

## Κεφάλαιο 2

# Μοντελοποίηση Ενσωματωμένων Συστημάτων

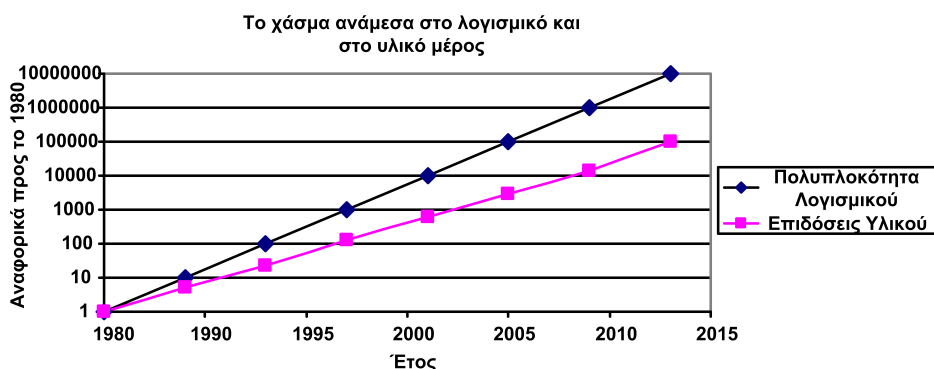
Ο σχεδιασμός ενός ενσωματωμένου συστήματος δεν είναι μια τυποποιημένη διαδικασία που καταλήγει πάντα στην ίδια υλοποίηση. Σπάνια μπορεί να βρεθεί μια βέλτιστη λύση -αν υπάρχει- ως προς κάθε απαίτηση του χρήστη ή του μηχανικού. Αντιθέτως, ο σχεδιαστής πρέπει να λαμβάνει υπόψη του όλες τις απαιτήσεις της εφαρμογής, τα χαρακτηριστικά της διαθέσιμης αρχιτεκτονικής και τους στόχους, και να προτείνει λύσεις που να ικανοποιούν όσο το δυνατόν περισσότερο τις ανάγκες του. Ο σχεδιαστής θα πρέπει διαρκώς να αναλύει και να εκτιμά τους πιθανούς συμβιβασμούς (trade-offs), ενώ ταυτόχρονα θα πρέπει να βελτιστοποιεί όσο το δυνατόν περισσότερο συγκεκριμένα χαρακτηριστικά της εφαρμογής του. Όσον αφορά το σχεδιασμό ενσωματωμένων συστημάτων, ο σχεδιαστής καλείται να μεγιστοποιεί την επίδοση του συστήματος με ταυτόχρονη ελαχιστοποίηση του κόστους. Αυτή η σύγκρουση προκαλεί και την ύπαρξη μιας πληθώρας λύσεων, οι οποίες ικανοποιούν και διαφορετικές ανάγκες. Για παράδειγμα, ένας μηχανικός που θέλει να σχεδιάσει ένα ψηφιακό ρολόι, θα προσπαθήσει να ελαχιστοποιήσει το κόστος και ταυτόχρονα να καλύψει τις ελάχιστες επεξεργαστικές και λειτουργικές απαιτήσεις. Από την άλλη πλευρά, ένας μηχανικός που θα κληθεί να σχεδιάσει έναν ψηφιακό προσωπικό βοηθό (Personal Digital Assistant), θα προσπαθήσει να μεγιστοποιήσει τις επιδόσεις της συσκευής από άλλες παρόμοιες συσκευές, ώστε να έχει καλύτερες επιδόσεις, αλλά με λιγότερη πίεση στο κόστος της συσκευής. Σε όλες λοιπόν τις περιπτώσεις ο στόχος είναι ένας: Μεγιστοποίηση της επίδοσης με όσο το δυνατόν μικρότερο κόστος. Αυτό επιτυγχάνεται με τη μοντελοποίηση.

## 2.1 Η ανάγκη για τη μοντελοποίηση

Η ραγδαία ανάπτυξη των ενσωματωμένων συστημάτων, όπως είδαμε προηγουμένως, δεν οφείλεται μόνο στην πρόοδο που επιτελέστηκε στο υλικό μέρος (π.χ. αύξηση επιδόσεων των επεξεργαστών, αύξηση βαθμού ολοκλήρωσης), αλλά και από το λογισμικό μέρος. Το λογισμικό μέρος είναι αυτό που οργανώνει, κατευθύνει και ελέγχει το υλικό μέρος για τη σωστή και έγκυρη λειτουργία του συστήματος. Όπως τα χαρακτηριστικά του υλικού μέρους επηρεάζουν σημαντικά την ταχύτητα και την κατανάλωση ενέργειας του συστήματος (π.χ. η χρήση ενός πολύ αργού επεξεργαστή θα έχει ως συνέπεια το όλο σύστημα να λειτουργεί με μικρή ταχύτητα), έτσι και το λογισμικό μέρος παίζει έναν κυρίαρχο ρόλο στη διαμόρφωση όλων των μετρικών ενός συστήματος. Ένα λογισμικό μέρος καλά προγραμματισμένο θα βοηθήσει το υλικό μέρος να αποδώσει τα μέγιστα των δυνατοτήτων του, ενώ σε αντίθετη περίπτωση, θα κάνει και το ενσωματωμένο σύστημα με τα πιο γρήγορα υλικά μέρη να μειονεκτεί αισθητά έναντι παρόμοιων συστημάτων.

