Audio	ελέγχει την παρουσίαση ενός ήχου, που ορίζεται ως μία στοιχειώδης ροή σε ένα ρεύμα Stream component.
CurrentTime	χρησιμοποιείται για την λήψη της τρέχουσας ώρας
Indicate	παρέχει έναν απλό μηχανισμό ψηφοφορίας χρησιμοποιώντας ένα return path .
Random	χρησιμοποιείται για την λήψη ενός τυχαίου αριθμού.
ReturnPath	παρέχει πρόσβαση σε μια διαδρομή επιστροφής.
SecureReturnPath	παρέχει την πρόσβαση σε έναν ασφαλή μηχανισμό διαδρομής επιστροφής .
Stream	ελέγχει τη σύνδεση σε μια σύνθεση στοιχειωδών πολυμεσικών ροών (βίντεο, ήχος, υπότιτλοι κ.λπ .) που παρουσιάζονται συγχρονισμένες .
StreamEvent	ελέγχει μια πηγή stream γεγονότων από την ροή στοιχειωδών δεδομένων από ένα Stream component.
Timer	χρησιμοποιείται για την δημιουργία μιας ανακοίνωσης χρόνου σε μία συγκεκριμένη στιγμή στο μέλλον.
Transaction	ελέγχει την ανταλλαγή δεδομένων πάνω σε μία διαδρομή επιστροφής .

Πίνακας 22. Μη – οπτικά λειτουργικά δομικά στοιχεία .

7.5.3.2 Τα δομικά στοιχεία μεταβλητή και cookie

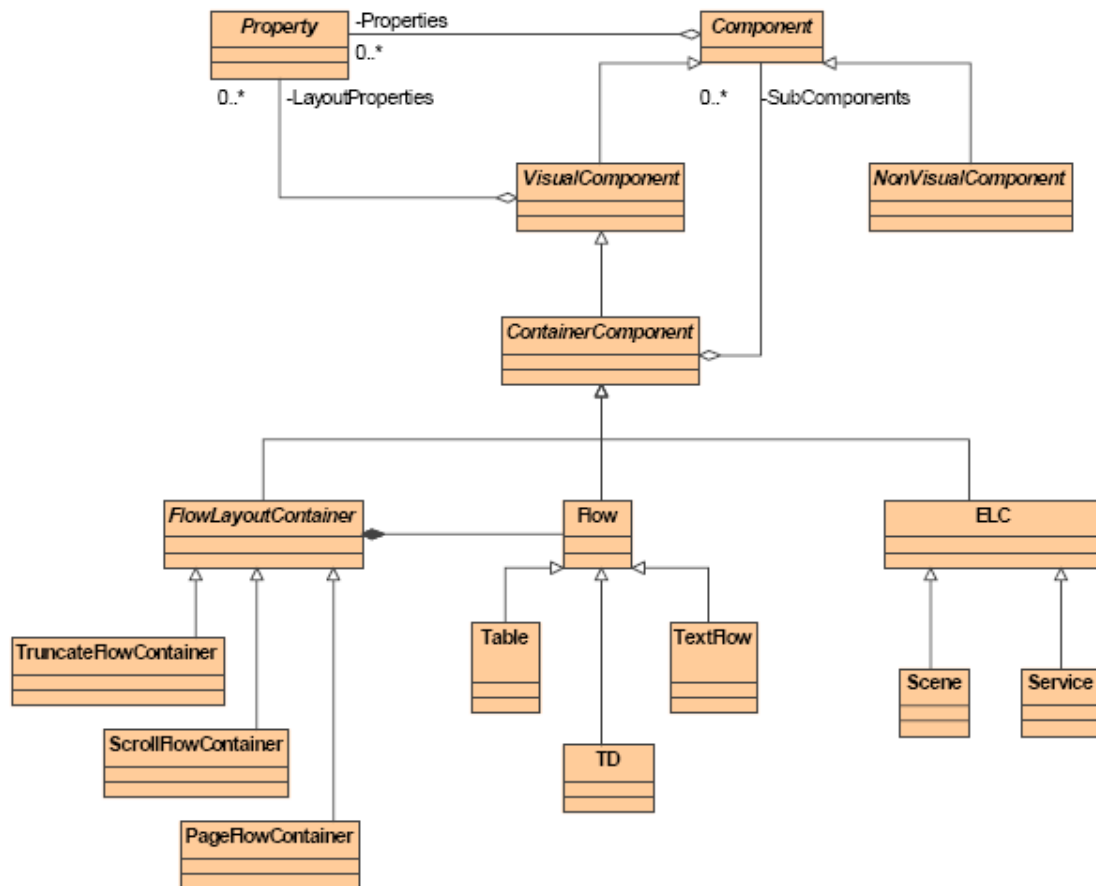
Ο πίνακας 23 απαριθμεί τα δομικά στοιχεία variable και cookie που είναι διαθέσιμα στην PCF.

Component Type Description	Component Type Description
BooleanVar	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής boolean τιμής κατά τη διάρκεια ζωής της ενθυλάκωσης ενός Container component.
DateTimeVar	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής παρουσίασης των τιμών date και time κατά τη διάρκεια ζωής της ενθυλάκωσης ενός Container component.
IntegerVar	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής integer τιμής κατά τη διάρκεια ζωής της ενθυλάκωσης ενός Container component.
StringVar	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής string τιμής κατά τη διάρκεια ζωής της ενθυλάκωσης ενός Container component.
BooleanCookie	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής boolean τιμής έτσι ώστε να μπορεί να διατηρείται πέρα από τη διάρκεια συνόδου μιας υπηρεσίας
DateTimeCookie	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής παρουσίασης των τιμών date και time έτσι ώστε μπορούν να διατηρούνται πέρα από τη διάρκεια συνόδου μιας υπηρεσίας
IntegerCookie	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής integer τιμής έτσι ώστε να μπορεί να διατηρείται πέρα από τη διάρκεια συνόδου μιας υπηρεσίας
StringCookie	χρησιμοποιείται για την αποθήκευση και την ανάκτηση μιας μοναδικής string τιμής έτσι ώστε μπορεί να διατηρείται πέρα από τη διάρκεια συνόδου μιας υπηρεσίας

Πίνακας 23. Τα δομικά στοιχεία variable και cookie .

7.5.4 Το δομικό στοιχείο Container

Το σχήμα παρουσιάζει την εννοιολογική ιεραρχία ταξινόμησης των δομικών στοιχείων μέσα στο τυποποιημένο σύνολο των PCF component :



Εικόνα 36 . Μοντέλο αντικειμένων των δομικών στοιχείων.

7.6 Συνήθη δομικά στοιχεία

Εκτός από τους component τύπους που ορίζονται ως πρότυποι μέσα στις προδιαγραφές της PCF, είναι δυνατό να δημιουργηθούν custom components για να εφαρμογή ειδικών λειτουργιών ή για την εκμετάλλευση συγκεκριμένων χαρακτηριστικών γνωρισμάτων τα οποία είναι διαθέσιμα σε μια πλατφόρμα στόχο.

Δεδομένου ότι η PCF χρησιμοποιεί μια γενική διεπαφή δομικών στοιχείων για τα τυποποιημένα component που έχει, οι περιπτώσεις custom components μπορούν να ορισθούν μέσα σε μια υπηρεσία PCF με τον ίδιο ακριβώς τρόπο.

Παράδειγμα 1:

Τα Foo Components έχουν δημιουργήσει ένα διαδραστικό παιχνίδι και θα το ονόμασουν ως εξής :

```
provider="foocomponents.com" name="Game"
```

Παράδειγμα 2:

Η εταιρία Big Broadcasting έχει δημιουργήσει 2 κουίζ και θα τα ονομάσει ως εξής :

```
provider="bigbroadcastingcompany.com/gameshows/strongestchain " name="quiz"  
provider="bigbroadcastingcompany.com/gameshows/testthecontinent" name="quiz"
```

7.7 Τα δομικά στοιχεία Schema

Το γενικευμένο component μοντέλο ισχύει για όλες τις κατηγορίες των δομικών στοιχείων. Έτσι ώστε να είναι δυνατό να δηλωθεί μία περίπτωση component οποιασδήποτε κατηγορίας χρησιμοποιώντας το στοιχείο **Component** του γενικευμένου component μοντέλου και καθορίζοντας ιδιότητες αυτής της περίπτωσης component με τα διάφορα **ValueItems**.

Εκτός από αυτήν την μέθοδο δήλωσης θα είναι δυνατό να ορισθούν τυποποιημένα PCF components χρησιμοποιώντας στοιχεία κάποιας συγκεκριμένης κατηγορίας .

Αυτό θα επιτρέπει σε έναν συντάκτη PCF να γράφει λιγότερο φλύαρα, αλλά λειτουργικά ισοδύναμα PCF. Τα components που μπορούν να χρησιμοποιηθούν με αυτό τον τρόπο αναφέρονται συλλογικά ως **schema components**..

Τα στοιχεία τα οποία χρησιμοποιούνται για την δήλωση κάθε τύπου στο **schema components** συμπεριλαμβάνονται μέσα στο σχήμα της PCF. Το στοιχείο για κάθε schema component θα είναι το ίδιο με το όνομα της component κατηγορίας .

Παράδειγμα:

Ένα **Rectangle** component μπορεί να δηλωθεί γενικά χρησιμοποιώντας το παρακάτω PCF πλαίσιο:

```
<Component class="Rectangle">  
  <Color name="fillcolor" value="#0000FF"/>  
  <Color name="linecolor" value="#FFFFFF"/>  
  <Position name="origin" value="100 100"/>  
  <Size name="size" value="50 30"/>  
</Component>
```

Εναλλακτικά η λειτουργικά ισοδύναμη δήλωση είναι η εξής :

```
<Rectangle>  
  <Color name="fillcolor" value="#0000FF"/>  
  <Color name="linecolor" value="#FFFFFF"/>  
  <Position name="origin" value="100 100"/>  
  <Size name="size" value="50 30"/>  
</Rectangle>
```

8 Προδιαγραφές Σχεδιαγραμμάτων

8.1 Εισαγωγή

Οι προδιαγραφές σχεδιαγραμμάτων της PCF καθορίζουν το γενικό προσδιορισμό θέσης και την εμφάνιση των οπτικών δομικών στοιχείων της PCF τα οποία περιλαμβάνονται σε ένα σχεδιάγραμμα (layout) μιας υπηρεσίας PCF.

Ένα σχεδιάγραμμα με οπτικά δομικά στοιχεία της PCF μπορεί να παρουσιαστεί με δύο τρόπους:

- explicit layout
- flow layout.

Το explicit layout υποστηρίζει τον ακριβή προσδιορισμό της θέσης pixel των graphical και textual αντικειμένων στην οθόνη μέσα σε ένα explicit layout container.

Το Flow layout υποστηρίζει την εμφάνιση του περιεχομένου που ρέει μέσα σε μια καθορισμένη ορθογώνια περιοχή σύμφωνα με ένα αυτοματοποιημένο αλγόριθμο. Υπάρχουν τρία διαφορετικά είδη **Flow**:

- Flow: Το περιεχόμενο ροής μπορεί να περιλαμβάνει κείμενο, ευθύγραμμες γραφικές παραστάσεις και όλα τα οπτικά δομικά στοιχεία. Ένα Flow component μπορεί να περιέχει θυγατρικά visual components. Ένα Flow component μπορεί ακόμα να περιέχει MarkedUpText περιεχόμενο .
- TextFlow: Το TextFlow είναι ένα υποσύνολο του αντικειμένου Flow όπου υποστηρίζεται μόνο το περιεχόμενο SimpleMarkedUpText. Ένα TextFlow δεν περιέχει θυγατρικά στοιχεία .
- Table: Οι πίνακες είναι ένα υποσύνολο του αντικειμένου Flow όπου το περιεχόμενο ροής σχεδιάζεται σε μια δισδιάστατη συνοπτική δομή .

Το περιεχόμενο ροής παρουσιάζεται σε ένα από τα τρία είδη του Flow container:

- **TFC**(Truncate Flow Container): Μια ορθογώνια περιοχή στην οποία μπορεί να εμφανιστεί μια ροή περιεχομένου. Το TFC μπορεί να επεκταθεί μέσα σε καθορισμένα όρια, για την προσαρμογή του Flow. Το περιεχόμενο το οποίο δεν μπορεί να προσαρμοστεί μέσα στο TFC θα περικοπεί.
- **SFC**(Scroll Flow Container): Μια ορθογώνια περιοχή στην οποία μια ροή περιεχομένου μπορεί να εμφανιστεί. Το SFC παρουσιάζει έναν scrollbar για

να επιτρέπει στο θεατή να κυλίζει πάνω-κάτω το περιεχόμενο το οποίο δεν μπορεί να προσαρμοστεί στο SFC.

- **PFC**(Page Flow Container): Μια ορθογώνια περιοχή στην οποία μπορεί να εμφανιστεί μια ροή περιεχομένου. Η PFC επιτρέπει τον έλεγχο αριστερης και δεξιάς πλοήγησης ώστε να μπορεί θεατής να δει το περιεχόμενο το οποίο δεν μπορεί να προσαρμοστεί στην PFC.

8.2 Αναλυτικά Σχεδιαγράμματα

8.2.1 Εισαγωγή

Ένα PCF **explicit layout container** component είναι ένα δομικό στοιχείο container το οποίο επιτρέπει στα θυγατρικά του δομικά στοιχεία να έχουν ακριβή θέση και μέγεθος χρησιμοποιώντας τις τιμές συντεταγμένων και μεγέθους που έχουν ρητά οριστεί. Ένα explicit layout container είναι ένα πλαίσιο το οποίο χρησιμοποιείται για την ομαδοποίηση των θυγατρικών components. Τα θυγατρικά components τοποθετούνται με θέση αναφοράς το αρχικό explicit container. Ένα explicit layout container δεν έχει δική του οπτική εμφάνιση ή μέγεθος. Έχει μια ιδιότητα θέσης **origin** που ορίζει την αρχή στο σύστημα συντεταγμένων όπου χρησιμοποιείται ως θέση αναφοράς για την τοποθέτηση των θυγατρικών components.

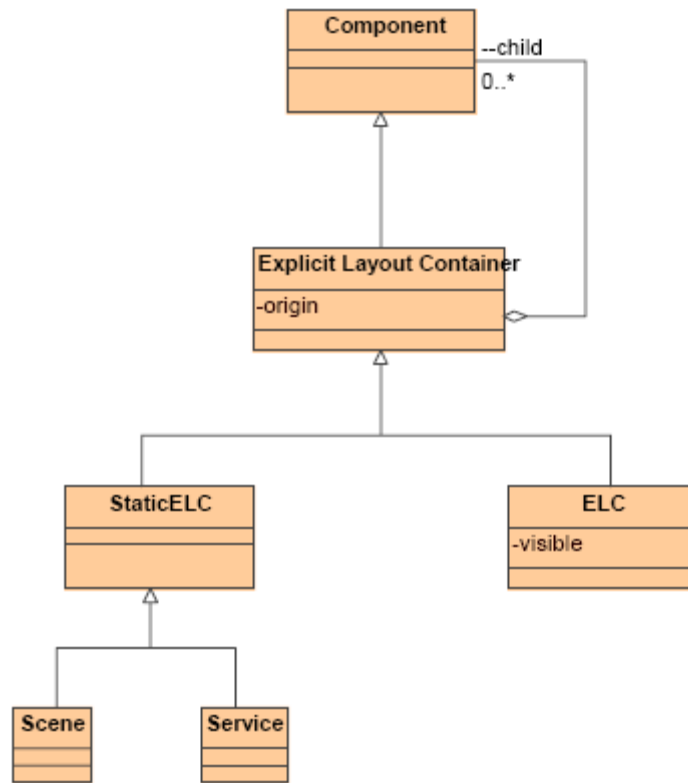
Τα δομικά στοιχεία τα οποία βρίσκονται μέσα σε ένα explicit layout container δεν επηρεάζουν το ένα την οριζόντια και κάθετη συντεταγμένη του άλλου.

8.2.2 Στοιχεία και χαρακτηριστικά του Explicit layout container

Υπάρχουν δύο διαθέσιμοι τύποι του explicit layout container: ο στατικός (**StaticELC**) και ο δυναμικός (**ELC**).

Το στατικό container επιτρέπει στα θυγατρικά του δομικά στοιχεία να σχεδιαστούν στην έναρξη, αλλά στην συνέχεια το container δεν μπορεί να επανατοποθετηθεί ή να είναι κρυμμένο. Αντίθετα, το δυναμικό container μπορεί να επανατοποθετηθεί, να εμφανιστεί ή να είναι κρυμμένο όπως οποιοδήποτε άλλο οπτικό δομικό στοιχείο.

Η σχέση μεταξύ αυτών και άλλων σχετικών component παρουσιάζεται στην εικόνα 37.

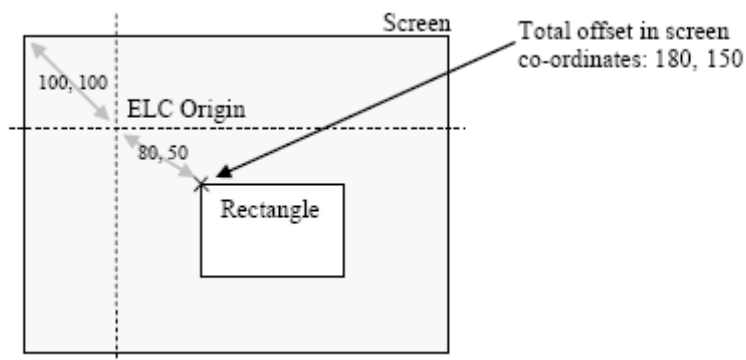


Εικόνα 37. Ιεραρχία του δομικού στοιχείου explicit layout

Παράδειγμα 1:

Ένα explicit layout container έχει μία origin θέση 100,100. Το container έχει ένα μοναδικό θυγατρικό component το **Rectangle** το οποίο έχει μία ιδιότητα origin με τιμή 80,50. Η θέση του **Rectangle** στην οθόνη θα είναι αναφορική με την τιμή origin του layout container, επομένως θα βρίσκεται στις συντεταγμένες της οθόνης 180,150.

Αυτό είναι διευκρινίζεται στο σχήμα



Εικόνα 38. Σχετικός προσδιορισμός θέσης των θυγατρικών δομικών στοιχείων ενός explicit layout.

Ένα StaticELC μπορεί μόνο να αρχικοποιηθεί. Μόλις αρχικοποιηθεί δεν μπορεί να μετακινηθεί ή να κρυφτεί. Κάθε ανεξάρτητο θυγατρικό component μπορεί να εμφανιστεί να κρυφτεί και να μετακινηθεί.

Ένα αντικείμενο Scene είναι ένα StaticELC. Όλα τα scenes έχουν μία σταθερή τιμή origin στο σύστημα συντεταγμένων της οθόνης την (0,0) .

Ένα δυναμικό ELC έχει όλες τις ιδιότητες ενός StaticELC. Επιπλέον είναι δυνατό το container να εμφανίζεται και να κρύβεται να μετακινείται στην origin τιμή του και να επανατοποθετείται στο display stack. Όταν ένα δυναμικό explicit layout container εμφανίζεται κρύβεται ή μετακινείται όλα τα θυγατρικά του component μπορούν επίσης να εμφανιστούν ,να κρυφτούν ή να μετακινηθούν ανάλογα.

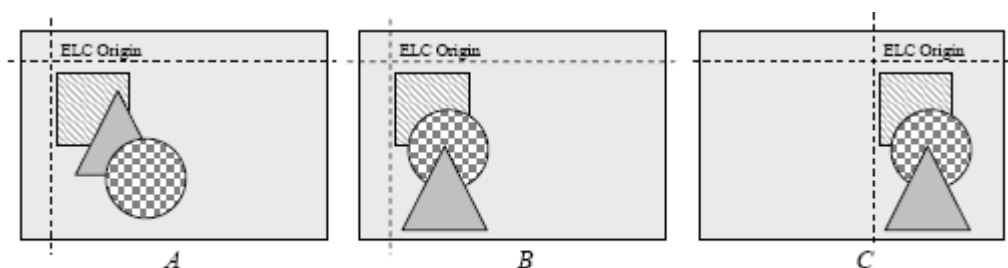
Είναι εφικτό ένα explicit layout container να περιέχει περαιτέρω explicit layout containers.

Σε κάθε περίπτωση οι origin συντεταγμένες των θυγατρικών containers καθορίζουν το σημείο μηδέν για τα θυγατρικά τους στοιχεία. Εντούτοις, δεν είναι δυνατό για ένα StaticELC να τοποθετηθεί μέσα σε ένα ELC. Αυτό συμβαίνει επειδή το στατικό container δεν μπορεί να μετακινηθεί, και έτσι είναι αδύνατο να μετακινηθεί εάν το δυναμικό γονικό container μετακινηθεί.

Τα Explicit layout containers μπορεί να περιέχουν Flow layout containers αλλά δεν μπορούν να περιέχονται σε Flow layout containers. Εάν ένα ή περισσότερα components μέσα σε ένα ELC κινούνται ανεξάρτητα, είτε στους X και Y άξονες είτε πάνω-κάτω στον display stack και στην συνέχεια μετακινείται και το ELC, τότε τα θυγατρικά components θα διατηρήσουν τις νέες σχετικές τους θέσεις μέσα στο ELC.

Παράδειγμα 2:

Το σχήμα παρουσιάζει ένα ELC το οποίο περιέχει τρία θυγατρικά components: ένα **Rectangle**, ένα **Ellipse** και ένα τρίγωνο(το οποίο μπορεί να δημιουργηθεί χρησιμοποιώντας το **Polygon** component). Αρχικά διατάσσονται, όπως διευκρινίζονται στο σχήμα A, από πίσω προς τα εμπρός ορθογώνιο-τρίγωνο - έλλειψη. Στη συνέχεια, το component τρίγωνο εμφανίζεται μπροστά στο σωρό. Το τρίγωνο και η έλλειψη μετακινούνται. Αυτό φαίνεται στο σχήμα B. Τέλος, το ELC κινείται προς τα δεξιά. Όπως φαίνεται στο σχήμα C, οι τροποποιημένες σχετικές χωρικές σχέσεις μεταξύ των θυγατρικών components διατηρούνται μετά την μετακίνηση του γονέα.

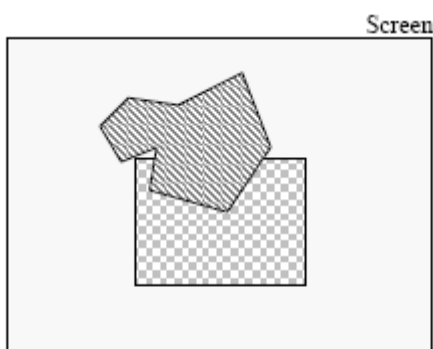


Εικόνα 39. Μετακίνηση θυγατρικών δομικών στοιχείων μέσα σε ένα ELC.

Κατά τη αρχικοποίηση τα δομικά στοιχεία θα τοποθετηθούν στο z-επίπεδο με την σειρά με την οποία έχουν δηλωθεί, τα δομικά στοιχεία που έχουν δηλωθεί πρώτα τοποθετούνται πίσω. Εάν δύο δομικά στοιχεία καταλαμβάνουν το ίδιο διάστημα στην οθόνη, το δομικό στοιχείο το οποίο έχει δηλωθεί αργότερα, και έτσι βρίσκεται υψηλότερα στον σωρό εμφάνισης, θα κρύψει το δομικό στοιχείο το οποίο θα τοποθετηθεί χαμηλότερα στο σωρό εμφάνισης

Παράδειγμα 3 :

Ένα explicit layout container έχει δύο θυγατρικά components, ένα **Rectangle** και ένα **Polygon**. Το **Rectangle** δηλώνεται πριν από το **Polygon** στην περιγραφή PCF. Το **Polygon** επομένως θα είναι πλησιέστερα στο z-επίπεδο του σωρού εμφάνισης .

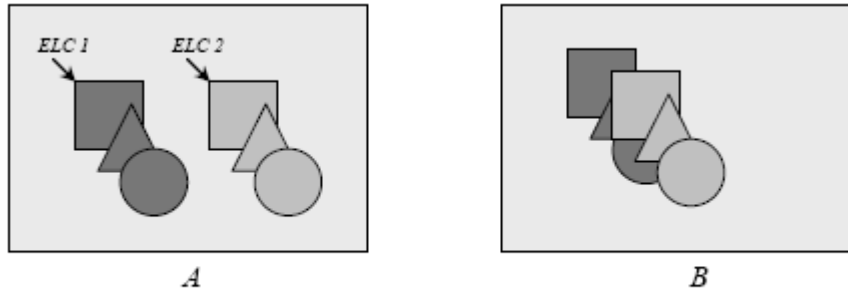


Εικόνα 40 . Σωρός εμφάνισης μετά την τοποθέτηση των θυγατρικών δομικών στοιχείων explicit layout.

Τα θυγατρικά components μέσα σε ένα explicit layout container θα καταλάβουν την ίδια «ζώνη» στο σωρό εμφάνισης. Εάν δύο explicit layout containers είναι θυγατρικά containers ενός άλλου explicit layout container, τότε αυτά τα θυγατρικά containers θα καταλάβουν την δικιά τους ξεχωριστή «ζώνη» στο σωρό εμφάνισης. Όλα τα θυγατρικά στοιχεία του layout container που βρίσκονται πίσω θα τοποθετηθούν κάτω από όλα τα θυγατρικά στοιχεία του μπροστινού layout container .

Παράδειγμα 4 :

Στο διάγραμμα A υπάρχουν δύο Explicit Layout Containers, τα ELC 1 και ELC 2. Και τα δύο έχουν τρία θυγατρικά components τα οποία βρίσκονται σε διαφορετικές θέσεις στο z-επίπεδο. Το ELC 2 βρίσκεται μπροστά από το ELC 1 στο z-επίπεδο. Εάν τα containers επανατοποθετηθούν έτσι ώστε να επικαλύπτονται όπως στο διάγραμμα B, το τελευταίο δομικό στοιχείο του ELC 2 είναι ακόμα μπροστά από το πιο μπροστινό δομικό στοιχείο του ELC 1.



Εικόνα 41. Επικαλυπτόμενα *Explicit Layout Containers*.

8.3 Διαγράμματα ροής

8.3.1 Εισαγωγή

Τα διαγράμματα ροής της PCF παρέχουν έναν μηχανισμό με τον οποίο δομικά στοιχεία και περιεχόμενο που δεν απαιτούν την ακρίβεια διάταξης ενός explicit layout container μπορούν να "κυλίσουν" σε μια ορθογώνια περιοχή και να τοποθετηθούν αναφορικά το ένα με το άλλο σύμφωνα με τον αλγόριθμο Flow layout.

Ένα δομικό στοιχείο **Flow** διατάσσει τα θυγατρικά του στοιχεία με τον ίδιο τρόπο όπως στην HTML. Τα σύνολα χαρακτήρων απαιτούν ροή από αριστερά προς δεξιά και δεξιά προς τα αριστερά και αυτά είναι διαχειρίσιμα σε μία βάση στοιχείο-προς-στοιχείο. Μέσα σε αυτό το βασικό πλαίσιο, ο ακριβέστερος έλεγχος τοποθέτησης επιτυγχάνεται με την χρήση της ευθυγράμμισης, κενών διαστημάτων και διαχωριστικών μπλοκ από ανεξάρτητα στοιχεία.

