ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Μια Μοντελοστρεφης Προσεγγιση για την Αυτοματοποιηση της Προσομοιωσης SysML Μοντελων Συστηματων

Γεώργιος-Δημήτριος Σ. Κάπος

ΑΘΗΝΑ

19 Σεπτεμβρίου 2016

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Μία Μοντελοστρεφής Προσέγγιση για την Αυτοματοποίηση της Προσομοίωσης SysML Μοντέλων Συστημάτων

Γεώργιος-Δημήτριος Σ. Κάπος

ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ: **Μάρα Νικολαΐδου**, Καθηγήτρια ΧΠ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Μάρα Νικολαΐδου, Καθηγήτρια ΧΠ
Δημοσθένης Αναγνωστόπουλος, Καθηγητής ΧΠ
Γεώργιος Δημητρακόπουλος, Επίκουρος Καθηγητής ΧΠ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Μάρα Νικολαΐδου,
Καθηγήτρια ΧΠ

Δημοσθένης Αναγνωστόπουλος,
Καθηγητής ΧΠ

Γεώργιος Δημητρακόπουλος,
Επίκουρος Καθηγητής ΧΠ

Τσερπές Κωνσταντίνος,
Επίκουρος Καθηγητής ΧΠ

Μαρία Βίρβου,
Καθηγήτρια Πανεπιστημίου Πειραιά

Ελένη Καρατζά,
Καθηγήτρια Αριστοτέλειου Πανεπιστημίου Θεσσαλονίκης

Αφροδίτη Τσαλγατίδου,
Αναπληρώτρια Καθηγήτρια Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών

Ημερομηνία εξέτασης 19 Σεπτεμβρίου 2016

Περιληψη

Η προσομοίωση είναι μία μέθοδος, η οποία διαχρονικά υποστηρίζει με ουσιαστικό τρόπο το σχεδιασμό συστημάτων. Παρέχει δυνατότητες μελέτης της συμπεριφοράς των συστημάτων πολύ πριν αυτά υλοποιηθούν, υποστηρίζοντας τη διαδικασία επιλογής κατάλληλων σχεδιαστικών λύσεων. Αυτό αποκτά ιδιαίτερη σημασία στις περιπτώσεις όπου οι αναλυτικές προσεγγίσεις δεν είναι πρόσφορες για τη μελέτη της συμπεριφοράς των συστημάτων. Η επιτυχής αξιοποίηση της προσομοίωσης εξαρτάται σε μεγάλο βαθμό από την αντιστοιχία του μοντέλου προσομοίωσης με το πραγματικό σύστημα (υπαρκτό ή υπό σχεδίαση). Επομένως, η επιτυχής παραγωγή μοντέλου προσομοίωσης είναι καθοριστική για την επιτυχή μελέτη του συστήματος. Με τη σύγχρονη τάση για Βασισμένη-σε-Μοντέλα Ανάπτυξη Συστημάτων (Model-based Systems Engineering) (MBSE), η παραγωγή μοντέλων προσομοίωσης από μοντέλα συστήματος αποκτά μία νέα δυναμική. Παράλληλα, η Οδηγούμενη από Μοντέλα Αρχιτεκτονική (Model Driven Architecture) (MDA) προσφέρει μία σειρά από πρότυπα, γλώσσες και εργαλεία, που στοχεύουν στο χειρισμό μοντέλων. Σε αυτό το περιβάλλον είναι δυνατό να τεθούν στόχοι για την αλματώδη βελτίωση των δυνατοτήτων παραγωγής μοντέλων προσομοίωσης από μοντέλα συστήματος. Επίσης, η αναβάθμιση της έννοιας και του ρόλου του μοντέλου στη MBSE δημιουργεί τις προϋποθέσεις για τη διερεύνηση των δυνατοτήτων αξιοποίησης των αποτελεσμάτων της προσομοίωσης σε συνδυασμό με στοιχεία και απαιτήσεις, που έχουν ήδη προδιαγραφεί στο μοντέλο συστήματος. Στις δυναμικές και γεμάτες προκλήσεις συνθήκες που περιγράφονται παραπάνω, η παρούσα διατριβή προσεγγίζει το ζήτημα της αυτοματοποίησης της παραγωγής εκτελέσιμων μοντέλων προσομοίωσης από τα μοντέλα συστήματος. Συγκεκριμένα, προτείνεται μία γενική μεθοδολογία για την αξιόπιστη και αποδοτική αντιμετώπιση των προκλήσεων της παραγωγής εκτελέσιμων μοντέλων προσομοίωσης, καθώς και οι προδιαγραφές για την υλοποίηση πλαισίων, τα οποία να την υποστηρίζουν. Έτσι, επιτυγχάνεται ουσιαστική αναβάθμιση των δυνατοτήτων εκτίμησης χαρακτηριστικών απόδοσης μοντέλων συστημάτων. Αυτά ορίζονται σε ένα γενικό επίπεδο, ανεξάρτητα από το πεδίο εφαρμογής και τη μεθοδολογία ή γλώσσα προσομοίωσης. Για την επίδειξη της εφαρμοσιμότητας της μεθοδολογίας, παρουσιάζεται ένα πλαίσιο που υλοποιήθηκε για τη μεθοδολογία προσομοίωσης Προδιαγραφή Συστημάτων Διακριτού Χρόνου (Discrete Event System Specification) (DEVS), καθώς και δύο εφαρμογές, που αξιοποιούν το πλαίσιο αυτό με διαφορετικό τρόπο.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μοντελοποίηση συστημάτων και αυτοματοποίηση της προσομοίωσης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μοντελοποίηση συστημάτων, αυτοματοποίηση παραγωγής εκτελέσιμων μοντέλων προσομοίωσης, μετασχηματισμοί μοντέλων, SysML, MOF, QVT, DEVS

Abstract

Simulation is an approach that traditionally supports decision making in regard with system design solutions under consideration. It provides the means to study the behaviour of systems, long before their realization, guiding the process of selecting appropriate design solutions. This becomes more important in cases where analytical approaches cannot be applied for studying systems behaviour. Successful utilization of simulation largely depends on the correspondence of the simulation model with the actual system or system model. Therefore, precise generation of the simulation model may determine the success of the system model study. Generation of simulation models from system models may dramatically develop, in the emerging era of MBSE. In parallel, MDA delivers a series of standards, languages and tools, focusing on models management. This context is a fertile infrastructure for approaches aiming at substantially enhancing generation of simulation models from system models. Additionally, the upgraded concept and role of the *model* in MBSE ensures that the preconditions for assessing the utilization of simulation results, in combination with attributes and requirements, already defined in the system model, are met. This thesis approaches the issue of automating the generation of executable simulation models from system models, in the dynamic and challenging environment, described above. To be more specific, a generic methodology towards this direction is proposed, along with the specifications for the implementation of frameworks supporting it. The latter are defined in a generic manner, regardless the domain of application or the simulation language. An implemented framework for the DEVS simulation methodology and two different case studies demonstrate the feasibility of the approach.

SUBJECT AREA: Systems modelling and automation of simulation

KEYWORDS: Systems modelling, automation of executable simulation models generation, model transformations, SysML, MOF, QVT, DEVS

Αφιερωση

Η εργασία αυτή αφιερώνεται

σε όσους αγωνίζονται για την πρόοδο

Ευχαριστιες

Θα ήθελα να ευχαριστήσω την Τριμελή Συμβουλευτική Επιτροπή για την πολύτιμη καθοδήγηση και στήριξη καθ' όλη τη διάρκεια της εκπόνησης της Διατριβής και ιδιαίτερα την Επιβλέπουσα Καθηγήτρια ΧΠ Μάρα Νικολαΐδου και τον Καθηγητή ΧΠ Δημοσθένη Αναγνωστόπουλο για τη πολυετή συνεργασία και τα επαναλαμβανόμενα εναύσματα για έρευνα.

Σημαντική ήταν και η παραγωγική συνεργασία στα πλαίσια της στενής ερευνητικής ομάδας στην οποία συμμετείχαν ο Διδάκτορας Βασίλης Δαλάκας, ο Υποψήφιος Διδάκτορας Ανάργυρος Τσαδήμας και, ως πρόσφατο μέλος, ο απόφοιτος του Μεταπτυχιακού Προγράμματος Σπουδών Χρήστος Κοτρώνης. Ως απόδειξη της υγιούς και αειφόρου ανατροφοδότησης της ερευνητικής διαδικασίας οφείλω να αναφέρω και τη συγγενή ερευνητική συνεργασία με τους Διδάκτορα Marco DiMaio, Charles Allen και Niklas Klusmann, που προέκυψε από την προβολή της έρευνας της διατριβής, αλλά και που μου διεύρυνε την οπτική στα τελευταία στάδια της διατριβής.

Η πορεία αυτή θεμελιώθηκε στο Ίδρυμα όπου έλαβα τις Προπτυχιακές και Μεταπτυχιακές μου σπουδές (ΕΚΠΑ) και μου παρασχέθηκαν απλόχερα η γνώση και η έμπνευση. Χαρακτηριστικές περιπτώσεις είναι (α) η εμπειρία και η ενδελεχής γνώση του Καθηγητή Μιχάλη Χαντζόπουλου, που είχα την τύχη να απολαύσω ως φοιτητής και ως ερευνητής, (β) η ανάδειξη της αφαιρετικής προσέγγισης με γλώσσες μοντελοποίησης (UML) από την Αναπληρώτρια Καθηγήτρια Αφροδίτη Τσαλγατίδου, (γ) η μεθοδικότητα και η πνευματική διαύγεια της Επίκουρης Καθηγήτριας Ιζαμπούς Καράλη και (δ) ο ενθουσιασμός και το πάθος για την έρευνα και τις τεχνολογίες αντικειμενοστρεφούς και βασισμένης-σε-συνιστώσες ανάπτυξης συστημάτων του Διδάκτορα Δημήτρη Θεοτόκη.

Καθώς η γνώση και η αγάπη για μάθηση κατακτώνται σταδιακά, ευχαριστώ τους δασκάλους και καθηγητές της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, ξεχωρίζοντας το ζήλο και τη μεταδοτικότητα του Καθηγητή Φυσικής του 46ου Λυκείου Αθηνών, Φωκίωνα Συρέλη, και τον πολύπλευρο και πολυδιάστατο τρόπο εκπαίδευσης του Δασκάλου του 116ου Δημοτικού Σχολείου Αθηνών, Πολυχρόνη Σαββουλίδη.

Ευχαριστώ τους δύο μικρούς γιους μου, Σπύρο και Σωτήρη, για την αξιοθαύμαστη υπομονή που επέδειξαν όλες εκείνες τις ώρες που η Διατριβή είχε προτεραιότητα, αλλά κυρίως για τα μαθήματα έμπνευσης, ενθουσιασμού και ευρηματικότητας, που μου παραδίδουν. Η αμέριστη συμπαράσταση, υποστήριξη και αγάπη από τη σύζυγο μου, Θώμη, ήταν η δύναμη που συγκράτησε την πορεία αυτή εντός τροχιάς. Τέλος, ευχαριστώ τους γονείς μου για τις αρχές που μου εμφύσησαν, τις εμπειρίες που μου διεύρυναν τους ορίζοντες και για το ότι υπό δύσκολες συνθήκες μου αγόρασαν στο γυμνάσιο (1989) τον πρώτο υπολογιστή.

Περιεχομενα

Πρόλογος

Το κείμενο αυτό είναι η τελική, ολοκληρωμένη και αναλυτική αποτύπωση της έρευνας που έγινε στα πλαίσια της διδακτορικής διατριβής με τίτλο "Αυτοματοποίηση της Προσομοίωσης στη Βασισμένη-σε-Μοντέλα Ανάπτυξη Συστημάτων". Το επιστημονικό περιεχόμενο και τα αποτελέσματα παρουσιάζονται στα κεφάλαια του κειμένου.

Στην *εισαγωγή* επιχειρείται μία επιτελική σύνοψη του περιεχομένου και της συνεισφοράς της διατριβής, ώστε να προετοιμαστεί ο αναγνώστης για την ανάγνωση του υπολοίπου κειμένου. Στη συνέχεια (Κεφάλαιο 2), περιγράφεται η υφιστάμενη κατάσταση στο χώρο της βασισμένης-σε-μοντέλα ανάπτυξης συστημάτων και της προσομοίωσης, ορίζοντας το υπόβαθρο στο οποίο βασίζεται η έρευνα. Προχωρώντας ένα βήμα παραπέρα, στο Κεφάλαιο 3 περιγράφονται οι πρόσφατες ερευνητικές προσπάθειες σε σχετικά αντικείμενα. Έτσι, δίνεται μία αίσθηση του ενδιαφέροντος που συγκεντρώνει το συγκεκριμένο ερευνητικό πεδίο και των αποτελεσμάτων που έχουν επιτευχθεί, αναδεικνύοντας τα κίνητρα και τις δυνατότητες συνεισφοράς της παρούσας διατριβής.

Έχοντας περιγράψει μέχρι αυτό το σημείο τις βασικές έννοιες, αλλά και τις ανταγωνιστικές προτάσεις, στο Κεφάλαιο 4 παρουσιάζεται η κύρια πρόταση και συνεισφορά της διατριβής: (α) μία γενική μεθοδολογία αποτίμησης σχεδιαστικών λύσεων για συστήματα, που βασίζεται στην αυτοματοποιημένη προσομοίωση μοντέλων συστημάτων και (β) οι γενικές προδιαγραφές πλαισίων, που μπορούν να υποστηρίξουν τη μεθοδολογία αυτή. Καθώς οι προδιαγραφές των πλαισίων είναι γενικές, ενώ για τη χρήση της μεθοδολογίας απαιτείται η αξιοποίηση συγκεκριμένων προσομοιωτών, στο Κεφάλαιο 5 παρουσιάζεται η υλοποίηση του πλαισίου για τη μεθοδολογία προσομοίωσης DEVS, που αποδεικνύει την εφαρμοσιμότητα της πρότασης και αποτελεί μία πρόσθετη συνεισφορά.

Για την περαιτέρω επίδειξη των πρακτικών θεμάτων εφαρμογής και χρήσης της μεθοδολογίας, στα Κεφάλαια 6 και 7 παρουσιάζονται δύο περιπτώσεις εφαρμογών, όπου αξιοποιείται το υλοποιημένο πλαίσιο με διαφορετικό τρόπο. Έτσι, εξετάζονται διαφορετικές πτυχές των δυνατοτήτων υλοποίησης των πλαισίων, καθώς και της εφαρμογής και χρήσης της μεθοδολογίας. Στο Κεφάλαιο 8 επιχειρείται η συγκέντρωση των πλέων ουσιαστικών στοιχείων από την εμπειρία του ορισμού των προδιαγραφών, την ανάπτυξη του πλαισίου και τις σχετικές εφαρμογές. Οι πληροφορίες αυτές μπορούν να διευκολύνουν και να κατευθύνουν μελλοντικές σχετικές προσπάθειες, αποτελώντας μία πρόσθετη συνεισφορά.

Τέλος, στο Κεφάλαιο 9, γίνεται απολογισμός των αποτελεσμάτων της διατριβής και παράθεση των προοπτικών για μελλοντικές επεκτάσεις.

Όπως σε κάθε δραστηριότητα, έτσι και στην περίπτωση της παρούσας διατριβής, οι εμπειρίες που αποκομίστηκαν κατά τη διάρκειά της ήταν πολύτιμες και έχουν επηρεάσει το κείμενο αυτό. Πρόκειται για την πνευματική απολαβή που αποκομίζει σταδιακά ο υποψήφιος διδάκτορας στην πορεία που διαγράφει από τη σύλληψη της αρχικής ιδέας, μέχρι τη σύνταξη του κειμένου της διατριβής και την προετοιμασία για την υποστήριξή της. Ενδιαφέρον, ενθουσιασμός, απογοήτευση, υπομονή, επιμονή, εκτίμηση, αμφισβήτηση, γνωριμίες με μέλη της επιστημονικής κοινότητας -άμεσες ή μέσω της δημοσιευμένης δουλειάς τους- είναι ορισμένα από αυτά. Σημαντικότερη ίσως όλων είναι η εμπειρία της σταδιακής ένταξης στην επιστημονική κοινότητα με έναν τρόπο πιο βαθύ και ουσιαστικό σε σχέση με προηγούμενα επίπεδα σπουδών. Αυτό γίνεται με την εμπέδωση της συμμετοχής στην κοινότητα μέσω συνεισφοράς προτάσεων και ιδεών, αλλά και λήψης χρήσιμων συμβουλών και κατευθύνσεων από αυτή, πάντα στα πλαίσια της ειλικρινούς και θετικής διαλλακτικής θεώρησης των πραγμάτων.

Η επιρροή από το εγγύτερο επιστημονικό περιβάλλον επίβλεψης της διατριβής και συνεργασίας ήταν φυσικά διαρκής και άμεση. Η επιλογή του θέματος, της κατεύθυνσης των προτεινόμενων λύσεων, των σχετικών επιστημονικών κοινοτήτων και η αξιοποίηση των σχετιζόμενων εργασιών ήταν ορισμένα από τα θέματα που καθόρισαν την πορεία και την ποιότητα της παρούσας διατριβής. Ακόμα και η αλληλεπίδραση με τους προπτυχιακούς και μεταπτυχιακούς φοιτητές προσέφερε χρήσιμες εναλλακτικές οπτικές και προσεγγίσεις.

Φυσικά, η επιστημονική προσέγγιση στη διατριβή είναι μεν ανεξάρτητη από άλλους παράγοντες της ζωής, αλλά δεν μπορεί να αποστειρωθεί πλήρως από τις υπόλοιπες σκέψεις και εμπειρίες του υποψήφιου διδάκτορα. Ο οικογενειακός και φιλικός κύκλος, το εργασιακό περιβάλλον και οι προκλήσεις του, καθώς και άλλες δραστηριότητες καθ' όλη τη διάρκεια του διδακτορικού, αλλά και πριν από αυτό, συν-διαμορφώνουν το ευρύτερο περιβάλλον στα πλαίσια του οποίου εκπονείται η διατριβή.

Κεφαλαιο 1

Εισαγωγη

1.1 Γενικά

Η προσομοίωση είναι μία επιστημονική μέθοδος (ή τεχνική) που στοχεύει στην εκτίμηση της συμπεριφοράς συστημάτων, όπως προκύπτει από την εκτέλεση σεναρίων λειτουργίας μοντέλων προσομοίωσης, που αντιστοιχούν στα συστήματα αυτά. Προσφέρεται έτσι μία εναλλακτική προσέγγιση πειραματισμού με το πραγματικό σύστημα, αντί της αναλυτικής, στην οποία η εκτιμώμενη συμπεριφορά των συστημάτων προκύπτει από την επεξεργασία των μαθηματικών μοντέλων τους. Προφανώς, οι παράγοντες που καθορίζουν την ακρίβεια της εκτίμησης της προσομοίωσης είναι κυρίως η αξιοπιστία της μεθοδολογίας προσομοίωσης και η επιτυχής αποτύπωση των χαρακτηριστικών του συστήματος στο μοντέλο προσομοίωσης. Επομένως, η έννοια του μοντέλου είναι κυρίαρχη στην προσομοίωση και η ορθότητά του, σε σχέση με το πραγματικό/σχεδιαζόμενο σύστημα, καθοριστική για την επιτυχή αξιοποίησή της.

Στην πράξη, η χρήση της προσομοίωσης γίνεται συνήθως σε κάποιο στάδιο σχεδιασμού του συστήματος και προϋποθέτει τον ορισμό εκτελέσιμου μοντέλου προσομοίωσης, σε πλήρη αντιστοιχία με τη διαμορφωμένη εικόνα για το υπό ανάπτυξη σύστημα. Έτσι, δημιουργείται παράλληλα με τις υπόλοιπες δραστηριότητες ανάπτυξης ένα μοντέλο προσομοίωσης, το οποίο πρέπει να έχει αντίστοιχα χαρακτηριστικά με αυτά του συστήματος. Σε κάποιες περιπτώσεις, επιχειρείται η αυτόματη παραγωγή μοντέλου προσομοίωσης, βασισμένη σε μοντέλα του συστήματος, που έχουν αναπτυχθεί στα πλαίσια άλλης δραστηριότητας, π.χ. λογική σχεδίαση συστήματος. Σε αυτές τις περιπτώσεις, παράγεται αυτόματα κάποιος σκελετός προγράμματος προσομοίωσης, που πρέπει στη συνέχεια να εμπλουτιστεί. Κατ' αυτό τον τρόπο, ενυπάρχει πιθανότητα εισαγωγής αποκλίσεων στο μοντέλο προσομοίωσης σε σχέση με το διαφαινόμενο πραγματικό σύστημα, με τις αρνητικές επιπτώσεις που αυτό συνεπάγεται.

Στη MBSE ένα κεντρικό μοντέλο χρησιμοποιείται σε όλη τη διάρκεια της ανάπτυξης των συστημάτων, σε ένα ευρύ φάσμα δραστηριοτήτων. Καθώς οι διακριτές δραστηριότητες έχουν ειδικές απαιτήσεις και υποβοηθούνται από διαφορετικά προγράμματα/εφαρμογές, το κεντρικό μοντέλο του συστήματος πρέπει να εμπλουτίζεται και να μπορεί να μετασχηματιστεί σε μορφές αξιοποιήσιμες από τα προγράμματα αυτά.

Σε αυτά τα πλαίσια, θεωρούμε τη διαδικασία εκτίμησης (estimation) της απόδοσης ενός συστήματος με προσομοίωση, ως μία τέτοια δραστηριότητα, που μπορεί να εντάσσεται και να εξυπηρετεί τη γενικότερη δραστηριότητα των δοκιμών (testing) και της επαλήθευσης απαιτήσεων (requirements verification). Λαμβάνοντας υπόψη την κυρίαρχη θέση της έννοιας του μοντέλου στην προσομοίωση, προκύπτει εύλογα το κίνητρο της έρευνας για προσδιορισμό και αυτοματοποίηση αυτής της δραστηριότητας.

1.2 Επισκόπηση Διατριβής

Η διατριβή εκπονήθηκε στο πεδίο της εκτίμησης της συμπεριφοράς μοντέλων συστημάτων, που διαμορφώνονται κατά τη βασισμένη σε μοντέλα ανάπτυξη συστημάτων και συγκεκριμένα στην αυτοματοποίηση της παραγωγής εκτελέσιμου κώδικα προσομοίωσης. Τα μοντέλα συστημάτων αποτελούν αναπαραστάσεις υπαρκτών ή υπό ανάπτυξη συστημάτων, με σκοπό την αποτύπωση των δομικών χαρακτηριστικών τους και της δυναμικής συμπεριφοράς τους, αλλά και την προαγωγή της διαδικασίας σχεδιασμού, ανάπτυξης και ελέγχου συστημάτων, γενικότερα, με την περαιτέρω επεξεργασία τους. Υπάρχουν πολλοί τρόποι μοντελοποίησης συστημάτων, ανάλογα με το σκοπό που εξυπηρετείται σε κάθε περίπτωση (ανάλυση συγκεκριμένης κατηγορίας συστημάτων, θεωρητική/μαθηματική θεμελίωση, προγραμματιστικό περιβάλλον μελέτης και επεξεργασίας).

Η εκτίμηση της συμπεριφοράς των συστημάτων γίνεται συνήθως είτε με αναλυτικό τρόπο, είτε με χρήση προσομοίωσης. Στην πρώτη περίπτωση (αναλυτική μέθοδος), το μοντέλο του συστήματος αναλύεται στα επιμέρους στοιχεία του και μελετώνται τα χαρακτηριστικά και οι σχέσεις τους, συνήθως εκφρασμένες με μαθηματικές σχέσεις. Στη δεύτερη περίπτωση (προσομοίωση), γίνεται πειραματισμός και μετρήσεις κατά την τεχνητή / εικονική λειτουργία του συστήματος. Καθώς η συμπεριφορά των συστημάτων συνήθως δεν είναι απόλυτα προβλέψιμη, κατά την εκτέλεση της προσομοίωσης, χρησιμοποιούνται τυχαία δείγματα από κατάλληλες συναρτήσεις κατανομών. Επίσης, το μοντέλο προσομοίωσης εκτελείται πολλές φορές και τα αποτελέσματα είναι αξιοποιήσιμα μετά από κατάλληλη στατιστική επεξεργασία.

Υπάρχει πληθώρα λύσεων και στις δύο προσεγγίσεις, ενώ και το εξεταζόμενο πεδίο έχει αποτελέσει αντικείμενο εκτεταμένης έρευνας. Οι περισσότερες από τις προσεγγίσεις όμως εμφανίζουν έναν ή περισσότερους από τους ακόλουθους περιορισμούς:

- παράγουν κώδικα προσομοίωσης, που απαιτεί συμπλήρωση για να γίνει εκτελέσιμος
- χρησιμοποιούν κλειστούς ή ειδικού σκοπού τρόπους αναπαράστασης των μοντέλων συστημάτων, περιορίζοντας το εύρος εφαρμογής και τις δυνατότητες διαλειτουργικότητας

- χρησιμοποιούν κλειστούς ή χαμηλού επιπέδου τρόπους αναπαράστασης των μοντέλων προσομοίωσης, περιορίζοντας τις δυνατότητες διαλειτουργικότητας σε επίπεδο προσομοιωτών
- χρησιμοποιούν χαμηλού επιπέδου τρόπους μετασχηματισμού για την παραγωγή των μοντέλων προσομοίωσης, επιβαρύνοντας την ανάπτυξη, τον ελέγχου έλεγχο και τη συντήρηση του μετασχηματισμού
- αντιμετωπίζουν επιμέρους ζητήματα της εξεταζόμενης περιοχής και δεν παρέχουν ένα συνολικό πλαίσιο για την ανάπτυξη και χρήση τέτοιων λύσεων, παρέχοντας ουσιαστική αναβάθμιση των συνθηκών εργασίας των σχεδιαστών συστημάτων

Στην παρούσα διατριβή, με βάση τους περιορισμούς άλλων ερευνητικών προσπαθειών, τέθηκαν οι ακόλουθοι στόχοι για την αντιμετώπιση του ζητήματος της αυτοματοποίησης της προσομοίωσης στη βασισμένη σε μοντέλα ανάπτυξη συστημάτων:

- **Γενικότητα εφαρμογής**: Η πρόταση δεν πρέπει να περιορίζεται σε συστήματα συγκεκριμένου πεδίου εφαρμογής, αλλά να μπορεί να εφαρμοστεί στην ευρύτερη δυνατή γκάμα συστημάτων.
- **Γενικότητα αναπαράστασης**: Δεν πρέπει να εισάγονται περιορισμοί στον τρόπο αναπαράστασης των μοντέλων συστημάτων, όπως proprietary γλώσσες συγκεκριμένων εμπορικών λύσεων, αλλά να υπάρχει, κατά το δυνατό, ανοιχτή προσέγγιση.
- **Αξιοπιστία εκτίμησης**: Η εκτιμώμενη συμπεριφορά θα πρέπει να είναι κατά το δυνατό αξιόπιστη, ώστε να μπορεί να χρησιμοποιηθεί για την αξιολόγηση των μοντέλων συστημάτων με τρόπο αποτελεσματικό.
- **Αναβάθμιση της διαδικασίας**: Η διαδικασία εκτίμησης μοντέλων συστημάτων θα πρέπει να απλοποιείται και να υποβοηθείται, ώστε να είναι αποδοτικότερο το έργο των σχεδιαστών συστημάτων, τόσο στην εκτέλεση προσομοίωσης όσο και στην αξιοποίηση των παραγόμενων αποτελεσμάτων.

Για την επίτευξη των παραπάνω στόχων, η διατριβή κινήθηκε με βάση δύο κύριες επιλογές:

- **Επιλογή SysML**: Ο τρόπος αναπαράστασης μοντέλων που επιλέχθηκε είναι η γενικού σκοπού γλώσσα περιγραφής συστημάτων SysML. Η συγκεκριμένη επιλογή προσδίδει μία σειρά από πλεονεκτήματα στην προσέγγιση, αφού η SysML ισορροπεί μεταξύ θεωρητικής θεμελίωσης, πρακτικότητας, γενικότητας και εμπορικής υποστήριξης. Αυτό επιτυγχάνεται, αφού η SysML είναι ορισμένη ως ένα προφίλ στην Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language) (UML), σύμφωνα με το μετα-μοντέλο της τελευταίας. Το γεγονός αυτό καθιστά τη SysML εκφραστική και επεξεργάσιμη σε μοντελοθεωρητικό επίπεδο, ενώ παράλληλα, ως πρότυπο του Object Management Group (OMG), υποστηρίζεται από πληθώρα εταιρειών και προϊόντων. Έτσι, εξασφαλίζεται η βασική υποδομή για γενικότητα εφαρμογής και αναπαράστασης, ενώ δίνεται η δυνατότητα αξιοποίησης μεθοδολογιών και εργαλείων ορισμού και επεξεργασίας των μοντέλων.

- **Επιλογή DEVS**: Για την προσομοίωση στις εφαρμογές, που υλοποιήθηκαν για την επίδειξη της εφαρμοσιμότητας της προσέγγισης, επιλέχθηκε το DEVS, το οποίο είναι μία θεμελιωμένη θεωρητικά μεθοδολογία προσομοίωσης, που υποστηρίζεται από σχετικούς προσομοιωτές. Επίσης, ο τρόπος δόμησης, ιεράρχησης και διασύνδεσης των στοιχείων των μοντέλων προσομοίωσης DEVS είναι αντίστοιχος με τη SysML, καθιστώντας την επιλογή κατάλληλη, δεδομένης της επιλογής της SysML.

Δεδομένων των βασικών επιλογών, για την επίτευξη των ανωτέρω στόχων έγιναν μία σειρά από υποθέσεις, ενέργειες και περαιτέρω επιλογές, οι οποίες αναφέρονται ακολούθως και κατέληξαν σε συγκεκριμένες προτάσεις, οι οποίες δημοσιεύθηκαν:

- **Πρόταση μεθοδολογίας και προδιαγραφών πλαισίων [1, 2]**: Προτάθηκε μία γενική μεθοδολογία για την αυτοματοποίηση της προσομοίωσης στη βασισμένη σε μοντέλα ανάπτυξη συστημάτων. Έμφαση δόθηκε στην ελαχιστοποίηση και απλοποίηση των απαιτούμενων από το σχεδιαστή ενεργειών, στη γενικότητα εφαρμογής και αναπαράστασης των δεδομένων και στην ενσωμάτωση των αποτελεσμάτων προσομοίωσης στο περιβάλλον μοντελοποίησης συστημάτων. Παράλληλα παρουσιάστηκαν γενικές προδιαγραφές πλαισίων, ανεξάρτητα από το περιβάλλον προσομοίωσης, που θα επιτρέπουν την ομαλή εκτέλεση της μεθοδολογίας.

- **Ανάπτυξη SysML προφίλ για DEVS [3]**: Η SysML είναι μία γενική και επεκτάσιμη γλώσσα μοντελοποίησης συστημάτων, λόγω του μετα-μοντέλου της UML, στο οποίο βασίζεται και το οποίο με τη σειρά του είναι ορισμένο σύμφωνα με την Υποδομή Μετα-Αντικειμένων (Meta-Object Facility) (MOF). Ωστόσο, η SysML δεν υποστηρίζει άμεσα την προσομοίωση των μοντέλων συστημάτων. Έτσι, αναπτύχθηκε ένα προφίλ στη SysML, ώστε να είναι δυνατή η εξειδίκευση και ο εμπλουτισμός των μοντέλων συστημάτων με τα απαιτούμενα χαρακτηριστικά προσομοίωσης. Το προφίλ αποτελείται από ένα σύνολο στερεοτύπων (stereotypes), για το χαρακτηρισμό των στοιχείων των μοντέλων SysML, και ένα σύνολο περιορισμών (constraints), που διασφαλίζουν την εγκυρότητα -ως προς τα χαρακτηριστικά προσομοίωσης- των μοντέλων SysML. Οι περιορισμοί είτε ορίστηκαν δηλωτικά με προτάσεις ορισμένες στη Γλώσσα Περιορισμών Αντικειμένων (Object Constraint Language) (OCL), είτε υλοποιήθηκαν προγραμματιστικά, ως plug-in στο εργαλείο μοντελοποίησης MagicDraw Modeling Tool by No Magic, Inc. (MagicDraw).

- **Ορισμός μετα-μοντέλου DEVS [4]**: Το SysML προφίλ για DEVS εξασφαλίζει ότι τα έγκυρα -σύμφωνα με το προφίλ- μοντέλα μπορούν να προσομοιωθούν. Ωστόσο, για να εκτελεστεί η προσομοίωση πρέπει να παραχθούν τα αντίστοιχα μοντέλα προσομοίωσης DEVS. Σε αυτό το σημείο υπήρχε ένα κενό, γιατί αν και το DEVS έχει συγκεκριμένο τρόπο αναπαράστασης με θεωρία συνόλων και συγκεκριμένα προγραμματιστικά περιβάλλοντα που το υποστηρίζουν (π.χ. DEVSJava, DEVSC++), δεν υπήρχε ένας υψηλού επιπέδου τρόπος προσδιορισμού μοντέλων DEVS, που να περιγράφει εκτελέσιμα μοντέλα DEVS και να είναι και διαχειρίσιμος σε επίπεδο μοντελοποίησης. Το κενό αυτό εξακολουθούσε

παρά τις επιμέρους προσπάθειες για ορισμό αναπαραστάσεων περιορισμένων μορφών του DEVS στην Επεκτάσιμη Γλώσσα Σημάνσεων (Extensible Markup Language) (XML) και συμπληρώθηκε με τον ορισμό του μετα-μοντέλου για το DEVS. Το τελευταίο είναι ορισμένο σύμφωνα με το πρότυπο MOF, το οποίο αποτελεί και κοινή υποδομή με τη SysML/UML, καθιστώντας την υψηλού επιπέδου επεξεργασία και διαχείριση μοντέλων DEVS εφικτή μέσα από μία σειρά προτυποποιημένων μεθοδολογιών και εργαλείων. Τα στοιχεία αυτά καθιστούν αυτό το μετα-μοντέλο κατάλληλο ως σημείο αναφοράς στην πληθώρα προσεγγίσεων και εργαλείων που υποστηρίζουν το DEVS.

- **Ορισμός Query/View/Transform (QVT) μετασχηματισμού μοντέλων SysML σε μοντέλα DEVS [5]**: Δεδομένης της ανάπτυξης του SysML προφίλ για DEVS (δυνατότητα ορισμού μοντέλων SysML με χαρακτηριστικά προσομοίωσης DEVS) και του ορισμού του μετα-μοντέλου DEVS (σημείο αναφοράς για την εκτέλεση της προσομοίωσης), ορίστηκε ο μετασχηματισμός μοντέλων SysML (μετα-μοντέλο UML με SysML και DEVS προφίλ) σε μοντέλα DEVS (μετα-μοντέλο DEVS). Λόγω της κοινής υποδομής των μετα-μοντέλων (MOF), ο ορισμός έγινε με χρήση QVT-Relations (QVT-R), η οποία είναι ένα πρότυπο του OMG και έχει δημιουργηθεί για αυτόν ακριβώς το σκοπό, παρέχοντας υψηλού επιπέδου και ισχυρές δυνατότητες ενδοσκόπησης των μοντέλων και δηλωτικού προσδιορισμού του μετασχηματισμού.

- **Εφαρμογές μεθοδολογίας [6, 7]**: Παρουσιάστηκαν εφαρμογές της μεθοδολογίας, αξιοποιώντας το υλοποιημένο πλαίσιο για το περιβάλλον προσομοίωσης DEVS, σε δύο τελείως διαφορετικά πεδία: "Αξιολόγηση Στρατηγικών Διεξαγωγής Μάχης" και "Εκτίμηση Απόδοσης σε Εταιρικά Πληροφοριακά Συστήματα (Enterprise Information Systems) (EIS)". Συγκεκριμένα, το πλαίσιο αξιοποιήθηκε με δύο εναλλακτικούς τρόπους. Στη μία περίπτωση, δόθηκε έμφαση στην εξ ολοκλήρου δήλωση της συμπεριφοράς των συστημάτων στο περιβάλλον μοντελοποίησης συστημάτων, οδηγώντας όμως σε εκτέλεση προσομοίωσης. Στην άλλη περίπτωση, εξετάστηκε η λογική της αξιοποίησης βιβλιοθηκών συστατικών λογισμικού (library components) προσομοίωσης, για τη διευκόλυνση της εφαρμογής σε μοντέλα συστημάτων μεγάλου μεγέθους και τη μελέτη της αξιοποίησης των αποτελεσμάτων προσομοίωσης.

- **Βέλτιστες πρακτικές [8]**: Προτάθηκαν βέλτιστες πρακτικές για ενδεχόμενες σχετικές μελλοντικές απόπειρες έρευνας, αλλά και εφαρμογών στο πεδίο. Για τη διαμόρφωση των προτεινόμενων πρακτικών έγινε συγκέντρωση και οργάνωση των σημαντικών στοιχείων και των εμπειριών από την έρευνα στα πλαίσια της διατριβής και, κυρίως, από τις εφαρμογές που έγιναν.

Συμπερασματικά, στα πλαίσια της διατριβής αντιμετωπίστηκε ουσιαστικά το θέμα της αναβάθμισης της διαδικασίας εκτίμησης της συμπεριφοράς μοντέλων συστημάτων, με χρήση προσομοίωσης. Η προσέγγιση και οι επιμέρους λύσεις που προτάθηκαν και υλοποιήθηκαν δίνουν τη δυνατότητα προσομοίωσης μοντέλων συστημάτων ορισμένων στην ευρέως διαδεδομένη SysML.

Προτάθηκε μία γενική (ως προς το πεδίο εφαρμογής, το περιβάλλον προσομοίωσης και τα χρησιμοποιούμενα εργαλεία) μεθοδολογία προς αυτή την κατεύθυνση και γενικές προδιαγραφές πλαισίων για την υποστήριξή της. Υλοποιήθηκε ένα τέτοιο πλαίσιο, το οποίο αξιοποιεί διαθέσιμους προσομοιωτές, συμβατούς με το φορμαλισμό για προδιαγραφή και προσομοίωση συστημάτων διακριτού χρόνου, DEVS. Η υφιστάμενη μαθηματική θεμελίωση της συγκεκριμένης μεθοδολογίας προσομοίωσης παρέχει ένα συμπαγές και σαφώς ορισμένο θεωρητικό μοντέλο, το οποίο εξυπηρετεί τη διερεύνηση και τον ορισμό των στοιχείων του πλαισίου. Συγκεκριμένα, για την υλοποίηση του πλαισίου ορίστηκαν (α) ένα γενικό, υψηλού επιπέδου, αλλά και εκτελέσιμο μετα-μοντέλο για DEVS σε όρους MOF, (β) ένα προφίλ για DEVS στη SysML και (γ) ο γενικός μετασχηματισμός μοντέλων SysML (σύμφωνα με το ανωτέρω προφίλ) σε εκτελέσιμα μοντέλα DEVS με χρήση της γλώσσας ορισμού μετασχηματισμών QVT-R. Τέλος, υλοποιήθηκαν και παρουσιάστηκαν εφαρμογές της μεθοδολογίας σε διαφορετικά πεδία, ενώ επιχειρήθηκε και η συγκέντρωση και παρουσίαση των ουσιαστικότερων πρακτικών ζητημάτων στο εξεταζόμενο πεδίο.

1.3 Διάρθρωση Κειμένου

Μετά τη γενική εισαγωγή στο αντικείμενο της διατριβής, που επιχειρείται σε αυτό το κεφάλαιο, το περιεχόμενο του κειμένου της διατριβής διαρθρώνεται ως ακολούθως. Στο Κεφάλαιο 2 αποτυπώνεται η υπάρχουσα κατάσταση στο χώρο της βασισμένης-σε-μοντέλα ανάπτυξης συστημάτων και της προσομοίωσης συστημάτων, ώστε να είναι σαφής η δεδομένη υφιστάμενη κατάσταση. Στο Κεφάλαιο 3 αναλύονται ερευνητικές προσπάθειες που σχετίζονται με την παρούσα, ως προς το στόχο ή τα χρησιμοποιούμενα μέσα. Στο Κεφάλαιο 4 αναλύονται η βασική ιδέα της διατριβής, η προτεινόμενη μεθοδολογία και οι προδιαγραφές για τη δημιουργία πλαισίων τα οποία να υποστηρίζουν τη λειτουργία της μεθοδολογίας. Στο Κεφάλαιο 5 παρουσιάζεται το πλαίσιο που υλοποιήθηκε για την εφαρμογή της μεθοδολογίας αξιοποιώντας το πλαίσιο προσομοίωσης DEVS. Στα Κεφάλαια 6 και 7 παρουσιάζονται δύο περιπτώσεις εφαρμογής της πρότασης (α) με περιγραφή της συμπεριφοράς προσομοίωσης των συστατικών στοιχείων των μοντέλων και (β) με χρήση διαθέσιμων συστατικών λογισμικού προσομοίωσης, αντίστοιχα. Στο Κεφάλαιο 8 παρουσιάζονται προτεινόμενες πρακτικές, για ενδεχόμενες απόπειρες αντίστοιχων προσεγγίσεων και εφαρμογών. Τέλος, στο κεφάλαιο 9 ανακεφαλαιώνονται συμπερασματικές παρατηρήσεις, σχολιάζεται η συνεισφορά της διατριβής, σε σχέση με την υφιστάμενη κατάσταση, αλλά και με τις σχετικές ερευνητικές προσπάθειες και αναφέρονται ανοιχτές, ερευνητικές κατευθύνσεις.

Κεφαλαιο 2

Σχετικο Επιστημονικο Υποβαθρο

Σε αυτό το κεφάλαιο περιγράφονται έννοιες και θέματα από το χώρο της έρευνας και της βιομηχανίας (industry) που διαμορφώνουν την υπάρχουσα κατάσταση στο χώρο, όπου εκπονείται η διατριβή. Έμφαση δίνεται σε θέματα που αφορούν τη MBSE και στη θεωρία και τις πρακτικές στην προσομοίωση συστημάτων.

2.1 Βασισμένη σε Μοντέλα Ανάπτυξη Συστημάτων

Στη MBSE, όπως ορίζεται από το Διεθνές Συμβούλιο Μηχανικής Συστημάτων (International Council on Systems Engineering) (INCOSE) [9], ένα κεντρικό μοντέλο συστήματος χρησιμοποιείται σαν σημείο αναφοράς για την εκτέλεση όλων των δραστηριοτήτων ανάπτυξης, όπως ο ορισμός προδιαγραφών, ο σχεδιασμός, η ολοκλήρωση, η επικύρωση και η λειτουργία ενός συστήματος. Όμως, οι περισσότερες δραστηριότητες γίνονται με χρήση διακριτών, ανεξάρτητα ορισμένων μοντέλων συστημάτων. Για παράδειγμα, η επαλήθευση απαιτήσεων συστημάτων είναι μια δραστηριότητα που γίνεται συχνά βασισμένη σε μοντέλα, με προσομοίωση [10], κυρίως όταν το σύστημα δεν είναι διαθέσιμο ή οι δοκιμές με το πραγματικό σύστημα δεν είναι δυνατές ή είναι ασύμφορες.

Η SysML [11] προτάθηκε ως μία γενικού σκοπού, γραφική γλώσσα μοντελοποίησης, για την περιγραφή των μοντέλων αναφοράς, που χρησιμοποιούνται για την εκτέλεση όλων των

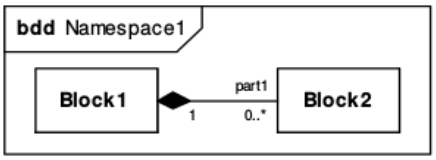
δραστηριοτήτων ανάπτυξης ενός μεγάλου εύρους *συστημάτων* και *συστημάτων από συστήματα (systems of systems)*. Τα *συστήματα από συστήματα* αποτελούνται από ένα σύνολο άλλων αυτόνομων συστημάτων, τα οποία λειτουργούν συνδυαστικά και συνεργατικά, με σκοπό την επίτευξη ενός στόχου [12]. Συγκεκριμένες δραστηριότητες ανάπτυξης μπορούν να ολοκληρωθούν από ένα μηχανικό συστημάτων είτε χρησιμοποιώντας ένα εργαλείο μοντελοποίησης για SysML (π.χ. σχεδιασμός συστήματος), είτε από εξωτερικά εργαλεία (π.χ. επικύρωση συστήματος) ή ακόμα και από συνδυαστική χρήση τους. Τα μοντέλα συστήματος σε SysML πρέπει να ορίζονται ανεξάρτητα από εργαλεία, που επικεντρώνουν σε κάποια συγκεκριμένη δραστηριότητα, αλλά να υποστηρίζουν διαφορετικά επίπεδα λεπτομέρειας για την υποστήριξη όλων των δραστηριοτήτων ανάπτυξης. Τα μοντέλα αυτά πρέπει να εμπλουτίζονται κατάλληλα, ώστε να μπορούν να υποστηρίξουν συγκεκριμένες δραστηριότητες. Ένα τέτοιο παράδειγμα μπορεί να είναι η υποστήριξη και διευκόλυνση της επικύρωσης μοντέλων συστημάτων από εξωτερικά εργαλεία. Η SysML παρέχει τα μέσα για τον εμπλουτισμό μοντέλων, με κατάλληλη αξιοποίηση των μηχανισμών επέκτασης της UML, όπως τα προφίλ και τα στερεότυπα [13].

2.1.1 Επισκόπηση SysML

Τα κύρια δομικά στοιχεία για την περιγραφή συστημάτων με την SysML είναι τα *blocks*, τα οποία μπορεί να περιέχουν:

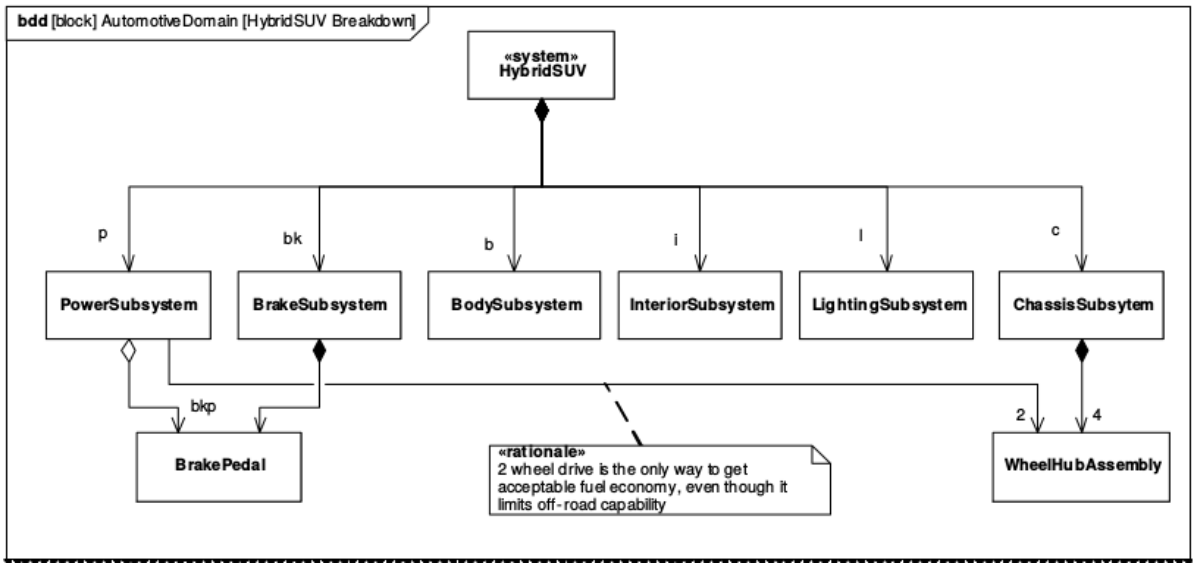
- *ιδιότητες τιμών (value properties)*, δηλαδή μεταβλητές για τη διατήρηση τιμών
- *ιδιότητες τμημάτων (part properties)*, δηλαδή άλλα blocks, που περιέχονται εντός του περιγραφόμενου
- *ιδιότητες αναφοράς (reference properties)*, δηλαδή αναφορές προς άλλα blocks, τα οποία χρησιμοποιούνται από το περιγραφόμενο
- *πόρτες (ports)*, που χρησιμοποιούνται ως τερματικά σημεία (endpoints) για τις διασυνδέσεις των blocks. Διαχωρίζονται σε:
  - *standard ports*, τα οποία χρησιμοποιούνται για την ανταλλαγή διακριτών μηνυμάτων / γεγονότων (events) και
  - *flow ports*, τα οποία χρησιμοποιούνται για τη μοντελοποίηση συνεχούς ροής στοιχείων ή υλικών (π.χ. υγρά, αέρια, ενέργεια)
- *περιορισμοί (constraints)*, δηλαδή σχέσεις μεταξύ των properties των blocks.

Η δομή του συστήματος ορίζεται σε Διαγράμματα Ορισμού Μπλοκ (Block Definition Diagrams) (BDDs), Διαγράμματα Εσωτερικού Μπλοκ (Internal Block Diagrams) (IBDs) και Διαγράμματα Παραμετροποίησης (Parametric Diagrams) (PDs). Τα BDDs παρέχουν μία συνολική, ιεραρχική αναπαράσταση της δομής του συστήματος, ως block, που αποτελείται από άλλα blocks-μέρη. Το σχήμα 2.1 δείχνει πώς μπορεί σε ένα BDD να οριστεί ένα block (Block1) ως σύνθεση πολλών (0..*) block άλλου τύπου (Block2).



Σχήμα 2.1: Υπόδειγμα BDD (από το SysML specification v1.3 σελ. 32).

Το παράδειγμα BDD του σχήματος 2.2 παρουσιάζει ένα μοντέλο συστήματος υβριδικού αυτοκινήτου, το οποίο ορίζεται ως σύνθεση μίας σειράς υποσυστημάτων (blocks): για την ισχύ, την πέδηση, το σώμα του οχήματος, το εσωτερικό, το φωτισμό και την πλατφόρμα (chassis).

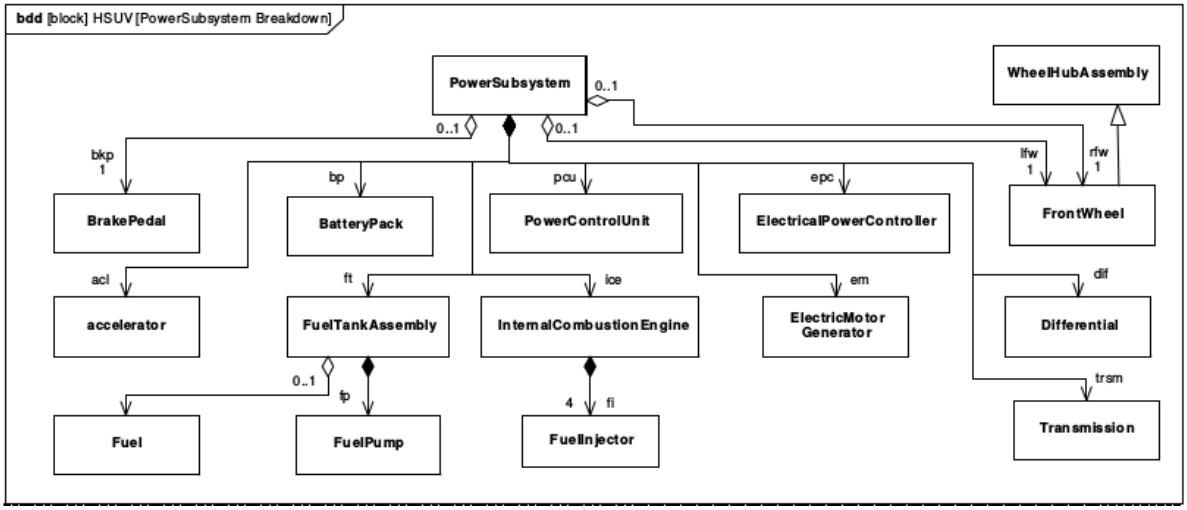


Σχήμα 2.2: BDD για την περιγραφή υβριδικού αυτοκινήτου (από το SysML specification v1.3 σελ. 194).

Στις συνδέσεις, που δείχνουν αυτή τη σχέση σύνθεσης, χρησιμοποιείται ο ρόμβος στην πλευρά της σύνθετης οντότητας. Όταν ο ρόμβος είναι όλος σκούρος, σημαίνει ότι η απλή οντότητα είναι μέρος (part) της σύνθετης, ενώ όταν το ο ρόμβος είναι λευκός στο εσωτερικό του, η απλή οντότητα δεν είναι μέρος της σύνθετης, απλά χρησιμοποιείται από αυτή (reference). Έτσι, το πεντάλ του φρένου είναι μέρος του συστήματος πέδησης, αλλά χρησιμοποιείται από το σύστημα ισχύος, για την αποθήκευση ενέργειας, που παράγεται ως τριβή κατά την πέδηση. Επίσης, στις σχέσεις αυτές μπορούν να χρησιμοποιούνται αριθμοί, που δείχνουν την το πλήθος των περιεχόμενων υποσυστημάτων. Για παράδειγμα, η πλατφόρμα (chassis) περιλαμβάνει 4 υποσυστήματα βάσης-τροχού, ενώ το σύστημα ισχύος συσχετίζεται μόνο με 2 από αυτά, για λόγους οικονομίας καυσίμου, όπως εξηγείται στο σημείωμα τύπου rationale.

Η ιεραρχική αυτή περιγραφή της σύνθεσης των συστημάτων δεν έχει περιορισμό στον αριθμό των επιπέδων. Για λόγους χρηστικότητας ή παρουσίασης είναι δυνατό σε διαφορετικά BDDs να παρουσιάζονται διαφορετικά τμήματα του γράφου αυτού. Για παράδειγμα, η σύνθεση του συστήματος ισχύος αποτυπώνεται στο σχήμα 2.3.

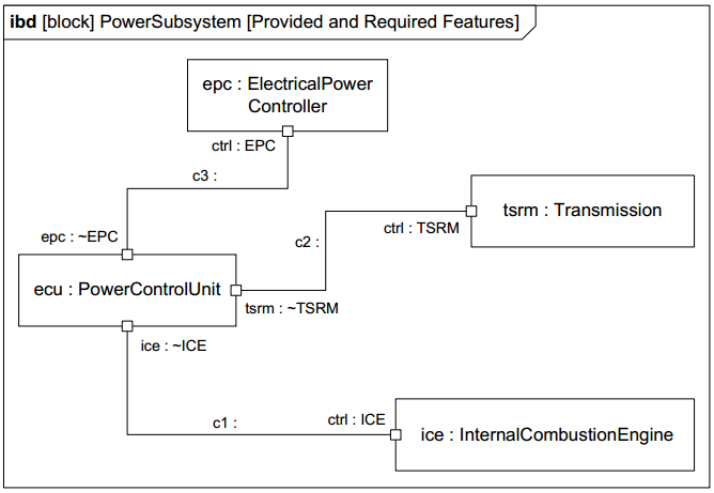
Τα IBDs εστιάζουν στην εσωτερική περιγραφή των σύνθετων blocks (αυτών που περιέχουν άλλα blocks). Εδώ προσδιορίζονται οι διασυνδέσεις μεταξύ των περιεχόμενων blocks και, συ-



Σχήμα 2.3: BDD για την περιγραφή του υποσυστήματος ισχύος (από το SysML specification v1.3 σελ. 195).

γκεκριμένα, μέσω των ports που διαθέτουν. Το σχήμα 2.4 παρουσιάζει τη διασύνδεση μερικών από τα μέρη του υποσυστήματος ισχύος. Η μονάδα ελέγχου, λόγω της φύσης της συνδέεται με όλα τα υποσυστήματα. Είναι εμφανής η διαφορά μεταξύ BDDs και IBDs, όπου στα πρώτα καθορίζεται η σύνθεση των συστημάτων από άλλα υποσυστήματα, ενώ στα δεύτερα ορίζονται οι διασυνδέσεις μεταξύ των υποσυστημάτων.

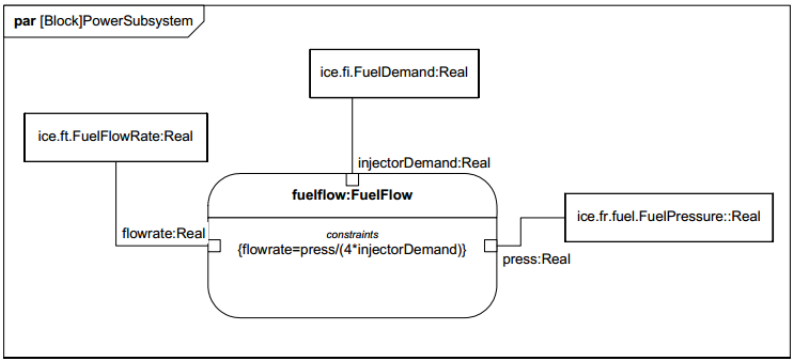
[!hb]



Σχήμα 2.4: Παράδειγμα IBD (από το SysML specification v1.3, σελ. 74).

Στα PDs συνδυάζονται μεταβλητές περιορισμών, που έχουν οριστεί για ένα σύστημα, με ιδιότητες τιμών του συστήματος. Δηλαδή, συσχετίζονται συγκεκριμένα block value properties (π.χ. ροή καυσίμου, ταχύτητα κ.λ.π.) με μεταβλητές των constraint blocks (που συμμετέχουν σε εξισώσεις). Έτσι είναι δυνατό να αποτυπωθούν οι επιτρεπτές συνθήκες λειτουργίας του συστήματος. Παράλληλα, με βάση αυτή την πληροφορία, καθίσταται εφικτός, κατά τη φάση της επικύρωσης του μοντέλου συστήματος, ο έλεγχος, αν η εκτίμηση της συμπεριφοράς του συστήματος (συγκεκριμένες τιμές για τα value properties) βρίσκεται σε επιτρεπτά πλαίσια λειτουργίας.

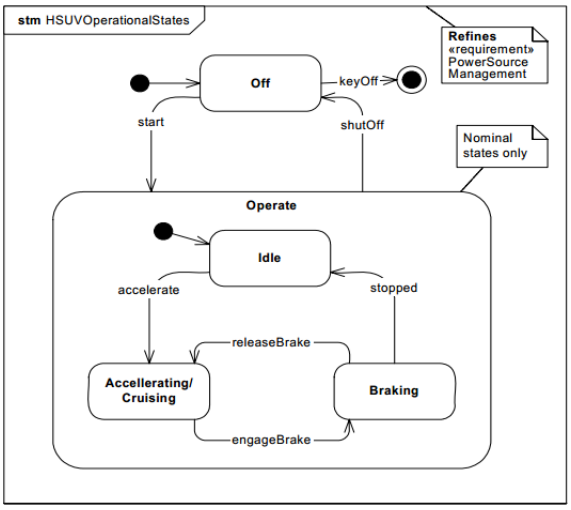
Το σχήμα 2.5 παρουσιάζει ένα παράδειγμα PD, όπου 3 value properties (η ροή καυσίμου, η απαίτηση καυσίμου και η πίεση καυσίμου) συσχετίζονται με 3 μεταβλητές (flowrate, injectorDemand και press) μίας εξίσωσης περιορισμού του συστήματος. Αν υπάρξει εκτίμηση τιμών για τα 3 value properties, π.χ. μέσω προσομοίωσης, τότε μπορεί να ελεγχθεί αν οι τιμές αυτές επαληθεύουν την εξίσωση.



Σχήμα 2.5: Παράδειγμα PD (από το SysML specification v1.3, σελ. 199).

Η συμπεριφορά του συστήματος περιγράφεται με μία σειρά από διαγράμματα. Στα Διαγράμματα Μηχανών Καταστάσεων (State Machine Diagrams) (SMDs) ορίζονται οι καταστάσεις των blocks, οι μεταβάσεις από κατάσταση σε κατάσταση και οι συνθήκες εφαρμογής για κάθε μετάβαση.

Το σχήμα 2.6 παρουσιάζει τις καταστάσεις λειτουργίας του αυτοκινήτου, που χρησιμοποιήθηκε και στα προηγούμενα παραδείγματα. Οι καταστάσεις ορίζονται σε ορθογώνια πλαίσια με στρογγυλεμένες γωνίες και οι μεταβάσεις με βέλη, που συνδέουν καταστάσεις. Οι συνθήκες μετάβασης γράφονται στο βέλος μετάβασης.



Σχήμα 2.6: Παράδειγμα SMD (από το SysML specification v1.3, σελ. 188).

Υπάρχουν ειδικές καταστάσεις, όπως:

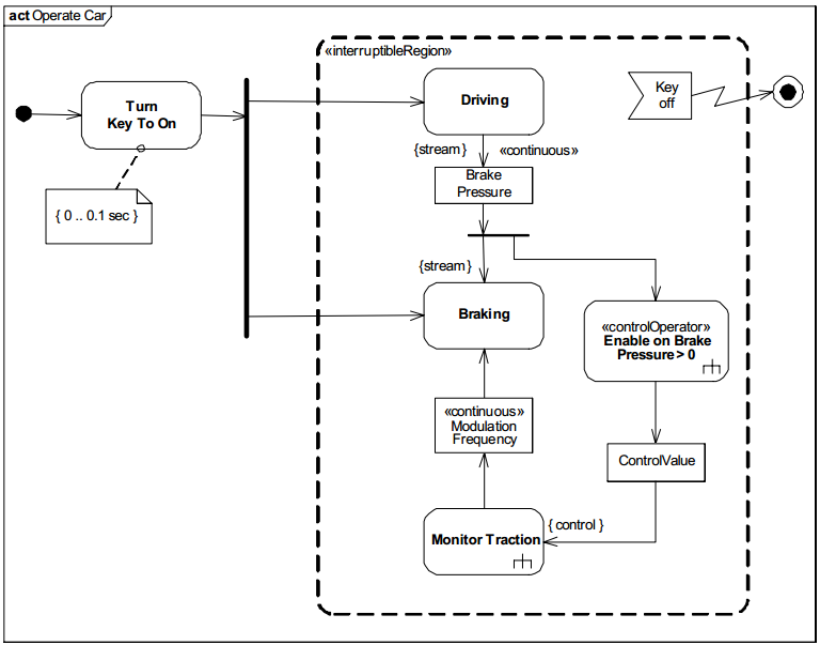
- η αρχική, που είναι μοναδική και συμβολίζεται με συμπαγή κύκλο και ορίζει την αρχική κατάσταση, στην οποία οδηγεί με μετάβαση (“Off“ στο παράδειγμα)

- η τελική, που συμβολίζεται με συμπαγή κύκλο με εξωτερικό δακτύλιο και δείχνει μία καταληκτική κατάσταση του συστήματος

Στο σχήμα 2.6 φαίνεται ότι είναι δυνατός ο ορισμός σύνθετων καταστάσεων (εδώ Operate), οι οποίες έχουν τη δική τους αρχική κατάσταση και ένα σύνολο υπο-καταστάσεων.

Στο παράδειγμα του σχήματος 2.6, υπάρχουν δύο καταστάσεις: "Off" και "Operate". Αρχική κατάσταση είναι η "Off" και το μοντέλο καταστάσεων τερματίζει όταν το σύστημα βρίσκεται στην κατάσταση "Off" και τεθεί το αφαιρεθεί κλειδί (keyOff). Η κατάσταση "Operate" έχει 3 υπο-καταστάσεις (Idle, Accelerating/Cruising και Braking). Όταν το σύστημα βρεθεί σε κατάσταση "Operate", τότε βρίσκεται αρχικά στην κατάσταση "Idle". Με τις αντίστοιχες ενέργειες (accelarate, engageBrake, releaseBrake και stopped) μπορεί να εναλλάσσεται στις υπο-καταστάσεις της "operate", ενώ με "shutOff", το σύστημα πηγαίνει σε κατάσταση "Off".

Τα Διαγράμματα Ενεργειών (Activity Diagrams) (ADs) δίνουν έμφαση στην οργάνωση και τον έλεγχο των ενεργειών που εκτελούνται στα blocks. Σε ένα AD περιγράφεται μία δραστηριότητα (activity) του συστήματος, ως ένα σύνολο ενεργειών (actions). Ένα παράδειγμα AD παρουσιάζεται στο σχήμα 2.7.



Σχήμα 2.7: Παράδειγμα AD (από το SysML specification v1.3, σελ. 109).