Το λογισμικό μέρος συμπληρώνει το υλικό μέρος. Η σχέση που συνδέει αυτά τα δύο μέρη είναι παρόμοια (σε ένα αρκετά υψηλό επίπεδο αφαίρεσης), με τη σχέση που έχει ένα όχημα με τον οδηγό του. Αν και το όχημα έχει όλη τη λειτουργικότητα που απαιτείται, εντούτοις περιμένουμε από τον οδηγό να το εκκινήσει, να κάνει τις κατάλληλες ενέργειες, ώστε το όχημα να μετακινηθεί και να οδηγηθεί σε κάποια τοποθεσία που είχε τεθεί ως προορισμός. Μπορεί να θεωρηθεί ότι ο οδηγός παριστάνει το λογισμικό μέρος, ενώ το υλικό μέρος συνδέεται με το όχημα. Έτσι, η συνύπαρξη και των δύο αυτών μερών αποτελεί το συνολικό σύστημα. Παρόλο που και τα δύο είναι αναπόσπαστα μέρη του συστήματος, κατά τα τελευταία χρόνια έχει δοθεί μεγαλύτερη έμφαση στο λογισμικό και, ιδιαίτερα, στους τρόπους καλύτερης μοντελοποίησης και προγραμματισμού των εφαρμογών, ειδικά για ενσωματωμένα συστήματα. Αυτό οφείλεται κυρίως στην τάση που επικρατεί να χρησιμοποιούνται τυπικά εξαρτήματα υλικού μέρους από το ράφι (off-the-shelf components), όπως για παράδειγμα επεξεργαστές των εταιρειών ARM, ή TI, ή μνήμες των εταιρειών Motorola, ή Kingston, κ.τ.λ. Όπως παρατηρείται παγκοσμίως, το πρόβλημα της ανάπτυξης ενός συστήματος, κατευθύνεται περισσότερο προς τη συγγραφή του λογισμικού μέρους, που θα ενεργοποιήσει το υλικό μέρος. Έτσι, η μερική τυποποίηση των υλικών μερών των ενσωματωμένων συστημάτων αποτελεί έναν παράγοντα που έχει μετατοπίσει το πρόβλημα της ανάπτυξης ενσωματωμένων συστημάτων προς το λογισμικό μέρος. Όμως, αυτός δεν είναι ο μόνος λόγος.

Τα τελευταία χρόνια, η ανάπτυξη των εφαρμογών των ενσωματωμένων συστημάτων έχει γίνει ο κυρίαρχος διαμορφωτής του κόστους για έναν ακόμη παράγοντα: αυτός είναι η πολυπλοκότητα των εφαρμογών. Οι πρώτες εκδόσεις των ενσωματωμένων συστημάτων είχαν ένα μικρό πρόγραμμα, το οποίο



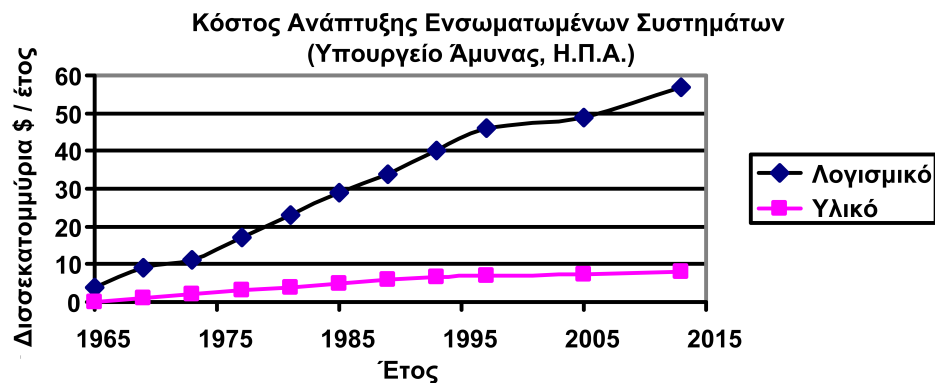
Σχήμα 2.1: Η πολυπλοκότητα των εφαρμογών ενσωματωμένων συστημάτων αυξάνεται με πολύ μεγαλύτερο ρυθμό σε σχέση με τις επιδόσεις των επεξεργαστών.

επιτελούσε κάποιες πολύ απλές λειτουργίες. Μαζί με την απαίτηση της αγοράς για ενσωματωμένα συστήματα με καλύτερες επιδόσεις στο υλικό μέρος, υπήρξε και η απαίτηση για πιο πολλές λειτουργίες, για προγράμματα φιλικά προς το χρήστη, και για εφαρμογές αρκετά απαιτητικές (όπως π.χ. η αναπαραγωγή πολυμέσων). Γι' αυτόν το λόγο υπήρξε και υπάρχει αρκετά μεγαλύτερη έμφαση στην ανάπτυξη των εφαρμογών ενσωματωμένων συστημάτων.

Ένας τρόπος μέτρησης της ανάπτυξης των εφαρμογών αποτελεί και το μέτρο της πολυπλοκότητας των εφαρμογών. Η πολυπλοκότητα μιας εφαρμογής εξαρτάται από τον αριθμό των εντολών και συναρτήσεων που την αποτελούν, από τις λειτουργίες που αυτή κάνει, από το μέγεθος κώδικα που αποθηκεύεται στη μνήμη εντολών, και από τον αποθηκευτικό όγκο σε μνήμη δεδομένων. Αν βγάλουμε χονδρικά ένα μέσο όρο της πολυπλοκότητας των εφαρμογών θα παρατηρήσουμε ότι ακολουθεί μια εκθετική αύξηση χρονολογικά (Σχήμα 2.1). Αν στο ίδιο γράφημα τοποθετήσουμε και τις επιδόσεις των επεξεργαστών, θα δούμε ότι αυτές υστερούν, και δεν μπορούν να καλύψουν τις τρέχουσες απαιτήσεις των εφαρμογών. Βλέπουμε, λοιπόν, ότι υπάρχει ένα χάσμα ανάμεσα στο τι μπορεί να εκτελεστεί σε ένα σύστημα (επιδόσεις υλικού μέρους) και στο τι απαιτούν οι εφαρμογές για να εκτελεστούν (πολυπλοκότητα λογισμικού μέρους).

Το πρόβλημα της δημιουργίας λογισμικού ενσωματωμένων συστημάτων είναι τόσο σημαντικό, ώστε μεγάλες εταιρείες, όπως η Microsoft και η Sun (εταιρείες που έχουν κυριαρχήσει στο λογισμικό μέρος των επιτραπέζιων συστημάτων, ή των διακομιστών), έχουν κατευθύνει μέρος του προϋπολογισμού τους στην έρευνα και ανάπτυξη εφαρμογών ενσωματωμένων συστημάτων. Για παράδειγμα, η Microsoft αναπτύσσει τα Windows Mobile, η Google το Android, και η Sun τη Java Virtual Machine, που όλα συμβάλλουν στην ανάπτυξη λογισμικού ενσωματωμένων συστημάτων.

