Φεσάκης Γ., Χαλάτσης, Κ., Καλαβάσης Φ. & Δημητρακοπούλου Α. (2000). Οι περιορισμοί του μοντέλου υπολογισμών Von Neumann στο μάθημα «Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον» και τα δίκτυα Petri. Στο (Επιμ.) Β. Κόμης, *Πρακτικά 2ου Συνεδρίου Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση*, Πάτρα Οκτώβριος 2000., Πανεπιστήμιο Πατρών, σελ. 136-144.

ΟΙ ΠΕΡΙΟΡΙΣΜΟΙ ΤΟΥ ΜΟΝΤΕΛΟΥ ΥΠΟΛΟΓΙΣΜΟΥ VON NEUMANN ΣΤΟ ΜΑΘΗΜΑ *"ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ"* ΚΑΙ ΤΑ ΔΙΚΤΥΑ PETRI.

Γεώργιος Φεσάκης
Καθ. Πληροφορικής ΠΕ19, Υπ. Διδάκτορας Π.Τ.Ν Πανεπιστημίου Αιγαίου
Γ. Γρίβα 23, Ρόδος, 85100
gfesakis@rhodes.aegean.gr

Κωνσταντίνος Χαλάτσης
Καθηγητής Τ. Πληροφορικής Ε.Κ. Πανεπιστημίου Αθηνών
Κτίρια ΤΥΠΑ, Πανεπιστημιούπολη Ζωγράφου
halatsis@di.uoa.gr

Φ. Καλαβάσης
Καθηγητής Π.Τ.Ν Πανεπιστημίου Αιγαίου
Λ. Δημοκρατίας 1, Ρόδος, 85100
kalabas@rhodes.aegean.gr

Α. Δημητρακοπούλου
Επίκουρος Καθηγήτρια Π.Τ.Ν Πανεπιστημίου Αιγαίου
Λ. Δημοκρατίας 1, Ρόδος, 85100
adimitr@rhodes.aegean.gr

ΠΛΗΡΟΦΟΡΙΕΣ: Γ. Φεσάκης
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: διδασκαλία προγραμματισμού, μοντελοποίηση συστημάτων, δίκτυα Petri, μοντέλο υπολογισμού von Neumann
ΘΕΜΑ: Παρουσιάσεις εργασιών νέων ερευνητών, **ΚΑΤΗΓΟΡΙΑ:** Πληροφορική στην Εκπαίδευση ή Προγράμματα Σπουδών στην Πληροφορική ή Διδακτική της Πληροφορικής

Περίληψη

Στην εργασία αυτή διατυπώνονται προβληματισμοί για το μάθημα "Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον" της τεχνολογικής κατεύθυνσης του Ενιαίου Λυκείου. Υποστηρίζεται ότι η αδυναμία του μοντέλου von Neumann, ως πλατφόρμα υλοποίησης της αλγοριθμικής μοντελοποίησης, να αναπαραστήσει με ευκολία παράλληλα, ασύγχρονα και κατανεμημένα συστήματα περιορίζει τις δυνατότητες επίτευξης των διδακτικών στόχων του μαθήματος. Για την αντιμετώπιση του τελευταίου προβλήματος προτείνεται η υιοθέτηση των περιβαλλόντων μοντελοποίησης γενικού σκοπού δικτύων Petri ως πλατφόρμα υλοποίησης του μαθήματος.

Abstract

This paper is dealing with the limits that are imposed by the procedural programming paradigm to the Greek Lyceum course "Application development in programming environment". The difficulties of the Von Neumann computing architecture to model parallel and/or distributed and/or asynchronous systems do not permit that kind of systems' use in teaching scenarios. In order to achieve the course goals the use of Petri-nets is proposed as a general purpose modeling environment covering a wider range of systems.

1. Εισαγωγή

Η αναγνώριση της εκπαιδευτικής αξίας της αλγοριθμικής μοντελοποίησης προβλημάτων έχει ως αποτέλεσμα την εμφάνιση αντίστοιχων μαθημάτων γενικής παιδείας στα σύγχρονα εκπαιδευτικά συστήματα. Στη χώρα μας διδάχθηκε για πρώτη φορά το σχολικό έτος 1999-2000 το μάθημα με τίτλο "Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον" στην τεχνολογική κατεύθυνση του ενιαίου λυκείου ως υποχρεωτικό μάθημα για τον κύκλο Πληροφορικής και Υπηρεσιών. Στην παρούσα εργασία θα παρατεθούν προβληματισμοί σχετικά με τους περιορισμούς που προκύπτουν από την επιλογή του μοντέλου υπολογισμού von Neumann ως πλατφόρμα μοντελοποίησης των αλγορίθμων.

- Θα παρουσιαστούν αρχικά οι στόχοι του μαθήματος όπως είναι διατυπωμένοι στο ενιαίο πρόγραμμα σπουδών και θα αναλυθούν ιδιαίτερα αυτοί που έρχονται σε σύγκρουση με την επιλογή του μοντέλου von Neuman.
- Για την ανάδειξη των προβληματισμών, θα δοθούν κατάλληλα παραδείγματα συστημάτων η προβληματική των οποίων δεν μπορεί να αντιμετωπισθεί με ευκολία από τις διαδικαστικές γλώσσες προγραμματισμού που συνοδεύουν το μοντέλο von Neumann.
- Κατόπιν θα παρουσιαστεί η προοπτική της μοντελοποίησης γενικού σκοπού που προτείνεται να εφαρμοστεί σε ένα μάθημα επίλυσης προβλημάτων γενικής παιδείας.
- Τελικά, θα προταθούν τα δίκτυα Petri ως ένα περιβάλλον μοντελοποίησης γενικού σκοπού που καλύπτει τις διαδικαστικές γλώσσες προγραμματισμού και μπορεί να υποστηρίξει καλύτερα τους διδακτικούς στόχους του μαθήματος.