8.3.2 Τα στοιχεία σχεδιαγράμματα ροής

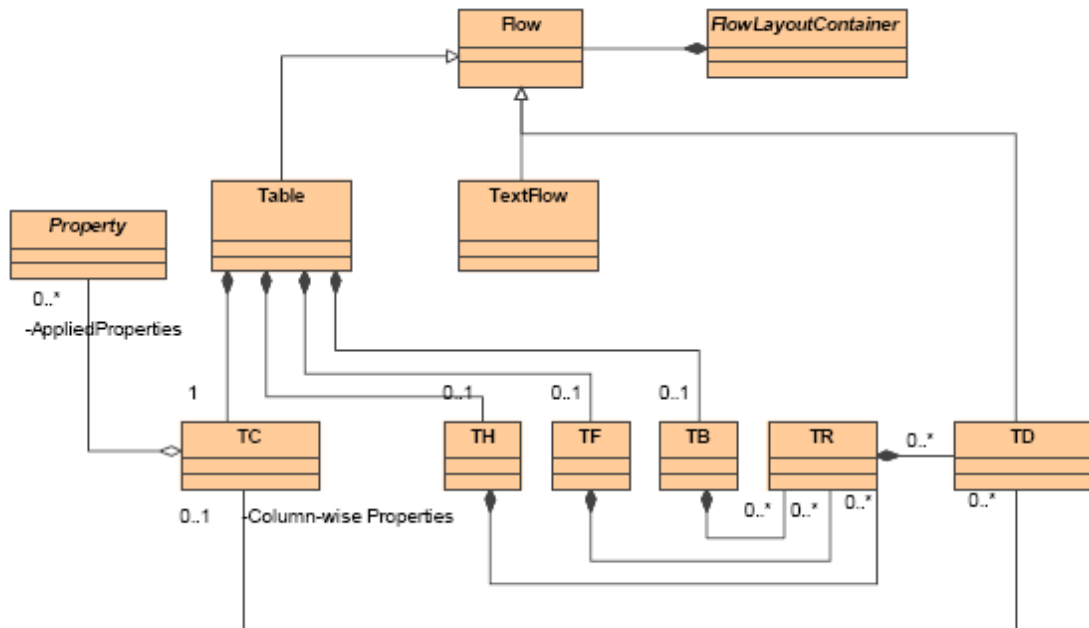
Μία PCF **Scene** περιγραφή διαμορφώνει ένα περιγραφικό δέντρο .

Ένα **Flow** είναι ένας κόμβος σε αυτό το περιγραφικό δέντρο .

Ένας πίνακας είναι μια ειδίκευση ενός Flow, και περιγράφεται στην παράγραφο 8.5.

Ένα **Flow** θα ενσωματώσει τα οπτικά δομικά στοιχεία και το `marked up` κείμενο κάτω από το την ιεραρχία του δέντρου περιγραφής.

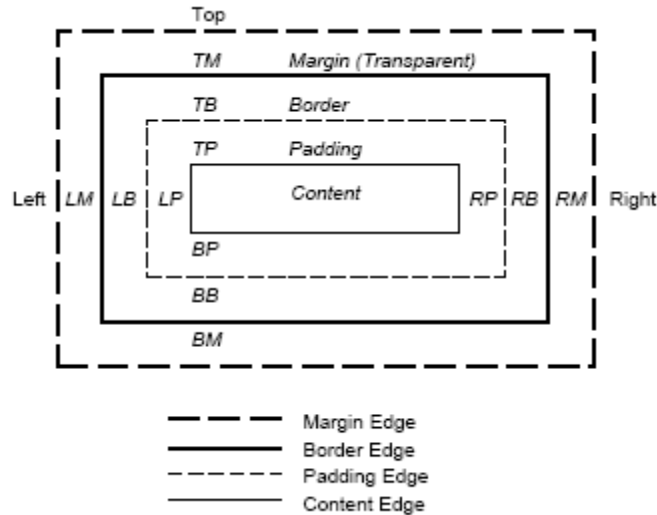
Αυτά ονομάζονται **PCF flow layout elements** και οι σχέσεις τους διευκρινίζονται στην εικόνα 42. Αυτά τα στοιχεία διαμορφώνουν ένα υποσύνολο κόμβων μέσα στο δέντρο και αυτό το υποσύνολο μπορεί να θεωρηθεί ισοδύναμο με το δέντρο εγγράφων που περιγράφεται στις CSS2 προδιαγραφές.



Εικόνα 42. Ιεραρχία των δομικών στοιχείων διαγράμματα ροής.

8.3.3 Το μοντέλο πλαισίων διαγραμμάτων ροής

Μία διάταξη ροής PCF θα διατάξει τα θυγατρικά δομικά στοιχεία και το περιεχόμενο χρησιμοποιώντας ένα εννοιολογικό **box model**. Σε αυτό το μοντέλο κάθε αναγνωρίσιμο θυγατρικό στοιχείο αντιμετωπίζεται σαν να περιλαμβάνεται μέσα στο **box model**. Κάθε εννοιολογικό κουτί έχει τις ιδιότητες που εμφανίζονται στη εικόνα 43.



Εικόνα 43. Ιδιότητες του layout box .

Ένα layout box για ένα στοιχείο έχει τέσσερις ομόκεντρες περιοχές: την **content area**, την **padding area**, την **border area**, και την **margin area**.

Η περίμετρος κάθε περιοχής καλείται "πλευρά", έτσι κάθε κουτί έχει τέσσερις πλευρές:

- Η **content edge** : αυτή η πλευρά περιβάλλει το περιεχόμενο του στοιχείου που δίνεται. Εξ ορισμού το ύψος και το πλάτος του είναι τα ελάχιστα που απαιτούνται ώστε να περιέχεται το περιεχόμενο, αν και αυτό μπορεί να αγνοηθεί με τις τιμές που έχουν δηλωθεί ρητά. Το background style μιας περιεχόμενης περιοχής είναι η ιδιότητα **background** ενός παραγόμενου στοιχείου.
- Η **padding edge**: αυτή η πλευρά περιβάλλει το γέμισμα του κιβωτίου. Εάν το πλάτος του γεμίματος είναι 0 τότε αυτή η πλευρά είναι ίδια με την content edge. Το background style μιας περιεχόμενης περιοχής είναι η ιδιότητα **background** ενός παραγόμενου στοιχείου.
- Η **border edge**: αυτή η πλευρά περιβάλλει τα σύνορα του κιβωτίου. Εάν το πλάτος των συνόρων είναι 0 τότε αυτή η πλευρά είναι η ίδια με **padding edge**. Το ύψος της padding περιοχής ορίζεται από τις ιδιότητες **border** του παραγόμενου στοιχείου.
- Η **margin edge**: αυτή η πλευρά περιβάλλει το περιθώριο του κιβωτίου. Εάν το πλάτος περιθωρίου είναι 0 τότε αυτή η άκρη είναι η ίδια με την border edge.

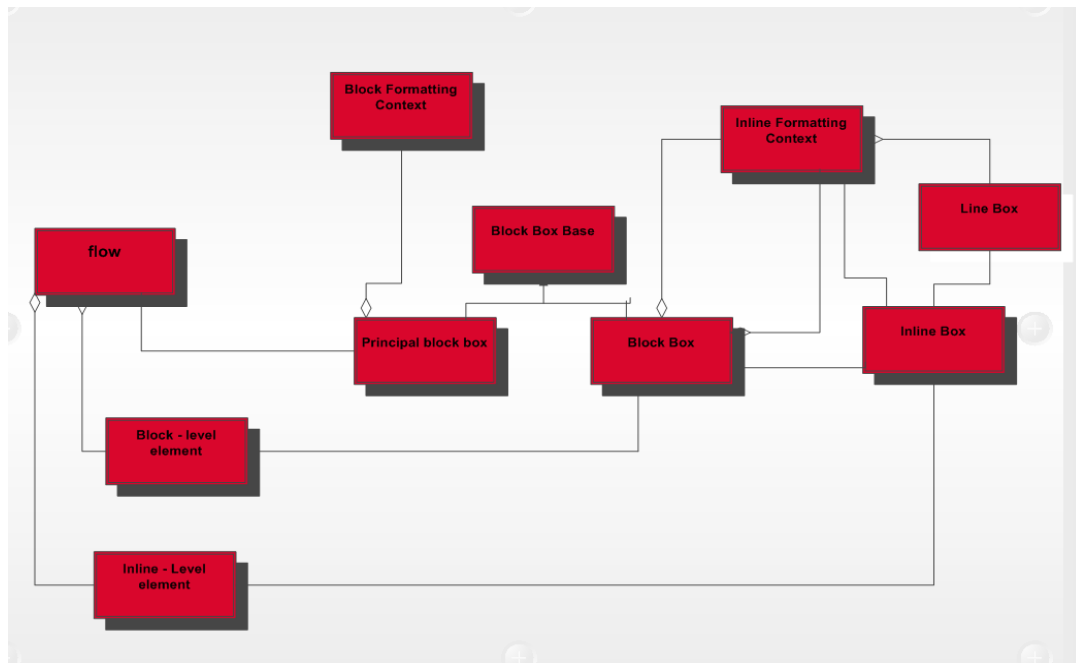
Οι ιδιότητες σχετικά με κάθε μια από τις πλευρές μπορούν να οριστούν χρησιμοποιώντας τα αντικείμενα value που αναφέρονται σε **top**, **bottom**, **left** και **right** μεμονωμένα τμήματα.

8.3.4 Τύποι πλαισίων των διαγραμμάτων ροής

8.3.4.1 Επισκόπηση

Ο αλγόριθμος διάταξης ενός δομικού στοιχείου flow χρησιμοποιεί διάφορους εικονικούς τύπους box. Αυτά δεν είναι αντικείμενα περιγραφής υπηρεσιών της PCF. Είναι εννοιολογικά αντικείμενα τα οποία χρησιμοποιούνται για να περιγράψουν το μοντέλο σχεδιαστικής αποτύπωσης των δομικών στοιχείων της PCF και του μαρκαρισμένου κειμένου μέσα σε ένα δομικό στοιχείο ροής.

Αυτά τα αντικείμενα, και οι σχέσεις μεταξύ τους, είναι διευκρινίζονται στη εικόνα 44.



Εικόνα 44. Μοντέλο αντικειμένων διαγραμμάτων ροής.

Τα τρία πιο αριστερά αντικείμενα τα οποία βρίσκονται μέσα στην εικόνα 44 αντιπροσωπεύουν τα στοιχεία μέσα σε ένα περιγραφικό δέντρο της PCF. Αυτά είναι:

- Flow: Είναι το Flow component. Αυτό περιλαμβάνει τα θυγατρικά στοιχεία τα οποία πρέπει να σχεδιαστούν χρησιμοποιώντας το flow layoutmodel.
- Block-level element: οποιοδήποτε θυγατρικό στοιχείο που απαιτεί οπτική μορφοποίηση ως ξεχωριστό μπλοκ .
- Inline-level element: οποιοδήποτε θυγατρικό στοιχείο που δεν μορφοποιεί έναν νέο μπλοκ με περιεχόμενο. Για παράδειγμα: υπογραμμισμένο κείμενο μέσα σε μια παράγραφο, ευθύγραμμες εικόνες και ευθύγραμμα components .

Τα υπόλοιπα αντικείμενα μέσα στο διάγραμμα είναι εννοιολογικά αντικείμενα τα οποία χρησιμοποιούνται για τον καθορισμό του flow layout model. Αυτά είναι:

- Block box: ένα layout box το οποίο χρησιμοποιείται για να περιέχει ένα στοιχείο block-level.
- Principle block box : τα block-level στοιχεία θα παράγουν ένα principal block box το οποίο περιέχει μόνο block boxes αντικείμενα. Το principal block box δημιουργεί ένα μπλοκ που θα περιέχει τους απογόνους .
- Block formatting context : τα κύρια block boxes δημιουργούν ένα Block formatting context το οποίο χρησιμοποιείται για την διάταξη των block boxes απόγονων.
- Inline box: ένα layout box που χρησιμοποιείται για να περιέχει ένα inline-level στοιχείο. Τα Inline boxes περιλαμβάνουν επίσης ένα block box το οποίο διαμορφώνει τα περιεχόμενα μπλοκ τους.
- Inline formatting context: το περιέχον block box καθιερώνει ένα inline formatting context για την διάταξη των απόγονων inline boxes..
- Line box: ένα inline formatting context ευθύγραμμίζει τα boxes σε μία η περισσότερες οριζόντιες γραμμές. Το πλαίσιο μορφοποίησης παράγει ένα line box για την παρουσίαση μίας ορθογώνιας περιοχής που περιέχει boxes που διαμορφώνουν μία μοναδική γραμμή.

8.3.4.2 Περιεχόμενα μπλοκ

Τα στοιχεία μέσα σε ένα αντικείμενο Flow περιέχονται μέσα σε μια δεντρική ιεραρχική δομή. Το αντικείμενο Flow περιέχει κάποια θυγατρικά στοιχεία και κάποια από αυτά τα θυγατρικά στοιχεία μπορεί να περιέχουν δικά τους θυγατρικά στοιχεία.

Μέσα σε αυτήν την δομή, το στοιχείο σε κάθε κόμβο του δέντρου σχεδιάζεται είτε ως ένα block box είτε ως ένα inline box.

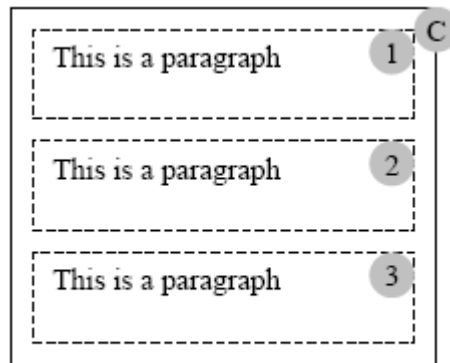
Το containing block ενός απόγονου box θα είναι η content-edge του κοντινότερου block-level προγόνου.

Το πλάτος του containing block που βρίσκεται στην κορυφή θα περιοριστεί από το πλάτος του container στο οποίο το αντικείμενο Flow θα εμφανιστεί. Το ύψος αυτού του block μπορεί επεκταθεί για να προσαρμόσει όλα boxes που είναι απογονοί του.

8.3.4.3 Τα στοιχεία Block-level

Τα στοιχεία **Block-level** είναι στοιχεία, όπως οι παράγραφοι, τα οποία απαιτούν οπτική μορφοποίηση ως ξεχωριστά block. Ένα block-level στοιχείο θα καταλαμβάνει την κάθετη περιοχή στην οποία βρίσκεται μέσα στο containing block

Η εικόνα 45 παρουσιάζει τρία block-level στοιχεία, που ονομάζονται 1, 2 και 3, μέσα στον containing block, το οποίο ονομάζεται C.



Εικόνα 45. Ένα σχεδιάγραμμα το οποίο αποτελείται block-level από στοιχεία

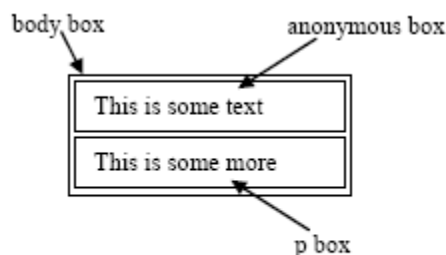
Τα στοιχεία Block-level θα παράγουν ένα **principal block box**, το οποίο περιέχει μόνο block boxes, και αυτά τα block boxes θα χρησιμοποιηθούν για την σχεδίαση των block level στοιχείων. Εάν ένα αντικείμενο Flow περιέχει μόνο inline-level στοιχεία, τότε ένα ανώνυμο block box θα δημιουργηθεί ώστε να περιέχονται σε αυτό. Εάν τα block-level και inline-level στοιχεία συνυπάρχουν στο ίδιο επίπεδο μέσα στο δέντρο τότε ένα ανώνυμο block box θα δημιουργηθεί για να συμπεριλάβει κάθε παρακείμενη ακολουθία inline-level στοιχείων.

Παράδειγμα:

Δίνεται το παρακάτω marked-up κείμενο

```
<body>
This is some text
  <p>This is some more text</p>
</body>
```

Το στοιχείο body περιλαμβάνει ένα inline και ένα block-level στοιχείο. Ένα ανώνυμο block θα δημιουργηθεί ώστε να περιλαμβάνει το κείμενο "This is some text".



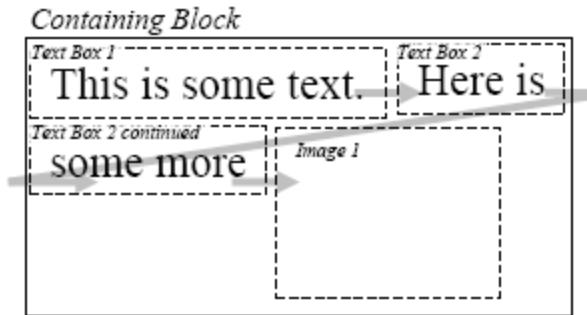
Εικόνα 46. Ένα σχεδιάγραμμα το οποίο αποτελείται από ανώνυμα *block-boxes*.

8.3.4.4 Κείμενο Μορφοποίησης

Μέσα σε ένα **block formatting context** τα block boxes θα διαταχθούν κάθετα το ένα μετά το άλλο ξεκινώντας από την κορυφή του containing block. Η κάθετη απόσταση μεταξύ των boxes που βρίσκονται στο ίδιο επίπεδο στο δέντρο θα καθοριστεί από τις ιδιότητες margin αυτών των boxes.

8.3.4.5 Τα στοιχεία Inline-level

Τα στοιχεία **Inline-level** δεν απαιτούν έναν ξεχωριστό block μορφοποίησης. Θα κατανεμηθούν σε γραμμές, όπου κάθε διαδοχικό στοιχείο τοποθετείται σε οριζόντια θέση σε σχέση με το προηγούμενο αμφιθαλή του στοιχείο. Η εικόνα 47 απεικονίζει τρία inline-level στοιχεία, Text Box 1, Text Box 2 και Image 1 τα οποία είναι διατεταγμένα μέσα στο containing block.



Εικόνα 47. . Ένα σχεδιάγραμμα το οποίο αποτελείται από στοιχεία *inline-level*.

Στην εικόνα 47 το δεύτερο στοιχείο, το οποίο ονομάζεται **Text Box 2**, έχει χωριστεί σε δύο γραμμές δεδομένου ότι διαφορετικά θα υπερέβαινε την δεξιά άκρη του containing block.

Μέσα σε ένα containing block box, μερικά inline boxes θα παραχθούν ρητά από τα inline-level στοιχεία όπως **em**, **big** ή **small** .

Σε περιπτώσεις όπου το κείμενο δηλώνεται άμεσα μέσα στο στοιχείο block-level, ένα **ανώνυμο inline box** θα δημιουργηθεί για να το συμπεριλάβει.

Παράδειγμα:

```
<p>This is <em>emphasized</em> text</p>
```

Το **p** παράγει ένα block box το οποίο περιέχει τρία inline boxes. Το box για την λέξη "emphasized" είναι ένα inline box το οποίο έχει δημιουργηθεί από ένα **em** inline στοιχείο. Τα άλλα inline boxes για τις λέξεις "This is" και "text" , έχουν δημιουργηθεί από το block-level στοιχείο, **p**, και είναι ανώνυμα inline boxes.

8.3.4.6 Ευθύγραμμο πλαίσιο μορφοποίησης

Σε ένα **inline formatting context**, τα boxes θα διαταχθούν οριζόντια, το ένα μετά από άλλο, ξεκινώντας από στην κορυφή ενός containing block. Τα οριζόντια περιθώρια, τα σύνορα, και τα γεμίσματα θα πρέπει να τηρηθούν μεταξύ των boxes.

Τα Boxes θα διαταχθούν κάθετα ανάλογα με την **vertical-align** ιδιότητα τους. Η ορθογώνια περιοχή που περιέχει τα boxes που διαμορφώνουν μια γραμμή ονομάζεται line box.

Το πλάτος ενός line box θα καθοριστεί από το περιέχον block. Ένα line box θα πρέπει πάντα να είναι αρκετά ψηλό για όλα τα boxes που περιέχει. Ωστόσο, μπορεί να είναι πιο ψηλό από το ψηλότερο box που περιέχει. Όταν το ύψος ενός box B είναι μικρότερο από το ύψος του line box που το περιέχει, η κάθετη ευθυγράμμιση του B μέσα στο line box θα καθοριστεί από την vertical-align ιδιότητα.

Όταν διάφορα inline boxes δεν μπορούν να χωρέσουν οριζόντια μέσα σε ένα line box, θα κατανεμηθούν μεταξύ δύο ή περισσοτέρων line boxes τα οποία βρίσκονται σε ένα κάθετο σωρό. Κατά συνέπεια, μια παράγραφος είναι ένας κάθετος σωρός από line boxes. Τα Line boxes θα τοποθετηθούν στο σωρό χωρίς κάθετο χωρισμό και χωρίς επικάλυψη. Η αριστερή άκρη ενός line box θα αγγίξει την αριστερή άκρη του περιέχον block και η δεξιά άκρη θα πρέπει να ακουμπάει την δεξιά άκρη του περιέχον block. Line boxes με το ίδιο ευθύγραμμο πλαίσιο μορφοποίησης μπορεί να διαφέρουν στο ύψος.

Παράδειγμα :

Μια γραμμή που περιέχει μια εικόνα που έχει μεγάλο ύψος, απαιτεί ένα ψηλότερο line box, από άλλες γραμμές που περιέχουν μόνο το κείμενο.

Όταν το συνολικό πλάτος των inline boxes σε μια γραμμή είναι μικρότερο από το πλάτος του line box που τα περιέχει, η οριζόντια κατανομή τους μέσα στο line box θα καθοριστεί από την ιδιότητα "text-align". Εάν αυτή η ιδιότητα έχει την τιμή "justify", ο user agent μπορεί να επεκτείνει τα inline boxes. Ένα inline box δεν μπορεί να υπερβαίνει το πλάτος ενός line box άρα τα inline boxes θα χωριστούν σε διάφορα boxes και αυτά τα boxes θα κατανεμηθούν σε διάφορα line boxes.

Όταν ένα inline box διαχωρίζεται, τα περιθώρια, τα σύνορα και τα γεμίσματα δεν θα επηρεαστούν οπτικά κατά τον διαχωρισμό. Η μορφοποίηση των περιθωρίων, σύνορων, και των γεμισμάτων δεν μπορούν να καθοριστούν πλήρως εάν ο διαχωρισμός συμβεί μέσα σε μια αμφίδρομη ενσωμάτωση.

8.3.5 Ιδιότητες Σχεδιαγραμμάτων

8.3.5.1 Γενικές ιδιότητες

Η ιδιότητα **v-align** καθορίζει την κάθετη ευθυγράμμιση ενός θυγατρικού στοιχείου. Εάν το στοιχείο είναι ένα inline-level στοιχείο τότε η ιδιότητα v-align θα καθορίσει την ευθυγράμμιση του στοιχείου μέσα στο περιέχον line-box. Αυτή η ιδιότητα δεν έχει καμία επίδραση εάν το θυγατρικό στοιχείο είναι ένα block-level στοιχείο. Η ιδιότητα v-align μπορεί να έχει τις ακόλουθες τιμές:

- **Baseline:** ευθυγραμμίζει την βασική γραμμή ενός box με την βασική γραμμή ενός box "γονέα". Εάν το box δεν έχει μια βασική γραμμή, ευθυγραμμίζεται το κατώ μέρος του box με τη βασική γραμμή του γονέα.
- **Middle:** ευθυγραμμίζει την μεσοκάθετο του box με τη βασική γραμμή του box γονέα συν το μισό X-ύψος του γονέα.
- **Sub:** χαμηλώνει τη βασική γραμμή ενός box στην κατάλληλη θέση για τους subscripts του γονέα box.
- **Super:** ανυψώνει τη βασική γραμμή ενός box στην κατάλληλη θέση για τους superscripts του γονέα box.
- **Text-top:** ευθυγραμμίζει την κορυφή ενός box με την κορυφή του γονικού στοιχείου font.
- **Text-bottom:** ευθυγραμμίζει το κατώτατο σημείο ενός box με το κατώτατο σημείο του γονικού στοιχείου font.
- **Top:** ευθυγραμμίζει την κορυφή ενός box με την κορυφή του line box.
- **Bottom:** ευθυγραμμίζει το κατώτατο σημείο ενός box με το κατώτατο σημείο του line box.

Η ιδιότητα **h-align** καθορίζει την οριζόντια ευθυγράμμιση των θυγατρικών στοιχείων μέσα σε ένα block-level στοιχείο. Λαμβάνοντας υπόψη ότι ένα block-level στοιχείο περιλαμβάνει ένα σωρό από ένα ή περισσότερα line boxes η ιδιότητα h-align θα καθορίσει ώστε να ευθυγραμμιστούν τα στοιχεία inline-level μέσα σε ένα line box .

Η ιδιότητα h-align μπορεί να έχει τις ακόλουθες τιμές:

- **Left:** τα στοιχεία inline-level θα ευθυγραμμιστούν στην αριστερή άκρη του containing line box που ανήκουν.
- **Right:** τα στοιχεία inline-level θα ευθυγραμμιστούν στην δεξιά άκρη του containing line box που ανήκουν.
- **Center:** τα στοιχεία inline-level θα ευθυγραμμιστούν ώστε να βρίσκονται στο κέντρο του containing line box που ανήκουν
- **Justify :** τα στοιχεία inline-level μπορούν να επεκταθούν ώστε να καταλάβουν ολόκληρο τον χώρο του containing line box που ανήκουν.

Η ιδιότητα **white-space** καθορίζει πώς αντιμετωπίζονται τα whitespace μέσα σε ένα στοιχείο .

8.3.5.2 Συγκεκριμένες ιδιότητες πλευρών

Οι ακόλουθες ιδιότητες μπορούν να χρησιμοποιηθούν στην γενική μορφή τους. Σε κάθε περίπτωση η ιδιότητα εφαρμόζεται και στις τέσσερις πλευρές (top, bottom, left και right) ενός box ή μπορούν να εφαρμοστούν σε μεμονωμένες πλευρές με την προσθήκη του κατάλληλου προθέματος .

Η ιδιότητα **margin** καθορίζει το πλάτος της περιθωρίου περιοχής ενός box. Η ίδια η ιδιότητα "margin" είναι μια στενογραφία για τον καθορισμό του πλάτους περιθωρίου και για τις τέσσερις πλευρές στην ίδια τιμή. Αυτές μπορούν επίσης να οριστούν χωριστά χρησιμοποιώντας τις ιδιότητες **margin-top**, **margin-bottom**, **margin-left** and **margin-right**. Όλες αυτές οι ιδιότητες καθορίζουν το πλάτος του περιθωρίου σε pixels .

Η ιδιότητα **padding** καθορίζει το πλάτος της περιοχής «γεμίσματος» ενός box. Η ίδια η ιδιότητα "padding" είναι μια στενογραφία για τον καθορισμό του πλάτους γεμίσματος και για τις τέσσερις πλευρές στην ίδια τιμή. Αυτές μπορούν επίσης να οριστούν χωριστά χρησιμοποιώντας τις ιδιότητες **padding-top**, **padding-bottom**, **padding-left** και **padding-right**. Όλες αυτές οι ιδιότητες καθορίζουν το πλάτος γεμίσματος σε pixels.

Το σύνολο των ιδιοτήτων **border** καθορίζουν το πλάτος, το χρώμα και το ύψος των συνόρων. Οι ιδιότητες border που προτάσσονται απλά ως "border" εφαρμόζουν την εν λόγω ιδιότητα και στις τέσσερις πλευρές. Οι ιδιότητες border μπορούν να εφαρμοστούν ξεχωριστά χρησιμοποιώντας τις ιδιότητες **border-top**, **border-bottom**, **border-left** και **border-right**.