Το λογισμικό μέρος των ενσωματωμένων συστημάτων, είναι ένας ακμάζων

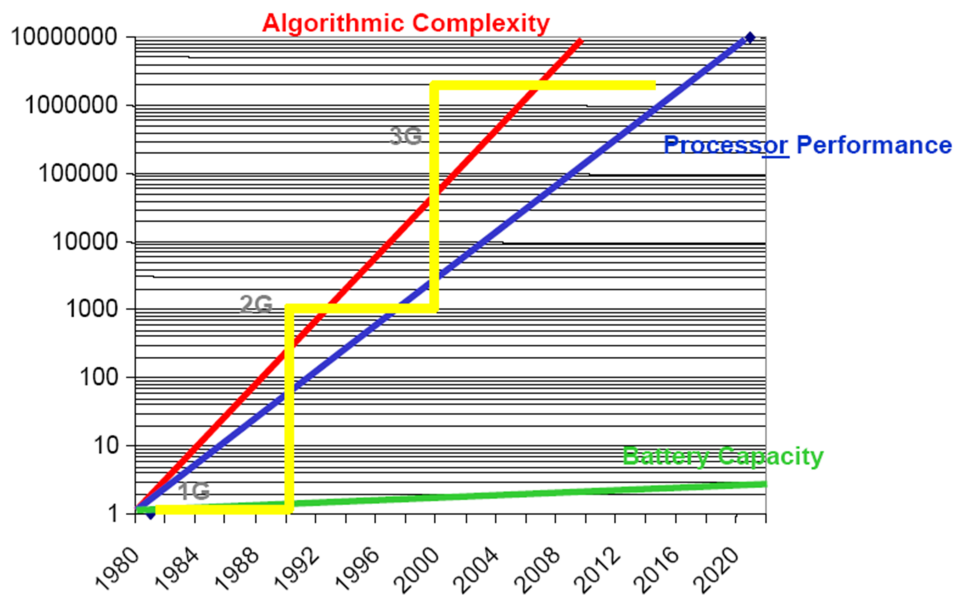


Σχήμα 2.2: Το λογισμικό μέρος κυριαρχεί στη διαμόρφωση του κόστους ενός ενσωματωμένου συστήματος.

τομέας, και αυτό φαίνεται στα οικονομικά μεγέθη που το συνοδεύουν. Για παράδειγμα, τα στοιχεία από το Υπουργείο Άμυνας των Η.Π.Α. – ένας από τους μεγαλύτερους παγκοσμίως αγοραστές ενσωματωμένων συστημάτων – (Σχήμα 2.2), δείχνουν ότι το κόστος ανάπτυξης του λογισμικού είναι έως και 6 φορές μεγαλύτερο από το κόστος ανάπτυξης των υλικών μερών. Είναι γνωστό, ότι το λογισμικό καταλαμβάνει περισσότερο από το 70% του κόστους ανάπτυξης πολύπλοκων ενσωματωμένων συστημάτων, όπως φορητά τερματικά, ηλεκτρονικά συστήματα αυτοκινήτων, συστήματα επικοινωνιών κτλ [9], [10], [11], [12].

Τέλος, η σπουδαιότητα του λογισμικού των ενσωματωμένων συστημάτων φαίνεται καθαρά και από τα στοιχεία που αναφέρονται στην έκθεση του Υπουργείου Εμπορίου και Βιομηχανίας της Ολλανδίας, στην οποία αναφέρεται ότι ο αριθμός των ατόμων ή εταιρειών που παίρνουν συμβόλαια για ανάπτυξη κώδικα ενσωματωμένων συστημάτων θα αυξηθεί από 2 εκατομμύρια το 1994 σε 13 εκατομμύρια το 2015, ενώ ο αριθμός των κατασκευαστών υλικού ενσωματωμένων συστημάτων θα αυξηθεί από 0,6 εκατομμύρια σε 1,1 εκατομμύρια, αντιστοίχως [13].

Είναι σημαντικό λοιπόν να αναλύεται η εφαρμογή, που θα εκτελεστεί στο ενσωματωμένο σύστημα, και να γίνεται συσχεδιασμός του υλικού και του λογισμικού μέρους. Η μεθοδολογία, σχεδιασμού ΕΣ αποτελείται από στάδια που βελτιώνουν το λογισμικό μέρος και από στάδια που βελτιώνουν το υλικό μέρος. Επίσης, μια σωστή μεθοδολογία έχει εφαρμοστεί με επιτυχία σε ένα πλήθος εφαρμογών, που εκτελούνται σε ενσωματωμένα συστήματα, γεγονός που δείχνει τη γενική χρήση που μπορεί αυτή να έχει.



Σχήμα 2.3: Η πολυπλοκότητα των εφαρμογών αυξάνεται λογαριθμικά.

## 2.2 Το NRE εξαρτάται από τη μοντελοποίηση

Όπως έχει αναφερθεί στην εισαγωγή του βιβλίου, το κόστος έρευνας και ανάπτυξης (non recurrent engineering cost, NRE) ενός ενσωματωμένου συστήματος είναι αποφασιστικής σημασίας για τη διαμόρφωση της τιμής του τελικού προϊόντος, αφού θα πρέπει να ενσωματωθεί στην τελική τιμή. Όσο μεγαλύτερο είναι αυτό το αρχικό κόστος, τόσο πιο πολύ θα αυξηθεί η τιμή των τελικών προϊόντων και ενδεχομένως θα επηρεάσει αρνητικά τον αριθμό πωλήσεων, προκαλώντας ίσως και ζημιά στην εταιρία που το προμηθεύει στην αγορά.