2. Οι παιδαγωγικοί στόχοι του μαθήματος "Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον" και οι περιορισμοί που επιβάλει το μοντέλο von Neumann.

2.1 Παιδαγωγικοί στόχοι

Στο ενιαίο πρόγραμμα σπουδών που έχει εκδοθεί από το Παιδαγωγικό Ινστιτούτο αναφέρεται:

"Ο γενικός σκοπός του μαθήματος είναι, να αναπτύξουν οι μαθητές αναλυτική και συνθετική σκέψη, να αποκτήσουν ικανότητες μεθοδολογικού χαρακτήρα και να μπορούν να επιλύουν απλά προβλήματα σε προγραμματιστικό περιβάλλον."

Αναφέρεται επίσης και ο βασικός πυρήνας γνώσεων και δεξιοτήτων που πρέπει να αποκτήσουν οι μαθητές που θα παρακολουθήσουν επιτυχώς το μάθημα:

"*Οι μαθητές που θα έχουν παρακολουθήσει με επιτυχία το μάθημα, πρέπει να:*

1. *μπορούν να διακρίνουν και να αναγνωρίζουν προβλήματα και καταστάσεις που επιλύονται/αντιμετωπίζονται σε προγραμματιστικό περιβάλλον*
2. *μπορούν να αποφασίζουν σχετικά με την πολυπλοκότητα προβλημάτων και καταστάσεων*
3. *μπορούν να αναλύουν ένα απλό πρόβλημα και να σχεδιάζουν τη λύση του*
4. *έχουν αναπτύξει ικανότητες μοντελοποίησης και αλγοριθμικής επίλυσης προβλημάτων*
5. *μπορούν να χρησιμοποιούν συμβολικές μεθόδους για την επίλυση προβλημάτων και την επεξεργασία δεδομένων*
6. *μπορούν να χρησιμοποιούν σύγχρονα προγραμματιστικά εργαλεία για την υλοποίηση αλγορίθμων*
7. *μπορούν να προσδιορίζουν τους απαιτούμενους πόρους του συστήματος."*

Η αξία των παραπάνω στόχων θεωρείται στην εργασία αυτή δεδομένη. Για τον ενδιαφερόμενο αναγνώστη προτείνονται εργασίες όπως οι [1], [4], [5] και [8] για μια πιο τεκμηριωμένη υποστήριξη της παιδαγωγικής αξίας της αλγοριθμικής επίλυσης προβλημάτων. Στο [1] υποστηρίζεται ότι ο προγραμματισμός είναι ένας τρόπος έλεγχου ενός αναδομήσιμου μέσου όπως ο γραπτός λόγος αλλά με πολύ εκτεταμένα αλληλεπιδραστικά χαρακτηριστικά. Επομένως, ένα πρόγραμμα είναι μια εξωτερίκευση της σκέψης του μαθητή για τη λύση ενός προβλήματος. Η εκτέλεση του προγράμματος αντιστοιχεί σε έλεγχο εγκυρότητας της λύσης και η αποσφαλμάτωση, σε μέσο συνειδητοποίησης των λαθών της σκέψης (Ανατροφοδότηση). Στο [8] παρουσιάζεται συνοπτικά ο κριτικός θετικισμός όπως τον εννοεί ο φιλόσοφος της επιστήμης Karl Popper και η αξία της επίλυσης προβλημάτων και της μοντελοποίησης στην σύγχρονη επιστημολογία. Προτείνεται τέλος η επίσκεψη στον WWW κόμβο με διεύθυνση http://pespme1.vub.ac.be όπου παρουσιάζονται αναλυτικά η συστημική προσέγγιση και ο ριζοσπαστικός κονστρουκτιβισμός όπως τον εννοεί ο Ernst von Glaserfeld.

2.2 Το μοντέλο Von Neumann και ο διαδικαστικός προγραμματισμός

Το μοντέλο υπολογισμού που υλοποιείται από τη συντριπτική πλειοψηφία των ψηφιακών Η/Υ γενικού σκοπού μέχρι σήμερα είναι γνωστό με το όνομα "von Neumann architecture". Η μεγάλη διάδοση του συχνά αποκρύπτει την ύπαρξη άλλων μοντέλων υπολογισμού. Οι βασικές αρχές του περιλαμβάνουν [9]:

1. Ένα και μοναδικό στοιχείο υπολογισμών που ενσωματώνει επεξεργαστή, επικοινωνίες και μνήμη
2. Σταθερού μεγέθους κελιά μνήμης γραμμικά οργανωμένα
3. Χώρο διευθύνσεων ενός επιπέδου
4. Γλώσσα μηχανής χαμηλού επιπέδου (εντολές που εκτελούν απλές λειτουργίες σε στοιχειώδη ορίσματα)
5. Ακολουθιακός και κεντρικοποιημένος έλεγχος του υπολογισμού.