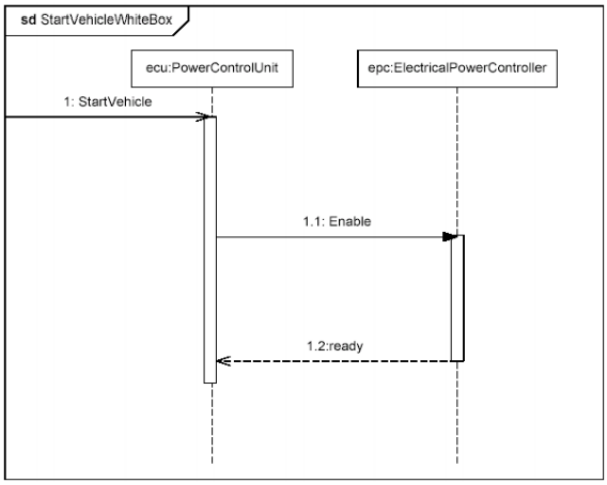
Όπως φαίνεται και στο παράδειγμα, οι δραστηριότητες περιγράφονται με:

- αρχικό κόμβο (συμπαγής κύκλος), που δείχνει το σημείο αφετηρίας της δραστηριότητας
- τελικό κόμβο (συμπαγής κύκλος με εξωτερικό δακτύλιο), που δείχνει τη λήξη της δραστηριότητας
- ενέργειες, που συμβολίζονται με ορθογώνια με στρογγυλεμένες γωνίες
- κόμβους δεδομένων (ορθογώνια), για τη μοντελοποίηση των δεδομένων που μεταφέρονται μεταξύ των ενεργειών

- κόμβους διαχωρισμού/συνένωσης (fork/join) της ροής (χοντρές οριζόντιες ή κάθετες γραμμές)

- κόμβους συμβάντων (ορθογώνια με εσοχή στα αριστερά), για τη μοντελοποίηση συμβάντων που επηρεάζουν την εκτέλεση της δραστηριότητας

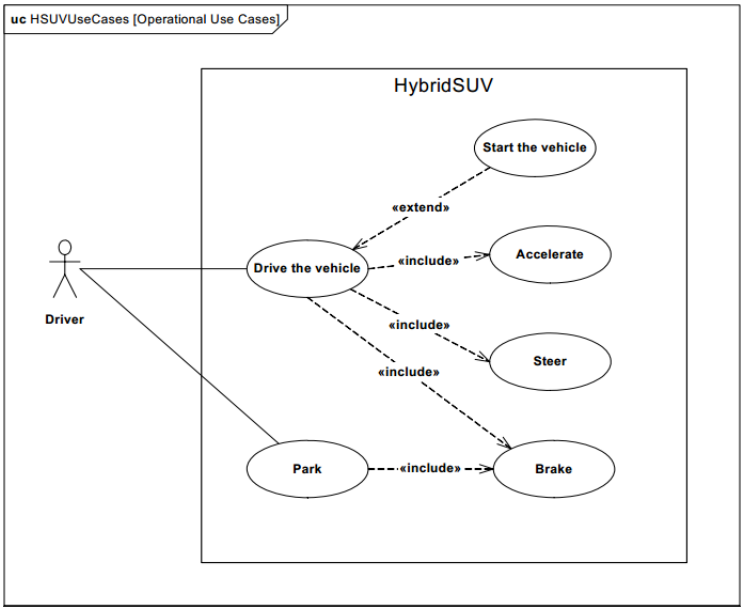
Τα Διαγράμματα Διαδοχής (Sequence Diagrams) (SDs) επικεντρώνουν σε θέματα συγχρονισμού των ενεργειών των blocks, καθώς και των παραγόμενων και των λαμβανόμενων γεγονότων (events). Όπως φαίνεται και στο σχήμα 2.8, στα SDs χρησιμοποιούνται blocks, για τα οποία εμφανίζεται μία γραμμή της δραστηριότητάς τους στο χρόνο (lifeline). Με αυτό τον τρόπο η διαδοχή της ανταλλαγής των μηνυμάτων αποτυπώνεται με σαφήνεια.



Σχήμα 2.8: Παράδειγμα SD (από το SysML specification v1.3, σελ. 189).

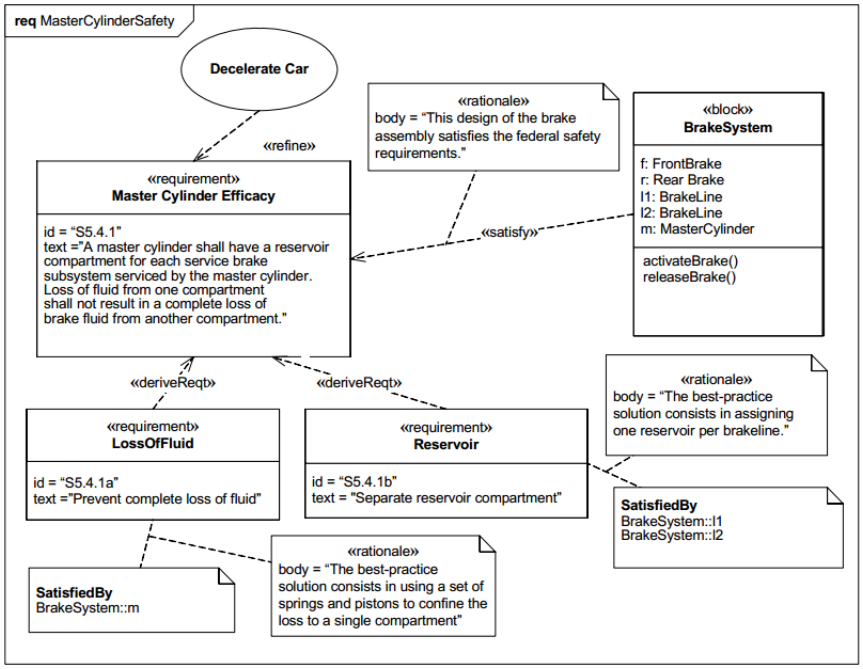
Τα Διαγράμματα Περιπτώσεων Χρήσης (Use Case Diagrams) (UCs) χρησιμοποιούνται για την καταγραφή περιπτώσεων χρήσης του συστήματος. Σε αυτά αναγνωρίζονται οι παράγοντες (actors), που αλληλεπιδρούν με το σύστημα και τα σενάρια χρήσεις στα πλαίσια του συστήματος. Ένα παράδειγμα UC φαίνεται στο σχήμα 2.9. Οι περιπτώσεις χρήσης μπορούν να σχετίζονται με σχέσεις τύπου "include" ή "extend". Στον πρώτο τύπο σχέσης (include), η περίπτωση χρήσης από την οποία ξεκινάει το βέλος (π.χ. "Drive the vehicle") περιλαμβάνει ως επιμέρους βήμα την πιο λεπτομερή (π.χ. "Accelerate"). Στον άλλο τύπο σχέσης (extends), η περίπτωση χρήσης από την οποία ξεκινάει το βέλος (π.χ. "Start the vehicle") επεκτείνει άλλη (π.χ. "Drive the vehicle"), προσθέτοντας ένα σενάριο χρήσης, που μπορεί να είναι απαραίτητο, υπό προϋποθέσεις. Για παράδειγμα, ενώ η οδήγηση περιλαμβάνει λειτουργίες, όπως η επιτάχυνση, η πέδηση και η διεύθυνση, μπορεί να χρειαστεί και την εκκίνηση του οχήματος, αν αυτό δεν είναι ήδη σε λειτουργία.

Τέλος, τα Διαγράμματα Απαιτήσεων (Requirement Diagrams) (RDs) έχουν περιληφθεί στη SysML για την μοντελοποίηση και διαχείριση των απαιτήσεων του συστήματος, που έχουν ιδιαίτερη σημασία στην ανάπτυξη συστημάτων. Οι απαιτήσεις ορίζονται ως αντικείμενα με ένα μοναδικό αναγνωριστικό (id) και μία περιγραφή (text), ενώ μπορούν να διασυνδέονται μεταξύ τους, αλλά και με άλλα στοιχεία του μοντέλου συστήματος. Έτσι, παρέχουν μία πιο



Σχήμα 2.9: Παράδειγμα UC (από το SysML specification v1.3, σελ. 186).

εμπεριστατωμένη εικόνα σχετικά με την επεξήγηση / αιτιολόγηση των απαιτήσεων, αλλά και την ικανοποίηση τους από τμήματα του συστήματος (σχήμα 2.10).



Σχήμα 2.10: Παράδειγμα RD (από το SysML specification v1.3, σελ. 151).

Στο RD του παραδείγματος παρουσιάζονται απαιτήσεις και τρεις βασικές σχέσεις:

- deriveReqt: Δείχνει ότι μία απαίτηση προέκυψε ως αποτέλεσμα της επεξεργασίας μίας άλλης προγενέστερης και γενικότερης απαίτησης
- refine: Χρησιμοποιείται για τη συσχέτιση μίας απαίτησης με άλλα στοιχεία του μοντέλου συστήματος (π.χ. use cases, activities), που περιγράφουν πιο αναλυτικά επιμέρους χαρακτηριστικά της απαίτησης.

- satisfy: Δείχνει ότι ένα στοιχείο του μοντέλου συστήματος ικανοποιεί μία απαίτηση.

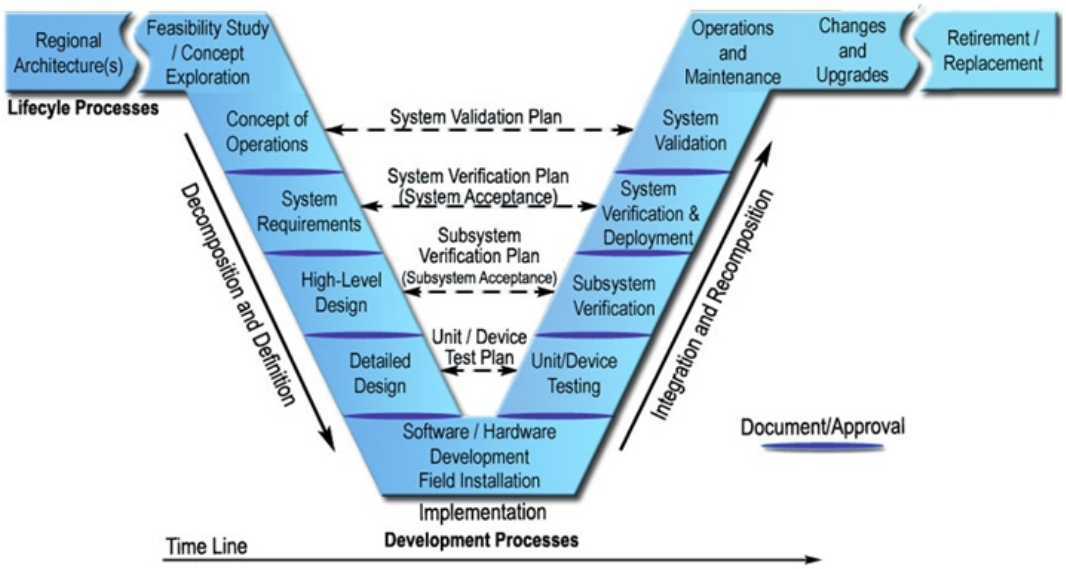
Σημειώσεις τύπου rationale χρησιμοποιούνται για την καλύτερη τεκμηρίωση της ανάλυσης απαιτήσεων.

2.1.2 Δραστηριότητες Ανάπτυξης Συστημάτων

Η ανάπτυξη συστημάτων είναι μία σύνθετη διαδικασία, η οποία διεξάγεται σε μία σειρά από φάσεις, που συνήθως περιλαμβάνουν [14]:

- Διερεύνηση και μελέτη σκοπιμότητας
- Εννοιολογική αποτύπωση των λειτουργιών του συστήματος
- Ανάλυση απαιτήσεων για το υπό ανάπτυξη σύστημα
- Υψηλού επιπέδου σχεδιασμός του συστήματος
- Αναλυτικός σχεδιασμός του συστήματος (λειτουργίες, δομή)
- Παραγωγή συστήματος
- Δοκιμές μονάδων
- Δοκιμές αποδοχής υποσυστημάτων
- Δοκιμές αποδοχής συστήματος
- Επαλήθευση επίτευξης σκοπού συστήματος
- Λειτουργία και συντήρηση συστήματος
- Αλλαγές και αναβαθμίσεις
- Απόρριψη/Ανακύκλωση/Αντικατάσταση συστήματος

Μία συνήθης αποτύπωση των φάσεων αυτών φαίνεται στο σχήμα 2.11. Εδώ οι φάσεις παρατίθενται σε διάταξη "V" και μοιράζονται σε αυτές που αφορούν την ανάλυση/αποσύνθεση και τον ορισμό του συστήματος (αριστερό σκέλος) και σε αυτές που αφορούν την ολοκλήρωση και την ανασύνθεση του συστήματος (δεξί σκέλος). Για κάθε ζευγάρι αντίστοιχης φάσης αποσύνθεσης και ολοκλήρωσης υπάρχουν οι σχετικές διαδικασίες ελέγχου, δοκιμών, επικύρωσης και επαλήθευσης του συστήματος.



Σχήμα 2.11: Το V-model της διαδικασίας ανάπτυξης συστημάτων (Clarus Concept of Operations. Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005)

Κάθε φάση ανάπτυξης περιλαμβάνει ένα σύνολο δραστηριοτήτων που διεξάγονται για την υλοποίησή της. Οι διακριτές φάσεις ανάπτυξης, ακόμα και οι επιμέρους δραστηριότητες μιας φάσης υποβοηθούνται από διαφορετικά εργαλεία/εφαρμογές που έχουν δημιουργηθεί από ειδικούς, ώστε να βελτιώνουν και να επιταχύνουν τις δραστηριότητες, διασφαλίζοντας υψηλά επίπεδα αξιοπιστίας στα παραγόμενα αποτελέσματα. Καθώς όμως τα εργαλεία χρησιμοποιούν διαφορετικά μοντέλα δεδομένων, η ομαλότητα της διαδικασίας ανάπτυξης συστημάτων εξαρτάται σε μεγάλο βαθμό από την απρόσκοπτη μετάβαση από το μοντέλο δεδομένων της μίας δραστηριότητας σε αυτό της επόμενης.

2.1.3 Διαχείριση Μοντέλων Συστημάτων

Όπως αναλύθηκε και στο 2.1.2, η ανάπτυξη συστημάτων είναι μία σύνθετη διαδικασία αποτελούμενη από πολλές δραστηριότητες. Κάθε δραστηριότητα υποστηρίζεται από εργαλεία που απαιτούν διαφορετικές μορφές του μοντέλου συστήματος, που αποκωδικοποιούν καλύτερα διαφορετικές όψεις και χαρακτηριστικά του. Επομένως, η διαχείριση των μοντέλων συστημάτων και η μετάβαση στις διαφορετικές μορφές που απαιτούνται σε κάθε δραστηριότητα καθίσταται κρίσιμη για την αποδοτική και επιτυχή ανάπτυξη συστημάτων.

Η παραγωγή διαφορετικών μορφών μοντέλων συστημάτων μπορεί να γίνει με:

- **εξ αρχής ορισμό της διαφορετικής μορφής** μοντέλου συστήματος από αναλυτή/μηχανικό συστημάτων.
- **δημιουργία ειδικού προγράμματος μετασχηματισμού** που αναγνωρίζει τις πληροφορίες που περιέχονται σε μια μορφή μοντέλου συστήματος και παράγει το αντίστοιχο μοντέλο σε διαφορετική μορφή.
- **υιοθέτηση μοντελο-κεντρικής προσέγγισης**, που επιτρέπει τον ορισμό μετασχηματισμών,

βασισμένων στην εννοιολογική αντιστοίχηση των δομών και των δεδομένων των διαφορετικών μορφών μοντέλων συστημάτων.

Ο ορισμός του μοντέλου συστήματος εκ νέου είναι απαραίτητος στις περιπτώσεις, που δεν υπάρχει διαθέσιμο μοντέλο του συστήματος, που να μπορεί να αξιοποιηθεί για την ηλεκτρονική επεξεργασία και αυτόματη παραγωγή αντίστοιχου μοντέλου προσομοίωσης. Αυτό συμβαίνει όταν η ανάλυση και ο σχεδιασμός του συστήματος είναι βασισμένα σε έγγραφα (document-based systems engineering) ή όταν τα μοντέλα είναι ορισμένα με χρήση κλειστών μηχανογραφικών εφαρμογών [15]. Η επιλογή αυτή παρουσιάζει μία σειρά από μειονεκτήματα, όπως:

- δεν αξιοποιούνται από προγράμματα οι υπάρχουσες μορφές μοντέλων συστημάτων
- υπάρχει αυξημένη πιθανότητα εισαγωγής σφαλμάτων και ασυμφωνιών σε σχέση με τη μορφή μοντέλου συστήματος προέλευσης
- είναι δύσκολη η επαλήθευση της αντιστοιχίας μεταξύ των διαφορετικών μορφών των μοντέλων συστημάτων
- απαιτείται πολλή χειρωνακτική εργασία από τον αναλυτή/μηχανικό συστημάτων

Τα παραπάνω μειονεκτήματα καθιστούν ουσιαστικά μη αποδοτική και πρακτική την επιλογή αυτή.

Η δημιουργία ειδικού προγράμματος μετασχηματισμού αντιμετωπίζει ορισμένα από τα προβλήματα της πρώτης επιλογής, αφού υποτίθεται ότι υπάρχει κεντρικό μοντέλο συστήματος και αξιοποιείται για την παραγωγή μέρους του μοντέλου προσομοίωσης. Έτσι μειώνεται η πιθανότητα εισαγωγής χρηστικού σφάλματος, λόγω του περιορισμού της χειρωνακτικής εργασίας. Εν τούτοις, υπάρχουν θέματα που δεν αντιμετωπίζονται επαρκώς:

- ο βαθμός δυσκολίας δημιουργίας του προγράμματος μετασχηματισμού μπορεί να αυξηθεί σημαντικά, λόγω σύνθετης αναπαράστασης των κεντρικών μοντέλων συστημάτων και των μοντέλων προσομοίωσης.
- αντίστοιχα, και η συντήρηση ενός προγράμματος μετασχηματισμού είναι δύσκολη, καθώς περιλαμβάνει πολλές εξαρτήσεις από λεπτομέρειες τόσο του κεντρικού μοντέλου συστήματος, όσο και του μοντέλου προσομοίωσης.
- η επικύρωση ενός προγράμματος μετασχηματισμού δεν είναι εύκολη με εξέταση του κώδικα, αφού απαιτείται σύνθετη τεχνική υλοποίηση, ακόμα και για απλές λογικές συσχετίσεις. Απαιτείται τουλάχιστον ανάλυση της δομής του κεντρικού μοντέλου, συνδυασμός διαφορετικών στοιχείων από αυτό και δημιουργία αντίστοιχης δομής στο μοντέλο προσομοίωσης. Ακόμα και αν η κατασκευή του προγράμματος μετασχηματισμού έχει γίνει στα πλαίσια κάποιας θεωρητικής προσέγγισης μετασχηματισμού, τυχόν θεωρητική απόδειξη της εγκυρότητας του μετασχηματισμού, εξετάζεται με επιφύλαξη, καθώς ο θεωρητικά ορθός μετασχηματισμός μπορεί να μην έχει μεταφερθεί σωστά στο τεχνικό επίπεδο της υλοποίησης.

- δεν αξιοποιούνται τυχόν κοινά αποδεκτές υποδομές μοντελοποίησης και διαχείρισης των μοντέλων, που να διευκολύνει τον ορισμό, την επικύρωση και την επικαιροποίηση μετασχηματισμών.

Κατ' αυτό τον τρόπο, η δημιουργία και χρήση ειδικού προγράμματος μετασχηματισμού μπορεί να διευκολύνει τη διαδικασία μετασχηματισμού μοντέλων. Έχει όμως περιορισμούς που καθιστούν τη δημιουργία, τον έλεγχο και τη συντήρηση τέτοιων προγραμμάτων ένα αυτοτελές έργο πληροφορικής με δυσκολίες, που περιορίζουν την έμφαση που θα έπρεπε να δίνεται στην εννοιολογική σημασία του μετασχηματισμού.

Τέλος, με την υιοθέτηση μοντελο-κεντρικής θεώρησης, η μετάβαση από μία μορφή μοντέλου σε άλλη υλοποιείται μέσω γενικών εργαλείων που αξιοποιούν την αποτυπωμένη γνώση σχετικά με τις σχέσεις δομής και περιεχομένου, ανάμεσα στις δύο μορφές. Σε αυτά τα πλαίσια, κάθε μετασχηματισμός δεν ξεκινά από μηδενική βάση όσον αφορά τη σύνταξη και τις εννοιολογικές δομές μοντελοποίησης της κάθε μορφής. Αντίθετα, προϋποθέτει την ύπαρξη κοινού υπόβαθρου μοντελοποίησης στις διακριτές μορφές των μοντέλων συστημάτων για τις διάφορες δραστηριότητες, που επιτρέπει την ανάπτυξη και χρήση ενός συνόλου γενικών γλωσσών, θεωριών και εργαλείων για τον ορισμό και την εκτέλεση των μετασχηματισμών [16–19]. Το υπόβαθρο αυτό είναι εν πολλοίς μία κοινή, γενική θεμελιώδης γλώσσα, με την οποία ορίζεται κάθε διαφορετικός τρόπος μοντελοποίησης για κάθε δραστηριότητα. Ο τρόπος ορισμού μοντέλων για κάθε δραστηριότητα ονομάζεται μετα-μοντέλο της δραστηριότητας, ενώ το θεμελιώδες υπόβαθρο ονομάζεται μετα-μετα-μοντέλο.

Με την επιλογή της μοντελο-κεντρικής θεώρησης αντιμετωπίζονται ουσιαστικά και τα προβλήματα της δημιουργίας ειδικού προγράμματος μετασχηματισμού, αφού είναι διαθέσιμες δομές και λειτουργίες, που έχουν αναπτυχθεί με βάση το κοινό υπόβαθρο και επιτρέπουν την ομαλή αντιμετώπιση των ιδιαιτεροτήτων του κάθε μετα-μοντέλου. Έτσι, οι μετασχηματισμοί μπορούν να ορίζονται σε ένα πιο υψηλό επίπεδο απλοποιώντας τη δημιουργία, τη συντήρηση, αλλά και την επικύρωση τους.

Ωστόσο, υπάρχουν κάποια θέματα που δεν επιτρέπουν πάντα ή δυσκολεύουν την απρόσκοπτη επιλογή της μοντελο-κεντρικής θεώρησης:

- επειδή όταν εξετάζεται η μοντελο-κεντρική προσέγγιση συνήθως είναι ήδη διαθέσιμες οι εφαρμογές υποστήριξης των δραστηριοτήτων ανάπτυξης, είναι πολύ πιθανό να μην έχουν μοντέλα δεδομένων, ορισμένα στα πλαίσια ενός κοινού υποβάθρου.

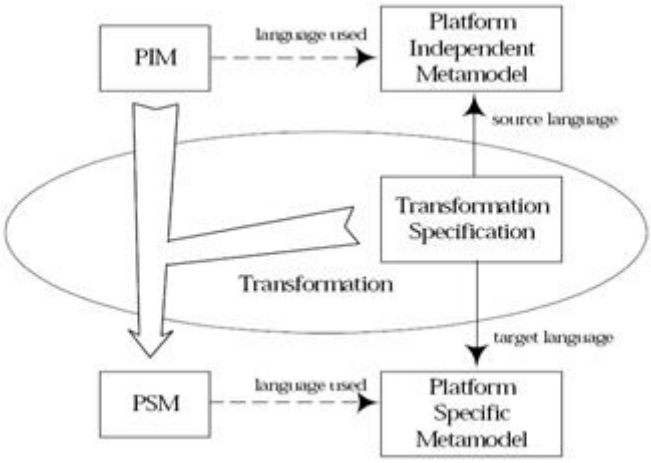
- η μοντελο-κεντρική προσέγγιση δεν προσανατολίζεται στις λεπτομέρειες υλοποίησης του αντικειμενικού στόχου της κάθε δραστηριότητας, η οποία υποστηρίζεται και τεκμηριώνεται σε λεπτομέρεια από κάποια εταιρεία ή κοινότητα. Αντίθετα, βασίζεται σε γενικές πρότυπες προδιαγραφές (standard specifications), οι οποίες υποστηρίζονται από πρότυπες εφαρμογές, συχνά ανοιχτού κώδικα, που δεν υποστηρίζονται πάντα και δεν τεκμηριώνονται πολύ αναλυτικά.

- επειδή η συγκεκριμένη λογική σχετίζεται ουσιαστικά με τη δια-λειτουργικότητα διακριτών εργαλείων (συχνά από διαφορετικές εταιρείες), δεν γίνονται συνήθως οι απαιτού-

μενες σχετικές προωθητικές ενέργειες, οι οποίες να ευνοούν την εκτεταμένη εφαρμογή της.

Object Management Group Model-Driven Architecture (OMG MDA)

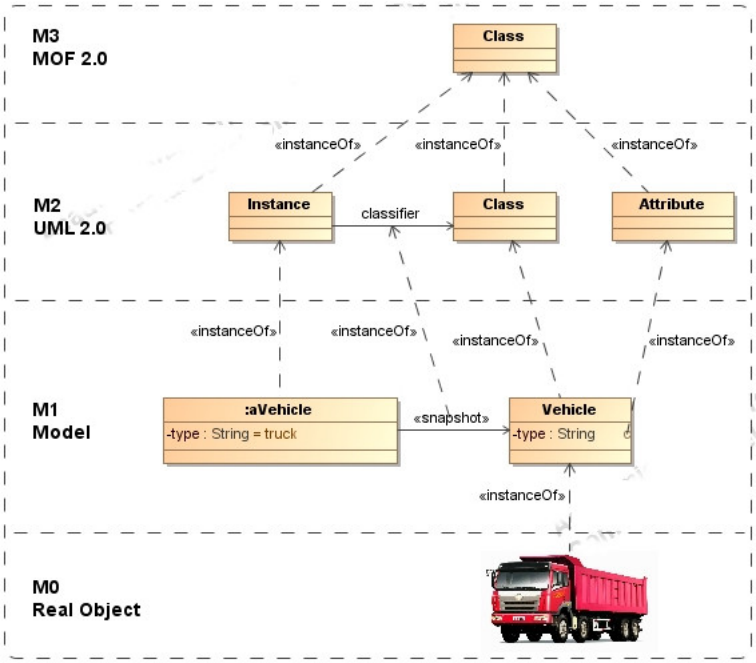
Μία ολοκληρωμένη πρόταση για μοντελο-κεντρική προσέγγιση στη μετάβαση από μία μορφή μοντέλου σε μία άλλη είναι η MDA [20], που προτάθηκε από τον οργανισμό OMG το 2001. Η MDA είναι μία προσέγγιση σχεδίασης για την ανάπτυξη συστημάτων λογισμικού. Παρέχει ένα σύνολο από οδηγίες για τη δόμηση των προδιαγραφών, οι οποίες εκφράζονται σαν μοντέλα, διαχωρίζοντας (α) τις λειτουργικές προδιαγραφές του συστήματος, που αποτυπώνονται σε ένα Μοντέλο Ανεξάρτητο Πλατφόρμας (Platform Independent Model) (PIM), από (β) την υλοποίηση της λειτουργικότητας σε μία συγκεκριμένη τεχνολογική πλατφόρμα, που αποτυπώνεται σε ένα Μοντέλο Συγκεκριμένης Πλατφόρμας (Platform Specific Model) (PSM), όπως απεικονίζεται και στην εικόνα 2.12.



Σχήμα 2.12: MDA PIMs και PSMs

Το υπόβαθρο που επιτρέπει τον ορισμό μετα-μοντέλων για τα PIMs και τα PSMs και, στη συνέχεια, υποστηρίζει τη δυνατότητα μετάβασης από το ένα στο άλλο είναι το μετα-μετα-μοντέλο MOF [21].

Το σχήμα 2.13 αποτυπώνει τα νοητά επίπεδα, που διαμορφώνονται στη μοντελοποίηση με το υπόβαθρο MOF. Σε γενικές γραμμές, με το MOF (επίπεδο M3) δίνεται η δυνατότητα ορισμού των βασικών στοιχείων (elements) ενός πεδίου μοντελοποίησης (modelling domain) με τα χαρακτηριστικά τους και τις μεταξύ τους συσχετίσεις, ως ένα μετα-μοντέλο (επίπεδο M2) για το πεδίο αυτό. Για παράδειγμα, το μετα-μοντέλο της UML περιλαμβάνει έννοιες (στιγμιότυπα της κλάσης του MOF), όπως η κλάση, το στιγμιότυπο και το χαρακτηριστικό. Έτσι, είναι δυνατός ο ορισμός μοντέλων (επίπεδο M1) στο εξεταζόμενο πεδίο. Στο παράδειγμά μας, μπορούν να οριστούν μοντέλα UML. Κάθε στοιχείο του μοντέλου UML (επίπεδο M1) είναι στιγμιότυπο των στοιχείων του μετα-μοντέλου UML (επίπεδο M2). Τα μοντέλα όμως ορίζονται για την μοντελοποίηση αντικειμένων του πραγματικού κόσμου (επίπεδο M0), τα οποία μπορούν να θεωρηθούν στιγμιότυπα της αντίστοιχης κλάσης του μοντέλου (επίπεδο



Σχήμα 2.13: Τα επίπεδα μοντελοποίησης με MOF

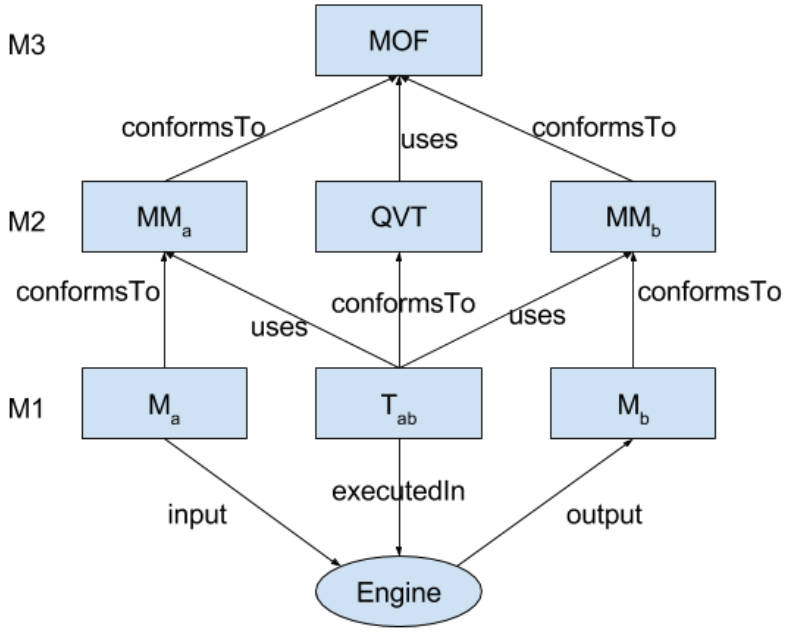
M1).

Στα πλαίσια της MDA, τα PIMs και τα PSMs είναι μοντέλα που ορίζονται βάσει ενός ή διαφορετικών μετα-μοντέλων και η μετάβαση από ένα PIM σε ένα PSM γίνεται μέσω ενός μετασχηματισμού. Ο μετασχηματισμός λαμβάνει ως είσοδο μοντέλα του ενός μετα-μοντέλου και δημιουργεί τα αντίστοιχα μοντέλα του μετα-μοντέλου προορισμού. Για τον ορισμό τέτοιων μετασχηματισμών έχει δημιουργηθεί το πρότυπο OMG QVT [22], το οποίο βασίζεται στο MOF. Το όνομα του προτύπου υποδηλώνει την ενιαία αντιμετώπιση του OMG όσον αφορά επερωτήσεις (queries), όψεις (views) και μετασχηματισμούς (transformations) σε μοντέλα, υπό το πρίσμα ότι κάθε ενέργεια σε ένα μοντέλο μπορεί να θεωρηθεί ως ένας "ειδικός" μετασχηματισμός.

Το σχήμα 2.14 αποτυπώνει διαγραμματικά τις έννοιες που εμπλέκονται σε ένα μετασχηματισμό QVT. Το MOF έχει κεντρικό ρόλο καθώς η QVT βασίζεται σε αυτή την υποδομή, ενώ τα μετα-μοντέλα αφετηρίας (MMa) και προορισμού (MMb) συμμορφώνονται με αυτό (conformsTo). Έτσι, είναι δυνατό να ορίζονται, με την QVT, μετασχηματισμοί, που βασίζονται στα μετα-μοντέλα αφετηρίας (MMa) και προορισμού (MMb). Οι μετασχηματισμοί, δηλαδή, αποκωδικοποιούν τις επιτρεπτές δομές μοντέλων και ορίζουν σχέσεις μεταξύ τους. Επομένως, μία μηχανή εκτέλεσης μετασχηματισμών QVT μπορεί να λαμβάνει ως είσοδο ένα μοντέλο (Ma), που συμμορφώνεται με το μετα-μοντέλο αφετηρίας (MMa), ένα μετασχηματισμό QVT (Tab), βασισμένο στα μετα-μοντέλα αφετηρίας (MMa) και προορισμού (MMb) και να παράγει το αντίστοιχο μοντέλο προορισμού (Mb).

Το πρότυπο QVT ορίζει τρεις γλώσσες μετασχηματισμού, σύμφωνα και με την αρχιτεκτονική του σχήματος 2.15:

- QVT-Operational (QVT-O): Γλώσσα συμπερασμών που επιτρέπει τον ορισμό μονόδρομων μετασχηματισμών με διαδικαστικό τρόπο, ως ένα σύνολο συναρτήσεων, που εκτελούνται

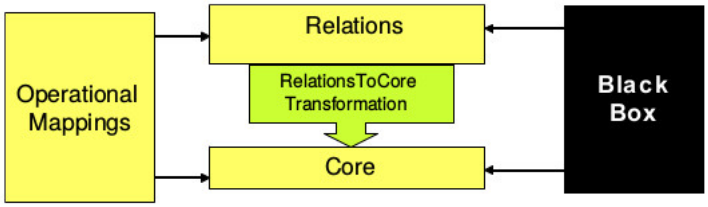


Σχήμα 2.14: Μετασχηματισμοί με OMG MOF και QVT

εξετάζοντας τμήματα του μοντέλου αφετηρίας και παράγοντας τα αντίστοιχα τμήματα του μοντέλου προορισμού.

- QVT-R: Δηλωτική γλώσσα που επιτρέπει τον ορισμό τόσο μονόδρομων, όσο και αμφίδρομων μετασχηματισμών, ως ένα σύνολο σχέσεων που πρέπει να ισχύουν μεταξύ στοιχείων των μοντέλων αφετηρίας και προορισμού. Οι σχέσεις εφαρμόζονται επιλεκτικά (υπό προϋποθέσεις) σε στοιχεία των μοντέλων αφετηρίας και προορισμού και όταν εφαρμοστούν έχουν σαν συνέπεια την απόπειρα εφαρμογής και άλλων σχέσεων.

- QVT-Core (QVT-C): Δηλωτική γλώσσα, που είναι σχεδιασμένη ώστε να είναι απλή και να αποτελεί τον προορισμό της μετάφρασης από QVT-R. Ωστόσο, η QVT-C δεν είναι όσο εκφραστική είναι η QVT-R, οπότε στο σχήμα 2.15 δεν διατηρείται η σημασιολογία κατά τη μετάβαση από την QVT-R στην QVT-C.



Σχήμα 2.15: Αρχιτεκτονική της QVT (Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification Version 1.1 - January 2011, σελ. 9)

Ο μηχανισμός QVT-BlackBox έχει προβλεφθεί για την κλήση λειτουργιών μετασχηματισμών άλλων γλωσσών (π.χ. EXtensible Stylesheet Language Transformations (XSLT) [23] ή Γλώσσα Επερωτήσεων XML (XML Query Language) (XQuery) [24]). Επίσης, η QVT αξιοποιεί την OCL [25], επεκτείνοντάς την με δυνατότητες συμπερασμού (imperative features).

2.1.4 Επικύρωση Μετασχηματισμών

Οι μετασχηματισμοί μοντέλων αποτελούν βασικό κομμάτι της βασισμένης-σε-μοντέλα ανάπτυξης συστημάτων. Επομένως, η ορθότητα και η ποιότητα των μετασχηματισμών έχουν σημαντικό αντίκτυπο στη διαδικασία ανάπτυξής συνολικά. Ο έλεγχος και η διασφάλιση της ορθότητας των μετασχηματισμών είναι ένα αντικείμενο που έχει απασχολήσει την ερευνητική κοινότητα. Στο [26] γίνεται μία ανασκόπηση της τρέχουσας κατάστασης στο χώρο. Η ορθότητα των μετασχηματισμών μπορεί να ελεγχθεί ως προς διάφορες παραμέτρους:

- **Τερματισμός**: Ο μετασχηματισμός θα πρέπει να ολοκληρώνεται, παράγοντας το μοντέλο προορισμού για κάθε μοντέλο αφετηρίας.
- **Ντετερμινισμός**: Για κάθε μοντέλο αφετηρίας θα παράγεται πάντα το ίδιο μοντέλο προορισμού σε πολλαπλές εκτελέσεις.
- **΄Ελεγχος Τύπων (Typing)**: Ο μετασχηματισμός είναι συντακτικά ορθός, ως προς τη γλώσσα μετασχηματισμού.
- **Διατήρηση Σημασιολογίας Εκτέλεσης**: Ο μετασχηματισμός θα πρέπει να συμπεριφέρεται σύμφωνα με τη σημασιολογία της γλώσσας μετασχηματισμού.
- **Συμμόρφωση και Τύποι Μοντέλων**: Τα παραγόμενα μοντέλα θα πρέπει να είναι συντακτικά ορθά, σύμφωνα με το μετα-μοντέλο προορισμού.
- **Δομική Αντιστοίχηση**: Παραγωγή του μοντέλου προορισμού μεταφέροντας στοιχεία και περιορισμούς της συνολικής δομής του μοντέλου αφετηρίας (όχι μόνο επιμέρους δομικούς μετασχηματισμούς).
- **Σημασιολογική Ορθότητα**: Διασφάλιση της σημασιολογικής αντιστοίχισης του παραγόμενου μοντέλου με το μοντέλο αφετηρίας.
- **Συντακτική Πληρότητα (Completeness)**: Διασφάλιση ότι ο μετασχηματισμός καλύπτει όλα τα δυνατά μοντέλα αφετηρίας/προορισμού.

Η επικύρωση των μετασχηματισμών μπορεί να γίνει με κάποια από τις ακόλουθες μεθόδους:

- **Εξαγωγή λογικών συμπερασμάτων (Απόδειξη θεωρημάτων)**: Χρήση μαθηματικής αναπαράστασης του συστήματος και των χαρακτηριστικών που πρέπει να επικυρωθούν, καθώς και μίας λογικής στο σημασιολογικό πεδίο, που επιτρέπει συμπερασμούς στην αναπαράσταση, οδηγώντας από υποθέσεις σε συμπεράσματα.
- **Θεωρητικός έλεγχος μοντέλων**: Γίνεται επίσης με μαθηματική αναπαράσταση του συστήματος. Οι αποδείξεις γίνονται με συστηματική, εξαντλητική εξερεύνηση του μαθηματικού μοντέλου.
- **Δοκιμές**: Υλοποιούνται με διαδοχικές εκτελέσεις του μετασχηματισμού σε μοντέλα που παρήχθησαν βάσει κάποιας στρατηγικής. Με αυτή τη μέθοδο μπορούν να αναδειχθούν προβλήματα, αλλά όχι απαραίτητα και οι αιτίες τους.

- **Στατική ανάλυση (static analysis)**: Γίνεται με μαθηματική ανάλυση του μετασχηματισμού (βασισμένου κυρίως σε γράφους), ως μεταβολή από μία μορφή σε μία άλλη, που πρέπει να διατηρεί συγκεκριμένα χαρακτηριστικά (π.χ. σημασιολογία).
- **Εκ κατασκευής (by construction)**: Εφαρμόζεται όταν η προδιαγραφή του μετασχηματισμού έχει δοθεί σε τέτοια μορφή, που εκ κατασκευής εξασφαλίζεται η ορθότητα.

2.2 Προσομοίωση Συστημάτων

Προσομοίωση είναι η τεχνητή αναπαράσταση της λειτουργίας ενός πραγματικού συστήματος ή διαδικασίας [27] για κάποιο χρονικό διάστημα. Η εκτέλεση της προσομοίωσης προϋποθέτει την ανάπτυξη ενός μοντέλου, που αναπαριστά τα κύρια χαρακτηριστικά και τη συμπεριφορά του συγκεκριμένου υπαρκτού ή θεωρητικού συστήματος ή διαδικασίας. Το μοντέλο αναπαριστά το σύστημα, ενώ η προσομοίωση τη λειτουργία του συστήματος για κάποιο χρονικό διάστημα. Η προσομοίωση χρησιμοποιείται σε διάφορες περιπτώσεις, όπως βελτιστοποίηση απόδοσης, διαχείριση ασφάλειας, δοκιμές, εκπαίδευση, για να αποτιμηθεί η συμπεριφορά του συστήματος σε συγκεκριμένες συνθήκες εκτέλεσης της προσομοίωσης. Η προσομοίωση χρησιμοποιείται και όταν το πραγματικό σύστημα δεν έχει κατασκευαστεί ακόμα ή δεν είναι διαθέσιμο ή η πρόσβαση σε αυτό δεν είναι ενδεδειγμένη, π.χ. λόγω θεμάτων ασφαλείας.

Σημεία-κλειδιά για την προσομοίωση είναι η κατάλληλη προδιαγραφή χαρακτηριστικών και συμπεριφοράς, η ισορροπημένη χρήση απλοποιήσεων και υποθέσεων στο βαθμό που διευκολύνουν την ανάδειξή τους και χωρίς να επηρεάζουν την εγκυρότητα των αποτελεσμάτων της προσομοίωσης.

Στα πλαίσια αυτής της διατριβής μας απασχολεί η προσομοίωση με υπολογιστή (computer simulation), δηλαδή η λειτουργία ενός μοντέλου συστήματος σε ένα υπολογιστικό περιβάλλον προσομοίωσης με συγκεκριμένες συνθήκες.

2.2.1 Τύποι Μοντέλων και Είδη Προσομοίωσης

Η προσομοίωση διαφοροποιείται ανάλογα με τις ιδιαιτερότητες του τρόπου αναπαράστασης των μοντέλων προσομοίωσης [10]. Έτσι, μπορούμε να διακρίνουμε σε:

- Προσομοίωση με **στοχαστικά** ή **ντετερμινιστικά** μοντέλα: Στα στοχαστικά μοντέλα η συμπεριφορά ορισμένων στοιχείων του μοντέλου συστήματος χαρακτηρίζονται από το στοιχείο της τυχαιότητας. Σε αυτές τις περιπτώσεις χρησιμοποιούνται συναρτήσεις παραγωγής τυχαίων δειγμάτων, σύμφωνα με την κατανομή, η οποία εκφράζει καλύτερα την τυχαία συμπεριφορά. Στα ντετερμινιστικά μοντέλα δεν υπάρχει καθόλου το στοιχείο της τυχαιότητας, που είναι και η τετριμμένη περίπτωση του στοχαστικού μοντέλου.
- Προσομοίωση με **σταθερά** ή **δυναμικά** μοντέλα: Τα σταθερά μοντέλα παρουσιάζουν σταθερή συμπεριφορά καθώς εξελίσσεται ο χρόνος προσομοίωσης. Τα δυναμικά μοντέλα περιγράφουν συστήματα, τα οποία αποκτούν διαφορετική κατάσταση με την πάροδο του

χρόνου προσομοίωσης, με αποτέλεσμα να μεταβάλλεται και η συμπεριφορά τους. Και εδώ, ένα σταθερό μοντέλο μπορεί να θεωρηθεί ως η τετριμμένη περίπτωση δυναμικού.

- Προσομοίωση με μοντέλα, όπου η συμπεριφορά ορίζεται με **συνεχή** ή **διακριτό** τρόπο ή πιο συγκεκριμένα με **διακριτά συμβάντα**: Στα μοντέλα με συνεχή τρόπο μοντελοποίησης, η συμπεριφορά εκφράζεται με διαφορικές εξισώσεις, οι οποίες αποτυπώνουν πώς η κατάσταση του μοντέλου μεταβάλλεται κατά την πάροδο του χρόνου με συνεχή τρόπο. Αντίθετα, στα μοντέλα διακριτών συμβάντων η κατάσταση του μοντέλου τροποποιείται στιγμιαία σε διακριτές χρονικές στιγμές, σύμφωνα με υπολογισμούς, που λαμβάνουν υπόψη την τρέχουσα κατάσταση, το χρόνο που έχει παρέλθει και το είδος του συμβάντος. Μπορούν να υπάρξουν και συνδυαστικές περιπτώσεις διακριτών-συνεχών μοντέλων προσομοίωσης, όπου διακριτά συμβάντα μπορούν να μεταβάλλουν τιμές συνεχών μεταβλητών κατάστασης ή να αλλάξουν την ισχύουσα σχέση μεταξύ συνεχών μεταβλητών κατάστασης, ενώ και υπέρβαση ενός ορίου (threshold) μιας συνεχούς μεταβλητής κατάστασης θα μπορούσε να πυροδοτήσει ένα διακριτό συμβάν.

Επίσης, η προσομοίωση μπορεί να διαφοροποιηθεί ανάλογα με τον τρόπο εκτέλεσης: **σειριακό** ή **παράλληλο/κατανεμημένο**. Στο σειριακό τρόπο εκτέλεσης, υπάρχει κεντρικός έλεγχος και εκτέλεση της προσομοίωσης. Επομένως, η προώθηση του χρόνου προσομοίωσης, ο υπολογισμός του επόμενου συμβάντος και η μεταβολή των μεταβλητών κατάστασης του μοντέλου προσομοίωσης εκτελούνται σε έναν επεξεργαστή σύμφωνα με κάποιο σειριακό αλγόριθμο. Στην περίπτωση της παράλληλης/κατανεμημένης εκτέλεσης της προσομοίωσης είναι δυνατό η εκτέλεση των επιμέρους λειτουργιών της προσομοίωσης (η προώθηση του χρόνου προσομοίωσης, ο υπολογισμός του επόμενου συμβάντος και η μεταβολή των μεταβλητών κατάστασης) να κατανεμηθεί σε διαφορετικούς επεξεργαστές. Εναλλακτικός τρόπος κατανομής της εκτέλεσης της προσομοίωσης είναι η κατάτμηση του μοντέλου σε υπο-μοντέλα τα οποία προσομοιώνονται αυτόνομα μεν, αλλά με μηχανισμούς συγχρονισμού δε, σε διαφορετικούς επεξεργαστές.

Στα πλαίσια της διατριβής, ασχολούμαστε με **δυναμικά, στοχαστικά μοντέλα προσομοίωσης διακριτών συμβάντων που εκτελούνται σειριακά**.

2.2.2 Περιβάλλοντα Εκτέλεσης Προσομοίωσης

Η εκτέλεση προσομοίωσης σε υπολογιστή γίνεται με την εκτέλεση προγράμματος το οποίο υλοποιεί την αναπαράσταση της λειτουργίας του συστήματος. Στη βασισμένη σε συμβάντα προσομοίωση, το πρόγραμμα παρακολουθεί και προωθεί το χρόνο προσομοίωσης, υπολογίζει το επόμενο συμβάν και μεταβάλει αναλόγως τις μεταβλητές κατάστασης. Το πρόγραμμα αυτό μπορεί να είναι ειδικού σκοπού πρόγραμμα, κατασκευασμένο ώστε να υλοποιεί τον απαιτούμενο αλγόριθμο για ένα συγκεκριμένο σύστημα. Ωστόσο, με την ανάπτυξη και τη διάδοση της προσομοίωσης έχουν δημιουργηθεί πολλά περιβάλλοντα προσομοίωσης τα οποία διαχωρίζουν την υλοποίηση του αλγορίθμου εκτέλεσης της προσομοίωσης από το μοντέλο του συγκεκριμένου, αλλά διαφορετικού κάθε φορά μοντέλου προσομοίωσης. Ο διαχωρισμός αυτός έχει τα εξής πλεονεκτήματα:

- Ο μηχανικός προσομοίωσης επικεντρώνει στα χαρακτηριστικά του μοντέλου και στις συνθήκες προσομοίωσης και όχι σε θέματα και δυσκολίες που αφορούν την υλοποίηση του προγράμματος προσομοίωσης.

- Ευνοείται η βελτιστοποίηση του προγράμματος εκτέλεσης της προσομοίωσης, αφού το ίδιο χρησιμοποιείται σε πολλές εφαρμογές προσομοίωσης σε διαφορετικά πεδία. Έτσι, αναδεικνύονται οι όποιες αδυναμίες και αντιμετωπίζονται, ενώ βελτιώνονται και θέματα απόδοσης και ακρίβειας των παραγόμενων αποτελεσμάτων.

- Δίνεται η δυνατότητα γενικής υλοποίησης, δοκιμής και σύγκρισης στην πράξη διαφορετικών προσεγγίσεων προσομοίωσης σε διαφορετικά πεδία εφαρμογής.

Επομένως, δεδομένων των πλεονεκτημάτων και της διάδοσης των περιβαλλόντων εκτέλεσης προσομοίωσης, αυτά αξιοποιούνται συνήθως στην πλειονότητα των γενικών περιπτώσεων, όπου απαιτείται προσομοίωση. Όταν υπάρχουν συγκεκριμένες και αυστηρές συνθήκες και απαιτήσεις για τον τρόπο, την ακρίβεια, την απόδοση και άλλες παραμέτρους της εκτέλεσης της προσομοίωσης (π.χ. στρατιωτικές/διαστημικές εφαρμογές ή συστήματα πραγματικού χρόνου), τότε μπορεί τα περιβάλλοντα εκτέλεσης προσομοίωσης να μην μπορούν να εγγυηθούν τις απαιτούμενες συνθήκες και να προκρίνεται η ανάπτυξη προγραμμάτων ειδικού σκοπού.

Ένα περιβάλλον εκτέλεσης προσομοίωσης συνήθως:

- υποστηρίζεται από μία θεωρητική προσέγγιση προσομοίωσης

- υλοποιεί τη θεωρητική προσέγγιση πλήρως ή με κάποιους περιορισμούς

- παρέχει μία ή περισσότερες γλώσσες ή σημειολογίες για τον ορισμό των μοντέλων προσομοίωσης

- παρέχει ένα ή περισσότερα περιβάλλοντα εκτέλεσης προσομοίωσης (προσομοιωτές), οι οποίοι μπορούν να αξιοποιούν έγκυρα μοντέλα προσομοίωσης

Η αναφερόμενη πολλαπλότητα στον τρόπο αναπαράστασης των μοντέλων προσομοίωσης του ίδιου πλαισίου προσομοίωσης, σε συνδυασμό με τους πολλούς και διαφορετικούς προσομοιωτές, εισάγει μία ασάφεια σχετικά με τις δυνατότητες του πλαισίου προσομοίωσης, ενώ δεν επιτρέπει τη δια-λειτουργικότητα μεταξύ των προσομοιωτών και περιορίζει τη δυνατότητα επαναχρησιμοποίησης των μοντέλων προσομοίωσης. Για παράδειγμα, στο DEVS [28], υπάρχει ένα θεωρητικό υπόβαθρο, βασισμένο σε θεωρία συνόλων, αλλά πληθώρα προσομοιωτών, όπως DEVS-C++ [29], DEVS-Java [30], cell-DEVS [31], DEVS-RMI [32], XLSC [33]. Καθένας από τους παραπάνω προσομοιωτές έχει δική του γλώσσα περιγραφής μοντέλων συστημάτων (πρόγραμμα σε Java, C++ ή XML αναπαράσταση).

Σε αυτό το κεφάλαιο αναλύθηκαν έννοιες σχετικά με τη βασισμένη-σε-μοντέλα ανάπτυξη συστημάτων και την προσομοίωση. Κυρίαρχη και στις δύο περιπτώσεις είναι η έννοια του *μοντέλου*, αλλά για διαφορετικούς λόγους. Στην πρώτη (MBSE), το μοντέλο πρέπει να περιέχει το ευρύτερο δυνατό σύνολο πληροφοριών σχετικά με το σύστημα, ώστε να είναι δυνατή η

αξιοποίησή του σε μεγάλο εύρος δραστηριοτήτων ανάπτυξης του συστήματος. Αυτό πρέπει να γίνεται χωρίς να χάνει τη συνοχή του και διευκολύνοντας την ομαλή μετάβαση από τη μία δραστηριότητα στην επόμενη. Στη δεύτερη περίπτωση (προσομοίωση), η ακρίβεια και η πιστότητα του μοντέλου προσομοίωσης καθορίζει το βαθμό επιτυχίας της αξιοποίησής της. Ένα μοντέλο, που έχει οριστεί μερικώς ή με αποκλίσεις από το πραγματικό σύστημα είναι σίγουρο ότι θα οδηγήσει σε λανθασμένα συμπεράσματα.

Αναμενόμενο είναι λοιπόν να επιζητεί η επιστημονική και τεχνολογική κοινότητα την αξιοποίηση της προσομοίωσης στην ανάπτυξη συστημάτων, εντάσσοντάς την στην MBSE με τρόπο, που να βελτιώνει ουσιαστικά τους όρους εργασίας των σχεδιαστών συστημάτων. Να παρέχει, δηλαδή, προσιτούς και αξιόπιστους τρόπους χρήσης της προσομοίωσης για τη μελέτη των σχεδιαστικών λύσεων, αξιοποιώντας στο μεγαλύτερο δυνατό βαθμό το διαθέσιμο μοντέλο συστήματος, όπως αυτό έχει διαμορφωθεί από προηγούμενες δραστηριότητες ανάπτυξης. Στη συνέχεια, αναλύονται σχετικές ερευνητικές προσπάθειες, που αφορούν είτε την προσομοίωση μοντέλων ορισμένων με SysML, είτε την αυτοματοποίηση της παραγωγής εκτελέσιμων μοντέλων ή προγραμμάτων προσομοίωσης.

Κεφαλαιο 3

Σχετικες Ερευνητικες Προσπαθειες

3.1 Αξιοποίηση Μοντέλων Συστημάτων στην Προσομοίωση

Η SysML είναι η γλώσσα που προτείνεται από το OMG για τη μοντελοποίηση συστημάτων, επιτρέποντας τον ορισμό απλών και σύνθετων συστημάτων με γραφικό τρόπο, μέσω εργαλείων μοντελοποίησης που υποστηρίζουν την πρότυπη γλώσσα UML και τις επεκτάσεις της. Η SysML υποστηρίζει την περιγραφή της δομής, της συμπεριφοράς του συστήματος, αλλά και των απαιτήσεων κατά τη λειτουργία του.

Δεδομένης της διάδοσης και της σημασίας της επικύρωσης μοντέλων συστημάτων μέσω προσομοίωσης, η ανάγκη για ολοκλήρωση εργαλείων μοντελοποίησης με SysML και εργαλείων προσομοίωσης είναι προφανής. Προς αυτή την κατεύθυνση έχουν γίνει αρκετές προσπάθειες τόσο από την ερευνητική κοινότητα, όσο και από τον επιχειρηματικό κόσμο που δραστηριοποιείται στο χώρο της ανάπτυξης συστημάτων και της προσομοίωσης. Στην πλειονότητα των σχετικών προτάσεων, τα μοντέλα SysML ορίζονται σε ένα εργαλείο μοντελοποίησης και εξάγονται σε κάποια μορφή XML. Στη συνέχεια μετασχηματίζονται σε μοντέλα προσομοίωσης για κάποιο συγκεκριμένο περιβάλλον προσομοίωσης και προωθούνται σε αυτό για εκτέλεση της προσομοίωσης.

Η SysML υποστηρίζει μία ποικιλία διαγραμμάτων για την περιγραφή της δομής και χαρακτηριστικών της συμπεριφοράς του συστήματος, που συνήθως απαιτούνται για την εκτέλεση προσομοίωσης. Ανάλογα με τη φύση και συγκεκριμένα χαρακτηριστικά του πεδίου εφαρμογής των συστημάτων υπό εξέταση, υπάρχει μία ποικιλία προσεγγίσεων για την προσομοίωση μο-

ντέλων, που έχουν οριστεί με SysML, αξιοποιώντας διαφορετικά διαγράμματα της γλώσσας. Μία μέθοδος για την προσομοίωση της συμπεριφοράς συνεχών συστημάτων χρησιμοποιώντας μαθηματική προσομοίωση προτάθηκε από τους Peak et al [34], αξιοποιώντας τα PDs, τα οποία επιτρέπουν την περιγραφή σύνθετων μαθηματικών εξισώσεων. Τα μοντέλα συστημάτων προσομοιώνονται με χρήση συνθέσιμων (composable) αντικειμένων (COBs) [35]. Πρέπει να σημειωθεί ότι, σε κάθε περίπτωση, τα μοντέλα SysML πρέπει να ορίζονται με τρόπο που να διευκολύνει την προσομοίωση τους [36]. Συνήθως, χρησιμοποιούνται κατάλληλα UML προφίλ για την ενσωμάτωση χαρακτηριστικών, που αφορούν ένα συγκεκριμένο είδος προσομοίωσης, στα μοντέλα συστήματος. Στην έρευνα που παρουσιάζεται από τους Paredis et al [37], η προσομοίωση εκτελείται με χρήση της Modelica. Για τη διασφάλιση της δυνατότητας παραγωγής ενός ολοκληρωμένου μοντέλου Modelica με ακρίβεια από ένα μοντέλο SysML, προτείνεται ένα αντίστοιχο προφίλ για τον εμπλουτισμό μοντέλων SysML με χαρακτηριστικά που αφορούν προσομοίωση. Σε αυτά τα πλαίσια αξιοποιήθηκαν οι περιορισμοί που μπορούν να εφαρμοστούν σε συστατικά λογισμικού και τα παραμετρικά διαγράμματα (PD), ώστε να μοντελοποιείται η συμπεριφορά του συστήματος με τις μαθηματικές εξισώσεις [38].

Το *SysML4Modelica* είναι ένα προφίλ, που επιτρέπει το σχολιασμό των μοντέλων SysML με χαρακτηριστικά της Modelica και της προσομοίωσης σε ένα περιβάλλον εκτέλεσης προσομοίωσης Modelica, αξιοποιώντας βιβλιοθήκες εκτελέσιμων μοντέλων προσομοίωσης, όταν αυτά είναι διαθέσιμα. Μάλιστα, ο ορισμός του προφίλ *SysML4Modelica* τυγχάνει της υποστήριξης του OMG [39]. Η γλώσσα QVT-O έχει υιοθετηθεί για το μετασχηματισμό των μοντέλων SysML σε μοντέλα Modelica, ενώ σχετικές προδιαγραφές και λογισμικό έχουν γίνει διαθέσιμα πρόσφατα για χρήση από την κοινότητα.

Οι προαναφερθείσες προσεγγίσεις είναι κατάλληλες για πεδία συστημάτων, που προσομοιώνονται με μοντέλα προσομοίωσης συνεχούς συμπεριφοράς. Ωστόσο, η προσομοίωση συστημάτων διακριτών συμβάντων (discrete event systems) είναι επίσης δυνατή για μοντέλα SysML, όπου η συμπεριφορά περιγράφεται σε διαγράμματα τύπου activity, sequence ή state.

Στην έρευνα των McGinnis και Ustun [40], μοντέλα συστήματος, ορισμένα σε SysML μεταφράζονται ώστε να μπορούν να προσομοιωθούν στο λογισμικό Arena. Τα μοντέλα SysML δεν εμπλουτίζονται με χαρακτηριστικά προσομοίωσης, ενώ δίνεται έμφαση στη δομή των συστημάτων, έναντι της συμπεριφοράς τους. Τα SysML μοντέλα εξάγονται από εργαλεία μοντελοποίησης UML, αξιοποιώντας έννοιες και εργαλεία της MDA. Στη συνέχεια, μετασχηματίζονται σε μοντέλα Arena, τα οποία πρέπει να εμπλουτιστούν με χαρακτηριστικά συμπεριφοράς των μοντέλων για να γίνουν εκτελέσιμα. Αυτό μπορεί να γίνει από το μηχανικό συστημάτων είτε με τη χρήση υπαρχόντων βιβλιοθηκών μοντέλων προσομοίωσης, είτε με τη συγγραφή κώδικα προσομοίωσης για Arena.

Στη μελέτη των Batarseh και McGinnis [41], παρουσιάστηκε το προφίλ *SysML4Arena* για τον ορισμό μοντέλων συστημάτων που εξάγονται και μετασχηματίζονται με Γλώσσα Μετασχηματισμών ATLAS (ATLAS Transformation Language) (ATL) σε μοντέλα προσομοίωσης για Arena. Αυτή η προσέγγιση δίνει έμφαση σε γλώσσες συγκεκριμένου πεδίου εφαρμογών και υπάρχουσες βιβλιοθήκες συστατικών λογισμικού.

Στην έρευνα των Wang και Dagli [42] προτείνεται η αξιοποίηση Colored Petri Nets για την προσομοίωση μοντέλων SysML. Τα τελευταία μπορούν να προσομοιωθούν με προσομοίωση διακριτών συμβάντων, μέσω Petri Nets, εφόσον η συμπεριφορά του συστήματος έχει περιγραφεί με διαγράμματα activity και sequence.

Η εταιρεία Intercax προτείνει το SLIM [43], ένα εμπορικό περιβάλλον συνεργαρτικής και βασισμένης σε μοντέλα μηχανικής συστημάτων. Με το SLIM αξιοποιείται η SysML, από τα αρχικά στάδια της ανάπτυξης συστημάτων, ως εξωτερικό επίπεδο για την ενορχήστρωση των δραστηριοτήτων ανάπτυξης. Διαφορετικές δραστηριότητες της διαδικασίας ανάπτυξης συστημάτων καλύπτονται κυρίως μέσω ανάπτυξης plugins για τις εφαρμογές μοντελοποίησης SysML, δηλαδή όχι μέσω ανοιχτών προσεγγίσεων βασισμένων σε πρότυπα. Το ModelCenter [44], ένα εμπορικό εργαλείο της εταιρείας Phoenix Integration, υιοθετεί μία παρεμφερή προσέγιση. Η πρόθεσή του είναι να ολοκληρώσει μοντέλα συστημάτων, αξιοποιώντας τη SysML, και να τα προσομοιώσει χρησιμοποιώντας συγκεκριμένα περιβάλλοντα προσομοίωσης, όπως η Matlab. Το ModelCenter χρησιμοποιεί κυρίως τα PDs, όπου ειδικά constraint blocks οδηγούν σε ανάλυση black-box.

Τέλος, ολοκλήρωση με περιβάλλοντα προσομοίωσης όπως MATLAB/Simulink, Mathematica και OpenModelica παρέχεται σε αρκετά εμπορικά εργαλεία. Όμως, σε αυτές τις περιπτώσεις τα περιβάλλοντα προσομοίωσης χρησιμοποιούνται ως εργαλεία επίλυσης μαθηματικών προβλημάτων και όχι ως μέθοδος επικύρωσης μοντέλων.

3.2 Αυτοματοποίηση Προσομοίωσης

Στην προηγούμενη ενότητα αναφέρθηκαν αρκετές από την πληθώρα προσεγγίσεων για την προσομοίωση μοντέλων συστημάτων. Εντούτοις, καμία από τις προαναφερθείσες προσεγγίσεις, οι οποίες στοχεύουν σε προσομοίωση διακριτών συμβάντων, δεν υποστηρίζει την πλήρως αυτοματοποιημένη παραγωγή κώδικα προσομοίωσης, ο οποίος να εκτελείται στο αντίστοιχο περιβάλλον προσομοίωσης. Η υποστήριξη αυτοματοποίησης είναι μικρότερη στις περιπτώσεις, όπου δεν αξιοποιούνται βιβλιοθήκες μοντέλων προσομοίωσης. Οι περισσότερες από αυτές υιοθετούν τον ορισμό ενός σχετιζόμενου με την προσομοίωση προφίλ για τον εμπλουτισμό των μοντέλων SysML με χαρακτηριστικά, που είναι απαραίτητα για την προσομοίωσή τους, σύμφωνα με την επιλεγμένη μεθοδολογία προσομοίωσης [40], ειδικά σε περιπτώσεις όπου η συμπεριφορά των μοντέλων περιγράφεται με SysML [42].

Επιπρόσθετα, η χρήση εργαλείων μετασχηματισμού μοντέλων, που βασίζονται στην MDA για την παραγωγή μοντέλων προσομοίωσης από μοντέλα συστήματος SysML, ενισχύεται όλο και περισσότερο. Το κύριο εμπόδιο στην υλοποίηση προσεγγίσεων αυτού του τύπου είναι η έλλειψη πρότυπων μετα-μοντέλων για τις μεθοδολογίες προσομοίωσης. Το μετα-μοντέλο αποτελεί σημείο αναφοράς σε μία τέτοια προσέγγιση και η έλλειψή του, πέρα από το κόστος για τον ορισμό ενός νέου μετα-μοντέλου για το περιβάλλον προσομοίωσης και του επιπρόσθετου κόστους ανάπτυξης λογισμικού, που θα καθιστούν τα μοντέλα εκτελέσιμα, περιορίζει

πλεονεκτήματα όπως δια-λειτουργικότητα σε πολλούς προσομοιωτές.