Μπορούν να οριστούν οι ακόλουθες ιδιότητες συνόρων :

- **Width:** το πλάτος των συνόρων σε pixels .
- **Color:** το προεπιλεγμένο χρώμα των συνόρων.
- **Focuscolor:** το χρώμα των συνόρων όταν το στοιχείο έχει την ιδιότητα focus
- **Disabledcolor:** το χρώμα των συνόρων όταν το στοιχείο είναι ανενεργό.
- **Activecolor:** το χρώμα των συνόρων όταν το στοιχείο είναι ενεργό.
- **Idlecolor:** το χρώμα των συνόρων όταν το στοιχείο είναι αδρανοποιημένο.
- **Shadowcolor:** το χρώμα της σκίασης όταν στα σύνορα εφαρμόζεται ένα πολύχρωμο ύψος .
- **Highlightcolor:** το πιο εντυπωσιακό χρώμα που χρησιμοποιείται όταν στα σύνορα εφαρμόζεται ένα πολύχρωμο ύψος .
- **Linestyle-singlecolor:** το ενιαίο linestyle χρώμα που χρησιμοποιείται στα σύνορα. Η τιμή της μπορεί να είναι **solid**, **dashed**, **dotted** ή **double**.
- **Linestyle-multicolor:** το πολύχρωμο linestyle που χρησιμοποιείται στα σύνορα. Η τιμή της μπορεί να είναι **groove**, **ridge**, **bevelled-outset** και **bevelled-inset**.

8.4 Κείμενο ροής

Το TextFlow component είναι μια εξειδίκευση του Flow component και υπακούει σε όλους τους κανόνες ενός Flow component αλλά ο content τύπος του θα είναι SimpleMarkedUpText και δεν θα περιλαμβάνει θυγατρικά στοιχεία.

8.5 Σχεδιάγραμμα πίνακα

8.5.1 Εισαγωγή

Ένα Table layout είναι μια εξειδίκευση ενός Flow layout, το οποίο παρέχει ένα μηχανισμό όπου τα θυγατρικά components τοποθετούνται σε μια δισδιάστατη δομή.

Το component **Table** καθορίζει τον αλγόριθμο για την διάταξη ενός πλέγματος, και υπάρχει ανεξάρτητα από τα αντικείμενα Flow layout container που χρησιμοποιούνται για την παρουσίαση των Flows και tables .

Η οπτική εμφάνιση ενός **Table** διαχειρίζεται με ένα από τα τρία containers:

- **SFC** (Scroll Flow Container) .
- **PCF** (Page Flow Container).
- **TFC** (Truncate Flow Container).

8.5.2 Αλγόριθμοι σχεδιαγραμμάτων πινάκων

Η PCF δεν καθορίζει κάποιον βέλτιστο table layout αλγόριθμο. Οι εφαρμογές της PCF μπορούν να χρησιμοποιήσουν οποιοδήποτε κατάλληλο layout αλγόριθμο συμπεριλαμβανομένων εκείνων που δίνουν προτεραιότητα στην ταχύτητα πέρα από την ακρίβεια. Η μόνη εξαίρεση σε αυτό είναι μόνο εκεί όπου ο συντάκτης υπηρεσιών έχει ορίσει ότι ένας "fixed" table layout αλγόριθμος πρέπει να χρησιμοποιηθεί.

8.5.2.1 Σταθερο σχεδιάγραμμα πίνακα

Κατα την χρήση ενός "fixed" table layout αλγόριθμου , το οριζόντιο πλάτος του πίνακα και οι στήλες δεν εξαρτώνται από το περιεχόμενο των κελιών . Αντ' αυτού το πλάτος του πίνακα εξαρτάται μόνο από το πλάτος του container που εμφανίζεται, και το πλάτος των στηλών εξαρτάται από τις τιμές του πλάτους των στηλών, των συνόρων και των διαστημάτων των κελιών που έχουν δηλωθεί. Το γενικό πλάτος του πίνακα θα προέλθει από το διαθέσιμο εσωτερικό πλάτος του flow container, δηλαδή αφαιρούμε από το πραγματικό πλάτος του container, οποιοδήποτε γέμισμα.

Το πλάτος κάθε στήλης στον πίνακα θα καθοριστεί ως εξής:

- 1) Μια ιδιότητα **width** ενός στοιχείου column η οποία έχει τιμή διάφορη της "auto" ορίζει το πλάτος για αυτή τη στήλη.
- 2) Όπου **το πλάτος των** στηλών δεν έχει οριστεί , ένα κελί στην πρώτη γραμμή το οποίο έχει τιμή διάφορη της "auto" για την ιδιότητα **width** θα καθορίσει το πλάτος για αυτή την στήλη. Εάν το κελί εκτείνεται σε περισσότερες από μια στήλες, το πλάτος διαιρείται ομοιόμορφα στις στήλες που εκτείνονται.
- 3) Οι υπόλοιπες στήλες διαιρούν εξίσου τα υπόλοιπα οριζόντια διάστημα του πίνακα .Όταν το διαθέσιμο πλάτος ενός container παρουσίασης είναι μεγαλύτερο από το άθροισμα των τιμών των ιδιοτήτων **width** για κάθε μια από τις στήλες ,των συνόρων και των διαστημάτων των κελιών ,ο πίνακας θα τοποθετηθεί μέσα στο container σύμφωνα με την ευθυγράμμιση ροής για αυτό το container. Όταν το διαθέσιμο πλάτος ενός container παρουσίασης είναι μικρότερο από το άθροισμα των τιμών των ιδιοτήτων **width** για κάθε μια από τις στήλες ,των συνόρων και των διαστημάτων των κελιών η εφαρμογή PCF θα επιστρέψει ένα λάθος.

8.5.2.2 Αυτόματο σχεδιάγραμμα πίνακα

Όταν χρησιμοποιείται ένας "αυτόματος" table layout αλγόριθμος το οριζόντιο πλάτος του πίνακα προέρχεται από διαθέσιμο εσωτερικό πλάτος του flow container του. Ο «αυτόματος» table layout αλγόριθμος εφαρμόζεται μόνο όταν μία ή περισσότερες στήλες δεν έχουν ορισμένη την ιδιότητα width.

Τα πλάτη των στηλών καθορίζονται ως εξής:

- 1) Το ελάχιστο πλάτος για κάθε κελί στον **πίνακα** θα υπολογιστεί σύμφωνα με τους κανόνες που διευκρινίζονται στην παράγραφο 8.5.2.1. Εάν το πλάτος που έχει οριστεί για ένα κελί είναι μεγαλύτερο από το ελάχιστο πλάτος του περιεχομένου, τότε το πλάτος που έχει οριστεί καθορίζει το πλάτος του μικρότερου κελίου.Είναι

λάθος το πλάτος που έχει οριστεί να είναι μικρότερο από το ελάχιστο πλάτος περιεχομένου. Μία τιμή "auto" για την ιδιότητα **width** ενός κελιού σημαίνει ότι το ελάχιστο content πλάτος καθορίζει το ελάχιστο πλάτος κελιού.

2) Το μέγιστο πλάτος ενός κελιού θα υπολογιστεί με τη μορφοποίηση του περιεχομένου σε κάθε κελί.

3) Για κάθε στήλη, ένα ελάχιστο και ένα μέγιστο πλάτος στήλης θα καθοριστούν από τα κελιά που εκτείνονται μόνο σε αυτή την στήλη. Το ελάχιστο πλάτος στήλης είναι το μεγαλύτερο από τα ελάχιστα πλάτη των κελιών σε αυτή την στήλη. Το μέγιστο πλάτος στήλης είναι το μεγαλύτερο από τα μέγιστα πλάτη των κελιών σε αυτή την στήλη.

4) Εάν μια ιδιότητα πλάτους ορίζεται για μια στήλη και η τιμή της ιδιότητας πλάτους είναι μεγαλύτερη από το ελάχιστο πλάτος στήλης τότε η ιδιότητα πλάτους θα καθορίσει το πλάτος των στηλών. Είναι λάθος εάν η ιδιότητα πλάτους για τη στήλη να είναι μικρότερη από το ελάχιστο πλάτος στηλών.

5) Για κάθε κελί που εκτείνεται σε περισσότερες από μία στήλες, το άθροισμα των ελάχιστων πλατών στηλών για τις εκτεταμένες στήλες πρέπει να είναι μεγαλύτερο από το ελάχιστο πλάτος κελιών .

6) Εάν ο κανόνας 5 δεν ικανοποιείται, τότε το ελάχιστο πλάτος στηλών για κάθε εκτεταμένη στήλη που δεν έχει ορισμένη την ιδιότητα width θα αυξηθεί ομοιόμορφα έτσι ώστε να ικανοποιείται ο κανόνας 5. Η PCF δεν καθορίζει πώς αυξάνονται τα πλάτη των στηλών σε καταστάσεις όπου ο αριθμός των πρόσθετων pixels που απαιτούνται δεν είναι πολλαπλάσιο του αριθμού των στηλών που αυξάνονται. Είναι λάθος εάν κανένα ελάχιστο πλάτος στηλών δεν μπορεί να αυξηθεί.

Οι παραπάνω κανόνες παρέχουν το ελάχιστο και μέγιστο πλάτος για κάθε στήλη . Το πλάτος του container θα καθορίσει τα τελικά πλάτη των στηλών ως εξής:

1) Εάν το πλάτος του container είναι μεγαλύτερο από το άθροισμα των μεγίστων πλατών που απαιτούνται από όλες τις στήλες των συνόρων και των διαστημάτων τότε το τελικό πλάτος για κάθε στήλη θα είναι το μέγιστο πλάτος στηλών συν κάποιο πρόσθετο πλάτος. Το πρόσθετο πλάτος θα είναι η διαφορά μεταξύ του αθροίσματος όλων των μεγίστων πλατών των στηλών και του πλάτους του container ,το οποίο διανέμεται ομοιόμορφα σε όλες τις στήλες που δεν έχουν ορισμένη την ιδιότητα του πλάτους .

2) Εάν το πλάτος του container είναι μικρότερο από το άθροισμα των μεγίστων πλατών που απαιτούνται από όλες τις στήλες των συνόρων και των διαστημάτων, και μεγαλύτερο από το άθροισμα των ελαχίστων πλατών που απαιτούνται για όλες τις στήλες των συνόρων και των διαστημάτων τότε το τελικό πλάτος για κάθε στήλη θα είναι το ελάχιστο πλάτος στήλης συν ένα πρόσθετο πλάτος. Το πρόσθετο πλάτος θα είναι η διαφορά μεταξύ του αθροίσματος όλων των ελαχίστων πλατών των στηλών

και του πλάτους του container το οποίο κατανέμεται ομοιόμορφα σε όλες τις στήλες που δεν έχουν ορισμένη την ιδιότητα του πλάτους .

3) Είναι λάθος εάν το διαθέσιμο πλάτος του container παρουσίασης είναι μικρότερο από το άθροισμα των ελάχιστων πλατών που απαιτείται από όλες τις στήλες ,των συνόρων και των διαστημάτων.

8.5.2.3 Αλγόριθμος ύψους πίνακα

Το ύψος του πίνακα καθορίζεται από το άθροισμα των υψών των γραμμών , των συνόρων και των διαστημάτων .

8.5.2.4 Αλγόριθμος ύψους γραμμών

Το ύψος των γραμμών καθορίζεται ως εξής:

- 1) όταν η ιδιότητα **height** έχει μία τιμή διάφορη της "auto", το ύψος των γραμμών θα καθοριστεί από την τιμή αυτής της ιδιότητας .
- 2) όταν δεν έχει οριστεί η ιδιότητα **height** , ή έχει την τιμή "auto", το ύψος των γραμμών υπολογίζεται αφού όλα τα κελιά σε αυτή την σειρά διαταχθούν, και θα είναι ίσο με το ελάχιστο ύψος που απαιτείται από το πιο ψηλό κελι στη γραμμή.

8.5.2.5 Αλγόριθμος ύψους κελιών

Τα κελιά ενός πίνακα μπορούν να έχουν ορισμένη την ιδιότητα **height** ή μπορούν να κληρονομήσουν μια ιδιότητα **height** από τη γονική γραμμή του πίνακα.

Όταν σε ένα κελί η ιδιότητα **height** έχει δηλωθεί ή έχει ή κληρονομηθεί το περιεχόμενο σε αυτό το κελί δεν θα αναδιπλωθεί , εκτός αν το ύψος του κελιού που έχει δηλωθεί είναι αρκετά μεγάλο για να προσαρμόσει το αναδιπλωμένο περιεχόμενο. Το περιεχόμενο το οποίο έχει υπερχειλήσει και δεν μπορεί να προσαρμοστεί μέσα στα σύνορα ενός κελιού του οποίου το ύψος έχει οριστεί, θα περικοπεί.

Όταν οι ιδιότητες **height ενός κελιού** και **height μίας γραμμής** δεν έχουν δηλωθεί, ή έχουν την τιμή "auto" ,το περιεχόμενο σε αυτό το κελί θα είναι flowed χρησιμοποιώντας τον αλγόριθμο flow και το ύψος των κελιών θα είναι το ελάχιστο ύψος που απαιτείται για το περιεχόμενο αυτού του κελιού.

Όταν ένα κελί επεκτείνεται σε δύο ή περισσότερες γραμμές, το ύψος του κελιού θα προέλθει από το άθροισμα των τιμών **height** που έχουν δηλωθεί των επεκταμένων γραμμών. Εάν οι επεκταμένες γραμμές δεν έχουν δηλωμένη την ιδιότητα **height**, ή έχουν την τιμή "auto" για την ιδιότητα **height** , τότε το άθροισμα των υψών των γραμμών θα πρέπει να είναι αρκετά μεγάλο για να καλύψει το ελάχιστο ύψος του κελιού που επεκτείνεται στις γραμμές .

Όταν το ύψος των γραμμών δεν έχει δηλωθεί, και πρέπει να υπολογιστεί, ο υπολογισμός εξαρτάται από την κάθετη ευθυγράμμιση των κελιών μέσα σε αυτή την γραμμή .Η ιδιότητα vertical-align κάθε κελιού καθορίζει την ευθυγράμμισή του μέσα στη γραμμή.

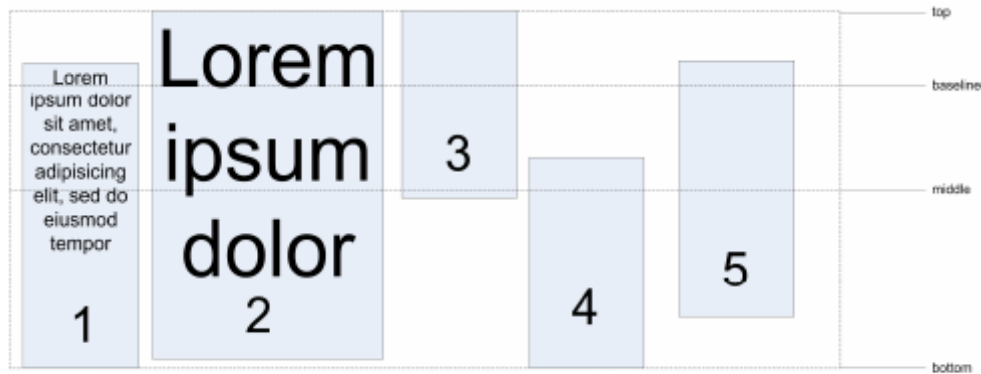
Οι τιμές για ιδιότητα vertical-align είναι οι ακόλουθες:

- **Baseline** - το κελί τοποθετείται έτσι ώστε η βασική γραμμή του περιεχομένου κελιού να ευθυγραμμίζεται με τη βασική γραμμή της περιέχουσας σειράς ,ή με την βασική γραμμή των αρχικών επεκταμένων γραμμών .
- **Top** - η κορυφή ενός block κελιού ευθυγραμμίζεται με την κορυφή της περιέχουσας γραμμής , ή με την κορυφή των αρχικών επεκταμένων γραμμών .
- **Bottom**- το κάτω μέρος ενός block κελιού ευθυγραμμίζεται με το κάτω μέρος της περιέχουσας γραμμής , ή με τις τελευταίες επεκταμένες γραμμές .
- **Middle** - το κέντρο του κελιού ευθυγραμμίζεται με το κέντρο της περιέχουσας γραμμής ή με το κεντρικό σημείο αρκετών επεκταμένων γραμμών.

Η βασική γραμμή ενός κελιού είναι η βασική γραμμή της πρώτης γραμμής του περιεχομένου σε αυτό το κελί. Εάν στο κελί δεν περιέχεται κείμενο, η βασική γραμμή θα είναι η βασική γραμμή οποιουδήποτε περιεχομένου είναι στο κελί. Εάν το κελί είναι κενό, η βασική γραμμή θα είναι το κάτω μέρος του cell block. Η μέγιστη απόσταση μεταξύ της κορυφής ενός cell block και της βασικής γραμμής όλων των κελιών που έχουν την τιμή "baseline" στην ιδιότητα **vertical-align** χρησιμοποιείται για τον ορισμό της βασικής γραμμής της σειράς .

Παράδειγμα:

Στο σχήμα τα cell blocks 1 και 2 είναι ευθυγραμμισμένα στην βασική γραμμή. Το Cell block 2 έχει το μεγαλύτερο ύψος πάνω στην βασική γραμμή, και καθορίζεται η βασική γραμμή για τη σειρά.



Εικόνα 48. Κάθετη ευθυγράμμιση κελιών

8.5.2.6 Ευθυγράμμιση περιεχομένου κελιού

8.5.2.6.1 Οριζόντια ευθυγράμμιση

Η οριζόντια ευθυγράμμιση του περιεχόμενου ενός κελιού θα καθοριστεί από την τιμή της ιδιότητα **horizontal-alignment** για αυτό το κελί.

Οι τιμές για τις ιδιότητες horizontal-alignment είναι:

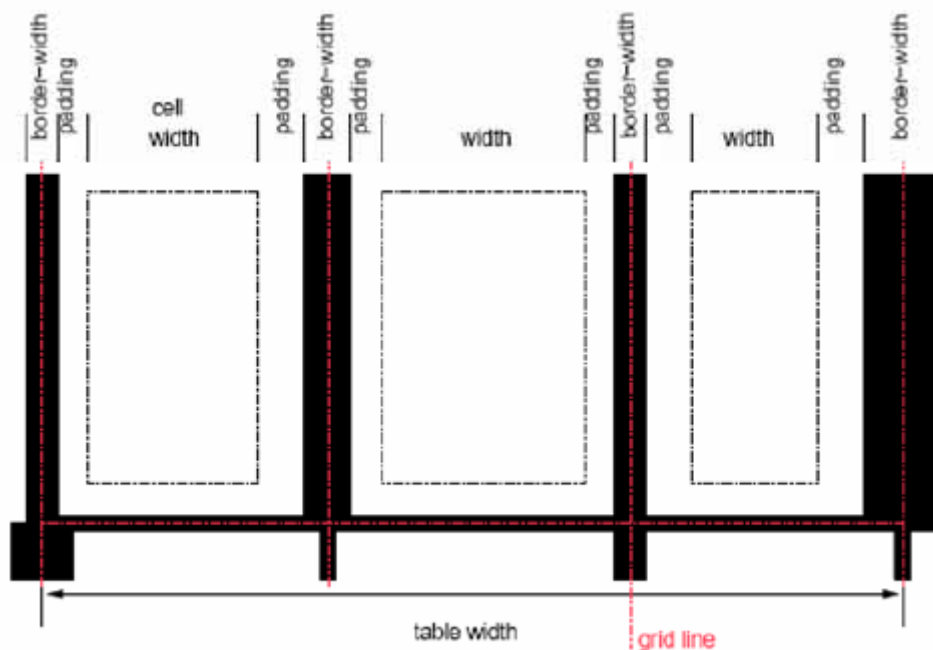
- **left**- το περιεχόμενο ευθυγραμμίζεται αριστερά στο κελί.
- **centre** - το περιεχόμενο τοποθετείται στο κέντρο του κελιού. Όταν το κελί περιέχει το αναδιπλωμένο περιεχόμενο, κάθε γραμμή περιεχομένου θα τοποθετηθεί στο κέντρο.
- **right** - το περιεχόμενο ευθυγραμμίζεται δεξιά στο κελί.
- **justify** - το περιεχόμενο θα στοιχιστεί με επιπλέον κενά διαστήματα έτσι ώστε το κείμενο να καταλάβει όλο το πλάτος του κελιού.

8.5.3 Σύνορα

Το πλάτος των συνόρων αλληλεπιδρά με το πλάτος του πίνακα, το πλάτος των κελιών, τα γεμίσματα και τα διαστήματα των κελιών. Η σχέση περιγράφεται από την παρακάτω εξίσωση, η οποία ισχύει για κάθε γραμμή του πίνακα :

Πλάτος γραμμών = $(0,5 * \text{πλάτος συνόρου}) + \text{αριστερό περιθώριο}_1 + \text{αριστερό γεμίσιμα}_1 + \text{πλάτος}_1 + \text{δεξιό γεμίσιμα}_1 + \text{δεξιό περιθώριο}_1 + \text{πλάτος συνόρων}_1 + \text{πλάτος συνόρων}_2 + \text{αριστερό περιθώριο}_2 + \dots + \text{δεξιό περιθώριο}_n + (0,5 * \text{πλάτος συνόρου})$.

Στην παραπάνω εξίσωση το n είναι ο αριθμός των κελιών σε μία γραμμή και το πλάτος των συνόρων αναφέρεται στα σύνορα των κελιών i και $i+1$



Εικόνα 49. Σχεδιαστικό διάγραμμα πλαισίων της PCF

8.6 Flow layout container components

Το μπλοκ που βρίσκεται στην κορυφή όλων των Flows θα έχει ένα σταθερό πλάτος, που ορίζεται ως το διαθέσιμο πλάτος της περιεχόμενης περιοχής ενός container στο οποίο εμφανίζεται ένα αντικείμενο Flow. Ωστόσο το ύψος του μπλοκ που βρίσκεται στη κορυφή θα μεγαλώσει ώστε να περιλαμβάνει όλα τα θυγατρικά στοιχεία που περιέχονται στο αντικείμενο Flow.

Το ύψος του μπλοκ που βρίσκεται στην κορυφή μπορεί να υπερβεί το ύψος της διαθέσιμης περιοχής ενός container component. Όταν αυτό συμβαίνει, είναι επιθυμητό να ακολουθηθεί ένας αριθμός διαφορετικών πολιτικών :

- περικόπεται το περιεχόμενο που υπερχειλίζει από την διαθέσιμη περιοχή
- αυξάνει ο ύψος του flow layout container ώστε προσαρμοστεί το πρόσθετο ύψος, μέχρι ένα ορισμένο μέγιστο μέγεθος .
- παρέχετε ένα scroll bar ώστε να επιτρέπετε στο χρήστη να κυλίσει προς τα κάτω το περιεχόμενο το οποίο έχει υποστεί υπερχείλιση .
- παρέχετε έναν μηχανισμό σελιδοποίησης ώστε να επιτρέπετε στο χρήστη τοποθετήσει σε σελίδα το περιεχόμενο το οποίο έχει υποστεί υπερχείλιση .

Αυτές οι τέσσερις πολιτικές εφαρμόζονται χρησιμοποιώντας τρία διαφορετικά flow layout container components:

- **SFC** (Scroll Flow Container);
- **PCF** (Page Flow Container);
- **TFC** (Truncate Flow Container).

Το **SFC** component υποστηρίζει κάθετη κύλιση του περιεχομένου . Εάν το περιεχόμενο ροής υπερβαίνει τη διαθέσιμη content περιοχή, ο χρήστης μπορεί να κυλίσει το περιεχόμενο χρησιμοποιώντας ένα ζεύγος προκαθορισμένων up/down κλειδιών .

Το **PFC** component χωρίζει ένα Flow σε διάφορες σελίδες, και επιτρέπει έπειτα στο χρήστη να πλοηγηθεί σε αυτές τις σελίδες χρησιμοποιώντας ένα ζευγάρι προκαθορισμένων previous/next κλειδιών.

Το **TFC** component δε εφαρμόζει ούτε σελιδοποίηση ούτε κύλιση του περιεχομένου. Εξ ορισμού ένα TFC component περικόπτει οποιοδήποτε περιεχόμενο υπερβαίνει τη διαθέσιμη content περιοχή.

8.7 Μοντέλο οθόνης αναφοράς

8.7.1 Η οθόνη αναφοράς

Υπάρχει μία σημαντική παραλλαγή στην ανάλυση της οθόνης στις πιθανές συσκευές στόχους .

Για την υποστήριξη του ορισμού της ακριβής θέσης των pixels στην οθόνη η PCF έχει υιοθετήσει ένα **reference screen model**.

Όλες οι PCF θέσεις μιας οθόνης καθορίζονται βάση μίας οθόνης αναφοράς. Αυτό είναι ένα ορθογώνιο σύστημα συντεταγμένων που περιλαμβάνει την οθόνη αναφοράς στην οποία θα τοποθετηθούν όλες οι θέσεις της οθόνης μέσα σε μία περιγραφή υπηρεσίας. Η ιδιότητα **referenceScreen** ενός PCF αντικειμένου υπηρεσίας καθορίζει το μέγεθος της οθόνης αναφοράς για μια συγκεκριμένη περιγραφή υπηρεσιών. Η χαρτογράφηση της οθόνης αναφοράς σε μια συσκευή στόχο είναι ευθύνη ενός PCF αποκωδικοποιητή.

8.7.2 Χαρτογράφηση της οθόνης αναφοράς σε μία συσκευή στόχο.

8.7.2.1 Η αναλύση της οθόνης της συσκευής στόχου είναι ίδια με την ανάλυση της οθόνης αναφοράς .

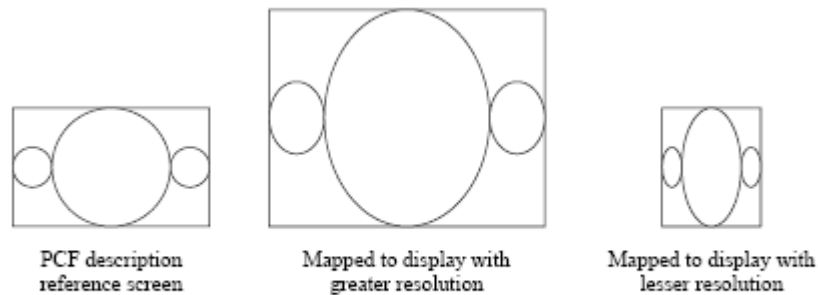
Όταν η ανάλυση της οθόνης της συσκευής στόχου και της οθόνης αναφοράς είναι η ίδια, οι συντεταγμένες των pixels στην οθόνη αναφοράς θα χαρτογραφηθούν άμεσα στις ισοδύναμες συντεταγμένες στην οθόνη της συσκευής. Ένα PCF αποκωδικοποιητής μπορεί να αγνοήσει τις ιδιότητες **referenceScreenMapping**, **referenceScreenAlignment** και **referenceScreenSurround** μιας υπηρεσίας PCF.

8.7.2.2 Η αναλύση της οθόνης της συσκευής στόχου είναι διαφορετική από την ανάλυση της οθόνης αναφοράς .