Το μοντέλο υπολογισμού von Neumann έχει δεχτεί αρκετή κριτική για τις δυσκολίες που επιβάλει στην εκμετάλλευση παράλληλων υπολογισμών. Οι δυσκολίες οφείλονται στο ένα και μοναδικό επεξεργαστικό στοιχείο και στον κεντρικοποιημένο και ακολουθιακό έλεγχο. Στο [2] υποστηρίζεται επιπλέον ότι το μοντέλο Von Neumann διατηρήθηκε παρά την άρση των τεχνολογικών χαρακτηριστικών που το καθιστούσαν αποδοτικό για την προστασία των επενδύσεων σε λογισμικό λόγω του κόστους εκπαίδευσης των προγραμματιστών. Πιο τεκμηριωμένη κριτική στο Von Neumann μοντέλο και εναλλακτικά μοντέλα υπολογισμού όπως το καθοδηγούμενο από τα δεδομένα (Data Driven) και το κατ' απαίτηση (Demand-Driven) μοντέλο περιγράφονται στο [9]. Η κριτική του μοντέλου Von Neumann είναι σήμερα περισσότερο επίκαιρη με την ύπαρξη μεγάλων κατανεμημένων συστημάτων όπως το Internet και τη συνειδητοποίηση των ορίων των σειριακών επεξεργαστών.

Τα υποδείγματα προγραμματισμού όμως έχουν γνωρίσει μεγάλη εξέλιξη, πέρα από το διαδικαστικό που υιοθετείται στο μάθημα. Πιο συγκεκριμένα:

- έχουν εμφανιστεί υποδείγματα (programming paradigms) όπως: προγραμματισμός καθοδηγούμενος από γεγονότα (event driven programming), εικονικός προγραμματισμός (visual programming), αντικειμενοστραφής προγραμματισμός (object oriented programming), λογικός-δηλωτικός προγραμματισμός (logic-declarative programming), προγραμματισμός με περιορισμούς (constraint programming) κ.α έτσι ακόμα και η ωφελιμιστική θεώρηση ενός κλασσικού μαθήματος προγραμματισμού μπορεί να αμφισβητηθεί.
- υπάρχει η τάση διεθνώς να δοθεί πρόσβαση στην υπολογιστική ισχύ των Η/Υ χωρίς να είναι αναγκαία η τριβή με τις λεπτομέρειες του προγραμματισμού. Στην κατεύθυνση αυτή έχουν αναπτυχθεί περιβάλλοντα μοντελοποίησης υψηλού επιπέδου (Δίκτυα Petris, PowerSim, Stella,) και γενικού σκοπού που επιτρέπουν στους χρήστες να μοντελοποιήσουν συστήματα και να επιλύσουν προβλήματα χωρίς την ιδιαίτερη γνώση τεχνικών λεπτομερειών.

Για πολλά από τα παραπάνω συστήματα μοντελοποίησης η αρχιτεκτονική von Neumann δε αποτελεί το φυσικότερο περιβάλλον υλοποίησης. Η δυνατότητα να προσομοιώνονται οι σημασιολογικές τους απαιτήσεις από τις σειριακές μηχανές μας επιτρέπει να χαρακτηρίσουμε την χρήση τους στην εκπαίδευση ως ρεαλιστικό στόχο.

2.3 Τα όρια της εκφραστικότητας των διαδικαστικών γλωσσών

Στη συνέχεια, θα αναδείξουμε τις αδυναμίες των διαδικαστικών γλωσσών να μοντελοποιήσουν με ευκολία συστήματα στα οποία εξελίσσονται παράλληλα γεγονότα ανεξάρτητα αλλά και με απαιτήσεις συγχρονισμού με μερικά απλά παραδείγματα.

Παράδειγμα 1. Το φιλοσοφικό συμπόσιο

Πέντε φιλόσοφοι κάθονται σε στρογγυλό τραπέζι και γευματίζουν μακαρόνια. Υπάρχει ένα πιάτο και ένα πιρούνι για κάθε φιλόσοφο. Κάθε φιλόσοφος εναλλάσσεται μεταξύ δύο καταστάσεων στη μία τρωει και στην άλλη σκέφτεται. Τα μακαρόνια γλιστρούν και για να φαει ένας φιλόσοφος πρέπει να χρησιμοποιήσει και το πιρούνι του διπλανού του. Όταν ένας φιλόσοφος πεινάσει προσπαθεί να πιάσει τα πιρούνια στα αριστερά και στα δεξιά του πιάνοντας ένα κάθε στιγμή με οποιαδήποτε σειρά. Αν τα καταφέρει να πιάσει και τα δύο πιρούνια, τότε τρωει για λίγο και κατόπιν τα αφήνει κάτω και συνεχίζει να σκέφτεται. Το ζήτημα είναι να γραφεί ένα πρόγραμμα που να περιγράφει τη συμπεριφορά κάθε φιλοσόφου και να μην οδηγεί σε αδιέξοδο (κατάσταση από την οποία δεν μπορούμε να μεταβούμε με κανένα τρόπο σε άλλη).

Το πρόβλημα αυτό προτάθηκε και επιλύθηκε το 1965 από τον Dijkstra για να αναδείξει τη λύση που είχε εφεύρει για το συγχρονισμό διεργασιών στα λειτουργικά συστήματα. Από τότε όλες οι αντιμετωπίσεις του προβλήματος του συγχρονισμού επιδεικνύονται με βάση αυτό το πρόβλημα.

Μια καλή περιγραφή του προβλήματος μαζί με την παρουσίαση της λύσης του μπορεί να βρεθεί στο [7]. Στο [7] παρουσιάζονται επίσης τα προβλήματα του αδιεξόδου (deadlock) της λιμοκτονίας (starvation) και της ελλιπούς απόδοσης της χρήσης σηματοφόρων για την λήψη κάθε πιρουνιού.