Η μείωση του κόστους NRE είναι λοιπόν ζωτικής σημασίας, αφού η ίδια η βιωσιμότητα μιας εταιρίας εξαρτάται από αυτό, και θα πρέπει να είναι όσο το δυνατόν μικρότερο. Το ίδιο προϊόν μπορεί σε δυο διαφορετικές εταιρίες να συνδέονται με NRE κόστη που έχουν πολύ μεγάλη διαφορά. Το κόστος NRE για το ίδιο προϊόν σε δυο διαφορετικές εταιρίες είναι διαφορετικό. Για να ελαχιστοποιηθεί αυτό το κόστος θα πρέπει να θυμηθούμε (1ο κεφάλαιο) τους παράγοντες που το διαμορφώνουν. Ο πρώτος παράγοντας είναι το κόστος μισθοδοσίας του προσωπικού. Μια εταιρία μπορεί να έχει υψηλό μισθολογικό κόστος λόγω νοοτροπίας ή φορολογίας και ασφαλιστικού, ενώ μια εταιρία σε άλλη περιοχή του πλανήτη μπορεί να έχει υποπολλαπλάσιο κόστος. Για αυτό αρκετές εταιρίες του Δυτικού Κόσμου, προβαίνουν σε εξωτερικές αναθέσεις (outsourcing) ή ανοίγουν παραρτήματα σε χώρες με χαμηλό μισθολογικό κόστος, όπως η Ινδία ή η Κίνα. Ο επόμενος παράγοντας είναι τα πάγια έξοδα των κτιρίων. Ομοίως, αυτό μπο-

ρεί να μειωθεί σε μετεγκατάσταση της εταιρίας είτε σε κατάλληλη οικοδομική ζώνη που παρέχει σε μικρότερο κόστος κτιριακή επιφάνεια, νερό, ρεύμα και διαδίκτυο, ή σε μετεγκατάσταση σε άλλη πόλη ή χώρα. Ένας άλλος παράγοντας, είναι το κόστος των υλικών για τη δημιουργία του πρωτοτύπου. Αν και αυτό το κόστος είναι συνήθως μικρότερο από τα δυο προηγούμενα, εντούτοις διαμορφώνει και αυτό το NRE, ιδιαίτερα αν πρέπει να γίνουν πολλά και μακροχρόνια πειράματα, δοκιμάζοντας πολλά διαφορετικά σενάρια ή σχεδιασμούς. Για να μειωθεί αυτό το κόστος, εκτός από την προμήθεια των υλικών από πάροχο που έχει τις χαμηλότερες τιμές και ταυτόχρονα καλύπτει τα επίπεδα αξιοπιστίας, ποιότητας και ταχύτητας παράδοσης, θα πρέπει να ακολουθηθεί μια τεχνική που θα επιτύχει μείωση του χρόνου, μείωση των λανθασμένων σχεδιαστικών επιλογών και μείωση των προβληματικών πρωτοτύπων. Αυτή η τεχνική ονομάζεται μοντελοποίηση, και είναι πάντα το πρώτο βήμα σε οποιαδήποτε μεθοδολογία σχεδιασμού, είτε ενσωματωμένων συστημάτων είτε οποιοδήποτε άλλου προϊόντος.

### 2.3 Η σημασία της μοντελοποίησης

Η μοντελοποίηση είναι μια τεχνική (ή για να είμαστε ακριβείς, ένα σύνολο από τεχνικές) που επιτρέπουν ένα σχεδιαστή να πετύχει το στόχο του με την πρώτη προσπάθεια. Αν και έχει μεγάλη σημασία, αρκετές φορές παραλείπεται από τα επίσημα διαγράμματα σχεδίασης, λόγω χρονικής πίεσης, αφού κάποιιοι σχεδιαστές πιστεύουν ότι μπορεί να καταφέρουν το τελικό αποτέλεσμα (π.χ. την ανάπτυξη ενός ΕΣ) με τη σύνδεση των δομοστοιχείων και τη συγγραφή του κώδικα. Αν και μπορεί να το πετύχουν και να σχεδιάσουν κάτι που να λειτουργεί, εντούτοις η μεγαλύτερη πιθανότητα είναι ότι δε θα το καταφέρουν και θα χάσουν το παράθυρο εισαγωγής στην αγορά, με δυσάρεστες οικονομικές συνέπειες. Αλλά, ακόμη και να τα καταφέρουν, δε σημαίνει ότι το τελικό αποτέλεσμα θα είναι στιβαρό και θα λειτουργεί σε οποιαδήποτε συνθήκη.

Παραδείγματα που δείχνουν τη σημασία της μοντελοποίησης μπορούμε να βρούμε σε κάθε χώρο. Ένα προσωπικό απλό παράδειγμα του συγγραφέα κατά την επίβλεψη μιας εργασίας φοιτητή, ήταν όταν έπρεπε να δημιουργηθεί μια συσκευή συνδεδεμένη στο Internet που έλεγχε τέσσερα ρελέ. Ο σχεδιασμός έγινε χωρίς μοντελοποίηση και εξέταση των αναγκών από το φοιτητή. Το πρωτότυπο συνδέθηκε στην τροφοδοσία και λειτούργησε με επιτυχία για 1 συσκευή. Όταν όμως συνδέθηκε και δεύτερη συσκευή, τότε παρουσιάστηκε δυσλειτουργία με τυχαία συμπεριφορά. Το πρόβλημα ήταν ότι δεν είχε μοντελοποιηθεί η ζητούμενη απαίτηση ρεύματος σε πλήρη λειτουργία (και με τα 4 ρελέ), οπότε το τροφοδοτικό των 300 mA, ενώ επαρκούσε για τον έλεγχο ενός ρελέ, δεν επαρκούσε για την ικανοποίηση της χειρότερης κατάστασης. Ένα άλλο πα-