3.2.1 Αυτοματοποίηση Επαλήθευσης Χαρακτηριστικών Μοντέλων Συστημάτων Πραγματικού Χρόνου

Επικεντρώνοντας στο πεδίο των ενσωματωμένων (embedded) συστημάτων πραγματικού χρόνου, έχει παρουσιαστεί η TEPE [45], μία γλώσσα γραφικών εκφράσεων, η οποία βασίζεται στα parametric diagrams της SysML. Με αυτή είναι δυνατή η αναπαράσταση λειτουργικών και μη λειτουργικών απαιτήσεων με ένα τυπικό, αλλά μη προτυποποιημένο τρόπο, επιτρέποντας την αυτοματοποιημένη επαλήθευσή τους. Μια προσαρμοσμένη μεθοδολογία χρησιμοποιείται για τη συλλογή απαιτήσεων, ενώ ο σχεδιασμός του συστήματος βασίζεται σε SysML blocks και η συμπεριφορά σε state machines. Τέλος, η επαλήθευση γίνεται με χρήση της UPPAAL [46]. Αυτή η προσπάθεια υποστηρίζεται από ένα προσαρμοσμένο εργαλείο, το TTool, που μπορεί να επικοινωνεί με εργαλεία επαλήθευσης, που υλοποιούν reachability analysis και model-checking. Το DIPLODOCUS, μία μηχανή προσομοίωσης, που στοχεύει στο σχεδιασμό System-on-Chip, είναι ενσωματωμένη στο TTool. Το TTool υποστηρίζει επίσης και το AVATAR [47], ένα περιβάλλον βασισμένο στη SysML, που μπορεί να αποτυπώσει στοιχεία ασφάλειας και προστασίας στο ίδιο το μοντέλο SysML. Το AVATAR επικεντρώνει στο χειρισμό νέων απειλών κατά της ασφάλειας, διατηρώντας ένα υψηλό επίπεδο προστασίας σε κρίσιμα, ενσωματωμένα συστήματα. Η επαλήθευση γίνεται με το εργαλείο UPPAAL.

3.2.2 Αυτοματοποίηση Προσομοίωσης σε DEVS

Στην περίπτωση του DEVS έχουν γίνει διάφορες προσπάθειες υποστήριξης μετασχηματισμού μοντέλων προσομοίωσης σε εκτελέσιμο κώδικα προσομοίωσης. Το κίνητρο για αυτές τις προσπάθειες ήταν σε μεγάλο βαθμό η ύπαρξη πολλών προσομοιωτών DEVS, ανεπτυγμένων από διαφορετικές ερευνητικές ομάδες. Καθώς οι προσομοιωτές αναπτύχθηκαν ανεξάρτητα, προέκυψε εκ των υστέρων η ανάγκη δια-λειτουργικότητας μεταξύ τους. Έτσι, πολλές προσεγγίσεις στοχεύουν στον ορισμό τρόπων περιγραφής μοντέλων DEVS, βασισμένων σε XML, και στο μετασχηματισμό τους σε διαφορετικές γλώσσες προγραμματισμού [33, 48–51].

Στο [50] προτείνεται μία αποτύπωση XML, ως μία μέθοδος επικοινωνιακής ενοποίησης των οντοτήτων σε οποιοδήποτε αρχιτεκτονική Συστημάτων-από-Συστήματα (Systems-of-Systems). Στο [51] το πρόβλημα της δια-λειτουργικότητας αντιμετωπίζεται με τη εισαγωγή της DEVSML. Τα παραχθέντα μοντέλα επικυρώνονται με Document Type Definitions (DTDs) για ατομικά και συζευγμένα μοντέλα DEVS. Σε αυτή την περίπτωση, δεν δίνεται έμφαση στη συμπεριφορά των μοντέλων.

Σε κάθε περίπτωση, τα εκτελέσιμα μοντέλα είναι στις γλώσσες προγραμματισμού C++ ή Java και οι προτεινόμενες αναπαραστάσεις XML προσανατολίζονται στη χαμηλού επιπέδου περιγραφή του εκτελέσιμου κώδικα DEVS, συμπεριλαμβανομένων εντολών και αναθέσεων τιμών σε μεταβλητές. Στο [33] παρουσιάζεται η XLSC DEVS, μία γλώσσα XML για τη μοντελοποίηση ατομικών και συζευγμένων μοντέλων DEVS. Με την XLSC μπορούν να αποτυπωθούν τόσο η

δομή, όσο και η συμπεριφορά των μοντέλων, ενώ ένα XLSC μοντέλο μπορεί να προσομοιωθεί. Ένας πρωτότυπος διερμηνέας υλοποιήθηκε σε Java και αξιοποιήθηκε για την άμεση εκτέλεση μοντέλων. Σε αυτή την περίπτωση, η συμπεριφορά των ατομικών μοντέλων περιγράφεται σε XML, με χρήση ενός συνόλου εντολών χαμηλού επιπέδου, που είναι ωστόσο ανεξάρτητες από κάποια συγκεκριμένη γλώσσα προγραμματισμού.

Στο [48], προτάθηκε η DEVS-XML, ως μία ανεξάρτητη πλατφόρμας, βασισμένη σε XML μορφή περιγραφής μοντέλων DEVS. Μία περιγραφή DEVS-XML μπορεί να μετατρέπεται σε εκτελέσιμο κώδικα για υπάρχοντες προσομοιωτές DEVS, χρησιμοποιώντας μεταγλωττιστές, όπως αυτοί που προτείνονται στο [48] για τον προσομοιωτή DEVSJava και οι οποίοι υλοποιήθηκαν μόνο για τα συζευγμένα μοντέλα DEVS. Έτσι, δεν υποστηρίζεται ο μετασχηματισμός της συμπεριφοράς των μοντέλων DEVS. Η DEVS-XML προτάθηκε για την επίτευξη φορητότητας μοντέλων DEVS και την προαγωγή της δια-λειτουργικότητας μεταξύ διαφορετικών προσομοιωτών DEVS, ανεξάρτητα από τη γλώσσα προγραμματισμού υλοποίησης και τον τρόπο λειτουργίας (σε κατανεμημένο ή κεντρικό περιβάλλον). Ωστόσο, δεν υπάρχουν πολλά εργαλεία που να υποστηρίζουν την DEVS-XML μέχρι τώρα.

Μεταξύ των συμβατών με DEVS-XML εργαλείων, το πιο ολοκληρωμένο είναι αυτό που παρουσιάζεται στο [49]. Επί του παρόντος χρησιμοποιείται για το μετασχηματισμό κώδικα DEVSJava σε XML και αντίστροφα, για ένα υποσύνολο του φορμαλισμού DEVS, το Finite Deterministic DEVS (FD-DEVS) [52]. Η συγκεκριμένη έκδοση της DEVS-XML αναφέρεται ως XFD-DEVS και παρέχονται τα αντίστοιχα XSDs και ένα εργαλείο για το μετασχηματισμό.

Η DEVS-XML προσφέρει μία υψηλού επιπέδου αναπαράσταση συμπεριφοράς DEVS, βασισμένη στις μεταβάσεις καταστάσεων του συστήματος, κατά αντιστοιχία με τα SMDs της SysML. Επομένως, η αξιοποίηση της DEVS-XML ως ένας ενδιάμεσος προορισμός κατά το μετασχηματισμό μοντέλων SysML σε εκτελέσιμο κώδικα προσομοίωσης DEVS, φαίνεται να έχει κάποια προοπτική. Ωστόσο, τίθεται μία σειρά από θέματα. Πρώτον, η DEVS-XML δεν αντιμετωπίζει την σχέση μεταξύ τιμών των μεταβλητών κατάστασης και καταστάσεων [28]. Δεύτερον, για να είναι ανεξάρτητη από την υλοποίηση, η DEVS-XML είναι αρκετά γενική και, ως εκ τούτου, δεν χειρίζεται σύνθετες εκφράσεις τιμών με κάποιο συγκεκριμένο τρόπο. Τρίτον, η DEVS-XML είναι -όπως μαρτυράει και το όνομά της- η συντακτική προδιαγραφή μιας XML αναπαράστασης μοντέλων DEVS και όχι ένα μετα-μοντέλο για πιο πλούσια εννοιολογικά μοντέλα DEVS. Τέταρτον, στην DEVS-XML σπάνια αξιοποιούνται XML attributes στα elements, οδηγώντας σε υπερπληθυσμό XML elements και αναίτια σύνθετη δομή. Τέλος, δοκιμές και χρήση εργαλείων συμβατών με DEVS-XML εργαλείων είναι μάλλον δύσκολη, από πλευράς διαθεσιμότητας και λειτουργικότητας. Ως εκ τούτου, παραμένει η ανάγκη για μία πρότυπη αναπαράσταση μοντέλων DEVS, η οποία να είναι (α) συνεπής με τη θεωρία του DEVS, (β) έτοιμη να εκτελεστεί σε περιβάλλοντα προσομοίωσης DEVS, και (γ) συμβατή με το μετα-μοντέλο της SysML/UML (δηλαδή να είναι ορισμένη σε όρους MOF), για να υποβοηθείται ο μετασχηματισμός των μοντέλων SysML σε μοντέλα DEVS με χρήση υπαρχόντων εργαλείων, που υποστηρίζουν το πρότυπο MOF. Κατά τον ορισμό ενός μετα-μοντέλου, σύμφωνα με τα παραπάνω, θα μπορούσε να αξιοποιηθεί η προεργασία που έχει γίνει με τις διάφορες προτάσεις για αναπαραστάσεις μοντέλων

DEVS σε XML.

3.3 Ανοιχτά Ζητήματα

Συμπερασματικά, λαμβάνοντας υπόψη τα ουσιαστικά αντικείμενα που αντιμετωπίζουν οι σχετικές ερευνητικές προσπάθειες, μπορεί κανείς να συνοψίσει στα θέματα που έχουν καλυφθεί εξαντλητικά ή σε μεγάλο βαθμό και σε αυτά που δεν έχουν αναγνωριστεί ή δεν έχουν αντιμετωπιστεί επαρκώς.

Η διερεύνηση της βιβλιογραφίας δείχνει ότι το θέμα της προσομοίωσης μοντέλων συστημάτων έχει αναγνωριστεί, αλλά δεν έχει αντιμετωπιστεί πλήρως. Ενώ η SysML είναι η επικρατούσα γλώσσα περιγραφής συστημάτων, τα μοντέλα SysML δεν μπορούν εν γένει να προσομοιωθούν. Οι περισσότερες προσεγγίσεις είτε περιορίζονται μόνο στη δομή και απαιτούν την εξαγωγή και περαιτέρω επεξεργασία των μοντέλων συστημάτων, για να μπορέσουν να προσομοιωθούν, είτε περιορίζονται σε συγκεκριμένο πεδίο εφαρμογής. Σε κάθε περίπτωση, δεν υποστηρίζεται ο ορισμός του μοντέλου συστήματος με όλες τις απαιτούμενες για την προσομοίωσή πληροφορίες στο εργαλείο μοντελοποίησης συστημάτων.

Επομένως, δεν καθίσταται εφικτή η αυτοματοποίηση της προσομοίωσης μοντέλων συστημάτων, χωρίς πρόσθετες παρεμβάσεις και χρήση τρίτων εργαλείων μοντελοποίησης ή κωδικοποίησης προσομοίωσης. Κάτι τέτοιο περιορίζει τα περιθώρια βελτίωσης σε θέματα όπως αποδοτικότητα, ευχρηστία και αντιστοιχία μοντέλου συστήματος και μοντέλου προσομοίωσης. Επίσης, η αξιοποίηση των αποτελεσμάτων προσομοίωσης είτε δεν αναγνωρίζεται ως θέμα, είτε γίνεται στα πλαίσια κάποιου συγκεκριμένου εργαλείου προσομοίωσης, που καλύπτει συγκεκριμένα πεδία εφαρμογών.

Κυρίως, τα θέματα αυτά δεν αντιμετωπίζονται συνολικά στα πλαίσια μίας μεθοδολογίας με τρόπο ανοικτό και βασισμένο σε πρότυπα, που να επιτρέπει την αξιοποίηση εναλλακτικών συμβατών επιλογών, συνθέτοντας δυναμικές λύσεις με πλούσια λειτουργικότητα.

Κεφαλαιο 4

Προταση

4.1 Βασική Ιδέα

Δεδομένων των σχετικών ερευνητικών προσπαθειών και των ανοιχτών θεμάτων στον τομέα της αυτοματοποίησης της προσομοίωσης μοντέλων συστημάτων, η παρούσα διατριβή προτείνει μία προσέγγιση, που οδηγεί με τρόπο ανοικτό και συμβατό με τα σχετικά πρότυπα στην έγκυρη, αποδοτική αυτοματοποιημένη προσομοίωση μοντέλων συστημάτων, ανεξάρτητα από το πεδίο εφαρμογής και τη μεθοδολογία προσομοίωσης. Κύριος άξονας της προσέγγισης είναι η πλήρης και αποκλειστική αξιοποίηση του μοντέλου συστήματος, που έχει οριστεί σύμφωνα με την πρότυπη γλώσσα SysML, στην αποτύπωση των χαρακτηριστικών του συστήματος, συμπεριλαμβανομένων των χαρακτηριστικών απόδοσης, που καθιστούν εφικτή την προσομοίωση. Κατά αυτό τον τρόπο, απαλείφεται η απαίτηση για εκ των υστέρων παρεμβάσεις και συμπληρώσεις, που εισάγουν καθυστερήσεις και πιθανά λάθη στην παραγωγή του μοντέλου προσομοίωσης. Επομένως, η προσέγγιση οδηγεί σε μία νέα προτεινόμενη διαδικασία σχεδιασμού συστημάτων, η οποία περιλαμβάνει και την αποτίμηση της αναμενόμενης συμπεριφοράς του μοντέλου συστήματος, μέσω προσομοίωσης. Επίσης, προτείνεται ακόμα και η αξιοποίηση

των αποτελεσμάτων προσομοίωσης να ξεκινάει από το μοντέλο συστήματος, όπου εκτός από τις παραγόμενες εκτιμήσεις για τη συμπεριφορά του συστήματος, περιλαμβάνεται και πλήθος άλλων πληροφοριών.

Για την εφαρμογή της προτεινόμενης διαδικασίας, δηλαδή την παραγωγή και εκτέλεση μοντέλων προσομοίωσης με την αποκλειστική χρήση του μοντέλου συστήματος είναι απαραίτητη η ανάπτυξη και αξιοποίηση ενός πλαισίου, που προδιαγράφεται αναλυτικά στην ενότητα 4.3. Συνοπτικά, το πλαίσιο αυτό:

- εξασφαλίζει την πλήρη και ορθή αποτύπωση των χαρακτηριστικών που αφορούν την προσομοίωση, μεσώ κατάλληλου προφίλ UML/SysML. Έτσι διασφαλίζεται η δυνατότητα παραγωγής εκτελέσιμων μοντέλων προσομοίωσης.
- περιλαμβάνει το μετα-μοντέλο αναφοράς για τη χρησιμοποιούμενη μεθοδολογία προσομοίωσης.
- παρέχει τον απαιτούμενο μετασχηματισμό για την παραγωγή μοντέλων προσομοίωσης (σύμφωνα με το μετα-μοντέλο αναφοράς) από τα μοντέλα συστημάτων.
- προωθεί τα μοντέλα προσομοίωσης για εκτέλεση και παραγωγή αποτελεσμάτων σε συμβατά περιβάλλοντα εκτέλεσης προσομοίωσης.
- παρέχει τους απαιτούμενους μηχανισμούς για τη συμπλήρωση των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος.

Η ανάπτυξη και αξιοποίηση πλαισίων για την εφαρμογή της διαδικασίας, σύμφωνα με τις προτεινόμενες προδιαγραφές, εμφανίζει μία σειρά από πλεονεκτήματα, όπως:

- Είναι δυνατή η χρήση όλων των συμβατών με τα πρότυπα (SysML) περιβαλλόντων μοντελοποίησης συστημάτων, αφού δεν απαιτείται η χρήση συγκεκριμένου και προσαρμοσμένου περιβάλλοντος μοντελοποίησης.
- Μπορούν να αξιοποιηθούν προϋπάρχοντα μοντέλα συστημάτων, αφού προκρίνεται η συμμόρφωση με τα πρότυπα, έναντι των ειδικών τύπων μοντέλων συστημάτων.
- Η ύπαρξη του μετα-μοντέλου αναφοράς για τη μεθοδολογία προσομοίωσης προάγει τη δια-λειτουργικότητα μεταξύ διαφορετικών προσομοιωτών της ίδιας μεθοδολογίας προσομοίωσης.
- Τα απαιτούμενα επιμέρους τμήματα του πλαισίου ορίζονται/υλοποιούνται άπαξ και αξιοποιούνται πολλαπλώς, κατά την εφαρμογή της μεθοδολογίας σε διαφορετικά μοντέλα.
- Επίσης, το κόστος ανάπτυξης των τμημάτων του πλαισίου ανταποδίδεται και από τη δυνατότητα αξιοποίησής τους και εκτός της μεθοδολογίας, καθώς εντάσσονται στη λογική της προτυποποίησης και της δια-λειτουργικότητας διαδικασιών σχεδιασμού και περιβαλλόντων προσομοίωσης.

4.2 Προτεινόμενη Διαδικασία Αυτοματοποιημένης Επικύρωσης Μοντέλων Συστημάτων

Η προτεινόμενη διαδικασία ενδυναμώνει ουσιαστικά το σχεδιασμό συστημάτων, δίνοντας τη δυνατότητα για αποτίμηση της αναμενόμενης συμπεριφοράς των μοντέλων συστημάτων, με χρήση αυτοματοποιημένης παραγωγής και εκτέλεσης κώδικα προσομοίωσης. Η μοντελοποίηση συστημάτων προσεγγίζεται ως μία κυκλική, επαναληπτική διαδικασία, η οποία ανατροφοδοτείται με έγκυρα στοιχεία εκτίμησης της συμπεριφοράς του συστήματος, διευκολύνοντας τον εντοπισμό ενδεχόμενων δυσλειτουργιών του συστήματος κατά το σχεδιασμό. Έτσι, υιοθετείται το γνωστό σχήμα της ανατροφοδότησης της κύριας διαδικασίας με στοιχεία εκτίμησης της ποιότητάς της, οδηγώντας σε συνολική βελτίωση του παραγόμενου αποτελέσματος.

Το σχήμα αυτό έχει υιοθετηθεί με επιτυχία σε πολλές και ποικίλες περιπτώσεις και μορφές, όπως:

- στην παρακολούθηση και ακριβή πρόβλεψη της συμπεριφοράς κρίσιμων συστημάτων πραγματικού χρόνου (real-time), μέσω της παράλληλης με τη λειτουργία του συστήματος εκτέλεσης ενός διαρκώς προσαρμοζόμενου μοντέλου προσομοίωσης [53]. Σε αυτή την περίπτωση το μοντέλο προσομοίωσης αναπροσαρμόζεται, όποτε παρατηρείται απόκλιση των δεδομένων λειτουργίας του συστήματος από τα αποτελέσματα της προσομοίωσης, ενώ η επαναληπτική διαδικασία ελέγχου δεν τερματίζεται όσο εξακολουθεί να λειτουργεί το σύστημα.

- στην απλοποίηση και επιτάχυνση του σχεδιασμού, της παραμετροποίησης, της διαχείρισης, της βελτιστοποίησης και της αποκατάστασης δικτύων κινητών συσκευών, όπως περιγράφεται στο [54].

- Τα φίλτρα Kalman [55], όπου διατηρείται μία εκτίμηση της κατάστασης του συστήματος και ένας δείκτης αβεβαιότητας για την εκτίμηση. Η εκτίμηση κατάστασης ενημερώνεται με τη χρήση ενός μοντέλου μεταβάσεων καταστάσεων, καθώς και μετρήσεων.

Ειδικότερα, στην προτεινόμενη διαδικασία, η αυτοματοποίηση των βημάτων που μεσολαβούν μεταξύ μοντελοποίησης του συστήματος και αποτύπωσης της εκτιμώμενης συμπεριφοράς, επιτρέπει στο σχεδιαστή συστημάτων να επικεντρώνει απερίσπαστος στο σχεδιασμό, αλλά και να έχει στη διάθεσή του εκτίμηση της συμπεριφοράς του υπό διαμόρφωση μοντέλου συστήματος.

Στο σχήμα 4.1 δίνεται μία γραφική αποτύπωση της προτεινόμενης διαδικασίας.

Όπως φαίνεται και στο σχήμα 4.1, ο σχεδιαστής ορίζει το μοντέλο του συστήματος χρησιμοποιώντας ένα εργαλείο μοντελοποίησης και μετά από μία σειρά αυτοματοποιημένων βημάτων, έχει στη διάθεσή του, εντός του μοντέλου συστήματος, μία έγκυρη αποτίμηση της συμπεριφοράς του. Αυτό επιτυγχάνεται με την εκτέλεση των πέντε ακόλουθων βημάτων:

1. Ορισμός του μοντέλου συστήματος (περιλαμβανομένων και των στοιχείων που απαιτούνται για την προσομοίωση) από το σχεδιαστή σε περιβάλλον μοντελοποίησης συστημάτων.

Η τυπική ορθότητα και η πληρότητα του μοντέλου ελέγχεται από τους προβλεπόμενους μηχανισμούς στο περιβάλλον μοντελοποίησης.

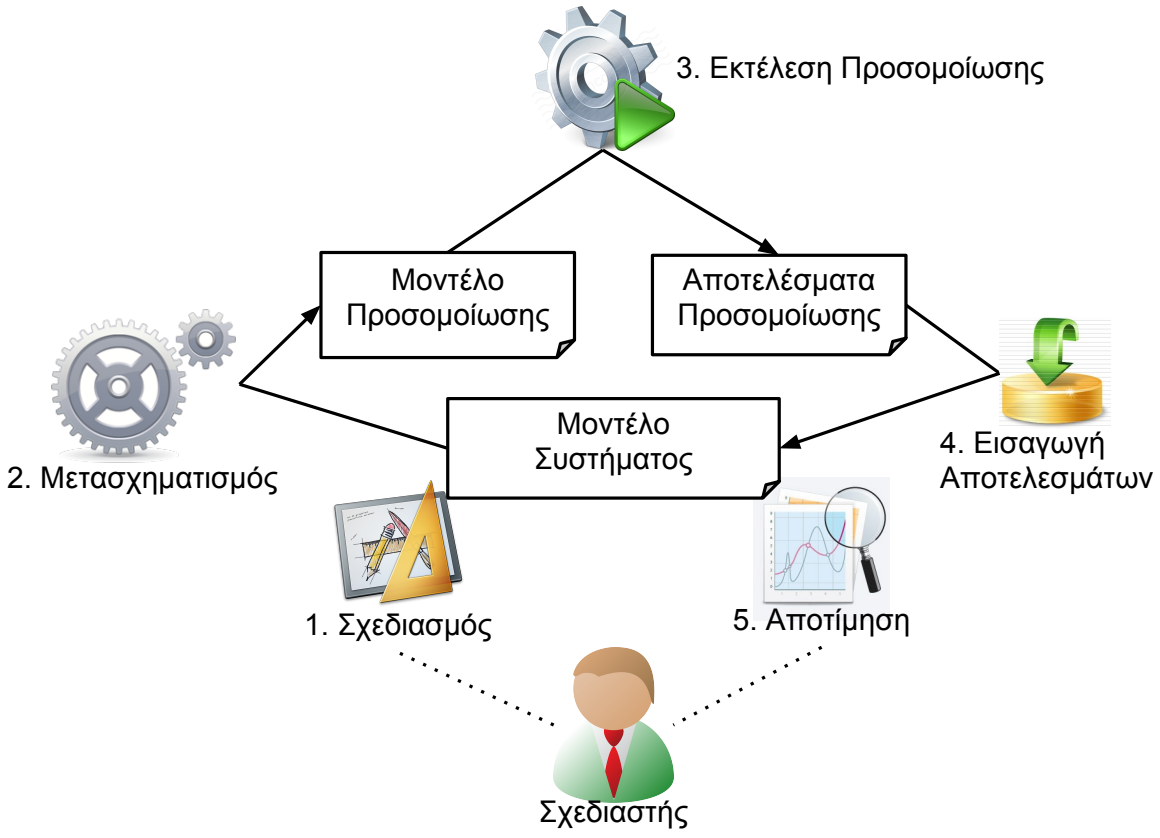
2. Το μοντέλο συστήματος μετασχηματίζεται σε εκτελέσιμο μοντέλο προσομοίωσης.

3. Η προσομοίωση εκτελείται και παράγονται αποτελέσματα, δηλαδή καταγράφονται μετρήσεις από την προσομοιούμενη λειτουργία του συστήματος.

4. Τα αποτελέσματα προσομοίωσης εισάγονται σε θέσεις που έχουν προβλεφθεί, στο μοντέλο συστήματος.

5. Ο σχεδιαστής αποτιμά τις εκτιμήσεις της συμπεριφοράς του μοντέλου συστήματος. Σε περίπτωση που η αποτίμηση αποβεί αρνητική, ο σχεδιαστής αποφασίζει σχετικά με απαιτούμενες σχεδιαστικές τροποποιήσεις στο μοντέλο του συστήματος. Ξεκινάει δηλαδή ένα νέο κύκλο της προτεινόμενης διαδικασίας.

Η προτεινόμενη προσέγγιση έχει σαν κύριο στόχο τον περιορισμό της απαιτούμενης ανάμειξης του σχεδιαστή στα σημεία όπου αυτό είναι απαραίτητο. Έτσι, ο σχεδιαστής εμπλέκεται αποκλειστικά στο πρώτο και το τελευταίο στάδιο της διαδικασίας. Στο πρώτο στάδιο λαμβάνει χώρα ο πλήρης ορισμός του μοντέλου, συμπεριλαμβανομένων και των χαρακτηριστικών που αφορούν την προσομοίωση, ενώ στο τελευταίο, η εκτίμηση της συμπεριφοράς είναι διαθέσιμα στο σχεδιαστή για αξιολόγηση. Επίσης, τόσο ο σχεδιασμός, όσο και η αποτίμηση γίνονται στο



Σχήμα 4.1: Μεθοδολογία προσομοίωσης μοντέλων συστημάτων

ίδιο περιβάλλον μοντελοποίησης συστημάτων. Τα χαρακτηριστικά αυτά προσδίδουν πλεονεκτήματα στην προσέγγιση, όπως τα ακόλουθα:

- Ο σχεδιαστής διευκολύνεται ουσιαστικά, αφού δεν απαιτείται να επανέρχεται και να συμπληρώνει το μοντέλο προσομοίωσης σε διαφορετικές χρονικές στιγμές και μέσω διαφορετικών περιβαλλόντων μοντελοποίησης και προγραμματισμού, που προσεγγίζουν το μοντέλο από διαφορετική οπτική γωνία.

- Η διαδικασία απαλλάσσεται από εισαγωγή λαθών εκ παραδρομής σε ενδιάμεσα στάδια μέχρι την εκτέλεση της προσομοίωσης, καθώς τα ενδιάμεσα στάδια είναι αυτοματοποιημένα.

- Οι αλλαγές στο μοντέλο του συστήματος, μπορούν να ελέγχονται ως προς την απόδοσή τους χωρίς την απαίτηση επίπονων, επιμέρους παρεμβάσεων από το σχεδιαστή.

- Η αποτίμηση της συμπεριφοράς του μοντέλου συστήματος γίνεται στο ίδιο περιβάλλον μοντελοποίησης συστημάτων, διευκολύνοντας την επιλογή και εφαρμογή των απαιτούμενων τροποποιήσεων στο μοντέλο του συστήματος. Επίσης, η αξιοποίηση των αποτελεσμάτων προσομοίωσης μπορεί να είναι πιο ουσιαστική και πολύπλευρη, καθώς μπορεί να συνδυαστεί με άλλες πληροφορίες του μοντέλου.

Για την εξασφάλιση της δυνατότητας παροχής αυτοματοποιημένης προσομοίωσης του μοντέλου συστήματος απαιτούνται:

- Ο πλήρης και ορθός ορισμός του μοντέλου συστήματος και των χαρακτηριστικών απόδοσης.

- Η ύπαρξη μηχανισμού μετασχηματισμού του μοντέλου συστήματος σε μοντέλο προσομοίωσης, για συγκεκριμένη μεθοδολογία προσομοίωσης.

- Η ύπαρξη προσομοιωτή, ο οποίος να μπορεί να εκτελέσει προσομοίωση, βάσει του ανωτέρω μοντέλου προσομοίωσης.

Η δυνατότητα ενσωμάτωσης των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος απαιτεί κατ' ελάχιστο τις ακόλουθες προϋποθέσεις:

- Την πρόβλεψη συγκεκριμένων πεδίων, που θα είναι διαθέσιμα για την συμπλήρωση με τα αποτελέσματα της εκτέλεσης της προσομοίωσης. Τα πεδία αυτά θα δημιουργηθούν ως επέκταση των στοιχείων του μοντέλου συστήματος.

- Την αποθήκευση των αποτελεσμάτων προσομοίωσης με χρήση αναγνωριστικών που θα διατηρούν τον σύνδεσμο με τα αντίστοιχα στοιχεία του μοντέλου συστήματος.

- Την ύπαρξη μηχανισμού ενσωμάτωσης των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος.

Ο δεύτερος βασικός άξονας της προσέγγισης είναι η γενικότητα ως προς τη δυνατότητα ένταξης και αξιοποίησης διαφορετικών μεθοδολογιών, περιβαλλόντων και εργαλείων για την προσομοίωση. Καθοριστικό ρόλο προς αυτή την κατεύθυνση διαδραματίζει η απαίτηση για τη δηλωτική αποτύπωση του μοντέλου προσομοίωσης, αντί για την απευθείας παραγωγή εκτελέσιμου κώδικα προσομοίωσης.

Για την αξιοποίηση μίας συγκεκριμένης μεθοδολογίας προσομοίωσης, στα πλαίσια της προτεινόμενης γενικής προσέγγισης, απαιτούνται:

- Να είναι διαθέσιμος κάποιος δηλωτικός τρόπος αναπαράστασης μοντέλων προσομοίωσης για την εν λόγω μεθοδολογία προσομοίωσης.
- Να υπάρχει τουλάχιστον ένα περιβάλλον εκτέλεσης, μοντέλων προσομοίωσης αποτυπωμένων σύμφωνα με τον ανωτέρω τρόπο.

4.3 Προτεινόμενο Πλαίσιο

Η προτεινόμενη προσέγγιση στοχεύει στην αναβάθμιση της διαδικασίας σχεδιασμού, μέσω αυτοματοποιημένης προσομοίωσης μοντέλων SysML, ανεξάρτητα από τη γλώσσα ή το περιβάλλον προσομοίωσης. Ωστόσο, η εφαρμογή και χρήση της γενικής διαδικασίας με χρήση μίας συγκεκριμένης μεθοδολογίας προσομοίωσης απαιτεί τη δημιουργία και αξιοποίηση ενός πλαισίου, το οποίο να την υποστηρίζει. Σε αυτή την ενότητα προδιαγράφονται τα χαρακτηριστικά ενός τέτοιου πλαισίου σε γενικό επίπεδο. Το πλαίσιο περιλαμβάνει ένα σύνολο από συστατικά στοιχεία, τα οποία διαμορφώνουν μία υποδομή που επιτρέπει το σχεδιασμό συστημάτων με τους όρους της προτεινόμενης διαδικασίας.

4.3.1 Επισκόπηση Πλαισίου

Στη λογική της διασφάλισης της γενικότητας της προσέγγισης και της δια-λειτουργικότητας σε διαφορετικά εργαλεία μοντελοποίησης, μετασχηματισμού και προσομοίωσης, θεμελιώδη ρόλο στην προδιαγραφή του πλαισίου παίζουν:

- Η υποδομή του MOF, το οποίο επιτρέπει τον ορισμό γλωσσών (μετα-μοντέλων), για τη μοντελοποίηση οποιουδήποτε πεδίου εφαρμογής.
- Η UML, η οποία είναι μία γλώσσα (ορισμένη ως ένα μετα-μοντέλο MOF) για ανάλυση και σχεδιασμό συστημάτων λογισμικού και διαθέτει μηχανισμό επέκτασης: τον ορισμό προφίλ.
- Η SysML, η οποία είναι μία γλώσσα μοντελοποίησης συστημάτων, ορισμένη ως ένα προφίλ της UML.
- Η QVT, η οποία είναι μία γλώσσα μετασχηματισμού μοντέλων από ένα μετα-μοντέλο αφετηρίας σε ένα μετα-μοντέλο προορισμού.

Τα συστατικά του προτεινόμενου πλαισίου, όπως φαίνονται και στο σχήμα 4.2, είναι:

1. Ένα προφίλ που να επεκτείνει και να θέτει συγκεκριμένους περιορισμούς στη SysML κατά τέτοιο τρόπο, ώστε να εξασφαλίζεται ότι περιλαμβάνεται όλη η απαιτούμενη πληροφορία για την προσομοίωση του μοντέλου συστήματος, σύμφωνα με μία μεθοδολογία προσομοίωσης. Επίσης, το προφίλ θα πρέπει να προδιαγράφει τα πεδία στα οποία θα τοποθετηθούν τα αποτελέσματα της προσομοίωσης.

2. Το μετα-μοντέλο αναφοράς της χρησιμοποιούμενης μεθοδολογίας προσομοίωσης, ορισμένο σύμφωνα με το MOF.

3. Ένας μετασχηματισμός QVT, ο οποίος να δέχεται ως είσοδο τα μοντέλα συστήματος και να παράγει μοντέλα προσομοίωσης.

4. Ένα περιβάλλον προσομοίωσης, το οποίο να μπορεί να εκτελέσει μοντέλα προσομοίωσης, συμβατά με το μετα-μοντέλο αναφοράς.

5. Το μετα-μοντέλο για την αποθήκευση των αποτελεσμάτων προσομοίωσης, ορισμένο σύμφωνα με το MOF.

6. Ένας μετασχηματισμός (QVT) ή κάποιος άλλος μηχανισμός, ο οποίος θα ενσωματώνει τα αποτελέσματα προσομοίωσης στο μοντέλο συστήματος.

Στη συνέχεια αναλύεται ο ρόλος, οι απαιτήσεις και ο προτεινόμενος τρόπος προσέγγισης για κάθε ένα από τα συστατικά του προτεινόμενου πλαισίου.

4.3.2 Επέκταση Μοντέλου Συστήματος: Προφίλ Προσομοίωσης

Τα μοντέλα συστημάτων (SysML) δεν μπορούν να προσομοιώνονται εν γένει. Αφ' ενός, ενώ περιέχουν προδιαγραφές που αφορούν τη συμπεριφορά των στοιχείων του μοντέλου, αυτές δεν αντιστοιχούν σαφώς σε κώδικα κάποιας μεθοδολογίας προσομοίωσης. Αφ' εταίρου, στο μοντέλο συστήματος δεν περιλαμβάνονται προδιαγραφές που αφορούν την εκτέλεση της προσομοίωσης, όπως π.χ. χρόνο προσομοίωσης ή αριθμό επαναληπτικών εκτελέσεων της προσομοίωσης.

Επομένως, είναι απαραίτητη η επέκταση των μοντέλων συστημάτων, κατά τέτοιο τρόπο ώστε:

- να διατίθενται οι απαιτούμενες δομές αποτύπωσης των χαρακτηριστικών που είναι απαραίτητα για την προσομοίωση (δομή και συμπεριφορά στοιχείων του μοντέλου, καθώς και παραμέτρους εκτέλεσης προσομοίωσης).

- να ελέγχεται η ορθή και ολοκληρωμένη συμπλήρωση των δομών αυτών.

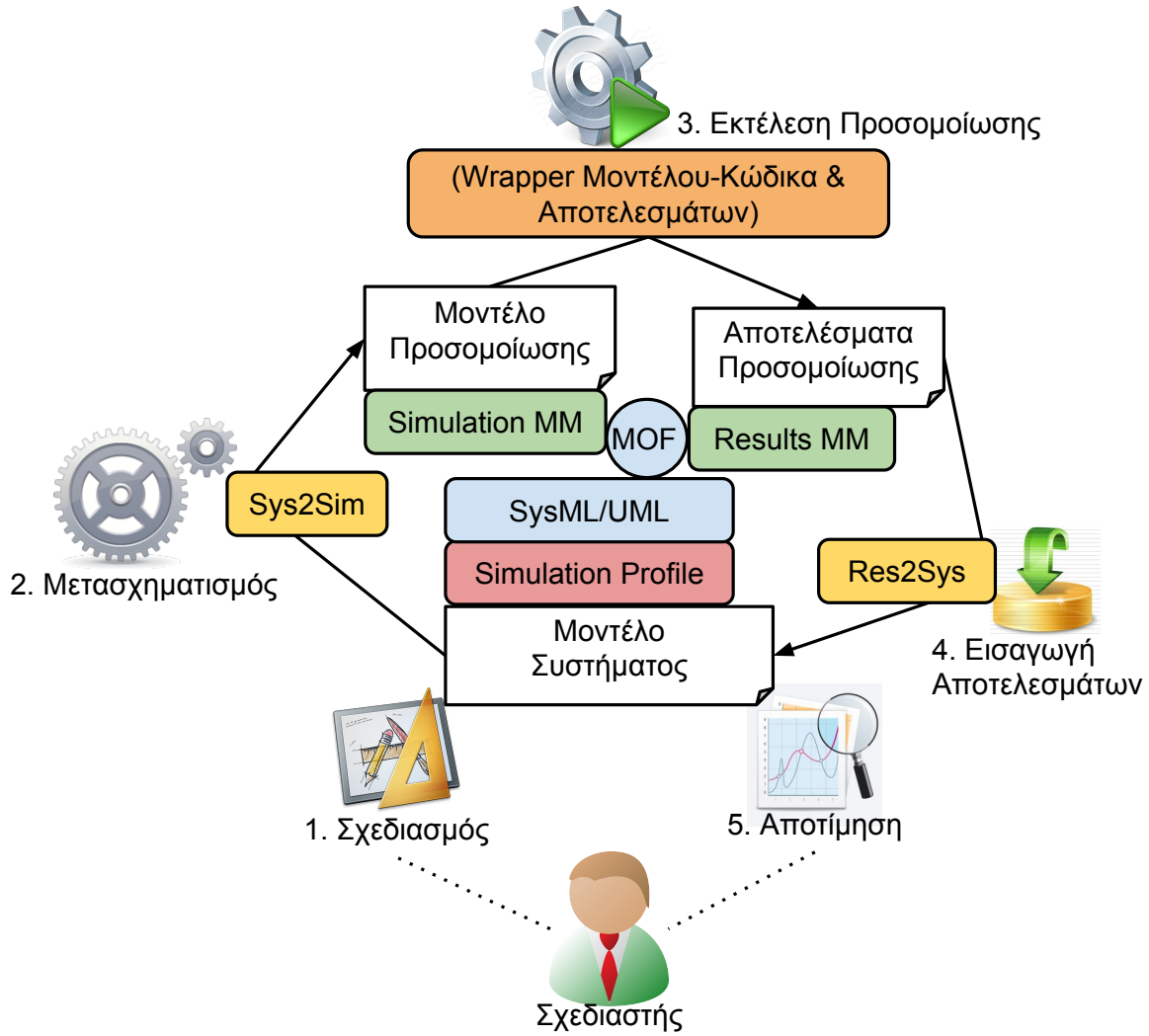
Ο ενδεδειγμένος τρόπος για τη διασφάλιση των δύο αυτών απαιτήσεων, διατηρώντας τη συμβατότητα με τα περιβάλλοντα μοντελοποίησης συστημάτων, είναι η διαμόρφωση μίας εξειδίκευσης των μοντέλων συστημάτων (SysML), μέσω του μηχανισμού *profiling*. Αυτός είναι ένας μηχανισμός της UML, ο οποίος κινείται κυρίως προς τις δύο ακόλουθες κατευθύνσεις:

- την επέκταση της UML με ορισμό στερεοτύπων (stereotypes), που χαρακτηρίζουν υπάρχουσες δομές της UML και μπορούν να ορίζουν επιπρόσθετα πεδία (tagged values).
- τον περιορισμό της UML με ορισμό ενός συνόλου περιορισμών σχετικά με τη δομή και τις τιμές των στοιχείων του μοντέλου.

Τέτοια προφίλ -για συγκεκριμένη μεθοδολογία προσομοίωσης- θα μπορούν να αξιοποιούνται απροβλημάτιστα στα υπάρχοντα εργαλεία μοντελοποίησης συστημάτων. Άλλωστε και η ίδια η SysML είναι ορισμένη ως ένα προφίλ της UML και ως τέτοιο υποστηρίζεται στα εργαλεία μοντελοποίησης. Το προφίλ συγκεκριμένης μεθοδολογίας προσομοίωσης θα πρέπει προφανώς να επεκτείνει δομικά στοιχεία της SysML κυρίως, στοχεύοντας στην προσομοίωση μοντέλων συστημάτων.

Πλεονεκτήματα:

- Ο ορισμός και η χρήση κατάλληλων προφίλ επιτρέπει την προσθήκη των σχετικών με προσομοίωση στοιχείων στο ίδιο μοντέλο συστήματος, αντί για τον επανορισμό ενός νέου μοντέλου προσομοίωσης.



Σχήμα 4.2: Πλαίσιο για τη μεθοδολογία προσομοίωσης μοντέλων συστημάτων: Μετα-μοντέλα, προφίλ και πρότυπα.

- Η υιοθέτηση μίας πρότυπης μεθόδου όπως το profiling, προσδίδει αξιοπιστία στην προσέγγιση και προωθεί την εκμάθηση και αξιοποίηση κοινών εργαλείων μοντελοποίησης, χωρίς την ανάγκη εξειδικευμένων γνώσεων για συγκεκριμένα περιβάλλοντα προσομοίωσης.
- Υποστηρίζεται η δυνατότητα επικύρωσης του εμπλουτισμένου μοντέλου συστήματος, μέσω περιορισμών που μπορούν να οριστούν στο προφίλ.
- Η εστίαση παραμένει στην ανάλυση και το σχεδιασμό του συστήματος, χωρίς να αποσπά την προσοχή του σχεδιαστή σε εργαλεία προσομοίωσης.

Θέματα προς Εξέταση:

- Ο ορισμός ενός προφίλ SysML για κάποια μεθοδολογία προσομοίωσης δεν είναι επουσιώδης διαδικασία. Τα απαιτούμενα χαρακτηριστικά για προσομοίωση πρέπει να αναγνωριστούν και να τοποθετηθούν στα κατάλληλα στοιχεία του μοντέλου συστήματος.
- Οι δομικές αντιστοιχίες μεταξύ SysML και μοντέλων προσομοίωσης δεν επαρκούν για τον ορισμό του προφίλ. Η σημασιολογία της μεθοδολογίας προσομοίωσης, ιδίως σε ότι έχει να κάνει με την περιγραφή της συμπεριφοράς των στοιχείων, πρέπει να λαμβάνεται υπόψη.
- Τέτοιου είδους προφίλ SysML δεν είναι διαθέσιμα για την πλειοψηφία των μεθοδολογιών προσομοίωσης. Επομένως, για να είναι δυνατή η χρήση τους, απαιτείται ο ορισμός τους.
- Η ορθότητα και η πληρότητα ενός προφίλ SysML αυτού του είδους δεν είναι εύκολο να αποδειχθεί εξ αρχής. Ενδεχομένως απαιτείται η σύνθεση όλου του πλαισίου και η διεξαγωγή δοκιμών, ώστε να διαπιστωθεί η ορθότητα του προφίλ.
- Ο ορισμός του τμήματος της πληροφορίας προσομοίωσης που έχει να κάνει με τη συμπεριφορά των στοιχείων του μοντέλου με ένα γενικό και συγχρόνως ακριβή τρόπο είναι εξαιρετικά δύσκολη διαδικασία.

4.3.3 Μετα-Μοντέλο Αναφοράς Προσομοίωσης

Για την επίτευξη της προσομοίωσης του μοντέλου συστήματος, απαιτείται ο σαφής προσδιορισμός του τρόπου περιγραφής του μοντέλου προσομοίωσης. Για να εξυπηρετηθεί καλύτερα η διαδικασία του μετασχηματισμού του μοντέλου συστήματος ενδείκνυται ο προσδιορισμός να γίνεται με τρόπο δηλωτικό. Έτσι, η ”εύκολη” λύση της επιλογής μιας γλώσσας προγραμματισμού που χρησιμοποιείται για τη μεθοδολογία προσομοίωσης δεν προκρίνεται, αφού αυτές είναι γλώσσες χαμηλού επιπέδου και όχι δηλωτικές. Άλλωστε, σε περιπτώσεις μεθοδολογιών προσομοίωσης που εξυπηρετούνται από περισσότερες της μίας γλώσσες προγραμματισμού, θα απαιτούνταν πολλοί και διαφορετικοί μετασχηματισμοί για κάθε μία από αυτές.

Επομένως, ο στόχος δεν πρέπει να είναι, κατ’ αρχήν, η παραγωγή ενός προγράμματος προσομοίωσης. Αντίθετα, η διαδικασία προσανατολίζεται στρατηγικά στον ορισμό και την αξιοποίηση μίας δηλωτικής, προδιαγραφής μοντέλων προσομοίωσης. Ασφαλώς, η προδιαγραφή αυτή

θα πρέπει να περιέχει το σύνολο της απαιτούμενης πληροφορίας για την εκτέλεση προσομοίωσης και είτε να εκτελείται σε συμβατούς με αυτήν προσομοιωτές, είτε να μετατρέπεται σε δεύτερο χρόνο σε εκτελέσιμο πρόγραμμα προσομοίωσης.

Σε αυτό το σημείο είναι σημαντικό να αποσαφηνιστεί η διαφοροποίηση του εμπλουτισμένου μοντέλου συστήματος, σύμφωνα με το προφίλ της μεθοδολογίας προσομοίωσης, από το μοντέλο προσομοίωσης. Διατυπώθηκε τόσο για το πρώτο όσο και για το δεύτερο ότι περιέχουν το σύνολο της απαιτούμενης πληροφορίας για την εκτέλεση προσομοίωσης. Ωστόσο, υπάρχει μία κεφαλαιώδης και ειδοποιός διαφορά. Το εμπλουτισμένο μοντέλο συστήματος περιέχει την πληροφορία αυτή κατακερματισμένη σε ένα σύνολο στοιχείων και πεδίων του συστήματος, που απλά έχουν επεκταθεί (με το προφίλ), χωρίς να έχει προσαρμοστεί και αλλοιωθεί η δομή και η σημασιολογία του μοντέλου συστήματος από τη μεθοδολογία προσομοίωσης. Αντίθετα, η προδιαγραφή μοντέλων προσομοίωσης ορίζεται αποκλειστικά με γνώμονα τα χαρακτηριστικά της μεθοδολογίας προσομοίωσης, αφού σκοπός της είναι η παροχή ενός μέσου για τη συμπαγή και ακριβή αποτύπωση εκτελέσιμων μοντέλων προσομοίωσης. Επομένως, ακόμα και στην περίπτωση που τα μοντέλα προσομοίωσης δεν εκτελούνται άμεσα, θα πρέπει να μετατρέπονται με απλό τρόπο σε εκτελέσιμο κώδικα προσομοίωσης.

Παράλληλα, για να είναι δυνατή η αξιοποίηση ισχυρών -εννοιολογικά και λειτουργικά- δομών και μηχανισμών μετασχηματισμού κατά την παραγωγή του μοντέλου προσομοίωσης, είναι ιδιαίτερα χρήσιμη η ύπαρξη ενός κοινού υποβάθρου. Αυτό δεν θα μπορούσε να είναι άλλο από το MOF, το οποίο:

- είναι μία πρότυπη γλώσσα ορισμού μετα-μοντέλων (προδιαγραφών μοντέλων).
- αποτελεί την υποδομή με την οποία έχει οριστεί η UML και, επομένως, η SysML.
- πλαισιώνεται από πρότυπες γλώσσες και εργαλεία μετασχηματισμού, όπως η QVT.

Επομένως, το προτεινόμενο πλαίσιο απαιτεί την ύπαρξη ενός μετα-μοντέλου αναφοράς MOF, το οποίο θα προδιαγράφει τον τρόπο ορισμού των μοντέλων προσομοίωσης, σύμφωνα με μία συγκεκριμένη μεθοδολογία προσομοίωσης. Το μετα-μοντέλο αναφοράς θα πρέπει να εξασφαλίζει ότι τα μοντέλα προσομοίωσης θα περιλαμβάνουν πέρα από την πλήρη και αναλυτική περιγραφή της δομής και της συμπεριφοράς του μοντέλου προσομοίωσης και τις απαιτούμενες πληροφορίες για την εκτέλεση της προσομοίωσης, όπως π.χ. συνολικός χρόνος προσομοίωσης ή άλλη συνθήκη τερματισμού. Σε περίπτωση που το μετα-μοντέλο αυτό δεν είναι ήδη διαθέσιμο, θα πρέπει να οριστεί με τέτοιο τρόπο ώστε τα μοντέλα προσομοίωσης να είναι συμπαγή, περιεκτικά και έτοιμα για εκτέλεση από έναν ή περισσότερους προσομοιωτές.

Πλεονεκτήματα:

- Η ένταξη στο πλαίσιο ενός μετα-μοντέλου προσομοίωσης MOF επιτρέπει την ανάπτυξη και αξιοποίηση ισχυρών σημασιολογικά μετασχηματισμών QVT.
- Η σημαντική παρουσία του μετα-μοντέλου προσομοίωσης MOF προάγει και αξιοποιεί τη δια-λειτουργικότητα διαφορετικών προσομοιωτών της μεθοδολογίας προσομοίωσης.

Θέματα προς Εξέταση:

- Τα απαιτούμενα μετα-μοντέλα δεν είναι διαθέσιμα για την πλειονότητα των μεθοδολογιών προσομοίωσης, οπότε απαιτείται ο ορισμός τους, για την εφαρμογή του πλαισίου.

- Συνήθως δεν είναι διαθέσιμοι προσομοιωτές συμβατοί με τα μετα-μοντέλα αναφοράς, οπότε απαιτείται η δημιουργία εργαλείων μετατροπής μοντέλων προσομοίωσης σε κώδικα προσομοίωσης.

- Τα μετα-μοντέλα προσομοίωσης αναφοράς θα αξιοποιούνται πλήρως όταν θα τυγχάνουν ευρείας αποδοχής, χρήσης, αλλά και προτυποποίησης. Κάτι τέτοιο δεν γίνεται συνήθως με συνοπτικές διαδικασίες.

4.3.4 Μετασχηματισμός Μοντέλου Συστήματος σε Μοντέλο Προσομοίωσης

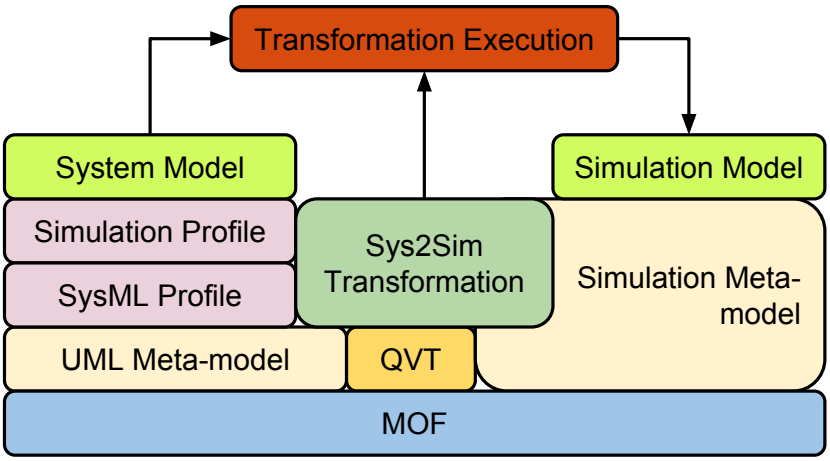
Για τη διασφάλιση της γενικότητας και της ευρύτητας εφαρμογής της διαδικασίας, ο μετασχηματισμός των μοντέλων συστημάτων σε μοντέλα προσομοίωσης πρέπει να βασίζεται στις έννοιες της MDA, αξιοποιώντας γλώσσες, όπως η QVT [22] και σχετικά εργαλεία, που την υποστηρίζουν.

Κατ' αρχήν, η χρήση μίας γλώσσας η οποία έχει σχεδιαστεί για τον ορισμό μετασχηματισμών είναι προφανώς καταλληλότερη, από άλλες προσεγγίσεις, όπως η συγγραφή προγραμμάτων μετασχηματισμού σε γλώσσα γενικού σκοπού. Γλώσσες ορισμού μετασχηματισμών, όπως η QVT, προσφέρουν υψηλού επιπέδου δομές αντιστοίχησης, που οδηγούν στη δημιουργία απλών και συντηρήσιμων μετασχηματισμών. Ειδικότερα, στην περίπτωση που το υπόβαθρο προσδιορισμού των μοντέλων τόσο αφετηρίας όσο και προορισμού είναι κοινό (MOF), μία γλώσσα που βασίζεται στο ίδιο υπόβαθρο (QVT) είναι η πλέον κατάλληλη. Επίσης, η προτυποποίηση των γλωσσών αυτών δεν περιορίζει την εκτέλεση των μετασχηματισμών σε συγκεκριμένα, κλειστά, εμπορικά εργαλεία. Αντίθετα, υπάρχει η δυνατότητα χρήσης οποιουδήποτε εργαλείου υποστηρίζει το πρότυπο της QVT.

Η αξιοποίηση μετασχηματισμών QVT προϋποθέτει την ύπαρξη, αποδοχή και χρήση ενός μετα-μοντέλου αναφοράς MOF για τη χρησιμοποιούμενη μεθοδολογία προσομοίωσης, όπως περιγράφεται στο 4.3.3. Προφανώς, για να είναι λειτουργικό το πλαίσιο στο σύνολό του θα πρέπει να είναι διαθέσιμα εργαλεία και προσομοιωτές, που να μπορούν να εκτελέσουν μοντέλα προσομοίωσης, ορισμένα σύμφωνα με το μετα-μοντέλο αναφοράς.

Οι μετασχηματισμοί QVT ορίζονται με βάση τα μετα-μοντέλα (MOF) αφετηρίας και προορισμού, όπως φαίνεται και στο σχήμα 4.3. Στην περίπτωση του προτεινόμενου πλαισίου, το μετα-μοντέλο αφετηρίας είναι αυτό της UML, ενώ το μετα-μοντέλο προορισμού είναι το μετα-μοντέλο αναφοράς για τη μεθοδολογία προσομοίωσης. Ο μετασχηματισμός μπορεί επίσης να αξιοποιεί πρόσθετα χαρακτηριστικά στοιχεία του μοντέλου αφετηρίας, όπως αυτά που ορίζονται στα προφίλ της SysML και του προφίλ προσομοίωσης. Κατά την εκτέλεση του μετασχηματισμού λαμβάνονται όλες οι πληροφορίες που περιέχονται στο μοντέλο συστήματος και

διαμορφώνεται αντίστοιχα το μοντέλο προσομοίωσης, το οποίο είναι εκ κατασκευής συμβατό με το μετα-μοντέλο αναφοράς της μεθοδολογίας προσομοίωσης.



Σχήμα 4.3: Διαστρωμάτωση υποδομών, μετα-μοντέλων και προφίλ για μετασχηματισμό μοντέλων.

Πιο αναλυτικά, στο σχήμα 4.3 η QVT παρουσιάζεται ένα επίπεδο πάνω από το MOF, γιατί βασίζεται σε αυτό. Αυτό συμβαίνει κυρίως γιατί η QVT επιτρέπει τον ορισμό μετασχηματισμών από ένα μετα-μοντέλο (MOF) προς ένα άλλο μετα-μοντέλο (MOF). Μάλιστα, δεδομένων των μετα-μοντέλων αφετηρίας και προορισμού, η QVT προβλέπει συντακτικό έλεγχο του οριζόμενου μετασχηματισμού, ώστε οι δομές του μοντέλου αφετηρίας να αναγνωρίζονται με τρόπο σύμφωνο με το μετα-μοντέλο του και οι δομές του μοντέλου προορισμού συντίθενται, σύμφωνα με το μετα-μοντέλο του. Κατ' αυτό τον τρόπο, αποφεύγονται τυχόν άσκοπες και άστοχες συντακτικά αναζητήσεις (εξ ορισμού αποτυχημένες), ενώ εξασφαλίζεται και η συντακτική συμμόρφωση του παραγόμενου μοντέλου με το μετα-μοντέλο του (μετα-μοντέλο προσομοίωσης στην περίπτωσή μας).

Ο μετασχηματισμός (Sys2Sim) τοποθετείται πάνω από την QVT, αλλά και τα δύο μετα-μοντέλα, αφού είναι διαθέσιμη η αξιοποίηση των δομών τους, όπως περιγράφηκε παραπάνω. Στα αριστερά του μετασχηματισμού τοποθετούνται τα προφίλ της SysML και της μεθοδολογίας προσομοίωσης. Η αξιοποίηση στοιχείων των προφίλ κάνει το μετασχηματισμό πιο σαφή, απλό και αποδοτικό, αφού επιλέγονται συγκεκριμένα στοιχεία του μοντέλου, π.χ. αυτά που είναι χαρακτηρισμένα με κάποιο στερεότυπο. Ωστόσο, τα πρόσθετα αυτά στοιχεία αξιοποιούνται με τρόπο διερευνητικό και όχι καθοριστικό, όπως συμβαίνει με τα μετα-μοντέλα.

Τα μοντέλα τοποθετούνται πάνω από τα μετα-μοντέλα τους και τυχόν προφίλ (μοντέλο συστήματος), που αξιοποιούνται για τον ορισμό τους. Τέλος, η εκτέλεση του μετασχηματισμού δεν είναι τίποτα άλλο από την αξιοποίηση του μοντέλου συστήματος και του QVT μετασχηματισμού για την παραγωγή του αντίστοιχου μοντέλου προσομοίωσης.

Πλεονεκτήματα:

- Ο ορισμός των μετασχηματισμών με QVT είναι απλούστερος σε σχέση με την υλοποίησή τους ως ειδικού σκοπού προγραμμάτων.

- Ο ορισμός του μετασχηματισμού γίνεται με βάση τα μετα-μοντέλα συστήματος και προσομοίωσης, γεγονός που διασφαλίζει τη συντακτική ορθότητα του παραγόμενου μοντέλου προσομοίωσης.

Θέματα προς Εξέταση:

- Ο έλεγχος της πληρότητας ενός τέτοιου μετασχηματισμού δεν είναι πάντα εύκολος και απαιτεί τουλάχιστον ένα εκτεταμένο σύνολο δοκιμών.

Προσέγγιση Ορθότητας Μετασχηματισμών - Εναλλακτικές

Στο προτεινόμενο πλαίσιο, ο μετασχηματισμός μοντέλων συστημάτων προς μοντέλα προσομοίωσης προδιαγράφεται με standards της MDA και υλοποιείται με συμβατά εργαλεία, όπως περιγράφηκε παραπάνω. Η ίδια η επιλογή της QVT για την προδιαγραφή του μετασχηματισμού εξασφαλίζει εξ αρχής ορισμένες από τις συνθήκες ορθότητας του μετασχηματισμού ή διευκολύνει με πολύ ουσιαστικό τρόπο τον έλεγχο και την επαλήθευση της ορθότητάς του.

Συγκεκριμένα, η **συντακτική ορθότητα των παραγόμενων μοντέλων προορισμού** είναι ούτως ή άλλως διασφαλισμένη. Η υποδομή που παρέχει το MOF είναι θεμελιώδες χαρακτηριστικό της QVT, ενώ το μετα-μοντέλο προορισμού προσδιορίζεται κατά τον ορισμό του μετασχηματισμού. Επομένως, τα συμβατά με QVT εργαλεία μετασχηματισμού έχουν τη δυνατότητα να ελέγχουν αν ένας μετασχηματισμός οδηγεί σε επιτρεπτά, από συντακτική άποψη παραγόμενα μοντέλα. Έτσι, διασφαλίζεται η συμμόρφωση με το μετα-μοντέλο προορισμού.

Επίσης, οι σχεδιαστικές επιλογές της προσέγγισης, με έμφαση στην προτυποποίηση (MDA / MOF / QVT), διαμορφώνει ένα **πλαίσιο που ευνοεί την εφαρμογή τεχνικών ελέγχου**. Το γεγονός ότι τόσο η γλώσσα ορισμού του μετασχηματισμού (QVT), όσο και η υποδομή της προδιαγραφής του μετα-μοντέλου προορισμού (MOF) είναι πρότυπα, επιτρέπει τον ορισμό αυτοματοποιημένων μηχανισμών ελέγχου. Τέτοιοι έλεγχοι μπορούν να υλοποιηθούν με εργαλεία, διευκολύνοντας τη διεξαγωγή εξαντλητικών ελέγχων του μετασχηματισμού και θα μπορούσαν να αφορούν την πληρότητα του παραγόμενου μοντέλου ή το ντετερμινισμό του μετασχηματισμού. Παράλληλα, η χρήση προτύπων (MOF/QVT) καθιστά την **προσέγγιση ανοιχτή και εύληπτη στην ευρύτερη κοινότητα, διευρύνοντας τις δυνατότητες ελέγχου**.

Επιπρόσθετα, η QVT και ιδιαίτερα η QVT-R, με το δηλωτικό τρόπο αποτύπωσης, μπορεί να θεωρηθεί ως κατ' ουσίαν μία **γλώσσα προσδιορισμού προδιαγραφών αντιστοιχήσεων μετα-μοντέλων και όχι ως απλά μία γλώσσα ορισμού μετασχηματισμών**. Με την QVT-R, τα δύο μοντέλα αντιπαρατίθενται σε σχέσεις, οι οποίες αποτυπώνουν δομικές αντιστοιχίες, καθώς και τις τιμές που πρέπει να αποδίδονται στα στοιχεία του μοντέλου προορισμού. Οι σχέσεις αυτές όμως δεν εφαρμόζονται σε όλα τα στοιχεία των μοντέλων αφετηρίας και προορισμού, παρά μόνο όταν συντρέχουν οι κατάλληλες συνθήκες (*when* clause). Επίσης, η επιτυχής εφαρμογή μίας σχέσης σε ένα συνδυασμό στοιχείων των μοντέλων αφετηρίας και προορισμού συνεπάγεται την απόπειρα εφαρμογής επόμενων σχέσεων σε κατάλληλα στοιχεία των δύο μοντέλων (*where* clause). Με αυτή την έννοια, ένας QVT-R μετασχηματισμός δεν περιγράφει

μόνο πώς γίνεται ο μετασχηματισμός, αλλά συγχρόνως και ποιες συνθήκες θα έπρεπε να συντρέχουν για να γίνει ο συγκεκριμένος μετασχηματισμός. Επομένως, **ένας μετασχηματισμός QVT-R διατηρεί την επιθυμητή σημασιολογική συνέπεια στο βαθμό που ο δημιουργός του εκφράζει με ορθότητα, ακρίβεια και πληρότητα τις προϋποθέσεις και τις συνέπειες των αντιστοιχήσεων**.

Συγκεκριμένες ιδιότητες του μετασχηματισμού θα μπορούσαν να εξεταστούν και με αναλυτικό τρόπο. Μία τέτοια προσέγγιση θα μπορούσε να θεωρηθεί συμπληρωματική στις προαναφερθείσες τεχνικές, κυρίως όταν δίνεται έμφαση στη θεωρητική θεμελίωση, έναντι της πρακτικής/τεχνικής αντιμετώπισης. Αρχικά, απαιτείται η μαθηματική αποτύπωση συγκεκριμένων ιδιοτήτων που εξασφαλίζονται από το μετα-μοντέλο UML και τα προφίλ SysML και προσομοίωσης. Στη συνέχεια, απαιτείται η μαθηματική αποτύπωση του μετα-μοντέλου προορισμού και του μετασχηματισμού στα πλαίσια της QVT. Έτσι, μπορούμε να οδηγηθούμε σε μαθηματικές εκφράσεις των παραγόμενων μοντέλων και να μελετήσουμε τη διατήρηση ή μη των ιδιοτήτων του μοντέλου, αξιολογώντας κατ' αυτό τον τρόπο το μετασχηματισμό. Και σε αυτή την περίπτωση, η εκτενής αξιοποίηση προτύπων (MOF/UML/SysML/QVT) μπορεί να οδηγήσει σε προτυποποίηση βασικών αναλυτικών θεωρήσεων, οι οποίες θα εξειδικεύονται για κάθε μεθοδολογία προσομοίωσης και μετασχηματισμό.