Οι δειπνούντες φιλόσοφοι αποτελούν χαρακτηριστικό παράδειγμα συστήματος παράλληλου, κατανεμημένου, ασύγχρονου με απαιτήσεις συγχρονισμού. Κάθε φιλόσοφος αποτελεί ένα ανεξάρτητο υποσύστημα με δικό του νήμα ελέγχου (ασύγχρονο υποσύστημα). Ο αλγόριθμος κάθε φιλοσόφου εκτελείται στο ίδιο υποσύστημα ανεξάρτητα από τα υπόλοιπα (κατανεμημένος

υπολογισμός). Είναι δυνατό δύο γεγονότα στο όλο σύστημα να συμβαίνουν ταυτόχρονα (παραλληλία). Οι φιλόσοφοι πρέπει να συγχρονίσουν ως ένα βαθμό τις ενέργειες τους, χωρίς τη βοήθεια κεντρικού ελέγχου ώστε να διαμοιραστούν τους κοινούς πόρους τους (πιρούνια) προκειμένου να αποφύγουν το αδιέξοδο και τη λιμοκτονία. Λίγη εμπειρία στον διαδικαστικό προγραμματισμό και/ή μια ανασκόπηση του [7] είναι αρκετή για να αντιληφθεί κανείς τη δυσκολία αντιμετώπισης του προβλήματος με τον προγραμματισμό κλασικού σειριακού Η/Υ.

Παράδειγμα 2. Το πρόβλημα του εκτελεστικού αποσπάσματος.

Στο [2] αναφέρεται το πρόβλημα του εκτελεστικού αποσπάσματος.

"Είστε ο επικεφαλής ενός εκτελεστικού αποσπάσματος που το αποτελούν στρατιώτες παρατεταγμένοι σε μια εξαιρετικά μεγάλη γραμμή, και πρέπει να τους δώσετε το παράγγελμα να πυροβολήσουν. Η γραμμή έχει τόσο μεγάλο μήκος ώστε εάν απλώς φωνάξετε "πυρ" δεν είναι δυνατό να σας ακούσουν όλοι. Έτσι δεν έχετε άλλη επιλογή παρά να δώσετε τη διαταγή σας στον πρώτο στρατιώτη και να του ζητήσετε να την επαναλάβει στον διπλανό του κ.ο.κ. Στο βάθος ακούγεται σταθερά ο ρυθμός ενός τύμπανου, αλλά δεν μπορείτε να διατάξετε τους στρατιώτες να πυροβολήσουν έπειτα από συγκεκριμένο αριθμό κτυπημάτων δεδομένου ότι αγνοείτε πόσοι στρατιώτες αποτελούν την γραμμή. Το πρόβλημα λοιπόν είναι να βρεθεί ένας τρόπος να πυροβολήσουν οι στρατιώτες ταυτόχρονα."

Το πρόβλημα περιγράφει ένα σύγχρονο κατανεμημένο σύστημα (κοινό τύμπανο) με περιορισμένες δυνατότητες επικοινωνίας. Η λύση του προβλήματος αυτού είναι πολύ ευκολότερη όταν κάθε στρατιώτης μοντελοποιηθεί ως μια μηχανή πεπερασμένων καταστάσεων που επικοινωνεί με τις διπλανές της (ο πρώτος και ο τελευταίος στρατιώτης μπορούν να διαφοροποιούνται).

Παραδείγματα από σχολικό βιβλίο και τη βιβλιογραφία

Άλλα προβλήματα που μπορούν να αναδείξουν τους περιορισμούς του μοντέλου von Neumann μπορούν να βρεθούν στο βιβλίο του μαθήματος "Τεχνολογία υπολογιστικών συστημάτων & Λειτουργικά Συστήματα" της Τεχνολογικής κατεύθυνσης του Ενιαίου Λυκείου, στο κεφάλαιο που αφορά στην αναπαράσταση παράλληλων διεργασιών με τους γράφους προβαδίσματος. Ως πιο πρακτικό παράδειγμα αναφέρεται η προσομοίωση τμήματος κυκλοφοριακού δικτύου μια πόλης (π.χ δύο συνεχόμενα σταυροδρόμια).

Στην βιβλιογραφία μπορεί κανείς να βρει πολλούς τρόπους αντιμετώπισης των παραπάνω προβλημάτων σε σειριακές μηχανές. Το πρόβλημα είναι ότι η προσομοίωση της παραλληλίας στις σειριακές μηχανές είναι μια πολυπλοκότητα που δεν δικαιολογείται στα πλαίσια ενός μαθήματος γενικής παιδείας στην επίλυση προβλημάτων με μοντελοποίηση. Επιπλέον τα παιδιά είναι συνηθισμένα να περιγράφουν σχέδια με παράλληλα γεγονότα, όταν για παράδειγμα σχεδιάζουν μια επίθεση σε ένα παιχνίδι μπάσκετ κ.α.

Επομένως το επιλεγμένο μοντέλο είναι περιοριστικό σχετικά με τους στόχους του μαθήματος. Συγκεκριμένα σε σχέση με τον πρώτο παιδαγωγικό στόχο είναι δυνατό οι μαθητές να μήν αναγνωρίζουν ως επιλύσιμα σε υπολογιστικό περιβάλλον προβλήματα που αφορούν κατανεμημένα και παράλληλα συστήματα. Οι ικανότητες μοντελοποίησης που θα αναπτύξει ένας μαθητής αφορούν το διαδικαστικό μόνο υπόδειγμα μοντελοποίησης και έτσι ο $4^{ος}$ στόχος δεν μπορεί να επιτευχθεί πλήρως.