ράδειγμα, αφορά το σχεδιασμό ενός ιστοχώρου εξέτασης φοιτητών, όπου ενώ λειτουργούσε σωστά όταν ήταν συνδεδεμένοι 1-2 φοιτητές, σε περίπτωση που συνδέονταν και οι 100 φοιτητές, δε λειτουργούσε καθόλου επειδή ο φόρτος του διακομιστή από 1% γίνονται 3000%. Το πρόβλημα ήταν στον αριθμό των ερωτημάτων ανά σελίδα που ήταν υπερβολικά πολλά, επειδή δεν είχε μοντελοποιηθεί πως να κλιμακώνεται. Με τη βελτίωση της σελίδας και τη μείωση των ερωτημάτων, το σύστημα μπόρεσε και λειτούργησε. Σε αυτά τα παραδείγματα φάνηκε ότι ακόμη και να λειτουργεί, αν δεν έχει προηγηθεί μια ανάλυση και μοντελοποίηση, τότε μπορεί να υπάρξουν περιπτώσεις που θα αποτύχει ή απαιτείται επιπρόσθετος χρόνος σχεδίασης για να βρεθεί το πρόβλημα στο τελικό προϊόν και να διορθωθεί. Ακόμα, αν αυτό το προϊόν κατασκευαστεί και διατεθεί, και αποτύχει σε μια κρίσιμη στιγμή, μπορεί να υπάρξει και απώλεια υλικής αξίας ή ανθρώπινης ζωής.

Η μοντελοποίηση δεν ανήκει στα βήματα της μεθοδολογίας που θα ήταν καλό να γίνουν ('nice to do'). Ανήκουν στα βήματα που πρέπει να γίνουν ('must do'). Η σημαντικότητα αυξάνεται όσο περισσότερο αυξάνονται η πολυπλοκότητα και οι απαιτήσεις. Αν κάποιος αποφασίσουν να κατασκευάσουν μια αποθήκη με ξύλα, αρκεί να σημειώσουν σε ένα χαρτί τις διαστάσεις των τοίχων, τα εργαλεία και τα υλικά που θα χρησιμοποιήσουν. Σε περίπτωση που δεν ταιριάζουν κάποια υλικά μεταξύ τους μπορεί να γίνουν κάποια μπαλώματα για να τα καλύψουν. Αν όμως απαιτείται η κατασκευή ενός σπιτιού με πολλούς χώρους, τότε αυξάνεται η πολυπλοκότητα και θα πρέπει να ληφθούν υπόψιν πολλά περισσότερα στοιχεία, όπως διελεύσεις των καλωδίων, των δικτύων ύδρευσης και αποχέτευσης, τα πάχη των τοίχων μαζί με τη μόνωση, τα κουφώματα και πλήθος άλλων λεπτομερειών. Αν δε γίνει αυτό, στο τέλος μπορεί να μην καλύπτονται οι προδιαγραφές κατοίκησης και να μην έχει σωστή μόνωση ή μπορεί να ξεχάσουν να βάλουν βρύσες στο μπάνιο ή ακόμη αν είναι πολυκατοικία να υπάρχει κατάρρευση, αν δεν έχει γίνει μοντελοποίηση των στατικών φορτίων. Για αυτό είναι απαραίτητο να μοντελοποιηθεί η κατασκευή ενός τέτοιου σύνθετου σχεδιασμού και μάλιστα, όχι μόνο με μια κάτοψη. Θα πρέπει να γίνει μια θερμική μελέτη, ηλεκτρολογική μελέτη, μελέτη ύδρευσης και αποχέτευσης, στατική μελέτη και άλλα. Σε περίπτωση που υπάρξει πρόβλημα σε μια τέτοια κατασκευή μπορεί να είναι δύσκολο να διορθωθεί, με αποτέλεσμα να υπάρχει σημαντική οικονομική καταστροφή.

Αν δεν υπάρξει η μοντελοποίηση, το τελικό αποτέλεσμα βασίζεται στην τύχη και σε υπερβολική εργασία της ομάδας να διορθώσει τα προβλήματα καθώς εμφανίζονται. Στις περισσότερες περιπτώσεις όμως, δεν υπάρχει τύχη, η ομάδα ασχολείται ήδη πολλές εργατώρες με το προϊόν και δε μπορεί να διαθέσει άλλες ή πλησιάζει κάποια κρίσιμη καταληκτική ημερομηνία, και λόγω κακού προγραμματισμού δεν υπάρχουν όλες οι πρώτες ύλες, που είναι απαραίτητες για την κατασκευή του πρωτοτύπου. Σχεδιαστικά project μεγάλων προ-



υπολογισμών (πάνω από 1.000.000 ευρώ) που χρηματοδοτούνται συμμετοχικά από ανώνυμους χρήστες (crowd-funding), έχουν αποτύχει λόγω έλλειψης μοντελοποίησης, όπως για παράδειγμα η εταιρία Pirate3D που μάζεψε \$1.5 εκατομμύρια δολάρια για την κατασκευή οικονομικών τρισδιάστατων εκτυπωτών, αλλά τελικά το κόστος τελικών προϊόντων ήταν πολύ μεγαλύτερο από το αναμενόμενο, ή η εταιρία Zano Drones που συγκέντρωσε \$3.5 εκατομμύρια δολάρια για την ανάπτυξη μικρών τετρακόπτερων που ελέγχονται από το smartphone, αλλά μετά από μήνες έρευνας χρεοκόπησε χωρίς να κατασκευάσει τίποτα. Σίγουρα για την ολοκληρωτική αποτυχία δεν έφταιγε μόνο η έλλειψη σωστής μοντελοποίησης των προϊόντων που ήθελαν να σχεδιάσουν, αλλά μια ελλιπής μοντελοποίηση ή λάθος εκτιμήσεις οδηγούν σε αυτό το αποτέλεσμα.