Οι Cabot κ.α. [56] προτείνουν μία μέθοδο επικύρωσης και επαλήθευσης δηλωτικών μετασχηματισμών μοντέλων, μέσω σταθερών (invariants), εκπεφρασμένων σε όρους OCL. Η μέθοδος αφορά μετασχηματισμούς QVT-R και Γραμματικές Τριπλών Γράφων (Triple Graph Grammars) (TGGs). Στην περίπτωση της QVT-R, ορίζεται μία σταθερά για τις top relations και μία για τις απλές relations. Με βάση τις σταθερές αυτές, υποστηρίζεται:

- **αυτοματοποιημένη επαλήθευση του μετασχηματισμού**: με τον ορισμό ενός συνόλου κατηγορημάτων (predicates) μπορούν να εκφραστούν συγκεκριμένα ποιοτικά χαρακτηριστικά των σχέσεων, που γενικεύονται και στο μετασχηματισμό (εφαρμοσιμότητα, ασθενής/κανονική/ισχυρή εκτελεσιμότητα, καθολικότητα, ντετερμινισμός, λειτουργικότητα (functionality), εξαντλητικότητα, αμφιμονοτιμία, αμφιμονοσήμανση, υπαγωγικότητα) και τον έλεγχο των αντίστοιχων OCL εκφράσεων από εργαλεία.

- **υποβοηθούμενη επικύρωση του μετασχηματισμού**: καθώς η απάντηση στο ερώτημα, αν ο μετασχηματισμός είναι ο σωστός/αναμενόμενος, προϋποθέτει τη συμμετοχή του σχεδιαστή, η αυτοματοποίηση είναι αδύνατη. Ωστόσο, μπορεί να υποστηριχθεί από εργαλεία (α) η πρόταση σεναρίων ελέγχου και (β) η εύρεση αντιπαραδειγμάτων που να ακυρώνουν κανόνες που θέτει ο σχεδιαστής (με σταθερές OCL).

Δεδομένων των προαναφερθέντων χαρακτηριστικών, που διευκολύνουν την εξ αρχής συγκροτημένη και ορθή από διάφορες απόψεις αποτύπωση του μετασχηματισμού, η σύγκριση με εναλλακτικούς τρόπους προσδιορισμού των μετασχηματισμών, αναδεικνύει μία σειρά πλεονεκτημάτων. Μία εναλλακτική προσέγγιση είναι η δημιουργία του μετασχηματισμού με συγγραφή ad-hoc προγράμματος με γλώσσα γενικού σκοπού. Σε αυτή την περίπτωση δεν υπάρχει εγγενής υποστήριξη του κοινού υποβάθρου MOF και συντακτική αξιοποίηση των μετα-

μοντέλων αφετηρίας και προορισμού. Επίσης, ο ορισμός του μετασχηματισμού δεν μπορεί να γίνει με δηλωτικό τρόπο, όπως στην περίπτωση της QVT-R, ενώ δεν αναμένεται να είναι διαθέσιμες υψηλού επιπέδου δομές, όπως συσχετίσεις και εντολές λογικής εξαγωγής συμπερασμάτων (inference). Επομένως, οι μετασχηματισμοί ορίζονται σε ένα αρκετά χαμηλότερο επίπεδο και με λιγότερο ισχυρά μέσα, οδηγώντας μοιραία σε πιο σύνθετους, πολύπλοκους και εκτενείς μετασχηματισμούς. Η πιθανότητα εισαγωγής λαθών είναι αυξημένη και ο εντοπισμός τους πιο δύσκολος, ενώ υστερεί και η συντηρησιμότητά τους. Τέλος, η δυνατότητα αξιοποίησης και επαναχρησιμοποίησης τους είναι πιο περιορισμένη, αφού δεν έχουν οριστεί στα πλαίσια ενός ευρέως αποδεκτού προτύπου.

Ορισμένες από τις αρνητικές πτυχές αυτής της επιλογής θα μπορούσαν να περιοριστούν σε κάποιο βαθμό. Για το θέμα της μη εγγενούς υποστήριξης του MOF, θα μπορούσαν να αναπτυχθούν βιβλιοθήκες, οι οποίες να παρέχουν βασικές λειτουργίες πρόσβασης, αναζήτησης και επεξεργασίας σε δομές μοντέλων MOF. Έτσι, ένα υψηλότερο επίπεδο χειρισμού των μοντέλων θα ήταν διαθέσιμο. Τις ιδιότητες αυτές έχει η QVT-O.

Για τη δηλωτική αποτύπωση του μετασχηματισμού, θα μπορούσε να εξεταστεί η χρήση άλλων γλωσσών μετασχηματισμού, που βασίζονται σε πρότυπα (patterns), όπως π.χ. η XSLT. Αυτή όμως αναγνωρίζει απλές συντακτικές δομές XML. Αντίθετα, η XML αναπαράσταση μοντέλων βασισμένων σε MOF (Ανταλλαγή Μεταδεδομένων XML (XML Metadata Interchange) (XMI)) στοχεύει στην αναπαράσταση πολλαπλών και σύνθετων συσχετίσεων μεταξύ των στοιχείων των μοντέλων. Οι συσχετίσεις αυτές αποτυπώνονται κυρίως με αλληλοαναφορές σε μοναδικά αναγνωριστικά (IDs) των στοιχείων και όχι με δομές XML, οι οποίες θα γίνονταν εξαιρετικά πολύπλοκες και με πολλές επαναλήψεις. Επομένως, η χρησιμότητα της XSLT για μετασχηματισμούς σύνθετων μοντέλων XMI είναι περιορισμένη.

Θα μπορούσαν να εξεταστούν και άλλες προσεγγίσεις, με σταθερότερο θεωρητικό υπόβαθρο, για την αντιστοίχηση και το μετασχηματισμό μοντέλων, όπως οι TGGs [57,58]. Σε αυτή την περίπτωση όμως θα έπρεπε να αντιμετωπιστούν ζητήματα όπως η πολυπλοκότητα των γράφων σε σύνθετα μετα-μοντέλα, η έλλειψη εγγενούς υποστήριξης του MOF και η μη επαρκής και ομοιογενής υποστήριξη από εργαλεία [59].

Η ATL είναι η πρόταση της Eclipse για τις προκλήσεις που θέτει η QVT. Υποστηρίζεται από το Πλαίσιο Μοντελοποίησης Eclipse (Eclipse Modeling Framework) (EMF) και είναι μία γλώσσα μετασχηματισμών για μοντέλα που είναι βασισμένα σε MOF ή Βασικό Μετα-μοντέλο του Πλαισίου Μοντελοποίησης Eclipse (Eclipse Modeling Framework Core Meta-Model) (Ecore). Η ATL τυγχάνει της υποστήριξης της ευρέως διαδεδομένης πλατφόρμας ανάπτυξης Eclipse και αποτελεί μία επιλογή για υλοποίηση μετασχηματισμών μοντέλων. Ωστόσο, δεν είναι πρότυπο, ενώ εκφράζει μία πιο πρακτική λογική και κινείται σε ένα λιγότερο αφηρημένο και δηλωτικό επίπεδο, σε σχέση με την QVT-R [60].

Συμπερασματικά, η QVT αποτελεί μία ασφαλή επιλογή για την υλοποίηση του μετασχηματισμού των μοντέλων συστημάτων σε μοντέλα προσομοίωσης. Τα πλεονεκτήματα της επιλογής αυτής είναι -συνοπτικά- ότι πρόκειται για πρότυπο, υποστηρίζει εγγενώς το MOF, παρέχει τη δυνατότητα δηλωτικού ορισμού του μετασχηματισμού (QVT-R), αλλά διαθέτει και διαδικα-

στικά χαρακτηριστικά (QVT-O) και επιτρέπει την ανάπτυξη τεχνικών και εργαλείων ελέγχου των μετασχηματισμών.

4.3.5 Εκτέλεση Προσομοίωσης - Παραγωγή Αποτελεσμάτων

Η παραγωγή του μοντέλου προσομοίωσης ως προϊόν του μετασχηματισμού από το μοντέλο συστήματος εξασφαλίζει τη δυνατότητα εκτέλεσης της προσομοίωσης. Όπως περιγράφεται και στο 4.3.3, το μοντέλο προσομοίωσης περιλαμβάνει όλα εκείνα τα στοιχεία που καθιστούν εφικτή την εκτέλεση της προσομοίωσης. Αυτό εξασφαλίζεται από το συνδυασμό των περιορισμών του προφίλ, του μετα-μοντέλου προσομοίωσης και του μετασχηματισμού.

Σύμφωνα με τη λογική της προσέγγισης, η ύπαρξη περιβάλλοντος προσομοίωσης, συμβατού με το μετα-μοντέλο προσομοίωσης, θεωρείται δεδομένη. Ωστόσο, σε μία πρακτική διάσταση, στην περίπτωση κατά την οποία κανένας από τους προσομοιωτές της συγκεκριμένης μεθοδολογίας προσομοίωσης δεν δέχεται ως είσοδο μοντέλα αυτής της μορφής, είναι απαραίτητη η ύπαρξη ενός wrapper, ο οποίος θα αναλάβει τη μετάφραση των μοντέλων προσομοίωσης σε κατάλληλο κώδικα προσομοίωσης. Επίσης, πρόνοια θα πρέπει να ληφθεί για την παραγωγή των αποτελεσμάτων προσομοίωσης σε μορφή εκμεταλλεύσιμη από τα υπόλοιπα στοιχεία του πλαισίου, δηλαδή σύμφωνα με το μετα-μοντέλο αποτελεσμάτων.

Δεδομένων (α) της σημασιολογικής συνάφειας ή και ταύτισης του μετα-μοντέλου προσομοίωσης με την αντίστοιχη γλώσσα προσομοίωσης (πρόκειται για την ίδια μεθοδολογία προσομοίωσης) και (β) του γεγονότος ότι η μετάβαση γίνεται από μία πιο αφηρημένη και σημασιολογικά πυκνή προς μία πιο απλή και χαμηλού επιπέδου αναπαράσταση, η υλοποίηση του wrapper θα πρέπει να θεωρείται μία μάλλον τετριμμένη διαδικασία. Αν κάτι τέτοιο δεν ισχύει, τότε το θέμα που αναδεικνύεται σε αυτό το σημείο έχει τα αίτιά του είτε στο μη ιδιαίτερα πετυχημένο ορισμό του μετα-μοντέλου προσομοίωσης, είτε στην υλοποίηση των προσομοιωτών με πολλές και ακραίες ιδιαιτερότητες σε σχέση με τη μεθοδολογία προσομοίωσης. Τα ζητήματα αυτά θα πρέπει να επιλυθούν ανεξάρτητα και πριν από αυτό το στάδιο.

Καθώς το μοντέλο προσομοίωσης (α) αποθηκεύεται σε XMI (μία μορφή XML) και (β) σε αντίθεση με το αρχικό μοντέλο συστήματος, είναι συμπαγές σημασιολογικά και δομικά, στα πλαίσια της μεθοδολογίας προσομοίωσης, μπορούν να αξιοποιηθούν απλές τεχνικές μετασχηματισμού, βασισμένες σε αναγνώριση XML προτύπων (patterns), όπως η XSLT.

Για την απόδοση των αποτελεσμάτων προσομοίωσης σε μορφή, σύμφωνη με το απλό μετα-μοντέλο αποτελεσμάτων, θα πρέπει να αξιοποιηθεί το μοναδικό αναγνωριστικό κάθε στοιχείου του μοντέλου συστήματος, το όνομα του μετρήσιμου στοιχείου και η παραγόμενη τιμή. Έτσι, συντίθενται τα αποτελέσματα σε μορφή αξιοποιήσιμη από άλλα στοιχεία του πλαισίου.

4.3.6 Ενσωμάτωση Αποτελεσμάτων - Αποτίμηση

Για την εισαγωγή των αποτελεσμάτων προσομοίωσης, που έχουν αποτυπωθεί σύμφωνα με το μετα-μοντέλο αποτελεσμάτων απαιτείται ένας απλός μετασχηματισμός. Ο μετασχηματισμός θα πρέπει να λαμβάνει το μοντέλο αποτελεσμάτων ως πηγή και το αρχικό μοντέλο

συστήματος ως προορισμό. Τα κατάλληλα στοιχεία του μοντέλου συστήματος θα βρίσκονται, με το μοναδικό αναγνωριστικό στοιχείο που συνοδεύει τα αποτελέσματα, ενώ οι τιμές των αποτελεσμάτων θα τίθενται στις αντίστοιχες θέσεις, που έχουν προβλεφθεί ως tagged values στο προφίλ προσομοίωσης. Κάτι τέτοιο υποστηρίζεται από την QVT, αφού το μοντέλο στο οποίο επέρχονται μεταβολές (enforce domain) δύναται να έχει περιεχόμενο στην αρχή του μετασχηματισμού (τροποποίηση και όχι δημιουργία εξ αρχής). Σε αυτή την περίπτωση η χρήση της QVT είναι σχεδόν τετριμμένη, αφού δεν τίθεται θέμα αντιστοίχησης εννοιών, αλλά εντοπισμό συγκεκριμένων στοιχείων του μοντέλου σύμφωνα με το αναγνωριστικό τους και, στη συνέχεια, απόδοση τιμών σε συγκεκριμένα και προκαθορισμένα πεδία.

Με την ολοκλήρωση της εισαγωγής των αποτελεσμάτων στο μοντέλο συστήματος, καθίσταται εφικτή η αποτίμηση της συμπεριφοράς του. Η αποτίμηση μπορεί να γίνει με μία σειρά από διαφορετικές προσεγγίσεις ως προς τη στόχευση και τα μέσα. Μία λίστα από τέτοιες προσεγγίσεις, που μπορούν να αξιοποιηθούν εναλλακτικά ή συμπληρωματικά, αναφέρονται ακολούθως:

- **Αυτοματοποιημένος έλεγχος ικανοποίησης μη λειτουργικών προδιαγραφών συγκεκριμένων στοιχείων του μοντέλου.** Για την υλοποίηση αυτού του ελέγχου πρέπει να πληρούνται τρεις προϋποθέσεις. Αρχικά, απαιτείται πρόβλεψη θέσεων για τον ορισμό ορίων για τα ελεγχόμενα στοιχεία. Επίσης, ο σχεδιαστής θα πρέπει να έχει θέσει συγκεκριμένες τιμές σε αυτά τα όρια. Τέλος, θα πρέπει να έχουν οριστεί μηχανισμοί ελέγχου της εκτιμώμενης συμπεριφοράς των στοιχείων του μοντέλου συστήματος σε σχέση με τα όρια. Ενδεδειγμένη λύση τόσο για την πρόβλεψη θέσεων για τα όρια, όσο και για το μηχανισμό ελέγχου εκτίμησης-ορίων είναι η συμπερίληψη τους σε κατάλληλο προφίλ αποτίμησης, με μορφή tagged values για τα πρώτα και περιορισμούς OCL για το δεύτερο. Ο έλεγχος περιορισμών OCL υποστηρίζεται άμεσα από τα περιβάλλοντα μοντελοποίησης SysML/UML.

- **Αυτοματοποιημένος έλεγχος ικανοποίησης μη λειτουργικών προδιαγραφών σύνθετων δομών του μοντέλου.** Πρόκειται για μία πιο σύνθετη περίπτωση της προηγούμενης, όπου οι έλεγχοι δεν αφορούν ατομικά στοιχεία του μοντέλου, αλλά σύνθετες δομές. Η μελέτη της συμπεριφοράς των δομών αυτών και ο έλεγχος ικανοποίησης των προδιαγραφών απαιτεί μία ευρύτερη και συνδυαστική αντιμετώπιση της λειτουργίας του συστήματος. Και αυτή η λειτουργικότητα θα μπορούσε να υλοποιηθεί σε κάποιο βαθμό με εγγενή μέσα (OCL) ή με ανάπτυξη κατάλληλων plugins στα εργαλεία μοντελοποίησης, για πιο ελεύθερη αντιμετώπιση.

- **Αξιοποίηση εξωτερικών μεθοδολογιών και εργαλείων.** Η πληροφορία που περιέχεται στο μοντέλο του συστήματος με τα στοιχεία της ποσοτικής εκτίμησης της συμπεριφοράς μπορεί να εξαχθεί και να μετασχηματιστεί, ώστε να χρησιμοποιηθεί σε εξειδικευμένα εξωτερικά εργαλεία για περαιτέρω μελέτη.

- **Διαισθητική αποτίμηση.** Η ενσωμάτωση των αποτελεσμάτων προσομοίωσης στο μοντέλο

συστήματος μπορεί να υποβοηθήσει τη διαισθητική αποτίμηση του σχεδιαστή, μέσω της επισκόπησης του μοντέλου.

Εφόσον, η αποτίμηση ικανοποιεί αυτοματοποιημένους και ατομικούς ελέγχους, δεν απαιτούνται άλλες ενέργειες στη φάση του σχεδιασμού. Αντίθετα, αν οι έλεγχοι αναδείξουν ανεπάρκεια στην ικανοποίηση συγκεκριμένων προδιαγραφών, απαιτούνται τροποποιήσεις στο μοντέλο του συστήματος και μία νέα διαδικασία ελέγχου του τροποποιημένου μοντέλου. Η ενσωμάτωση των αποτελεσμάτων στο μοντέλο συστήματος μπορεί να βοηθήσει την αξιοποίηση τεχνικών εντοπισμού των αδυναμιών του σχεδιασμού. Σε αυτά τα πλαίσια, οι απαιτούμενες βελτιώσεις στο σχεδιασμό του συστήματος με σκοπό τη βελτιστοποίησή του μοντέλου συστήματος θα μπορούσαν να προκύψουν με αυτόματο τρόπο σε κάποιο βαθμό.

Κεφαλαιο 5

Αυτοματοποιημενη Προσομοιωση Μοντελων SysML με DEVS

Η προτεινόμενη διαδικασία αυτοματοποιημένης επικύρωσης μοντέλων συστημάτων (SysML) εφαρμόστηκε επιτυχώς με αξιοποίηση της μεθοδολογίας προσομοίωσης DEVS. Όπως προβλέπεται, υλοποιήθηκε το προτεινόμενο πλαίσιο, που υποστηρίζει τη διαδικασία, σύμφωνα με τις προδιαγραφές της ενότητας 4.3.

Σε αυτό το κεφάλαιο, αρχικά αναλύονται θέματα σχετικά με την επιλογή της συγκεκριμένης μεθοδολογίας προσομοίωσης. Στη συνέχεια, παρουσιάζονται τα στοιχεία του πλαισίου που υλοποιήθηκε και συγκεκριμένα:

- το DEVS προφίλ για SysML
- το μετα-μοντέλο MOF για DEVS
- ο μετασχηματισμός μοντέλων συστημάτων σε μοντέλα DEVS
- ο μετασχηματισμός μοντέλων DEVS σε εκτελέσιμο κώδικα DEVS
- το μετα-μοντέλο αποτελεσμάτων προσομοίωσης

- ο μηχανισμός παραγωγής των αποτελεσμάτων προσομοίωσης στην μορφή που καθορίζει το μετα-μοντέλο αποτελεσμάτων

- η διαδικασία εισαγωγής των αποτελεσμάτων στο μοντέλο συστήματος

5.1 Επιλογή DEVS

Μία από τις πρώτες επιλογές που πρέπει να γίνουν για την έναρξη της υλοποίησης του πλαισίου είναι αυτή της μεθοδολογίας προσομοίωσης, που θα αξιοποιηθεί. Εν προκειμένω, η επιλογή του DEVS έγινε για μία σειρά από λόγους. Αρχικά, πρόκειται για μία προσέγγιση για την προσομοίωση συστημάτων διακριτού χρόνου με στέρεο θεωρητικό υπόβαθρο. Αυτή η θεωρητική θεμελίωση προσδίδει αξιοπιστία στην προσομοίωση και ακρίβεια στα αποτελέσματά της.

Επίσης, όπως αναφέρεται και στο 3.2.2, υπήρχαν ήδη αρκετές προσπάθειες αποτύπωσης μοντέλων DEVS σε μορφές XML, αντί για μαθηματικές εκφράσεις συνόλων. Παρά το γεγονός οι προσπάθειες αυτές δεν είχαν καταλήξει σε ένα πρότυπο μετα-μοντέλο DEVS, είχαν διαμορφώσει ένα πλαίσιο διερεύνησης των δυνατοτήτων για την αποτύπωση μοντέλων DEVS με τρόπο δηλωτικό, σε μορφή XML.

Παράλληλα με τους δηλωτικούς τρόπους αναπαράστασης, αναπτύσσονταν και τεχνικές για τη δυνατότητα μετατροπής των αναπαραστάσεων αυτών σε εκτελέσιμο κώδικα DEVS (σε Java ή C++). Οι τεχνικές αυτές δημιούργησαν τις προϋποθέσεις για τη δυνατότητα εκτέλεσης δηλωτικών περιγραφών μοντέλων DEVS, που είναι ιδιαίτερα χρήσιμη στην εφαρμογή του πλαισίου.

Επιπρόσθετα, η δομή των μοντέλων DEVS παρουσιάζει αρκετές ομοιότητες με τη δομή των μοντέλων SysML. Το γεγονός αυτό απλοποιεί τον ορισμό, αλλά και τη χρήση του SysML προφίλ για DEVS, ενώ διευκολύνει και την υλοποίηση του μετασχηματισμού μοντέλων συστημάτων σε μοντέλα DEVS.

Τέλος, γνώση, εξοικείωση και εμπειρία με την μεθοδολογία προσομοίωσης DEVS, ήταν διαθέσιμη και μπορούσε να αξιοποιηθεί στα πλαίσια της υλοποίησης του πλαισίου.

5.1.1 Επισκόπηση DEVS

Σύμφωνα με το φορμαλισμό DEVS [61], μαθηματικά σύνολα χρησιμοποιούνται για την περιγραφή της δομής και της συμπεριφοράς των μοντέλων. Τα μοντέλα προσδιορίζονται με ένα τμηματικό και ιεραρχικό τρόπο, ορίζοντας δύο είδη μοντέλων:

- τα *atomic models*, που δίνουν έμφαση σε θέματα συμπεριφοράς

- τα *coupled models*, που δίνουν έμφαση σε θέματα δομής, περιγράφοντας πώς διασυνδέονται τα atomic και άλλα coupled μοντέλα, με έναν ιεραρχικό τρόπο, για το σχηματισμό πιο σύνθετων δομών.

Ένα atomic DEVS μοντέλο περιγράφεται από μία πλειάδα επτά στοιχείων της μορφής:

$$M =< X, Y, S, ta, \delta_{ext}, \delta_{int}, \lambda >$$

όπου:

- X είναι το σύνολο γεγονότων εισόδου
- Y είναι το σύνολο γεγονότων εξόδου
- S είναι το σύνολο καταστάσεων του μοντέλου
- $ta : S \to \mathbb{T}^{\infty}$ είναι η συνάρτηση προώθησης χρόνου (time advance), η οποία χρησιμοποιείται για τον προσδιορισμό της χρονικής διάρκειας παραμονής σε κάθε κατάσταση
- $\delta_{ext} : Q \times X \to S$ είναι η συνάρτηση εξωτερικών μεταβάσεων, η οποία προσδιορίζει την επόμενη κατάσταση του συστήματος, όταν ληφθεί κάποιο εξωτερικό συμβάν, όπου $Q = \{(s, t_e) | s \in S, t_e \in (\mathbb{T} \cap [0, ta(s)])$, δηλαδή η επόμενη κατάσταση εξαρτάται από την τρέχουσα κατάσταση, το χρόνο παραμονής σε αυτή και το εξωτερικό συμβάν.
- $\delta_{int} : S \to S$ είναι η συνάρτηση εσωτερικών μεταβάσεων, η οποία προσδιορίζει την επόμενη, κάθε φορά, κατάσταση, στην οποία θα μεταβεί το σύστημα, όταν έχει πληρωθεί ο χρόνος παραμονής στην τρέχουσα κατάσταση
- $\lambda : S \to Y^{\varnothing}$ είναι η συνάρτηση εξόδου, όπου $Y^{\varnothing} = Y \cup \{\varnothing\}$ και $\varnothing \notin Y$ είναι το μη αντιληπτό γεγονός. Η συνάρτηση αυτή προσδιορίζει το παραγόμενο γεγονός εξόδου, όταν πληρωθεί ο χρόνος παραμονής του συστήματος σε μία κατάσταση.

Ένα coupled DEVS μοντέλο περιγράφεται με μία πλειάδα οκτώ στοιχείων της μορφής:

$$N =< X, Y, D, \{M_i\}, C_{xx}, C_{yx}, C_{yy}, Select >$$

όπου:

- X είναι το σύνολο γεγονότων εισόδου
- Y είναι το σύνολο γεγονότων εξόδου
- D είναι το σύνολο των ονομάτων των περιεχόμενων στοιχείων
- $\{M_i\}$ είναι το σύνολο των περιεχόμενων στοιχείων, όπου $\forall i \in D, M_i$ είναι ένα atomic ή ένα coupled DEVS μοντέλο.
- $C_{xx} \subseteq X \bigcup_{i \in D} X_i$ είναι το σύνολο των εξωτερικών συζεύξεων εισόδων
- $C_{yx} \bigcup_{i \in D} Y_i \times \bigcup_{i \in D} X_i$ είναι το σύνολο των εσωτερικών συζεύξεων
- $C_{yy} : \bigcup_{i \in D} Y_i \to Y^{\varnothing}$ είναι η συνάρτηση εξωτερικών συζεύξεων εξόδων
- $Select : 2^D \to D$ είναι η συνάρτηση επίλυσης προτεραιότητας σε περίπτωση ταυτόχρονων συμβάντων

Συνοψίζοντας, από την μία πλευρά, το DEVS επιτρέπει την ιεραρχική δόμηση μοντέλων μέσω των coupled μοντέλων, τα οποία μπορούν να περιέχουν άλλα μοντέλα DEVS (atomic ή coupled), διασυνδέοντας κατάλληλα τις εισόδους και εξόδους των μοντέλων (του coupled και των περιεχόμενων μοντέλων). Από την άλλη πλευρά, η συμπεριφορά των μοντέλων ορίζεται τελικά σε επίπεδο atomic μοντέλων. Τα τελευταία, ελλείψει εξωτερικών γεγονότων, παραμένουν σε μία από τις προδιαγεγραμμένες καταστάσεις για συγκεκριμένο χρονικό διάστημα και παράγουν κάποιο γεγονός εξόδου με την παρέλευση του διαστήματος, πριν μεταβούν στην επόμενη προβλεπόμενη κατάσταση. Σε περίπτωση εξωτερικού γεγονότος, ανάλογα με το είδος αυτού, την τρέχουσα κατάσταση και το χρόνο παραμονής του μοντέλου σε αυτή, το μοντέλο μεταβαίνει στην επόμενη προβλεπόμενη κατάσταση.

Από μία πρακτική οπτική γωνία, τα περισσότερα περιβάλλοντα προσομοίωσης DEVS, αλλά και οι σχετικές προσπάθειες δηλωτικής αποτύπωσης εκτελέσιμων μοντέλων DEVS αντιμετωπίζουν τα μοντέλα DEVS ως δύο ειδών συστατικά (atomic και coupled). Και στις δύο περιπτώσεις τα στοιχεία των συνόλων X και Y αντιμετωπίζονται ως input και output ports, διακρίνοντας έτσι τις διαφορετικές κατηγορίες εισόδων και εξόδων. Στα atomic μοντέλα το σύνολο των καταστάσεων αντιμετωπίζεται με μεταβλητές κατάστασης, ενώ οι συναρτήσεις $(ta, \delta_{ext}, \delta_{int}, \lambda)$ με αντίστοιχα functions. Στα coupled μοντέλα δίνεται η δυνατότητα δήλωσης περιεχόμενων συστατικών και των διασυνδέσεων των ports.

5.1.2 Ομοιότητες Μεταξύ SysML και DEVS

Στο DEVS, τα coupled μοντέλα επιτρέπουν τη δημιουργία πολύπλοκων μοντέλων, μέσω της ιεραρχικής σύνθεσης απλούστερων μοντέλων και κατάλληλης σύνδεσης των ports εισόδου και εξόδου. Αντίστοιχα, στη SysML, η έννοια της ιεραρχικής σύνθεσης στοιχείων παραπέμπει άμεσα στα blocks, τα οποία μπορούν να περιέχουν ή να αναφέρονται σε άλλα blocks. Στα BDDs περιγράφονται οι ιεραρχίες των blocks, ενώ σε ένα IBD αναλύονται οι διασυνδέσεις των ports ενός σύνθετου block και των περιεχόμενων σε αυτό blocks.

Ομοιότητες μπορούν να βρεθούν και στα στοιχεία των atomic μοντέλων DEVS με στοιχεία των SysML blocks. Στα atomic DEVS μοντέλα η κατάσταση περιγράφεται με ένα σύνολο state variables και τις μεταξύ τους συσχετίσεις. Αντίστοιχα, τα SysML blocks διαθέτουν value properties, τα οποία μπορούν να συσχετίζονται μέσω constraint blocks σε PDs.

Πέρα όμως από την πληροφορία της κατάστασης και τη δομή, η εκτέλεση της προσομοίωσης εξαρτάται και από την περιγραφή της συμπεριφοράς του μοντέλου. Στο DEVS αυτό γίνεται με τις συναρτήσεις των atomic μοντέλων. Η συνάρτηση εσωτερικών μεταβάσεων ορίζει τις προϋποθέσεις μετάβασης από κάθε κατάσταση στις υποψήφιες διαδόχους της. Αντίστοιχη λειτουργία στη SysML επιτελούν τα SMDs, όπου τα guard conditions εκφράζουν τις προϋποθέσεις για την ενεργοποίηση της μετάβασης. Πρόσθετα χαρακτηριστικά των SMDs, όπως τα duration guard conditions και οι επιπτώσεις (effects) των μεταβάσεων, καθιστούν αυτό τον τύπο διαγράμματος κατάλληλο για την περιγραφή των συναρτήσεων προώθησης χρόνου και εξόδου των atomic μοντέλων. Κάτι τέτοιο είναι αναμενόμενο, αφού οι τελευταίες είναι πολύ

Πίνακας 5.1: Αντιστοίχηση στοιχείων DEVS και SysML

Στοιχείο DEVS	**Στοιχείο SysML**
Μοντέλο (atomic, coupled)	Block
Port εισόδου	Flow port με κατεύθυνση *in*
Port εξόδου	Flow port με κατεύθυνση *out*
Atomic μοντέλο	BDDs
Μεταβλητή κατάστασης	Value property
Σχέσεις μεταξύ μεταβλητών κατάστασης	Constraints σε PDs
$ta, \delta_{int}, \lambda$	SMDs
δ_{ext}	ADs
Coupled μοντέλο	BDDs, IBDs
Περιεχόμενα μοντέλα	Block parts
Εξωτερικές συζεύξεις εισόδων	Συνδέσεις από *in* flow ports του περιέχοντος block προς *in* flow ports των περιεχόμενων blocks
Εσωτερικές συζεύξεις	Συνδέσεις από *out* flow ports περιεχόμενων blocks προς *in* flow ports άλλων περιεχόμενων blocks
Συνάρτηση εξωτερικών εξόδων	Συνδέσεις από *out* flow ports περιεχόμενων blocks προς *out* flow ports του περιέχοντος block

στενά συνδεδεμένες με τις καταστάσεις του μοντέλου και τις μεταβάσεις τους.

Από την άλλη πλευρά, υπάρχει και η συνάρτηση εξωτερικών μεταβάσεων, που ενεργοποιείται από εξωτερικά γεγονότα και οδηγεί σε μεταβολή κατάστασης. Κάτι τέτοιο μπορεί να περιγραφεί σε ADs, όπου προβλέπεται και η αναπαράσταση εξωτερικής εισόδου.

Ο πίνακας 5.1 συνοψίζει τις αντιστοιχίσεις μεταξύ DEVS και SysML σε ένα γενικό επίπεδο. Το SysML προφίλ για DEVS βασίζεται στη λογική των αναφερόμενων ομοιοτήτων και αντιστοιχήσεων, αλλά παρουσιάζεται στη συνέχεια με τον ορισμό των στερεοτύπων (stereotypes) και των περιορισμών (constraints) του.

5.2 SysML Προφίλ για DEVS

Ο τυπικός τρόπος επέκτασης της UML (ή και επεκτάσεων αυτής, όπως η SysML) για τη βελτιστοποίηση της μοντελοποίησης σε ένα συγκεκριμένο πλαίσιο ή πεδίο, είναι ο ορισμός ενός συνόλου *στερεοτύπων*. Αυτά χαρακτηρίζουν συγκεκριμένα στοιχεία της SysML, δίνουν τη δυνατότητα ορισμού πρόσθετων πεδίων και μαζί με ένα σύνολο *περιορισμών (constraints)*, οργανώνονται σε ένα προφίλ. Με αυτό τον τρόπο εμπλουτίζονται οι εκφραστικές δυνατότητες της SysML, αλλά ελέγχεται και η ορθότητα του μοντέλου μέσω της ικανοποίησης των περιορισμών.

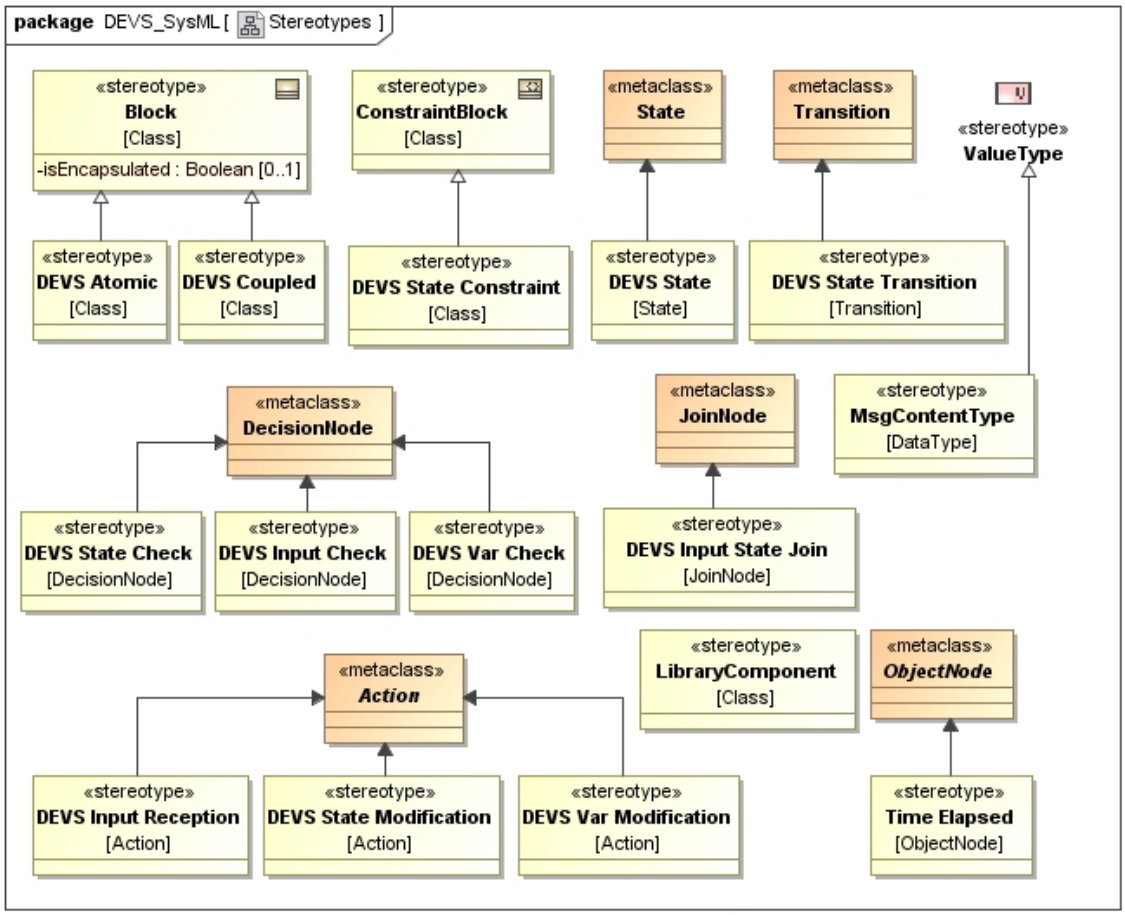
Το SysML προφίλ για DEVS, ορίζει ένα σύνολο επεκτάσεων και περιορισμών στα διαγράμματα και τα στοιχεία μοντελοποίησης της SysML. Οι επεκτάσεις παρέχουν τη δυνατότητα

συμπλήρωσης του μοντέλου συστήματος με χαρακτηριστικά προσομοίωσης, ενώ οι περιορισμοί διασφαλίζουν ότι η συμπλήρωση έχει γίνει επιτυχώς, ώστε να εξαλειφθούν οι ασυνέπειες στο εκτελέσιμο μοντέλο προσομοίωσης, που θα παραχθεί στη συνέχεια.

Το SysML προφίλ για DEVS έχει οριστεί στο MagicDraw, όπου δηλώθηκαν τα προτεινόμενα στερεότυπα, σύμφωνα με τις ομοιότητες μεταξύ SysML και DEVS, που περιγράφονται στο 5.1.2. Οι περιορισμοί του προφίλ ορίστηκαν με χρήση OCL, με την αξιοποίηση των δυνατοτήτων προσαρμογής του εργαλείου, καθώς και με ανάπτυξη plugins, σύμφωνα με την παρεχόμενη Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface) (API).

5.2.1 Στερεότυπα και Περιορισμοί του SysML Προφίλ για DEVS

Τα στερεότυπα του SysML προφίλ για DEVS παρουσιάζονται συνοπτικά στο σχήμα 5.1. Στη συνέχεια, τα στερεότυπα και οι περιορισμοί αναλύονται ανά νοηματική ομάδα σε αντίστοιχους πίνακες. Οι πίνακες περιγράφουν το υπό εξέταση στοιχείο DEVS, τη βασική κλάση ή το στερεότυπο UML/SysML, που επεκτείνεται, το όνομα του νέου στερεοτύπου, τα tagged values του και τους περιορισμούς, όπου υπάρχουν.



Σχήμα 5.1: Τα στερεότυπα του SysML προφίλ για DEVS.

Στερεότυπα και Περιορισμοί για τη Δομή στο SysML προφίλ για DEVS Στον πίνακα 5.2 παρουσιάζονται τα στερεότυπα και οι περιορισμοί του SysML προφίλ για DEVS που αφο-

ρούν τα βασικά δομικά χαρακτηριστικά των μοντέλων, σύμφωνα με τις ομοιότητες και τις αντιστοιχίσεις μεταξύ DEVS και SysML, που αναλύθηκαν στο 5.1.2.

Το μοντέλο DEVS περιλαμβάνεται σε ένα BDD που χαρακτηρίζεται ως «DEVS Model Diagram». Σε αυτό περιέχεται ένα block χαρακτηρισμένο ως «Simulation Specs», το οποίο περιγράφει τα χαρακτηριστικά εκτέλεσης της προσομοίωσης (χρόνο προσομοίωσης και επαναλήψεις). Εντός του διαγράμματος τα blocks χαρακτηρίζονται ως «DEVS Atomic» ή ως «DEVS Coupled», ανάλογα με το αν είναι απλά blocks, για τα οποία περιγράφεται η συμπεριφορά, ή σύνθετα blocks, αποτελούμενα από άλλα «DEVS» blocks.

Τόσο τα Atomic όσο και τα Coupled blocks, έχουν ports, που χαρακτηρίζονται ως «DEVS Input Port» ή ως «DEVS Output Port». Στα ports έχουν προβλεφθεί πεδία, για τη συμπλήρωση των αποτελεσμάτων της προσομοίωσης (συνολικό πλήθος, ελάχιστος και μέγιστος αριθμός, μέσος όρος μηνυμάτων και απόκλιση).

Στον πίνακα 5.3 παρουσιάζονται τα στερεότυπα και οι περιορισμοί του SysML προφίλ για DEVS που αφορούν την ιεραρχική σύνθεση των coupled μοντέλων, σύμφωνα με τις ομοιότητες και τις αντιστοιχίσεις μεταξύ DEVS και SysML, που αναλύθηκαν στο 5.1.2.

Οι διασυνδέσεις μεταξύ των ports των περιεχόμενων blocks ορίζονται σε ένα IBD, χαρακτηρισμένο ως «DEVS Coupled Internal Diagram». Όλες οι συνδέσεις είναι χαρακτηρισμένες με τα αντίστοιχα στερεότυπα, ανάλογα με το αν πρόκειται για σύνδεση μεταξύ ports των περιεχόμενων blocks ή μεταξύ ports του Coupled μοντέλου και εσωτερικών blocks.

Στον πίνακα 5.4 παρουσιάζονται τα στερεότυπα και οι περιορισμοί του SysML προφίλ για DEVS που αφορούν τον ορισμό της κατάστασης των Atomic μοντέλων.

Οι καταστάσεις ορίζονται σε ένα BDD, που περιέχεται στο Atomic μοντέλο και είναι χαρακτηρισμένο ως «DEVS States Definition». Κάθε κατάσταση ορίζεται ως ένα «DEVS State Constraint» block, με τουλάχιστον ένα constraint και ένα parameter. Επίσης, έχουν προβλεφθεί πεδία, για τη συμπλήρωση των αποτελεσμάτων της προσομοίωσης (συνολικό πλήθος, ελάχιστος και μέγιστος αριθμός, μέσος όρος περιπτώσεων κατά τις οποίες το μοντέλο έλαβε τη συγκεκριμένη κατάσταση και απόκλιση). Το στερεότυπο «DEVS State Variable» χρησιμοποιείται για το χαρακτηρισμό των μεταβλητών που καθορίζουν την κατάσταση. Σε ένα PD συσχετίζονται οι μεταβλητές κατάστασης με τα parameters των «DEVS State Constraint» blocks, εκφράζοντας τις σχέσεις που υπάρχουν μεταξύ τιμών μεταβλητών κατάστασης και καταστάσεων.

Στον πίνακα 5.5 παρουσιάζονται τα στερεότυπα και οι περιορισμοί του SysML προφίλ για DEVS που αφορούν τις εσωτερικές μεταβάσεις κατάστασης των Atomic μοντέλων DEVS.

Οι εσωτερικές μεταβάσεις κατάστασης ορίζονται σε ένα SMD («DEVS Atomic Internal Model»), που περιέχεται στο Atomic μοντέλο. Σε αυτό υπάρχει ένα «DEVS State» για κάθε «DEVS State Constraint», που ορίστηκε στο «DEVS State Definition». Από κάθε «DEVS State» μπορεί να ξεκινάει μόνο ένα «DEVS State Transition» προς ένα άλλο ή το ίδιο «DEVS State». Στη μετάβαση συμπληρώνονται μόνο το time guard condition, που καθορίζει το χρόνο παραμονής στην κατάσταση αφετηρίας, και ένα output effect για τον προσδιορισμό του παραγόμενου output από συγκεκριμένο «DEVS Output Port» του Atomic μοντέλου.

Στον πίνακα 5.6 παρουσιάζονται τα στερεότυπα και οι περιορισμοί του SysML προφίλ για

Πίνακας 5.2: Στερεότυπα και περιορισμοί του SysML προφίλ για DEVS για τη δομή

Στοιχείο DEVS	**Βασική κλάση**	**DEVS Stereotype**
Tagged Values:	**Τύπος**	**Όνομα**
Constraints:	**Ορισμός**	
Περιγραφή μοντέλου DEVS	bdd	«DEVS Model Diagram»
Constraints:	Δεν επιτρέπεται τα blocks που ορίζονται σε ένα «DEVS Model Diagram» (σε οποιοδήποτε βάθος) να περιέχουν bdd με το ίδιο στερεότυπο («DEVS Model Diagram»).	
	Περιέχει υποχρεωτικά μόνο ένα «Simulation Specs» block.	
Στοιχεία Προσομοίωσης	SysML::Block	«Simulation Specs»
Tagged Values:	UML::Real	simulationTime
	UML::Integer	simulationRuns
Μοντέλο DEVS	SysML::Block	Abstract «DEVS»
Tagged Values:	UML::String	modelName
DEVS port	SysML::Ports&Flows::ItemFlow	«DEVS Port»
Tagged Values:	UML::String	portName
	UML::String	msgType
	UML::Integer	total
	UML::Integer	min
	UML::Integer	max
	UML::Float	avg
	UML::Float	deviation
DEVS input	DEVSys::DEVS Port	«DEVS Input Port»
Constraints:	Το port πρέπει να έχει direction *in*.	
DEVS output	DEVSysML::DEVS Port	«DEVS Output Port»
Constraints:	Το port πρέπει να έχει direction *out*.	
Atomic Μοντέλο DEVS	DEVSys::DEVS	«DEVS Atomic»
Constraints:	Ένα έγκυρο «DEVS Atomic» block πρέπει να έχει τα εξής τέσσερα διαγράμματα, συσχετισμένα με αυτό, για την περιγραφή της συμπεριφοράς του: «DEVS States Definition», «DEVS States Association», «DEVS Atomic Internal» και «DEVS Atomic External».	
Coupled Μοντέλο DEVS	DEVSys::DEVS	«DEVS Coupled»
Constraints:	Ένα έγκυρο «DEVS Coupled» block πρέπει να έχει το διάγραμμα «DEVS Coupled Internal», συσχετισμένο με αυτό, για την περιγραφή των διασυνδέσεων των περιεχόμενων σε αυτό blocks.	

Πίνακας 5.3: Στερεότυπα και περιορισμοί του SysML προφίλ για DEVS για την ιεραρχική σύνθεση των coupled μοντέλων

Στοιχείο DEVS ***Tagged Values:*** ***Constraints:***	**Βασική κλάση** **Τύπος** **Ορισμός**	**DEVS Stereotype** **Όνομα**
Εσωτερικές Διασυνδέσεις DEVS Coupled	ibd	«DEVS Coupled Internal Diagram»
Σύνδεση DEVS	SysML::Ports&Flows::ItemFlow	«DEVS Connection»
Constraints:	Τα δύο άκρα της σύνδεσης πρέπει να ονομάζονται *from* και *to* και συνδέονται με κάποιο «DEVS Port».	
Σύνδεση Εξωτερικής Εισόδου	DEVSys::DEVS Connection	«External Input Connection»
Constraints:	Το άκρο *from* συνδέεται σε ένα «DEVS Input Port» του αναλυόμενου «DEVS Coupled» block και το άκρο *to* συνδέεται σε ένα «DEVS Input Port» ενός περιεχόμενου «DEVS» block.	
Εσωτερική Σύνδεση	DEVSys::DEVS Connection	«Internal Connection»
Constraints:	Το άκρο *from* συνδέεται σε ένα «DEVS Output Port» ενός περιεχόμενου «DEVS» block και το άκρο *to* συνδέεται σε ένα «DEVS Input Port» ενός άλλου περιεχόμενου «DEVS» block.	
Σύνδεση Εξωτερικής Εξόδου	DEVSys::DEVS Connection	«External Output Connection»
Constraints:	Το άκρο *from* συνδέεται σε ένα «DEVS Output Port» ενός περιεχόμενου «DEVS» block και το άκρο *to* συνδέεται σε ένα «DEVS Output Port» του «DEVS Coupled» block, που αναλύεται.	

Πίνακας 5.4: Στερεότυπα και περιορισμοί του SysML προφίλ για DEVS για τον ορισμό της κατάστασης Atomic μοντέλων

Στοιχείο DEVS ***Tagged Values:*** ***Constraints:***	**Βασική κλάση** **Τύπος** **Ορισμός**	**DEVS Stereotype** **Όνομα**
Ορισμός καταστάσεων «Atomic DEVS»	bdd	«DEVS States Definition»
Constraints:	Πρέπει να είναι συσχετισμένο με ένα «DEVS Atomic» block.	
Κατάσταση Atomic μοντέλου	SysML::ConstraintBlocks::ConstraintBlock	«DEVS State Constraint»
Tagged Values:	UML::Integer	total
	UML::Integer	min
	UML::Integer	max
	UML::Float	avg
	UML::Float	deviation
Constraints:	Έχει τουλάχιστον ένα constraint. Έχει τουλάχιστον ένα parameter.	
Μεταβλητή Κατάστασης	SysML::Blocks::ValueProperty	«DEVS State Variable»
Συσχέτιση καταστάσεων με μεταβλητές κατάστασης	pd	«DEVS State Association Diagram»
Constraints:	Συσχετίζει όλα τα parameters των καταστάσεων DEVS με μεταβλητές κατάστασης.	

Πίνακας 5.5: Στερεότυπα και περιορισμοί του SysML προφίλ για DEVS για τις εσωτερικές μεταβάσεις κατάστασης

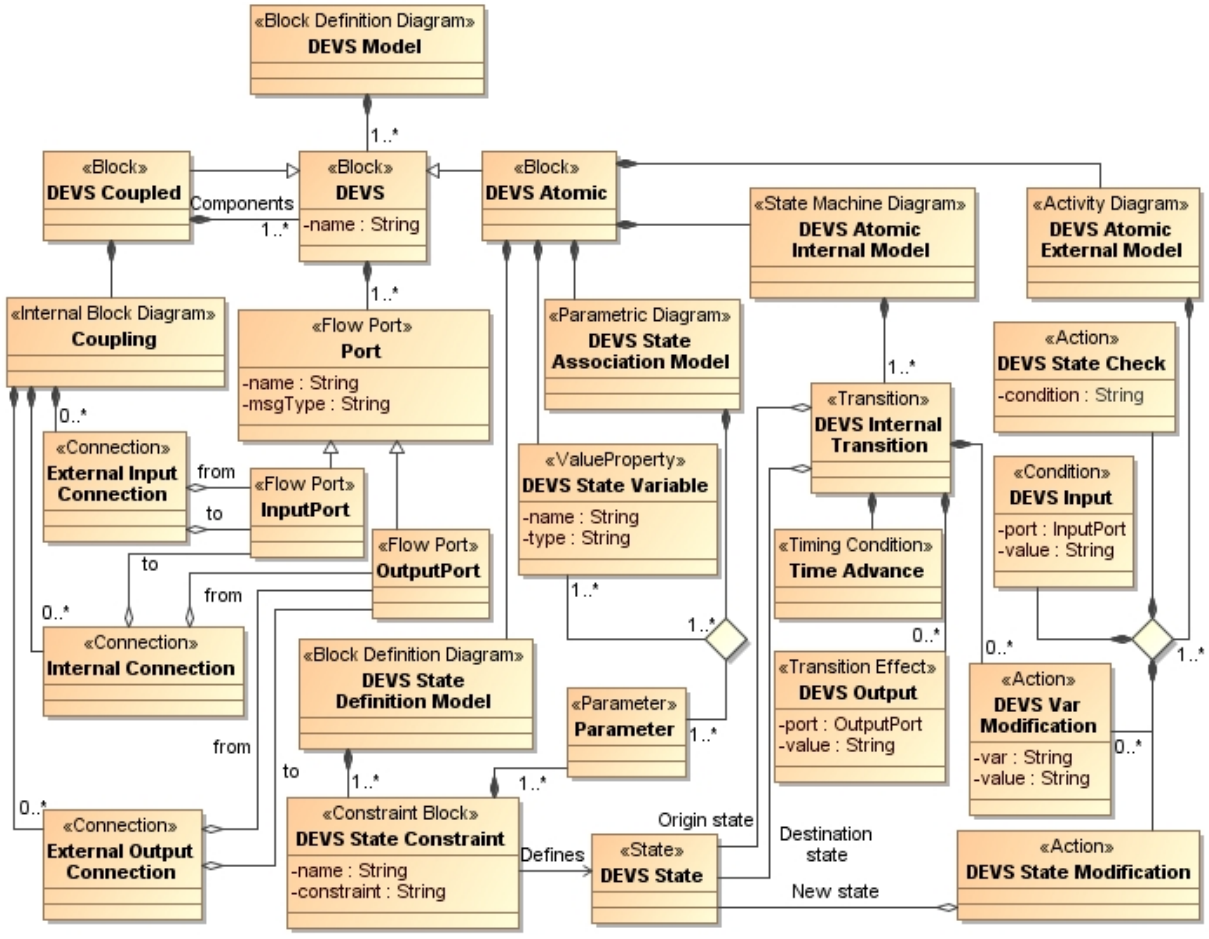
Στοιχείο DEVS ***Tagged Values:*** ***Constraints:***	**Βασική κλάση** **Τύπος** **Ορισμός**	**DEVS Stereotype** **Όνομα**
Εσωτερικές μεταβάσεις Atomic μοντέλου	smd	«DEVS Atomic Internal Model»
Constraints:	Περιέχει έναν αρχικό κόμβο και ένα «DEVS State» για κάθε «DEVS State Constraint» που έχει οριστεί στο «DEVS States Definition» του Atomic μοντέλου.	
Κατάσταση DEVS	UML::StateMachines::State	«DEVS State»
Constraints:	Αντιστοιχεί σε ένα «DEVS State Constraint» και έχει το ίδιο όνομα με αυτό. Μπορεί να ξεκινάει μόνο ένα transition από ένα «DEVS State».	
Εσωτερική μετάβαση κατάστασης Atomic μοντέλου	UML::StateMachines::Transition	«DEVS State Transition»
Constraints:	Μπορεί να έχει συμπληρωμένα μόνο την time guard condition (ta) και ένα output effect, όπου ανατίθεται τιμή σε ένα «DEVS Output Port».	

DEVS που αφορούν την ανταπόκριση ενός Atomic μοντέλου σε εξωτερικά γεγονότα.

Η περιγραφή της συμπεριφοράς σε εξωτερικά συμβάντα γίνεται σε ένα AD, το «DEVS Atomic External Model», που περιέχεται στο εξεταζόμενο Atomic μοντέλο. Για την επιλογή των προβλεπόμενων ενεργειών γίνεται συνδυασμός της εισόδου («DEVS In Port») με την τρέχουσα κατάσταση («DEVS State Check» και «DEVS State Fork») με κόμβους τύπου «DEVS State Input Join». Οι ενέργειες μεταβάλουν είτε μεταβλητές κατάστασης («DEVS State Modification Action»), είτε απλές μεταβλητές («DEVS Variable Modification Action»).

5.2.2 Σύνοψη του SysML Προφίλ για DEVS

Το σχήμα 5.2 παρέχει σε ένα class diagram μία σύνοψη των στερεοτύπων του SysML προφίλ για DEVS, αλλά και τον τρόπο με τον οποίο αυτά συσχετίζονται μεταξύ τους. Έτσι, διαφαίνονται και αρκετοί από τους περιορισμούς του προφίλ, που αποτυπώθηκαν στους προηγούμενους πίνακες.



Σχήμα 5.2: Το class diagram με τις συσχετίσεις των στερεοτύπων του SysML προφίλ για DEVS.

Πίνακας 5.6: Στερεότυπα και περιορισμοί του SysML προφίλ για DEVS, για τη συμπεριφορά του μοντέλου στην παραλαβή εξωτερικών συμβάντων

Στοιχείο DEVS ***Tagged Values:*** ***Constraints:***	**Βασική κλάση** **Τύπος** **Ορισμός**	**DEVS Stereotype** **Όνομα**
Συμπεριφορά Atomic μοντέλου σε εξωτερικά συμβάντα	ad	«DEVS Atomic External Model»
Constraints:	Περιλαμβάνει μόνο έναν αρχικό κόμβο, ένα «DEVS State Check», παραμέτρους «DEVS In Port» (αντίστοιχες των input ports του block) και στοιχεία «DEVS State Fork», «DEVS State Input Join» και «DEVS State Modification Actions», καθώς και τελικούς κόμβους.	
Έλεγχος κατάστασης μοντέλου	UML::Activities::DecisionNode	«DEVS State Check»
Constraints:	Οι guard conditions που ξεκινούν από αυτό το κόμβο ελέγχουν τα state variables και καταλήγουν σε κόμβους «DEVS State Fork» ή «DEVS State Input Join».	
Είσοδος εξωτερικού συμβάντος	UML::Activities ActivityParameterNode	«DEVS In Port»
Constraints:	Πρέπει να συνδέεται μόνο προς ένα «DEVS State Input Join».	
Πολλαπλή εμφάνιση καταστάσεων για συνδυασμό με είσοδο	UML::Activities ForkNode	«DEVS State Fork»
Constraints:	Συνδέεται προς ένα «DEVS State Input Join».	
Συνδυασμός εισόδου και κατάστασης	UML::Activities::JoinNode	«DEVS State Input Join»
Constraints:	Συνδέεται μόνο προς ένα «DEVS State Modification Action» ή προς ένα τελικό κόμβο.	
Μεταβολή κατάστασης μοντέλου	UML::Actions::Action	«DEVS State Modification Action»
Constraints:	Αναθέτει μία τιμή σε μία μεταβλητή κατάστασης του block. Συνδέεται μόνο προς μία άλλη «DEVS State Modification Action», προς μία «DEVS Variable Modification Action» ή προς έναν τελικό κόμβο.	
Μεταβολή μεταβλητών μοντέλου	UML::Actions::Action	«DEVS Variable Modification Action»
Constraints:	Αναθέτει μία τιμή σε μία value property του block. Συνδέεται μόνο προς μία άλλη «DEVS Variable Modification Action» ή προς έναν τελικό κόμβο.	

5.3 Μετα-μοντέλο MOF για το DEVS

Απαραίτητο συστατικό και με δεσπόζουσα σημασία στην υλοποίηση του πλαισίου, σύμφωνα με την προτεινόμενη μεθοδολογία είναι η επιλογή ή δημιουργία ενός μετα-μοντέλου MOF για την επιλεγμένη μεθοδολογία προσομοίωσης. Στην περίπτωση του DEVS, έχουν γίνει διάφορες προτάσεις για την αναπαράσταση μοντέλων με δηλωτικό τρόπο. Όπως όμως αναλύεται και στο 3.2.2, καμία από αυτές δεν είναι βασισμένη σε ένα μετα-μοντέλο MOF.

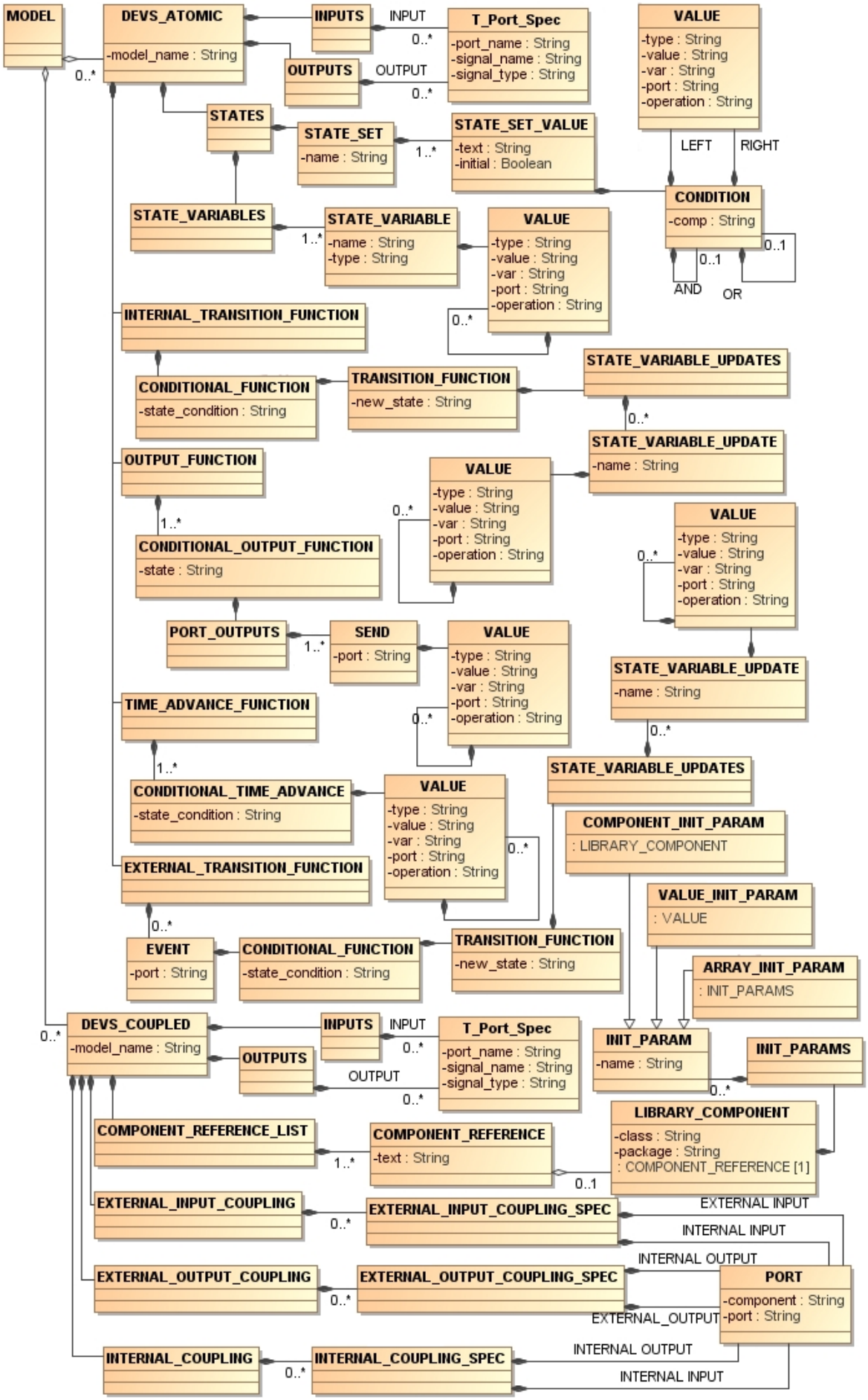
Έτσι, για την κάλυψη των απαιτήσεων του πλαισίου δημιουργήθηκε ένα μετα-μοντέλο MOF για DEVS. Για τον ορισμό του μετα-μοντέλου αξιοποιήθηκε σε κάποιο βαθμό η DEVS-XML [48], που αποτελούσε την πιο ολοκληρωμένη από τις προτάσεις. Στο Διάγραμμα Κλάσεων (Class Diagram) (CD) του σχήματος 5.3 παρουσιάζεται σχηματικά το υλοποιημένο μετα-μοντέλο για DEVS. Περιλαμβάνει στοιχεία για τον ορισμό της δομής και της συμπεριφοράς των Atomic DEVS μοντέλων (ports, states, internal transition, output, time advance και external transition functions) και των Coupled DEVS μοντέλων (ports και coupling).

Η δομή τόσο των Atomic, όσο και των Coupled μοντέλων ορίζεται με παρόμοιο τρόπο, όπως προτείνεται και στο [48]. Η συμπεριφορά των μοντέλων DEVS, δηλαδή οι συναρτήσεις των Atomic DEVS μοντέλων, περιγράφεται με βάση τις μεταβάσεις κατάστασης του συστήματος, επίσης σε συμφωνία με την DEVS-XML. Όμως το νέο μετα-μοντέλο για DEVS περιλαμβάνει και τον ορισμό των καταστάσεων του συστήματος με βάση τις μεταβλητές κατάστασης. Έτσι, είναι δυνατή η αξιοποίηση είτε των μεταβλητών κατάστασης, είτε των καταστάσεων του συστήματος για την περιγραφή των συναρτήσεων συμπεριφοράς, ανάλογα με τον χρησιμοποιούμενο προσομοιωτή.

Οι περισσότερες από τις συσχετίσεις σύνθεσης (composition associations) του CD, που περιγράφει το μετα-μοντέλο MOF για DEVS δεν έχουν ονομαστεί. Σε αυτές τις περιπτώσεις υπονοείται ότι το όνομα της περιεχόμενης οντότητας χρησιμοποιείται για την αναφορά της, στο context της περιέχουσας οντότητας. Έτσι, επιτυγχάνεται μία απλοποίηση του διαγράμματος. Ωστόσο, σε περιπτώσεις όπου υπάρχει αναφορά σε μία κλάση, δύο ή περισσότερες φορές, από την ίδια οντότητα (π.χ. input και output ports), ένα διακριτό όνομα χρησιμοποιείται για κάθε συσχέτιση σύνθεσης.

Επισημαίνεται ότι ο ορισμός των κλάσεων CONDITION και VALUE είναι αναδρομικός. Για παράδειγμα, μία οντότητα VALUE μπορεί να συντεθεί από μία ή περισσότερες άλλες οντότητες VALUE. Αυτό είναι χρήσιμο και επιτρέπεται μόνο στον ορισμό μίας πράξης, ώστε να είναι δυνατός ο ορισμός μίας σύνθετης έκφρασης (π.χ. $v_1 + v_2$). Στην περίπτωση του CONDITION, AND ή OR επιμέρους συνθήκες επιτρέπονται για τη δήλωση σύνθετων συνθηκών (π.χ. $c_1\ AND\ c_2$).