3. Η μοντελοποίηση συστημάτων με τα δίκτυα Petri.

Υπάρχει όμως κάποιος τρόπος να επιτρέψουμε στα παιδιά να αποκτήσουν ικανότητες μοντελοποίησης και επίλυσης προβλημάτων με τυπικό τρόπο χωρίς την χρήση των διαδικαστικών γλωσσών προγραμματισμού και του μοντέλου von Neumann με τις εγγενής του αδυναμίες. Ο σκοπός της εργασία αυτής είναι να προτείνει ως καταλληλότερα για τους στόχους του μαθήματος τα γενικού σκοπού περιβάλλοντα μοντελοποίησης που βασίζονται σε κάποιο τυπικό μοντέλο όπως τα δίκτυα Petri.

3.1 Τι είναι τα δίκτυα Petri;

Λόγω του περιορισμένου της έκτασης της εργασίας δεν είναι δυνατό να δοθεί μια ολοκληρωμένη εισαγωγή στα δίκτυα Petri. Για τον σκοπό αυτό προτείνεται στον αναγνώστη το [3] και ο κόμβος http://www.daimi.aau.dk/PetriNets.

Ιστορικά τα δίκτυα Petri προέρχονται από τη διατριβή του Carl Adam Petri που υποβλήθηκε το 1962 στην σχολή μαθηματικών και φυσικής του τεχνικού πανεπιστημίου του Darmastadt της Γερμανίας. Τα δίκτυα Petri είναι ένα γραφικό και μαθηματικό εργαλείο μοντελοποίησης εφαρμόσιμο σε πληθώρα συστημάτων. Αποτελούν ένα πολλά υποσχόμενο εργαλείο για την περιγραφή και τη μελέτη συστημάτων επεξεργασίας πληροφορίας που χαρακτηρίζονται ως:

συνεξελισσόμενα (concurrent), ασύγχρονα (asynchronous), κατανεμημένα (distributed), παράλληλα (parallel), μη αιτιοκρατικά (non deterministic), και/η στοχαστικά (stochastic).

Η γραφική γλώσσα των δικτύων Petri μπορεί να χρησιμοποιηθεί σαν οπτικό μέσο επικοινωνίας παρόμοιο με τα διαγράμματα ροής (flow charts), τα διαγράμματα τμημάτων (block diagrams) και τα δίκτυα (networks). Επιπλέον, στα δίκτυα Petri χρησιμοποιούνται πεσσοί (tokens) για την προσομοίωση των δυναμικών και συνεξελισσόμενων δραστηριοτήτων των συστημάτων.

Σαν μαθηματικό εργαλείο, τα δίκτυα Petri, επιτρέπουν τον ορισμό εξισώσεων καταστάσεων (state equations), αλγεβρικών εξισώσεων και άλλων μαθηματικών μοντέλων για τη συμπεριφορά των περιγραφόμενων συστημάτων. Η μαθηματική βάση των δικτύων Petri επιτρέπει να αποδειχθούν τυπικά κάποια από τα χαρακτηριστικά των παραγόμενων μοντέλων, για παράδειγμα είναι δυνατό να αποδείξει κανείς ότι η λύση που προτείνει για το πρόβλημα των δειπνούντων φιλοσόφων δεν οδηγεί σε αδιέξοδο.

Για την αποδοτική χρήση των δικτύων Petri είναι επιβεβλημένη η χρήση λογισμικών εργαλείων. Είναι κοινή πρακτική κάθε ερευνητική ομάδα δικτύων Petri να ορίζει μια κλάση δικτύων Petri και να κατασκευάζει και τα δικά της εργαλεία για τη διαχείριση των δικτύων Petri.

3.2 Ορισμός των απλών δικτύωνPetri

Ένα **δίκτυο petri** PN είναι μια πεντάδα PN=(P,T,F,W,M0) όπου:

- P={p1,p2,...,pn} ένα πεπερασμένο και μη κενό σύνολο **θέσεων** (places)
- T={t1,t2,...,tn} ένα πεπερασμένο μη κενό σύνολο **μεταβάσεων** (transitions)
- F$\subseteq$ (PXT)$\cup$(TXP) ένα πεπερασμένο και μη κενό σύνολο **προσανατολισμένων συνδέσεων** (arcs)
- W: F$\rightarrow${1,2,3,....} μια συνάρτηση ανάθεσης **βαρών** στις συνδέσεις του PN.
- M0: P$\rightarrow${0,1,2,3,....} ένα αρχικό μαρκάρισμα των θέσεων του PN με **πεσσούς**.

  Έτσι ώστε P $\cap$ T=$\varnothing$.

Στη γραφική αναπαράσταση των δικτύων Petri οι θέσεις απεικονίζονται με κύκλους, οι μεταβάσεις με ορθογώνια οι συνδέσεις με προσανατολισμένες ακμές και οι πεσσοί με παχιές τελείες μέσα στις θέσεις.

Για να συμπληρωθεί ο ορισμός των δικτύων Petri πρέπει να συνοδευτεί και με τον κανόνα πυροδότησης.