Ο σχεδιασμός και η κατασκευή ενός προϊόντος δεν πρέπει να βασίζεται στην τύχη. Θα πρέπει να έχει προηγηθεί μια σειρά μοντελοποιήσεων σε διάφορα επίπεδα, ώστε να βρεθούν από πριν τα προβλήματα, τα υλικά, τα εργαλεία και το κόστος που τα συνοδεύει. Ενώ το κάθε αποτυχημένο σχεδιαστικό έργο αποτυγχάνει με ποικίλους τρόπους, όλα τα σωστά σχεδιασμένα προϊόντα έχουν ένα κοινό στοιχείο: μια προσεκτική και σε βάθος περίοδο μοντελοποίησης. Η μοντελοποίηση έχει αποδειχθεί ότι είναι απαραίτητο στοιχείο και εργαλείο των μηχανικών. Η μοντελοποίηση δεν αφορά μόνο τον κατασκευαστικό τομέα. Χρησιμοποιείται παντού: από την οικοδομική κατασκευή έως την κατασκευή νέων οχημάτων ή αεροπλάνων ή και δορυφόρων, από την κατασκευή νέων επεξεργαστών, έως και μεγάλων συστοιχιών με δεκάδες χιλιάδες επεξεργαστές, από τη μοντελοποίηση ενός τοπικού δικτύου, ως και τη μοντελοποίηση του διαδικτύου κ.ο.κ. Οποιοδήποτε σύστημα μπορεί να μοντελοποιηθεί, με πολλούς τρόπους και με πολλαπλά επίπεδα λεπτομέρειας. Μπορεί να χρησιμοποιηθεί ένα μαθηματικό μοντέλο, ένα δομικό μοντέλο, ένα ιεραρχικό μοντέλο, ένα μοντέλο συμπεριφοράς, ένα μοντέλο απεικόνισης, ένα μοντέλο επικοινωνίας κ.ο.κ. Γενικά, ένα μοντέλο είναι οτιδήποτε μπορεί να απλοποιήσει και να απεικονίσει την πραγματικότητα, και ο λόγος που τα κατασκευάζουμε είναι για την καλύτερη κατανόηση του συστήματος που σχεδιάζουμε.

Με τη χρήση των μοντέλων επιτυγχάνουμε τους παρακάτω στόχους:

- Απεικονίζεται το σύστημα όπως είναι ή όπως το σκεφτόμαστε.
- Προσδιορίζουμε τη δομή και την ιεραρχία ή τη συμπεριφορά.
- Μας οδηγεί και μας υπενθυμίζει τα βήματα προς την ολοκλήρωση του συστήματος.
- Καταγράφει λεπτομέρειες και αποφάσεις που έχουμε πάρει.
- Μας επιτρέπει να δημιουργήσουμε πολλαπλά αντίγραφα του ίδιου προϊόντος.



- Μας βοηθάει να επιβεβαιώσουμε τις λειτουργικές και άλλες προδιαγραφές, πριν ακόμη κατασκευαστεί.
- Μας επιτρέπει να κάνουμε μια γρήγορη σχεδιαστική αναζήτηση, αν υπάρχουν πολλές εναλλακτικές.
- Μας επιτρέπει να έχουμε μια κοινή εικόνα για το τελικό προϊόν σε όλα τα μέλη της ομάδας.

Όσο πιο πολύπλοκο είναι ένα σύστημα τόσο πιο σημαντική είναι η μοντελοποίηση, και αυτό γιατί υπάρχει αδυναμία να κατανοηθεί η έντονη πολυπλοκότητα και οι αλληλεξαρτήσεις από τους μηχανικούς. Ιεραρχικά και τμηματικά όμως μπορούμε να το επιτύχουμε, αφού η προσέγγιση ‘διαίρεση-και-κατάκτηση’ που είχε προτείνει ο Edsger Dijkstra [14], δηλαδή της διαίρεσης ενός σύνθετου προβλήματος σε πολλά επιμέρους βήματα, τα οποία ξεχωριστά μπορούν να επιλυθούν, δίνουν το καλύτερο αποτέλεσμα. Αν επιτευχθεί η επίλυση όλων των μικρών προβλημάτων, τότε θα επιτευχθεί και η επίλυση του μεγάλου πολύπλοκου προβλήματος. Η πρόταση του Dijkstra υλοποιείται με τη μοντελοποίηση.

Αφού έγινε κατανοητή η σημασία της μοντελοποίησης, το επόμενο βήμα είναι να βρεθεί ο σωστός τρόπος μοντελοποίησης. Κάποιοι σχεδιαστές ΕΣ, μπορούν να μοντελοποιήσουν ένα σύστημα σε ένα κομμάτι χαρτιού, στον πίνακα, σε ένα σχεδιαστικό πρόγραμμα στο κινητό τους ή ακόμη και σε μια χαρτοπετσέτα<sup>1</sup>.

Ασφαλώς, και μόνο που γίνεται η μοντελοποίηση, είναι ένα θετικό στοιχείο, αν μπορεί να το ακολουθήσει και να τον οδηγήσει σε μια σωστή υλοποίηση. Όμως, το πρόβλημα με αυτά τα αυθαίρετα μοντέλα είναι ότι τα καταλαβαίνει μόνο ο δημιουργός τους και από την άλλη υπάρχει μια δυσκολία διαμοιρασμού σε άλλους, και συνεπώς και δυσκολία επεξεργασίας και σχολιασμού από αυτούς. Οπότε, το σημείο κλειδί είναι η δημιουργία ενός μοντέλου σε προτυποποιημένη (formal) γλώσσα, που την κατανοούν και άλλοι και μπορούν να τη χρησιμοποιήσουν για τις δικές τους ανάγκες. Επομένως, θα πρέπει η σχεδιαστική ομάδα να επιλέξει μια ή παραπάνω γλώσσες μοντελοποίησης και να τις χρησιμοποιεί έως την κατασκευή του προϊόντος ή ακόμη και μετά για τη βελτιστοποίηση και αναθεώρηση.

---

<sup>1</sup>Μερικές φορές χρησιμοποιείται η φράση ‘handkerchief design’ ή ‘handkerchief idea’, που σημαίνει ιδέα ή σχεδιασμός της χαρτοπετσέτας και συμβαίνει όταν κάποιος βρίσκεται σε κάποιο γεύμα και είτε μόνος του, είτε σε συζήτηση με συναδέλφους, σκέφτεται ένα καινοτόμο προϊόν και σχεδιάζει τα βασικά του τμήματα. Επειδή, δεν υπάρχει χαρτί σημειώσεων σε ένα γεύμα, προκειμένου να μη ξεχάσει την ιδέα, βρίσκει το πρώτο υλικό πάνω στο οποίο μπορεί να γράψει και το χρησιμοποιεί. Αυτό είναι μια χαρτοπετσέτα.