Αν η ανάλυση της οθόνης της συσκευής στόχου είναι διαφορετική από την ανάλυση της οθόνης αναφοράς τότε χρησιμοποιείται η ιδιότητα **referenceScreenMapping** ενός στοιχείου PCF υπηρεσίας για να δείξει πώς ένα PCF αποκωδικοποιητής θα χειριστεί αυτή την κατάσταση.

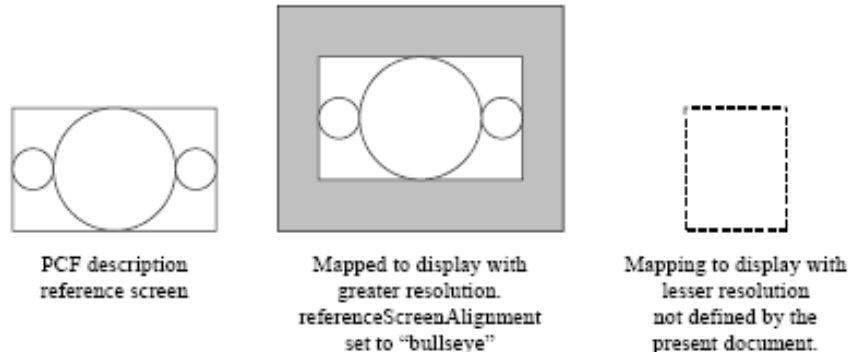
Εάν η ιδιότητα **referenceScreenMapping** έχει οριστεί ως "**display-anamorphic**" τότε η οθόνη αναφοράς θα κλιμακωθεί ώστε να γεμίσει την οθόνη μιας συσκευής

στόχου. Ωστόσο επειδή η οριζόντια κλιμάκωση μπορεί να είναι διαφορετική από την κάθετη η εμφάνιση που θα προκύψει μπορεί να είναι αναμορφωστική .



Εικόνα 50. Η ιδιότητα referenceScreenMapping έχει τεθεί ως "display-anamorphic"

Αν η ιδιότητα **referenceScreenMapping** έχει τεθεί "pixel" και η ανάλυση της οθόνης της συσκευής είναι μεγαλύτερη από την ανάλυση της οθόνης αναφοράς τότε η οθόνη αναφοράς δεν θα κλιμακωθεί. Η ιδιότητα referenceScreenAlignment ενός στοιχείου υπηρεσίας θα χρησιμοποιηθεί για να τοποθετήσει την οθόνη αναφοράς μέσα στην οθόνη της συσκευής και την ιδιότητα referenceScreenSurround του στοιχείου υπηρεσίας θα χρησιμοποιηθεί για να γεμίσει οποιοδήποτε μέρος οθόνης της συσκευής που δεν έχει χαρτογραφηθεί από την οθόνη αναφοράς.



Εικόνα 51. Η ιδιότητα referenceScreenMapping έχει τεθεί ως "pixel".

8.7.2.3 Κλιμάκωση της οθόνης αναφοράς .

Η PCF δεν διευκρινίζει τον αλγόριθμο που χρησιμοποιείται κατά τη χαρτογράφηση των συντεταγμένων από την οθόνη αναφοράς στην οθόνη της συσκευής σε περιπτώσεις όπου η ανάλυση δεν είναι η ίδια. Οι PCF αποκωδικοποιητές μπορούν να εφαρμόσουν αλγορίθμους κλιμάκωσης όπου είναι απαραίτητο σε συγκεκριμένες πλατφόρμες στόχους .

8.8 Εγγραφή βίντεο και γραφικών.

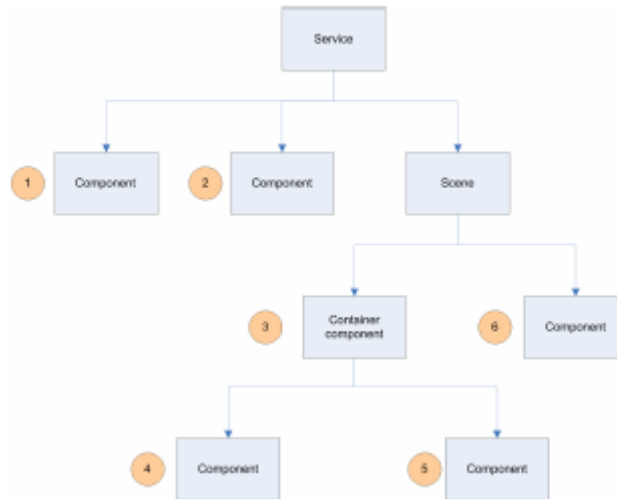
Η PCF δεν κάνει καμία υπόθεση σχετικά με τη φυσική εφαρμογή των συσκευών στόχων και χειρίζεται όλα τα ορατά components μέσα σε ένα χώρο συντεταγμένων, πχ μια οθόνη αναφοράς.

Σε αυτό το μοντέλο εγγραφής γραφικών σε ένα συγκεκριμένο σημείο στο video επιτυγχάνεται πάντα. Εντούτοις, στην πράξη το βίντεο και τα γραφικά μπορούν να αντιμετωπιστούν διαφορετικά μέσα στο δέκτη και οι απαιτήσεις για την επίτευξη του σωστού λόγου διάστασης για οποιοδήποτε βίντεο και γραφικών δεν επιτυγχάνεται πάντα

Οι ιδιότητες **serviceAspectRatio** και **videoHandlingPriority** ενός στοιχείου υπηρεσιών PCF μπορούν να χρησιμοποιηθούν από ένα PCF αποκωδικοποιητή για να βελτιστοποιήσει την παρουσίαση μιας υπηρεσίας σε μια συγκεκριμένη συσκευή στόχο.

8.9 Μοντέλο εμφάνισης σωρού

8.9.1 Αρχικοποίηση του σωρού εμφάνισης



Εικόνα 52.Ταξινόμηση στοιχείων συγκεκριμένου τύπου



Εικόνα 53.Μοντέλο επίδειξης επιπέδων σωρού

8.9.2 Χειρισμός του σωρού εμφάνισης

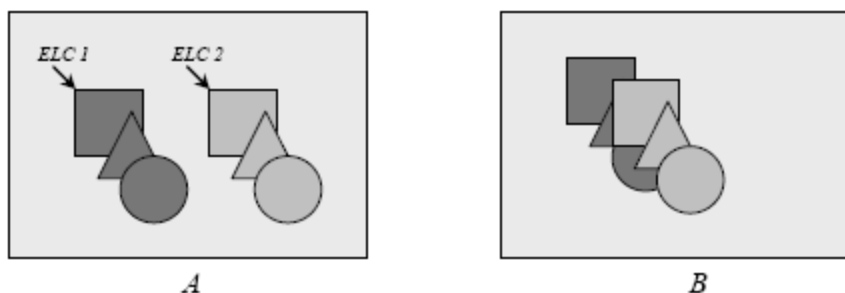
Τα οπτικά components μπορούν να μετακινηθούν πάνω-κάτω μέσα στο display stack χρησιμοποιώντας τις ακόλουθες ενέργειες:

- **infront** - για να μετακινηθεί ένα αντικείμενο μπροστά από ένα άλλο
- **behind** - για να μετακινηθεί ένα αντικείμενο πίσω από ένα άλλο
- **top** - για να μετακινηθεί ένα αντικείμενο στην κορυφή του σωρού.
- **bottom**- για να μετακινηθεί ένα αντικείμενο στο κάτω μέρος του σωρού .

Όταν ένα οπτικό component κινείται κατ' αυτό τον τρόπο, άλλα οπτικά component μπορούν μερικώς ή πλήρως να κρυφτούν ή να αποκαλυφθούν. Τα Container components μπορούν επίσης να μετακινηθούν πάνω-κάτω μέσα στο display stack όπου αυτό επιτρέπεται στον καθορισμό της δράσης τους.

Παράδειγμα:

Το σχήμα επεξηγεί το μοντέλο display stack Στο διάγραμμα *A* υπάρχουν δύο Explicit Layout Containers, τα ELC1 και ELC2. Και τα δύο έχουν τρία επικαλύπτομενα components. Το ELC 2 δηλώθηκε μετά από το ELC1, και επομένως τα θυγατρικά του components εμφανίζονται επάνω από ELC1 στον display stack Αυτό είναι διευκρινίζεται στο διάγραμμα *B*, όπου το component του ELC2 που βρίσκεται πιο πίσω από τα υπόλοιπα είναι μπροστά από το πιο μπροστινό component του ELC1.



Εικόνα 54. Συμπεριφορά του σωρού εμφάνισης .

8.10 Επιλογή γραμματοσειράς

Η PCF υποστηρίζει τα ακόλουθα κριτήρια επιλογής γραμματοσειράς .

- **font-family** - ορίζει την οικογένεια των στοιχείων που μπορούν να χρησιμοποιηθούν για την απόδοση του κείμενου.
- **font-emphasis** - ορίζει όπου χρειάζεται αν τα στοιχεία θα εμφανιστούν χρησιμοποιώντας κανονική πλάγια έντονη μορφή.
- **font-emphasis** - ορίζει την έμφαση για τα στοιχεία .
- **font-variant**- ορίζει εάν τα στοιχεία πρέπει να αποδοθούν χρησιμοποιώντας κανονικά ή μικρα γραφικά σύμβολα
- **font-weight** – ορίζει την ευκρίνεια των γραφικών συμβόλων που χρησιμοποιούνται για την απόδοση του κείμενου.
- **font-stretch** - ορίζει το βαθμό συμπίκνωσης ή επέκτασης των γραφικών συμβόλων στα glyphs που χρησιμοποιούνται για την απόδοση του κείμενου.
- **font-size**- ορίζει το μέγεθος των στοιχείων .

9 Προδιαγραφές συμπεριφοράς

Στον πυρήνα μιας διαδραστικής υπηρεσίας υπάρχει ένα σύστημα καθοδηγούμενων γεγονότων, που επιτρέπουν σε εξωτερικά γεγονότα να καθορίσουν τη διαδρομή κατευθείαν στο σύστημα παρά να ακολουθήσουν κάποια εσωτερική διαδρομή. Η πολυπλοκότητα μιας διαδραστικής υπηρεσίας προέρχεται από γεγονότα/ενέργειες που συνδέονται συχνά με θέματα συναγωνισμού και συγχρονισμού. Το σχήμα συμπεριφοράς της PCF παρέχει ένα σύντομο τρόπο σύλληψης αυτής της πολυπλοκότητας με τρόπο που μπορεί εύκολα και αξιόπιστα να εφαρμοστεί, να αναλυθεί και να ελεγχθεί.

9.1 Εισαγωγή

Η συμπεριφορά της PCF προσφέρει σε ένα συντάκτη υπηρεσίας:

- μια περιγραφή των διαφορετικών τύπων **event** και πώς αυτά μεταδίδονται μέσα σε μια δομή υπηρεσίας .
- την δυνατότητα να περιγράψει ένα συγκεκριμένο component το οποίο έχει την ιδιότητα **focus** που επηρεάζει τη μέθοδο μετάδοσης του γεγονότος.
- μια **action language** για να περιγράψει την απόκριση σε ένα γεγονός.
- μια μέθοδο για την περιγραφή συγκεκριμένων στιγμών στη διάρκεια ζωής μιας διαδραστικής υπηρεσίας οι οποίες μπορούν « συλληφθούν» και να περιγραφούν ως μία κατάσταση - **state**.
- μια μέθοδο για την περιγραφή μιας ακολουθίας states και actions που εμφανίζονται ως απόκριση σε συγκεκριμένα γεγονότα βασισμένα σε **statemachines** .
- μια γραφική αναπαράσταση των statemachines υπό μορφή **statecharts**
- μια XML κειμενική αναπαράσταση των statemachines.

Το μοντέλο συμπεριφοράς είναι υψηλού επιπέδου, ελευθερώνοντας μια υπηρεσία από τον περιορισμό των λεπτομερειών της εφαρμογής και διευκολύνει τους συντάκτες υπηρεσιών να εκφράσουν περιληπτικά την πρόθεσή τους.

9.1.1 Ενδογενής συμπεριφορά δομικών στοιχείων

Η PCF παρέχει ένα περιεκτικό σύνολο δομικών στοιχείων όπου ένας συντάκτης υπηρεσιών μπορεί να χρησιμοποιήσει για να δημιουργήσει μια υπηρεσία. Μερικά από αυτά τα δομικά στοιχεία έχουν την ενδογενή συμπεριφορά που τα αναγκάζει να αντιδρούν σε συγκεκριμένα γεγονότα χωρίς χρειάζονται κάποιες πρόσθετες PCF δηλώσεις συμπεριφοράς.

Παράδειγμα 1:

Ως απόκριση στα πλήκτρα του τηλεχειριστηρίου UP και DOWN το highlight σε ένα μενού που αποτελείται από components θα μετακινήσει πάνω - κάτω τήν λίστα των ετικετών με τις διαθέσιμες επιλογές .

Παράδειγμα 2:

Ο συντάκτης υπηρεσίας μπορεί να ελέγξει το χρώμα που συνδέεται με την τονισμένη ετικέτα επιλογών παρέχοντας μία τιμή για την ιδιότητα **textcolor-focus** ενός μενού από components .

9.1.2 Ανεξάρτητη συμπεριφορά

Όταν τα PCF components δεν παρέχουν την απαραίτητη behaviour για την δημιουργία μια συγκεκριμένης προοριζόμενη εμπειρίας του χρήστη, ο συντάκτης υπηρεσίας μπορεί να προσθέσει τη συμπεριφορά τους. Αυτό θα επιτευχθεί χρησιμοποιώντας έναν συνδυασμό PCF statemachines και action language περιγραφών με :

- την δημιουργία προσαρμοσμένων δομικών στοιχείων με την επιθυμητή συμπεριφορά,
- την σύνδεση ανεξάρτητων statemachines,

9.2 Γεγονότα

9.2.1 Run-time event model

Τα γεγονότα στην PCF είναι αντικείμενα που έχουν μια παροδική ύπαρξη κατά τον χρόνο εκτέλεσης, και μπορούν μόνο να προσεγγιστούν από την action language που προκαλείται από αυτό το γεγονός.

Ένα γεγονός έχει ένα **eventtype** που καθορίζει ένα σύνολο ονομασμένων ιδιοτήτων. Μέσα σε μία δήλωση ενός component, το ενδιαφέρον για ένα γεγονός δηλώνεται μέσω του αντικειμένου **trigger** μέσα σε ένα **onEvent** αντικείμενο ή σε ένα **transition** αντικείμενο κατάστασης.

Το αντικείμενο onEvent ή το αντικείμενο κατάστασης **transition** καθορίζουν ένα **name** για το γεγονός που υπάρχει στο περιεχόμενο του component. Για τους σκοπούς της action language, τα γεγονότα που υπάρχουν στο περιεχόμενο ενός δομικού στοιχείου θα θεωρηθούν ότι είναι components τα οποία έχουν οριστεί στο ίδιο περιεχόμενο. Κατά συνέπεια το όνομα γεγονότος μπορεί να χρησιμοποιηθεί για να υπάρχει άμεση πρόσβαση στις ιδιότητες που περιλαμβάνονται μέσα στο γεγονός.

Ένα γεγονός μπορεί να περιέχει μια ιδιότητα **source** που δείχνει το πηγαίο component που είναι υπεύθυνο για τη μετάδοση του γεγονότος σε ένα context στο οποίο εκτελείται η action language.

9.3 Μοντέλο μετάδοσης γεγονότων

Το μοντέλο μετάδοσης γεγονότων είναι απαραίτητο μέσα στην PCF έτσι ώστε ένας αποκωδικοποιητής να είναι σε θέση να αντιληφθεί την PCF συμπεριφορά που έχει καθοριστεί. Σε αυτό το αφηρημένο μοντέλο περιλαμβάνονται οι έννοιες **system events**, **user input events** και **component event** και πως αυτές επηρεάζονται από την ιδιότητα focus.

9.3.1 Εισαγωγή

9.3.1.1 Μοντέλο Αντικειμένων

Το μοντέλο αντικειμένων που παρουσιάζεται στο σχήμα εξηγεί τη σχέση μεταξύ ενός γεγονότος και της δομής της υπηρεσίας .

Μια statemachine μπορεί να ενδιαφέρεται για κανένα ή περισσότερα γεγονότα.

Ένα component συνδέεται με ένα statemachine. Ως συνέπεια ένα component ενδιαφέρεται για εκείνα τα γεγονότα που συνδέονται με την statemachine του.

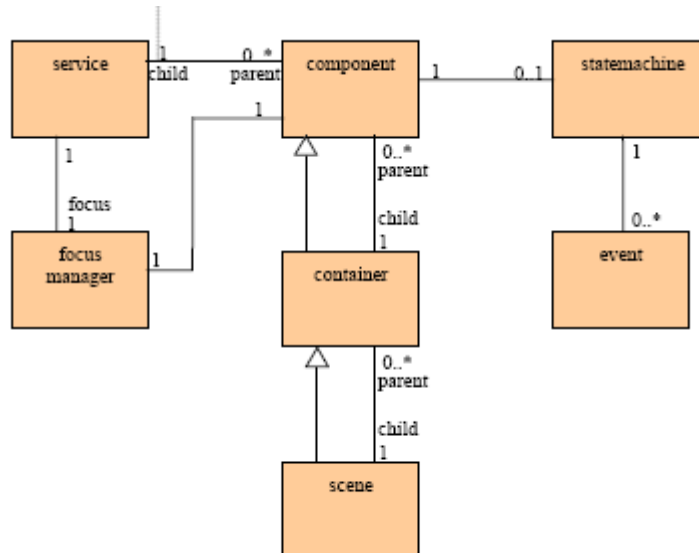
Ένα container είναι ένας τύπος component και συνδέεται με κανένα ή περισσότερα component .

Ένα scene είναι ένα container και, επομένως, είναι ένας τύπος component.

Μια υπηρεσία συνδέεται με ένα ή περισσότερα components.

Ένα **focus manager** θα χρησιμοποιηθεί για τον προσδιορισμό του component μέσα σε μια υπηρεσία που έχει την ιδιότητα focus. Μόνο ένα component θα έχει την

ιδιότητα focus οποιαδήποτε στιγμή κατά τη διάρκεια της παροχής μιας υπηρεσίας σε έναν δέκτη. Αυτό το component μπορεί να δηλωθεί σε ένα επίπεδο scene σε ένα επίπεδο container ή σε ένα basic component επίπεδο.



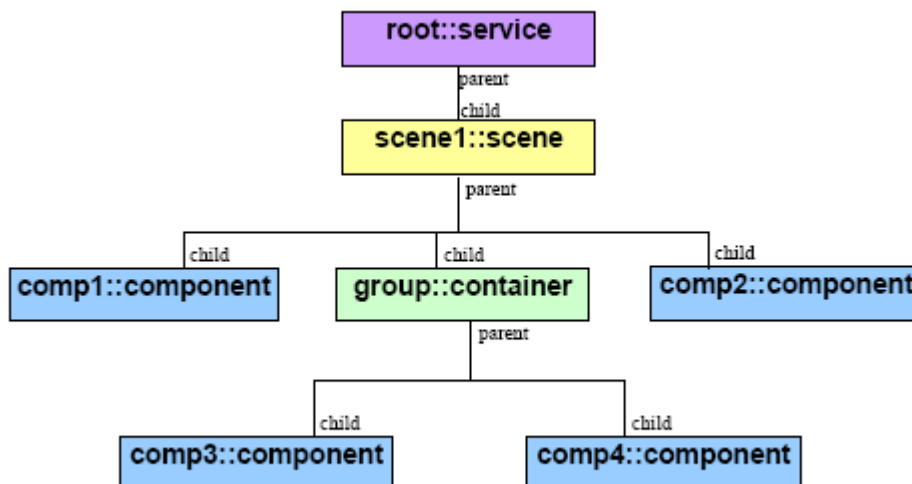
Εικόνα 55. Το μοντέλο αντικειμένων παρουσιάζει πως συνδέονται τα γεγονότα με άλλες πτυχές της υπηρεσίας

9.3.1.2 Ιεραρχία περιέξεως των δομικών στοιχείων .

Ένα scene θα κατασκευαστεί από έναν συνδυασμό λογικών σχηματισμών διάφορων components. Αυτό οδηγεί σε μια component ιεραρχία. Υπάρχει μια parent-child σχέση περιέξεως μεταξύ των components που βρίσκονται σε διαφορετικά επίπεδα στην ιεραρχία.

Παράδειγμα:

Η ιεραρχία των component που απεικονίζεται στο σχήμα παρουσιάζει μια σχέση συγκράτησης μέσα σε μία υπηρεσία που ονομάζεται "root". Παραδείγματος χάριν, το "group" container είναι ο γονέας του "comp3" και του "comp4". Αυτό είναι απόγονος του "scene1". Οι απόγονοι ενός component περιλαμβάνουν όλες τις γενιές περιέξεως. Επομένως, οι απόγονοι του "scene"1 είναι τα "comp1", "group", "comp2" τα θυγατρικά αντικείμενα του "group" είναι τα "comp3" and "comp4". Όλα τα components μέσα στην ιεραρχία ενός scene είναι ενεργά όταν το αντικείμενο scene είναι ενεργό.



Εικόνα 56. Αφηρημένη ιεραρχία των δομικών στοιχείων.

9.3.1.3 Μετάδοση Γεγονότων

Ένα γεγονός της PCF είναι ένα περιστατικό μέσα στη διάρκεια ζωής μιας υπηρεσίας που μπορεί να επικαλεσθεί με κάποιο τρόπο κάποια απόκριση. Οι τύποι γεγονότων περιλαμβάνουν τα system events, τα γεγονότα που εισάγονται από χρήστες και τα component events.

Το μοντέλο διάδοσης γεγονότος καθορίζει τους κανόνες που προσδιορίζουν τη δρομολόγηση των γεγονότων γύρω από την ιεραρχία περιέλιξης των component. Το μοντέλο παρέχει μια συνεπή λύση και για τους τρεις τύπους γεγονότων αναγνωρίζοντας ταυτόχρονα τους διαφορετικούς σκοπούς τους.

9.3.2 Γεγονότα συστήματος

9.3.2.1 Επισκόπηση

Τα System events είναι ανεξάρτητα από την αλληλεπίδραση του χρήστη με την υπηρεσία και έτσι δεν συνδέονται με το component που έχει την ιδιότητα focus. Αντ' αυτού, μια απόκριση μπορεί να κερδηθεί από ένα interested component μέσα στη ιεραρχία των component.

Τα System events περιλαμβάνουν:

- αναπροσαρμογές αρχείων/αντικειμένων στην μετάδοσης ροής
- συγχρονισμός με τις ροές
- in-band ωθήσεις.

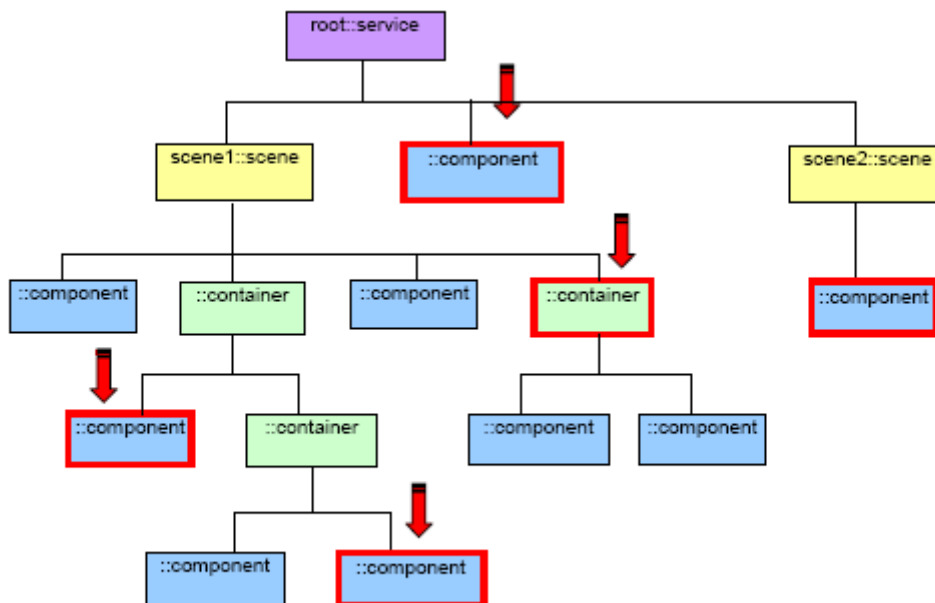
9.3.2.2 Κανόνες μετάδοσης των γεγονότων συστήματος

Ένα γεγονός θα στοχεύσει αρχικά σε ένα Service component.

- Το γεγονός θα στοχεύσει στο ενεργό scene και σε όλους τους απογόνους του. Όλα τα components που ενδιαφέρονται για αυτό το γεγονός μπορεί να αποκριθούν.
- Το γεγονός θα στοχεύσει σε όλους τους απογόνους ενός **Service** component αποκλείοντας οποιαδήποτε μη ενεργά scene components.

Παράδειγμα:

Υπάρχουν πέντε components που ενδιαφέρουν ένα συγκεκριμένο system event στην ιεραρχία των components και παρουσιάζονται στην εικόνα 57. Το system event στοχεύει σε κάθε ένα από τα τρία components μέσα στο ενεργό scene (scene1) και στοχεύει στους άμεσους απογόνους της υπηρεσίας που υποδεικνύεται με τα κόκκινα βέλη. Το highlighted component που περιλαμβάνεται μέσα στο scene2 δεν στοχεύεται.



Εικόνα 57. Επίπεδα δομικών στοιχείων όπου ένα γεγονός στοχεύει στο δομικό στοιχείο service και στα δομικά στοιχεία των ενεργών scene.

9.3.3 Γεγονότα τα οποία εισάγονται από τον χρήστη

9.3.3.1 Επισκόπηση

Τα γεγονότα User input παράγονται ως αποτέλεσμα της αλληλεπίδρασης με τον χρήστη.

Το πεδίο της PCF εξετάζει μόνο:

- Remote control key press (εικονικά κλειδιά).

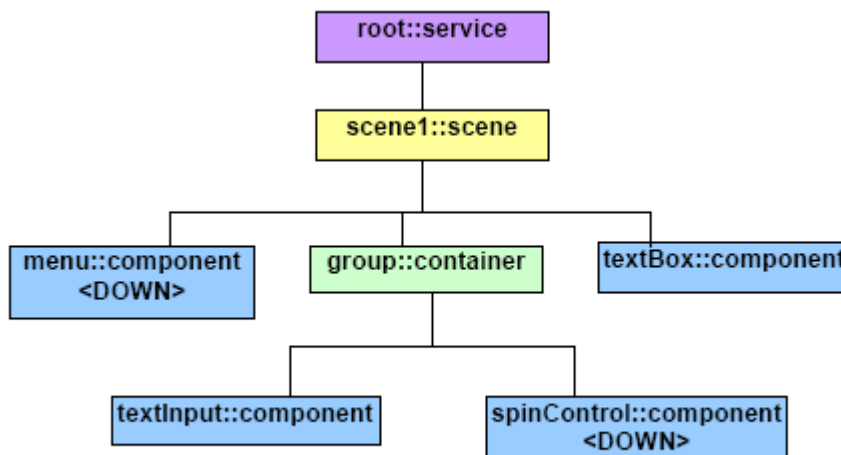
Είναι η έννοια της εστίασης που καθιστά το μοντέλο διάδοσης γεγονότος για τα user input events διαφορετικό από τα system events.