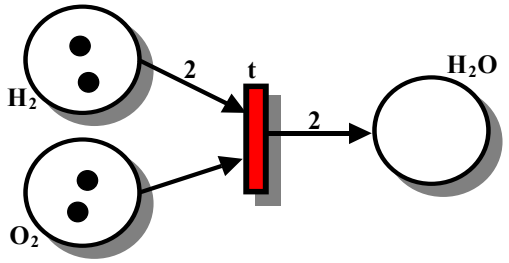
Ο κανόνας πυροδότησης

- *Η πυροδότηση μιας ενεργοποιημένης μετάβασης t αφαιρεί $w(p_i,t)$ πεσσούς από κάθε θέση εισόδου p_i της t και προσθέτει $w(t,p_o)$ πεσσούς σε κάθε θέση εξόδου της t, όπου $w(p_i,t)$ και $w(t,p_o)$ είναι τα βάρη στις αντίστοιχες συνδέσεις.*
- Μια μετάβαση t ονομάζεται **ενεργοποιημένη** όταν κάθε θέση εισόδου p_i της t περιέχει τουλάχιστον $w(p_i,t)$ πεσσούς, όπου $w(p_i,t)$ είναι το βάρος της σύνδεσης της θέσης p_i με την μετάβαση t.
- Μια ενεργοποιημένη μετάβαση δεν πυροδοτείται κατ' ανάγκη

Εφαρμόζοντας τον κανόνα πυροδότησης σε ένα δίκτυο Petri προσομοιώνουμε τη δυναμική του συμπεριφορά. Για να γίνουν πιο σαφή τα προηγούμενα θα παρουσιαστούν μερικά απλά παραδείγματα δικτύων Petri.

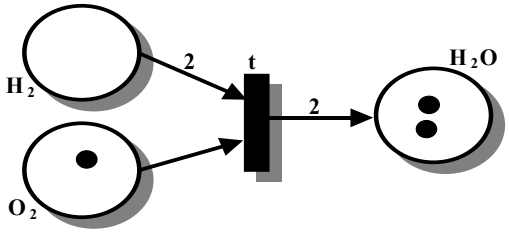
3.3 Παραδείγματα δικτύων Petri

Στο σχήμα 1 φαίνεται ένα δίκτυο Petri για τη χημική αντίδραση $2H_2+O_2-->2H_2O$. Το μοντέλο περιλαμβάνει: **1.** τις θέσεις: H_2, O_2 και H_2O που περιέχουν τις συγκεντρώσεις των αντίστοιχων αντιδρώντων και προϊόντων. **2.** Τη μετάβαση t που αντιστοιχεί στην παραγωγή ενός μορίου H_2O από 2 μόρια H_2 και ένα μόριο O_2. Η απαίτηση για δύο μόρια H_2 αναπαρίσταται με το βάρος με τιμή 2 στη σύνδεση από την θέση H_2 προς την μετάβαση t. **3.** Τα βάρη με τιμή 1 δεν εμφανίζονται ρητά πάνω στις αντίστοιχες συνδέσεις. **4.** Τρεις συνδέσεις (δύο συνδέσεις εισόδου στην t και μία εξόδου) **5.** Τέσσερις πεσσούς, δύο σε κάθε θέση εισόδου της t που σημαίνουν ότι υπάρχουν αρχικά από δύο μόρια H_2 και O_2 στα αντιδρώντα στοιχεία.



Σχήμα 1. Ένα δίκτυο Petri για τη μοντελοποίηση της χημικής αντίδρασης: $2H_2+O_2-->2H_2O$

Όσο αφορά στον κανόνα πυροδότησης, η μετάβαση t είναι ενεργοποιημένη γιατί υπάρχει ο απαραίτητος αριθμός πεσσών σε κάθε θέση εισόδου. Είναι δυνατό λοιπόν να πυροδοτηθεί και να προκύψει μετά την πυροδότηση το δίκτυο Petri του σχήματος 2.



Σχήμα 2. Το δίκτυο Petri του σχήματος 1 μετά την πυροδότηση της t, ένα μόριο νερού γεννήθηκε.

Στο δίκτυο Petri του σχήματος 2 δεν υπάρχει μετάβαση ενεργοποιημένη και κανένα γεγονός δε μπορεί πια να λάβει χώρα αφού η συγκέντρωση των αντιδρώντων δεν είναι αρκετή για την παραγωγή και άλλου μορίου νερού.

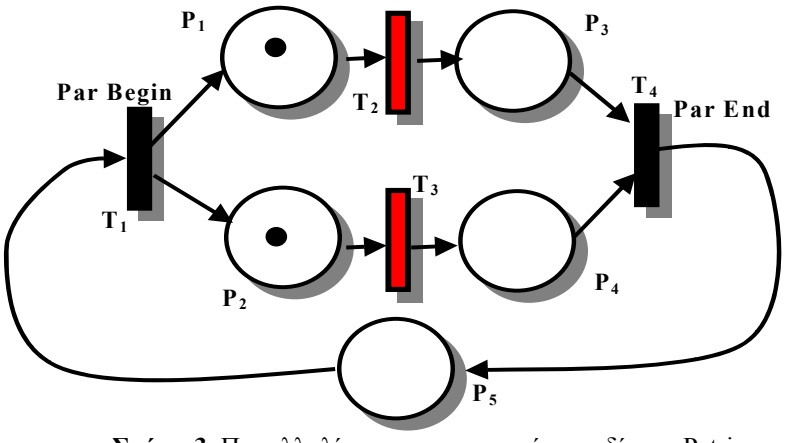
Κατά την χρήση των δικτύων Petri οι θέσεις και οι μεταβάσεις μπορούν να αντιστοιχούν σε διάφορες έννοιες όπως για παράδειγμα στον επόμενο πίνακα:

ΘΕΣΕΙΣ ΕΙΣΟΔΟΥ	ΜΕΤΑΒΑΣΕΙΣ	ΘΕΣΕΙΣ ΕΞΟΔΟΥ
Προϋποθέσεις (Precoditions)	Γεγονός (Event)	Νέες συνθήκες (Post conditions)
Δεδομένα εισόδου	Υπολογιστικό βήμα	Δεδομένα εξόδου
Σήματα εισόδου	Επεξεργαστής σήματος	Σήματα εξόδου
Απαιτούμενοι πόροι	Εργασία (Task, job)	Πόροι που διατίθενται
Συνθήκες	Λογική πρόταση	Συμπεράσματα
Ενδιάμεση μνήμη (Buffer)	Επεξεργαστής	Ενδιάμεση μνήμη (Buffer)

Πίνακας 1. Συνήθης χρήση αφαιρετικών συμβάσεων για μεταβάσεις και θέσεις.

3.4 Παραλληλία και συγχρονισμός

Στο σχήμα 3 αναπαρίσταται η υλοποίηση της δομής parabegin-parend που εισήγαγε ο Dijkstra για την περιγραφή παράλληλων προγραμμάτων. Τα δίκτυα Petri μπορούν να αναπαραστήσουν με ευκολία οποιαδήποτε σχέση γεγονότων στο χρόνο.

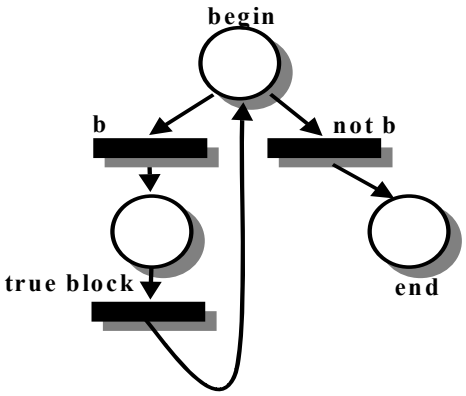


Σχήμα 3. Παραλληλία και συγχρονισμός στα δίκτυα Petri

3.5 Δίκτυα Petri και διαδικαστικός προγραμματισμός

Κάθε δομή του διαδικαστικού προγραμματισμού (ακολουθία, επιλογή, επανάληψη) μπορεί να αναπαρασταθεί με απλές δομές δικτύων Petri. Για παράδειγμα το επόμενο δίκτυο Petri αναπαριστά τη δομή «όσο <*συνθήκη*> επανέλαβε».

Είναι έτσι φανερό ότι κάθε πρόγραμμα που μπορεί να γραφτεί σε διαδικαστική γλώσσα προγραμματισμού (π.χ C, Pascal, Basic κλπ), μπορεί να διατυπωθεί και ως δίκτυο Petri.



Σχήμα 4. Η δομή ***όσο*** <*συνθήκη*> ***επανέλαβε.***

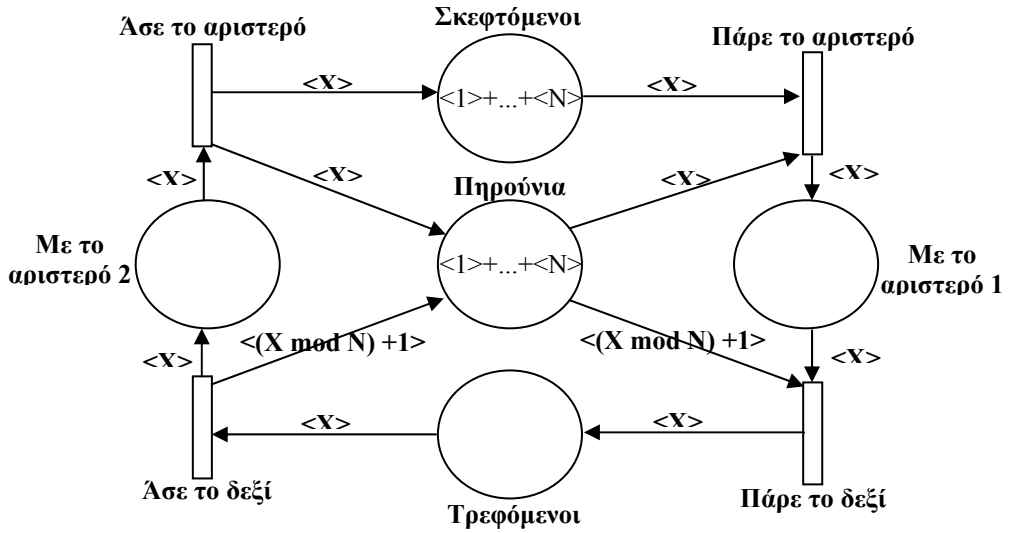
3.6 Μέθοδοι ανάλυσης

Για κάθε δίκτυο Petri ορίζονται διάφορα χαρακτηριστικά που είναι δυνατό να αναλυθούν με τυπικές μαθηματικές μεθόδους [3]. Μερικά από τα βασικά χαρακτηριστικά είναι: **1. προσβασιμότητα (reachability):** Ύπαρξη ακολουθίας πυροδοτήσεων που να οδηγεί σε κάποιο συγκεκριμένο μαρκάρισμα. **2.Ύπαρξη πάνω φράγματος (boundedness):** Ύπαρξη φράγματος στο πλήθος των πεσσών για συγκεκριμένες θέσεις **3. ζωντάνια (liveness):** ορίζονται διάφορα επίπεδα και αφορούν την δυνατότητα ενεργοποίησης κάποιων μεταβάσεων καθώς και την συχνότητα πυροδότησης. Ερωτήσεις που αφορούν αμοιβαίο αποκλεισμό και τα αδιέξοδα μεταφράζονται σε ερωτήσεις ζωντάνιας.