## 2.4 Οι 4 αρχές της μοντελοποίησης

Υπάρχουν τέσσερις βασικές αρχές στη μοντελοποίηση:

1. Η επιλογή του μοντέλου επιδρά στον τρόπο που αντιμετωπίζεται το πρόβλημα και στη λύση που διαμορφώνεται.
2. Κάθε μοντέλο μπορεί να εκφραστεί με διαφορετικά επίπεδα ακρίβειας.
3. Τα καλύτερα μοντέλα είναι συνδεδεμένα με την πραγματικότητα.
4. Δεν υπάρχει ένα μοντέλο για όλα. Υπάρχουν πολλαπλά μοντέλα, και θα πρέπει να χρησιμοποιούνται παραπάνω από ένα σε κάθε πρόβλημα.

Η πρώτη αρχή σημαίνει ότι η σχεδιαστική ομάδα θα πρέπει να γνωρίζει αρκετούς τρόπους μοντελοποίησης για να επιλέξει το κατάλληλο εργαλείο. Τα σωστά μοντέλα θα διαφωτίσουν και θα φέρουν στην επιφάνεια σχεδιαστικά θέματα που ίσως δεν θα είχαν γίνει κατανοητά, αν είχε επιλεγεί κάποιο άλλο μοντέλο. Για παράδειγμα, αν κάποιος μοντελοποιήσει ένα προϊόν (π.χ. μια ανεμογεννήτρια) με το να κατασκευάσει μια μακέτα μικρογραφίας και να την τοποθετήσει σε ένα πειραματικό τούνελ αέρος, ενώ κάποιος άλλος το μοντελοποιήσει με μαθηματικές εξισώσεις και κάνει προσομοίωση, ο δεύτερος θα μπορέσει να κάνει και μια καλύτερη ανάλυση, αφού θα μπορεί να αλλάξει κάποιες παραμέτρους (π.χ. μήκος φτερών) για να διαπιστώσει αν έχει όφελος ή όχι η τροποποίηση. Αυτός που θα επιλέξει το μοντέλο θα πρέπει να έχει γενική γνώση και να μην είναι εξειδικευμένος σε ένα θέμα μόνο. Για παράδειγμα, ένας σχεδιαστής βάσης δεδομένων που μοντελοποιεί ένα σύστημα, μπορεί να γνωρίζει μόνο μοντέλα οντοτήτων και εξαρτήσεων, ενώ ένας προγραμματιστής αντικειμενοστραφών γλωσσών, μπορεί να γνωρίζει μόνο για μοντέλα αντικειμένων και ιδιοτήτων.

Η δεύτερη αρχή σημαίνει ότι το ίδιο στοιχείο μπορεί να εκφραστεί με πολλά επίπεδα λεπτομέρειας. Από ένα απλό εκτελέσιμο που δέχεται κάποιους εισόδους και εμφανίζει στην έξοδο κάποια αποτελέσματα, έως το επίπεδο τρανζίστορ που δείχνει πως συμπεριφέρεται κάθε πύλη. Βέβαια, όσες πιο πολλές λεπτομέρειες έχει ένα μοντέλο, τόσο πιο δύσκολη είναι η ανάπτυξη και βελτίωση του. Θα πρέπει να επιλέγεται το κατάλληλο επίπεδο λεπτομέρειας κάθε φορά. Συνήθως, ακολουθείται μια σχεδιαστική προσέγγιση top-bottom και bottom-up. Δηλαδή, όταν απαιτείται ο σχεδιασμός ενός προϊόντος, αρχικά γνωρίζουμε την υψηλού επιπέδου περιγραφή, στη συνέχεια αρχίζουμε να κατεβαίνουμε προς το επίπεδο της υλοποίησης (top-bottom) για να εξάγουμε τα βασικά δομοστοιχεία που αυτή θα αποτελείται. Μετά την εξαγωγή ακολουθούμε μια πορεία

bottom-up, δηλαδή μοντελοποιούμε, κάθε δομοστοιχείο ξεχωριστά, δομοστοιχεία που έχουν μοντελοποιηθεί συνδέονται και μοντελοποιούν μεγαλύτερα, μέχρι να φτάσουμε στο ανώτερο επίπεδο που είναι το τελικό προϊόν.

Η τρίτη αρχή ερμηνεύεται ότι θα πρέπει τα μοντέλα που επιλέγονται να έχουν μεγάλη σχέση με την πραγματικότητα και να μη γίνονται αβάσιμες εκτιμήσεις λειτουργίας. Στα μοντέλα αυτά θα πρέπει να εξετάζονται και οι ακραίες συνθήκες και όχι μόνο οι τυπικές. Όλα τα μοντέλα απλοποιούν την πραγματικότητα. Θα πρέπει να προσέχουμε όμως, να μην υπερ-απλοποιούμε την πραγματικότητα καλύπτοντας σημαντικές πτυχές, που μπορεί να επηρεάσουν αρνητικά το προϊόν. Για παράδειγμα, αν μοντελοποιήσουμε ένα ΕΣ που θα σταλεί σε κάποιον δορυφόρο, με τον ίδιο τρόπο που θα μοντελοποιηθεί ένα παρόμοιο ΕΣ στη γη, τότε θα αγνοηθεί ο παράγοντας της κοσμικής ηλεκτρομαγνητικής ακτινοβολίας. Στη γη η κοσμική ακτινοβολία επηρεάζει ελάχιστα τα ΕΣ, ενώ στο διάστημα είναι πολύ έντονες οι συνέπειες της. Είναι κατανοητό, ότι κάποιος θα πρέπει να επιλέξει το μοντέλο ύστερα από έρευνα, ιδιαίτερα αν αφορά σχεδιασμό σε περιοχή που δεν έχει ασχοληθεί ξανά.