Σε σύγκριση με την DEVS-XML [48], το νέο μετα-μοντέλο MOF για DEVS αντιμετωπίζει τη σχέση μεταξύ μεταβλητών κατάστασης και καταστάσεων. Αυτό το χαρακτηριστικό δίνει τη δυνατότητα εκτέλεσης μοντέλων DEVS σε μία ευρύτερη γκάμα προσομοιωτών DEVS, υλοποιημένων σε C++ ή Java.



Σχήμα 5.3: Το μετα-μοντέλο για DEVS.

Επίσης, αντιμετωπίζει πολύπλοκες εκφράσεις τιμών, αξιοποιώντας την κλάση VALUE, που χρησιμοποιείται για τη δημιουργία συνθηκών καταστάσεων και ενημερώσεων τιμών μεταβλητών κατάστασης, παρέχοντας παράλληλα πιο απλή δομή και εννοιολογική συνοχή. Τιμές για σύνθετους τύπους θα μπορούσαν να υποστηριχθούν με τη σειριοποίησή τους σε ένα String value property. Ωστόσο αυτή η δυνατότητα δεν έχει εξεταστεί διεξοδικά.

Τέλος, στο σχήμα 5.3 παρουσιάζεται η επεκτεταμένη μορφή του μετα-μοντέλου, όπου σε ένα Coupled μοντέλο, εκτός από άλλα Coupled και Atomic μοντέλα, μπορούν να περιλαμβάνονται και αναφορές σε υλοποιημένα συστατικά λογισμικού προσομοίωσης. Αυτό προσδιορίζεται με τη χρήση του στοιχείου *LIBRARY_COMPONENT* και των απαιτούμενων παραμέτρων αρχικοποίησης (στοιχεία *INIT_PARAMS*, που μπορεί να περιλαμβάνει ένα σύνολο από στοιχεία *INIT_PARAM*). Οι παράμετροι αρχικοποίησης μπορούν να είναι (α) τιμές (*VALUE_INIT_PARAM*), (β) διανύσματα τιμών (*ARRAY_INIT_PARAM* ή (γ) άλλο συστατικό λογισμικού προσομοίωσης (*COMPONENT_INIT_PARAM*).

Οι ανωτέρω δυνατότητες βελτιώνουν την περιγραφική δυνατότητα του μετα-μοντέλου και διευκολύνουν το μετασχηματισμό σε εκτελέσιμο DEVS κώδικα. Έτσι, το μετα-μοντέλο αυτό παρέχει μία πρότυπη, συμπαγή, αξιοποιήσιμη θεμελίωση για τον ορισμό μοντέλων DEVS. Από πρακτική άποψη, το γεγονός ότι πρόκειται για ένα μετα-μοντέλο, που έχει οριστεί με όρους MOF, το καθιστά αξιοποιήσιμο σε συμβατά με πρότυπα εργαλεία διαχείρισης μοντέλων, όπως απαιτείται από το πλαίσιο.

5.4 Μετασχηματισμός: Μοντέλα SysML με προφίλ DEVS προς μοντέλα DEVS

Για τη μετατροπή των μοντέλων συστημάτων (SysML με προφίλ DEVS) σε μοντέλα DEVS, απαιτείται ο κατάλληλος μετασχηματισμός από το μετα-μοντέλο της UML στο μετα-μοντέλο DEVS. Η QVT είναι το κατάλληλο πρότυπο για τον ορισμό τέτοιων μετασχηματισμών, καθώς ο κύριος σκοπός της είναι ο ορισμός αντιστοιχήσεων και μετασχηματισμών ανάμεσα σε δύο μετα-μοντέλα MOF.

Στα πλαίσια της διαμόρφωσης του προτεινόμενου πλαισίου, ορίστηκε ένα σύνολο από σχέσεις QVT, που υλοποιούν το μετασχηματισμό, ανάμεσα σε (α) στοιχεία SysML με προφίλ DEVS και (β) στοιχεία του μετα-μοντέλου DEVS. Η χρήση σχέσεων QVT για τη ενδοσκόπηση του μοντέλου συστήματος (SysML) διευκολύνει τον εντοπισμό και την αξιοποίηση των στοιχείων που αφορούν προσομοίωση, με βάση τα στερεότυπα τους και τις σχέσεις τους με άλλα στοιχεία, ανεξάρτητα από τον αριθμό και την πολυπλοκότητα των διαγραμμάτων του μοντέλου συστήματος. Αυτό συμβαίνει διότι σε κάθε σχέση QVT, λαμβάνονται υπόψη οι παράμετροι ενός συγκεκριμένου υποσυνόλου του μοντέλου συστήματος και, σε δεύτερο χρόνο, είναι δυνατή η εξέταση και άλλων υποσυνόλων του μοντέλου συστήματος με μετάβαση σε άλλες σχέσεις QVT που προσδιορίζονται στο μέρος *where* της υπό εξέταση σχέσης.

Στο σχήμα 5.4 απεικονίζεται συνοπτικά ο μετασχηματισμός μοντέλων DEVS SysML (αριστερή πλευρά) σε μοντέλα DEVS (δεξιά πλευρά). Με το σχήμα αυτό παρέχεται μία σχηματική

αναπαράσταση των αντιστοιχήσεων τμημάτων του μετα-μοντέλου συστήματος σε τμήματα του μετα-μοντέλου προσομοίωσης. Λεπτομέρειες των δύο μετα-μοντέλων έχουν απαλειφθεί, για την απλοποίηση της αναπαράστασης.

Τα αριθμημένα βέλη, που υποδεικνύουν επιμέρους τμήματα του μετασχηματισμού, περιγράφονται στη συνέχεια, χωρίς να υπονοείται και αντίστοιχη σειρά εκτέλεσης του μετασχηματισμού:

1. Μετασχηματισμός μοντέλου: Το υψηλότερου επιπέδου τμήμα του μετασχηματισμού. Το μοντέλο DEVS SysML περιέχει όλα τα blocks τύπου *DEVS ATOMIC* και *DEVS Coupled*, τα οποία θα μετασχηματιστούν στα αντίστοιχα στοιχεία *DEVS_ATOMIC* και *DEVS_COUPLED* του μετα-μοντέλου DEVS. Υλοποιείται από μία σχέση QVT, η οποία εντοπίζει όλα τα μοντέλα στο DEVS SysML (φυσιολογικά θα υπάρχει μόνο ένα) και δημιουργεί ένα μοντέλο DEVS, για καθένα από αυτά. Σε αυτό το σημείο, οι σχέσεις *AtomicBlock2DevsAtomic* και *CoupledBlock2Devs Coupled* εφαρμόζονται σε κάθε μοντέλο, ώστε να αναγνωριστούν όλα τα στοιχεία DEVS.

2. Μετασχηματισμός των κοινών στοιχείων DEVS: Αυτό το τμήμα μετασχηματίζει τα στοιχεία που είναι κοινά σε atomic και coupled μοντέλα DEVS, π.χ. όνομα μοντέλου και πόρτες εισόδου/εξόδου.

3. Μετασχηματισμός ορισμού κατάστασης atomic DEVS: Μετασχηματίζει τα DEVS Constraint Blocks, που ορίζουν το σύνολο των καταστάσεων.

4. Μετασχηματισμός ορισμού μεταβλητών κατάστασης DEVS atomic: Μετασχηματίζει τα DEVS State Variable properties σε DEVS State Variables.

5. Μετασχηματισμός συνδέσμων καταστάσεων DEVS atomic: Μετασχηματίζει το DEVS State Association Model σε συνθήκες συνημμένες σε τιμές του συνόλου καταστάσεων.

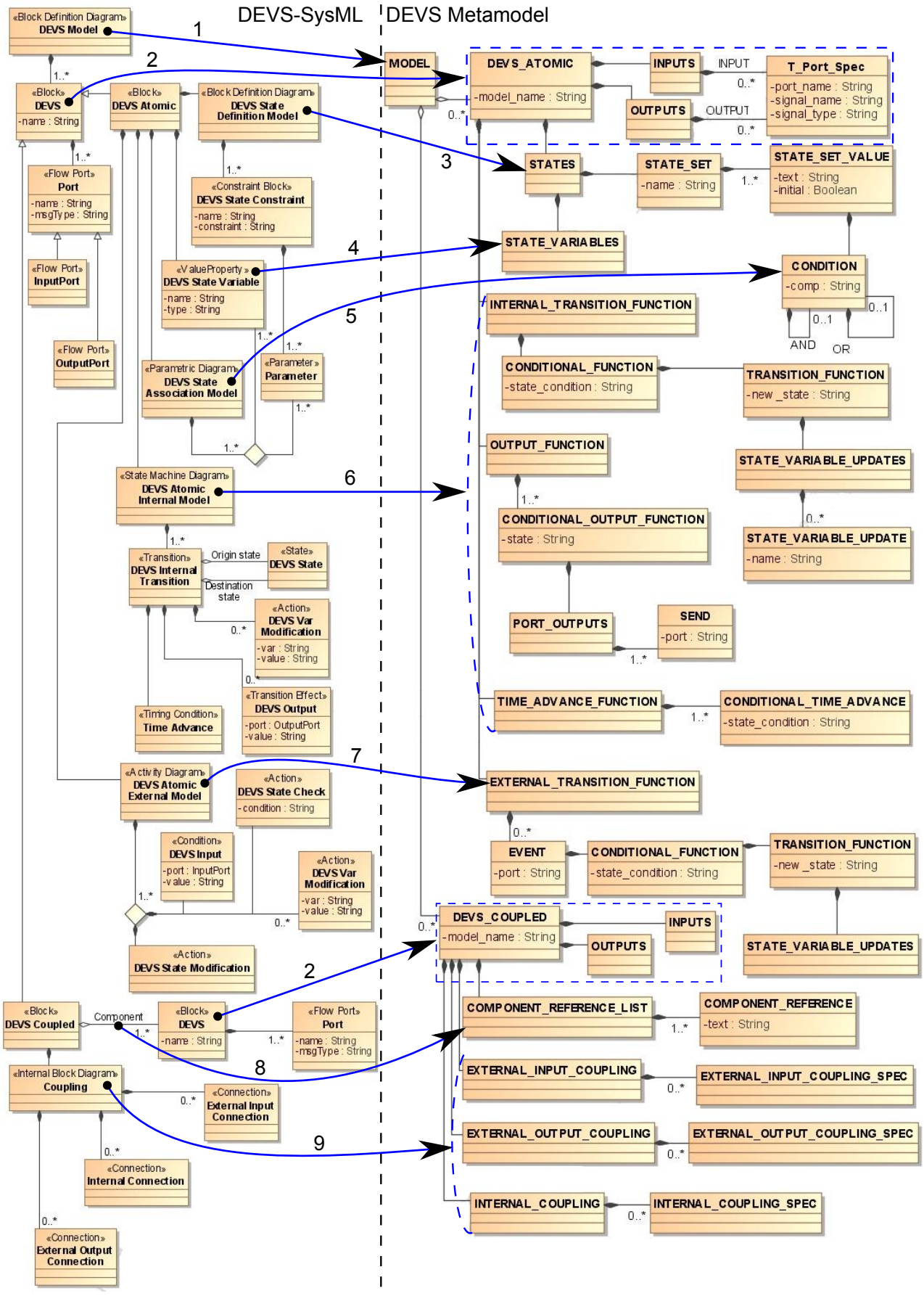
6. Μετασχηματισμός εσωτερικού μοντέλου DEVS atomic: Μετασχηματίζει το DEVS Atomic Internal Model (SMD) σε Internal Transition Function, Output Function και Time Advance Function.

7. Μετασχηματισμός εξωτερικού μοντέλου DEVS atomic: Μετασχηματίζει το DEVS Atomic External Model (AD) σε External Transition Function.

8. Μετασχηματισμός των στοιχείων DEVS coupled: Μετασχηματίζει τους συνδέσμους σύνθεσης τύπου *Component* στη λίστα αναφορών συστατικών.

9. Μετασχηματισμός συζεύξεων DEVS coupled: Μετασχηματίζει τις συνδέσεις *port* από το IBD σε External Input Coupling, External Output Coupling και Internal Coupling.

Σύμφωνα με το σχήμα 5.4, ο μετασχηματισμός DEVS Atomic Internal Model αντιστοιχεί στο βέλος 6. Το DEVS Atomic Internal Model, που έχει οριστεί ως ένα στερεότυπο του UML2 SMD, μετασχηματίζεται στις οντότητες INTERNAL_TRANSITION_FUNCTION, OUTPUT_FUNCTION



Σχήμα 5.4: Η αντιστοίχιση στοιχείων SysML με προφίλ DEVS σε στοιχεία μετα-μοντέλου DEVS.

και TIME_ADVANCE_FUNCTION του μετα-μοντέλου DEVS, αναπαριστώντας τις σχετικές συναρτήσεις DEVS. Κάθε μία από αυτές κατασκευάζεται χρησιμοποιώντας πληροφορία που περιλαμβάνεται στο αντίστοιχο SMD. Οι σχέσεις QVT που υλοποιούν αυτό τμήμα του μετασχηματισμού περιλαμβάνονται στο Παράρτημα Α΄.1.

Ένα εργαλείο υποστήριξης μετασχηματισμών QVT (Medini), βασισμένο στην πλατφόρμα ανάπτυξης εφαρμογών Eclipse, χρησιμοποιήθηκε με επιτυχία για το μετασχηματισμό διαφόρων μοντέλων SysML με προφίλ DEVS σε μοντέλα DEVS.

5.5 Μετασχηματισμός: Μοντέλα DEVS προς Εκτελέσιμο Κώδικα Προσομοίωσης

Με τα στοιχεία του πλαισίου *DEVSys*, που περιγράφηκαν μέχρι τώρα, είναι εφικτή η παραγωγή μοντέλων DEVS από μοντέλα συστήματος (SysML). Καθώς, το μετα-μοντέλο DEVS είναι νέα πρόταση, δεν υποστηρίζεται από κάποιο προσομοιωτή DEVS. Επομένως, για την εκτέλεση των αυτόματα παραγόμενων μοντέλων προσομοίωσης απαιτείται η επέκταση ενός υπάρχοντος προσομοιωτή DEVS ή ο περαιτέρω μετασχηματισμός του μοντέλου προσομοίωσης DEVS, σε μορφή εκτελέσιμη σε κάποιο προσομοιωτή.

Στη συγκεκριμένη περίπτωση επιλέχθηκε η πλατφόρμα *Βασισμένη στην XML Γλώσσα για Συστατικά Λογισμικού Προσομοίωσης (XML-based Language for Simulation Components) (XLSC) DEVS* [33], η οποία παρουσιάζει τα ακόλουθα χαρακτηριστικά:

- έχει έναν αρκετά υψηλού επιπέδου τρόπο ορισμού των μοντέλων, γεγονός που απλοποιεί την αντιστοίχηση στοιχείων του μοντέλου DEVS σε στοιχεία της XLSC.
- ο ορισμός των εκτελέσιμων προγραμμάτων προσομοίωσης γίνεται σε κάποια μορφή XML, οπότε είναι δυνατή η αξιοποίηση τεχνικών, γλωσσών και εργαλείων μετασχηματισμού XML αναπαραστάσεων.
- είναι διαθέσιμη προς χρήση, οπότε δεν τίθεται θέμα δυνατότητας αξιοποίησής της.

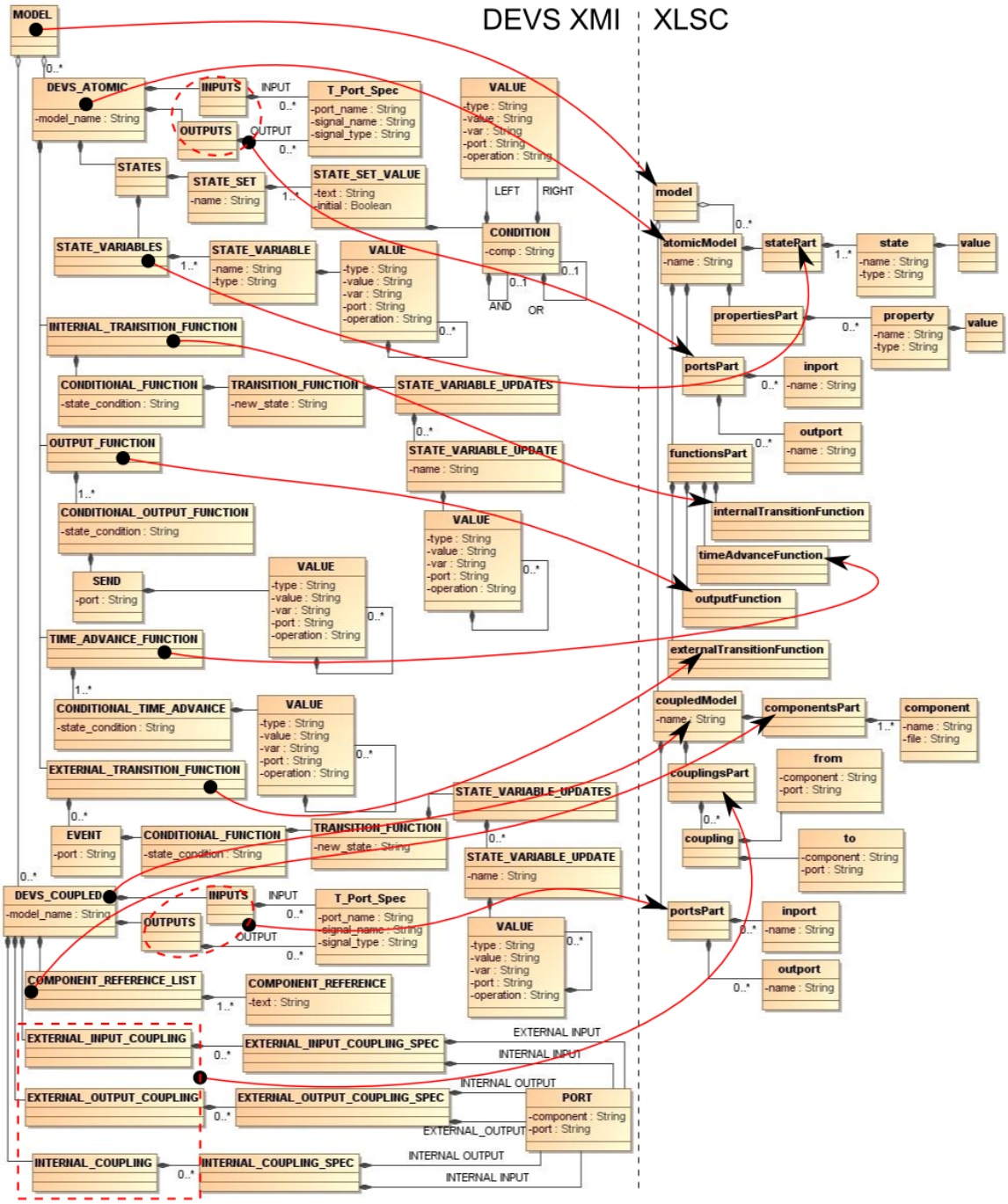
Επομένως, υλοποιήθηκε ένας μετασχηματισμός XSLT, για την παραγωγή αρχείων XLSC από μοντέλα DEVS σε μορφή XMI. Καθώς οι δύο μορφές αναπαραστάσεις μοντέλων DEVS έχουν κυρίως απλές συντακτικές διαφοροποιήσεις, ο μετασχηματισμός υλοποιήθηκε με ένα σύνολο από XSLT templates, τα οποία ταιριάζουν με συγκεκριμένα στοιχεία DEVS και παράγουν τα αντίστοιχα στοιχεία XLSC με τις κατάλληλες τιμές των χαρακτηριστικών τους. Το σχήμα 5.5 αποτυπώνει το μετα-μοντέλο DEVS (αριστερά), τα βασικά δομικά στοιχεία της XLSC (δεξιά) και τα κύρια σημεία του μετασχηματισμού.

Ο μετασχηματισμός XSLT είναι αρκετά απλός, ενώ περισσότερες λεπτομέρειες σχετικά παρέχονται στο Α΄.2. Ωστόσο, αξίζει να σημειωθεί ότι:

- Στην XLSC δεν υπάρχει ορισμός του συνόλου καταστάσεων. Ορίζονται μόνο οι μεταβλητές κατάστασης.

- Στην XLSC όλες οι συναρτήσεις των DEVS atomic (internal transition, output, time advance και external transition) ορίζονται με ένα χαμηλού επιπέδου, διαδικαστικό τρόπο. Ως εκ τούτου, δεν αναλύονται περαιτέρω στο σχήμα 5.5. Στον προτεινόμενο μετασχηματισμό XSLT, βασικά κατασκευάζονται τα διαδικαστικά στοιχεία της XLSC, που υλοποιούν τη συμπεριφορά που δηλώνεται στα αντίστοιχα στοιχεία του μοντέλου DEVS.

- Το coupling αναπαρίσταται με έναν πιο απλό και μονοδιάστατο τρόπο στην XLSC. Δεν υπάρχει διάκριση σε εσωτερικά και εξωτερικά coupling. Για κάθε coupling προσδιορίζονται η πηγή (component, port) και ο προορισμός (component, port). Όταν η πηγή ή ο προορισμός είναι ένα μοντέλο DEVS coupled, τότε ως τιμή component, δίνεται το "this".



Σχήμα 5.5: Η αντιστοίχηση στοιχείων του μετα-μοντέλου DEVS σε στοιχεία XLSC.

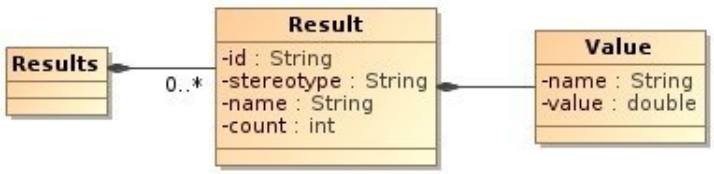
Αλλιώς χρησιμοποιείται το όνομα του μοντέλου DEVS.

- Κατά τη δημιουργία του αρχείου XLSC, λαμβάνεται πρόνοια για την παραγωγή χρήσιμων αποτελεσμάτων προσομοίωσης. Αυτό γίνεται (α) με τη μεταφορά και διατήρηση στα στοιχεία XLSC του αναγνωριστικού (id) των στοιχείων του μοντέλου DEVS και (β) τη δημιουργία πρόσθετων δομών για τη μέτρηση και την αποτύπωση των χαρακτηριστικών απόδοσης των στοιχείων XLSC.

5.6 Εκτέλεση Προσομοίωσης - Αποτελέσματα

Ο κώδικας XLSC που παράγεται από το μοντέλο DEVS εκτελείται από το διερμηνέα XLSC, ο οποίος παράγει δυναμικά DEVS Java κλάσεις για τα περιγραφόμενα μοντέλα DEVS (atomic και coupled). Έτσι, το μοντέλο προσομοίωσης μπορεί να εκτελείται σε περιβάλλοντα προσομοίωσης DEVS Java.

Τα πρόσθετα τμήματα που έχουν ενσωματωθεί στον κώδικα XLSC και ασχολούνται με τη μέτρηση και την καταγραφή των χαρακτηριστικών απόδοσης των στοιχείων παράγουν τα αναμενόμενα αποτελέσματα προσομοίωσης. Τα αποτελέσματα προσομοίωσης αποθηκεύονται σε μορφή σύμφωνη με το μετα-μοντέλο αποτελεσμάτων, που αποτυπώνεται στο σχήμα 5.6. Το στοιχείο αποτελεσμάτων (*Results*) περιλαμβάνουν ένα σύνολο από αποτελέσματα (*Result*). Κάθε *Result* αφορά ένα property ενός στοιχείου του μοντέλου συστήματος και αναφέρεται το μοναδικό αναγνωριστικό (*id*) και το *στερεότυπο* του στοιχείου, καθώς και το όνομα του property (*name*) και το πλήθος των ατομικών αποτελεσμάτων που αξιοποιήθηκαν για την παραγωγή των τελικών συγκεντρωτικών αποτελεσμάτων. Τα συγκεντρωτικά αποτελέσματα (*Value*) περιλαμβάνουν τιμές όπως μέσος όρος, μέγιστο, ελάχιστο και τυπική απόκλιση.

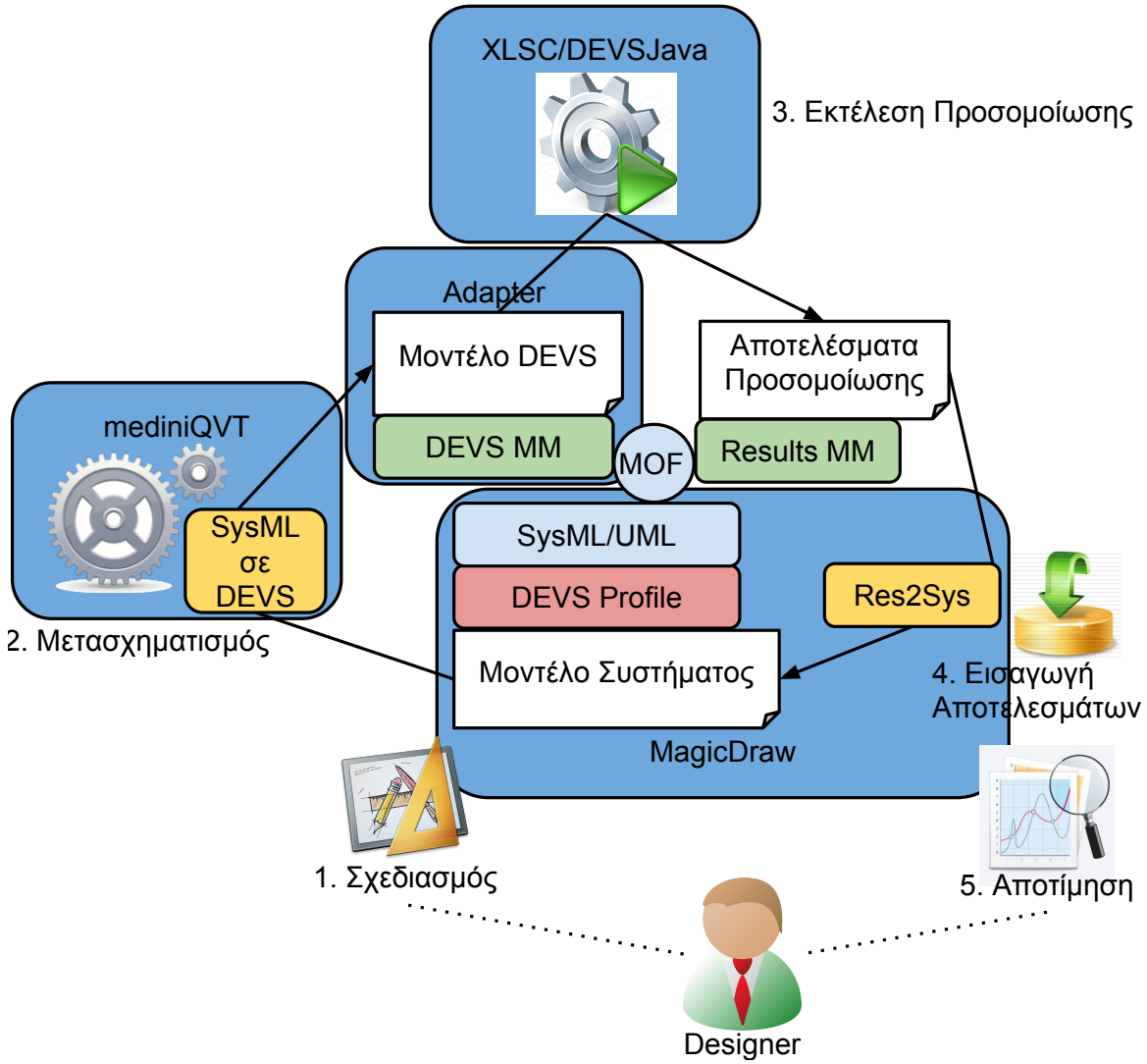


Σχήμα 5.6: Το μετα-μοντέλο για την αποθήκευση των αποτελεσμάτων προσομοίωσης στο DEVSys.

Η εισαγωγή των αποτελεσμάτων αυτών στο μοντέλο συστήματος γίνεται με plugin που αναπτύχθηκε για το εργαλείο μοντελοποίησης UML MagicDraw. Η λύση αυτή προτιμήθηκε έναντι της δημιουργίας ενός άλλου QVT μετασχηματισμού, καθώς πρόκειται για μία απλή ανάθεση τιμών σε προκαθορισμένα σημεία του μοντέλου συστήματος.

5.7 Σχολιασμός Εφαρμογής σε SysML και DEVS

Η εφαρμογή της προσέγγισης για το περιβάλλον προσομοίωσης DEVS ήταν επιτυχής. Το πλαίσιο υλοποιήθηκε στο σύνολό του επιτρέποντας τη λειτουργία της μεθοδολογίας, με όλα τα βήματά της. Έτσι, δόθηκε η δυνατότητα έγκυρου και ολοκληρωμένου ορισμού των μοντέλων συστημάτων και με την αυτοματοποίηση του μετασχηματισμού για την παραγωγή μοντέλων προσομοίωσης, την εκτέλεση της προσομοίωσης και την εισαγωγή των αποτελεσμάτων, αυτά είναι διαθέσιμα στο σχεδιαστή για περαιτέρω αξιολόγηση εντός του μοντέλου συστήματος. Στο σχήμα 5.7 απεικονίζονται τα επιμέρους στοιχεία του υλοποιημένου και λειτουργικού πλαισίου.



Σχήμα 5.7: Πλαίσιο για τη μεθοδολογία προσομοίωσης μοντέλων συστημάτων σε προσομοιωτές DEVS.

Η εμπειρία από την εφαρμογή αυτή ανέδειξε ορισμένα θέματα σχετικά με τις προοπτικές περαιτέρω εφαρμογής της προσέγγισης. Αρχικά, οι προϋπάρχουσες προσπάθειες εξεύρεσης δηλωτικών τρόπων ορισμού μοντέλων προσομοίωσης DEVS, ανεξάρτητα από συγκεκριμένους προσομοιωτές DEVS, είχε οδηγήσει στην οριοθέτηση των θεμελιωδών στοιχείων της συγκεκριμένης μεθοδολογίας προσομοίωσης. Αυτό δημιούργησε ένα υπόβαθρο για την προδιαγραφή του μετα-μοντέλου DEVS, καθώς και του προφίλ SysML/UML για DEVS.

Επίσης, η ύπαρξη εργαλείων για την προσομοίωση μοντέλων DEVS που έχουν οριστεί με δηλωτικό τρόπο αναπαράστασης (XML), όπως π.χ. η XLSC, υποβοήθησε την εκτέλεση των μοντέλων DEVS. Στη συγκεκριμένη περίπτωση, ένας σχετικά απλός XSLT μετασχηματισμός της XMI αναπαράστασης των μοντέλων DEVS σε XLSC XML ήταν αρκετός.

Τέλος, η εισαγωγή των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος με βασισμένο σε πρότυπα τρόπο παρουσίασε κάποιες πρακτικές δυσκολίες. Ενώ σαν προδιαγραφή μετασχηματισμού ο εμπλουτισμός του μοντέλου συστήματος με τα αποτελέσματα προσομοίωσης είναι απλός, στην πράξη η επανεισαγωγή στο εργαλείο μοντελοποίησης του εμπλουτισμένου πλέον με τα στοιχεία προσομοίωσης μοντέλου δεν είναι απροβλημάτιστη. Ως εκ τούτου προτιμήθηκε η υλοποίηση ενός plugin για το χρησιμοποιούμενο εργαλείο μοντελοποίησης (MagicDraw).

Κεφαλαιο 6

Εφαρμογη Προτασης με Περιγραφη Συμπεριφορας - Η Μαχη ως Συστημα

Το πλαίσιο που επιτρέπει την εφαρμογή της μεθοδολογίας υλοποιήθηκε για την περίπτωση της προσομοίωσης με DEVS, όπως περιγράφηκε στο κεφάλαιο 5. Το υλοποιημένο πλαίσιο αξιοποιήθηκε για τη δοκιμαστική εφαρμογή της μεθοδολογίας σε διάφορα μοντέλα συστημάτων. Με την ευρεία έννοια του όρου συστήματος, ως ένα σύνολο συνεργαζόμενων στοιχείων, που παρουσιάζει μία συνολική συμπεριφορά και αλληλεπιδρά με το περιβάλλον του, τα μοντέλα μπορεί να περιγράφουν συστήματα διαφόρων τύπων, όπως φυσικά, μηχανικά, ηλεκτρικά, ηλεκτρονικά, ψηφιακά, επιχειρησιακά, οικονομικά, κοινωνικά και άλλα.

Εδώ παρουσιάζεται ένα παράδειγμα εφαρμογής της μεθοδολογίας για τη μελέτη της απόδοσης της επιχειρησιακής δράσης, κατά τη μάχη μεταξύ δύο πολεμικών μονάδων (divisions), μίας φιλικής και μίας εχθρικής. Η φιλική μονάδα αποτελείται από ένα σύνταγμα (regiment), που ελέγχει μία μονάδα πεζικού (infantry) και μία μονάδα πυροβολικού (artillery). Η εχθρική μονάδα μοντελοποιείται αφαιρετικά ως ένα στοιχείο με συνολική ισχύ, χωρίς να αναλύεται περαιτέρω.

Ο στόχος του συγκεκριμένου παραδείγματος είναι η μελέτη της απόδοσης δύο εναλλακτικών στρατηγικών επίθεσης, που μπορεί να εφαρμόσει η φιλική μονάδα:

- Attack1by1: Το σύνταγμα επιτίθεται προς την εχθρική μονάδα πρώτα με τη μονάδα πυροβολικού και στη συνέχεια με τη μονάδα πεζικού.
- AttackAll: Το σύνταγμα επιτίθεται στον εχθρό με όλες τις διαθέσιμες δυνάμεις ταυτόχρονα.

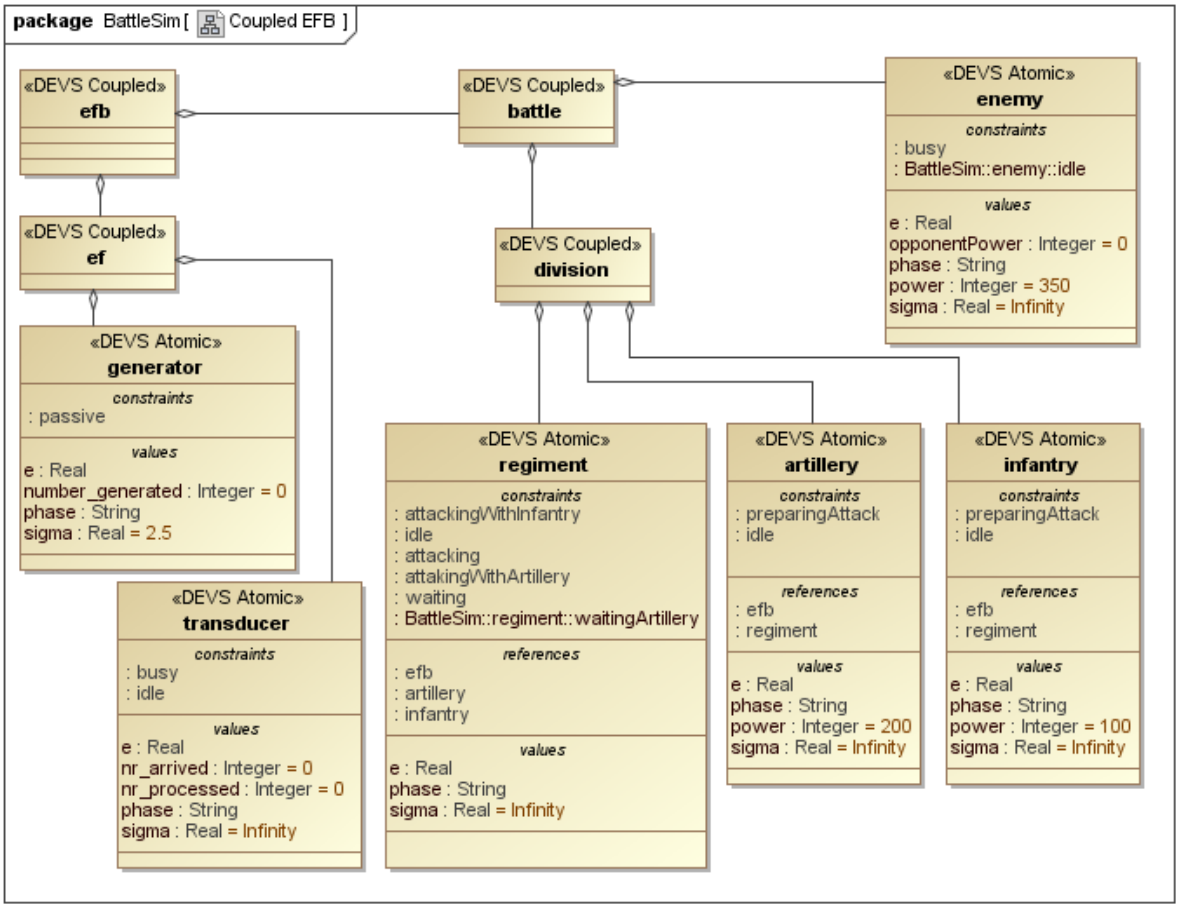
Στο παράδειγμα εφαρμογής που περιγράφεται εδώ, το σύνολο της δομής και της συμπεριφοράς του συστήματος περιγράφεται στο εμπλουτισμένο μοντέλο συστήματος SysML. Δεν

γράφτηκε ούτε μία γραμμή κώδικα προσομοίωσης και δεν αξιοποιήθηκαν έτοιμα συστατικά λογισμικού προσομοίωσης, πέρα από το περιβάλλον εκτέλεσης προσομοίωσης.

Επομένως, η εφαρμογή αυτή δίνει έμφαση στη δυνατότητα παραγωγής και εκτέλεσης μοντέλων προσομοίωσης, χωρίς την προσθήκη κώδικα προσομοίωσης. Σε αυτή την εφαρμογή δεν δόθηκε έμφαση σε θέματα όπως η εισαγωγή αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος και η αποτίμηση της σχεδιασμένης λύσης. Έτσι, το παράδειγμα παρουσιάζεται στη συνέχεια σε μία ενότητα για κάθε ένα από τα τρία πρώτα βήματα της μεθοδολογίας.

6.1 Σχεδιασμός Συστήματος με Χαρακτηριστικά Απόδοσης

Ο σχεδιασμός του συστήματος γίνεται σε εργαλείο που υποστηρίζει τη UML και τις επεκτάσεις της (προφίλ για SysML και DEVS). Στη συγκεκριμένη περίπτωση χρησιμοποιήθηκε το MagicDraw. Για τη διαμόρφωση του μοντέλου, αρχικά δημιουργείται ένα BDD, το οποίο περιέχει τα στοιχεία του συστήματος, υποδεικνύοντας και τις μεταξύ τους σχέσεις. Τα στοιχεία του παραδείγματος περιγράφονται στο BDD του σχήματος 6.1.



Σχήμα 6.1: Το block definition diagram με τα στοιχεία του συστήματος μάχης.

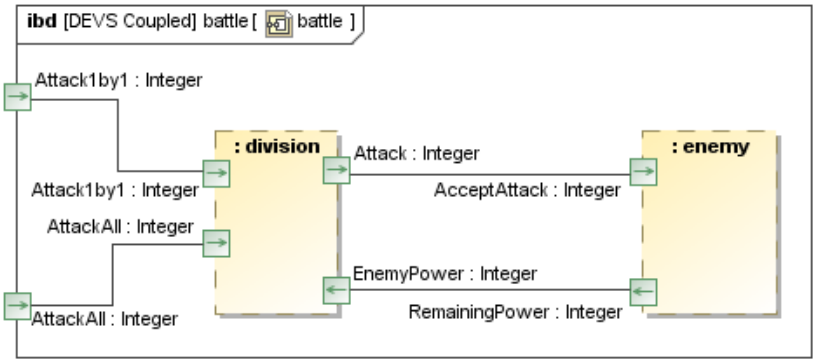
Το σύστημα που περιγράφει το σενάριο μάχης είναι η ιεραρχία block με αρχικό κόμβο το *battle*. Το block *battle* περιέχει την εχθρική μονάδα (*enemy*) και τη φιλική (*division*), η οποία περιλαμβάνει τα blocks *regiment*, *artillery* και *infantry*. Όλα τα blocks έχουν χαρακτηριστεί με

τα στερεότυπα του προφίλ και συγκεκριμένα, με το *DEVS_Coupled* αυτά που είναι ψηλά στην ιεραρχία και με το *DEVS_Atomic* τα blocks, που δεν περιέχουν άλλα στοιχεία *DEVS*.

Από την άλλη πλευρά το block *ef* (experimental framework) περιλαμβάνει (α) μία γεννήτρια γεγονότων (*generator*) για την ενεργοποίηση της μάχης, καθώς τα προαναφερθέντα στοιχεία είναι παθητικά και (β) έναν αποδέκτη της ένδειξης ολοκλήρωσης της μάχης (*transducer*).

Στην κορυφή της ιεραρχίας του μοντέλου βρίσκεται το block *efb* (experimental frame-work-battle), το οποίο περιέχει το experimental framework (*ef*) και το σύστημα μάχης (*battle*).

Ενώ τα περιεχόμενα του μοντέλου και η ιεραρχία των στοιχείων του περιγράφονται στο BDD, οι διασυνδέσεις των ports των περιεχόμενων blocks σε ένα *DEVS Coupled* block προσδιορίζεται με άνεση σε ένα IBD. Για παράδειγμα, οι ports του DEVS Coupled block *battle* (*Atack1by1* και *AttackAll*) συνδέονται με ports των περιεχόμενων blocks (*division* και *enemy*), όπως φαίνεται στο σχήμα 6.2. Στο IBD προσδιορίζονται και οι διασυνδέσεις μεταξύ των περιεχόμενων blocks.



Σχήμα 6.2: Το internal block diagram του συστήματος μάχης.

Με αυτό τον τρόπο καθορίζονται όλες οι συνδέσεις εντός των DEVS Coupled blocks, στα αντίστοιχα IBDs. Έτσι, στο σχήμα 6.3 ορίζονται οι εσωτερικές συνδέσεις του block *division*. Η κατεύθυνση των ports είναι σημαντική για τις συνδέσεις, όπως ορίζεται και στο προφίλ SysML για DEVS (βλ. πίνακα 5.3). Για διασυνδέσεις περιεχόμενων blocks, επιτρέπονται μόνο output-σε-input συνδέσεις. Αντίθετα, στην περίπτωση σύνδεσης port του εξωτερικού block με port κάποιου περιεχόμενου block, η κατεύθυνση πρέπει να είναι η ίδια, αφού ουσιαστικά γίνεται προώθηση των γεγονότων προς το εσωτερικό ή το εξωτερικό του εξωτερικού block.

Για τα DEVS Atomic blocks ορίζεται ένα σύνολο από χαρακτηριστικά συμπεριφοράς (καταστάσεις, εσωτερικές μεταβάσεις, προώθηση χρόνου, έξοδος και εξωτερικές μεταβάσεις). Εδώ εξετάζεται το DEVS Atomic block *regiment* για την επίδειξη των χαρακτηριστικών αυτών. Οι καταστάσεις ορίζονται ρητά σε ένα εσωτερικό BDD, που ονομάζεται *States*. Όπως φαίνεται και στο σχήμα 6.4, οι διακριτές καταστάσεις ορίζονται ως DEVS State Constraints, όπου μία παράμετρος (*s* περιορίζεται σε μία συγκεκριμένη τιμή.

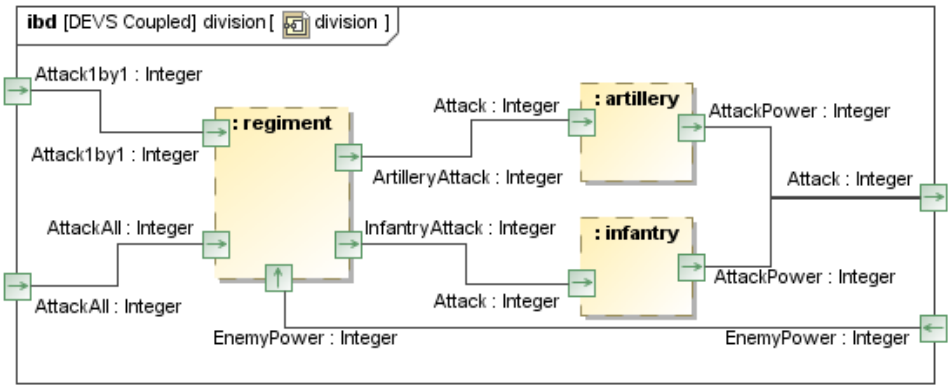
Στη συνέχεια, τα ανεξάρτητα state constraints του BDD *States* συσχετίζονται με τις μεταβλητές κατάστασης σε ένα PD. Στην περίπτωση του συντάγματος, χρησιμοποιείται η μεταβλητή κατάστασης *phase*, η οποία συσχετίζεται με τα state constraints στο PD του σχήματος 6.5. Η κατάσταση του συντάγματος καθορίζεται μόνο από τη μεταβλητή *phase*, οπότε η τελευταία

συσχετίζεται με την παράμετρο *s* σε όλα state constraints.

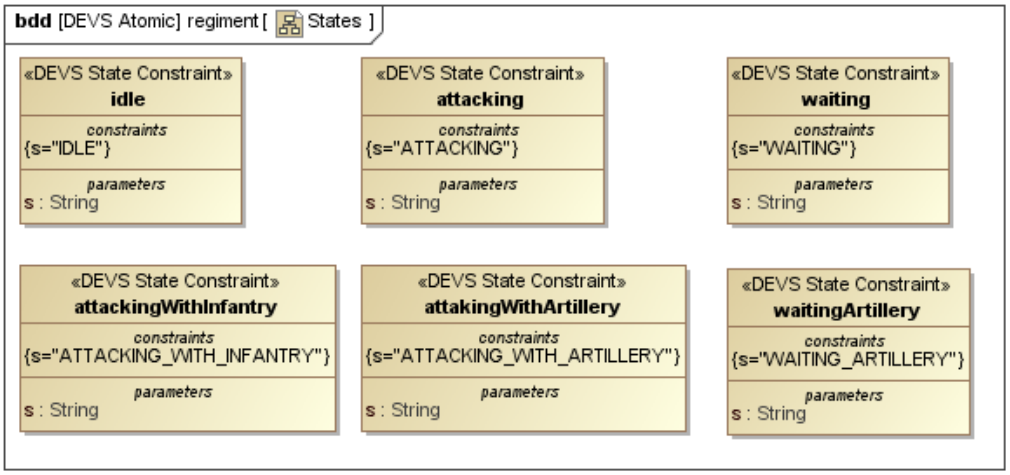
Οι πληροφορίες σχετικά με τις εσωτερικές μεταβάσεις κατάστασης, την παραγωγή εξόδων και της προώθησης χρόνου ενός block DEVS Atomic περιέχονται σε ένα SMD, που ονομάζεται *Internal Transition*. Το σχήμα 6.6 παρουσιάζει το συγκεκριμένο διάγραμμα για το block *regiment*. Σε αυτό υπάρχει ένα *DEVS State* για κάθε *DEVS State Constraint* στο BDD του σχήματος 6.4. Ένας αρχικός κόμβος (κύκλος, χρωματισμένος με μαύρο) χρησιμοποιείται για την ένδειξη της αρχικής κατάστασης (*idle*).

Όλες οι δυνατές εσωτερικές μεταβάσεις ορίζονται με κατευθυνόμενες συνδέσεις τύπου *DEVS State Transition*. Δεν είναι απαραίτητη η ύπαρξη συνδέσεων προς όλες τις καταστάσεις, καθώς κάποιες μπορεί να επέρχονται μόνο ύστερα από κάποιο εξωτερικό γεγονός, όπως π.χ. *attacking*, *attackingWithArtillery* και *attackingWithInfantry*. Παρομοίως, καταστάσεις οι οποίες εγκαταλείπονται μόνο σε περίπτωση παραλαβής κάποιου εξωτερικού γεγονότος, δεν έχουν εξερχόμενες συνδέσεις, όπως π.χ. *idle*, *waiting* και *waitingArtillery*. Το πολύ μία εξερχόμενη *DEVS State Transition* μπορεί να ορίζεται σε κάθε *DEVS State*, όπως ορίζεται στο φορμαλισμό DEVS και στο προφίλ DEVS για SysML.

Η προώθηση χρόνου προκύπτει από την time duration guard condition, που ορίζεται σε



Σχήμα 6.3: Το internal block diagram της φιλικής μονάδας.



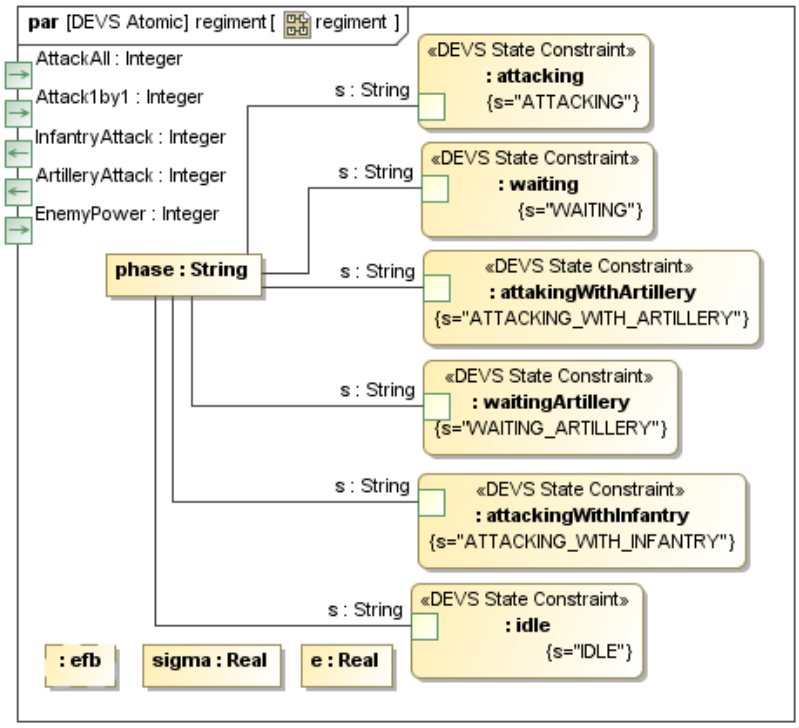
Σχήμα 6.4: Το constraint block definition diagram με τις καταστάσεις του συντάγματος.

κάθε *DEVS State Transition*. Για κάθε μετάβαση, αυτή η συνθήκη ορίζει τη χρονική διάρκεια παραμονής του block στην κατάσταση αφετηρίας της μετάβασης.

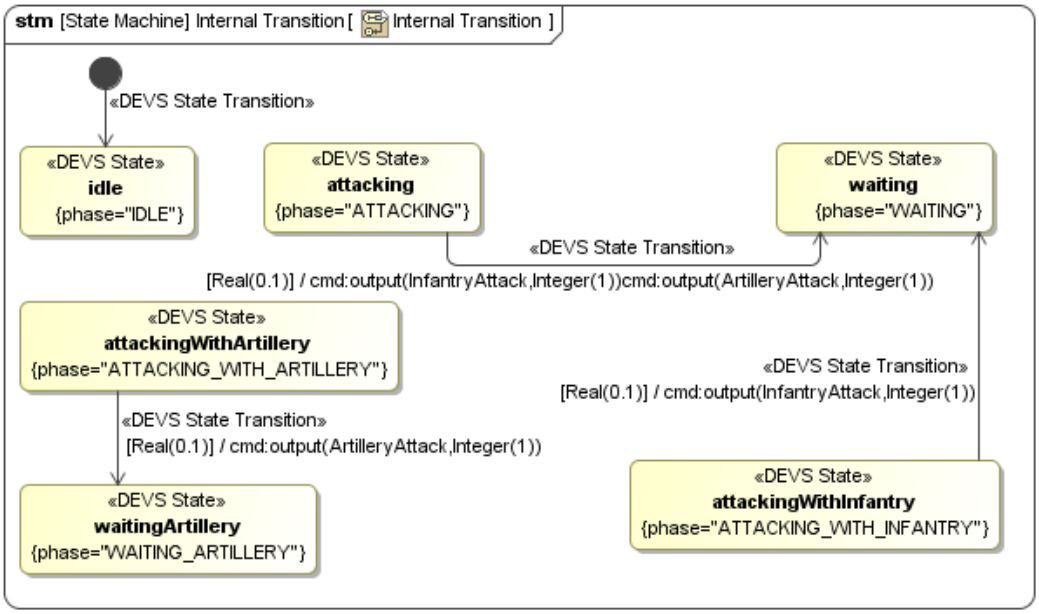
Η συνάρτηση εξόδου ορίζεται επίσης μέσω των *DEVS State Transitions*, αλλά με έναν μάλλον διαδικαστικό τρόπο. Στο σώμα συμπεριφοράς (behavior body) κάθε μετάβασης, δηλώνονται εντολές εξόδου, υποδεικνύοντας την πόρτα εξόδου και την τιμή εξόδου (απόλυτη ή σε έκφραση υπολογισμού) που πρέπει να αποσταλεί.

Τέλος, η συνάρτηση εξωτερικών μεταβάσεων ορίζεται μέσω ενός AD για κάθε block *DEVS Atomic*. Το σχήμα 6.7 παρουσιάζει αυτό το διάγραμμα για το DEVS Atomic *regiment*. Τα γεγονότα εισόδου από τα ports (π.χ. *AttackAll*, *Attack1by1* και *EnemyPower*) συνδυάζονται με ενδεχόμενες τρέχουσες (κατά την παραλαβή του γεγονότος) καταστάσεις (*DEVS State Check*) σε κόμβους τύπου *DEVS Input State Join* για τον ορισμό των μεταβάσεων στις νέες καταστάσεις (ενέργειες *DEVS State Modification*).

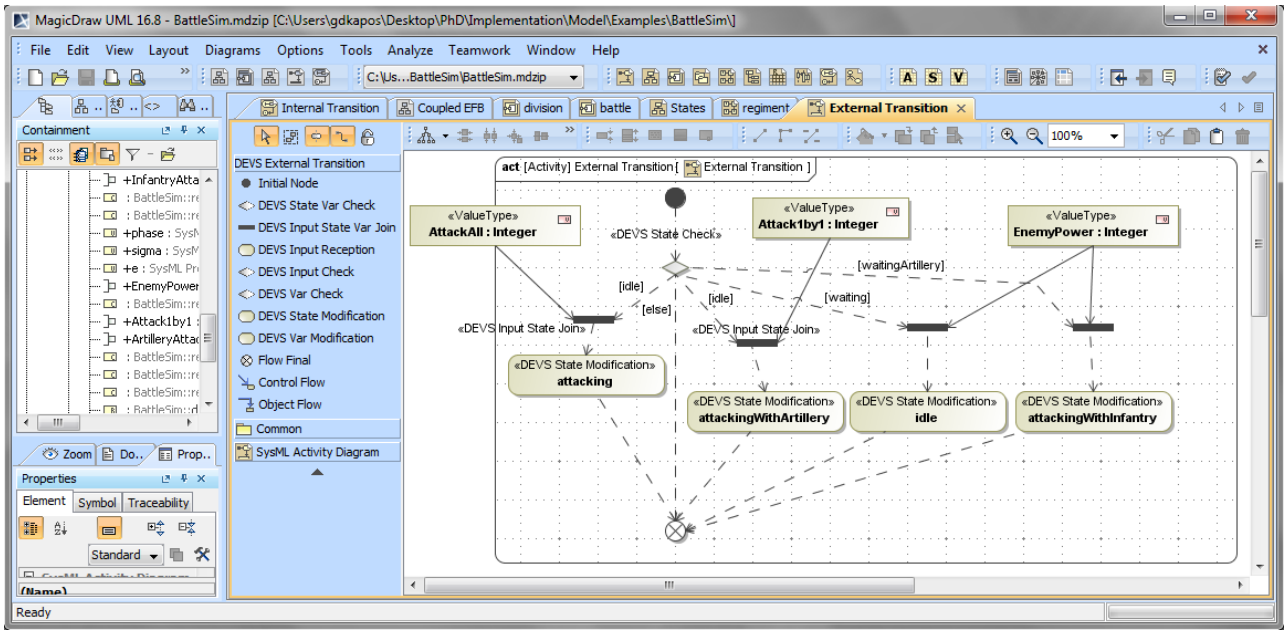
Το σχήμα 6.7 δείχνει επίσης τη διεπαφή χρήστη για τον ορισμό μοντέλων DEVS-SysML σε ένα εργαλείο μοντελοποίησης UML (*MagicDraw*). Η χρήση των απαιτούμενων στοιχείων μοντελοποίησης DEVS μπορούν να υποβοηθηθούν από τα αντίστοιχα buttons στη Γραφική Διεπαφή Χρήστη (Graphical User Interface) (GUI). Επιπρόσθετα, η εγκυρότητα του μοντέλου, ως προς τα χαρακτηριστικά DEVS μπορούν να ελέγχονται με την επαλήθευση των ορισμένων στο προφίλ περιορισμών OCL. Πρόσθετοι έλεγχοι και λειτουργικότητα μπορεί να επιτευχθεί και με την υλοποίηση plugins για το εργαλείο μοντελοποίησης, χρησιμοποιώντας το παρεχόμενο σχετικό API.



Σχήμα 6.5: Το parametric diagram με τις συσχετίσεις μεταξύ καταστάσεων και μεταβλητών κατάστασης του συντάγματος.



Σχήμα 6.6: Το state machine diagram με τις μεταβάσεις κατάστασης, τις εξόδους και την πρόοδο χρόνου του συντάγματος.

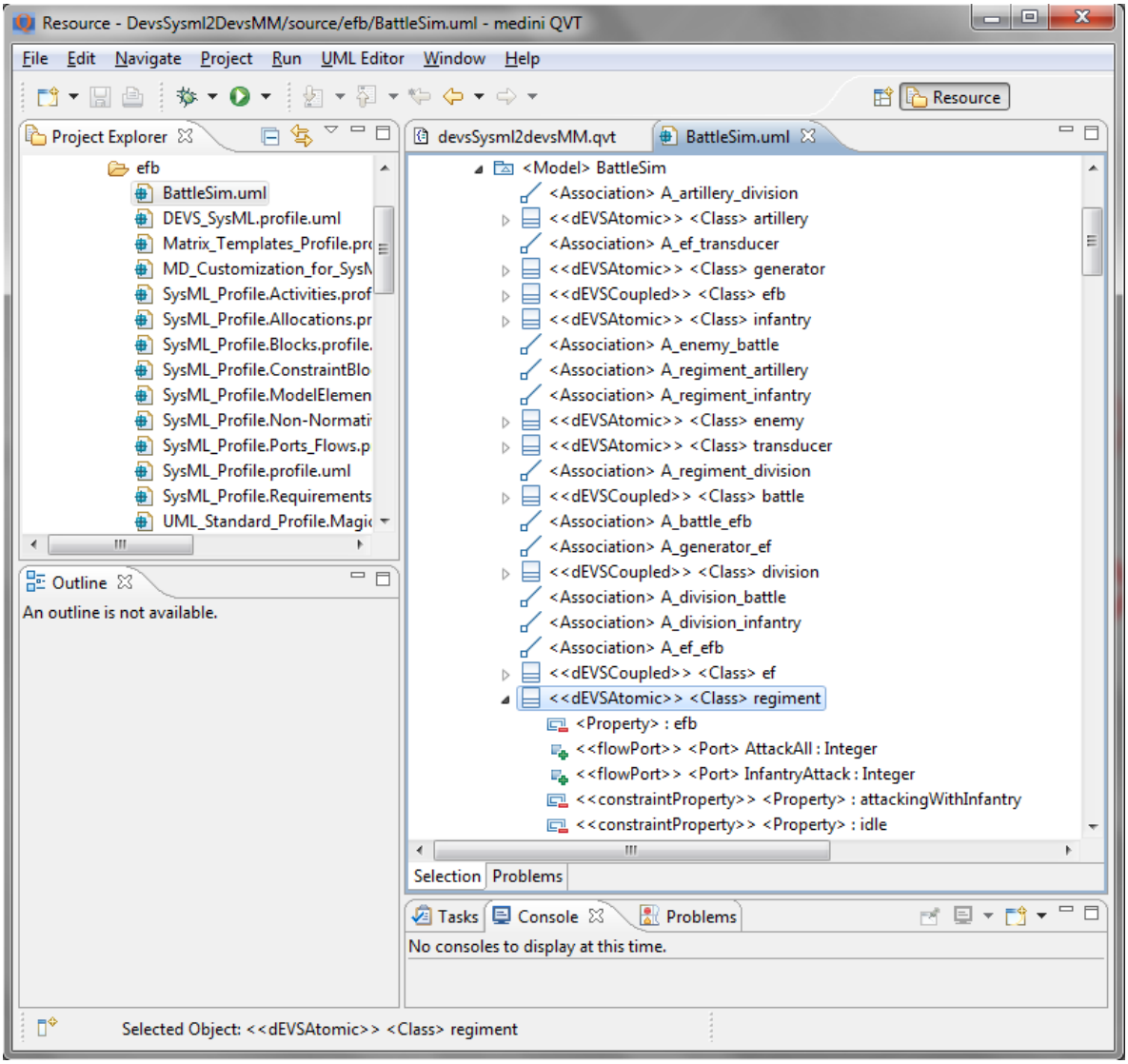


Σχήμα 6.7: Το activity diagram με τη συμπεριφορά του συντάγματος στην παραλαβή εξωτερικών γεγονότων.

6.2 Μετασχηματισμός Μοντέλου Συστήματος σε Μοντέλο DEVS

Το εμπλουτισμένο με στοιχεία συμπεριφοράς και απόδοσης μοντέλο συστήματος, που δημιουργήθηκε στο εργαλείο μοντελοποίησης UML, μπορεί να αξιοποιηθεί στο δεύτερο βήμα της μεθοδολογίας, το μετασχηματισμό. Το μοντέλο συστήματος μετασχηματίζεται σε ένα καθαρό μοντέλο DEVS που είναι έτοιμο για εκτέλεση, με την έννοια ότι θα μπορούσε να εκτελεστεί σε ένα συμβατό προσομοιωτή DEVS ή να προσαρμοστεί για να εκτελεστεί σε άλλους προσομοιωτές DEVS. Επίσης, αυτό το βήμα δεν απαιτεί ουσιαστική συνεισφορά από το σχεδιαστή, καθώς ο μετασχηματισμός έχει ήδη οριστεί και μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο DEVS-SysML.