9.3.3.2 Έλεγχος εστίασης

Μόνο ένα component θα έχει την ιδιότητα focus οποιαδήποτε στιγμή. Ένα user event θα στοχεύσει αρχικά στο component το οποίο έχει προσωρινά την ιδιότητα focus. Εάν κανένα component δεν έχει αναγνωριστεί με την ιδιότητα focus τότε εξ ορισμού, το scene υποθέσει την εστίαση και ένα user event θα στοχεύσει αρχικά στο scene. Η PCF action language θα διευκολύνει τη μετακίνηση της εστίασης μεταξύ των components

Παράδειγμα:

Η ιεραρχία των component που παρουσιάζεται στο σχήμα περιλαμβάνει ένα Menu και ένα SpinControl που αποκρίνονται στο user event DOWN. Εάν το Menu έχει την ιδιότητα focus όταν δημιουργείται το γεγονός DOWN τότε το γεγονός θα στοχεύσει αρχικά στο component Menu. Το Menu ανταποκρίνεται στο γεγονός. Το SpinControl component παρόλο που ενδιαφέρει το γεγονός δεν αποκρίνεται.



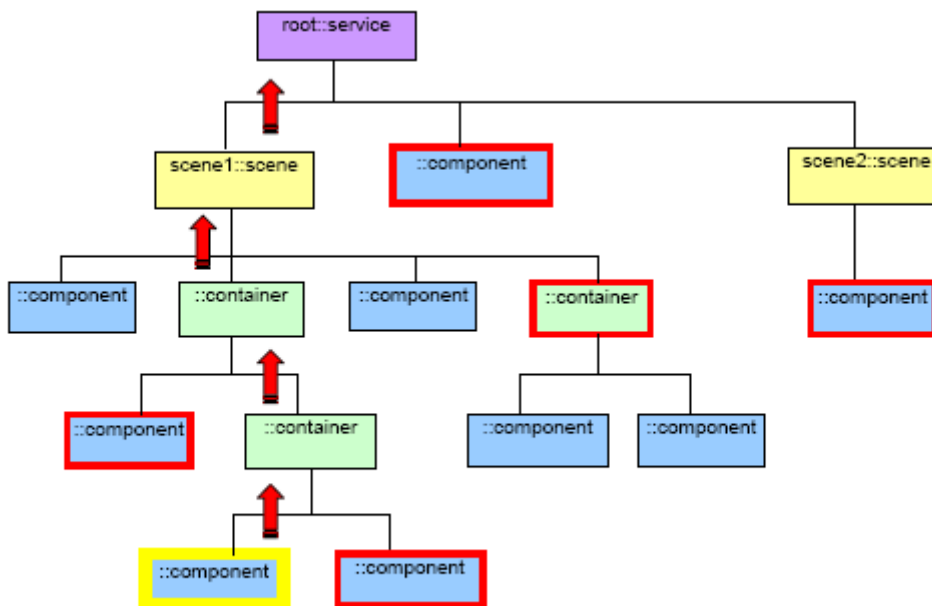
Εικόνα 58. Ιεραρχία των δομικών στοιχείων μέσα σε ένα παράδειγμα scene

9.3.3.3 Κανόνες μετάδοσης γεγονότων που εισάγονται από τον χρήστη

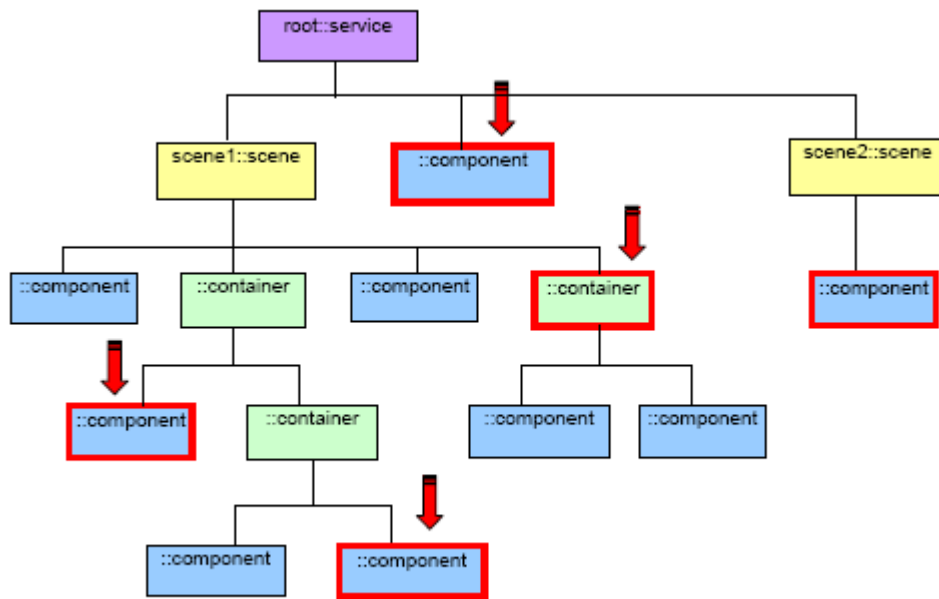
- 1) Ένα γεγονός θα στοχεύσει αρχικά στο focused component.
- 2) Εάν ένα component ενδιαφέρει το γεγονός τότε η μετάδοση του γεγονότος θα ολοκληρωθεί και το component μπορεί να αποκριθεί στο γεγονός.
- 3) Εάν ένα component δεν ενδιαφέρει το γεγονός τότε το γεγονός θα στοχεύσει στον πρόγονο του τρέχοντος component .
- 4) Τα βήματα 2 και 3 επαναλαμβάνουν μέχρι :
 - ένα component να ενδιαφέρει το γεγονός και η μετάδοση να ολοκληρωθεί(βήμα 2) ή
 - το component στόχος είναι το Service component.
- 5) Εάν το Service component δεν ενδιαφέρει το γεγονός τότε η μετάδοση του γεγονότος συνεχίζεται σαν ήταν ένα system event.

Παράδειγμα:

Το component με την ιδιότητα focus στο σχήμα 39 είναι τονισμένο με κίτρινα σύνορα. Εκείνα τα components που ενδιαφέρουν ένα συγκεκριμένο γεγονός έχουν χρωματιστεί με κόκκινα σύνορα. Το γεγονός στοχεύει πρώτα στο focused component. Αυτό το component δεν ενδιαφέρει στο συγκεκριμένο γεγονός και έτσι το γεγονός στοχεύει στον απόγονο αυτού του component. Κανένα από τα component στην διαδρομή με προορισμό το Service component δεν ενδιαφέρει το γεγονός. Όταν το γεγονός φθάνει στο Service component θα μεταδοθεί σαν να ήταν ένα system event. Στοχεύει όλα εκείνα τα components μέσα στο ενεργό scene που ενδιαφέρουν το συγκεκριμένο γεγονός. Αυτό παρουσιάζεται στο σχήμα όπου το γεγονός στοχεύει στα τρία components μέσα στο scene1. Το γεγονός δεν στοχεύει στο component μέσα στο scene2 παρόλο που ενδιαφέρει το γεγονός.



Εικόνα 59. Μετάδοση ενός γεγονότος από ένα focused component σε ένα Service component



Εικόνα 60. Μετάδοση ενός γεγονότος αφού έχει φθάσει στο component scene .

9.3.4 Γεγονότα δομικών στοιχείων

9.3.4.1 Επισκόπηση

Ένα component παράγει component events για να παρέχει πληροφορίες για μια αλλαγή σε κάποια εσωτερική του κατάσταση. Δεν αναφέρονται όλες οι αλλαγές που δημιουργούνται σε εσωτερικές καταστάσεις, αλλά στις προδιαγραφές των component τύπων διευκρινίζεται ποια γεγονότα παράγονται από συγκεκριμένους τύπους component και κάτω από ποιες συνθήκες αυτά τα γεγονότα θα συμβούν.

Παράδειγμα 1:

Ένα component **Button** παράγει ένα γεγονός **OnSelect** όταν ο χρήστης πιέζει το επιλεγμένο πλήκτρο ενώ το πλήκτρο είναι ενεργό και έχει την ιδιότητα focus.

Ο σκοπός των component events είναι διπλός:

- Να παρέχει ανακοινώσεις που επιτρέπουν την επέκταση της εγγενούς συμπεριφορά ενός component με custom behaviour αντικείμενα .

- Να επιτρέπει στα parent container components να πραγματοποιούν κάποια ενέργεια ως απόκριση σε κάποιο γεγονός που παράγεται από κάποιο από τα θυγατρικά τους στοιχεία . Αυτό σημαίνει ότι ένα parent component μπορεί να συνδέσει τις συμπεριφορές δύο ή περισσότερων θυγατρικών του στοιχείων.

Παράδειγμα 2:

Όταν ο χρήστης αλλάζει το επιλεγμένο στοιχείο σε ένα **Menu** component το **Menu** παράγει ένα **OnSelect** γεγονός. Εάν ένα scene έχει ένα **Menu** και ένα **TextBox** ως θυγατρικά του στοιχεία μπορεί να συνδέσει τη συμπεριφορά αυτών των δύο components ανταποκρινόμενο στο γεγονός **Menu OnSelect** με την αλλαγή του κειμένου μέσα στο **TextBox**.

9.3.4.2 Κανόνες μετάδοσης γεγονότων δομικών στοιχείων

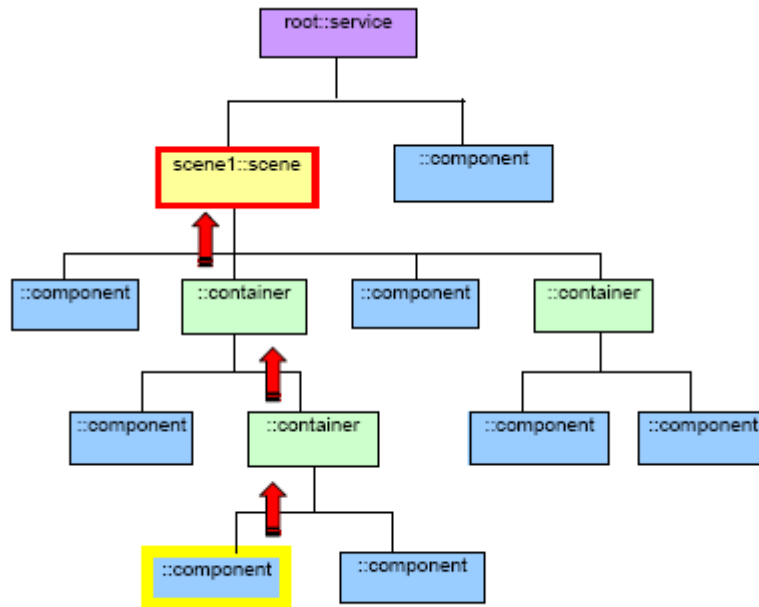
Όταν ένα component παράγει ένα γεγονός, θα πρέπει να δοθεί η ευκαιρία στους χειριστές του γεγονότος να ανταποκριθούν σε αυτό το γεγονός με τους παρακάτω κανόνες :

- 1) Ένα γεγονός θα στοχεύσει αρχικά στο component το οποίο έχει δημιουργηθεί.
- 2) Εάν το component έχει custom behaviour αντικείμενα που ενδιαφέρουν το γεγονός τότε η διάδοση του γεγονότος θα τερματιστεί και τα αντικείμενα behaviour θα ανταποκριθούν στο γεγονός .
- 3) Εάν το component δεν ενδιαφέρει το γεγονός τότε το γεγονός θα στοχεύσει στον πρόγονο του τρέχοντος component .
- 4) Τα βήματα 2 και 3 επαναλαμβάνουν μέχρι :
 - ένα component να ενδιαφέρει το γεγονός και η διάδοση ολοκληρώνεται (βήμα 2) ή
 - το component στόχος είναι το **Service** component .
- 5) Εάν το γεγονός διαδίδεται μέχρι το στοιχείο service και δεν αντιμετωπιστεί τότε απορρίπτεται.

Παράδειγμα:

Το component το οποίο παράγει ένα γεγονός είναι τονισμένο με κίτρινα σύνορα. Το parent component που ενδιαφέρει αυτό το γεγονός έχει ζωγραφιστεί με κόκκινα σύνορα. Σε αυτήν την περίπτωση είναι το scene. Το γεγονός πρώτα στοχεύει στο component το οποίο έχει δημιουργηθεί. Η συμπεριφορά που δηλώνεται μέσα σε αυτό το component δεν ενδιαφέρει το συγκεκριμένο γεγονός και έτσι το γεγονός στοχεύει στον πρόγονο του. Κανένας από τα components στη διαδρομή για το scene component δεν ενδιαφέρουν αυτό το γεγονός. Όταν το γεγονός φθάνει στο scene

component, τα αντικείμενα behaviour που έχουν δηλωθεί μέσα στο δηλώνονται μέσα στο scene ανταποκρίνονται στο γεγονός.



Εικόνα 61. Μετάδοση ενός γεγονότος από ένα component που έχει δημιουργηθεί σε ένα component Service

9.3.5 Γεγονότα Λαθών

Ένα γεγονός error είναι μια ειδίκευση ενός γεγονότος δομικού στοιχείου, και θα συμπεριφερθεί με τον ίδιο τρόπο όπως ένα component event για σκοπούς μετάδοσης και δήλωσης των trigger αντικειμένων για την λήψη τέτοιων γεγονότων.

9.3.5.1 Επίπεδα λαθών κατά του χρόνο εκτέλεσης και προκαθορισμένες αποκρίσεις.

Τα λάθη που μεταδίδονται μέσω μιας περιγραφής υπηρεσίας προσδιορίζονται σαν να βρίσκονται σε ένα από τα παρακάτω τρία επίπεδα.

- **error** - ένα γεγονός error αυτού του επιπέδου που διαδίδεται κατευθείαν στο **Service component** θεωρείται μοιραίο και προκαλεί τον τερματισμό της συνόδου υπηρεσίας.
- **warning** - ένα γεγονός error αυτού του επιπέδου που διαδίδεται κατευθείαν στο **Service component** καταγράφεται από την σύνοδο υπηρεσίας για να εξεταστεί αργότερα αλλά δεν επηρεάζει την σύνοδο υπηρεσίας.
- **notice** - ένα γεγονός error αυτού του επιπέδου που διαδίδεται κατευθείαν στο **Service component** απορρίπτεται και δεν επηρεάζει την σύνοδο υπηρεσίας .

9.3.5.2 Τύποι λαθών

Μια περιγραφή υπηρεσίας PCF μπορεί να υφίσταται :

- **reference errors**- όταν μια αναφορά λάθους δεν είναι επιλύσιμη κατά τον χρόνο εκτέλεσης
- **component errors**
- **component errors** - όταν η συνθήκη δεν μπορεί να αξιολογηθεί
- **action language errors**- αριθμητικά ή σημασιολογικά λάθη κατά τη διάρκεια της εκτέλεσης.
- Ένα λάθος αναφοράς θα οδηγήσει στην παραγωγή ενός **ReferenceError** γεγονότος .

Παράδειγμα 1:

Στο χρόνο εκτέλεσης, ένας δέκτης μπορεί να ανακαλύψει ότι ένα κομμάτι της περιγραφής υπηρεσίας PCF δεν μπορεί να βρεθεί κατά την ενεργοποίηση ενός scene που αναφέρεται σε αυτό .

Παράδειγμα 2:

Ένα action language λάθος μπορεί να εμφανιστεί εάν οι δείκτες των πινάκων είναι έξω από τα όρια, ή εάν μια μετατροπή τύπου αποτυχει ή εάν γίνει μια προσπάθεια να διαιρεθεί ένας αριθμός με το μηδέν.

9.4 Γλώσσα Δράσης

9.4.1 Εισαγωγή

Η action language επιτρέπει σε έναν συντάκτη υπηρεσιών PCF να περιγράψει μια ακολουθία ενεργειών που δημιουργούνται κατά τη διάρκεια ζωής μιας υπηρεσίας PCF, όπου αυτές οι ενέργειες ελέγχουν και τροποποιούν την τρέχουσα κατάσταση μιας διαδραστικής υπηρεσίας.

Παράδειγμα: Το ακόλουθο πλαίσιο της action language ορίζει το χρώμα γεμίσματος ενός ορθογωνίου όταν η μεταβλητή loadComplete είναι αληθής .

```
if (loadComplete == true)
statusRectangle.fillcolor = #7f7f7f;
```

Η action language έχει τα ακόλουθα χαρακτηριστικά :

- strongly typed τιμές που είναι συμβατές με τις στατικές περιγραφές υπηρεσιών PCF, και μια χαρτογράφηση μεταξύ των τιμών σε μια στατική περιγραφή υπηρεσίας και των τιμών σε μια ενεργό περιγραφή υπηρεσίας.
- μια βιβλιοθήκη συναρτήσεων για τη χρήση εκφράσεων που μπορούν να χρησιμοποιηθούν για τον χειρισμό των τιμών των τύπων δεδομένων της PCF.
- μια βιβλιοθήκη ενεργειών συστημάτων που μπορεί να κληθεί από οποιαδήποτε PCF action language .

9.4.2 Αναπαράσταση και εκτέλεση

Η Action language χρησιμοποιείται σε δύο περιπτώσεις στην PCF:

- για να καθορίσει τις ακολουθίες ενεργειών που εκτελούνται ως απόκριση σε μια μετάβαση γεγονότος ή κατάστασης .
- για να καθορίσει κάποιες συνθήκες .

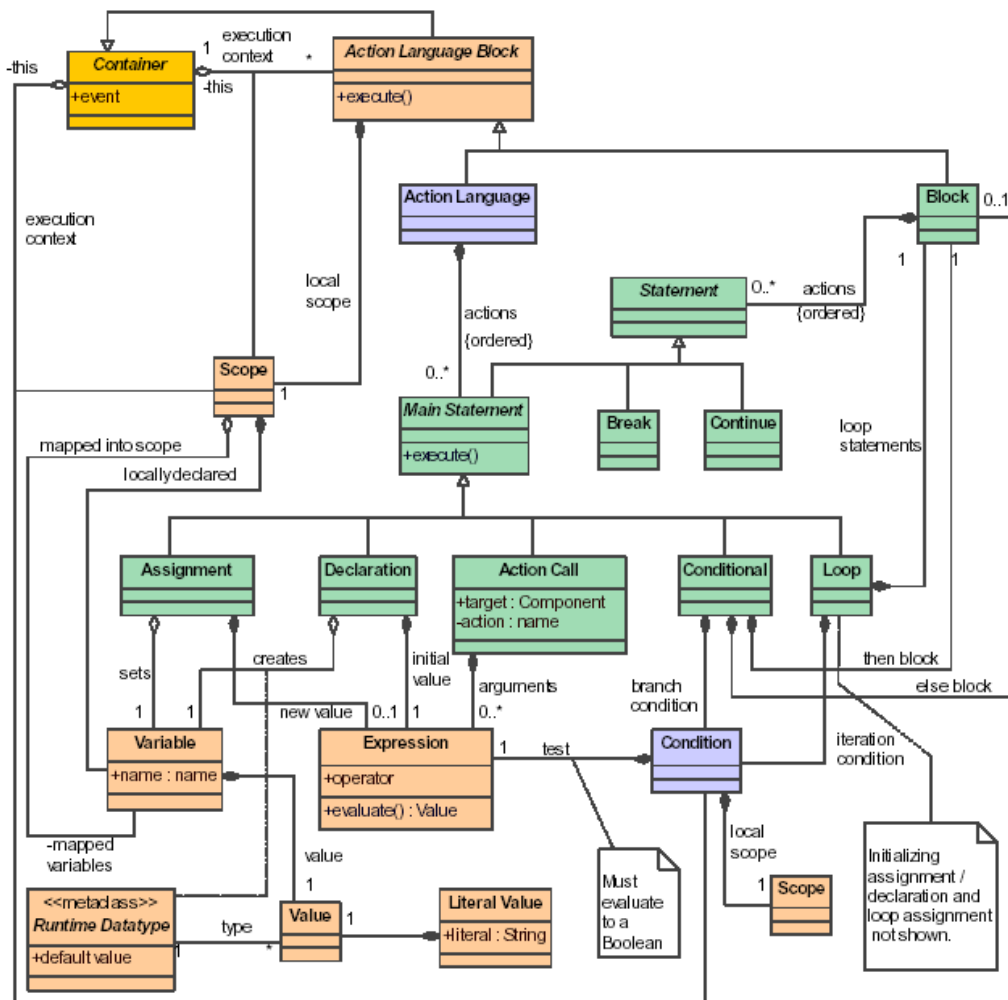


Figure 42: Action language model

Εικόνα 62. Το μοντέλο της γλώσσας δράσης

Το μοντέλο για την παρουσίαση της action language παρουσιάζεται στην εικόνα 62. Τα αντικείμενα τα οποία βρίσκονται στη κορυφή του μοντέλου είναι τα **action language** και τα αντικείμενα **condition**, τα οποία είναι γνωστά ως **action language αντικείμενα**. Αυτά περιλαμβάνονται μέσα σε ένα γενικό **container** αντικείμενο, τα οποία παρέχουν την **εκτέλεση του περιεχομένου** μιας action language περίπτωσης.

Τα αντικείμενα σε αυτό το μοντέλο περιγράφονται παρακάτω :

- Τα αντικείμενα **Action language block** παρουσιάζουν όλες τις ακολουθίες ενεργειών που μπορούν να εκτελεστούν. Μπορούν να ενεργήσουν ως container για άλλα action language blocks.
- Τα αντικείμενα **Action language** είναι ένα είδος του Action Language Block αντικειμένου και περιέχουν μια ακολουθία αντικειμένων **Statement** όπου

κάθε αντικείμενο statement θεωρείται ως μία ενέργεια που εκτελείται με την σειρά την οποία έχει δηλωθεί.

- Τα Action language block έχουν ένα πεδίο, γνωστό ως **τοπικό πεδίο** το οποίο περιέχει τις μεταβλητές ταξινομημένες μέσα στο πλαίσιο εκτέλεσης, και οποιεσδήποτε άλλες τοπικά δηλωμένες μεταβλητές.
- Τα αντικείμενα **Run-time datatype** παρουσιάζουν τους διαθέσιμους τύπους δεδομένων για τις τιμές σε οποιαδήποτε περίπτωση μιας action language .
- Τα αντικείμενα **Value** παρουσιάζουν τις τιμές που είναι μέλη των τύπων δεδομένων που εκτελούνται .
- Τα αντικείμενα **Literal value** είναι literals που παρουσιάζουν τις τιμές ενός δεδομένου τύπου δεδομένων που εκτελείται .
- Τα αντικείμενα **Variable** είναι ονομασμένες τιμές που περιλαμβάνονται στο πεδίο μιας action language περίπτωσης. Τα αντικείμενα Variable υπάρχουν σε ένα πεδίο για έναν από τους δύο λόγους:
 - Έχουν δηλωθεί τοπικά χρησιμοποιώντας μια δήλωση statement. Τα αντικείμενα τα οποία δηλώνονται με αυτό τον τρόπο είναι γνωστά ως **τοπικές μεταβλητές**
 - Έχουν χαρτογραφηθεί μέσα στο πεδίο. Τα αντικείμενα Variable που είναι διαθέσιμα μέσα σε μια action language περίπτωση από τη χαρτογράφηση είναι γνωστά ως **mapped variables**.
- Τα αντικείμενα **Expression** παρουσιάζουν μια αξιολόγηση που μπορεί να πραγματοποιηθεί ως τμήμα μιας δήλωσης ή μιας συνθήκης και επιστρέφουν μία τιμή που υπολογίζεται βάση της δομής της έκφρασης.
- Τα αντικείμενα **Condition** αποτελούνται από ένα αντικείμενο expression που αξιολογεί μία Boolean τιμή.
- Τα αντικείμενα **Block** είναι μια ακολουθία αντικειμένων statement που εκτελούνται στο τοπικό τους πεδίο.

9.4.3 Έγκυρη γλώσσα δράσης

Για να εκτελεσθούν σε ένα run-time περιβάλλον ως μέρος της εμπειρίας του χρηστή, τα αντικείμενα action language θα πρέπει να είναι έγκυρα . Αυτό σημαίνει ότι πρέπει να είναι συντακτικά και σημασιολογικά σωστά. Εάν η action language δεν είναι έγκυρη , δεν θα εκτελεσθεί και θα αναφερθεί ένα λάθος .Αυτό το λάθος είναι γνωστό ως **transcode-time error**.

9.4.4 Τύποι δεδομένων και αντικείμενα της γλώσσας δράσης

Ο τύπος δεδομένων **action language** παρουσιάζει το σύνολο όλων των πιθανών περιπτώσεων του action language μοντέλου. Στον τύπο δεδομένων action language ανήκουν τα αντικείμενα action language.

Παράδειγμα:

Τα Action language αντικείμενα παρουσιάζονται από το στοιχείο "ActionLanguage" στο PCF- XML σχήμα . Το ακόλουθο πλαίσιο XML επεξηγεί την επαναχρησιμοποίηση ενός αντικειμένου action language.

```
<ActionLanguage name="incrementScore">
    score.value++;
</ActionLanguage>

<Scene name="scoreDisplay">
    <IntegerVar name="score">
        <Integer name="value" value="0"/>
    </IntegerVar>

    <OnEvent name="select">
        <Trigger eventtype="KeyEvent">
            <UserKey name="key" value="VK_ENTER"/>
        </Trigger>
        <ActionLanguage href="#../incrementScore" context="original"/>
    </OnEvent>
</Scene>
```

9.4.5 Χαρτογράφηση δεδομένων κατά τον χρόνο εκτέλεσης

9.4.5.1 Περιεχόμενο εκτέλεσης

Τα αντικείμενα Action language δηλώνονται σε ένα rendering context το οποίο παρέχεται από το πιο στενό πρόγονο component αντικείμενο, και είναι γνωστά ως parent component αντικείμενα .

Το parent component αντικείμενο παρέχει όλες τις **mapped μεταβλητές** κατά την εκτέλεση .

Όλες οι mapped μεταβλητές για το τοπικό πεδίο ενός action language block με τις οποίες συνδέεται ένα parent component αντικείμενο θα καθοριστούν πριν από την εκτέλεση του action language block.

Παράδειγμα 1:

Το ακόλουθο XML πλαίσιο, που γράφεται σύμφωνα με το σχήμα PCF XML, επιδεικνύει πώς ένα δομικό στοιχείο μέσα σε ένα αντικείμενο collection μπορεί να παραπεμφθεί από ένα αντικείμενο action language .

```
<Scene>
  <Collection name="variables">
    <IntegerVar name="counter">
      <Integer name="value" value="1"/>
    </IntegerVar>
  </Collection>
  <OnEvent name="select">
    <Trigger eventtype="KeyEvent">
      <UserKey name="key" value="VK_ENTER"/>
    </Trigger>
    <ActionLanguage name="jumpCounter">
variables.counter.value += 10;
    </ActionLanguage>
  </OnEvent>
</Scene>
```

Παράδειγμα 2:

Το ακόλουθο XML πλαίσιο , που γράφεται σύμφωνα με το σχήμα PCF XML, επεξηγεί πως ένα δομικό στοιχείο που έχει τύπο πρόσβασης "readWrite" χαρτογραφείται μέσα σε ένα action language αντικείμενο και μπορεί να τροποποιηθεί. Επίσης γίνεται χρήση της ετικέτας XML CDATA [5] καθώς το action language παράδειγμα περιέχει ένα στοιχείο που μοιάζει με tag και δεν είναι μέρος του σχήματος PCF.