Η τέταρτη αρχή σημαίνει ότι για το ίδιο προϊόν απαιτούνται πολλά μοντέλα που δείχνουν μια διαφορετική εικόνα. Όπως σε ένα κατασκευαστικό έργο, υπάρχουν πολλά μοντέλα (ηλεκτρολογικό, στατικό, μηχανολογικό, θερμομονωτικό,...), έτσι και σε ένα έργο σχεδιασμού ενσωματωμένων συστημάτων, χρησιμοποιούνται πολλά μοντέλα που είναι ανεξάρτητα ως προς αυτό που εξετάζουν, αλλά συνδέονται με το τελικό προϊόν. Τυπικά μοντέλα που συναντώνται στα ΕΣ είναι το μοντέλο καταστάσεων χρήσης (use-cases) που περιγράφει τις λειτουργικές προδιαγραφές του συστήματος, το μοντέλο σχεδιασμού που παρουσιάζει τα φυσικά και ηλεκτρονικά δομοστοιχεία του κυκλώματος και πως συνδέονται, το μοντέλο υλοποίησης που δείχνει πως θα είναι το τελικό προϊόν, το μοντέλο λειτουργικής εμφάνισης, που δείχνει στον τελικό χρήστη πως θα επικοινωνεί μαζί του, και ασφαλώς μοντέλα περιγραφής προγράμματος, σχεδιασμού, και προσομοίωσης.

### 2.5 Γλώσσες Μοντελοποίησης

Στο σχεδιασμό ενσωματωμένων συστημάτων, υπάρχουν πολλές γλώσσες μοντελοποίησης. Οι πιο σημαντικές γλώσσες ή εργαλεία μοντελοποίησης που χρησιμοποιούνται, είναι οι παρακάτω:

- VHDL
- Verilog
- SystemC

- SystemVerilog
- UML
- MATLAB
- EDA TOOLS

## 2.6 Τρόποι Μοντελοποίησης

Τα εργαλεία που έχει στη διάθεσή του ένας μηχανικός για να μοντελοποιήσει ένα σύστημα συνήθως δέχονται διαφόρους τρόπους για την εισαγωγή της περιγραφής. Μπορεί να είναι είτε μια ειδική μορφή κειμένου σαν μια τυπική γλώσσα προγραμματισμού, όπως για παράδειγμα η VHDL ή η Verilog, είτε μια σχηματική μορφή που μοιάζει με διάγραμμα, όπως η UML, είτε πολύπλοκα περιβάλλοντα που έχουν και γραφική είσοδο και είσοδο σε μορφή προγράμματος, όπως το Matlab.

Τον τελευταίο καιρό έχουν παρουσιαστεί και ειδικά περιβάλλοντα μοντελοποίησης που παρέχουν τη δυνατότητα στη σχεδιαστική ομάδα να δημιουργήσει ένα αφαιρετικό λειτουργικό μοντέλο πριν ακόμη σχεδιαστεί το σύστημα. Αυτά τα περιβάλλοντα είναι ιδιαίτερα χρήσιμα σε αυτούς που σχεδιάζουν σύνθετα ΕΣ με διάφορους επεξεργαστές και δομοστοιχεία από πολλούς παρόχους, οπότε πριν αποφασίσουν και επιλέξουν αυτά τα στοιχεία, επιβεβαιώνουν τη συνεργασία τους. Ο σχεδιαστής μπορεί να περιγράψει τις οντότητες σε SystemC και να ορίσει τα πρότυπα επικοινωνίας. Το εργαλείο χρησιμοποιεί συνήθως μια υψηλού επιπέδου μοντελοποίηση, όπως το μοντέλο επιπέδου συναλλαγής (transaction level modeling, TLM), στο οποίο καθορίζονται τα δεδομένα που μεταφέρονται από μια οντότητα του συστήματος στην άλλη, και επίσης παρέχει μια βιβλιοθήκη από πολλά δομικά στοιχεία, όπως συγκεκριμένους επεξεργαστές, στοιχεία μνήμης, ελεγκτές, διαύλους και άλλα. Αφού τοποθετηθούν όλα τα απαραίτητα στοιχεία με τις περιγραφές λειτουργίας και τις διασυνδέσεις, δημιουργείται ένα εκτελέσιμο πρόγραμμα, το οποίο μπορεί να επιτύχει υψηλές ταχύτητες προσομοίωσης (έχουν αναφερθεί προσομοιώσεις μεγάλης λεπτομέρειας, με ταχύτητες που προσεγγίζουν το 40% της πραγματικής εκτέλεσης του συστήματος). Αυτό το πρόγραμμα δέχεται εισόδους και δείχνει με γραφικό τρόπο τα αποτελέσματα, όπως για παράδειγμα το γράφο κλήσεων συναρτήσεων, τις πληροφορίες καθυστέρησης των συναρτήσεων, τους χρόνους εξυπηρέτησης διακοπών, την κίνηση δεδομένων στους διαύλους και άλλα. Επειδή είναι εκτελέσιμο, ο μηχανικός μπορεί να επέμβει άμεσα και να τροποποιήσει παραμέτρους σε αυτό το μοντέλο μέσω μιας διεπαφής συμβατής με το GDB (gnu debugger) ή να αποσφαλματώσει και να διορθώσει τμήματα κώδικα με



## 2.7 Ανασκόπηση του Κεφαλαίου

Στις μέρες μας σχεδιάζονται σύνθετα και πολύπλοκα προϊόντα, ενώ ταυτόχρονα ο χρόνος εισαγωγής του προϊόντος στην αγορά (time-to-market) έχει μειωθεί και είναι περίπου 6 μήνες. Προκειμένου να τηρούν αυτό το μικρό χρόνο, οι σχεδιαστές πρέπει να μοντελοποιούν με διάφορους τρόπους και εργαλεία το τελικό προϊόν. Σε αυτό το κεφάλαιο περιγράψαμε τη σημαντικότητα της μοντελοποίησης του σχεδιασμού στα ενσωματωμένα συστήματα. Περιγράψαμε τις βασικές αρχές και τα επίπεδα περιγραφής, ενώ δώσαμε και μια ενδεικτική λίστα με τα εργαλεία και τις πιο σημαντικές γλώσσες μοντελοποίησης που πρέπει να γνωρίζει ένας μηχανικός που καλείται να μοντελοποιήσει ένα ενσωματωμένο σύστημα.