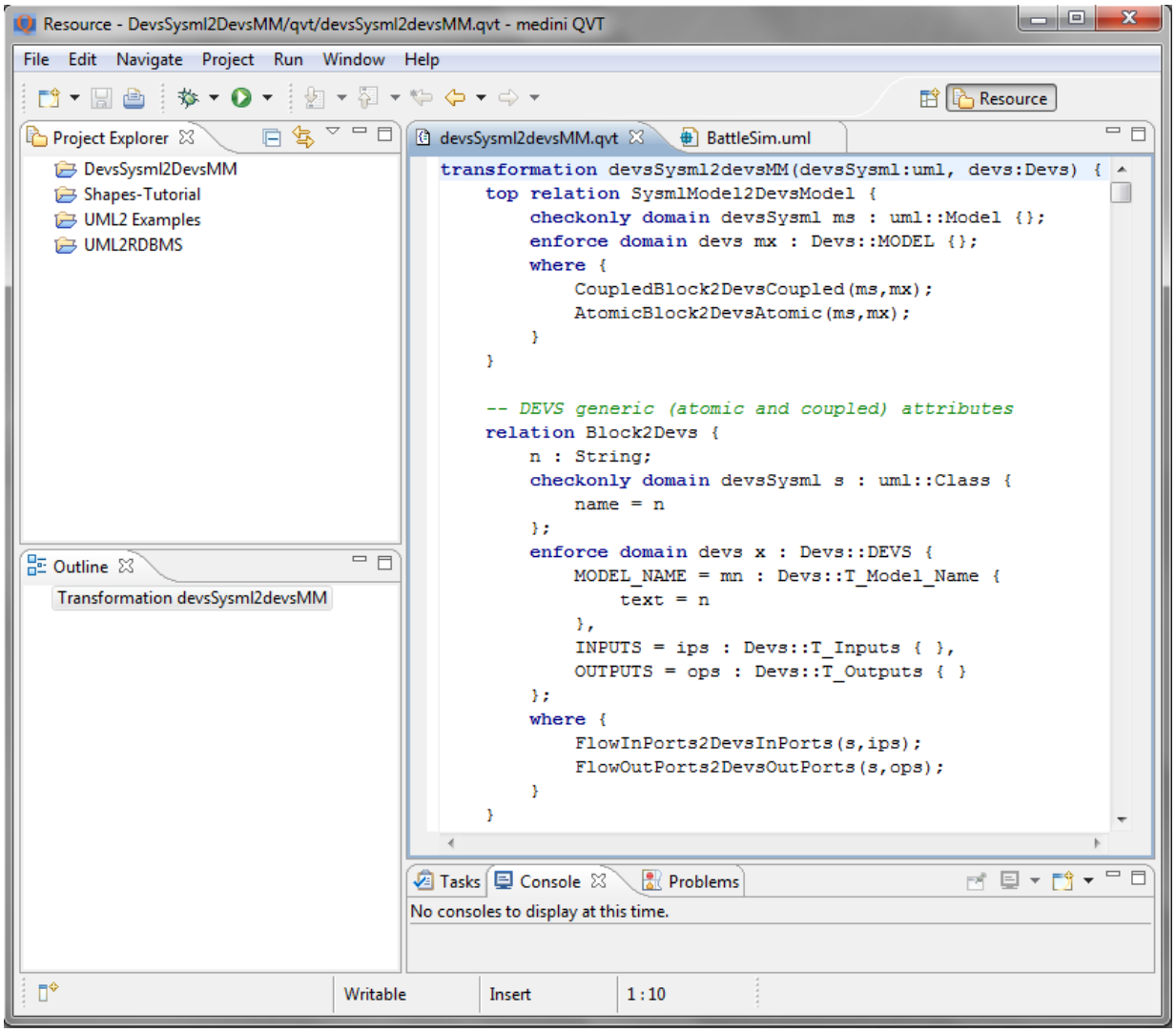
Για την εκτέλεση του μετασχηματισμού χρησιμοποιείται το εργαλείο Medini QVT (Medini) [62], που εκτελεί μετασχηματισμούς QVT. Το σχήμα 6.8 παρουσιάζει το μοντέλο συστήματος, όπως αυτό εμφανίζεται στο Medini, μετά από εξαγωγή από το εργαλείο μοντελοποίησης UML. Τα στερεότυπα του προφίλ DEVS και τα στοιχεία του μοντέλου διακρίνονται καθαρά.



Σχήμα 6.8: Το μοντέλο συστήματος που έχει εισαχθεί στο Medini QVT για μετασχηματισμό.

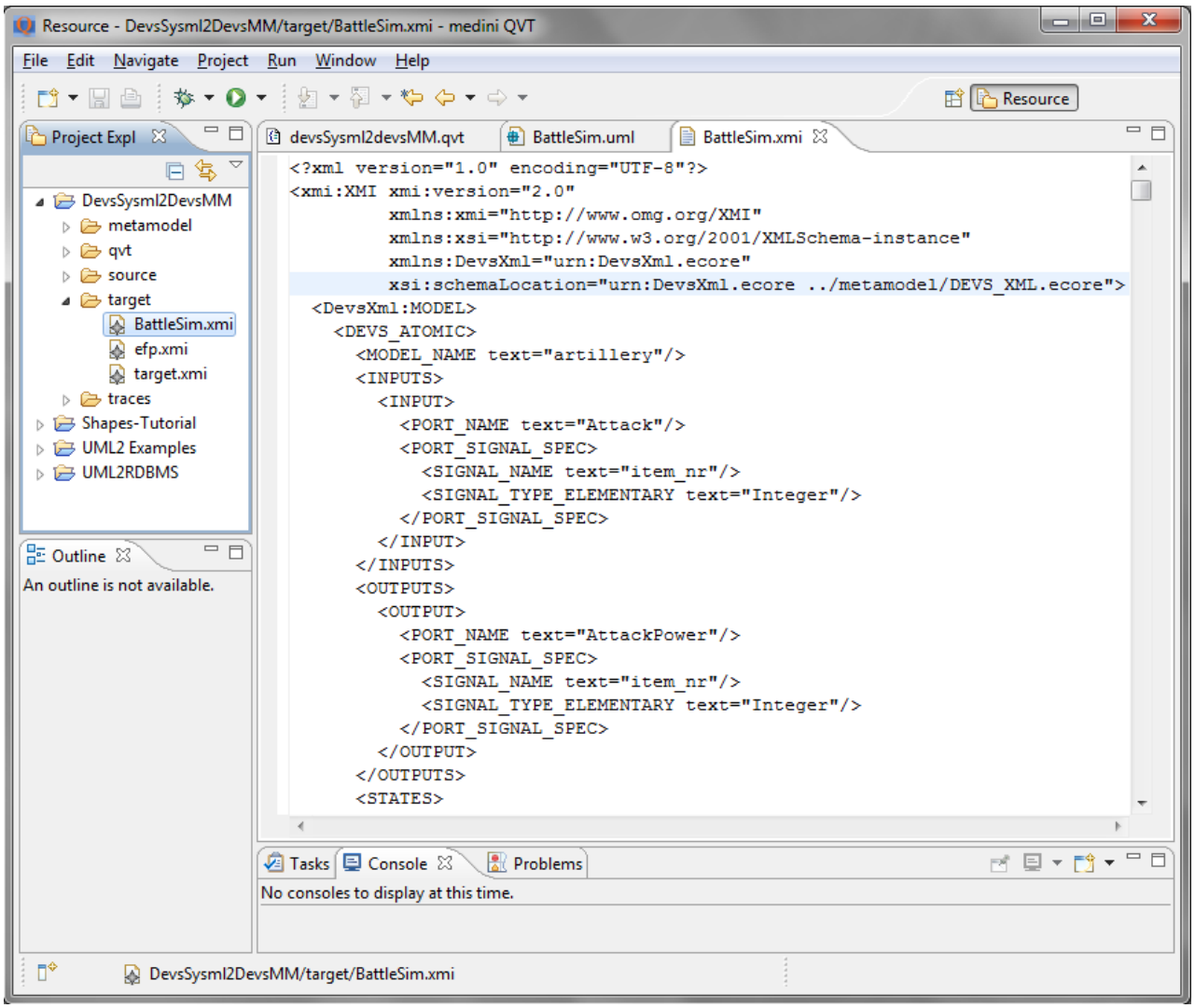
Ο μετασχηματισμός που απεικονίζεται στο σχήμα 5.4 έχει υλοποιηθεί ως ένα σύνολο από

σχέσεις QVT. Ένα μικρό μέρος των σχέσεων αυτών φαίνεται στο σχήμα 6.9.



Σχήμα 6.9: Οι σχέσεις QVT στο Medini QVT για το μετασχηματισμό του μοντέλου συστήματος σε μοντέλο προσομοίωσης.

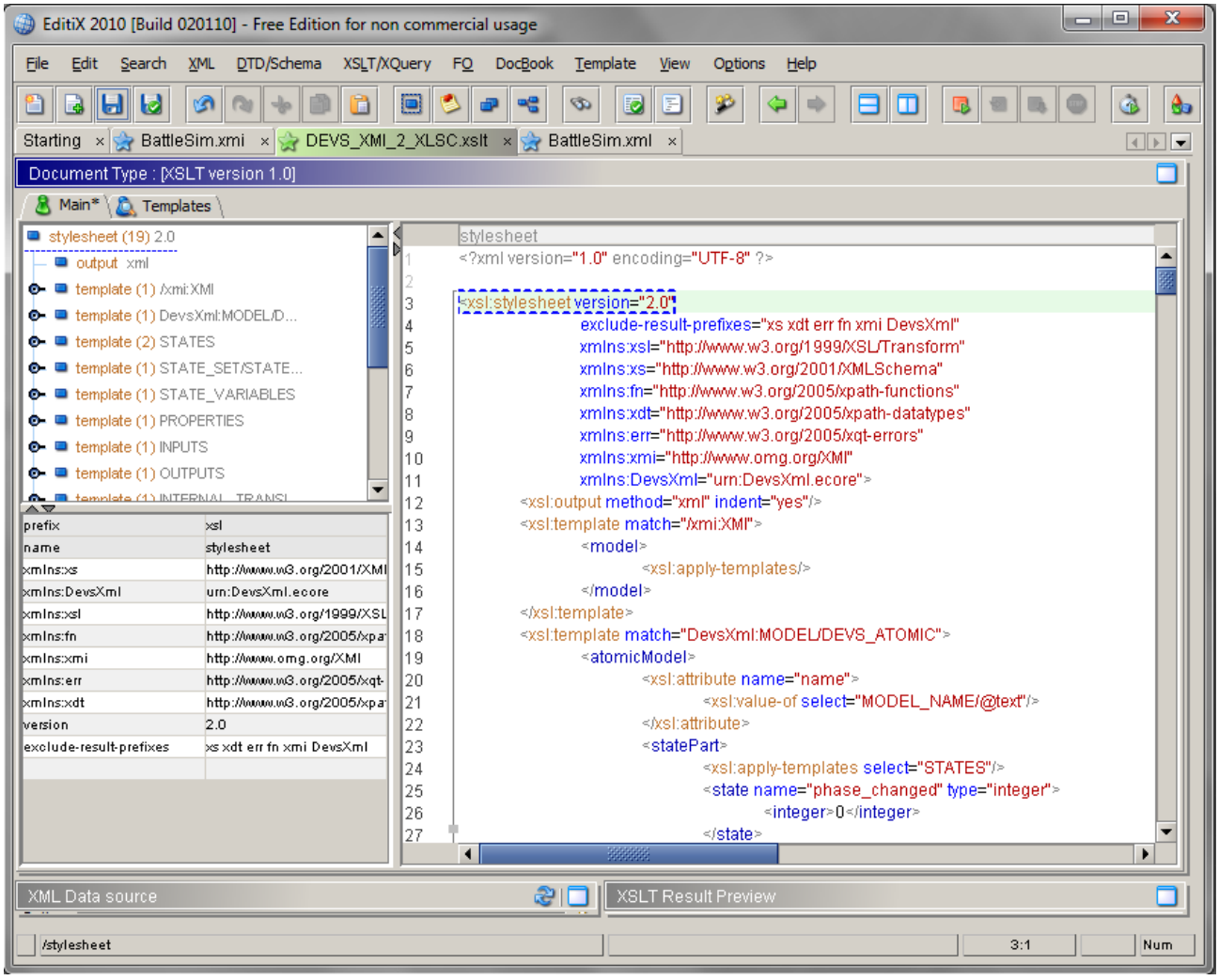
Με την εκτέλεση του μετασχηματισμού παράγεται το μοντέλο DEVS, όπως φαίνεται στο σχήμα 6.10. Το μοντέλο είναι σύμφωνο με το μετα-μοντέλο DEVS και μπορεί να αξιοποιηθεί περαιτέρω.



Σχήμα 6.10: Το μοντέλο προσομοίωσης DEVS που παρήχθη με το μετασχηματισμό στο εργαλείο Medini QVT.

6.3 Εκτέλεση Προσομοίωσης DEVS

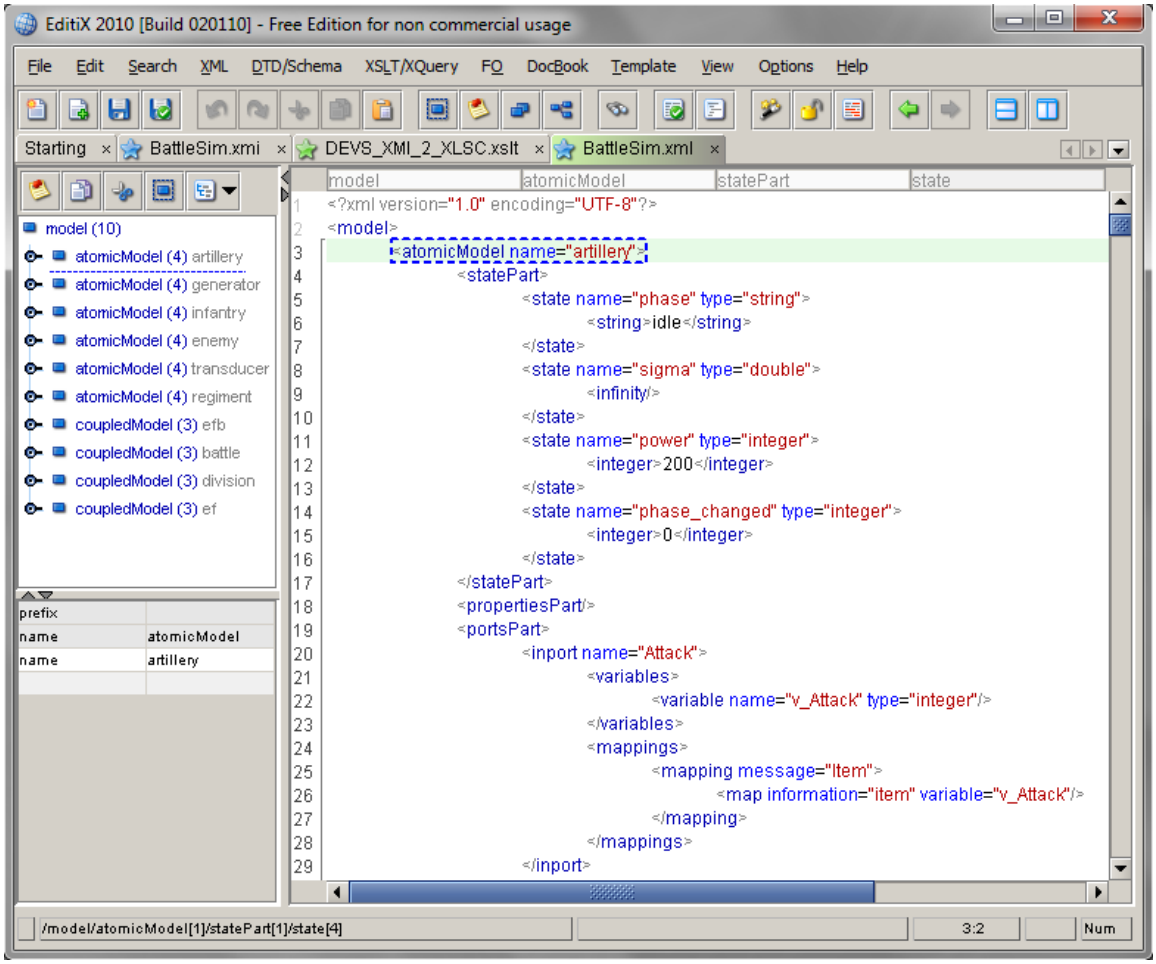
Το τρίτο βήμα της μεθοδολογίας περιλαμβάνει την εκτέλεση του μοντέλου προσομοίωσης. Σε αυτή την περίπτωση το περιβάλλον εκτέλεσης της προσομοίωσης είναι το DEVS XLSC. Η απαιτούμενη από το XLSC είσοδος είναι μία διαφορετική, χαμηλότερου επιπέδου XML αναπαράσταση των μοντέλων DEVS. Για τη μετατροπή της XMI αναπαράστασης των μοντέλων DEVS στην μορφή XLSC έχει οριστεί ένα μετασχηματισμός XSLT. Ένα τμήμα του μετασχηματισμού αυτού, όπως φαίνεται στην ελεύθερη έκδοση του εργαλείου EditiX [63], παρουσιάζεται στο σχήμα 6.11. Το EditiX είναι ένα εργαλείο διαχείρισης αρχείων XML, το οποίο εκτελεί μετασχηματισμούς XSLT. Και σε αυτό το βήμα της μεθοδολογίας δεν απαιτείται ουσιαστική παρέμβαση από το σχεδιαστή του μοντέλου συστήματος. Στη συγκεκριμένη υλοποίηση μάλιστα, ο μετασχηματισμός XSLT εκτελείται με τη χρήση ενός script, χωρίς την παρέμβαση του σχεδιαστή.



Σχήμα 6.11: Ο XSLT μετασχηματισμός για XMI μοντέλα DEVS σε XLSC στο εργαλείο EditiX.

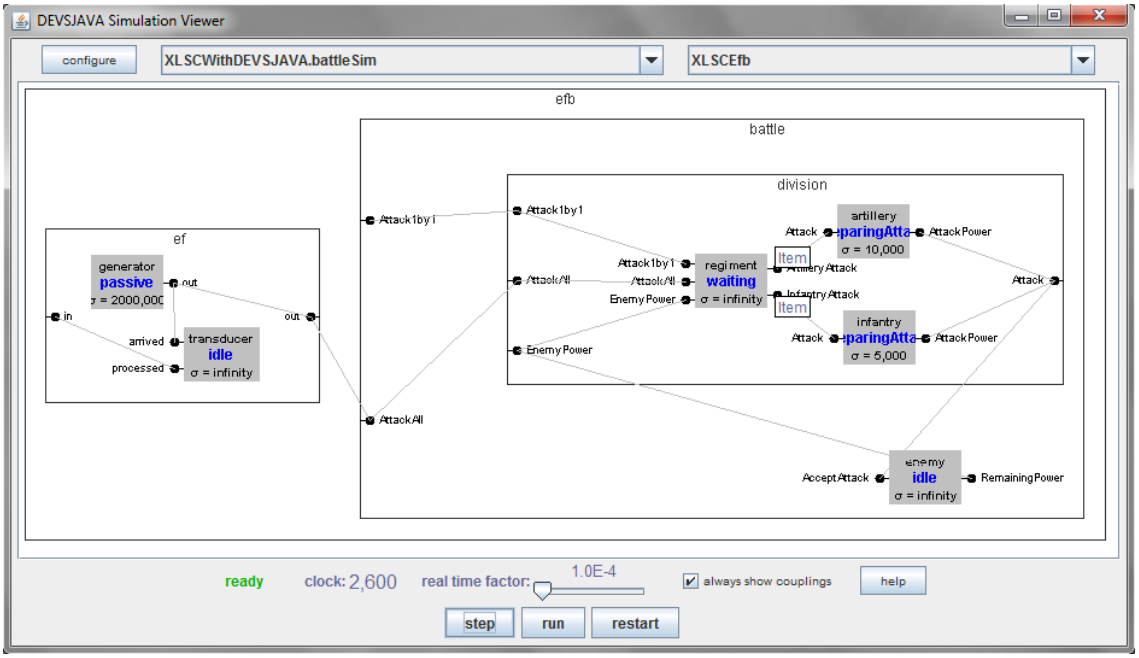
Τμήμα του αποτελέσματος του μετασχηματισμού από XMI αναπαράσταση μοντέλου DEVS (σχήμα 6.10) σε XLSC παρουσιάζεται στο σχήμα 6.12. Όπως φαίνεται έχει γίνει η απαιτούμενη προσαρμογή των στοιχείων XML, ώστε το αρχείο να μπορεί να εκτελεστεί στο περιβάλλον

XLSC/DEVSJava.



Σχήμα 6.12: Η παραγόμενη XLSC αναπαράσταση του μοντέλου DEVS στο εργαλείο EditiX.

Το σχήμα 6.13 παρουσιάζει μία εικόνα κατά την εκτέλεση του μοντέλου στο περιβάλλον XLSC/DEVSJava. Χρησιμοποιήθηκε το εργαλείο *SimView*, το οποίο παρέχει την βήμα-προς-βήμα γραφική αναπαράσταση της εκτέλεσης της προσομοίωσης. Στο συγκεκριμένο στιγμιότυπο, το συστατικό *regiment* έχει παραλάβει το γεγονός *AttackAll* και προωθεί ταυτοχρόνως γεγονότα *Attack* προς αμφότερες τις μονάδες *artillery* και *infantry*, οι οποίες μόλις εισήλθαν στην κατάσταση *preparingAttack*.



Σχήμα 6.13: Η εκτέλεση της προσομοίωσης στο γραφικό περιβάλλον εκτέλεσης DEVS-Java κώδικα SimView.

6.4 Σχολιασμός

Η ολοκληρωμένη αξιοποίηση των επιμέρους βημάτων της μεθοδολογίας στο απλό παράδειγμα ενός συστήματος μάχης, παρέχει μία ολοκληρωμένη και καθαρή εικόνα της εφαρμοσιμότητας της πρότασης. Αποκαλύπτει επίσης, πώς η έμφαση που δόθηκε σε κοινά αποδεκτά και προτυποποιημένα μοντέλα και μεθόδους προσδίδει στην προσέγγιση πλεονεκτήματα, όπως δια-λειτουργικότητα και υποστήριξη από πολλά εργαλεία.

Η ουσιαστική παρέμβαση του σχεδιαστή γίνεται μόνο στο μοντέλο συστήματος και μόνο μέσω λογισμικού γραφικής δημιουργίας μοντέλων συστημάτων σε προτυποποιημένες γλώσσες, όπως η SysML. Χωρίς να απαιτείται να συγγράψει ούτε μία γραμμή κώδικα προσομοίωσης, ο σχεδιαστής έχει στη διάθεσή του αποτελέσματα προσομοίωσης από προσομοιωτή DEVS, σύμφωνα με τα βήματα της μεθοδολογίας.

Κατά τον εμπλουτισμό του μοντέλου συστήματος, η πιθανότητα εισαγωγής λογικών και συντακτικών λαθών περιορίζεται, καθώς η διαδικασία γίνεται σε ένα πιο υψηλό επίπεδο και με οπτική απεικόνιση, σε σχέση με τη συγγραφή κώδικα προσομοίωσης. Επίσης, η συντακτική ορθότητα του εμπλουτισμένου μοντέλου διασφαλίζεται από τους περιορισμούς του SysML προφίλ για DEVS. Έτσι, μόνο έγκυρα εμπλουτισμένα μοντέλα αξιοποιούνται περαιτέρω.

Μία βασική γνώση του χρησιμοποιούμενου πλαισίου προσομοίωσης (DEVS) απαιτείται για τον εμπλουτισμό στοιχείων της συμπεριφοράς του μοντέλου συστήματος. Ωστόσο, καθώς η SysML και το DEVS προσεγγίζουν τα μοντέλα με αντίστοιχο τρόπο, περιγράφοντάς τα ως συστήματα συστημάτων, αυτό δεν θα πρέπει να είναι εμπόδιο. Επιπρόσθετα, υπάρχοντα διαγράμματα SMD και/ή AD θα μπορούσαν να αξιοποιηθούν, μειώνοντας τον αριθμό των απαιτούμενων τροποποιήσεων.

Παρόλα αυτά ο εμπλουτισμός του μοντέλου συστήματος είναι μία διόλου ευκαταφρόνητη εργασία, καθώς το SysML προφίλ για DEVS δημιουργήθηκε με γνώμονα την πιστή και ακριβή αναπαράσταση εννοιών του DEVS και όχι την απλούστευση της διαδικασίας εμπλουτισμού. Κάτι τέτοιο θα μπορούσε να επιτευχθεί π.χ. με περιορισμό της πληροφορίας που απαιτείται να δηλώνεται ρητά κατά τον εμπλουτισμό του μοντέλου. Για παράδειγμα, οι καταστάσεις ενός SMD κάποιου block θα μπορούσαν να μην χαρακτηρίζονται με συγκεκριμένο στερεότυπο (*DEVS State*), αλλά κατά το μετασχηματισμό του μοντέλου να γίνονται πρόσθετοι έλεγχοι οι οποίοι να καθορίζουν κατά πόσο κάθε state μπορεί να αξιοποιηθεί ως κατάσταση DEVS στο μοντέλο προσομοίωσης. Επιλογές σαν αυτή θα περιόριζαν σημαντικά το φόρτο του σχεδιαστή κατά τον εμπλουτισμό του μοντέλου, αλλά θα εισήγαγαν ασάφεια γύρω από το ποιο τμήμα του μοντέλου θα προσομοιωθεί τελικά.

Εναλλακτικά, ο σχεδιαστής θα μπορούσε να υποβοηθηθεί με την ανάπτυξη plugins για τα εργαλεία μοντελοποίησης SysML, τα οποία να εξετάζουν ένα μοντέλο ή τμήματά του και να εφαρμόζουν στερεότυπα του προφίλ προσομοίωσης σε όσα στοιχεία του μοντέλου αυτό είναι εφικτό.

Σε κάθε περίπτωση, για τη διαμόρφωση μίας ακριβούς ανάλυσης κόστους χρήσης της προσέγγισης, διάφοροι παράγοντες θα πρέπει να συνυπολογιστούν. Πρώτον, υπάρχει εξάρτηση

από το domain της εφαρμογής και την επανάληψη των στοιχείων του μοντέλου. Εμφάνιση πολλών επαναλήψεων μπορεί να οδηγήσει σε επαναχρησιμοποίηση των εμπλουτισμένων στοιχείων του μοντέλου, περιορίζοντας το συνολικό κόστος εμπλουτισμού. Δεύτερον, ο εμπλουτισμός του μοντέλου θα μπορούσε να υποβοηθηθεί από ειδικού σκοπού plugins, που μπορεί να αναπτυχθούν για συγκεκριμένα εργαλεία μοντελοποίησης UML/SysML. Τέτοια εργαλεία θα μπορούσαν μέχρι και να εξαλείψουν τις απαιτούμενες από το σχεδιαστή ενέργειες εμπλουτισμού, που μπορεί να φαίνονται περιττές. Τέλος, η αξιοποίηση διαθέσιμων συστατικών λογισμικού προσομοίωσης από αντίστοιχες βιβλιοθήκες θα μπορούσε να απλοποιήσει σημαντικά αυτή τη διαδικασία.

Κεφαλαιο 7

Εφαρμογη Προτασης με Χρηση Διαθεσιμων Συστατικων Λογισμικου Προσομοιωσης - Η Σχεδιαση ΠΣ ως Συστημα

Σε αυτό το κεφάλαιο παρουσιάζεται ένα παράδειγμα εφαρμογής της πρότασης με μία εναλλακτική προσέγγιση, σε σχέση με την απλή αξιοποίηση του πλαισίου DEVSys (κεφάλαιο 5), όπως το παράδειγμα που παρουσιάστηκε στο κεφάλαιο 6. Σύμφωνα με το γενικό πλαίσιο, που ορίζεται στο κεφάλαιο 4, είναι απαραίτητη η χρήση ενός προφίλ SysML. Σε αυτή την περίπτωση επελέγη ένα διαθέσιμο προφίλ για την περιγραφή συστημάτων ενός συγκεκριμένου πεδίου εφαρμογών, των εταιρικών πληροφοριακών συστημάτων (EIS). Επομένως, δεν εξετάστηκε η επιλογή ενός προφίλ για κάποια μεθοδολογία προσομοίωσης, αλλά αντίθετα, στο υπάρχον προφίλ για EIS αναζητήθηκαν οι δομικές ομοιότητες με το μετα-μοντέλο DEVS και τα χαρακτηριστικά εκείνα, που μπορούν να διακρίνουν τα διαφορετικού τύπου στοιχεία του μοντέλου και να καθορίσουν τις παραμέτρους προσομοίωσης.

Από το υλοποιημένο πλαίσιο DEVSys αξιοποιήθηκε το μετα-μοντέλο DEVS στην τελική του, επεκτεταμένη μορφή, όπου μπορούν να αξιοποιηθούν διαθέσιμα συστατικών λογισμικού σε βιβλιοθήκη προσομοίωσης και το μετα-μοντέλο αποτελεσμάτων. Συγκεκριμένα, τα σχετικά συστατικά λογισμικού προσομοίωσης δεν ήταν διαθέσιμα και υλοποιήθηκαν, σε κάθε περίπτωση όμως, εκτός των πλαισίων της διαδικασίας της προτεινόμενης μεθοδολογίας.

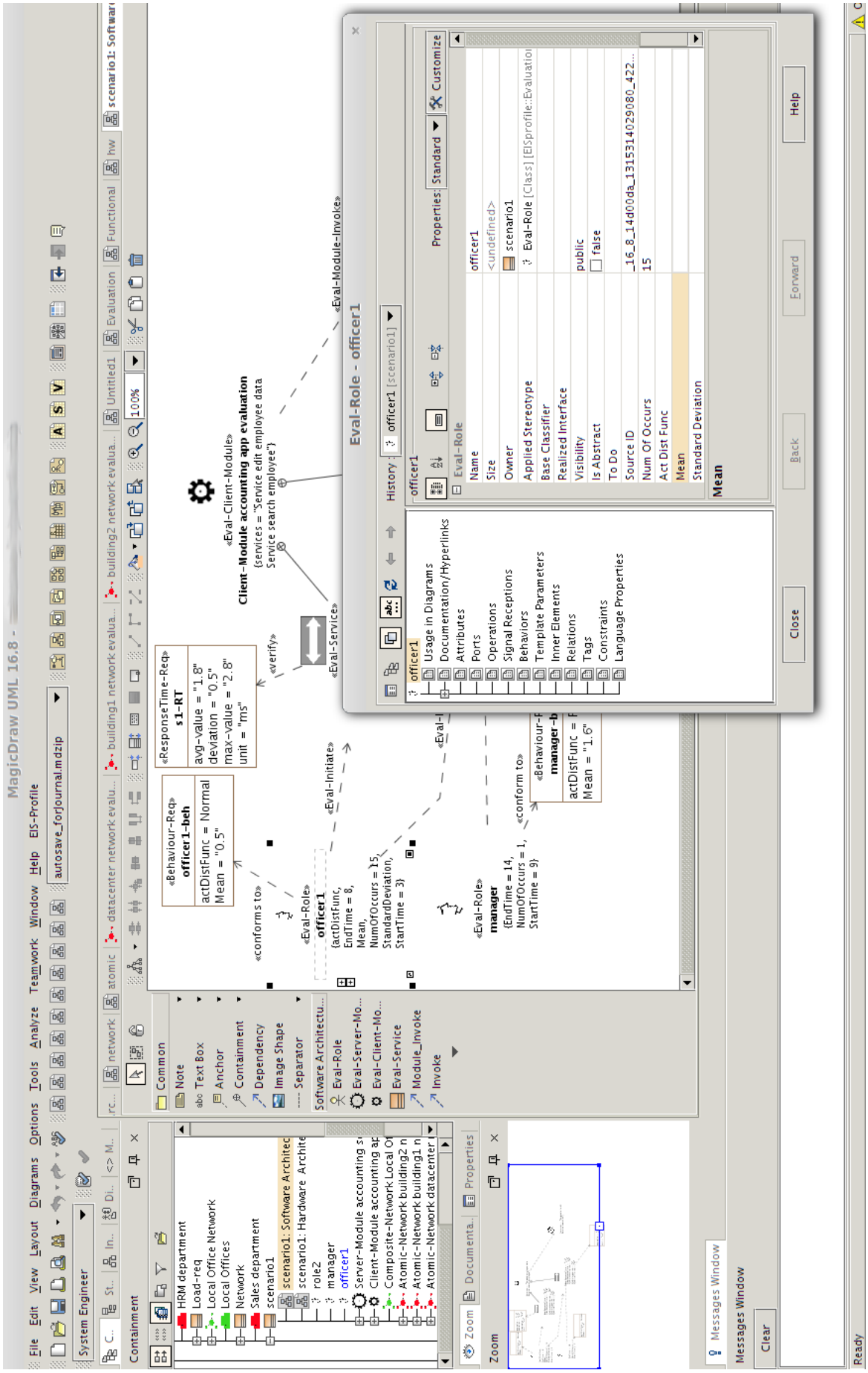
Για τη συγκεκριμένη εφαρμογή, εκτός από το προφίλ EIS, αξιοποιήθηκαν και άλλα στοιχεία της ερευνητικής προσπάθειας [64]. Σύμφωνα με αυτή, ο σχεδιαστής συστημάτων, όταν σχεδιάζει την αρχιτεκτονική ενός EIS, περιγράφει την αρχιτεκτονική τόσο του λογισμικού, όσο και του υλικού, με όρους των Functional, Topology και Network Infrastructure views. Μετά τον ορισμό της αρχιτεκτονικής EIS, η αποτίμηση γίνεται στο Evaluation view [65, 66]. Σε αυτό το view ορίζονται σενάρια Evaluation, ως στιγμιότυπα της ορισμένης αρχιτεκτονικής λογισμικού και υλικού. Στο Evaluation view είναι εφικτή η αυτοματοποιημένη επαλήθευση μη λειτουργικών απαιτήσεων, αρκεί να είναι διαθέσιμα αποτελέσματα προσομοίωσης. Αυτό επιτυγχάνεται με την εφαρμογή της προτεινόμενης μεθοδολογίας, η οποία παράγει εκτελέσιμα μοντέλα προσομοίωσης, τα εκτελεί, συγκεντρώνει τα αποτελέσματα προσομοίωσης και τα εισάγει στο μοντέλο συστήματος.

Σε αυτό το κεφάλαιο παρουσιάζεται η αποτίμηση ενός μοντέλου EIS για μία απλή εφαρμογή μητρώου, όπου οι εφαρμογές πελάτες επικοινωνούν με υπηρεσίες λογισμικού που εκτελούνται σε κατανεμημένους εξυπηρετητές βάσεων δεδομένων. Οι χρήστες της εφαρμογής χωρίζονται σε δύο κατηγορίες-ρόλους: *manager* και *staff*. Κάθε ρόλος μπορεί να χρησιμοποιήσει διαφορετικές υπηρεσίες, με καθορισμένες πιθανότητες.

7.1 Σχεδιασμός Συστήματος με Χαρακτηριστικά Απόδοσης

Το μοντέλο συστήματος EIS είχε οριστεί στο εργαλείο μοντελοποίησης MagicDraw, με τη χρήση του διαθέσιμου προφίλ SysML για EIS. Το σχήμα 7.1 παρουσιάζει μία εικόνα από το Software Evaluation view. Η συμμόρφωση κάποιου ρόλου με ένα *behavior requirement* καθορίζει κάποια χαρακτηριστικά της συμπεριφοράς του.

Ο σχεδιαστής συστημάτων δημιουργεί ένα σενάριο αποτίμησης (evaluation scenario) και, με την υλοποιημένη υποδομή για το σχεδιασμό EIS, παράγονται αυτόματα τα διαγράμματα software και hardware evaluation. Στα διαγράμματα αυτά αναπαρίσταται η τρέχουσα κατάσταση του μοντέλου συστήματος, ως στιγμιότυπο, όπως προκύπτει από όλες τις views, που έχουν χρησιμοποιηθεί για τη σχεδίασή του. Στη συνέχεια, ο σχεδιαστής εμπλουτίζει το σενάριο με πληροφορίες που αφορούν την προσομοίωση, όπως αριθμός επαναλήψεων, συνθήκες τερματισμού, επιτρέποντας τη δημιουργία ενός πλήρως εκτελέσιμου μοντέλου προσομοίωσης.



Σχήμα 7.1: EIS Evaluation View: Software architecture diagram.

7.2 Μετασχηματισμός Μοντέλου Συστήματος σε Μοντέλο DEVS

Για την εκτέλεση προσομοίωσης, στα πλαίσια της προτεινόμενης προσέγγισης, απαιτείται η παραγωγή ενός έγκυρου μοντέλου προσομοίωσης, σύμφωνα με το μετα-μοντέλο DEVS, που αναφέρεται στο 5.3. Αυτό επιτυγχάνεται με το μετασχηματισμό των μοντέλων EIS στα αντίστοιχα μοντέλα DEVS.

Προς αυτή την κατεύθυνση, μελετήθηκε η δομή των μοντέλων EIS και οι διασυνδέσεις μεταξύ των στοιχείων τους. Οι κύριες οντότητες, που επηρεάζουν την απόδοση του συνολικού συστήματος, εντοπίζονται στα Hardware και Software Evaluation Diagrams (περιλαμβάνονται στην Evaluation view). Παρά το γεγονός ότι για την ανάλυση και το σχεδιασμό συστημάτων EIS, είναι προτιμότερος ο ορισμός διαφορετικών χαρακτηριστικών του συστήματος σε διαφορετικά διαγράμματα και όψεις, το μοντέλο προσομοίωσης πρέπει να αποτυπώνει τη δομή και τα χαρακτηριστικά ενός συγκεκριμένου, συμπαγούς συστήματος, κάθε φορά. Επιπρόσθετα, η αποτελεσματικότητα της προσομοίωσης εξαρτάται σε μεγάλο βαθμό από την απλότητα του μοντέλου προσομοίωσης.

Η δομή των μοντέλων, τα χαρακτηριστικά των οντοτήτων και οι συσχετίσεις τους είναι διαθέσιμα για αξιοποίηση στο μετασχηματισμό, ανεξάρτητα από τα διαγράμματα στα οποία μπορεί να εμφανίζονται για λόγους οργάνωσης και παρουσίασης. Σε αυτά τα πλαίσια, ορίστηκε ένα σύνολο από συστατικά λογισμικού προσομοίωσης, σε αντιστοιχία με τα στοιχεία software και hardware του EIS, όπως θα μπορούσαν να συνδυαστούν σε σενάρια αποτίμησης.

Αυτό αποτυπώνεται σε γενικές γραμμές στο σχήμα 7.2, όπου τα στοιχεία EIS εμφανίζονται στο αριστερό τμήμα και τα αντίστοιχα στοιχεία DEVS στο δεξί. Το σχήμα παρουσιάζει την υψηλού επιπέδου εννοιολογική αντιστοίχιση ανάμεσα στα στοιχεία EIS και DEVS. Είναι εμφανές ότι τα στοιχεία προσομοίωσης προκύπτουν συνήθως από το συνδυασμό περισσοτέρων του ενός στοιχείων EIS, με την εξαίρεση του *DEVS_Simulation_Controller*. Το τελευταίο ορίζεται ανεξάρτητα από το σενάριο, ώστε να διαχωριστούν τα δομικά στοιχεία του μοντέλου προσομοίωσης από τις παραμέτρους εκτέλεσης της προσομοίωσης, όπως η διάρκεια της προσομοίωσης ή ο αριθμός επαναλήψεων.

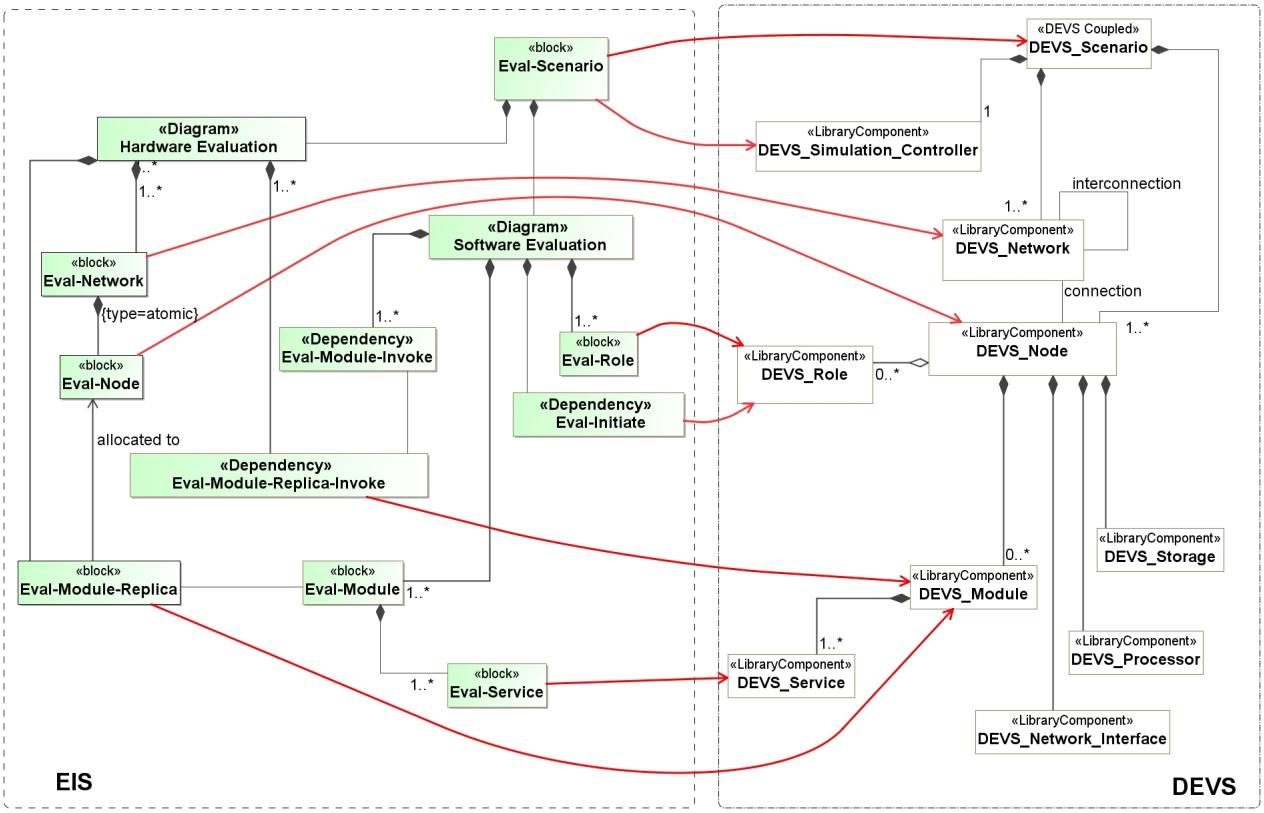
Το μέρος που αφορά το EIS στο σχήμα 7.2 περιλαμβάνει τις κύριες οντότητες του Hardware Evaluation Diagram (Eval-Network, Eval-Node, Eval-Module-Replica-Invoke), τις κύριες οντότητες του Software Evaluation Diagram (Eval-Role, Eval-Module, Eval-Service), τους συσχετισμούς και τις εξαρτήσεις τους. Επίσης, το block *Eval-Scenario* παρέχει πληροφορίες αναφορικά με τις συνθήκες αποτίμησης και χρησιμοποιούνται για την παραγωγή των αντίστοιχων, υψηλού επιπέδου στοιχείων *DEVS_ Scenario* και *DEVS_Simulation_Controller*.

Οι υπόλοιπες οντότητες EIS αξιοποιούνται για την παραγωγή των αντίστοιχων στοιχείων DEVS (*DEVS_Network*, *DEVS_Node*, κλπ.), ενώ οι συσχετίσεις και οι εξαρτήσεις υποδεικνύουν κυρίως τις διασυνδέσεις των στοιχείων του μοντέλου DEVS. Στο EIS, τα στοιχεία οργανώνονται σε διαγράμματα, τα οποία συνδυάζονται σε ένα σενάριο. Αντίθετα, τα στοιχεία DEVS σε μια πιο αυστηρή ιεραρχία. Το *DEVS_Scenario* είναι η ρίζα του ιεραρχικού μοντέλου, που περιέχει το *DEVS_Simulation_Controller*, ένα σύνολο από διασυνδεδεμένα στοιχεία *DEVS_Network* και ένα

σύνολο από στοιχεία *DEVS_Nodes*. Κάθε *DEVS_Node* αποτελείται από ένα *DEVS_Processor*, ένα *DEVS_ Storage*, ένα *DEVS_Network_Interface* Και ένα σύνολο από *DEVS_Modules*, που περιέχουν σύνολα από *DEVS_Services*. Επιπρόσθετα, προσδιορίζονται οι *DEVS_Roles* που χρησιμοποιούν κάθε node, όπως επίσης και το *DEVS_Network*, στο οποίο ανήκει κάθε κόμβος.

Σε ένα χαμηλότερο επίπεδο αυτού του σχήματος μετασχηματισμού, οι οντότητες EIS αναλύθηκαν περαιτέρω, ώστε να αναγνωριστούν τα βασικά χαρακτηριστικά τους, που καθορίζουν την απόδοση του συστήματος. Αντίστοιχα χαρακτηριστικά ορίστηκαν στα αντίστοιχα στοιχεία προσομοίωσης. Ο μετασχηματισμός επιτυγχάνει την κατάλληλη αρχικοποίησή τους, βασισμένη στις τιμές των αντίστοιχων χαρακτηριστικών των στοιχείων EIS. Οι πιθανές διασυνδέσεις των οντοτήτων εξετάστηκαν επίσης. Πρόσθετες πληροφορίες, που προκύπτουν από το συνδυασμό με άλλες οντότητες EIS, όπως οι εξαρτήσεις *Eval-Initiate*, υποδεικνύουν ποιες υπηρεσίες εκκινούνται από κάθε ρόλο. Στην πραγματικότητα, στο μετασχηματισμό αξιοποιείται ένα μεγάλο πλήθος χαρακτηριστικών και συσχετήσεων του μοντέλου EIS, που δεν είναι δυνατό να απεικονιστούν στην υψηλού επιπέδου αναπαράσταση του σχήματος 7.2.

Το γεγονός ότι και τα δύο μετα-μοντέλα (SysML και DEVS) είναι βασισμένα στο MOF, δίνει τη δυνατότητα χρήσης πρότυπων γλωσσών μετασχηματισμού, όπως η QVT. Ως εκ τούτου, οι κατάλληλες σχέσεις QVT ορίστηκαν για τη δημιουργία των κατάλληλων στοιχείων DEVS από τα αντίστοιχα στοιχεία EIS, καθώς και των διασυνδέσεών τους. Η πρώτη σχέση του



Σχήμα 7.2: Η αντιστοίχιση των στοιχείων του προφίλ για Enterprise Information Systems σε συστατικά λογισμικού προσομοίωσης DEVS από βιβλιοθήκη.

μετασχηματισμού QVT από EIS σε DEVS παρουσιάζεται στο listing 7.1.

Listing 7.1: Τμήμα του μετασχηματισμού QVT μοντέλων EIS σε μοντέλα DEVS

```
transformation eis2devsMM(eis:uml, devs:Devs) {
top relation eisScenario2DevsModel {
 scenarioName: String;
 checkonly domain eis scenario : uml::Class { name = scenarioName };
 enforce domain devs model : Devs::MODEL {
  DEVS_COUPLED = devsCoupled : Devs::DEVS_COUPLED {
   MODEL_NAME = modelName : Devs::T_Model_Name { text = scenarioName },
   COMPONENT_REFERENCE_LIST =
    componentReferenceList : Devs::T_Component_Reference_List { },
   INTERNAL_COUPLING = internalCoupling : Devs::T_Internal_Coupling { } } };
 when { scenario.getAppliedStereotype('EvaluationView::Evaluation Scenario')
    ->notEmpty(); }
 where {
  eisSimulationAttributes2ComponentReference(scenario,
     componentReferenceList,
    internalCoupling);
  eisEvalNetwork2ComponentReference(scenario,componentReferenceList,
     internalCoupling);
  eisIncludes2InternalCoupling(scenario.getModel(),internalCoupling);
  eisIncludesRev2InternalCoupling(scenario.getModel(),internalCoupling);
  eisEvalPTPConnection2InternalCoupling(scenario.getModel(),internalCoupling)
     ; } }
 ...
}
```

Τα εκτελέσιμα μοντέλα προσομοίωσης περιέχουν όλη την απαιτούμενη πληροφορία, καταδεικνύοντας με αυτό τον τρόπο τη σημασία της αυτοματοποιημένης παραγωγής τους από τα μοντέλα συστήματος, χωρίς ανθρώπινη παρέμβαση. Ένα τμήμα παραχθέντος μοντέλου προσομοίωσης DEVS παρουσιάζεται στο listing 7.2. Σε αυτό, ένα DEVS Coupled scenario περιέχει ένα *SimController* DEVS και ένα *Network* DEVS συστατικό λογισμικού. Οι παράμετροι αρχικοποίησης, όπως προκύπτουν από το μοντέλο EIS SysML, περιέχονται στο μοντέλο προσομοίωσης.

Listing 7.2: Τμήμα του μοντέλου προσομοίωσης DEVS για το μοντέλο EIS

```
<?xml version="1.0" encoding="UTF-8"?>
<Devs:MODEL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:Devs="urn:DEVS_MM.ecore"
xsi:schemaLocation="urn:DEVS_MM.ecore platform:/resource/Eis2DevsMM/metamodel/
   DEVS_MM.ecore">
  <DEVS_COUPLED>
    <MODEL_NAME text="aScenario"/>
    <COMPONENT_REFERENCE_LIST>
      <COMPONENT_REFERENCE xsi:type="Devs:T_Component_Reference" text="
```

```
        SimulationController">
      <LIBRARY_COMPONENT class="SimController" package="eis.library">
        <INIT_PARAMS>
          <INIT_PARAM xsi:type="Devs:T_Value_Init_Param" name="
              simulationTime">
            <VALUE type="Real" value="3600"/>
          </INIT_PARAM>
          <INIT_PARAM xsi:type="Devs:T_Value_Init_Param" name="
              simulationRuns">
            <VALUE type="Integer" value="1"/>
          </INIT_PARAM>
        </INIT_PARAMS>
      </LIBRARY_COMPONENT>
    </COMPONENT_REFERENCE>
    <COMPONENT_REFERENCE xsi:type="Devs:T_Component_Reference"
    text="Composite-Network␣regional␣office␣1␣net␣evaluation">
      <LIBRARY_COMPONENT class="Network" package="eis">
        <INIT_PARAMS>
          <INIT_PARAM xsi:type="Devs:T_Value_Init_Param" name="throughput">
            <VALUE type="Real" value="100"/>
          </INIT_PARAM>
          <INIT_PARAM xsi:type="Devs:T_Array_Init_Param" name="nodes">
            <INIT_PARAMS/>
          </INIT_PARAM>
          <INIT_PARAM xsi:type="Devs:T_Array_Init_Param" name="networks">
            <INIT_PARAMS>
              <INIT_PARAM xsi:type="Devs:T_Value_Init_Param" name="network">
                <VALUE type="String" value="Atomic-Network␣registry␣office␣1
                    ␣net␣evaluation"/>
              </INIT_PARAM>
              <INIT_PARAM xsi:type="Devs:T_Value_Init_Param" name="network">
                <VALUE type="String" value="Atomic-Network␣datatacenter␣1␣
                    net␣evaluation"/>
              </INIT_PARAM>
            </INIT_PARAMS>
          </INIT_PARAM>
        </INIT_PARAMS>
      </LIBRARY_COMPONENT>
    </COMPONENT_REFERENCE>
    ....
  </DEVS_COUPLED>
</Devs:MODEL>
```

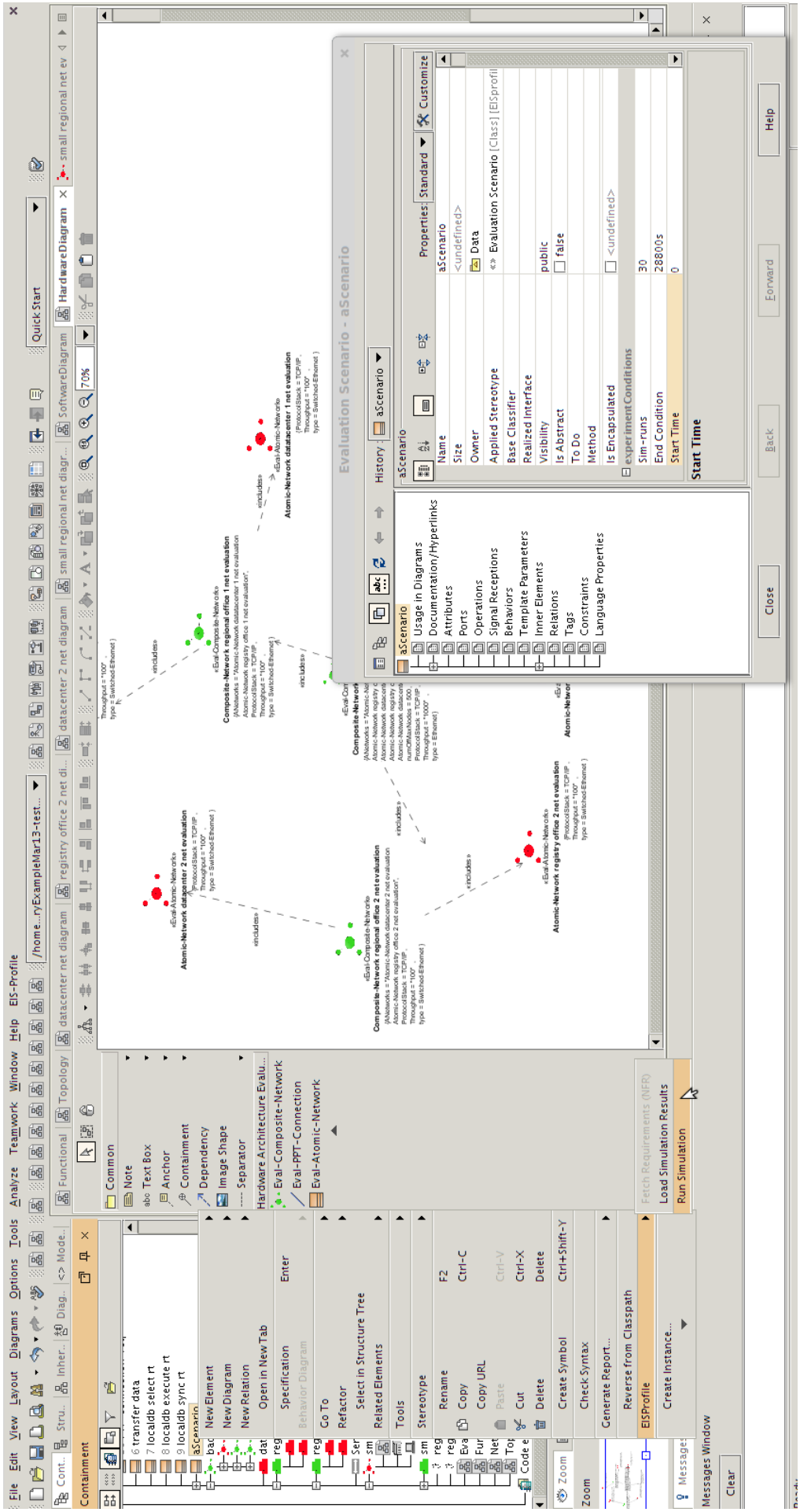
7.3 Εκτέλεση Προσομοίωσης DEVS

Για την εκτέλεση της προσομοίωσης απαιτείται ένα περιβάλλον εκτέλεσης DEVS, το οποίο να υποστηρίζει το μετα-μοντέλο DEVS. Ελλείψει ενός κατάλληλου περιβάλλοντος, επελέγη το DEVSJava και επεκτάθηκε μέσω ενός μετασχηματιστή, ο οποίος μετατρέπει σύνθετα μοντέλα DEVS σε εκτελέσιμο κώδικα DEVSJava.

Δεδομένου ότι τα απαιτούμενα συστατικά λογισμικού προσομοίωσης υλοποιήθηκαν σε μία βιβλιοθήκη DEVSJava, η ουσιαστική πληροφορία που περιλαμβάνεται στο παραχθέν μοντέλο DEVS είναι η αρχικοποίηση, οι διασυνδέσεις και η ρύθμιση τέτοιων συστατικών λογισμικού. Για αυτό το λόγο, υλοποιήθηκε ένας μετασχηματισμός μοντέλων DEVS (XMI) προς την αντίστοιχη κλάση αρχικοποίησης και διαρρύθμισης DEVSJava, η οποία χρησιμοποιεί και ρυθμίζει όλα τα απαιτούμενα σχετικά με EIS συστατικά λογισμικού DEVSJava. Έτσι, διαμορφώνεται ο εκτελέσιμος DEVSJava κώδικας. Αυτός ο μετασχηματισμός είναι ορισμένος σε XSLT, καθώς πρόκειται για έναν απλό συντακτικό μετασχηματισμό. Έτσι, η εκτέλεση της προσομοίωσης είναι εφικτή μετά από αυτό τον μετασχηματισμό,

Το *DEVS_Scenario*, το *DEVS_Node* και το *DEVS_Module* υλοποιήθηκαν ως DEVS Coupled συστατικά λογισμικού, αποτελούμενα από άλλα συστατικά. Όλα τα άλλα, τερματικά (ως προς την ιεραρχία του μοντέλου προσομοίωσης) στοιχεία υλοποιήθηκαν ως DEVS Atomic συστατικά λογισμικού με την αναμενόμενη συμπεριφορά. Κάθε συστατικό λογισμικού προσομοίωσης είναι μία κλάση Java, που επεκτείνει μία από τις δύο βασικές κλάσεις DEVSJava: ViewableAtomic και ViewableDigraph. Στις κλάσεις που επεκτείνουν την ViewableAtomic, η συμπεριφορά του συστατικού λογισμικού προσομοίωσης ορίζεται σε συνάρτηση με την κατάστασή του, όπως αυτή επηρεάζεται από εξωτερικά συμβάντα και την εσωτερική λειτουργικότητα. Στις κλάσεις που επεκτείνουν την ViewableDigraph, ορίζονται σύνθετα συστατικά λογισμικού προσομοίωσης, που περιέχουν άλλα διασυνδεδεμένα συστατικά λογισμικού. Η συμπεριφορά των σύνθετων συστατικών λογισμικού δεν ορίζεται ρητά, αλλά προκύπτει από τη σύνθεση των συμπεριφορών των περιεχόμενων συστατικών. Η εκτέλεση της προσομοίωσης μπορεί να επιθεωρηθεί μέσω του *SimView*, ενός ελεύθερου πλαισίου GUI για DEVSJava [30].

Στη συνέχεια, το σενάριο εκτελείται σε προσομοιωτή DEVS. Το σχήμα 7.3 παρουσιάζει το hardware architecture diagram του evaluation scenario, όπως είναι ακριβώς πριν την προσομοίωση. Σε αυτό το παράδειγμα, το διάγραμμα αποτυπώνει την αρχιτεκτονική δικτύου, στην οποία λειτουργεί η εφαρμογή.



Σχήμα 7.3: EIS Evaluation View: Hardware evaluation diagram.

7.4 Εισαγωγή Αποτελεσμάτων

Με την εκτέλεση της προσομοίωσης παράγονται τα αποτελέσματα και αποθηκεύονται σε αρχεία XML, όπως αυτό που παρουσιάζεται στο listing 7.3. Για κάθε αποτιμώμενη οντότητα, στα αποτελέσματα προσδιορίζεται ένα μοναδικό αναγνωριστικό (id, που χρησιμοποιείται και στο εργαλείο μοντελοποίησης SysML), το στερεότυπό της και οι μετρήσεις από την εκτέλεση της προσομοίωσης.

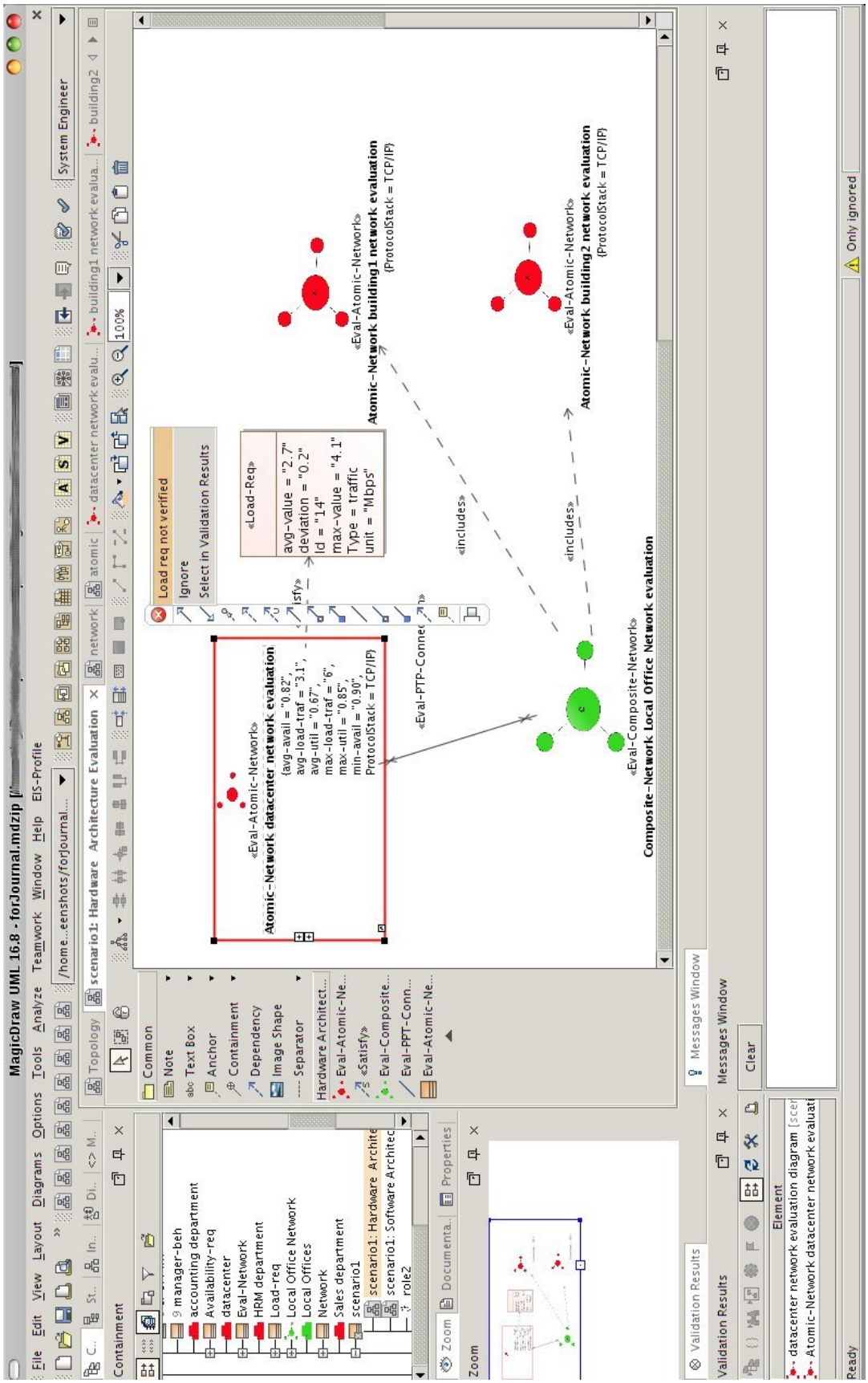
Listing 7.3: Δείγμα αποτελεσμάτων προσομοίωσης για μοντέλο EIS

```
<results>
    <result stereotype="Eval-Atomic-Network::Eval-Service-Replica"
name="update␣taxation␣entity" id="_16_8_14d00da_1366132365894_952758_15900"
count="6">
        <value value="0.00044780904330914747" name="avg-ResponseTime"/>
        <value value="0.00343737965521722799167037010192871093750"
name="max-ResponseTime"/>
    </result>
    <result stereotype="Eval-Atomic-Network::Eval-Service-Replica"
name="update␣taxation␣entity" id="_16_8_14d00da_1366132365900_571644_15928"
count="6">
        <value value="0.00022175926571496682" name="avg-ResponseTime"/>
        <value value="0.00210262284012424061074852943420410156250"
name="max-ResponseTime"/>
    </result>
    ...
</results>
```

Τα αποτελέσματα ενσωματώνονται από το EIS plugin, το οποίο τα προσαρτά στα αντίστοιχα στοιχεία του μοντέλου συστήματος. Το σχήμα 7.4 παρουσιάζει ένα απόσπασμα από την evaluation view, όπου τα αποτελέσματα προσομοίωσης έχουν ενσωματωθεί στα στοιχεία αποτίμησης του μοντέλου συστήματος.

7.5 Αποτίμηση Μοντέλου

Ένα τοπικό δίκτυο αποτελείται από υπολογιστές με συστατικά software να εκτελούνται σε αυτά. Για ένα συγκεκριμένο δίκτυο, η κίνηση που προκύπτει από τις απαιτήσεις έχει ένα μέσο όρο 2.7 Mbps με τυπική απόκλιση 0.2, όπως παρουσιάζεται στο σχήμα 7.4. Τα αποτελέσματα προσομοίωσης καταλήγουν σε μία τιμή 3.1 Mbps, η οποία είναι εκτός επιτρεπτών ορίων. Αυτή η μη επαληθευμένη απαίτηση υποδεικνύεται με κόκκινο χρώμα στο σχήμα 7.4, καθώς δεν ικανοποιείται από το αντίστοιχο στοιχείο του μοντέλου συστήματος. Σε περίπτωση που ο σχεδιαστής συστημάτων αποφασίσει να επέμβει στο μοντέλο συστήματος, για την επίλυση τέτοιων θεμάτων, θα μπορεί να δημιουργήσει εκ νέου το αντίστοιχα τροποποιημένο σενάριο αποτίμησης και να κάνει επαλήθευση απαιτήσεων με χρήση προσομοίωσης.



Σχήμα 7.4: Αυτοματοποιημένη επαλήθευση μη λειτουργικών απαιτήσεων στο Evaluation View του EIS.