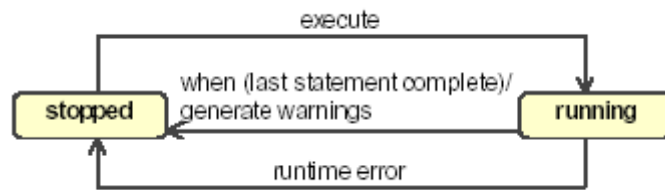
```
<Service>
  <Proportion name="serviceAspectRatio" value="4 3"/>
  <OnEvent name="aspect_shift">
    <Trigger eventtype="KeyEvent">
      <UserKey name="key" value="VK_ENTER"/>
    </Trigger>
    <ActionLanguage>
      <![CDATA[
        serviceAspectRatio = <proportion>16 9</proportion>;
      ]]>
    </ActionLanguage>
  </OnEvent>
</Service>
```

9.4.6 Μοντέλο κατά του χρόνο εκτέλεσης

9.4.6.1 Δηλώσεις

Τα Action language blocks αποτελούνται από μια ακολουθία αντικειμένων **statement** κάθε ένα από τα οποία εκτελείται σύμφωνα με την σειρά με την οποία έχει δηλωθεί, αρχίζοντας από τον πρώτο στην ακολουθία. Τα αντικείμενα Statements μέσα σε ένα action language block διαχωρίζονται με τον χαρακτήρα ";".

Τα κύρια αντικείμενα statements της PCF action language είναι assignment, declaration, action call, conditional και loop.



Εικόνα 63. Καταστάσεις του αντικειμένου action language.

Ένα αντικείμενο action language έχει δύο καταστάσεις εκτέλεσης, "stopped" και "running", όπως φαίνεται στην εικόνα 63. Όταν εκτελείται ένα action Language αντικείμενο το πρώτο statement που περιέχει θα πρέπει να εκτελεστεί άμεσα και το action language αντικείμενο θα μεταβεί στην κατάσταση "running".

Όταν το τελευταίο statement στην ακολουθία ολοκληρώσει την εκτελεσή του, το αντικείμενο action language θα μεταβεί στην κατάσταση "stopped". Όταν συμβεί ένα run-time error κατά την εκτέλεση ενός αντικειμένου action language, η εκτέλεση θα σταματήσει, και το αντικείμενο action language θα μεταβεί στην κατάσταση "stopped".

9.4.6.2 Δήλωση Ανάθεσης

Ένα **assignment** statement χρησιμοποιεί ένα αντικείμενο **variable** και, ανάλογα με τον τύπο του **assignment** statement, έχει έναν προαιρετικό αντικείμενο expression. Ορίζονται επτά τύποι του αντικειμένου **assignment** statement.

Όταν παρέχεται μία έκφραση η εκτέλεση ενός assignment statement θα ολοκληρωθεί όταν :

- η έκφραση έχει αποτιμηθεί.
- και η τιμή της μεταβλητής έχει πάρει το αποτέλεσμα της αποτίμησης, σύμφωνα με τον τύπο χορηγώντας τον τύπο του assignment statement.

Οι πέντε τύποι ενός assignment statement για τους οποίους μια έκφραση απαιτείται είναι:

- Άμεση ανάθεση ("="): Το αποτέλεσμα της αποτίμησης θα τεθεί ως ή νέα τιμή της μεταβλητής.
- Ανάθεση από προσθήκη ("+ ="): Το αποτέλεσμα της αποτίμησης θα προστεθεί στην τρέχουσα τιμή της μεταβλητής .
- Ανάθεση από αφαίρεση ("- ="): Το αποτέλεσμα της αποτίμησης θα αφαιρεθεί από την τρέχουσα τιμή της μεταβλητής .
- Ανάθεση από πολλαπλασιασμό ("* ="): Το αποτέλεσμα της αποτίμησης θα πολλαπλασιαστεί με την τιμή της μεταβλητής .
- Ανάθεση από διαίρεση("/ ="): Το αποτέλεσμα της αποτίμησης θα διαιρεθεί με την τιμή της μεταβλητής, .

9.4.6.3 Δήλωση ορισμού

Ένα declaration statement χρησιμοποιεί ένα run-time τύπο δεδομένων για την δημιουργία ενός καινούριου αντικειμένου variable και έχει μια προαιρετική έκφραση η οποία μπορεί να χρησιμοποιηθεί για την αρχικοποίηση της τιμής της μεταβλητής

Η εκτέλεση ενός declaration statement θα θεωρηθεί πλήρης όταν δημιουργηθεί μια νέα μεταβλητή μέσα τοπικό πεδίο της action language περίπτωσης που περιέχει το declaration statement. Η μεταβλητή θα έχει το όνομα που ορίζεται στο declaration statement..

Όταν παρέχεται μία προαιρετική έκφραση, αυτή θα αντιμετωπιστεί σαν ένα declaration statement το οποίο ακολουθείται από ένα assignment statement. .

Αυτό το assignment statement έχει ως μεταβλητή την πρόσφατα δημιουργημένη μεταβλητή και την declaration έκφραση. Η δήλωση θα θεωρηθεί πλήρης μόλις ολοκληρωθεί το assignment statement. Όταν δεν παρέχεται καμία έκφραση μαζί με την δήλωση στη μεταβλητή θα οριστεί η τιμή μηδέν για τους τύπους δεδομένων run-time που συνδέονται με αυτή την δήλωση .

Παράδειγμα:

Οι παρακάτω δύο action language ακολουθίες είναι ισοδύναμες :

Η πρώτη είναι μία δήλωση με μία έκφραση
integer x = 3;

Η δεύτερη είναι μία δήλωση με μία ανάθεση
integer x;
x = 3;

9.4.6.4 Action call statement

Ένα action call statement έχει δύο μορφές :

- Καλώντας ένα system-level action, γνωστό ως **system action** ,
- Χρησιμοποιώντας ένα component στόχο, που είναι μέσα στο πεδίο, καλώντας μία ενέργεια η οποία ορίζεται από τις προδιαγραφές component και ονομάζεται **component action**

Και στις δύο περιπτώσεις, το αντικείμενο action θα έχει ένα όνομα και μπορεί να έχει μια ακολουθία εκφράσεων που αξιολογούνται για την παροχή επιχειρημάτων για το αντικείμενο action .

Η εκτέλεση ενός action call statement θα έχει ολοκληρωθεί όταν:

- οποιοσδήποτε εκφράσεις που παρέχονται ως επιχειρήματα για τα αντικείμενα action έχουν αξιολογηθεί και έχουν περάσει στο αντικείμενο action .
- για τα system και component action, ένα action θα ολοκληρωθεί όταν θα εισαχθούν μία ή περισσότερες τιμές .

Ένα system action μπορεί να διακριθεί από ένα component action:

- Για τα system action αν δεν υπάρχει κανένα action του ονόματος που έχει οριστεί τότε η action language θα θεωρηθεί άκυρη .
- Για τα component action οι προδιαγραφές του component για τον τύπο του component που αναφέρεται η action language θα παρέχει ένα action του ονόματος που έχει δοθεί , ή διαφορετικά η action language θα θεωρείται άκυρη.

Παράδειγμα 1 :

Το ακόλουθο system action call statement προκαλεί την μετάβαση ενός service αντικειμένου σε ένα καινούριο scene:

```
SceneNavigate(<uri>#~/homepage</uri>, <enum>mark</enum>, nil);
```

Παράδειγμα 2:

Το ακόλουθο component action αποκρύπτει την εμφάνιση ενός **TextBox**.

```
help_text.hide();
```

9.4.6.5 Δήλωση Συνθήκης

Ένα conditional statement θα χρησιμοποιήσει την αποτίμηση μιας συνθήκης για να επιλέξει αν θα εκτελέσει ένα block statements ή όχι.

Όταν ένα block δεν εκτελείται και ένα προαιρετικό εναλλακτικό "else" block από statements παρέχεται, αυτό θα εκτελεσθεί αντ' αυτού.

Η εκτέλεση ενός conditional statement θα έχει ολοκληρωθεί όταν:

- η έκφραση της συνθήκης έχει αποτιμηθεί
- εάν η συνθήκη αποτιμηθεί ως true τότε θα πρέπει να εκτελεσθεί το "then" statement block και ολοκληρώνεται
- εάν η συνθήκη αποτιμηθεί ως false :
 - εάν παρέχεται ένα άλλο "else" block τότε θα εκτελεσθεί αυτό το statement block και ολοκληρώνεται .
 - εάν δεν υπάρχει άλλο "else" block τότε η δήλωση της συνθήκης έχει ολοκληρωθεί .

Παράδειγμα :

Το ακόλουθο action language statement παρουσιάζει μια δήλωση συνθήκης η οποία εμφανίζεται με την λέξη κλειδί "if" .

```
if ((a < b) && (b == 10)) a++;  
else { a = 0; b = 0; }
```

9.4.6.6 Δήλωση βρόχου και έλεγχος βρόχου

Ένα loop statement θα αποτελείται από:

- μια προαιρετική αρχική δήλωση
- μία απαραίτητη συνθήκη βρόχου
- ένα loop statement ή ένα block που αποτελείται από loop statements
- ένα προαιρετικό assignment βρόχου

Ένα loop block και τα blocks που είναι απόγονοι του μπορούν να περιέχουν τους ελέγχους break και continue για τις δηλώσεις των βρόχων .

Αυτοί ερμηνεύονται ως εξής:

- **break** - η εκτέλεση της δήλωσης break θα σταματήσει την εκτέλεση του loop block, θα σταματήσει την επανάληψη και θα ολοκληρωθεί ο βρόχος .
- **continue** - η εκτέλεση της δήλωσης continue θα σταματήσει την εκτέλεση του loop block, και θα πάει στο επόμενο βήμα της επανάληψης .

Παράδειγμα:

```
for ( int i = 0 ; i < 255 ; i += 4 ) {  
    if (i > maxIntensity) break;  
    integer[4] colorNums = [i, i, i, 255];  
    color fontColor = integerToColor(colorNums);  
    text3.textcolor = fontColor;  
}
```

9.4.6.7 Λάθη εκτέλεσης

Τα λάθη που συμβαίνουν κατά τον χρόνο εκτέλεσης οδηγούν σε "execution error" γεγονότα .

Τα λάθη που συμβαίνουν κατά την εκτέλεση και ανήκουν στο επίπεδο "error" είναι:

- διαίρεση με το μηδέν κατά την αποτίμηση μιας έκφρασης
- ο δείκτης του πίνακα να έχει τιμή που είναι έξω από τα όρια
- αποτυχία κλήσης μιας ενέργειας
- αποτυχία μιας συνάρτησης στη βιβλιοθήκη συναρτήσεων όπως μια συνάρτηση μετατροπής τύπων

Τα λάθη που συμβαίνουν κατά την εκτέλεση και ανήκουν στο επίπεδο "warning" είναι:

- υποχείλιση και υπερχείλιση ακέραιων αριθμών στην αποτίμηση μιας έκφρασης
- συγκεκριμένα μηνύματα προειδοποιήσεων που σχετίζονται με την πλατφόρμα

9.4.7 Εκφράσεις και συνθήκες

9.4.7.1 Αποτίμηση

Οι εκφράσεις είναι δομημένες σε ένα δέντρο με υποεκφράσεις . Οι εκφράσεις μπορούν να αξιολογηθούν ως μέρος της εκτέλεσης ενός action language statement όπου απαιτείται. Η αποτίμηση μιας έκφρασης θα είναι πλήρης όταν μία τιμή ενός γνωστού τύπου δεδομένων που εκτελείται έχει καθοριστεί χρησιμοποιώντας τους κανόνες σε αυτήν την παράγραφο .

Το περιεχόμενο των φύλλων των δέντρων μπορεί να είναι :

- literal τιμές,
- εκφράσεις δηλώσεων πινάκων
- διαδρομές παραπομπών που δεν περιλαμβάνουν δείκτες πινάκων
- διαδρομές παραπομπών που περιλαμβάνουν δείκτες πινάκων .
- το αποτέλεσμα της κλήσης μια συνάρτησης

Παράδειγμα:

: Η ακόλουθη action language παρουσιάζει τέσσερα statements που αντιστοιχούν στους τέσσερις τύπους περιεχομένων που αναφέρθηκαν παραπάνω .

```
string stringLiteral = "Example text.";
integer[3] birthday = [4/2, 10+1, 1972];
integer[3] anotherBirthday = birthday;
integer birthYear = birthday[3];
date birthDate = integerToDate(birthday);
```

9.4.7.2 Αριθμητικοί τελεστές

Οι τελεστές που παρουσιάζονται στην συνέχεια εκτελούν αριθμητικές λειτουργίες. Αυτές οι λειτουργίες ορίζονται μόνο για τους ακεραίους τύπους δεδομένων .

Ορίζονται οι παρακάτω αριθμητικοί τελεστές :

- **add (+)**- η αποτίμηση του τελεστή `add` θα είναι το αποτέλεσμα ενός ακεραίου ο οποίος θα προκύψει από το άθροισμα δύο ακεραίων τιμών.
- **subtract (-)** - η αποτίμηση του τελεστή `subtract` θα είναι το αποτέλεσμα ενός ακεραίου ο οποίος θα προκύψει από την διαφορά δύο ακεραίων τιμών.
- **multiply (*)** -η αποτίμηση του τελεστή `multiply` θα είναι το αποτέλεσμα ενός ακεραίου ο οποίος θα προκύψει από το πολλαπλασιασμό δύο ακεραίων τιμών.
- **divide (/)** -η αποτίμηση του τελεστή `divide` θα είναι ένας ακεραίος αριθμός ,το ηλίκο το οποίο θα προκύψει από την διαίρεση δύο ακεραίων τιμών.
- **modulus (%)** - η αποτίμηση του τελεστή `modulus` θα οδηγήσει σε ένα ακεραίο αριθμό ο οποίος θα είναι το υπόλοιπο της διαίρεσης δύο ακεραίων .
- **unary minus (-)** -η αποτίμηση του τελεστή `unary minus` θα είναι το αποτέλεσμα μιας ακεραίας τιμής που θα έχει αντίθετο πρόσημο από την δοσμένη τιμή .
- **increment (++)** - η αποτίμηση του τελεστή `increment` θα οδηγήσει σε μια ακεραία τιμή που θα είναι κατά μία μονάδα μεγαλύτερη από τον δοσμένη ακεραία τιμή .
- **decrement (--)** - η αποτίμηση του τελεστή `decrement` θα οδηγήσει σε μια ακεραία τιμή που θα είναι κατά μία μονάδα μικρότερη από τον δοσμένη ακεραία τιμή .

9.4.7.3 Λογικοί τελεστές

Οι παρακάτω τελεστές εκτελούν λογικές λειτουργίες. Αυτές οι λειτουργίες ορίζονται μόνο για τον `για τον` τύπο δεδομένων `boolean`.

- **and (&&)**- η αποτίμηση του τελεστή `and` θα είναι το αποτέλεσμα μιας `boolean` τιμής η οποία είναι `true` αν και μόνο αν και δύο οι `boolean` τιμές των απογόνων της είναι `true` .

- **or (||)** - η αποτίμηση του τελεστή `or` θα είναι το αποτέλεσμα μιας boolean τιμής η οποία είναι `false` αν και μόνο αν και δύο οι boolean τιμές των απογόνων της είναι `false`.
- **not (!)** - η αποτίμηση του τελεστή `not` θα είναι το αποτέλεσμα μιας boolean τιμής η οποία θα είναι `true` αν και μόνο η boolean τιμή του απογόνου είναι `false`.

9.4.7.4 Σχεσιακοί τελεστές

Οι σχεσιακοί τελεστές συγκρίνουν την τιμή δύο αντικειμένων, για να δουν εάν είναι ίσοι, άνισοι, μικρότερος ή μεγαλύτερος ο ένας από τον άλλο. Η τιμή που προκύπτει από την αποτίμηση ενός σχετικού τελεστή είναι πάντα τύπου `boolean`.

Οι ακόλουθοι σχεσιακοί τελεστές ορίζονται για όλους τους τύπους δεδομένων :

- **equals (==)** - η αποτίμηση του τελεστή `equals` θα είναι το αποτέλεσμα μιας boolean τιμής η οποία είναι `true` εάν η τιμή που είναι αριστερά από τον τελεστή είναι ίδια με την τιμή που βρίσκεται δεξιά του τελεστή και ανήκουν στον ίδιο τύπο δεδομένων.
- **not equals (!=)** - η αποτίμηση του τελεστή `not equals` θα είναι το αποτέλεσμα μιας boolean τιμής η οποία είναι `false` εάν η τιμή που είναι αριστερά από τον τελεστή είναι ίδια με την τιμή που βρίσκεται δεξιά του τελεστή και ανήκουν στον ίδιο τύπο δεδομένων.

Οι ακόλουθοι σχεσιακοί τελεστές ορίζονται μόνο για ακέραιους τύπους δεδομένων .

- **less (<)** - η αποτίμηση του τελεστή `less` θα οδηγήσει σε μια boolean τιμή η οποία είναι `true` αν και μόνο αν ο ακέραιος αριθμός που βρίσκεται αριστερά από τον τελεστή είναι μικρότερος από τον ακέραιο αριθμό που βρίσκεται δεξιά από τον τελεστή .
- **less or equal (<=)** - η αποτίμηση του τελεστή `less or equal` θα οδηγήσει σε μια boolean τιμή η οποία είναι `true` αν και μόνο αν ο ακέραιος αριθμός που βρίσκεται αριστερά από τον τελεστή είναι μικρότερος ή ίσος από τον ακέραιο αριθμό που βρίσκεται δεξιά από τον τελεστή .

- **greater (>)** - η αποτίμηση του τελεστή less or equal θα οδηγήσει σε μια boolean τιμή η οποία είναι true αν και μόνο αν ο ακέραιος αριθμός που βρίσκεται αριστερά από τον τελεστή είναι μεγαλύτερος από τον ακέραιο αριθμό που βρίσκεται δεξιά από τον τελεστή .
- **greater or equal (>=)** - η αποτίμηση του τελεστή less or equal θα οδηγήσει σε μια boolean τιμή η οποία είναι true αν και μόνο αν ο ακέραιος αριθμός που βρίσκεται αριστερά από τον τελεστή είναι μεγαλύτερος ή ίσος από τον ακέραιο αριθμό που βρίσκεται δεξιά από τον τελεστή .

9.5 Συντομεύσεις της γλώσσας δράσεις

Παράλληλα με την σύνταξη της PCF action language υπάρχει ένας αριθμός «συντομεύσεων» που παρέχουν έναν εναλλακτικό τρόπο περιγραφής ενεργειών που χρησιμοποιούνται συχνά .

9.6 Μηχανή καταστάσεων

Μία statemachine είναι μια περιγραφή συμπεριφοράς που ενθυλακώνει γεγονότα και την action language.

Οι Statemachines σχετίζονται συχνά με components αλλά μπορούν να περιγραφούν και ανεξάρτητα μέσα σε μια υπηρεσία PCF.

9.6.1 Εισαγωγή

Οι ακολουθίες των καταστάσεων και των ενεργειών που συμβαίνουν ως απόκριση σε διακριτά γεγονότα μπορούν να περιγραφούν χρησιμοποιώντας μία **statemachine**. Οι Statemachines μπορεί να διαμορφωθούν χρησιμοποιώντας ένα **statechart**, μία τεκμηριωμένη και αποδεκτή γραφική αναπαράσταση της προριζόμενης συμπεριφοράς .

Τα Statecharts παρέχουν έναν λογικό τρόπο για την περιγραφή της προοριζόμενης behaviour της εμπειρίας του χρήστη που είναι συμπληρωματική του περιεχόμενου, της πλοήγησης και της περιγραφής του σχεδιαγράμματος μιας υπηρεσίας PCF. Το μοντέλο συμπεριφοράς PCF διευκρινίζει διάφορα αντικείμενα πληροφοριών που αποτελούν μέρος του συνόλου της αρχιτεκτονικής πληροφοριών .

9.6.1.1 Ορισμός καταστάσεων

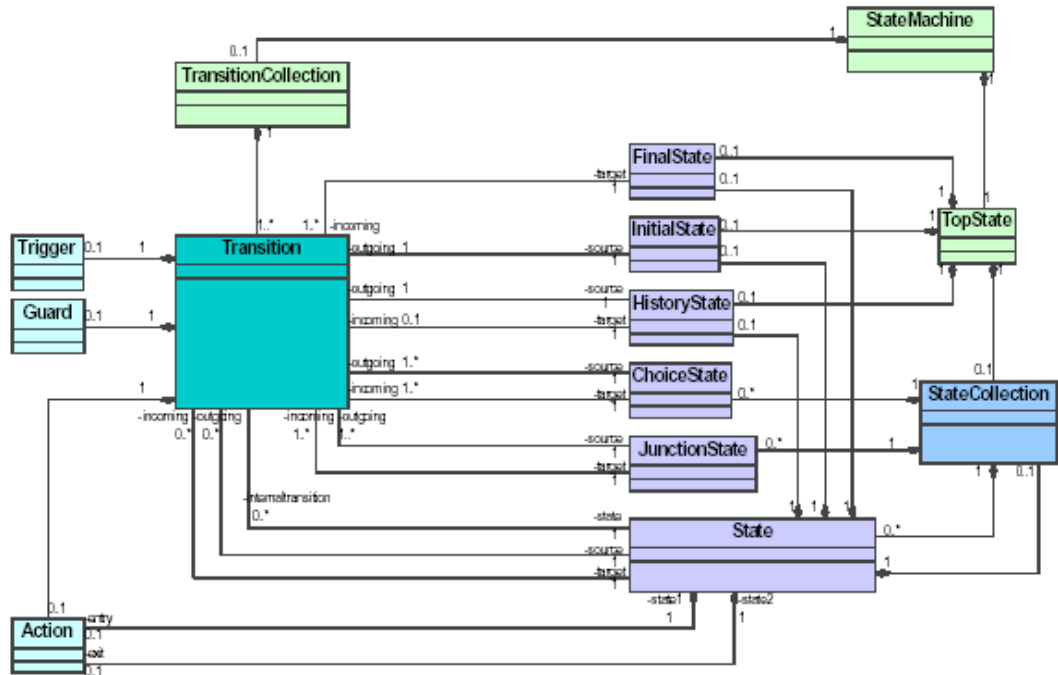
Μια κατάσταση είναι μια προσδιορισμένη περίοδος ηρεμίας κατά τη διάρκεια ζωής ενός αντικειμένου ή μιας αλληλεπίδρασης κατά τη διάρκεια της οποίας :

- ικανοποιεί κάποια συνθήκη
- εκτελεί κάποια **ενέργεια** ή
- περιμένει κάποιο **γεγονός**

Μπορούν να υπάρξουν πολλές διαφορετικές καταστάσεις συνδεδεμένες με μια συγκεκριμένη υπηρεσία. Μια αλλαγή της κατάστασης μπορεί να εμφανιστεί ως αποτέλεσμα ενός συγκεκριμένου γεγονότος. Αυτό υποδεικνύεται σε ένα statechart από ένα βέλος μετάβασης το οποίο κατευθύνεται μακριά από την κατάσταση **πηγή** προς την κατάσταση **στόχο**. Η μετάβαση περιλαμβάνει λεπτομέρειες του γεγονότος, οποιεσδήποτε περιοριστικές συνθήκες κάτω από τις οποίες η μετάβαση δεν πρέπει να πραγματοποιηθεί και οποιαδήποτε ενέργεια που πρέπει να πραγματοποιηθεί πριν η κατάσταση στόχος γίνει ενεργή .

9.6.1.3 Μοντέλο αντικειμένων

Στην εικόνα 64 φαίνεται ένα διάγραμμα το οποίο παρουσιάζει ένα PCF μοντέλο αντικειμένων για τις statemachines.



Εικόνα 64. Το μοντέλο αντικειμένων για την StateMachine

9.6.1.4 Transition and onevent object model

Η εικόνα 65 παρουσιάζει το PCF μοντέλο αντικειμένων για τα αντικείμενα transition και onEvent.

Και τα δύο προέρχονται από την αφηρημένη κατηγορία *TransitionBase*.

Το *TransitionBase* έχει:

- το πολύ ένα *Trigger*.
- το πολύ ένα *Guard*
- το πολύ ένα τμήμα της *ActionLanguage*.

Τα αντικείμενα Transition είναι τμήματα της συμπεριφοράς που προκαλεί μία αλλαγή κατάστασης από τη κατάσταση πηγή στην κατάσταση στόχο. Τα αντικείμενα OnEvent είναι περιγραφές συμπεριφοράς που δεν προκαλούν αλλαγή σε κάποια κατάσταση. Μπορούν να χρησιμοποιηθούν μέσα σε εσωτερικές μεταβάσεις ή μπορεί να χρησιμοποιηθεί σε οποιοδήποτε σημείο στην περιγραφή της PCF ως «συντόμευση» χωρίς την ανάγκη χρήσης μιας statemachine.

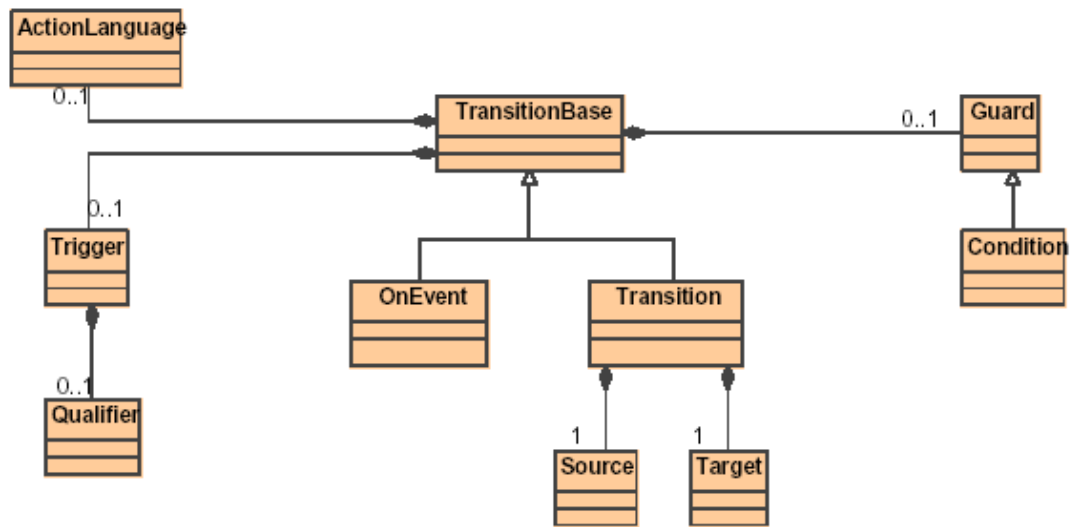


Figure 45: Transition and OnEvent object model

Εικόνα 65. Το μοντέλο των αντικειμένων transition και onevent

9.6.2 Το αντικείμενο Statemachine

Ένα αντικείμενο statemachine παρουσιάζει ακολουθίες καταστάσεων και ενεργειών που εμφανίζονται ως απόκριση σε διακριτά γεγονότα.

Οι ιδιότητες ενός αντικειμένου statemachine μπορούν να περιλαμβάνουν:

- ένα όνομα που είναι μοναδικό μέσα στο γονικό αντικείμενο στο οποίο περιέχεται.

Η περιγραφή ενός αντικείμενου `statemachine` περιέχει:

- το πολύ ένα `state transition collection` αντικείμενο .
- μόνο ένα αντικείμενο `topstate` .

Ένα `statechart` είναι μια γραφική αναπαράσταση ενός αντικειμένου `statemachine`.

Παράδειγμα:

Ένα αντικείμενο `statemachine` παρουσιάζεται στο παρακάτω πλαίσιο PCF XML χρησιμοποιώντας ένα στοιχείο με το όνομα `StateMachine`:

```
<StateMachine>
  <TopState>
    <!--state information -->
  </TopState>
  <TransitionCollection>
    <!--transition information -->
  </TransitionCollection>
</StateMachine>
```

9.6.3 Το αντικείμενο *Transition collection*

Ένα αντικείμενο `transition collection` περιέχει μία λίστα αντικειμένων `transition` τα οποία είναι διαθέσιμα στην `statemachine`.

Οι ιδιότητες ενός αντικειμένου `transition collection` μπορούν να περιλαμβάνουν:

- ένα όνομα που είναι μοναδικό μέσα στην `statemachine`.

Η περιγραφή ενός αντικειμένου `transition collection` περιέχει:

- τουλάχιστον ένα `transition` αντικείμενο .

Παράδειγμα:

```
<TransitionCollection>
  <Transition ...>
  <Transition ...>
</TransitionCollection>
```

Όλες οι μεταβάσεις μεταξύ διαφορετικών καταστάσεων πρέπει να δηλωθούν μέσα στο αντικείμενο TransitionCollection.

9.6.3.1 Το αντικείμενο Transition

Ένα αντικείμενο state transition είναι μια κατευθυνόμενη σχέση μεταξύ δύο αντικειμένων state (της κατάστασης πηγής και της κατάστασης στόχου) .

Οι ιδιότητες ενός αντικειμένου state transition περιλαμβάνουν:

- ένα προαιρετικό όνομα που θα είναι μοναδικό μέσα στην statemachine
- ακριβώς ένα αντικείμενο source state
- ακριβώς ένα αντικείμενο target state

Το αντικείμενο source state και το αντικείμενο target state θα προσδιορίσουν το όνομα του αντικειμένου state που υπάρχει μέσα στην statemachine στην οποία αναφέρονται. Το αντικείμενο transition διαμορφώνει την περιγραφή ενός εξερχόμενου αντικειμένου transition του αντικείμενο source state και ενός εισερχόμενου transition του αντικειμένου target state

Η περιγραφή ενός αντικειμένου state transition μπορεί να περιέχει :

- το πολύ ένα αντικείμενο trigger
- το πολύ ένα αντικείμενο guard
- το πολύ ένα αντικείμενο action

Το αντικείμενο guard θα υπολογιστεί πριν την εκτέλεση οποιουδήποτε αντικειμένου action. Το action θα εκτελεστεί μόνο εάν η συνθήκη guard αποτιμηθεί ως true. Όταν ένα αντικείμενο state transition παρουσιάζεται σε ένα statechart θα λάβει τη μορφή ενός βέλους όπως φαίνεται στην εικόνα 66.



Εικόνα 66. Αναπαράσταση ενός αντικειμένου transition σε ένα statechart .

Παράδειγμα:

Ένα αντικείμενο state transition παρουσιάζεται στο παρακάτω PCF- XML πλαίσιο χρησιμοποιώντας ένα στοιχείο με το όνομα Transition. Τα αντικείμενα source state και target state παρουσιάζονται από τις ιδιότητες source και target και αναφέρονται στις καταστάσεις "state1" και "state2" αντίστοιχα. Αυτή η μετάβαση είναι εξερχόμενη μετάβαση για την κατάσταση που ονομάζεται "state1" και εισερχόμενη για την κατάσταση που ονομάζεται "state2".

```
<Transition name="tran1" source="state1" target="state2">  
  <Trigger ... />  
  <Guard .../>  
  <ActionLanguage .../>  
</Transition>
```

9.6.3.2 Το αντικείμενο Trigger

Το αντικείμενο trigger δείχνει το γεγονός που προκαλεί μια μετάβαση κατάστασης.

Οι ιδιότητες ενός αντικειμένου trigger θα περιλαμβάνουν:

- ένα προαιρετικό όνομα που είναι μοναδικό μέσα στο γονικό αντικείμενο που περιλαμβάνεται .
- ένα event type από μία απαριθμημένη λίστα event types .

Η περιγραφή ενός αντικειμένου trigger μπορεί να περιέχει :

- κανένα ή περισσότερα αντικείμενα PCF για τον προσδιορισμό της τιμής του eventtype.

Παράδειγμα:

Ένα αντικείμενο trigger παρουσιάζεται στο παρακάτω PCF -XML σχήμα χρησιμοποιώντας το όνομα Trigger. Το παράδειγμα περιγράφει ένα Trigger το οποίο φιλτράρει KeyEvents, συγκεκριμένα το VK_LEFT και διαμορφώνει την τιμή της ιδιότητας "key" του γεγονότος KeyEvent.

```
<Trigger name="trigger1" eventtype="KeyEvent">
  <UserKey name="key" value="VK_LEFT"/>
</Trigger>
```

9.6.3.3 Το αντικείμενο Guard

Ένα αντικείμενο guard περιέχει μία συνθήκη. Μία μετάβαση με ένα αντικείμενο guard θα συμβεί μόνο αν αυτή η έκφραση αποτιμηθεί ως true .

Η περιγραφή ενός αντικειμένου guard θα περιέχει :

- ένα αντικείμενο Condition από το action language σχήμα της PCF.

Παράδειγμα:

Περιγραφή PCF XML για ένα αντικείμενο guard

```
<Guard>
  <Condition .../>
</Guard>
```

9.6.3.4 Το αντικείμενο Action

Ένα αντικείμενο action είναι ένα κομμάτι της γλώσσας δράσης της PCF. Το PCF XML σχήμα χρησιμοποιεί ένα μέρος της action language από το action language σχήμα .

9.6.4 Το αντικείμενο Top state

Ένα αντικείμενο top state είναι η ρίζα στην ιεραρχία των states .

Η περιγραφή ενός αντικείμενου top state μπορεί να περιέχει:

- ακριβώς ένα αντικείμενο initial state.
- το πολύ ένα αντικείμενο final state.
- το πολύ ένα αντικείμενο history state.
- ακριβώς ένα αντικείμενο state collection.

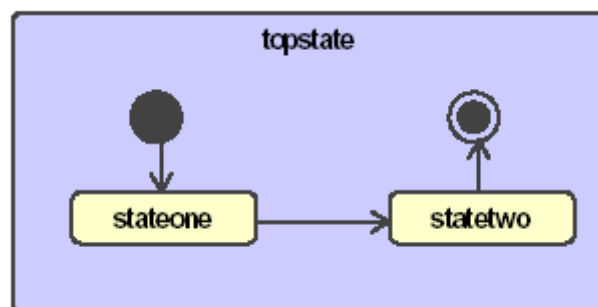
Ένα αντικείμενο top state δεν μπορεί να είναι η κατάσταση πηγής μιας μετάβασης και επομένως δεν μπορεί να περιέχει εξερχόμενες μεταβάσεις .Όταν ένα αντικείμενο top state παρουσιάζεται σε ένα statechart θα λάβει τη μορφή ενός ορθογώνιου με στρογγυλεμένες τις γωνίες του όπως παρουσιάζεται στην εικόνα 67.



Εικόνα 67. Αναπαράσταση ενός αντικειμένου top state σε ένα statechart .

Παράδειγμα 1:

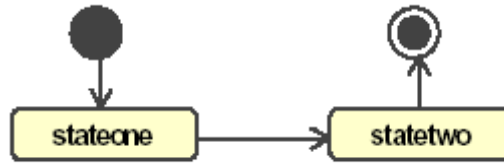
Το statechart στην εικόνα 68 παρουσιάζει ένα αντικείμενο top state το οποίο περιέχει ένα αντικείμενο αρχικής κατάστασης ένα αντικείμενο τελικής κατάστασης και δύο αντικείμενα state .



Εικόνα 68 .Το αντικείμενο top state είναι σχεδιασμένο

Παράδειγμα 2:

Η γραφική αναπαράσταση ενός top state αντικείμενου σε ένα statechart είναι προαιρετική. Το statechart που παρουσιάζεται μέσα το σχήμα είναι ισοδύναμο με αυτό στο σχήμα 48.



Εικόνα 69. Το αντικείμενο top state δεν έχει σχεδιαστεί αλλά υπονοείται η ύπαρξή του.

Παράδειγμα 3:

Ένα top state αντικείμενο παρουσιάζεται στο PCF XML χρησιμοποιώντας ένα στοιχείο με το όνομα TopState:

```
<TopState>
  <InitialState name="I_initial" .../>
  <FinalState name="F_initial" .../>
  <StateCollection name="otherstates" .../>
</TopState>
```

9.6.4.1 Το αντικείμενο αρχικής κατάστασης

Ένα αντικείμενο initial state είναι ένα source state μιας μετάβασης .

Οι ιδιότητες ενός initial state αντικείμενου μπορούν να περιλαμβάνουν :

- ένα όνομα που είναι μοναδικό μέσα στο αντικείμενο enclosing state

Ένα αντικείμενο initial state μπορεί να συνδεθεί με:

- ακριβώς ένα εξερχόμενο αντικείμενο transition όπου δεν ορίζεται κανένα αντικείμενο trigger .

Ένα αντικείμενο initial state δεν θα είναι η κατάσταση στόχος καμιάς μετάβασης και επομένως δεν θα συνδεθεί με κανένα εισερχόμενο αντικείμενο transition.

Όταν ένα αντικείμενο initial state που παρουσιάζεται σε ένα statechart θα λάβει τη μορφή ενός γεμισμένου κύκλου όπως φαίνεται στο σχήμα



Εικόνα 70. Αναπαράσταση ενός αντικειμένου initial state σε ένα statechart.

Παράδειγμα:

Ένα initial state αντικείμενο παρουσιάζεται στο παρακάτω PCF XML πλαίσιο χρησιμοποιώντας ένα στοιχείο με το όνομα InitialState:

```
<InitialState name="initial"/>
```

9.6.4.2 Το αντικείμενο τελικής κατάστασης

Ένα final state δείχνει ότι το αντικείμενο enclosing state έχει ολοκληρωθεί.

Οι ιδιότητες ενός final state αντικειμένου μπορεί να περιλαμβάνουν:

- ένα όνομα που είναι μοναδικό μέσα στο αντικείμενο enclosing state .

Ένα final state αντικείμενο μπορεί να συνδεθεί με :

- τουλάχιστον ένα εισερχόμενο αντικείμενο transition

Ένα αντικείμενο final state δεν θα είναι η κατάσταση πηγή οποιασδήποτε μετάβασης και, επομένως, δεν θα συνδεθεί με κανένα εξερχόμενο αντικείμενο transition .

Όταν ένα αντικείμενο final state παρουσιάζεται σε ένα statechart θα λάβει τη παρακάτω μορφή .



Εικόνα 71. Αναπαράσταση ενός αντικειμένου final state σε ένα statechart.

Παράδειγμα:

Ένα αντικείμενο final state παρουσιάζεται στο παρακάτω PCF XML πλαίσιο χρησιμοποιώντας ένα στοιχείο με το όνομα FinalState.

```
<FinalState name="final_2"/>
```

9.6.4.3 Το αντικείμενο History state

Ένα αντικείμενο history state μπορεί να χρησιμοποιηθεί αντί, ή από κοινού με το αντικείμενο initial state για να δείξει "enter-by-history". Η κατάσταση πηγή ενός αντικειμένου transition είναι αυτή που δείχνει το προκαθορισμένο εσωτερικό state αντικείμενο ενός enclosing state αντικειμένου .

Συγκεκριμένα, επιτρέπει στο προκαθορισμένο state αντικείμενο να γίνει το πιο πρόσφατα επισκέψιμο child state.

Οι ιδιότητες ενός history state αντικειμένου μπορεί να περιλαμβάνουν:

- ένα όνομα που είναι μοναδικό μέσα στο enclosing state αντικείμενο .

Ένα history state αντικείμενο :

- θα συνδεθεί με ακριβώς ένα εξερχόμενο transition αντικείμενο όπου δεν ορίζεται κάποιο αντικείμενο trigger.
- πρέπει να συνδεθείτε με τουλάχιστον ένα εισερχόμενο transition αντικείμενο.

Όταν ένα history state αντικείμενο παρουσιάζεται σε ένα statechart θα λάβει τη παρακάτω μορφή .



Εικόνα 72 . Αναπαράσταση ενός αντικειμένου history state σε ένα statechart.

Την πρώτη φορά που ένα enclosing state επισκέπτεται ένα θυγατρικό state το οποίο γίνεται ενεργό θα είναι το "default state".

Εάν η μετάβαση τερματισθεί στο history state τότε η default state υποδεικνύεται από την εξερχόμενη μετάβαση από το history state. Μια μετάβαση που ολοκληρώνεται στην άκρη του enclosing state είναι ισοδύναμη με μια μετάβαση που ολοκληρώνεται στο initial state.

Οι επόμενες επισκέψεις στο enclosing state μέσω του history state θα έχουν ως αποτέλεσμα το πιο πρόσφατα επισκέψιμο θυγατρικό state να γίνει το ενεργό state .

Εάν το πρόσφατα επισκέψιμο θυγατρικό state ήταν το final state τότε το ενεργό θυγατρικό θα είναι το default state.

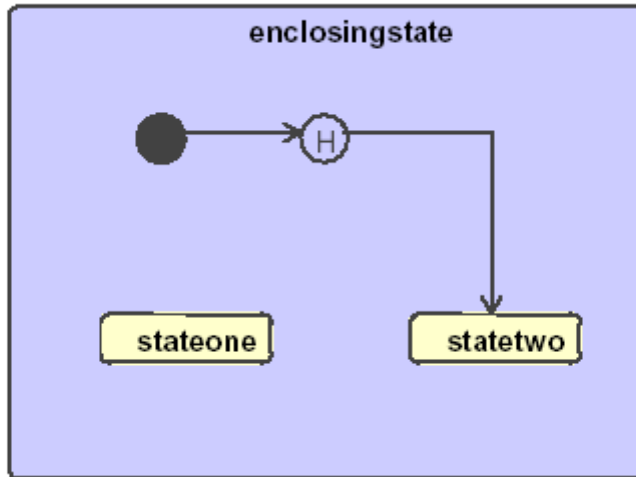
Οι επόμενες επισκέψεις στο enclosing state μέσω του initial state θα έχουν ως αποτέλεσμα το default state να υποδεικνύει το active state.

Εάν δεν υπάρχει κανένα initial state τότε όλες οι μεταβάσεις στο enclosing state θα τερματιστούν από το history state.

Παράδειγμα 1:

Ένα history state αντικείμενο παρουσιάζεται στο PCF XML πλαίσιο χρησιμοποιώντας ένα στοιχείο με το όνομα HistoryState. Αυτό το παράδειγμα παρουσιάζει μια PCF XML περιγραφή για το history state αντικείμενο που φαίνεται στο σχήμα

```
<HistoryState name="example1"/>
```

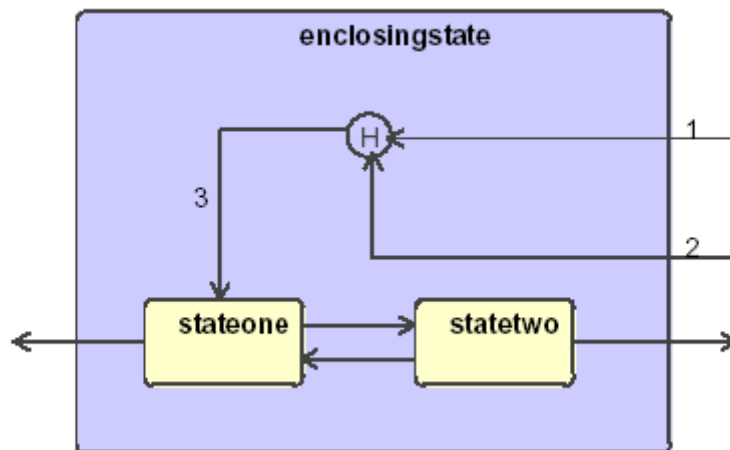


Εικόνα 73. Όλες οι επισκέψεις στο αντικείμενο enclosingstate εκμεταλλεύονται την συμπεριφορά του αντικειμένου history state.

Παράδειγμα 2:

Όταν υπάρχει ένα history state χωρίς ένα συνοδευτικό initial state τότε το enclosing state θα ενεργοποιηθεί μέσω των μεταβάσεων που κατευθύνονται στο history state του.

Το history state στην εικόνα 74 δεν συνοδεύεται από ένα initial state. Οι μεταβάσεις 1 και 2 κατευθύνονται στο history state. Η εξερχόμενη μετάβαση από το history state, η μετάβαση 3, δείχνει ότι το "stateone" είναι το θυγατρικό default state του "enclosingstate". Αυτό θα είναι το ενεργό θυγατρικό state την πρώτη φορά που γίνεται κάποια επίσκεψη στο "enclosingstate". Οι επόμενες επισκέψεις "στο enclosingstate" οδηγούν στο πιο πρόσφατα επισκέψιμο θυγατρικό state που γίνεται ενεργό.

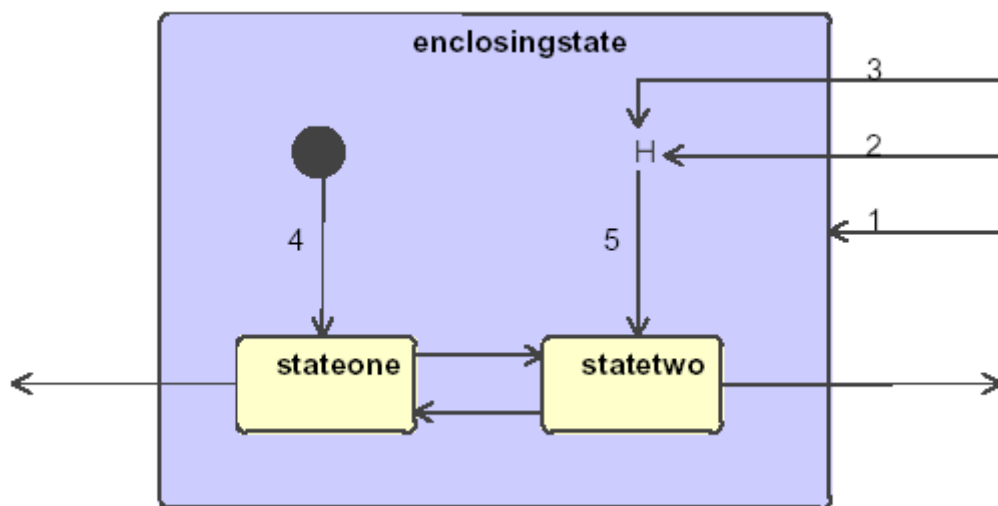


Εικόνα 74. Το αντικείμενο history state μπορεί να χρησιμοποιηθεί χωρίς το συνοδευτικό αντικείμενο initial state.

Παράδειγμα 3:

Το σχήμα 55 παρουσιάζει ένα history state που συνδέεται με ένα initial state.

Εάν η πρώτη επίσκεψη "στο enclosingstate" είναι μέσω της μετάβασης 1 τότε το ενεργό θυγατρικό state θα καθοριστεί από την εξωτερική μετάβαση από το initial state , την μετάβαση 4. Το ενεργό θυγατρικό state είναι το "stateone". Εάν η πρώτη επίσκεψη είναι μέσω της μετάβασης 2 ή 3 τότε το ενεργό θυγατρικό state θα καθοριστεί από την εξερχόμενη μετάβαση από το history state , την μετάβαση 5. Το ενεργό θυγατρικό state είναι το "statetwo". Οι επόμενες επισκέψεις "στο enclosingstate" μέσω της μετάβασης 1 στο initial state θα οδηγήσουν "στο stateone" το οποίο θα γίνει ενεργό . Οι επόμενες επισκέψεις μέσω της μετάβασης 2 ή 3 στο history state θα οδηγήσουν στο πιο πρόσφατα επισκέψιμο θυγατρικό state που γίνεται ενεργό .



Εικόνα 75. Τα αντικείμενα history και initial state έχουν και τα δύο μία εξωτερική μετάβαση.

9.6.5 Το αντικείμενο State collection

Ένα αντικείμενο state collection χρησιμοποιείται για την ομαδοποίηση των αντικειμένων state. Αυτό μπορεί να περιλαμβάνει αντικείμενα state , junction state και choice αλλά αποκλείονται τα αντικείμενα initial state, final state και history.

Οι ιδιότητες ενός state collection αντικειμένου μπορεί να περιλαμβάνουν :

- ένα όνομα που είναι μοναδικό μέσα στο enclosing state αντικείμενο

Η περιγραφή ενός αντικειμένου state collection μπορεί να περιέχει :

- κανένα ή πολλά αντικείμενα state
- κανένα ή πολλά αντικείμενα choice state
- κανένα ή πολλά αντικείμενα junction state

Ένα αντικείμενο state collection δεν έχει γραφική αναπαράσταση σε ένα statechart.

Παράδειγμα:

Ένα state collection αντικείμενο παρουσιάζεται στο PCF XML πλαίσιο χρησιμοποιώντας ένα στοιχείο με το όνομα StateCollection. Αυτό το παράδειγμα παρουσιάζει μια PCF XML περιγραφή ενός αντικειμένου state collection το οποίο περιέχει δύο αντικείμενα state ,ένα initial state αντικείμενο και ένα final state αντικείμενο .

```
<StateCollection>
  <State name="stateA" ... />
  <State name="stateB" ... />
  <JunctionState name="junctionA" .../>
  <ChoiceState name="choiceA" .../>
  <ChoiceState name="choiceB" .../>
</StateCollection>
```

9.6.5.1 Το αντικείμενο state .

Οι ιδιότητες ενός αντικειμένου state μπορεί να περιλαμβάνουν:

- ένα όνομα που είναι μοναδικό μέσα στο enclosing state αντικείμενο

Η περιγραφή ενός αντικειμένου state μπορεί να περιέχει :

- το πολύ ένα αντικείμενο state entry action
- το πολύ ένα αντικείμενο state exit action
- κανένα ή περισσότερα αντικείμενα internal state transition .
- το πολύ ένα αντικείμενο initial state
- το πολύ ένα αντικείμενο final state
- το πολύ ένα αντικείμενο history state
- το πολύ ένα αντικείμενο state collection

Ένα αντικείμενο state μπορεί να συνδεθεί με:

- κανένα ή περισσότερα εισερχόμενα αντικείμενα transition
- κανένα ή περισσότερα εξερχόμενα αντικείμενα transition κάθε ένα το οποίο έχει καθορισμένο ένα αντικείμενο trigger .

Όταν ένα state αντικείμενο παρουσιάζεται σε ένα statechart θα λάβει τη παρακάτω μορφή .

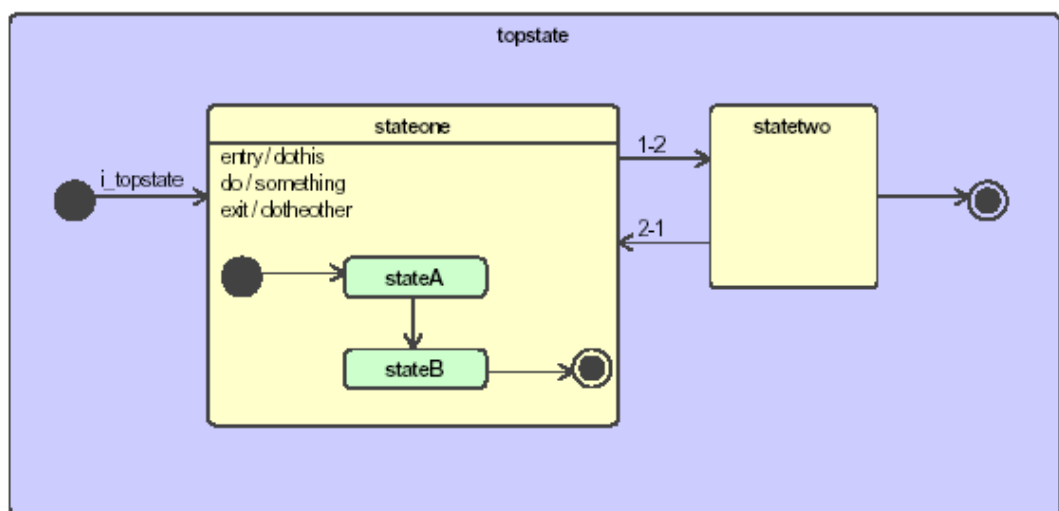


Εικόνα 76. Αναπαράσταση ενός αντικειμένου state σε ένα statechart.

Παράδειγμα:

Ένα αντικείμενο state παρουσιάζεται στο παρακάτω PCF -XML σχήμα χρησιμοποιώντας ένα στοιχείο με το ονόμα State . Αυτό το παράδειγμα παρουσιάζει την PCF XML περιγραφή ενός αντικειμένου state ,του "stateone" που παρουσιάζεται στην εικόνα 77 .

```
<State name="stateone">
  <StateEntry>
    <ActionLanguage> <!-- dothis --> </ActionLanguage>
  </StateEntry>
  <StateExit>
    <ActionLanguage> <!-- dotheother --> </ActionLanguage>
  </StateExit>
  <InternalTransitions>
    <OnEvent>
      <Trigger eventtype="KeyEvent"></Trigger>
      <ActionLanguage> <!-- something --> </ActionLanguage>
    </OnEvent>
  </InternalTransitions>
  <InitialState name="stateone_i"/>
  <FinalState name="stateone_f"/>
  <StateCollection>
    <State name="stateA"/>
    <State name="stateB"/>
  </StateCollection>
</State>
```



Εικόνα 77. Η περιγραφή του αντικειμένου «stateone» περιλαμβάνει μεταβάσεις και θυγατρικά αντικείμενα state.

9.6.5.1.1 Το αντικείμενο κατάστασης εισόδου .

Ένα αντικείμενο state entry περιέχει ένα μέρος της action language που θα ενεργοποιηθεί κατά την είσοδο στο αντικείμενο enclosing state .

Η περιγραφή ενός αντικειμένου state entry :

- περιέχει ακριβώς ένα αντικείμενο action .

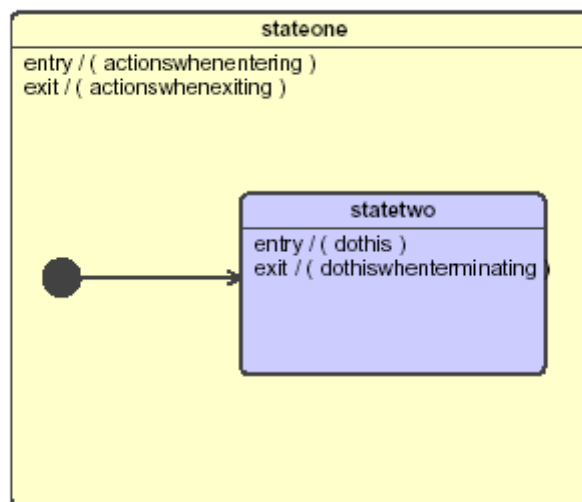
Παράδειγμα 1:

Ένα αντικείμενο state entry παρουσιάζεται στο PCF XML σχήμα χρησιμοποιώντας ένα στοιχείο με το όνομα StateEntry.

```
<StateEntry>  
  <ActionLanguage ... />  
</StateEntry>
```

Παράδειγμα 2:

Στην εικόνα 78 τα αντικείμενα entry actions τα οποία συνδέονται με το stateone και πραγματοποιούνται πριν από εκείνα που συνδέονται με θυγατρικό ενεργο initial state αντικείμενο statetwo .