7.6 Στοιχεία Υλοποίησης Εφαρμογής

Για την εφαρμογή της προσέγγισης αυτοματοποιημένης προσομοίωσης DEVS για μοντέλα EIS, αναπτύχθηκαν 9 συστατικά λογισμικού προσομοίωσης (1200 γραμμές κώδικα Java), τα οποία υλοποιούν τη συμπεριφορά των κύριων συσταικών λογισμικού EIS (network node, processing node, κλπ.). Αναπτύχθηκαν επίσης 5 βοηθητικές κλάσεις μοντελοποίησης και 4 κλάσεις για το χειρισμό των αποτελεσμάτων προσομοίωσης (540 γραμμές κώδικα Java). Ο μετασχηματισμός QVT υλοποιήθηκε ως ένα σύνολο 17 σχέσεων σε ένα αρχείο 1000 γραμμών. Τέλος, ένας μετασχηματισμός XSLT (360 γραμμές) ορίστηκε για τη μετατροπή μοντέλων DEVS σε εκτελέσιμο κώδικα προσομοίωσης.

Το μοντέλο EIS, που παρουσιάστηκε σε αυτό το κεφάλαιο, είχε 148 κλάσεις και 832 λοιπά στοιχεία (packages, abstractions, realizations, dependencies), που εξήχθησαν από το εργαλείο μοντελοποίησης MagicDraw ως ένα XML document 10573 γραμμών σε μορφή XMI. Ο μετασχηματισμός QVT από EIS σε DEVS εκτελέστηκε σε προσωπικό υπολογιστή με διπύρηνο επεξεργαστή Intel Pentium SU4100 (2M Cache, 1.30 GHz, 800 MHz FSB) και 4GB DDR3 RAM με λειτουργικό σύστημα Ubuntu 12.04 LTS. Το μοντέλο προσομοίωσης DEVS (αρχείο XML 4550 γραμμών σε μορφή XMI) παρήχθη σε 38 δευτερόλεπτα.

Ο μετασχηματισμός αναγνώρισε 9 δικτυακούς κόμβους, 16 εξυπηρετητές και σταθμούς εργασίας, 58 υπηρεσίες και 12 ρόλους, που περιλαμβάνονταν στο σενάριο αποτίμησης, και συγκέντρωσε πληροφορίες και από άλλα στοιχεία, που περιέχονται σε άλλες όψεις του μοντέλου EIS. Το μοντέλο προσομοίωσης DEVS μετατράπηκε, μέσω XSLT, σε εκτελέσιμο πρόγραμμα (2457 γραμμές κώδικα Java), που χρησιμοποιεί τα συστατικά λογισμικού προσομοίωσης. Εκτέλεση 8 ωρών χρόνου προσομοίωσης ολοκληρώθηκε σε 48 δευτερόλεπτα, παράγοντας αποτελέσματα προσομοίωσης για 30 οντότητες του evaluation view σε ένα αρχείο XML 123 γραμμών, σε μορφή XMI. Τα αποτελέσματα αυτά εισήχθησαν στο μοντέλο EIS, στο εργαλείο μοντελοποίησης, όπου και ολοκληρώθηκε η επαλήθευση απαιτήσεων και η οπτική απεικόνιση των προβλημάτων στο σχεδιαστή.

7.7 Σχολιασμός

Ανακεφαλαιώνοντας, για την επαλήθευση απαιτήσεων ενός σεναρίου αποτίμησης στα πλαίσια της εφαρμογής που παρουσιάστηκε σε αυτό το κεφάλαιο, απαιτείται η εκτέλεση των ακόλουθων βημάτων:

- Παραμετροποίηση προσομοίωσης: Ο σχεδιαστής προσδιορίζει τις παραμέτρους προσομοίωσης. Τέτοιες παράμετροι είναι η συνθήκη τερματισμού, η οποία ορίζεται με τη μορφή χρόνου, π.χ. 28800 δευτερόλεπτα, δηλαδή 8 ώρες χρόνου προσομοίωσης και ο αριθμός των επαναληπτικών εκτελέσεων της προσομοίωσης. Ο σχεδιαστής συστημάτων έχει τη δυνατότητα να εκτελέσει την προσομοίωση. Κατά την εκτέλεση της προσομοίωσης, μπορεί να παρακολουθεί την εκτέλεσή της στο περιβάλλον DEVSJava SimView.

- Όταν ολοκληρωθεί η προσομοίωση, ο σχεδιαστής ειδοποιείται για τη διαθεσιμότητα των αποτελεσμάτων προσομοίωσης και μπορεί να επιλέξει την εισαγωγή των αποτελεσμάτων στο σενάριο αποτίμησης.

- Μετά την εισαγωγή των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος, εκτελούνται κανόνες επαλήθευσης και τα στοιχεία του μοντέλου που αποτυγχάνουν σημειώνονται χαρακτηριστικά. Έτσι, ο σχεδιαστής συστημάτων μπορεί να εντοπίσει τα στοιχεία του μοντέλου συστήματος που χρειάζονται προσαρμογή.

Το πλεονέκτημα της προσέγγισης είναι ότι ο σχεδιαστής μπορεί να ορίζει μοντέλα συστήματος και το εργαλείο μοντελοποίησης είναι σε θέση να επαληθεύει μη λειτουργικές απαιτήσεις του μοντέλου. Η προσέγγιση αξιοποιεί 2100 γραμμές κώδικα προσομοίωσης και μετασχηματισμών, που αναπτύχθηκαν άπαξ και μπορούν να χειρίζονται κατά απαίτηση, επαναληπτικά μεγάλα μοντέλα συστημάτων (π.χ. 980 στοιχεία μοντέλου ή 10573 γραμμές) και να παράγουν μεγάλες ποσότητες εκτελέσιμου κώδικα προσομοίωσης (π.χ. 2457 γραμμές κώδικα Java). Δεν απαιτείται αλληλεπίδραση μεταξύ του σχεδιαστή και του περιβάλλοντος προσομοίωσης. Έτσι, τα εκτελέσιμα μοντέλα προσομοίωσης δεν απαιτούν τη γνώση ή την τροποποίηση από το σχεδιαστή. Επιπρόσθετα, ο σχεδιαστής ενημερώνεται για τη διαθεσιμότητα των αποτελεσμάτων προσομοίωσης, τα οποία και εισάγονται στο εργαλείο μοντελοποίησης. Έτσι, τα αποτελέσματα προσομοίωσης είναι διαθέσιμα για περαιτέρω επεξεργασία και αξιοποίηση σε συνδυασμό με άλλες πληροφορίες του μοντέλου συστήματος, όπως ποσοτικά προσδιορισμένες μη λειτουργικές απαιτήσεις, καθιστώντας δυνατή την αυτοματοποιημένη επαλήθευση μη λειτουργικών απαιτήσεων.

Επομένως, τα πλεονεκτήματα της προσέγγισης προσανατολίζονται σε δύο κατευθύνσεις. Η μία έχει να κάνει με την συγκέντρωση κρίσιμης για την προσομοίωση πληροφορίας στα συστατικά λογισμικού προσομοίωσης, όσον αφορά τη συμπεριφορά, και στο μετασχηματισμό, όσον αφορά τη δομή και στις διασυνδέσεις. Οι κωδικοποιημένες πληροφορίες αυτές είναι διαθέσιμες για αξιοποίηση από τους σχεδιαστές συστημάτων, σε οποιοδήποτε μοντέλο. Η άλλη κατεύθυνση έχει να κάνει με την ενσωμάτωση των αποτελεσμάτων στο μοντέλο συστήματος. Η διαθεσιμότητα των αποτελεσμάτων στο μοντέλο συστήματος καθιστά δυνατή την περαιτέρω επεξεργασία τους σε συνδυασμό και με άλλα στοιχεία του μοντέλου. Στη συγκεκριμένη περίπτωση αξιοποιήθηκαν οι ποσοτικά προσδιορισμένες μη λειτουργικές απαιτήσεις και οι περιορισμοί που είχαν οριστεί στα πλαίσια του προφίλ EIS, για την ανάδειξη των στοιχείων του μοντέλου στα οποία παρουσιάζεται μη επαλήθευση των απαιτήσεων.

Κεφαλαιο 8

Προτεινομενες Βελτιστες Πρακτικες

Στα πλαίσια της παρούσας διδακτορικής διατριβής προτάθηκε μία μεθοδολογία για την αυτοματοποίηση της προσομοίωσης κατά την MBSE. Επίσης, προτάθηκαν γενικές προδιαγραφές για πλαίσια που να επιτρέπουν την εφαρμογή της μεθοδολογίας, ενώ υλοποιήθηκε και ένα πλαίσιο, σύμφωνα με τις προδιαγραφές, για την περίπτωση των προσομοιωτών DEVS. Για τον έλεγχο της εφαρμοσιμότητας και της λειτουργικότητας της προτεινόμενης προσέγγισης, σχεδιάστηκαν, υλοποιήθηκαν και εκτελέστηκαν διάφορες πρακτικές εφαρμογές. Δύο από αυτές περιγράφονται στα κεφάλαια 6 και 7 και είναι αντιπροσωπευτικές των δύο παραλλαγών που ακολουθήθηκαν, κατά τις δοκιμαστικές εφαρμογές της προσέγγισης:

- παραγωγή και εκτέλεση μοντέλων προσομοίωσης, αποκλειστικά από το μοντέλο συστήματος, χωρίς την ανάγκη συγγραφής ή χρήσης κώδικα προσομοίωσης, για τα στοιχεία του εξεταζόμενου μοντέλου.

- αξιοποίηση βιβλιοθηκών λογισμικού με συστατικά προσομοίωσης για τον περιορισμό της επιβάρυνσης κατά την προσομοίωση μεγάλων, προ-υπαρχόντων μοντέλων συστημάτων.

Στη συνέχεια, με βάση τις εμπειρίες και τις παρατηρήσεις από τις πρακτικές εφαρμογές της πρότασης, γίνεται μία αποτύπωση προτεινόμενων βέλτιστων πρακτικών, για τη διευκόλυνση ανθρώπων που ενδιαφέρονται για την έρευνα και την εφαρμογή της αυτοματοποιημένης προσομοίωσης στην MBSE.

8.1 Μοντελοποίηση Συστημάτων Σύμφωνα με Πρότυπα

Είναι σχεδόν αυτονόητο, αλλά αναφέρεται εδώ για λόγους πληρότητας, ότι δεδομένης της επιλογής της MBSE, θα υιοθετηθεί κάποιο πρότυπο, βάσει του οποίου θα γίνεται ο ορισμός των μοντέλων συστημάτων. Η χρήση ειδικού σκοπού λύσεων μοντελοποίησης είναι θεμιτή, αλλά παρουσιάζει ένα σύνολο από περιορισμούς, οι οποίοι την καθιστούν ασύμφορη, ιδιαίτερα όταν η μοντελοποίηση του συστήματος γίνεται σε ένα υψηλό επίπεδο και όχι σε ένα επίπεδο επιμέρους υποσυστημάτων πολύ ειδικού σκοπού και με ιδιαιτερότητες.

Η SysML, η οποία είναι πρότυπο του OMG και προτεινόμενη από το INCOSE, είναι ο αδιαφιλονίκητος πρωταγωνιστής στο συγκεκριμένο τομέα. Το γεγονός ότι βασίζεται στη UML την καθιστά έτοιμη για χρήση σε πληθώρα εργαλείων μοντελοποίησης, ενώ της προσδίδει και όλες τις δυνατότητες επέκτασης και εξειδίκευσης της, περιορίζοντας ακόμα περισσότερο την ανάγκη για λύσεις ειδικού σκοπού.

Επομένως, **η χρήση της SysML για τη μοντελοποίηση των συστημάτων ή η δυνατότητα παραγωγής μοντέλων συστήματος σε μορφή SysML θεωρείται δεδομένη**.

8.2 Επιλογή Μεθοδολογίας Προσομοίωσης

Η προσπάθεια αναζήτησης λύσεων για την αυτοματοποίηση της προσομοίωσης στην MBSE μαρτυρά την ύπαρξη κάποιου σχετικού αντικειμενικού στόχου. Αυτός θα μπορούσε να είναι η αναβάθμιση και η διευκόλυνση των δυνατοτήτων αξιολόγησης προτεινόμενων σχεδιαστικών λύσεων για συστήματα μεγάλης κλίμακας ή έρευνα σε σχετικούς τομείς. Σε κάθε περίπτωση, θα πρέπει να αναζητηθούν τα χαρακτηριστικά εκείνα που καθιστούν κάποια μεθοδολογία προσομοίωσης κατάλληλη για τον εκάστοτε στόχο.

Για παράδειγμα, η μελέτη συστημάτων, των οποίων η κατάσταση και η συμπεριφορά μπορούν να αποτυπωθούν καλύτερα με νόμους της φυσικής, είναι προτιμότερη η αναζήτηση μεθόδων προσομοίωσης συνεχούς χρόνου. Αντίθετα, σε περιπτώσεις όπου η συμπεριφορά των συστημάτων περιγράφεται καλύτερα με συμβάντα που καθορίζουν κάθε φορά τη συμπερι-

φορά του συστήματος, είναι προτιμότερες λύσεις προσομοίωσης διακριτού χρόνου.

Η **επιλογή κατάλληλης μεθοδολογίας ή ομάδας λύσεων προσομοίωσης είναι καθοριστική** για την ορθή και ουσιαστική χρήση της προσομοίωσης ως μέθοδο εκτίμησης της συμπεριφοράς του υπό σχεδίαση συστήματος.

8.3 Μετα-Μοντέλο Προσομοίωσης

Οι προτεινόμενες προδιαγραφές πλαισίων περιλαμβάνουν ένα σύνολο από στοιχεία, τα οποία καθιστούν εφικτή την εφαρμογή της μεθοδολογίας. Ανάμεσα σε αυτά, **η ύπαρξη του μετα-μοντέλου προσομοίωσης είναι θεμελιώδης**, αφού επιλύει το ζήτημα των ενδεχόμενων παραλλαγών, που μπορεί να υπάρχουν στις γλώσσες κωδικοποίησης προσομοίωσης για μία συγκεκριμένη μεθοδολογία προσομοίωσης. Επίσης, το μετα-μοντέλο αποτελεί έναν υψηλού επιπέδου τρόπο αναπαράστασης μοντέλων προσομοίωσης, διευκολύνοντας την αυτόματη παραγωγή τους από μετασχηματισμό μοντέλων συστημάτων. Η αυτόματη παραγωγή των μοντέλων προσομοίωσης διευκολύνεται και διασφαλίζεται, **εφόσον το μετα-μοντέλο προσομοίωσης είναι ορισμένο σύμφωνα με ένα πρότυπο**, απλοποιώντας ζητήματα διαχείρισης και επεξεργασίας των μοντέλων. Στην περίπτωση της MBSE, το πρότυπο αυτό είναι συνήθως το MOF, που αποτελεί και την υποδομή πάνω στην οποία έχει οριστεί και η SysML.

Επομένως, η ύπαρξη του μετα-μοντέλο προσομοίωσης, ορισμένου σύμφωνα με κάποια πρότυπη μορφή (MOF) είναι βασική προϋπόθεση για την αυτοματοποίηση της προσομοίωσης στην MBSE.

8.4 Προσομοιωτές Συμβατοί με το Μετα-Μοντέλο Προσομοίωσης

Η ύπαρξη του μετα-μοντέλου δεν εξασφαλίζει από μόνη της τη δυνατότητα εκτέλεσης συμβατών μοντέλων προσομοίωσης. Ερευνητικές προσπάθειες συχνά στοχεύουν στη γενίκευση των χαρακτηριστικών που εμφανίζουν διαφορετικοί προσομοιωτές μίας συγκεκριμένης μεθοδολογίας προσομοίωσης, αλλά δεν καταλήγουν πάντα σε συγκεκριμένες, εφαρμόσιμες λύσεις. Σε περίπτωση που οι υπάρχοντες προσομοιωτές δεν μπορούν να ανταποκριθούν στην απαίτηση για την εκτέλεση υψηλού επιπέδου μοντέλων προσομοίωσης, Θα πρέπει να αναζητηθούν προσομοιωτές που προσφέρονται για επέκταση και προσαρμογή για την εξασφάλιση της λειτουργικότητας αυτής.

Τόσο το επιλεγμένο μετα-μοντέλο προσομοίωσης, όσο και οι προσομοιωτές αφορούν την ίδια μεθοδολογία, οπότε έχουν εννοιολογική συγγένεια. Επομένως, είναι αναμενόμενο να είναι εφικτή η εξεύρεση λύσης στο ζήτημα αυτό. Αν όμως αυτό δεν καταστεί εφικτό, τότε η προσέγγιση δεν μπορεί να είναι λειτουργική, οπότε θα πρέπει να αναθεωρηθούν επιλογές που έχουν γίνει και αφορούν τη μεθοδολογία και το μετα-μοντέλο προσομοίωσης.

Επομένως, **η ύπαρξη τουλάχιστον ενός προσομοιωτή, ο οποίος να μπορεί να λάβει ως είσοδο μοντέλα, σύμφωνα με το επιλεγμένο μετα-μοντέλο είναι απαραίτητη προϋπόθεση**

για τη λειτουργία της προσέγγισης.

8.5 Προφίλ Προσομοίωσης

Υπάρχουν κάποιες περιπτώσεις χρήσης της προτεινόμενης προσέγγισης, όπου είναι ιδιαίτερα σημαντικό η περιγραφή της συμπεριφοράς των στοιχείων ενός συστήματος (πέρα από τη δομή και τις διασυνδέσεις) να γίνει με όρους μοντελοποίησης και να αποφευχθεί η συγγραφή κώδικα προσομοίωσης. Παραδείγματα αυτής της περίπτωσης θα μπορούσαν να είναι:

- Στα πλαίσια κάποιας εκπαιδευτικής διαδικασίας, η χρήση διαγραμμάτων SysML για τον ορισμό της συμπεριφοράς των στοιχείων ενός συστήματος μπορεί να φανεί πιο αποδοτική σε σχέση με τη χρήση παραδειγμάτων κώδικα προσομοίωσης.
- Όταν υπάρχουν ειδικές συνθήκες (π.χ. ασφαλείας), οι οποίες προϋποθέτουν τη χρήση συγκεκριμένων συμβάσεων κατά την προδιαγραφή της συμπεριφοράς των μοντέλων. Η εφαρμογή των συμβάσεων και ο έλεγχος τήρησής τους μπορεί να διασφαλίζεται καλύτερα με την εφαρμογή συγκεκριμένων κανόνων, περιορισμών και ελέγχων που μπορούν να υλοποιηθούν στα πλαίσια του περιβάλλοντος προσομοίωσης.
- Όταν είναι διαθέσιμα και πρέπει να αξιοποιηθούν μοντέλα συστήματος, στα οποία έχει ήδη γίνει εκτεταμένη περιγραφή της συμπεριφοράς των στοιχείων των συστημάτων και απαιτείται μόνο κάποια προσαρμογή ή επιπλέον προσδιορισμός τους.

Σε αυτές τις περιπτώσεις, κύριος στόχος είναι η περιγραφή με όρους μοντελοποίησης τόσο της δομής και των διασυνδέσεων των στοιχείων ενός συστήματος, όσο και **της συμπεριφοράς** τους, που είναι συνήθως και πιο σύνθετη, καθώς εμπλέκει τη διαχείριση της κατάστασής τους, την αντιμετώπιση των εισόδων και την παραγωγή εξόδων. Ακριβώς επειδή η περιγραφή αυτή (α) πρέπει να είναι λεπτομερής και (β) πρέπει να οδηγεί σε εκτελέσιμα μοντέλα προσομοίωσης, θα πρέπει να υπάρχει συμβατότητα με συγκεκριμένη μεθοδολογία προσομοίωσης. Για την εξασφάλιση αυτών των προϋποθέσεων **απαιτείται η αξιοποίηση ενός προφίλ SysML, ειδικά ορισμένο για τη συγκεκριμένη μεθοδολογία προσομοίωσης**. Με αυτό τον τρόπο ο σχεδιαστής περιορίζεται στον ορισμό μοντέλων που είναι πλήρως και ορθά ορισμένα, ώστε να είναι δυνατή η παραγωγή εκτελέσιμων μοντέλων προσομοίωσης.

8.6 Μετασχηματισμός για Προφίλ Συγκεκριμένου Πεδίου

Κατά την εφαρμογή της προσέγγισης είναι ιδιαίτερα πιθανό να προκύψουν περιπτώσεις, όπου ισχύουν μία ή περισσότερες από τις ακόλουθες συνθήκες:

- Υπάρχουν διαθέσιμα πολλά και μεγάλου μεγέθους και πολυπλοκότητας μοντέλα συστημάτων κάποιου πεδίου (σύμφωνα με σχετικό προφίλ), τα οποία απαιτείται να προσομοιωθούν.

- Γίνεται πολλές φορές χρήση κάποιων τύπων στοιχείου με διαφορετικές παραμέτρους κάθε φορά.
- Κάποια κρίσιμα για την προσομοίωση τμήματα του μοντέλου συστήματος μπορεί να (επανα-)παράγονται αυτόματα από άλλα τμήματα του συστήματος, οπότε δεν είναι σταθερά και διαθέσιμα εξ αρχής.
- Δεν είναι διαθέσιμη εντός του μοντέλου συστήματος πληροφορία σχετικά με τη συμπεριφορά των στοιχείων.
- Είναι διαθέσιμες βιβλιοθήκες λογισμικού προσομοίωσης που έχουν υλοποιημένη τη συμπεριφορά των χρησιμοποιούμενων τύπων στοιχείων.

Σε αυτές τις περιπτώσεις, η προσπάθεια υιοθέτησης και εφαρμογής κάποιου προφίλ προσομοίωσης μπορεί να δημιουργήσει περισσότερα προβλήματα από όσα θα μπορούσε να λύσει. Ο εκτεταμένος εμπλουτισμός που απαιτείται για μεγάλα συστήματα μπορεί να εισάγει μεγάλη επιβάρυνση και καθυστερήσεις στο σχεδιαστή. Επίσης, σε περιπτώσεις κατά τις οποίες τμήματα του μοντέλου συστήματος μπορεί να επανα-παράγονται πολλές φορές από κάποια αυτοματοποιημένη διαδικασία, η επιβάρυνση αυτή είναι επαναλαμβανόμενη.

Όταν ισχύουν κάποιες ή αρκετές από τις παραπάνω συνθήκες, είναι ενδεδειγμένη:

- **η αξιοποίηση της πληροφορίας, που είναι διαθέσιμη μέσω του χρησιμοποιούμενου προφίλ συγκεκριμένου πεδίου**
- **η αξιοποίηση των διαθέσιμων βιβλιοθηκών λογισμικού προσομοίωσης**
- **η ανάπτυξη και χρήση μετασχηματισμού, ο οποίος μετατρέπει τα μοντέλα συστήματος σε αντίστοιχες συνθέσεις των κατάλληλα αρχικοποιημένων συστατικών προσομοίωσης** των διαθέσιμων βιβλιοθηκών.

Καθώς ο μετασχηματισμός ορίζεται άπαξ και μπορεί να χρησιμοποιείται πολλές φορές, χωρίς επιβάρυνση για το σχεδιαστή, αντιμετωπίζονται επαρκώς τα ενδεχόμενα επαναλαμβανόμενης αναπροσαρμογής του μοντέλου συστήματος και αυτόματης παραγωγής τμημάτων του.

8.7 Ανάπτυξη Μετασχηματισμών με QVT

Από τη μελέτη της εφαρμογής της μεθοδολογίας σε διαφορετικά πεδία εφαρμογών και την αντίστοιχη ανάπτυξη μετασχηματισμών από μοντέλα συστήματος πεδίου σε μοντέλα προσομοίωσης [1, 67, 68], προέκυψε μία λίστα προτεινόμενων βημάτων, ως μία λογική προσέγγιση για την αποτελεσματική και αποδοτική ανάπτυξη μετασχηματισμών QVT-R, καθώς και κάποιες επιμέρους χρήσιμες πρακτικές, που μπορούν να εφαρμόζονται σε διαφορετικά σημεία του μετασχηματισμού.

8.7.1 Προτεινόμενα Βήματα Ανάπτυξης Μετασχηματισμού QVT

Τα γενικά βήματα, που περιγράφονται στη συνέχεια, μπορούν να οδηγήσουν στην ανάπτυξη ενός μετασχηματισμού QVT μοντέλων συστημάτων σε μοντέλα προσομοίωσης με τρόπο μεθοδικό.

1. Εννοιολογική αντιστοίχιση πεδίου και συστατικών λογισμικού προσομοίωσης

Αρχικά, είναι σημαντική η εγκαθίδρυση μίας εννοιολογικής συσχέτισης μεταξύ των στοιχείων του μοντέλου συστήματος και των συστατικών λογισμικού προσομοίωσης. Αυτό απαιτεί επαρκή γνώση του πεδίου εφαρμογής και του σχετικού προφίλ, καθώς και των δυνατοτήτων των διαθέσιμων συστατικών λογισμικού προσομοίωσης. Έμφαση πρέπει να δίνεται στην ιεραρχική σύσταση των μοντέλων και τις διασυνδέσεις μεταξύ των στοιχείων. Η χρήση συνδυασμένων γραφικών όψεων με τυπικές ιεραρχίες μοντέλων συστημάτων και αντίστοιχες ιεραρχίες συστατικών λογισμικού προσομοίωσης, αναμένεται να προωθήσει την υλοποίηση της εννοιολογικής αντιστοίχισης. Ένα πρώτο βήμα μπορεί να είναι η σημείωση αντίστοιχων τμημάτων εκατέρωθεν και η σύνδεσή τους. Πρόσθετα σχόλια, παρατηρήσεις και περιορισμοί πρέπει να καταγράφονται στην εν λόγω γραφική αναπαράσταση και να συσχετίζονται με τα στοιχεία ή τις διασυνδέσεις που αφορούν. Αυτό θα αποτελέσει το πρώτο άτυπο, αλλά περιεκτικό και ουσιαστικό σημείο αναφοράς για την ανάπτυξη του μετασχηματισμού. Ενδέχεται να υπάρξουν περιπτώσεις, όπου η αναπαράσταση των αντιστοιχίσεων σε ένα διάγραμμα να είναι ιδιαίτερα πολύπλοκη. Σε αυτές τις περιπτώσεις το βασικό διάγραμμα θα πρέπει να περιλαμβάνει αφαιρετικά τμήματα, τα οποία αναλύονται σε άλλα διαγράμματα. Αν υπάρχουν διασυνδέσεις μεταξύ στοιχείων διαφορετικών τμημάτων, αυτές θα πρέπει να σημειωθούν με ιδιαίτερη προσοχή, για να μη παραληφθούν στη συνέχεια.

2. Αντιστοίχιση Στοιχείων από το Γενικό προς το Ειδικό

Ο μετασχηματισμός εκτελείται μέσω (α) την εφαρμογή σχέσεων QVT σε στοιχεία του μοντέλου συστήματος και του υπό διαμόρφωση μοντέλου προσομοίωσης, και (β) περαιτέρω προώθησης της εφαρμογής άλλων σχέσεων σε πιο λεπτομερή στοιχεία, όπως προσδιορίζονται στα τμήματα *where* των αρχικών σχέσεων. Το γεγονός αυτό παραπέμπει σε μία προσέγγιση από το γενικό προς το ειδικό, αντίστοιχη της υποδομής μοντελοποίησης προσομοίωσης (μετα-μοντέλο), που επιτρέπει την ιεραρχική σύνθεση του μοντέλου προσομοίωσης. Το μοντέλο συστήματος μπορεί να συντίθεται από πολλαπλώς διασυνδεδεμένα στοιχεία, αποκλίνοντας από τη λογική γενικό προς ειδικό. Ωστόσο, μία αντιστοίχιση στοιχείων του μοντέλου συστήματος (ή ομάδες συνδυασμένων αντιστοιχίσεων) σε στοιχεία το μοντέλου συστήματος πρέπει να υπακούν στη λογική γενικό προς ειδικό. Καθώς οι αντιστοιχίσεις αυτές υλοποιούνται ως σχέσεις QVT, ο μετασχηματισμός οργανώνεται και δομείται ως μία ιεραρχία σχέσεων.

3. Διαμόρφωση Μοντέλου Προσομοίωσης

Η δομή του μοντέλου προσομοίωσης (προορισμού) προδιαγράφεται στο *enforce domain* των σχέσεων του μετασχηματισμού, χρησιμοποιώντας τους τύπους, που είναι διαθέσιμοι στο μετα-μοντέλο προσομοίωσης. Απλές τιμές πεδίων, που είναι διαθέσιμες σε αντίστοιχα πεδία του μοντέλου συστήματος, μπορούν να ανατεθούν, με τη χρήση τοπικών (στη σχέση) μεταβλητών, όπως για παράδειγμα το *v_networkName* στο Listing 8.1 και τα *sp*, *sc*, *tp* και *tc* στο Listing 8.2. Τιμές από πεδία επέκτασης (stereotype tagged values) ή που προκύπτουν με άλλο, σύνθετο τρόπο, μπορούν επίσης να ανατεθούν με τη χρήση τοπικών μεταβλητών. Στις σύνθετες αυτές περιπτώσεις οι μεταβλητές αρχικοποιούνται στο τμήμα *where* της σχέσης (π.χ. *v_throughput* στο Listing 8.1).

4. Δήλωση Προϋποθέσεων

Όλοι οι λογικοί περιορισμοί, που σημειώθηκαν κατά την εννοιολογική αντιστοίχιση ή αναγνωρίστηκαν σε επόμενο στάδιο, πρέπει να δηλωθούν με τυπικό τρόπο στις αντίστοιχες σχέσεις. Ένας εγγενής τρόπος ορισμού περιορισμών για την επιλογή στοιχείων του μοντέλου συστήματος (αφετηρίας) είναι η επιβολή τους με τη χρήση συγκεκριμένων τύπων στο *checkonly* δομικό μέρος της σχέσης. Για παράδειγμα, στο Listing 8.2, το *packagedElement c*, θα μπορούσε να είναι οποιουδήποτε τύπου στοιχείο, που περιέχεται σε ένα πακέτο. Ωστόσο, λόγω του προσδιορισμού του αναμενόμενου τύπου (*InformationFlow*), αυτή η σχέση θα εφαρμοστεί μόνο σε περιεχόμενα στοιχεία αυτού του τύπου. Με αντίστοιχο τρόπο, άλλοι βασικοί περιορισμοί μπορούν να εφαρμόζονται στη δομή, τους τύπους και τις τιμές πεδίων του *checkonly* μέρους μίας σχέσης.

Πρόσθετες, ρητές προϋποθέσεις ορίζονται στο τμήμα *when* των σχέσεων, ακολουθώντας μία από τις ακόλουθες περιπτώσεις:

- Αποτίμηση Λογικών Εκφράσεων: Ένα εύρος λογικών περιορισμών μπορούν να δηλωθούν στα στοιχεία των μοντέλων συστήματος και προσομοίωσης, χρησιμοποιώντας εκφράσεις OCL. Η εφαρμογή σε ένα στοιχείο του μοντέλου συστήματος ενός απαιτούμενο stereotype είναι μία συχνά χρησιμοποιούμενη προϋπόθεση αυτή της μορφής. Το τμήμα *when* της σχέσης που παρουσιάζεται στο Listing 8.1 είναι ένα αντιπροσωπευτικό παράδειγμα.

- Διάδοση (Propagation) του Ελέγχου: Ο έλεγχος συγκεκριμένων στοιχείων ως προϋπόθεση για την εφαρμογή μίας σχέσης μπορεί να είναι πολύπλοκος και να απαιτεί σύνθετους επιμέρους ελέγχους. Σε αυτές τις περιπτώσεις ο έλεγχος γίνεται με την κλήση άλλων σχέσεων στο τμήμα *when* της σχέσης. Η βασική σχέση θα εφαρμοστεί μόνο εφόσον είναι επιτυχής η εφαρμογή των επιμέρους σχέσεων (βάσει των δικών τους προϋποθέσεων).

5. Δοκιμές

Η επιτυχής εκτέλεση ενός μετασχηματισμού QVT οδηγεί πάντα σε ένα μοντέλο, που συμμορφώνεται με το μετα-μοντέλο προορισμού. Ως εκ τούτου, είναι διασφαλισμένη η απουσία

Listing 8.1: Αρχικοποίηση συστατικών λογισμικού προσομοίωσης

```
relation eisEvalNetwork2ComponentReference {
    v_networkName: String;
    v_throughput: String;
  checkonly domain eis scenario : uml::Class {
    nestedClassifier = network : uml::Classifier {
      name = v_networkName } };
  enforce domain devs componentReferenceList : Devs::
     T_Component_Reference_List {
    COMPONENT_REFERENCE = componentReference : Devs::
       T_Component_Reference {
      text = v_networkName,
      LIBRARY_COMPONENT = libraryComponent : Devs::T_Library_Component
         {
        class = 'Network',
        _package = 'eis',
        INIT_PARAMS = ips : Devs::T_Init_Params {
          INIT_PARAM = ip : Devs::T_Value_Init_Param {
            name = 'throughput',
            VALUE = v : Devs::T_Value {
              type = 'Real',
              value = v_throughput } },
          INIT_PARAM = ip0 : Devs::T_Array_Init_Param {
            name = 'nodes',
            INIT_PARAMS = nodes :Devs::T_Init_Params { } }, ... } } } }
                ...
  when {
    network.getAppliedStereotype('Eval-Atomic-Network::Eval-Atomic-
       Network')->notEmpty(); ... }
  where {
    v_throughput = network.getValue(network.getAppliedStereotype('Eval-
       Atomic-Network::Eval-Atomic-Network'),'Throughput').oclAsType(
       String);
    eisEvalNode2ComponentReference(scenario, network, nodes,
       componentReferenceList); ... } }
```

Listing 8.2: Δημιουργία συνδέσεων συστατικών λογισμικού προσομοίωσης από information flows

```
relation informationFlows2internalCoupling {
  sc, sp, tc, tp: String;
  checkonly domain u pack: uml::Package {
    packagedElement = c : uml::InformationFlow {
      informationSource = is : uml::Port {
        owner = so : uml::Classifier {
          name = sc },
        name = sp },
      informationTarget = it : uml::Port {
        owner = to : uml::Classifier {
          name = tc },
        name = tp } } };
  enforce domain d ic: Devs::T_Internal_Coupling {
    INTERNAL_COUPLING_SPEC = ics : Devs::T_Internal_Coupling_Spec {
      INTERNAL_OUTPUT = io : Devs::T_Component_Port {
        component = sc,
        port = sp },
      INTERNAL_INPUT = ii : Devs::T_Component_Port {
        component = tc,
        port = tp } } }; }
```

συντακτικών λαθών. Αντίθετα, η εννοιολογική συνάφεια και αντιστοιχία μεταξύ του μοντέλου αφετηρίας και προορισμού πρέπει να ελεγχθεί. Έμφαση πρέπει να δίνεται στον έλεγχο της δομικής σύνθεσης και των διασυνδέσεων στο παραγόμενο μοντέλο προσομοίωσης. Καθώς τα μοντέλα συστημάτων και τα μοντέλα συστήματος μπορεί να είναι ιδιαίτερα σύνθετα με πολλούς τρόπους, συνιστάται καθ' όλη τη διάρκεια της ανάπτυξης του μετασχηματισμού. Έτσι, βασικά λάθη και παραλείψεις θα εντοπιστούν στην αρχή της διαδικασίας. Μία άλλη προτεινόμενη τακτική είναι η διεξαγωγή δοκιμών με δοκιμαστικά μοντέλα συστήματος με σταδιακά αυξανόμενη πολυπλοκότητα, ώστε τα σενάρια δοκιμών να είναι απλά στην αρχή και η πολυπλοκότητά του να αυξάνεται με ελεγχόμενο τρόπο.

8.7.2 Χρήσιμα Μοτίβα Χρήσης QVT

Πέρα από τα προτεινόμενα βήματα, που περιγράφονται στην υποενότητα 8.7.1, και είναι χρήσιμα κατά την προοδευτική ανάπτυξη ενός μετασχηματισμού QVT, υπάρχουν και ειδικά θέματα, τα οποία μπορεί να προκύψουν σε διαφορετικά μέρη του μετασχηματισμού. Τα μοτίβα χρήσης της QVT, που περιγράφονται στη συνέχεια μπορούν να εφαρμοστούν για την αντιμετώπιση τέτοιων περιπτώσεων.

Πολλαπλά Πεδία checkonly/enforce

Συνήθως, σε μία σχέση QVT χρησιμοποιούνται ένα πεδίο *checkonly* (αφετηρίας) και ένα *enforce* (προορισμού). Ωστόσο, υπάρχουν περιστάσεις, όπου απαιτείται η αξιοποίηση πληροφοριών, που βρίσκονται σε διαμετρικά αντίθετα σημεία του μοντέλου αφετηρίας, για την παραγωγή του μοντέλου προορισμού. Για την ικανοποίηση αυτής της απαίτησης, είναι δυνατό να οριστούν περισσότερα του ενός πεδία *checkonly* σε μία σχέση QVT. Για παράδειγμα, το Listing 8.3 παρουσιάζει τη σχέση QVT για την αρχικοποίηση των συστατικών λογισμικού τύπου *Role*. Σε αυτό το παράδειγμα, ο καταμερισμός ρόλων σε σταθμούς εργασίας ορίζεται σε ένα γενικό επίπεδο του μοντέλου, αλλά απαιτούνται και πληροφορίες για το δίκτυο και τον υπολογιστικό. Έτσι, χρησιμοποιούνται τρία πεδία *checkonly*, που καθιστούν τρεις όψεις του ίδιου μοντέλου EIS διαθέσιμες, για την αρχικοποίηση των συστατικών λογισμικού προσομοίωσης τύπου *Role*. Όμοια, πολλαπλά πεδία *enforce* μπορούν να χρησιμοποιηθούν σε μία σχέση QVT, για τον παράλληλο προσδιορισμό διαφορετικών μερών του μοντέλου προορισμού.

Αξιοποίηση Διαδικαστικών Χαρακτηριστικών

Οι σχέσεις QVT αναβαθμίζουν τους μετασχηματισμούς μοντέλων σε ένα υψηλό, δηλωτικό επίπεδο, με έμφαση στη δομή του μοντέλου και τους περιορισμούς. Ωστόσο, συγκεκριμένες τιμές, που θα έπρεπε να αποδοθούν σε χαρακτηριστικά του μοντέλου προορισμού, μπορεί να είναι προσβάσιμα, μέσω μίας σειράς από κλήσεις επερωτήσεων σε στοιχεία του μοντέλου αφετηρίας. Σε αυτές τις περιπτώσεις, μπορούν να χρησιμοποιηθούν τα διαδικαστικά χαρακτηριστικά της QVT, ώστε να περιοριστεί η πολυπλοκότητα των σχέσεων. Στο μετασχηματισμό από μοντέλα EIS σε μοντέλα DEVS, αρκετές τιμές πρόσθετων πεδίων (stereotype tagged values)

Listing 8.3: Χρήση πολλαπλών πεδίων checkonly

```
relation eisNodeRole2InitParam {
  evalRoleName : String;
  checkonly domain eis model : uml::Model {
    packagedElement = evalUsageAllocation : uml::Abstraction {
      client = evalRoleFromUsage : uml::NamedElement { },
      supplier = evalWorkstation : uml::NamedElement { } } };
  checkonly domain eis evalNetwork : uml::Class {
    nestedClassifier = evalRole : uml::Classifier {
      name = evalRoleName } };
  checkonly domain eis node : uml::Class { };
  enforce domain devs rolesIps : Devs::T_Init_Params { ... } }
```

Listing 8.4: Ορισμός QVT query

```
query getStringValueFromStereotypes(el: uml::Classifier, sters: Set(
    String), prop: String) : String {
     el.getValue(el.getAppliedStereotype(sters->any(i|el.
        getAppliedStereotype(i)->notEmpty())),prop).oclAsType(String) }
```

του μοντέλου EIS απαιτούνταν για την παραγωγή των αντίστοιχων στοιχείων του μοντέλου DEVS. Επίσης, υπάρχουν σχέσεις που επεξεργάζονται στοιχεία, που μπορεί να έχουν ένα εύρος από σχετικά stereotypes (π.χ. υπολογιστικοί κόμβοι, που μπορεί να είναι είτε σταθμοί εργασίας, είτε εξυπηρετητές στα EIS). Για την πρόσβαση σε επεκτεταμένες τιμές (με το ίδιο όνομα), ανεξάρτητα από το συγκεκριμένο stereotype ενός στοιχείου, ο κώδικας που περιέχεται στο *query* του Listing 8.4 πρέπει να χρησιμοποιηθεί. Η χρήση τέτοιων πολύπλοκων τμημάτων κώδικα μπορεί να γίνει σε σχέσεις QVT, μέσω απλών κλήσεων των δηλωμένων queries. Στο παραπάνω παράδειγμα, το query *getStringValueFromStereotypes* δέχεται τρεις παραμέτρους: το στοιχείο του μοντέλο, στο οποίο θα αναζητηθεί η επεκτεταμένη τιμή (*el*), το σύνολο των ενδεχόμενων stereotypes του στοιχείου (*sters*) και το όνομα του πεδίου (*prop*). Το query επιστρέφει την τιμή (string) της τιμής που εντοπίστηκε. Το Listing 8.5 παρουσιάζει μία κλήση του query, ώστε να ληφθεί η υπολογιστική ισχύς (*ProcPower*) του κόμβου (*node*), ανεξάρτητα από το αν είναι εξυπηρετητής (*Eval-Server*) ή σταθμός εργασίας (*Eval-Workstation*).

Listing 8.5: Χρήση QVT query

```
value = getStringValueFromStereotypes(node, Set{'Eval-Atomic-Network::
    Eval-Server','Eval-Atomic-Network::Eval-Workstation'}, 'ProcPower')
```

Αξιοποίηση Χαρακτηριστικών Συμπερασμού OCL

Συνήθως τα μοντέλα εξυπηρετούν την αναπαράσταση σύνθετων στοιχείων πραγματικών συστημάτων, που είναι διασυνδεδεμένα με διάφορους τρόπους. Επομένως, τα μοντέλα αναμένονται να είναι σύνθετα, τόσο σε επίπεδο δομής όσο και σε επίπεδο διασυνδέσεων. Αυτή η πολυπλοκότητα μπορεί να αντιμετωπιστεί αποτελεσματικά με τη δηλωτική προσέγγιση, που προσφέρει η QVT-R για τον ορισμό μετασχηματισμών μοντέλων, όταν συμπληρώνεται με τα χαρακτηριστικά συμπερασμού της OCL. Αυτά τα χαρακτηριστικά επιτρέπουν την εξαγωγή συμπερασμάτων, βάσει απλών, υψηλού επιπέδου λογικών εκφράσεων που αφορούν στοιχεία μοντέλων με πολλαπλές τιμές ή διασυνδέσεις. Για παράδειγμα, στο query,που παρουσιάστηκε στο Listing 8.4, κάθε στοιχείο του συνόλου των stereotypes που δίνεται ως παράμετρος (*sters*), ελέγχεται αν έχει τη ζητούμενη επεκτεταμένη τιμή (tagged value), με το όνομα που έχει η τιμή της παραμέτρου *prop*. Αξιοποιείται όποια (*any*) τιμή βρεθεί. Τέτοια χαρακτηριστικά παρέχουν ισχυρές δυνατότητες διερεύνησης των μοντέλων που προάγουν την απλότητα των σχέσεων QVT και την αποτελεσματικότητα της διαδικασίας ανάπτυξης του μετασχηματισμού.

8.8 Επαλήθευση Μη Λειτουργικών Απαιτήσεων

Για τη διευκόλυνση της αποτίμησης σχεδιαστικών λύσεων, προτείνεται η πρόβλεψη θέσεων στο μοντέλο συστήματος, όπου θα καταχωρούνται οι τιμές που προκύπτουν από τις μετρήσεις των χαρακτηριστικών λειτουργίας του συστήματος, κατά την εκτέλεση της προσομοίωσης, και την επεξεργασία τους.

Η εισαγωγή των τιμών αυτών είναι εφικτή με κατάλληλο, απλό μετασχηματισμό του μοντέλου συστήματος, ο οποίος θα διατηρεί το μοντέλο συστήματος ως έχει και θα το εμπλουτίζει με τις τιμές απόδοσης. Ωστόσο, η εξαγωγή ενός μοντέλου συστήματος από το εργαλείο μοντελοποίησης, ο μετασχηματισμός του και η επανεισαγωγή του σε αυτό ενδέχεται να δημιουργήσουν ζητήματα ανάγνωσης του μοντέλου από το εργαλείο μοντελοποίησης. Ως εκ τούτου, οι δοκιμές της συγκεκριμένης λειτουργικότητας είναι επιβεβλημένες. Σε περίπτωση που διαπιστωθούν προβλήματα, η κατάσταση μπορεί να αντιμετωπιστεί και με ανάπτυξη εργαλείου (plugin) ειδικού σκοπού στο εργαλείο μοντελοποίησης.

Με τις τιμές απόδοσης των στοιχείων του μοντέλου διαθέσιμες, είναι δυνατή η αυτοματοποίηση της επαλήθευσης των μη λειτουργικών απαιτήσεων. Αρκεί στο μοντέλο συστήματος να έχουν αποτυπωθεί με ποσοτικό τρόπο περιορισμοί, σχετικά με τις επιτρεπτές τιμές των στοιχείων του όσον αφορά χαρακτηριστικά, όπως ρυθμός παραλαβής και εξυπηρέτησης αιτημάτων, ποσοστό χρησιμοποίησης (utilization) κ.λ.π.

Επομένως, **η ενσωμάτωση των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος σε συνδυασμό με προδιαγραφή μη λειτουργικών απαιτήσεων με ποσοτικό τρόπο, μπορεί να οδηγήσει σε αυτοματοποίηση της επαλήθευσης** των τελευταίων.

8.9 Συμβατότητα Εργαλείων με Πρότυπα - Διαλειτουργικότητα

Για τη λειτουργία της προτεινόμενης μεθοδολογίας αξιοποιούνται πολλά και, ενδεχομένως, διαφορετικά εργαλεία:

- Για τον ορισμό μετα-μοντέλων
- Για τον ορισμό προφίλ
- Για τον ορισμό μοντέλων
- Για τον ορισμό μετασχηματισμών
- Για την εκτέλεση μετασχηματισμών
- Για την εκτέλεση προσομοίωσης

Οι προτεινόμενες λύσεις είναι προσανατολισμένες στην εκτεταμένη χρήση προτύπων με σκοπό την ελαχιστοποίηση των περιορισμών σχετικά με τη χρήση εργαλείων λογισμικού. Θεωρητικά, όλα τα εργαλεία που *υποστηρίζουν* κάποια πρότυπα, άμεσα ή μέσω λειτουργιών εισαγωγής/εξαγωγής, θα έπρεπε να είναι αξιοποιήσιμα. Εντούτοις, **η διαλειτουργικότητα δεν επιτυγχάνεται πάντα στην πράξη**. Για αυτό το λόγο, είναι **απαραίτητη η διεξαγωγή δοκιμαστικών ελέγχων των επιλεγμένων εργαλείων λογισμικού**, πριν την οριστική επιλογή τους.

Κεφαλαιο 9

Συμπερασματα

9.1 Σχολιασμός Προσέγγισης και Εφαρμογών

Στο παρόν κείμενο παρουσιάστηκε η διατριβή για την αυτοματοποίηση της προσομοίωσης στην MBSE. Αρχικά, δόθηκε ένα περίγραμμα του επιστημονικού υποβάθρου, που σχετίζεται με τη διατριβή. Αυτό σχετίζεται κυρίως με δύο επιστημονικά πεδία: τη βασισμένη σε μοντέλα ανάπτυξη συστημάτων και την προσομοίωση συστημάτων.

Στη συνέχεια παρουσιάστηκαν ερευνητικές προσπάθειες, οι οποίες έχουν γίνει ή γίνονται στα ανωτέρω βασικά πεδία και οι οποίες κινούνται σε κατευθύνσεις σχετικές με την παρούσα διατριβή. Από την εξέτασή τους προκύπτει ότι υπάρχουν κάποια θέματα που δεν αντιμετωπίζονται επαρκώς με τις ήδη προταθείσες προσεγγίσεις. Μία λίστα από τα σημαντικότερα θέματα, που παρουσιάζονται αθροιστικά ή διαζευκτικά στις σχετικές προτάσεις είναι:

- Οι προτεινόμενες λύσεις βασίζονται σε κλειστά (proprietary) εργαλεία, γλώσσες ή/και περιβάλλοντα, χωρίς να παρέχονται διεπαφές προς πρότυπα. Έτσι εισάγονται περιορισμοί σε λύσεις συγκεκριμένων εταιρειών και οι προτάσεις χάνουν τη γενικότητά τους.
- Η παραγωγή κώδικα προσομοίωσης δεν είναι πλήρως αυτοματοποιημένη και, επομένως, απαιτείται συγγραφή/συμπλήρωση του κώδικα προσομοίωσης. Έτσι αυξάνεται η πιθανότητα εισαγωγής σφαλμάτων.
- Οι λύσεις αφορούν και στοχεύουν σε συγκεκριμένα πεδία, π.χ. συστήματα πραγματικού χρόνου. Αυτές οι λύσεις είναι δύσκολο να εφαρμοστούν σε άλλα πεδία εφαρμογής.
- Τα αποτελέσματα της ανάλυσης με προσομοίωση δεν εισάγονται στο μοντέλο συστήματος. Κατ' αυτό τον τρόπο, δεν είναι δυνατή η περαιτέρω ανάλυση και αξιοποίηση των αποτελεσμάτων, σε συνδυασμό και με άλλα στοιχεία του μοντέλου συστήματος, για την εξαγωγή πρόσθετων συμπερασμάτων σχετικά με το μοντέλο.

Ακολούθως, αναλύθηκε η πρόταση της διατριβής, η οποία συνεισφέρει στην αντιμετώπιση των ανοικτών ζητημάτων, μέσω (α) μίας προτεινόμενης μεθοδολογίας και (β) των προδιαγραφών για την ανάπτυξη πλαισίων, τα οποία θα έχουν τη δυνατότητα να υποστηρίζουν τη μεθοδολογία αυτή. Τόσο η μεθοδολογία, όσο και οι προδιαγραφές των πλαισίων ορίστηκαν με ένα τρόπο γενικό σε σχέση με το πεδίο εφαρμογής, το περιβάλλον προσομοίωσης και τις χρησιμοποιούμενες γλώσσες αναπαράστασης μοντέλων συστημάτων και προσομοίωσης. Τα πρότυπα υιοθετήθηκαν και αξιοποιήθηκαν στο μέγιστο βαθμό, ώστε να προαχθεί η δια-λειτουργικότητα έναντι των κλειστών λύσεων, εφαρμογών και προσεγγίσεων.

Η προτεινόμενη μεθοδολογία είναι αρκετά γενική, ώστε να μπορεί να εφαρμοστεί σε ποικιλία πεδίων εφαρμογών και να λειτουργήσει με αξιοποίηση διαφορετικών προσομοιωτών. Κύριος στόχος της μεθοδολογίας είναι η αυτοματοποίηση όλων των βημάτων που αφορούν την παραγωγή και εκτέλεση των μοντέλων προσομοίωσης και η ενσωμάτωση των αποτελεσμάτων στο μοντέλο συστήματος. Με την αυτοματοποίηση βελτιώνεται ουσιαστικά η διαδικασία της αποτίμησης μίας προτεινόμενης λύσης, αφού η διαδικασία απλοποιείται, επιταχύνεται και αποσφαλματώνεται λόγω της αποφυγής ανθρώπινης παρέμβασης. Η εισαγωγή των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος τα καθιστά διαθέσιμα τόσο στο σχεδιαστή για αντιπαραβολή με οποιοδήποτε στοιχείο του μοντέλου, όσο και σε εργαλεία, που μπορούν να αναπτυχθούν, για την περαιτέρω αξιοποίησή τους (π.χ. αυτόματη επαλήθευση απαιτήσεων, αυτόματη παραγωγή προτάσεων για βελτιστοποίηση του μοντέλου κλπ.).

Και οι προδιαγραφές για τα απαιτούμενα πλαίσια υποστήριξης της προτεινόμενης μεθοδολογίας έχουν οριστεί με τρόπο γενικό, ώστε να μην εισάγονται περιορισμοί ως προς το χρησιμοποιούμενο περιβάλλον προσομοίωσης και τα ενδιάμεσα μοντέλα αναπαράστασης και εργαλεία. Οι προδιαγραφές αναδεικνύουν το σημαντικό ρόλο των μετα-μοντέλων προσομοίωσης, τα οποία προτείνεται να προδιαγράφονται σύμφωνα με την υποδομή MOF. Έτσι, εξασφαλίζεται η δυνατότητα αξιοποίησης γλωσσών μετασχηματισμού υψηλού επιπέδου, όπως η QVT. Έτσι, προσδίδονται σημαντικά πλεονεκτήματα στη διαδικασία μετασχηματισμού μοντέλων συστημάτων σε μοντέλα προσομοίωσης, αφού με την QVT η ενδοσκόπηση των μοντέλων αφετηρίας και προορισμού απλοποιείται. Επίσης, τα συντακτικά σφάλματα εκμηδενίζονται, καθώς τα μετα-μοντέλα αφετηρίας και προορισμού δίνονται ως παράμετροι κατά τον ορισμό του μετασχηματισμού.

Ως απόδειξη της δυνατότητας εφαρμογής της προσέγγισης, υλοποιήθηκε ένα πλαίσιο για την αξιοποίηση της μεθοδολογίας προσομοίωσης DEVS στη βασισμένη σε μοντέλα ανάπτυξη συστημάτων. Ορίστηκαν κατάλληλα προφίλ, μετα-μοντέλα, μετασχηματισμοί και λοιπά υποστηρικτικά εργαλεία, σύμφωνα με τα προβλεπόμενα στην προτεινόμενη προσέγγιση. Τέλος, παρουσιάστηκαν δύο εφαρμογές, όπου το υλοποιημένο πλαίσιο χρησιμοποιήθηκε και οδήγησε σε αυτοματοποιημένη παραγωγή εκτελέσιμων μοντέλων προσομοίωσης, τα οποία και εκτελέστηκαν. Οι δύο εφαρμογές, που παρουσιάστηκαν δεν είναι δύο πανομοιότυπες χρήσεις του πλαισίου που αναπτύχθηκε. Πρόκειται για δύο παραλλαγές στον τρόπο αξιοποίησης του βασικού πλαισίου, που υλοποιήθηκε για την υποστήριξη της προσομοίωσης με DEVS, οι οποίες παρουσιάζουν διαφορετικά χαρακτηριστικά.

Η μία εφαρμογή δίνει έμφαση στη δυνατότητα ορισμού του μοντέλου προσομοίωσης αποκλειστικά και καθ' ολοκληρία στο περιβάλλον μοντελοποίησης συστημάτων (εργαλείο UML / SysML). Σε αυτή την περίπτωση δεν απαιτείται συγγραφή κώδικα προσομοίωσης, ούτε κατά την δημιουργία του πλαισίου, ούτε κατά τη χρήση του, στην αποτίμηση συγκεκριμένων μοντέλων. Αυτό συνεπάγεται ότι τα μοντέλα συστήματος θα πρέπει να είναι εμπλουτισμένα επαρκώς, με την πλήρη προδιαγραφή της συμπεριφοράς του συνόλου των υποσυστημάτων, που το αποτελούν. Στη συγκεκριμένη περίπτωση και δεδομένου ότι το πλαίσιο έχει υλοποιηθεί για εκτέλεση της προσομοίωσης σε περιβάλλον DEVS, εκτός από τις διασυνδέσεις των υποσυστημάτων (αντίστοιχο του DEVS internal και external coupling), το μοντέλο συστήματος οφείλει να περιλαμβάνει και την απαιτούμενη πληροφορία για τον ορισμό των καταστάσεων και της συμπεριφοράς των τερματικών στοιχείων του μοντέλου, Αυτό διασφαλίζεται από τους περιορισμούς του προφίλ SysML για DEVS του πλαισίου.

Ωστόσο, καθώς η εφαρμογή είναι προσανατολισμένη στον τρόπο περιγραφής της δομής και της συμπεριφοράς στα πλαίσια του DEVS, ο σχεδιαστής συστημάτων πρέπει να έχει επαρκή εξοικείωση με αυτό. Αυτό δημιουργεί πρόσθετες απαιτήσεις και επιβαρύνει τη διαδικασία του σχεδιασμού. Η επιβάρυνση αυτή όμως είναι δικαιολογημένη για τα υπόλοιπα οφέλη που προσφέρει. Αν η προσομοίωση του μοντέλου συστήματος είναι απαραίτητη, τότε η εναλλακτική προσέγγιση θα ήταν με συγγραφή κώδικα προσομοίωσης, εκτός του μοντέλου συστήματος. Σε αυτή την περίπτωση αυτοματισμοί και έλεγχοι για τον περιορισμό εισαγωγής σφαλμάτων δεν θα ήταν διαθέσιμοι. Επίσης, από την οπτική του περιβάλλοντος προσομοίωσης, το DEVS δεν έχει διαθέσιμο κάποιο εργαλείο για τον ορισμό μοντέλων προσομοίωσης με γραφικό τρόπο. Η εφαρμογή αυτή δείχνει ότι το κενό αυτό μπορεί να καλυφθεί και μάλιστα με ένα τρόπο γενικό και συμβατό με πρότυπα και λύσεις της ευρύτερης περιοχής της μοντελοποίησης συστημάτων.

Η δεύτερη εφαρμογή που παρουσιάστηκε βασίστηκε στο πλαίσιο που αναπτύχθηκε για την υποστήριξη της μεθοδολογίας με χρήση του περιβάλλοντος προσομοίωσης DEVS, αλλά εισήχθησαν και κάποια πρόσθετα στοιχεία για την αντιμετώπιση ορισμένων από τα θέματα που αντιμετωπίστηκαν κατά την πρώτη εφαρμογή. Έγινε προσπάθεια ώστε να εισαχθεί αξιοποιήσιμη, κωδικοποιημένη πληροφορία για την προσομοίωση κατά την προεργασία της εφαρμογής, η οποία να είναι διαθέσιμη κατά την αποτίμηση συγκεκριμένων μοντέλων. Αυτό έγινε με:

- την εξέταση ενός πεδίου εφαρμογής (EIS) και τον εντοπισμό τμημάτων του μοντέλου, για τα οποία θα ήταν χρήσιμη η ύπαρξη και η αξιοποίησή συστατικών λογισμικού προσομοίωσης με συγκεκριμένη συμπεριφορά.
- τη δημιουργία βιβλιοθήκης με τα απαιτούμενα για το πεδίο εφαρμογής συστατικών λογισμικού προσομοίωσης.
- ενσωμάτωση στο μετασχηματισμό μοντέλων συστημάτων σε μοντέλα προσομοίωσης της αντιστοίχησης των τμημάτων του μοντέλου συστήματος σε συστατικά λογισμικού προσομοίωσης.

Με αυτό τον τρόπο το μεγαλύτερο τμήμα που αφορά την προσομοίωση ενσωματώνεται στο πλαίσιο και ο σχεδιασμός συστημάτων απλοποιείται σημαντικά, όντας απαλλαγμένος από

αυτά, ενώ η προσέγγιση εξακολουθεί να κινείται στη λογική της γενικής μεθοδολογίας και τις γενικές προδιαγραφές των υποστηρικτικών πλαισίων. Αυτό καθιστά την εφαρμογή της προσέγγισης σε μεγαλύτερα συστήματα πιο πρακτική, αφού δεν εισάγονται περιορισμοί και πρόσθετες απαιτήσεις, πέρα από αυτές που έχουν τεθεί ήδη για το εξεταζόμενο πεδίο.

Οι δύο εναλλακτικοί τρόποι εφαρμογής εμφανίζουν διαφορετικά χαρακτηριστικά, αναδεικνύοντας τη ποικιλομορφία στις δυνατότητες εφαρμογής της προσέγγισης, οι οποίες θα πρέπει να επιλέγονται, ανάλογα με τις απαιτήσεις. Ωστόσο, κοινά παραμένουν στοιχεία όπως τα βήματα της προτεινόμενης μεθοδολογίας, η χρήση του μετα-μοντέλου προσομοίωσης, ο ορισμός των συνθηκών εκτέλεσης της προσομοίωσης και η αξιοποίηση της δομής των μοντέλων συστήματος.

9.2 Δυνατότητες Μελλοντικών Επεκτάσεων

Η παρούσα διατριβή προτείνει μία μεθοδολογία για την αυτοματοποίηση της προσομοίωσης στην MBSE. Για την υποστήριξη της μεθοδολογίας προτείνεται η υλοποίηση πλαισίων, για τα οποία δίνονται γενικές προδιαγραφές, ενώ παρουσιάστηκαν και δύο περιπτώσεις εφαρμογής της μεθοδολογίας με τη χρήση ενός πλαισίου που υλοποιήθηκε για το περιβάλλον προσομοίωσης DEVS.

Ήδη οι δύο εφαρμογές ανέδειξαν χαρακτηριστικά σχετικά με τη χρήση της προσέγγισης σε συστήματα μικρότερου ή μεγαλύτερου μεγέθους. Η χρήση της προσέγγισης σε περισσότερα και ιδιαίτερα μεγάλου μεγέθους συστήματα, μπορεί να αναδείξει περισσότερα θέματα σχετικά με θέματα εφαρμοσιμότητας και απόδοσης της προσέγγισης.

Παρά το γεγονός ότι η μεθοδολογία και οι προδιαγραφές των υποστηρικτικών πλαισίων είναι ορισμένα με τρόπο γενικό, η υλοποίηση συγκεκριμένων πλαισίων απαιτεί την επιλογή συγκεκριμένου περιβάλλοντος στο οποίο θα εκτελείται η προσομοίωση. Προς το παρόν έχει υλοποιηθεί ένα πλαίσιο για τη μεθοδολογία προσομοίωσης DEVS, ενώ υπάρχουν και άλλες ερευνητικές προσπάθειες για τη χρήση άλλων γλωσσών προσομοίωσης κατά τη σχεδίαση μοντέλων συστημάτων. Η ανάπτυξη και άλλων πλαισίων για άλλες γλώσσες προσομοίωσης είναι μία πρώτη μελλοντική προσθήκη στην παρούσα προσπάθεια.

Με την ύπαρξη πολλαπλών υποστηρικτικών πλαισίων για διαφορετικά περιβάλλοντα προσομοίωσης, θα είχε ενδιαφέρον η μελέτη συγκεκριμένων παραδειγμάτων με περισσότερα του ενός πλαίσια. Αυτό θα εξυπηρετούσε στην ανάδειξη των ιδιαίτερων χαρακτηριστικών κάθε πλαισίου και στην εξαγωγή χρήσιμων, συγκριτικών συμπερασμάτων.

Επιπρόσθετα, η παράλληλη μελέτη των διαφορετικών πλαισίων με μία αφαιρετική οπτική θα μπορούσε να οδηγήσει στην ανάδειξη κοινών χαρακτηριστικών. Με αυτή τη λογική, θα μπορούσαν να υλοποιηθούν πλαίσια τα οποία να μην είναι προσανατολισμένα σε ένα περιβάλλον προσομοίωσης, αλλά σε μία ομάδα περιβαλλόντων ή μία ευρύτερη κατηγορία περιβαλλόντων προσομοίωσης. Αυτό θα καθιστούσε πιο γενικά τα χαρακτηριστικά των υλοποιημένων πλαισίων και λιγότερο εξειδικευμένες τις απαιτούμενες γνώσεις για τον εμπλουτισμό των μοντέλων

συστημάτων με στοιχεία προσομοίωσης. Δηλαδή, ο σχεδιασμός συστήματος αναμένεται να είναι πιο απλός και αποδοτικός, αφού θα στοχεύει σε γενικότερες έννοιες μοντελοποίησης της συμπεριφοράς, παρά σε συγκεκριμένες και λεπτομερείς έννοιες συγκεκριμένης μεθοδολογίας προσομοίωσης, Επίσης, μία τέτοια προσέγγιση θα είχε το πλεονέκτημα της παροχής πολλών επιλογών περιβάλλοντος εκτέλεσης της προσομοίωσης.

Υπάρχουν και επιμέρους ζητήματα τα οποία θα μπορούσαν να αντιμετωπιστούν με τρόπο πιο αποδοτικό, επιφέροντας ουσιαστικές βελτιώσεις σε προσεγγίσεις, όπως η προτεινόμενη στη διατριβή. Μία τέτοια περίπτωση είναι ο ορισμός των μετασχηματισμών QVT, οι οποίοι έχουν καταλυτική σημασία στη δημιουργία και λειτουργία των προτεινόμενων πλαισίων. Ενώ υπάρχει μία πρότυπη γραφική απεικόνιση για τις σχέσεις QVT, αυτή δεν είναι ιδιαίτερα πρακτική και δεν εξυπηρετεί στον ορισμό μεγάλων και σύνθετων μετασχηματισμών. Μία προσέγγιση για γραφικό ορισμό μετασχηματισμών QVT, προσανατολισμένη σε βασικούς τρόπους αναπαράστασης μοντέλων, όπως η UML και το MOF, θα εξυπηρετούσε την ουσιαστική αναβάθμιση των δυνατοτήτων ορισμού και ελέγχου των μετασχηματισμών.

Η προσέγγιση αυτή -του ορισμού μετασχηματισμών με προτυποποιημένο τρόπο, ως μοντέλα- θα διευκόλυνε και την ανάπτυξη μηχανισμών ελέγχου και πιστοποίησης των μετασχηματισμών, ως προς ένα σύνολο χαρακτηριστικών. Ιδιαίτερο ενδιαφέρον θα είχε η εξέταση χρήσης μετασχηματισμών QVT, για το σκοπό αυτό. Το μοντέλο του οριζόμενου μετασχηματισμού θα αποτελούσε την είσοδο του μετασχηματισμού-ελέγχου και η παραγόμενη έξοδος θα ήταν ένα μοντέλο ανάλυσης των ιδιοτήτων του οριζόμενου μετασχηματισμού. Αυτό προϋποθέτει τον ορισμό ενός μετα-μοντέλου ανάλυσης ιδιοτήτων μετασχηματισμών QVT.

Ένα άλλο ζήτημα που σχετίζεται με την προτεινόμενη προσέγγιση είναι η αυτοματοποίηση της επαλήθευσης απαιτήσεων. Με την πρόταση της διατριβής είναι δυνατή η αυτοματοποίηση της παραγωγής εκτιμήσεων της συμπεριφοράς των συστημάτων, μέσω προσομοίωσης. Οι εκτιμήσεις αυτές αξιοποιούνται στα πλαίσια της δεύτερης εφαρμογής, που παρουσιάστηκε στο κεφάλαιο 7, για την επαλήθευση μη λειτουργικών απαιτήσεων στο πεδίο των EIS. Θα ήταν χρήσιμη η αναζήτηση προτάσεων και λύσεων για την προτυποποίηση του ορισμού απαιτήσεων με ποσοτικά χαρακτηριστικά, αλλά και των μεθόδων επαλήθευσης, ανεξάρτητα από το πεδίο εφαρμογής.

Στην περίπτωση της αξιοποίησης βιβλιοθηκών προσομοίωσης και με δεδομένη την προώθηση της διαλειτουργικότητας των προσομοιωτών, που επιτυγχάνεται με προτάσεις όπως η παρούσα διατριβή, ενδιαφέρον θα είχε η διερεύνηση των δυνατοτήτων αυτοματοποιημένης αναζήτησης, εντοπισμού και σύνθεσης συστατικών προσομοίωσης, που μπορούν να είναι διαθέσιμα σε κατανεμημένες δομές ή σε περιβάλλοντα τύπου *cloud*.

Τέλος, θα είχε ενδιαφέρον η εστίαση της ανάλυσης μέσω προσομοίωσης στη Διαχείριση Κύκλου Ζωής Προϊόντων/Συστημάτων (Product Lifecycle Management) (PLM). Σε αυτή την περίπτωση περισσότερες και ευρύτερες πτυχές του εξεταζόμενου συστήματος μέσα στο περιβάλλον λειτουργίας του θα έπρεπε να ληφθούν υπόψη. Θέματα όπως η μέθοδος και το κόστος παραγωγής, η εγκατάσταση, η λειτουργία σε διαφορετικές συνθήκες και η διαδικασία απόσυρσης θα αποτελούσαν τα βασικά αντικείμενα ανάλυσης.

Βιβλιογραφια

1. G.-D. Kapos, A. Tsadimas, C. Kotronis, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "Transforming SysML models to simulation code: The role of QVT," *IEEE Systems Journal*, 2016, submitted for review.
2. G.-D. Kapos, "Enabling system models automated evaluation through cross-concept information utilization," in *9th IEEE International Conference on Research Challenges in Information Science, RCIS 2015, Athens, Greece, May 13-15, 2015*, 2015, pp. 504–509. [Online]. Available: http://dx.doi.org/10.1109/RCIS.2015.7128913
3. M. Nikolaidou, V. Dalakas, L. Mitsi, G.-D. Kapos, and D. Anagnostopoulos, "A SysML profile for classical DEVS simulators," in *Proceedings of the Third International Conference on Software Engineering Advances (ICSEA 2008)*. Malta: IEEE Computer Society, October 2008, pp. 445–450.
4. G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "An integrated framework for automated simulation of SysML models using DEVS," *SCS Transactions on Computer Simulation*, vol. 90, no. 6, pp. 717–744, June 2014.
5. G.-D. Kapos, V. Dalakas, A. Tsadimas, M. Nikolaidou, and D. Anagnostopoulos, "Model-based System Engineering using SysML: Deriving Executable Simulation Models with QVT," in *Proc. Eighth Annual Int. Systems Conf. (SysCon)*, 2014, pp. 531–538.
6. G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications*. Igi Global, 2013, ch. 10. An Integrated Framework to Simulate SysML Models Using DEVS Simulators.
7. A. Tsadimas, G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "Integrating Simulation Capabilities into SysML for Enterprise Information System Design," in *Proc. Eighth Int. Conf. on System of Systems Engineering (SoSE) 2014*, June 2014, pp. 0–0.
8. M. Nikolaidou, G.-D. Kapos, V. Dalakas, and D. Anagnostopoulos, "Basic Guidelines for Simulating SysML Models: An Experience Report," in *Proc. Seventh Int. Conf. on System of Systems Engineering (SoSE) 2012*, July 2012, pp. 95–100.
9. C. Haskins, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, v. 3*, International Council on Systems Engineering, June 2006, iNCOSE-TP-2003-002-03.
10. A. Law, *Simulation modeling and analysis*, 4th ed., ser. McGraw-Hill series in industrial engineering and management science. McGraw-Hill, 2006.
11. O. M. G. Inc, *Systems Modeling Language (SYSML) Specification, Version 1.3*, Std., June 2012, http://www.omg.org/spec/SysML/1.3/PDF.
12. M. Jamshidi, "System-of-systems engineering-a definition," *IEEE SMC*, pp. 10–12, 2005.
13. OMG, *OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1*, Std., August 2011, http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/.
14. A. Sage, *Systems engineering*, ser. A Wiley-Interscience publication. New York, NY [u.a.]: Wiley, 1992. [Online]. Available: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+110169247&sourceid=fbw_bibsonomy

15. D. Buede, *The Engineering Design of Systems: Models and Methods*, ser. Wiley Series in Systems Engineering and Management. Wiley, 2011. [Online]. Available: https://books.google.gr/books?id=4gHqQIaVncMC
16. S. Sendall and W. Kozaczynski, "Model transformation: the heart and soul of model-driven software development," *Software, IEEE*, vol. 20, no. 5, pp. 42 – 45, September 2003.
17. D. Li, X. Li, and V. Stolz, "QVT-based model transformation using XSLT," *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 1, pp. 1–8, Jan. 2011.
18. M. Alanen, I. Porres, T. Centre, and C. Science, "A relation between context-free grammars and meta object facility metamodels," Tech. Rep., 2003.
19. P. Stevens, "Bidirectional model transformations in QVT: Semantic issues and open questions," *Software and Systems Modeling*, vol. 9, no. 1, pp. 7–20, 2010.
20. OMG, "Model Driven Architecture. Version 1.0.1," *Available online via http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf*, June 2003.
21. OMG, "Meta Object Facility (MOF) Core Specification, Version 2.4.1," no. June, pp. 1–88, 2013. [Online]. Available: http://www.omg.org/spec/MOF/2.4.1/PDF/
22. OMG, "Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, Version 1.1," no. January, pp. 1–246, 2011. [Online]. Available: http://www.omg.org/spec/QVT/1.1/PDF/
23. World Wide Web Consortium (W3C), "Extensible Stylesheet Language Transformations (XSLT)," 2007. [Online]. Available: http://www.w3.org/TR/xslt20
24. ——, "XQuery 3.0: An XML Query Language," 2014. [Online]. Available: http://www.w3.org/TR/xquery-30/
25. OMG, "OMG Object Constraint Language (OCL), Version 2.3.1," February 2012. [Online]. Available: http://www.omg.org/spec/OCL/2.3.1/PDF/
26. D. Calegari and N. Szasz, "Verification of model transformations," *Electron. Notes Theor. Comput. Sci.*, vol. 292, pp. 5–25, Mar. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.entcs.2013.02.002
27. J. Banks, J. Carson, B. Nelson, and D. Nicol, *Discrete-event system simulation*, 5th ed. Prentice Hall, 2010.
28. B. P. Zeigler, H. Praehofer, and T. Kim, *Theory of Modeling and Simulation*, 2nd ed. Academic Press, 2000.
29. B. P. Zeigler, Y. Moon, D. Kim, and J. G. Kim, "DEVS-C++: A high performance modelling and simulation environment." in *HICSS (1)*, 1996, pp. 350–359. [Online]. Available: http://dblp.uni-trier.de/db/conf/hicss/hicss1996-1.html#ZeiglerMKK96
30. B. P. Zeigler and H. S. Sarjoughian, *Introduction to DEVS Modeling and Simulation with JAVA. DEVSJAVA Manual*, 2003. [Online]. Available: www.acims.arizona.edu/PUBLICATIONS/publications.shtml
31. G. A. Wainer and N. Giambiasi, *Timed Cell-DEVS: modelling and simulation of cell spaces*. Springer-Verlag: H. Sarjoughian, F. Cellier Eds., 2001, ch. 10. [Online]. Available: http://cell-devs.sce.carleton.ca/publications/2001/WG01b
32. M. Zhang, B. P. Zeigler, and P. Hammonds, "DEVS/RMI-an auto-adaptive and reconfigurable distributed simulation environment for engineering studies," *International Test and Evaluation Association Journal*, vol. 27, no. 1, pp. 49–60, 2005.
33. N. Meseth, P. Kirchhof, and T. Witte, "XML-based DEVS modeling and interpretation," in *SpringSim '09: Proceedings of the 2009 Spring Simulation Multiconference*. San Diego, CA, USA: Society for Computer Simulation International, 2009, pp. 1–9.
34. R. Peak, R. Burkhart, S. Friedenthal, M. Wilson, M. Bajaj, and I. Kim, "Simulation-based design

using sysml part 1: A parametrics primer." San Diego, CA, USA: INCOSE Intl. Symposium, 2009.

35. R. Peak, C. J. Paredis, and D. R. Tamburini, "The composable object (COB) knowledge representation: Enabling advanced collaborative engineering environments (CEEs), COB requirements & objectives (v1.0)," Georgia Institute of Technology, Atlanta, GA, Technical Report, Oct. 2005.
36. D. R. Tamburini, "Defining executable design & simulation models using SysML," *Available online via http://www.pslm.gatech.edu/topics/sysml/*, March 2006.
37. C. J. J. Paredis and T. Johnson, "Using OMG's SysML to support simulation," in *WSC '08: Proceedings of the 40th Conference on Winter Simulation*. Winter Simulation Conference, 2008, pp. 2350–2352.
38. A. A. Kerzhner, J. M. Jobe, and C. J. J. Paredis, "A formal framework for capturing knowledge to transform structural models into analysis models," *Journal of Simulation*, vol. 5, no. 3, pp. 202–216, 2011.
39. OMG, *SysML and Modelica Integration*, Dec. 2008. [Online]. Available: http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-modelica:sysml_and_modelica_integration
40. L. McGinnis and V. Ustun, "A simple example of SysML-driven simulation," in *Winter Simulation Conference (WSC), Proceedings of the 2009*. IEEE, 2009, pp. 1703–1710.
41. O. Batarseh and L. F. McGinnis, "System modeling in sysml and system analysis in arena," in *Proceedings of the Winter Simulation Conference*, ser. WSC '12. Winter Simulation Conference, 2012, pp. 258:1–258:12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2429759.2430107
42. R. Wang and C. Dagli, "An executable system architecture approach to discrete events system modeling using SysML in conjunction with colored petri nets," in *IEEE Systems Conference 2008*. Montreal: IEEE Computer Press, April 2008, pp. 1–8.
43. M. Bajaj, D. Zwemer, R. Peak, A. Phung, A. Scott, and M. Wilson, "Slim: collaborative model-based systems engineering workspace for next-generation complex systems," in *Aerospace Conference, 2011 IEEE*, 2011, pp. 1–15.
44. "ModelCenter Integrate," http://www.phoenix-int.com/modelcenter/integrate.php.
45. D. Knorreck, L. Apvrille, and P. de Saqui-Sannes, "Tepe: A sysml language for time-constrained property modeling and formal verification," *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 1, pp. 1–8, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1921532.1921556
46. G. Behrmann, A. David, K. G. Larsen, J. Hakansson, P. Petterson, W. Yi, and M. Hendriks, "Uppaal 4.0," in *Proceedings of the 3rd International Conference on the Quantitative Evaluation of Systems*, ser. QEST '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 125–126. [Online]. Available: http://dx.doi.org/10.1109/QEST.2006.59
47. G. Pedroza, L. Apvrille, and D. Knorreck, "Avatar: A sysml environment for the formal verification of safety and security properties," in *New Technologies of Distributed Systems (NOTERE), 2011 11th Annual International Conference on*, May 2011, pp. 1–10.
48. J. Martín, S. Mittal, M. López-Peña, and J. De la Cruz, "A W3C XML schema for DEVS scenarios," in *Proceedings of the 2007 spring simulation multiconference-Volume 2*. Society for Computer Simulation International, 2007, pp. 279–286.
49. J. L. Risco-Martín, J. M. De La Cruz, S. Mittal, and B. P. Zeigler, "eUDEVS: Executable UML with DEVS theory of modeling and simulation," *Simulation*, vol. 85, no. 11-12, pp. 750–777, 2009.
50. M. Hosking and F. Sahin, "An XML based system of systems discrete event simulation communications framework," in *SpringSim '09: Proceedings of the 2009 Spring Simulation Multiconference*. San Diego, CA, USA: Society for Computer Simulation International, 2009, pp. 1–9.
51. S. Mittal, J. L. Risco-Martín, and B. P. Zeigler, "DEVSML: Automating DEVS execution over SOA

towards transparent simulators," in *DEVS Symposium, Spring Simulation Multiconference*. ACIMS Publications, March 2007, pp. 287–295.

52. M. H. Hwang and B. Zeigler, "Reachability graph of finite and deterministic DEVS networks," *Automation Science and Engineering, IEEE Transactions on*, vol. 6, no. 3, pp. 468–478, July 2009.
53. D. Anagnostopoulos, V. Dalakas, G.-D. Kapos, and M. Nikolaidou, "An RT-UML Model for Building Faster-Than-Real-Time Simulators," in *Proceedings of Eurosim04*, Paris, September 2004.
54. S. Hämäläinen, H. Sanneck, and C. Sartori, *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Wiley, 2011. [Online]. Available: http://books.google.gr/books?id=vM2gUROYpNIC
55. R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
56. J. Cabot, R. Clarisó, E. Guerra, and J. de Lara, "Verification and validation of declarative model-to-model transformations through invariants," *J. Syst. Softw.*, vol. 83, no. 2, pp. 283–302, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2009.08.012
57. A. Schürr, "Specification of graph translators with triple graph grammars," in *in Proc. of the 20th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG '94), Herrsching (D.* Springer, 1995.
58. E. Kindler and R. Wagner, "Triple graph grammars: Concepts, extensions, implementations, and application scenarios," Software Engineering Group, Department of Computer Science, University of Paderborn, Tech. Rep., 2007.
59. S. Hildebrandt, L. Lambers, H. Giese, J. Rieke, J. Greenyer, W. Schäfer, M. Lauder, A. Anjorin, and A. Schürr, "A survey of triple graph grammar tools," in *Second International Workshop on Bidirectional Transformations (BX 2013)*, vol. 57. EC-EASST, 2013, pp. 1–18. [Online]. Available: http://jgreen.de/wp-content/documents/2013/bx13.pdf
60. F. Jouault and I. Kurtev, "On the architectural alignment of atl and qvt," in *Proceedings of the 2006 ACM Symposium on Applied Computing*, ser. SAC '06. New York, NY, USA: ACM, 2006, pp. 1188–1195. [Online]. Available: http://doi.acm.org/10.1145/1141277.1141561
61. A. Concepcion and B. Zeigler, "Devs formalism: a framework for hierarchical model development," *Software Engineering, IEEE Transactions on*, vol. 14, no. 2, pp. 228–241, Feb 1988.
62. "medini QVT," ikv++ technologies ag, 2013, http://projects.ikv.de/qvt.
63. "EditiX," http://www.editix.com/.
64. A. Tsadimas, M. Nikolaidou, and D. Anagnostopoulos, "Handling non-functional requirements in information system architecture design," in *ICSEA '09*, 2009, pp. 59–64.
65. ——, "Evaluating software architecture in a model-based approach for enterprise information system design," in *SHARK '10*, 2010, pp. 72–79.
66. ——, "Extending sysml to explore non-functional requirements: the case of information system design," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12, 2012, pp. 1057–1062. [Online]. Available: http://doi.acm.org/10.1145/2231936.2231941
67. A. Tsadimas, G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "Simulating simulation-agnostic SysML models for enterprise information systems via DEVS," *Simulation Modelling Practice and Theory*, vol. 66, pp. 243 – 259, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X16300259
68. C. Kotronis, A. Tsadimas, G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "Simulating SysML Transportation Models," in *IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC2016)*, Budapest, October 2016, (accepted to be presented).
69. M. Nikolaidou, G.-D. Kapos, A. Tsadimas, V. Dalakas, and D. Anagnostopoulos, "Challenges in SysML

model simulation," *Advances in Computer Science: an International Journal (ACSIJ)*, 2016, accepted.

70. ——, "Simulating sysml models: Overview and challenges," in *10th System of Systems Engineering Conference, SoSE 2015, San Antonio, TX, USA, May 17-20, 2015*, 2015, pp. 328–333. [Online]. Available: http://dx.doi.org/10.1109/SYSOSE.2015.7151961

Δημοσιευσεις

Σε αυτή την ενότητα γίνεται μία συνοπτική παρουσίαση των δημοσιεύσεων που έχουν γίνει στα πλαίσια της έρευνας που σχετίζεται με την παρούσα διατριβή. Για κάθε δημοσίευση δίνεται ο τίτλος, το έτος δημοσίευσης και τα επιμέρους ζητήματα στα οποία δίνεται έμφαση.

Δημοσιεύσεις σε Περιοδικές Εκδόσεις

- "An Integrated Framework for Automated Simulation of SysML Models using DEVS", 2014 [4]: Με την άνεση χώρου ενός άρθρου σε περιοδικό, γίνεται μία αναλυτική παρουσίαση του πλαισίου για αυτοματοποίηση της προσομοίωσης μοντέλων συστημάτων με DEVS. Έμφαση δίνεται στη μεθοδολογία, το μετα-μοντέλο για το DEVS και το μετασχηματισμό μοντέλων συστημάτων σε μοντέλα DEVS με QVT.

- "Simulating Simulation-Agnostic SysML Models for Enterprise Information Systems via DEVS", 2016 [67]: Παρουσιάζεται αναλυτικά η δυνατότητα, που παρέχει η προτεινόμενη προσέγγιση για αυτοματοποίηση της προσομοίωσης μοντέλων συστημάτων (SysML), χωρίς τον εμπλουτισμό τους με στοιχεία συμπεριφοράς για την προσομοίωση. Εξετάζεται η διαδικασία διαμόρφωσης του προτεινόμενου πλαισίου και αναλύονται οι ρόλοι των εμπλεκόμενων σε αυτή. Η παρουσίαση γίνεται στο πεδίο εφαρμογών των EIS με την αξιοποίηση αντίστοιχων βιβλιοθηκών συστατικών λογισμικού προσομοίωσης. Έμφαση δίνεται στην ενσωμάτωση των αποτελεσμάτων στο μοντέλο συστήματος και στις δυνατότητες αυτοματοποιημένης επαλήθευσης μη λειτουργικών απαιτήσεων των EIS με την αξιοποίησή τους.

- "Transforming SysML Models to Simulation Code: The Role of QVT", υπεβλήθη για κρίση το 2016 [1]: Παρουσιάζεται η προτεινόμενη μεθοδολογία στη γενική της μορφή και αναδεικνύεται ο κεντρικός ρόλος της QVT στην υλοποίηση της παραγωγής κώδικα προσομοίωσης, ως μία μετάβαση από το μετα-μοντέλο της SysML στο μετα-μοντέλο προσομοίωσης. Παρουσιάζονται τα αποτελέσματα της μελέτης της εφαρμογής της γενικής προσέγγισης σε δύο διαφορετικά πεδία εφαρμογής με το πλαίσιο που αναπτύχθηκε για τη μεθοδολογία προσομοίωσης DEVS. Εδώ προτείνονται και σχετικές οδηγίες για την ανάπτυξη μετασχηματισμών QVT.

- "Challenges in SysML Model Simulation", έγινε δεκτό για δημοσίευση το 2016 [69]: Σε αυτή τη δημοσίευση αναλύονται τα χαρακτηριστικά των κυριότερων προσεγγίσεων για

την προσομοίωση μοντέλων SysML. Περιγράφονται οι ομοιότητες και διαφορές μεταξύ των προσεγγίσεων και αναγνωρίζονται οι προκλήσεις της πλήρως αυτοματοποιημένης προσομοίωσης μοντέλων SysML.

Δημοσίευση σε Κεφάλαιο Βιβλίου

- "An Integrated Framework to Simulate SysML Models Using DEVS Simulators", 2013 [6]: Με την άνεση χώρου ενός κεφαλαίου σε βιβλίο, παρουσιάζονται αρκετές από τις πρακτικές πτυχές της εφαρμογής του πλαισίου αυτοματοποιημένης προσομοίωσης μοντέλων συστημάτων με DEVS. Το αντικείμενο είναι ένα σύστημα μάχης, αποτελούμενο από φιλικές και εχθρικές δυνάμεις.

Δημοσιεύσεις σε Συνέδρια

- "A SysML Profile for Classical DEVS Simulators", 2008 [3]: Σε αυτή τη φάση της έρευνας αναζητούνταν κατάλληλος τρόπος για την αναπαράσταση μοντέλων DEVS με όρους SysML. Η δημοσίευση παρουσιάζει το προφίλ SysML για τη μεθοδολογία προσομοίωσης DEVS, που αποτελεί τμήμα του συνολικού πλαισίου που αναπτύχθηκε για την υποστήριξη της προτεινόμενης μεθοδολογίας με προσομοίωση σε DEVS.

- "Basic Guidelines for Simulating SysML Models: An Experience Report", 2012 [8]: Έχοντας ήδη πειραματιστεί επαρκώς με την αυτοματοποιημένη παραγωγή κώδικα προσομοίωσης από εμπλουτισμένα (σύμφωνα με το προφίλ SysML για DEVS) μοντέλα συστήματος, σε αυτή τη δημοσίευση προτείνονται κάποιες βασικές οδηγίες για την προσέγγιση της προσομοίωσης μοντέλων SysML. Έτσι, προδιαγράφεται αρχικά και η προτεινόμενη στην παρούσα διατριβή μεθοδολογία.

- "Model-based System Engineering using SysML: Deriving Executable Simulation Models with QVT", 2014 [5]: Εδώ παρουσιάζεται το πλαίσιο, με έμφαση στο μετασχηματισμό μοντέλων SysML σε μοντέλα DEVS. Επισημαίνονται ο ρόλος και οι δυνατότητες της γλώσσας μετασχηματισμών QVT, που είναι καθοριστικά στην προσεγγιση συνολικά. Η δημοσίευση αυτή διακρίθηκε με "**Best Student Paper Award - Honorable Mention**".

- "Integrating Simulation Capabilities into SysML for Enterprise Information System Design", 2014 [7]: Παρουσιάζεται η προσθήκη δυνατοτήτων προσομοίωσης σε μοντέλα SysML, στο πεδίο εφαρμογών των EIS, με χρήση βιβλιοθηκών συστατικών λογισμικού προσομοίωσης. Ιδιαίτερη έμφαση δίνεται και στα θέματα της εισαγωγής των αποτελεσμάτων προσομοίωσης στο μοντέλο συστήματος και την επαλήθευση ποσοτικά προσδιορισμένων μη λειτουργικών απαιτήσεων.

- "Enabling System Models Automated Evaluation through Cross-Concept Information Utili-

zation", 2015 [2]: Γίνεται μία γενική παρουσίαση της προσέγγισης, αναδεικνύοντας τη συνεισφορά της διατριβής και τα πλεονεκτήματα που μπορούν να προκύψουν από την κατάλληλη αξιοποίηση της πληροφορίας, που είναι διαθέσιμη στα μοντέλα συστήματος για την αυτοματοποίηση της εκτίμησης της συμπεριφοράς τους.

- "Simulating SysML models: Overview and challenges", 2015 [70]: Σε αυτή τη δημοσίευση γίνεται μία επισκόπηση των κυριότερων προσεγγίσεων, που έχουν προταθεί για την προσομοίωση μοντέλων SysML. Οι προσεγγίσεις συγκρίνονται ως προς συγκεκριμένα κριτήρια και αναγνωρίζονται οι προκλήσεις της συγκεκριμένης ερευνητικής περιοχής.

- "Simulating SysML Transportation Models", εγκρίθηκε για δημοσίευση το 2016 [68]: Παρουσιάζεται η εφαρμογή της προσέγγισης στο πεδίο των συστημάτων μαζικής μεταφοράς σταθερής τροχιάς. Σύμφωνα με τις προδιαγραφές, αναπτύχθηκαν τα στοιχεία που απαιτούνται για το πεδίο εφαρμογής: ένας μετασχηματισμός μοντέλων συστημάτων μεταφοράς σε μοντέλα προσομοίωσης και βιβλιοθήκη με σχετικά συστατικά προσομοίωσης. Μοντέλα συστημάτων μεταφοράς μπορούν να ορίζονται σε SysML, αξιοποιώντας τις επεκτάσεις, που έχουν οριστεί σε ένα σχετικό προτεινόμενο προφίλ. Έγινε δοκιμαστική μοντελοποίηση και σχετικές μετρήσεις προσομοίωσης των γραμμών 1, 2 και 3 του αστικού δικτύου σταθερής τροχιάς της Αθήνας.

Βιογραφικο

Ο Γεώργιος-Δημήτριος Κάπος είναι αριστούχος πτυχιούχος του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού Καποδιστριακού Πανεπιστημίου Αθηνών και κάτοχος μεταπτυχιακού τίτλου με άριστα στα Προηγμένα Πληροφοριακά Συστήματα από το ίδιο Τμήμα.

Από το 2006 έως σήμερα εργάζεται σαν αναλυτής/προγραμματιστής στη Διεύθυνση Πληροφορικής του Ταμείου Παρακαταθηκών και Δανείων. Παράλληλα, από το 2012 συμμετέχει στη διδασκαλία μαθημάτων του Μεταπτυχιακού Προγράμματος Σπουδών του Τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου Αθηνών, στα πεδία της αρχιτεκτονικής πληροφοριακών συστημάτων, της ανάπτυξης εφαρμογών και της Model-Driven Architecture (MDA). Επίσης, από το 2015 παρέχει υπηρεσίες συμβουλευτικής στο ευρείας κλίμακας ερευνητικό πρόγραμμα DEMO, για την πρότυπη κατασκευή λειτουργικών εργοστασίων παραγωγής ενέργειας με πυρηνική σύντηξη και, συγκεκριμένα, στην απόπειρα υιοθέτησης της βασισμένης σε μοντέλα ανάπτυξης συστημάτων (MBSE) κατά την conceptual design activity.

Από το 2004 έως το 2006 εργάστηκε σαν αναλυτής/προγραμματιστής στη Διεύθυνση Πληροφορικής του Οργανισμού Ελληνικών Γεωργικών Ασφαλίσεων (ΕΛΓΑ). Από το 1999 έως το 2004 συμμετείχε σε εθνικά και κοινοτικά ερευνητικά προγράμματα, με θέματα όπως η ομογενοποίηση κατανεμημένων βάσεων στατιστικών δεδομένων, η ανάπτυξη ψηφιακών υπηρεσιών και περιεχομένου τηλε-εκπαίδευσης και η ανάπτυξη/βελτίωση δικτυακών υπηρεσιών.

Από το 1999 έως σήμερα έχει συμμετάσχει στη συγγραφή περισσότερων από 20 δημοσιεύσεων σε συνέδρια, περιοδικά και κεφάλαια βιβλίων, στο χώρο της πληροφορικής, των κατανεμημένων συστημάτων, της προσομοίωσης και της ανάπτυξης συστημάτων.

Ευρετηριο Σχηματων

Ευρετηριο Πινακων

Παραρτήματα

Παραρτημα Α΄

Ορισμοι Μετα-μοντελων και Μετασχηματισμοι

Α΄.1 Μετασχηματισμός Μοντέλων SysML σε Μοντέλα DEVS με QVT

Ο μετασχηματισμός μοντέλων συστήματος SysML σε μοντέλα προσομοίωσης DEVS είναι σημαντικό στοιχείο του προτεινόμενου πλαισίου DEVSys (κεφάλαιο 5). Για την παροχή μίας οπτικής πιο κοντά στην υλοποίηση, στο listing Α΄.1 δίνεται το μέρος του μετασχηματισμού QVT, που αναγνωρίζει τις μεταβάσεις καταστάσεων των SysML blocks και δημιουργεί τα αντίστοιχα χαρακτηριστικά συμπεριφοράς των συστατικών λογισμικού προσομοίωσης DEVS.

Στην QVT-R, ένας μετασχηματισμός ανάμεσα σε δύο πεδία ορίζεται μέσω ενός συνόλου σχέσεων, που αντιστοιχίζουν οντότητες του ενός πεδίου σε οντότητες του άλλου. Ο μετασχηματισμός ξεκινάει από την εφαρμογή των ανώτερων σχέσεων (top relations), οι οποίες λογικά θα αναφέρονται σε άλλες σχέσεις. Τα δύο πεδία (*devsSysml* και *devs*) δηλώνονται και χαρακτηρίζονται κατάλληλα. Το πεδίο *devsSysml* χαρακτηρίζεται ως *checkonly*, καθώς είναι το πεδίο αφετηρίας, το οποίο ελέγχεται αν πληροί τις προϋποθέσεις εφαρμογής των σχέσεων, όπως είναι (χωρίς να επέλθουν τροποποιήσεις ώστε να τις ικανοποιεί). Από την άλλη πλευρά, το πεδίο *devs* χαρακτηρίζεται ως *enforce*, αφού πρόκειται για το πεδίο προορισμού, το οποίο πρέπει να δημιουργηθεί σύμφωνα με τις συνθήκες των σχέσεων. Σε αυτή την περίπτωση η σχέση με τις συνθήκες της επιβάλλονται στο μοντέλο του πεδίου προορισμού.

Αναφορικά με το μετασχηματισμό των μεταβάσεων κατάστασης από τα SMDs των SysML, χρησιμοποιήθηκαν έξι (6) σχέσεις. Η σχέση *Transitions2ConditionalFunctions* δημιουργεί την *DEVS Atomic Internal Transition Function*, ενώ η *StringSequence2StateVarUpdate* χειρίζεται τις κατάλληλες προδιαγραφές μεταβολών για μεταβολές κατάστασης. Η σχέση *Transitions2ConditionalOutputFunctions* δημιουργεί την *DEVS Atomic Output Function* με τη βοήθεια της *StringSequence2PortOutputs*. Η σχέση *Transitions2ConditionalTimeAdvances* δημιουργεί την *DEVS Atomic Time Advance Function* για τις καταστάσεις με πεπερασμένο χρόνο παραμονής, ενώ η *NoTransitions2InfiniteConditionalTimeAdvances* ασχολείται με τις καταστάσεις εκείνες στις οποίες το σύστημα παραμένει επ' άπειρο, χωρίς εξωτερική παρέμβαση.

Στη συνέχεια, παρουσιάζεται σε μεγαλύτερη λεπτομέρεια η σχέση *Transitions2ConditionalOut-*

putFunctions. Συγκεκριμένα, αυτή η σχέση αναγνωρίζει τις *DEVS Internal Transitions*, τις καταστάσεις αφετηρίας τους (*Origin*) και τα *DEVS Outputs*, και τα ταιριάζει με τα αντίστοιχα στοιχεία *CONDITIONAL_OUTPUT_FUNCTION*, *SEND* και *VALUE*. Για κάθε πεδίο, δηλώνονται τα σημαντικά στοιχεία: η *transition* με το *source* και το *effect* της για το πεδίο *devsSysml* και η *CONDITIONAL_OUTPUT_FUNCTION* με τη *STATE_CONDITION* της για το πεδίο *devsXml*. Για την αντιστοίχιση τιμών χρησιμοποιούνται μεταβλητές, π.χ. η μεταβλητή *ns* χρησιμοποιείται για την απόδοση τιμής στο *CONDITIONAL_OUTPUT_FUNCTION.STATE_CONDITION.text* από το *transition.source.name*. Η συνθήκη *when* μίας σχέσης ορίζει τις προϋποθέσεις, που πρέπει να πληρούνται για να εφαρμοστεί η σχέση. Η συνθήκη αυτή μπορεί να περιέχει κλήση άλλων σχέσεων ή εκφράσεις OCL. Σε αυτό το παράδειγμα, το *tr* (η *transition*) πρέπει να έχει το στερεότυπο *DEVS State Transition*. Αυτό σημαίνει ότι το στοιχείο *CONDITIONAL_OUTPUT_FUNCTION* θα δημιουργηθεί μόνο για κάθε *transition*, η οποία έχει το στερεότυπο *DEVS State Transition*. Η συνθήκη *where* μιας σχέσης ορίζει τις εκφράσεις (πιθανά με άλλες σχέσεις), οι οποίες θα έπρεπε να εφαρμοστούν, εφόσον εφαρμοστεί αυτή η σχέση και δημιουργηθούν τα στοιχεία του πεδίου προορισμού. Η σχέση *StringSequence2Output* αναλύει την εντολή εξόδου (*output command*) *y* και δημιουργεί τα κατάλληλα στοιχεία κάτω από το στοιχείο state condition *sc*. Σε αυτή την περίπτωση, μία έκφραση OCL έχει σαν αποτέλεσμα τις πολλαπλές κλήσεις μίας σχέσης. Το πρώτο στοιχείο της sequence συμβολοσειρών *b* διαχωρίζεται σε όλες τις επιμέρους συμβολοσειρές της, που χωρίζονται με το διακριτικό *cmd:*, δημιουργώντας μία λίστα με όλες τις εντολές. Από αυτές, επιλέγονται μόνο αυτές που ξεκινάνε από *output*, δηλαδή μόνο οι εντολές εξόδου.

Listing Α΄.1: Μετασχηματισμός των DEVS Conditional Functions με QVT

```
transformation devsSysml2devsMM(devsSysml:uml, devs:Devs) {
  top relation SysmlModel2DevsModel {
    checkonly domain devsSysml ms : uml::Model {};
    enforce domain devs mx : Devs::MODEL {};
    where {
      CoupledBlock2DevsCoupled(ms,mx);
      AtomicBlock2DevsAtomic(ms,mx);
    }
  }
  ...
  relation Transitions2ConditionalFunctions {
    ns, nt : String;
    b : Sequence(String);
    checkonly domain devsSysml r : uml::Region {
      transition = tr : uml::Transition {
        source = s : uml::Vertex { name = ns },
        target = t : uml::Vertex { name = nt },
        effect = e : uml::FunctionBehavior { _body = b }
      }
    };
```

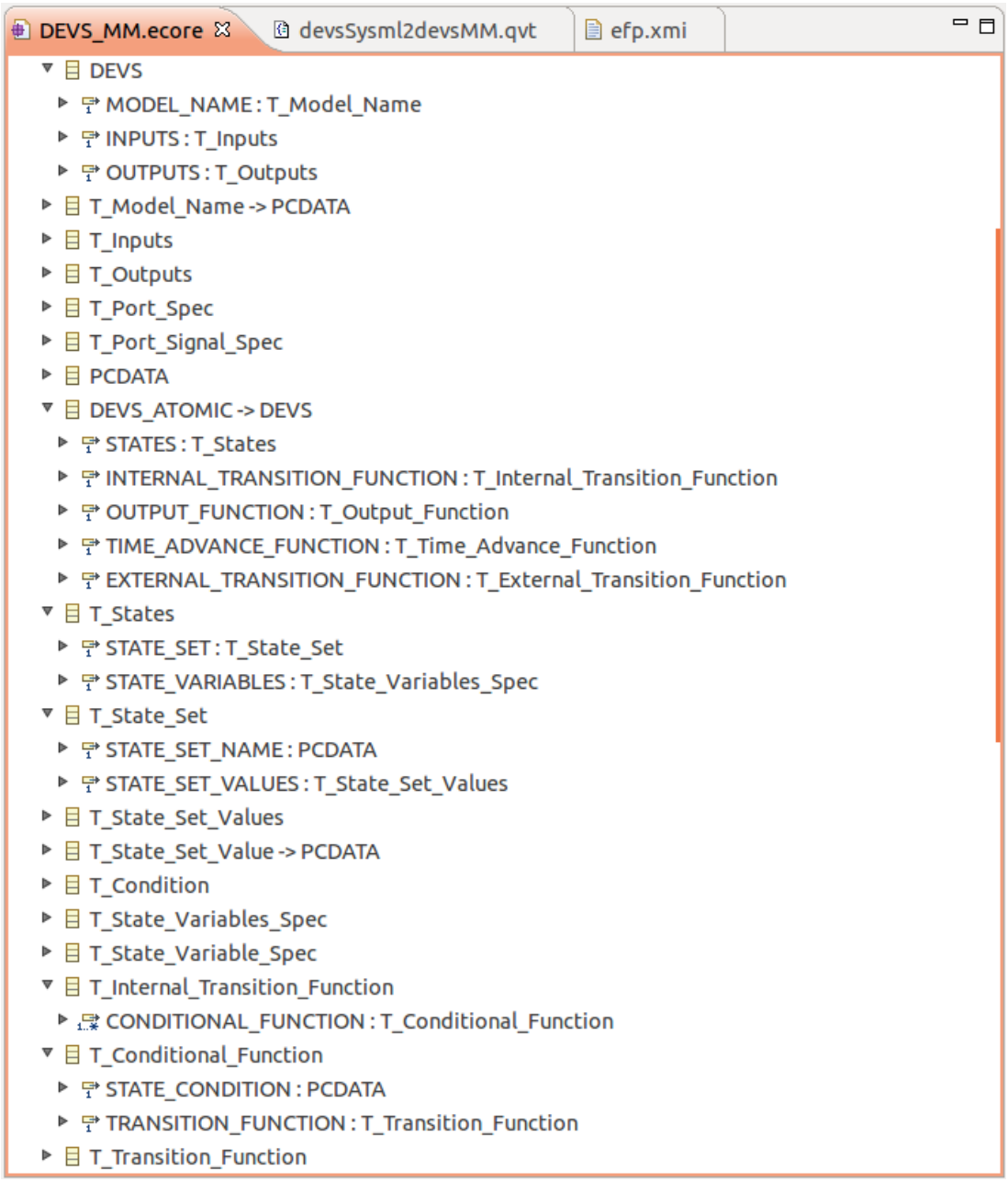
```
  enforce domain devs itf : Devs::T_Internal_Transition_Function {
    CONDITIONAL_FUNCTION = cf : Devs::T_Conditional_Function {
      STATE_CONDITION = sc : Devs::PCDATA { text = ns },
      TRANSITION_FUNCTION = tf : Devs::T_Transition_Function {
        NEW_STATE = nst : Devs::PCDATA { text = nt },
        STATE_VARIABLE_UPDATES =
          svus : Devs::T_State_Variable_Updates {}
      }
    }
  };
  when {
    not (tr.source.name = '');
    tr.getAppliedStereotype('DEVS_SysML::DEVS State Transition')->notEmpty()
      ;
  }
  where {
    b->first().split('cmd:')->select(x|x.startsWith('stateVarUpdate'))->
      forAll(y|StringSequence2StateVarUpdate(y,svus));
  }
}
relation StringSequence2StateVarUpdate {
  checkonly domain devsSysml b : String { };
  enforce domain devs svus : Devs::T_State_Variable_Updates {
    STATE_VARIABLE_UPDATE = svu : Devs::T_State_Variable_Update {
      name = b.substring(b.indexOf('(')+2,b.indexOf(',')),
      VALUE = v : Devs::T_Value {}
    }
  };
  where {
    value(b.substring(b.indexOf(',')+2,b.lastIndexOf(')')),v);
  }
}
relation Transitions2ConditionalOutputFunctions {
  ns : String;
  b : Sequence(String);
  checkonly domain devsSysml r : uml::Region {
    transition = tr : uml::Transition {
      source = s : uml::Vertex { name = ns },
      effect = e : uml::FunctionBehavior { _body = b }
    }
  };
  enforce domain devs of : Devs::T_Output_Function {
    CONDITIONAL_OUTPUT_FUNCTION = cof : Devs::T_Conditional_Output_Function
      {
      state = ns,
```

```
      PORT_OUTPUTS = po : Devs::T_Port_Outputs {}
    }
  };
  when {
    tr.getAppliedStereotype('DEVS_SysML::DEVS State Transition')->notEmpty()
      ;
    b->first().indexOf('cmd:output') > -1;
  }
  where {
    b->first().split('cmd:')->select(x|x.startsWith('output'))->
      forAll(y|StringSequence2PortOutputs(y,po));
  }
}
relation StringSequence2PortOutputs {
  checkonly domain devsSysml b : String { };
  enforce domain devs po : Devs::T_Port_Outputs {
    SEND = s : Devs::T_Send {
      port = b.substring(b.indexOf('(')+2,b.indexOf(',')),
      VALUE = v : Devs::T_Value {}
    }
  };
  where { value(b.substring(b.indexOf(',')+2,b.lastIndexOf(')')),v); }
}
relation Transitions2ConditionalTimeAdvances {
  ns, tav : String;
  checkonly domain devsSysml r : uml::Region {
    transition = tr : uml::Transition {
      source = s : uml::Vertex { name = ns },
      guard = g : uml::Constraint {
        specification = sp : uml::DurationInterval {
          min = m : uml::Duration {
            expr = e : uml::LiteralString { value = tav }
          }
        }
      }
    }
  };
  enforce domain devs taf : Devs::T_Time_Advance_Function {
    CONDITIONAL_TIME_ADVANCE = cta : Devs::T_Conditional_Time_Advance {
      STATE_CONDITION = sc : Devs::PCDATA { text = ns },
      TIME_ADVANCE = ta : Devs::T_Time_Advance {
        VALUE = v : Devs::T_Value {}
      }
    }
  };
```

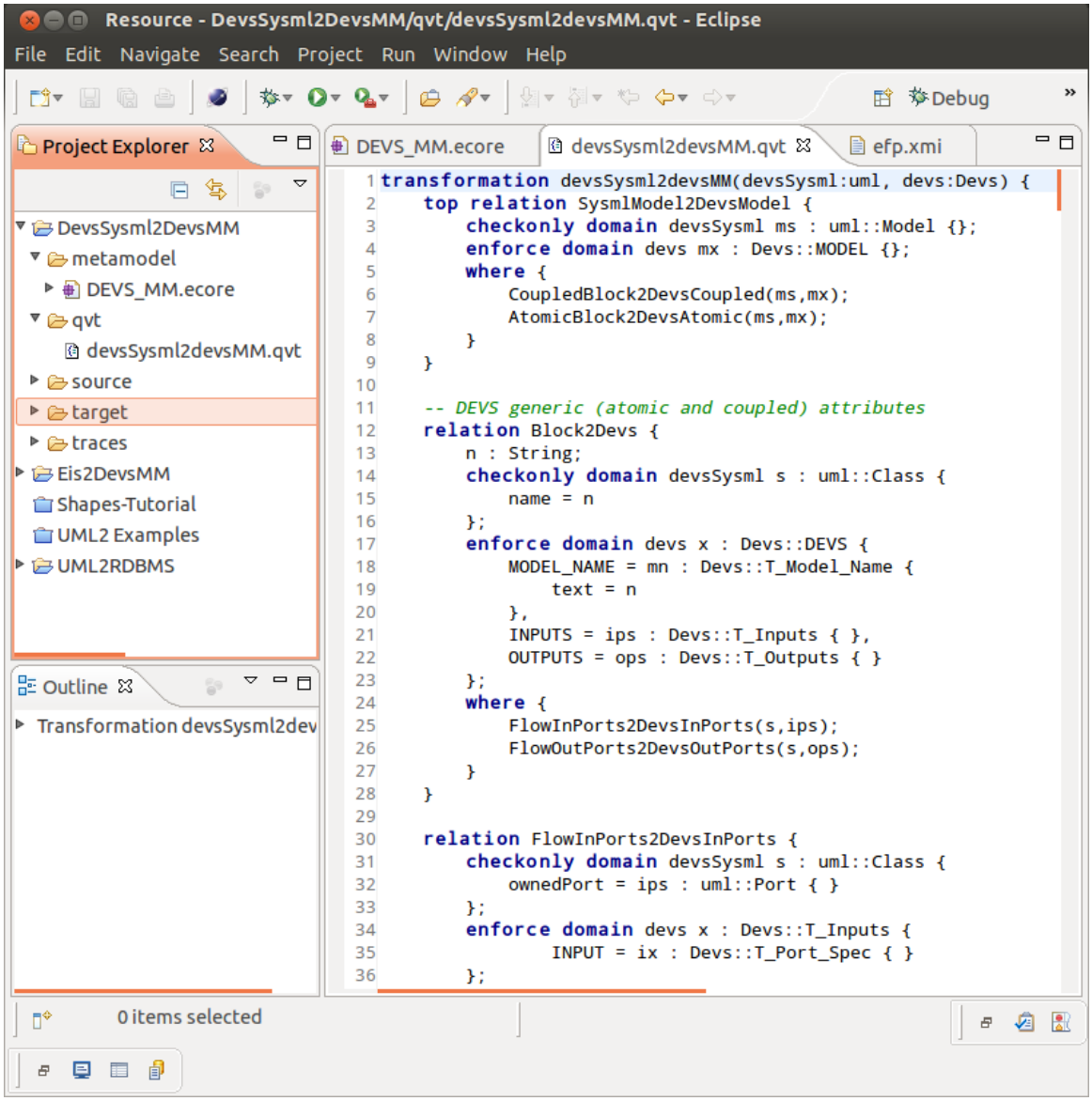
```
    when { tr.getAppliedStereotype('DEVS_SysML::DEVS State Transition')->
        notEmpty(); }
    where { value(tav,v); }
  }
  relation NoTransitions2InfiniteConditionalTimeAdvances {
    ns : String;
    checkonly domain devsSysml r : uml::Region {
      subvertex = sv : uml::Vertex { name = ns }
    };
    enforce domain devs taf : Devs::T_Time_Advance_Function {
      CONDITIONAL_TIME_ADVANCE = cta : Devs::T_Conditional_Time_Advance {
        STATE_CONDITION = sc : Devs::PCDATA { text = ns },
        TIME_ADVANCE = ta : Devs::T_Time_Advance {
          VALUE = ev : Devs::T_Value {
            type = 'Real',
            value = 'Infinity'
          }
        }
      }
    };
    when {
      sv.getAppliedStereotype('DEVS_SysML::DEVS State')->notEmpty();
      sv.getOutgoings()->isEmpty();
    }
  }
}
```

Το εργαλείο *Medini* χρησιμοποιήθηκε τόσο για τον ορισμό του μετα-μοντέλου DEVS, όσο και για τον ορισμό και την εκτέλεση του μετασχηματισμού QVT από μοντέλα SysML σε μοντέλα DEVS. Στο σχήμα Α΄.1 φαίνεται το GUI για τη δημιουργία και την προβολή μετα-μοντέλων MOF.

Το σχήμα Α΄.2 παρουσιάζει τον editor για συγγραφή σχέσεων QVT, του εργαλείου *Medini*. Όπως φαίνεται στο αριστερό τμήμα του σχήματος, το *DEVS_MM.ecore* είναι το αρχείο που περιέχει τον ορισμό του μετα-μοντέλου DEVS, ενώ το μετα-μοντέλο της UML είναι ενσωματωμένο στο εργαλείο. Το *Medini* επιτρέπει την εκτέλεση και το debugging μετασχηματισμών QVT.



Σχήμα Α΄.1: Ο ορισμός του μετα-μοντέλου DEVS στο εργαλείο Medini σε όρους MOF.



Σχήμα Α΄.2: Ο μετασχηματισμός από το μετα-μοντέλο UML στο μετα-μοντέλο DEVS με το εργαλείο Medini.

Α΄.2 Δημιουργία Εκτελέσιμων Μοντέλων XLSC από Μοντέλα DEVS με XSLT

Καθώς δεν υπάρχει διαθέσιμος προσομοιωτής DEVS, ο οποίος να δέχεται ως είσοδο μοντέλα DEVS, σύμφωνα με το προτεινόμενο μετα-μοντέλο, είναι απαραίτητη η δυνατότητα μετασχηματισμού των μοντέλων αυτών σε κάποια εκτελέσιμη μορφή. Για την υλοποίηση του πλαισίου αυτοματοποίησης προσομοίωσης με DEVS (κεφάλαιο 5), έχει επιλεγεί η εκτελέσιμη μορφή XLSC [33]. Η XSLT -το de facto πρότυπο για συντακτικούς μετασχηματισμούς εγγράφων XML διαφορετικής μορφής- χρησιμοποιήθηκε για αυτό το σκοπό. Στο ίδιο θέμα, που αφορούσε και το listing Α΄.1 της ενότητας Α΄.1, παρουσιάζονται στο listing Α΄.2 τα XSLT templates, που μετασχηματίζουν τις συναρτήσεις *DEVS Internal Transition*, *Time Advance* και *Output* στα αντίστοιχα στοιχεία XLSC.

Listing Α΄.2: Μετατροπή των συναρτήσεων DEVS Internal, Time Advance και Output σε XLSC με XSLT

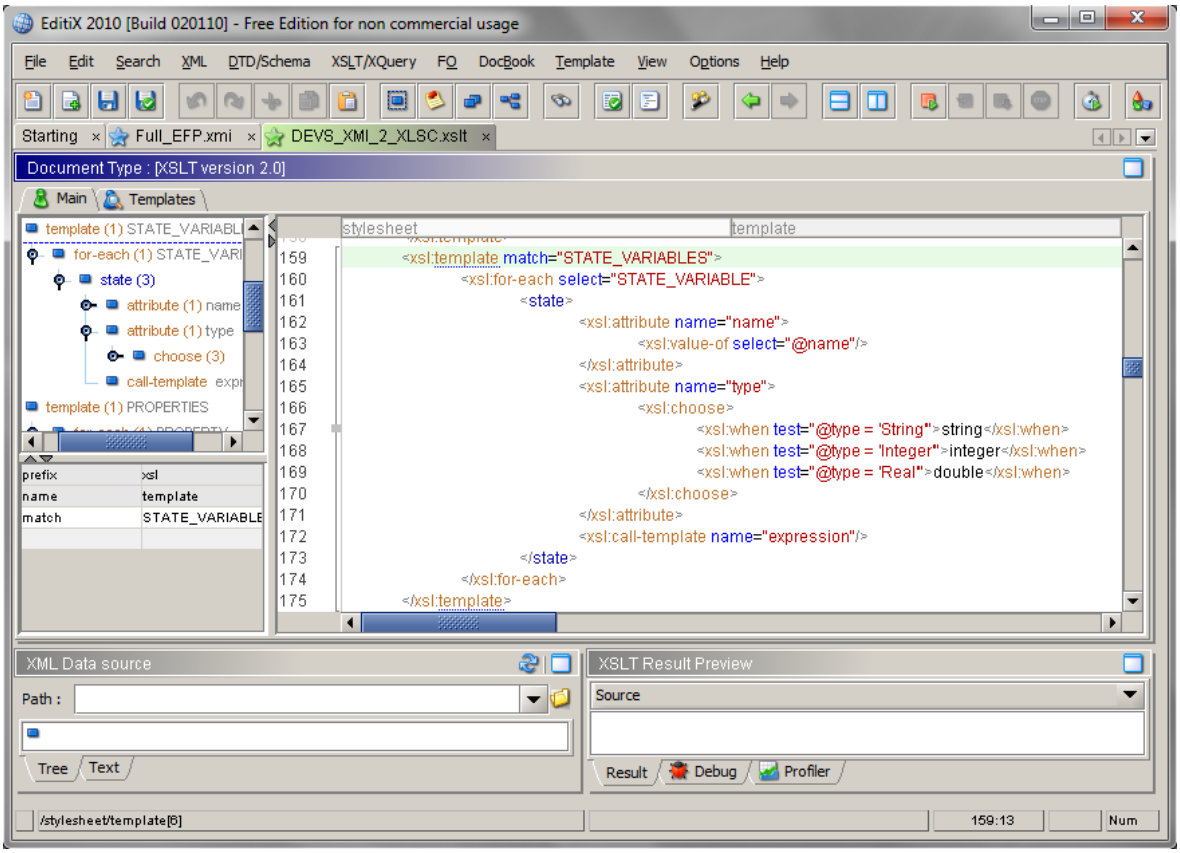
```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="2.0"
    exclude-result-prefixes="xs xdt err fn xmi DevsXml"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    xmlns:xdt="http://www.w3.org/2005/xpath-datatypes"
    xmlns:err="http://www.w3.org/2005/xqt-errors"
    xmlns:xmi="http://www.omg.org/XMI"
    xmlns:DevsXml="urn:DevsXml.ecore">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="/xmi:XMI">
  <model>
      <xsl:apply-templates/>
  </model>
  </xsl:template>
  ...
  <xsl:template match="INTERNAL_TRANSITION_FUNCTION">
    <internalTransitionFunction>
      <action>
        <xsl:for-each select="CONDITIONAL_FUNCTION">
          <if>
            <condition>
              <equal>
                <retrieve state="phase"/>
                <string>
                  <xsl:value-of select="STATE_CONDITION/@text"/>
                </string>
              </equal>
            </condition>
```

```
        <then>
            <update state="phase">
              <string>
                <xsl:value-of select="TRANSITION_FUNCTION/NEW_STATE/@text"/>
              </string>
            </update>
            <xsl:for-each
                  select="TRANSITION_FUNCTION/STATE_VARIABLE_UPDATES/
                     STATE_VARIABLE_UPDATE">
              <update>
                <xsl:attribute name="state">
                  <xsl:value-of select="@name"/>
                </xsl:attribute>
                <xsl:call-template name="expression"/>
              </update>
            </xsl:for-each>
        </then>
        </if>
      </xsl:for-each>
      <update state="phase_changed">
        <integer>1</integer>
      </update>
  </action>
  </internalTransitionFunction>
</xsl:template>
<xsl:template match="TIME_ADVANCE_FUNCTION">
  <timeAdvanceFunction>
    <action>
      <if>
        <condition>
          <equal>
            <retrieve state="phase_changed"/>
            <integer>1</integer>
          </equal>
        </condition>
        <then>
          <xsl:for-each select="CONDITIONAL_TIME_ADVANCE">
            <if>
              <condition>
                <equal>
                  <retrieve state="phase"/>
                  <string>
                    <xsl:value-of select="STATE_CONDITION/@text"/>
                  </string>
                </equal>
```

```
            </condition>
            <then>
              <update state="sigma">
                <xsl:for-each select="TIME_ADVANCE">
                  <xsl:call-template name="expression"/>
                </xsl:for-each>
              </update>
            </then>
          </if>
        </xsl:for-each>
        <update state="phase_changed">
          <integer>0</integer>
        </update>
      </then>
    </if>
  </action>
 </timeAdvanceFunction>
</xsl:template>
<xsl:template match="OUTPUT_FUNCTION">
 <outputFunction>
  <action>
   <xsl:for-each select="CONDITIONAL_OUTPUT_FUNCTION">
    <if>
     <condition>
      <equal>
       <retrieve state="phase"/>
       <string>
        <xsl:value-of select="@state"/>
       </string>
      </equal>
     </condition>
     <then>
      <xsl:for-each select="PORT_OUTPUTS/SEND">
       <output>
        <xsl:attribute name="port">
         <xsl:value-of select="@port"/>
        </xsl:attribute>
        <xsl:attribute name="variable">v_<xsl:value-of select="@port
            "/>
        </xsl:attribute>
        <xsl:call-template name="expression"/>
       </output>
       <send message="Item">
        <xsl:attribute name="port"><xsl:value-of select="@port"/></
            xsl:attribute>
```

```
                </send>
              </xsl:for-each>
            </then>
          </if>
        </xsl:for-each>
      </action>
    </outputFunction>
  </xsl:template>
</xsl:stylesheet>
```

Ο μετασχηματισμός XSLT ορίστηκε στο εργαλείο *EditiX*, το οποίο παρέχει δυνατότητα εκτέλεσης μετασχηματισμών XSLT και έχει διαθέσιμη δωρεάν έκδοση. Ένα screenshot του εργαλείου παρέχεται στο σχήμα Α΄.3. Σε αυτό παρουσιάζεται ένα απλό template μετασχηματισμού, που μετατρέπει στοιχεία *STATE_VARIABLES* του μοντέλου *DEVS* σε στοιχεία *StatePart* εγγράφων XML μορφής XLSC.



Σχήμα Α΄.3: Η μετατροπή μοντέλων DEVS σε έγγραφα XLSC με το εργαλείο EditiX.

Όπως φαίνεται, τα χρησιμοποιούμενα στοιχεία τύπου *xsl*, καθορίζουν τη ροή ελέγχου του μετασχηματισμού. Τα υπόλοιπα (απλά) στοιχεία διαμορφώνουν το σκελετό του αποτελέσματος του μετασχηματισμού και εμπλουτίζονται κατάλληλα από τα πρώτα (*xsl*) για την παραγωγή του τελικού αποτελέσματος. Για παράδειγμα, το πρώτο στοιχείο (*xsl:template*) υποδεικνύει ότι όλες οι εντολές, που περιέχονται σε αυτό, πρέπει να εκτελούνται για κάθε στοιχείο *STATE_VARIABLES*, που βρίσκονται στα πλαίσια αυτού του template. Το δεύτερο στοιχείο

(*xsl:for-each*) υποδεικνύει ότι όλες οι εντολές, που περιέχονται σε αυτό, πρέπει να εκτελεστούν για κάθε στοιχείο *STATE_VARIABLE*, που περιέχεται ακριβώς κάτω από το στοιχείο *VARIABLES*, που βρίσκεται στην προηγούμενη γραμμή. Το στοιχείο *state*, που ακολουθεί, θα προστεθεί στο *output* για κάθε στοιχείο *STATE_VARIABLE*. Οι εντολές *xsl:attribute*, που ακολουθούν, προσθέτουν δύο attributes (*name* και *type*) στο στοιχείο *state*. Η τιμή του πρώτου attribute (*name*) θα ληφθεί από το attribute *name* του στοιχείου *STATE_VARIABLE*. Η τιμή του άλλου attribute (*type*) προκύπτει από τον κατάλληλο μετασχηματισμό της τιμής του attribute *type* του στοιχείου *STATE_VARIABLE*, πως φαίνεται από τα στοιχεία *xsl:choose* και *xsl:when*. Τέλος, καλείται η έκφραση template (*xsl:call-template*), ώστε να αναζητηθεί η αρχική τιμή του στοιχείου *STATE_VARIABLE* και να μετασχηματιστεί στις κατάλληλες τιμές κατάστασης.

Ακρωνυμια

A

AD Διάγραμμα Ενεργειών (Activity Diagram). 27, 69, 76, 81, 92, 100

API Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface). 70, 92

ATL Γλώσσα Μετασχηματισμών ATLAS (ATLAS Transformation Language). 42, 61

B

BDD Διάγραμμα Ορισμού Μπλοκ (Block Definition Diagram). 25, 68, 69, 71, 89–91

C

CD Διάγραμμα Κλάσεων (Class Diagram). 78

D

DEVS Προδιαγραφή Συστημάτων Διακριτού Χρόνου (Discrete Event System Specification). 5, 6, 12, 13, 16, 20–22, 40, 41, 44, 45, 65–78, 80, 81, 83–92, 94, 95, 97, 100, 102, 105–107, 109, 113, 116–118, 124, 125, 129, 132, 136, 139

E

Ecore Βασικό Μετα-μοντέλο του Πλαισίου Μοντελοποίησης Eclipse (Eclipse Modeling Framework Core Meta-Model). 61

EIS Εταιρικά Πληροφοριακά Συστήματα (Enterprise Information Systems). 21, 102, 103, 105–107, 109, 111, 113, 114, 117, 119, 124

EMF Πλαίσιο Μοντελοποίησης Eclipse (Eclipse Modeling Framework). 61

G

GUI Γραφική Διεπαφή Χρήστη (Graphical User Interface). 92, 109, 136

I

IBD Διάγραμμα Εσωτερικού Μπλοκ (Internal Block Diagram). 25, 27, 68, 69, 71, 81, 90

INCOSE Διεθνές Συμβούλιο Μηχανικής Συστημάτων (International Council on Systems Engineering). 24

M

MagicDraw MagicDraw Modeling Tool by No Magic, Inc.. 21, 70, 85, 87, 89, 92, 103, 113

MBSE Βασισμένη-σε-Μοντέλα Ανάπτυξη Συστημάτων (Model-based Systems Engineering). 5, 6, 18, 24, 40

MDA Οδηγούμενη από Μοντέλα Αρχιτεκτονική (Model Driven Architecture). 5, 6, 34, 42, 43, 57, 59

Medini Medini QVT. 94

MOF Υποδομή Μετα-Αντικειμένων (Meta-Object Facility). 20–22, 34, 35, 45, 52, 53, 56–61, 65, 78, 80, 106, 116, 119, 136

O

OCL Γλώσσα Περιορισμών Αντικειμένων (Object Constraint Language). 21, 36, 60, 63, 70, 92, 133

OMG Object Management Group. 20, 21, 34, 41, 42

P

PD Διάγραμμα Παραμετροποίησης (Parametric Diagram). 25, 27, 42, 68, 69, 71, 90

PIM Μοντέλο Ανεξάρτητο Πλατφόρμας (Platform Independent Model). 34

PLM Διαχείριση Κύκλου Ζωής Προϊόντων/Συστημάτων (Product Lifecycle Management). 119

PSM Μοντέλο Συγκεκριμένης Πλατφόρμας (Platform Specific Model). 34

Q

QVT Query/View/Transform. 21, 34–36, 52, 53, 56–61, 63, 80, 81, 83, 85, 94, 95, 106, 107, 113, 116, 119, 124, 132, 136

QVT-C QVT-Core. 36

QVT-O QVT-Operational. 36, 42, 61, 62

QVT-R QVT-Relations. 21, 22, 36, 59–61, 132

R

RD Διάγραμμα Απαιτήσεων (Requirement Diagram). 29

S

SD Διάγραμμα Διαδοχής (Sequence Diagram). 29

SMD Διάγραμμα Μηχανών Καταστάσεων (State Machine Diagram). 27, 45, 68, 69, 71, 81, 83, 91, 100, 132

SysML Γλώσσα Μοντελοποίησης Συστημάτων (Systems Modeling Language). 13, 20–22, 24, 25, 40–48, 52–58, 60, 63, 65, 66, 68–77, 80, 81, 83, 86, 88–92, 94, 100–103, 106, 107, 117, 124, 125, 129, 132, 136

T

TGG Γραμματική Τριπλών Γράφων (Triple Graph Grammar). 60, 61

U

UCs Use Case Diagrams. 29

UML Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language). 20, 21, 25, 41, 42, 45, 48, 52–54, 56, 57, 60, 63, 69, 70, 80, 85, 86, 89, 92, 94, 101, 117, 119, 136

X

XLSC Βασισμένη στην XML Γλώσσα για Συστατικά Λογισμικού Προσομοίωσης (XML-based Language for Simulation Components). 83–85, 87, 97, 98, 127, 139, 142

XMI Ανταλλαγή Μεταδεδομένων XML (XML Metadata Interchange). 61, 62, 83, 87, 97, 109, 113

XML Επεκτάσιμη Γλώσσα Σημάνσεων (Extensible Markup Language). 21, 40, 41, 45, 61, 62, 66, 83, 87, 97, 111, 113, 139, 142

XQuery Γλώσσα Επερωτήσεων XML (XML Query Language). 36

XSLT EXtensible Stylesheet Language Transformations. 36, 61, 62, 83, 84, 87, 97, 109, 113, 139, 142