Εικόνα 78. Φωλιασμένα αντικείμενα states με καταστάσεις εισόδου και εξόδου.

9.6.5.1.2 Το αντικείμενο κατάστασης εξόδου

Ένα αντικείμενο state exit περιέχει ένα μέρος της action language που θα ενεργοποιηθεί κατά την έξοδο από το αντικείμενο enclosing state .

Η περιγραφή ενός αντικειμένου state exit :

- περιέχει ακριβώς ένα αντικείμενο action .

Παράδειγμα 1:

Ένα αντικείμενο state exit παρουσιάζεται στο παρακάτω PCF XML σχήμα χρησιμοποιώντας ένα στοιχείο με το όνομα StateExit

.

```
<StateExit>  
    <ActionLanguage ... />  
</StateExit>
```

Παράδειγμα 2:

Στην εικόνα 78, όταν το stateone τερματίζεται τα αντικείμενα exit actions τα οποία συνδέονται με το statetwo πραγματοποιούνται πριν από εκείνα που συνδέονται με stateone.

9.6.5.1.3 Το αντικείμενο εσωτερικών καταστάσεων

Ένα αντικείμενο internal transition δεν προκαλεί κάποια αλλαγή σε ένα state .

Οι ιδιότητες ενός αντικειμένου internal transition μπορεί να περιλαμβάνουν :

- ένα όνομα που είναι μοναδικό μέσα στο enclosing state αντικείμενο

Η περιγραφή ενός internal transition αντικείμενου :

- περιέχει ένα αντικείμενο trigger
- μπορεί να περιέχει ένα αντικείμενο guard
- μπορεί να περιέχει ένα αντικείμενο action

Παράδειγμα:

```
<State name="this state">
  <Internal Transitions>
    <OnEvent name="pressselect" source="thisstate" target="thisstate">
      <Trigger eventtype="KeyEvent">
        <UserKey name="key" value="VK_ENTER"/>
      </Trigger>
      <ActionLanguage> <!--actions _ </ActionLanguage>
    </OnEvent>
  </InternalTransitions>
</State>
```

9.6.5.2 Το αντικείμενο διακλάδωσης καταστάσεων

Ένα αντικείμενο junction state θα χρησιμοποιηθεί για να ενώσει ένα ή περισσότερα εισερχόμενα transitions σε ένα ή περισσότερα εξερχόμενα transitions. Συγκεκριμένα, θα χρησιμοποιηθεί όταν δεν συνδέεται κανένα αντικείμενο action με κάποιο εισερχόμενο transitions .

Οι ιδιότητες ενός junction state αντικείμενου μπορεί να περιλαμβάνουν:

- ένα όνομα που είναι μοναδικό μέσα στο enclosing state αντικείμενο

Ένα αντικείμενο junction state μπορεί να θα συνδεθεί με:

- τουλάχιστον ένα εισερχόμενο αντικείμενο transition
- τουλάχιστον ένα εξερχόμενο αντικείμενο transition

Η περιγραφή των εισερχόμενων αντικειμένων transition που συνδέονται με ένα junction state αντικείμενο :

- μπορεί να περιέχει ένα αντικείμενο trigger
- μπορεί να περιέχει ένα αντικείμενο guard
- δεν περιέχει κανένα αντικείμενο action

Η περιγραφή των εξερχόμενων αντικειμένων transition που συνδέονται με ένα junction state αντικείμενο :

- δεν περιέχει κανένα αντικείμενο trigger
- μπορεί να περιέχει ένα αντικείμενο guard
- μπορεί να περιέχει ένα αντικείμενο action

Όταν ένα αντικείμενο state παρουσιάζεται σε ένα statechart θα λάβει τη παρακάτω μορφή .



Εικόνα 79. Αναπαράσταση ενός αντικειμένου junction state σε ένα statechart.

Η αναπαράσταση ενός αντικειμένου junction state σε ένα statechart είναι γραφική συντόμευση για πολλαπλά state transitions τα οποία μοιράζονται ορισμένους guard conditions.

Το statechart στο σχήμα 80 χρησιμοποιεί ένα junction state αντικείμενο και είναι ισοδύναμο με το statechart στο σχήμα 81 που χρησιμοποιεί πολλαπλά transitions .

Παράδειγμα:

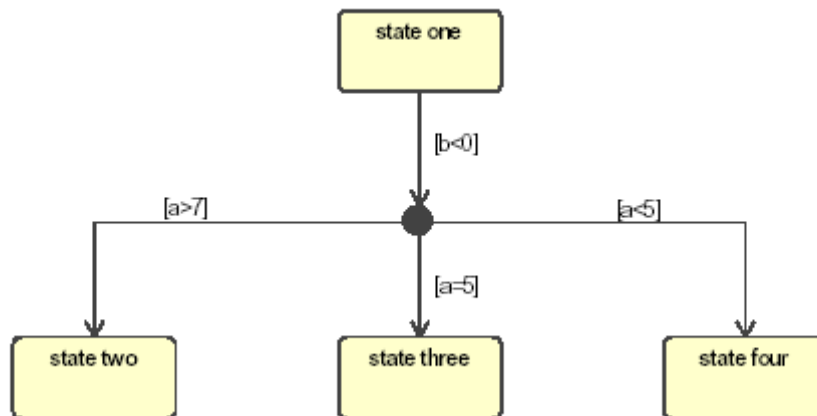
Ένα αντικείμενο junction state παρουσιάζεται στο PCF XML σχήμα χρησιμοποιώντας ένα στοιχείο με το όνομα JunctionState.

Αρχικά γίνεται αποτίμηση της guard condition[$b < 0$] . Εάν είναι true τότε αποτιμούνται τα guards αντικείμενα στα εξωτερικά transitions . Η συνθήκη που αποτιμάται ως true καθορίζει το νέο state . Εάν [$a = 6$) τότε δεν υπάρχει καμία μετάβαση από το state one

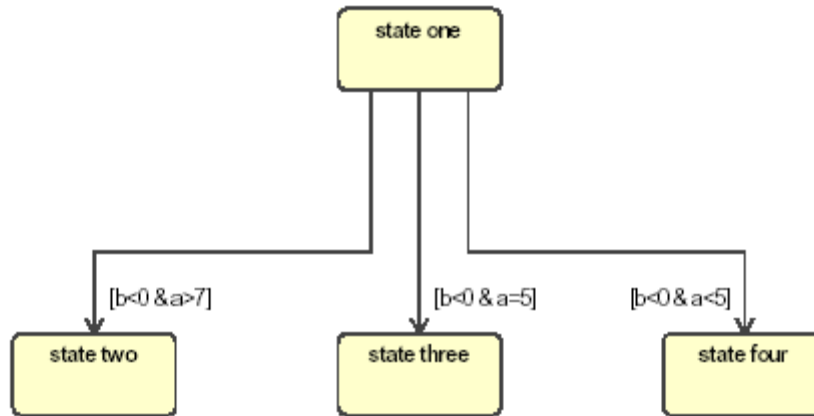
```

<JunctionState name="junction"/>
...
<TransitionCollection>
  <Transition name="in1" source="stateone" target="junction">
    <Guard <!-- b<0 --> />
    <Trigger .../>
    <!--no action for incoming transition-->
  </Transition>
  <Transition name="out1" source="junction" target="statetwo">
    <Guard <!--a>7 --> />
    <!--no trigger for outgoing transition -->
  </Transition>
  <Transition name="out2" source="junction" target="statethree">
    <Guard <!--a=5 --> />
    <!--no trigger for outgoing transition -->
  </Transition>
  <Transition name="out3" source="junction" target="statefour">
    <Guard <!-- a<5 --> />
    <!--no trigger for outgoing transition -->
  </Transition>
</TransitionCollection>

```



Εικόνα 80. αντικείμενο junction state χρησιμοποιείται για να απλοποιήσει τις πολλαπλές μεταβάσεις υπό συνθήκες.



Εικόνα 81. Πολλαπλές μεταβάσεις υπό συνθήκες .

9.6.5.3 Το αντικείμενο επιλογής κατάστασης

Ένα αντικείμενο choice state θα χρησιμοποιηθεί για να ενώσει ένα ή περισσότερα εισερχόμενα transitions με ένα ή περισσότερα εξερχόμενα transitions .

Συγκεκριμένα, θα χρησιμοποιηθεί όταν ένα αντικείμενο action συνδεθεί με ένα ή περισσότερα εισερχόμενα transitions .

Οι ιδιότητες ενός αντικειμένου choice state μπορεί να περιλαμβάνουν :

- ένα όνομα που είναι μοναδικό μέσα στο enclosing state αντικείμενο

Ένα αντικείμενο choice state μπορεί να θα συνδεθεί με:

- τουλάχιστον ένα εισερχόμενο αντικείμενο transition
- τουλάχιστον ένα εξερχόμενο αντικείμενο transition

Η περιγραφή των εισερχόμενων αντικειμένων transition που συνδέονται με ένα choice state αντικείμενο :

- μπορεί να περιέχει ένα αντικείμενο trigger
- μπορεί να περιέχει ένα αντικείμενο guard
- πρέπει να περιέχει ένα αντικείμενο action

Η περιγραφή των εξερχόμενων αντικειμένων transition που συνδέονται με ένα choice state αντικείμενο :

- δεν περιέχει κανένα αντικείμενο trigger.
- μπορεί να περιέχει ένα αντικείμενο guard.
- μπορεί να περιέχει ένα αντικείμενο action.

Όταν ένα state αντικείμενο παρουσιάζεται σε ένα statechart θα λάβει τη παρακάτω μορφή.

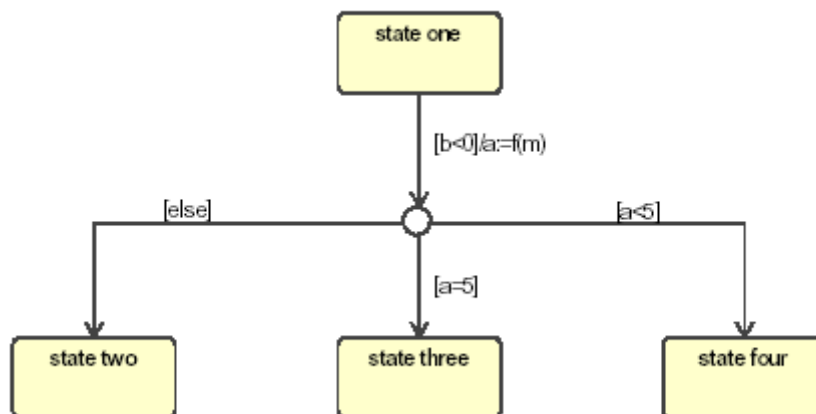


Εικόνα 82. Αναπαράσταση ενός αντικειμένου junction state σε ένα statechart.

Παράδειγμα 1:

Ένα αντικείμενο choice state παρουσιάζεται στο παρακάτω PCF XML πλαίσιο χρησιμοποιεί ένα στοιχείο με το όνομα ChoiceState. Αυτό το παράδειγμα παρουσιάζει μια PCF XML περιγραφή ενός choice state αντικειμένου το οποίο φαίνεται στο σχήμα 63 και ενώνει ένα εισερχόμενο transition με τρία εξερχόμενα transitions. Ένα αντικείμενο guard συνδέεται με κάθε transition. Ένα αντικείμενο action συνδέεται με το εισερχόμενο transition. Αρχικά γίνεται αποτίμηση της guard condition $[b < 0]$. Εάν είναι true τότε εκτελείται το αντικείμενο action $a := f(m)$. Μόνο μετά την εκτέλεση του αντικειμένου action γίνεται αποτίμηση του αντικειμένου guard στο εξερχόμενο transition. Η συνθήκη else διασφαλίζει όλες τις πιθανές τιμές του "a".

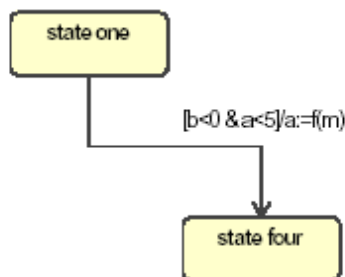
```
<ChoiceState name="choice"/>
...
<TransitionCollection>
  <Transition name="in1" source="stateone" target="choice">
    <Guard ... <!-- b<0 --> />
    <Trigger .../>
    <Action ...<!-- a:=f(m) --> />
  </Transition>
  <Transition name="out1" source="choice" target="statetwo">
    <Guard ... <!-- else --> />
  </Transition>
  <Transition name="out2" source="choice" target="statethree">
    <Guard ... <!-- a=5 --> />
  </Transition>
  <Transition name="out3" source="choice" target="statefour">
    <Guard ... <!-- a<5 --> />
  </Transition>
</TransitionCollection>
```



Εικόνα 83. Το αντικείμενο choice state

Παράδειγμα 2:

Η μετάβαση από το state one στο state four στο σχήμα 84 δεν είναι ισοδύναμη με την μετάβαση από το state one στο state four μέσω του αντικειμένου choice state στο σχήμα 83 .



Εικόνα 84. Μία μοναδική μετάβαση δεν μπορεί να αντικαταστήσει τις μεταβάσεις διαμέσου του αντικειμένου choice state.

9.7 Το αντικείμενο OnEvent – συντόμευση της statemachine .

Παράλληλα με τη statemachine της PCF υπάρχει μία συντόμευση η οποία παρέχει ένα εναλλακτικό τρόπο περιγραφής συμπεριφοράς που περιλαμβάνει τα αντικείμενα trigger, guard και action language. Το αντικείμενο onEvent είναι συνώνυμο με το εσωτερικό transition που περιγράφεται στην παράγραφο 9.6.5.1.3, και είναι κατάλληλο όταν δεν συμβαίνει κάποια αλλαγή σε κάποιο state .

Τα παρακάτω παραδείγματα δείχνουν πως μπορεί να χρησιμοποιηθεί ένα αντικείμενο OnEvent .

Παράδειγμα 1:

Όταν το αντικείμενο "scene1" είναι ενεργό και ένα VK_ENTER keypress παραλαμβάνεται, το scene πλοηγεί στο αντικείμενο "next_scene".

```
<Scene name="scene1">
  <!--visual components etc -->
  <Button name="button1">
    <!--Button properties -->
  </Button>
  <Button name="button2">
    <!--Button properties -->
  </Button>
  <OnEvent name="pressselect">
    <Trigger eventtype="KeyEvent">
      <UserKey name="key" value="VK_ENTER"/>
    </Trigger>
    <SceneNavigate>
      <URI name="target" value="#../next_scene"/>
    </SceneNavigate>
  </OnEvent>
</Scene>
```

Παράδειγμα 2:

Όταν το "scene2" είναι ενεργό, το scene πλοηγεί στο "next_scene" όταν επιλέγεται είτε το "button1" είτε το "button2"

```
<Scene name="scene2">
  <!--visual components etc -->
  <Button name="button1">
    <!--Button properties -->
  </Button>
  <Button name="button2">
    <!--Button properties -->
  </Button>
  <OnEvent name="pressselect">
    <Trigger eventtype="OnSelect"/>
    <SceneNavigate>
      <URI name="target" value="#../next_scene"/>
    </SceneNavigate>
  </OnEvent>
</Scene>
```

Παράδειγμα 3:

Όταν το "scene3" είναι ενεργό και επιλέγεται το button1 , το scene πλοηγεί στο "next_scene".

```
<Scene name="scene3">
  <!--visual components etc -->
  <Button name="button1">
    <!--Button properties -->
    <OnEvent name="pressselect">
      <Trigger eventtype="OnSelect"/>
      <SceneNavigate>
        <URI name="target" value="#../next_scene"/>
      </SceneNavigate>
    </OnEvent>
  </Button>
  <Button name="I_do_nothing">
    <!--Button properties -->
  </Button>
</Scene>
```

9.8 Συμπεριφορά που ορίζεται από τον χρήστη.

9.8.1 Κανόνες για την συμπεριφορά που ορίζεται από τον χρήστη.

Τα αντικείμενα User-defined behaviour μπορούν να εισαχθούν σε οποιοδήποτε επίπεδο στην ιεραρχία των component χρησιμοποιώντας statemachines ή events.

Τα αντικείμενα User-defined behaviour μπορούν :

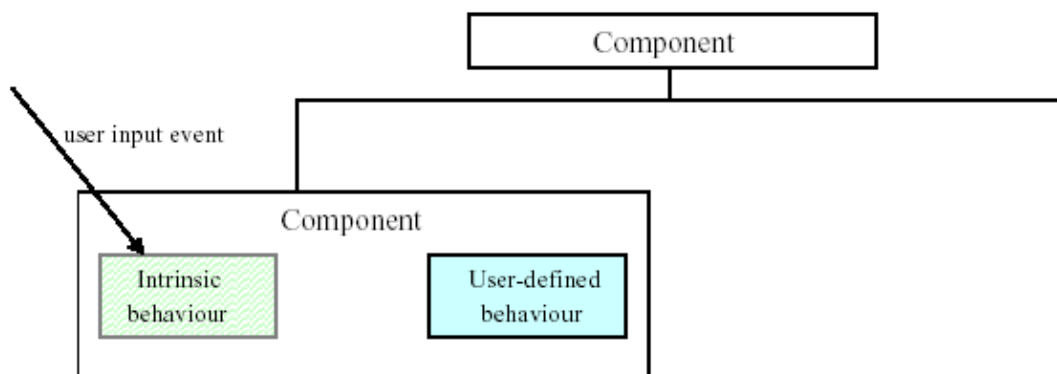
- να επεκτείνουν οποιαδήποτε εγγενή συμπεριφορά, αλλά δεν μπορούν να την αγνοήσουν .
- να υπακούσουν στους ίδιους scoring κανόνες με τα components της PCF

9.8.2 Μετάδοση γεγονότος κατά την οποία εμπλέκεται συμπεριφορά που ορίζεται από τον χρήστη.

Η μετάδοση ενός γεγονότος πραγματοποιείται όπως περιγράφεται στην πρόταση 9.3. Όταν ένα component έχει εγγενή και user-defined συμπεριφορά ένα event θα στοχεύσει αρχικά στην εγγενή συμπεριφορά. Μόνο εάν η εγγενής συμπεριφορά δεν αποκρίνεται το αντικείμενο event θα στοχεύσει στην user-defined συμπεριφορά .

Παράδειγμα 1:

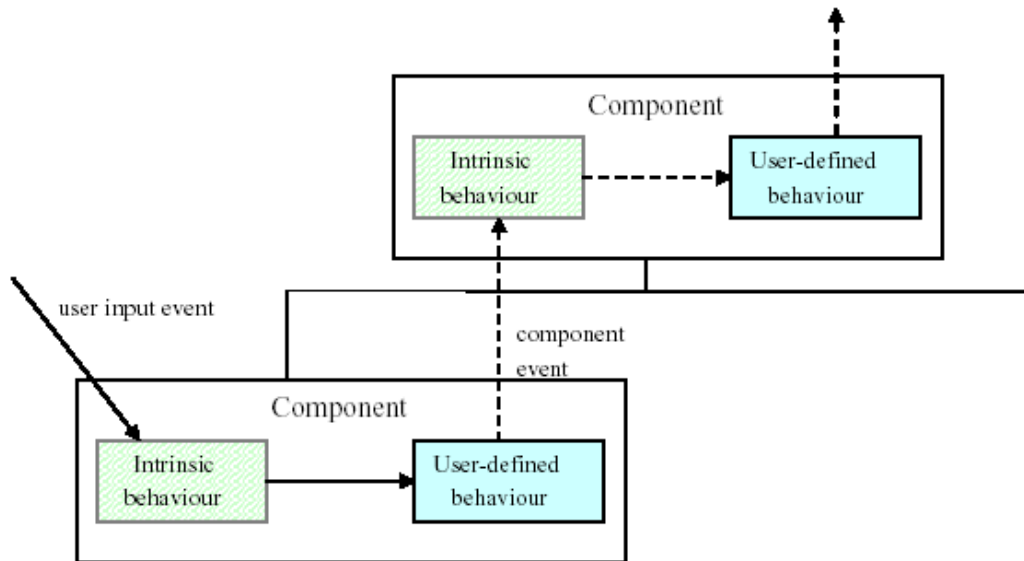
Το σχήμα 65 παρουσιάζει την εγγενή συμπεριφορά ενός focused component το οποίο αποκρίνεται σε ένα γεγονός που εισάγεται από τον χρήστη .



Εικόνα 85. Η μετάδοση ενός γεγονότος τερματίζεται στην εγγενή συμπεριφορά στο οποίο έχουμε εστιάσει.

Παράδειγμα 2:

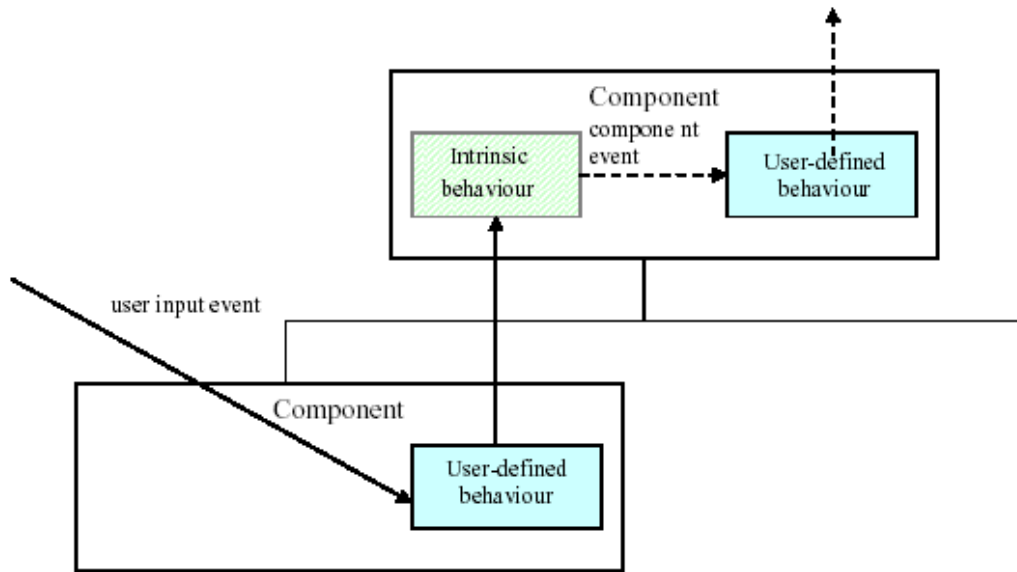
Το σχήμα 66 παρουσιάζει ένα event που εισάγει ο χρήστης και το οποίο στοχεύει αρχικά στην ενδογενή συμπεριφορά ενός focused component. Η ενδογενής συμπεριφορά δεν αποκρίνεται στο γεγονός και έτσι ο χρήστης εισάγει ένα γεγονός που διαδίδεται στην συμπεριφορά που έχει καθορίσει ο χρήστης. Η καθορισμένη από το χρήστη συμπεριφορά αποκρίνεται στο γεγονός που έχει εισαχθεί από τον χρήστη και ολοκληρώνεται. Αλλά η καθορισμένη από το χρήστη συμπεριφορά παράγει ένα component event που συνεχίζει να διαδίδεται προς τα επάνω στην ιεραρχία των component.



Εικόνα 86. Μετάδοση ενός γεγονότος όπου το δομικό στοιχείο στο οποίο έχουμε εστιάσει έχει εγγενή συμπεριφορά.

Παράδειγμα 3:

Το σχήμα 67 παρουσιάζει ένα focused component χωρίς να έχει καθορισμένη κάποια εγγενή συμπεριφορά. Αυτή η συμπεριφορά δεν αποκρίνεται στο γεγονός και έτσι προχωράει προς τα πάνω στην ιεραρχία των component. Ο γονέας του component intrinsic behaviour αποκρίνεται στο γεγονός που έχει εισαχθεί από τον χρήστη και παράγει ένα component event το οποίο περνά πρώτο από οποιοδήποτε defined behaviour που έχει καθοριστεί από τον χρήστη σε αυτό το component.



Εικόνα 87. Μετάδοση ενός γεγονότος όπου το δομικό στοιχείο στο οποίο έχουμε εστιάσει δεν έχει εγγενή συμπεριφορά.

10. Βιβλιογραφία

[1]ETSI EN 302 304 v.1.1.1 Standard. ETSI, 2004

[2] DVB-H Digital Broadcast Services to Handheld Devices, G. Faria, J. Henriksson, E. Stare, P. Talmola, IEEE, 2006

[3] DVB-H : Digital TV in the hands. Gerard FARIA TeamCast, France

[4] DVB-H The emerging standard for mobile data communication M. Kornfeld, U. Reimers, EBU Technical Report, 2005

[5] Εκπομπή δεδομένων (IP Datacasting) στην επίγεια και φορητή ψηφιακή τηλεόραση (DVB-T/ DVB-H) **Γεώργιος Γαρδικής** Εργαστήριο Ψηφιακών Τηλεπικοινωνιών Ινστ. Πληροφορικής & Τηλεπικοινωνιών, Ε.Κ.Ε.Φ.Ε. "Δημόκριτος"

[6] The Doppler Effect And Digital Video Broadcasting. Author: Michael Sobol

[7] Αμφίδρομη Επίγεια Ψηφιακή Τηλεόραση
Ξυλούρης Γιώργος

[8] Feasibility of DVB-H Deployment on Existing Wireless Infrastructure.
David Gómez-Barquero Mobile Communications Group Polytechnic University of Valencia (UPV). Aurelian Bria Radio Communications Systems Lab Royal Institute of Technology (KTH)

[9] DVB Project, A090 - DVB Portable Content Format (PCF) Commercial Requirements

[10] DVB Fact Sheet – January 2008. Portable Content Format. Enabling Cross-Platform Authoring for Interactive Applications.

[11] http://en.wikipedia.org/wiki/Portable_Content_Format#Objectives_of_PCF

[12] Semi-Automated Creation of Converged iTV Services:
From Macromedia Director Simulations to Services Ready for Broadcast

- [13] Portable Content Format
– a standard for describing an interactive digital television service
R. Bradbury, R. Cartwright, J. Hunter, S. Perrott, J.E. Rosser
- [14] Unlimited Mobile TV for the Mass Market
- [15] Mobile TV Production – Expectations, Challenges and Strategies
- [16] DVB-H Data Mapping
- [17] Focal.Press.Mobile.TV.DVB-H.DMB.3G.Systems.and.Rich.Media.Applications
- [18] DVB-H — IP Datacast over DVB-H: Architecture
- [19] DVB-H mobile digital TV
- [20] Television on a handheld receiver - broadcasting with DVB-H
- [21] Address challenges in DVB-H receiver design
- [22] Digital TV Transmission:Channel Coding and Modulation
- [23] DVB-H digital video broadcasting for handheld devices .
Based on a presentation given by Björn Forss, Magnus Melin
And a paper of AHG DVB TM-H
- [24] Handover in DVB-H
DVB-Seminar. University of Göttingen. John-Patrick Wowra
- [25] Broadcast/Satellite Communications
DVB-H architecture for mobile communications Systems.

[26] The Doppler Effect And Digital Video Broadcasting
Seminar: Digital Video Broadcasting SS2007
Dozent: Dr. Xiaoming Fu
Author: Michael Sobol

[27] DVB-H TO DELIVER DIGITAL TV TO HAND-HELD TERMINALS
Gerard FARIA (gerard.faria@teamcast.com)
TeamCast, France

[28] Unlimited Mobile TV for the Mass Market

[29] Digital Video Broadcasting (DVB);
Specification for Service Information (SI) in DVB systems

[30] Digital Video Broadcasting (DVB);
Transmission System for Handheld Terminals (DVB-H)