4. Η επίλυση του προβλήματος των φιλοσόφων με ένα δίκτυο Petri

Στην πράξη τα απλά δίκτυα Petri παράγουν μεγάλα μοντέλα ακόμα και για απλά συστήματα, για το λόγο αυτό χρησιμοποιούνται ισοδύναμες αλλά πιο εκφραστικές κλάσεις δικτύων Petri. Οι δειπνούντες φιλόσοφοι θα αναπαρασταθούν με το δίκτυο Petri κατηγορήματος-μετάβασης (Predicate-Transition net) του σχήματος 5. Στα δίκτυα αυτά οι πεσσοί μεταφέρουν δεδομένα, οι

μεταβάσεις όταν πυροδοτούνται εκτελούν διαδικαστικό κώδικα για τον υπολογισμό των πεσσών εξόδου και οι ακμές προσδιορίζονται με κατηγορήματα που περιγράφουν ιδιότητες που πρέπει να έχουν οι πεσσοί εισόδου προκειμένου μια μετάβαση να ενεργοποιηθεί. Η διαφορά με οποιαδήποτε λύση διαδικαστικού προγραμματισμού είναι φανερή. Επιπλέον είναι δυνατό χρησιμοποιώντας τις μεθόδους ανάλυσης των δικτύων Petri να αποδειχθεί ότι το προτεινόμενο σύστημα δεν είναι δυνατό να περιέλθει σε αδιέξοδο [6].



Σχήμα 5. Δίκτυο Petri για το πρόβλήμα των δειπνούντων φιλοσόφων από το εγχειρίδιο του συστήματος Prod (αρχικά και οι N φιλόσοφοι σκέφτονται)

5. Συμπεράσματα

Το μάθημα "Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον" είναι μια πρώτη προσπάθεια για τη διδασκαλία της μεθοδολογίας της μοντελοποίησης για την επίλυση προβλημάτων. Η σύγχρονη επιστημολογία έχει ανάγει την κατασκευή μοντέλων σε κεντρική επιστημονική δραστηριότητα που έρχεται να συμπληρώσει τις αναλυτικές μεθόδους. Η σκοπιμότητα του μαθήματος και η παιδαγωγική αξία των διδακτικών του στόχων συνάδουν με τις σύγχρονες κονστρουκτιβιστικές παιδαγωγικές και τη συστημική επιστημολογία.

Η επιλογή των διαδικαστικών γλωσσών προγραμματισμού ως πλατφόρμα υλοποίησης των παραγόμενων μοντέλων των μαθητών επιβάλει περιορισμούς στην ποικιλία των προβλημάτων που μπορούν να προσεγγιστούν. Μεγάλες κλάσεις συστημάτων όπως τα ασύγχρονα κατανεμημένα και παράλληλα συστήματα δεν προσεγγίζονται εύκολα με διαδικαστικό προγραμματισμό.

Για την άρση των παραπάνω περιορισμών προτείνεται η χρήση τυπικών περιβαλλόντων μοντελοποίησης γενικού σκοπού όπως τα δίκτυα Petri που δεν υποθέτουν το περιοριστικό σειριακό μοντέλο υπολογισμού von Neumann παρά το γεγονός ότι προσομοιώνονται από αυτό. Τα δίκτυα Petri: αποτελούν εργαλείο μοντελοποίησης και ανάλυσης με γραφικό και μαθηματικό τρόπο μεγάλου φάσματος πολύπλοκων συστημάτων. Υποστηρίζουν την ανάλυση του παραγόμενου μοντέλου με ισχυρά μαθηματικά εργαλεία όπως η θεωρία γραφημάτων και η άλγεβρα πινάκων. Επιτρέπουν την αλληλεπιδραστική προσομοίωση της δυναμικής συμπεριφορά του μοντελοποιημένου συστήματος Υπερκαλύπτουν τις δυνατότητες των διαδικαστικών γλωσσών προγραμματισμού και τέλος απαιτούν λογισμικό εργαλείο επεξεργασίας για την αποδοτική τους χρήση.

Η εργασία αυτή θα συνεχιστεί μελλοντικά με το σχεδιασμό και τη δοκιμή διδακτικών παρεμβάσεων για την χρήση των δικτύων Petri στο μάθημα "Ανάπτυξης εφαρμογών σε προγραμματιστικό περιβάλλον" αλλά και σε άλλα γνωστικά αντικείμενα. Βασικός στόχος είναι η ανάδειξη της επιστημολογικής αξίας των Η/Υ για την αυθόρμητη υιοθέτηση τους στη διδακτική και άλλων αντικειμένων όπως συμβαίνει στην τριτοβάθμια εκπαίδευση. Σε μια τέτοια προοπτική η μοντελοποίηση έχει κεντρικό ρόλο.

6. Βιβλιογραφία

1. DiSessa A. & Abelson H. (1986). BOXER: A Reconstructible computational medium, *Communications of the ACM*, V29, N9
2. Hillis D.(1999), *The pattern on the Stone*, Basic books (Ελληνική έκδοση με τίτλο: "Οι υπολογιστές στο παρόν και το μέλλον", μετάφραση Στ. Ζαχαρίου, Γ. Κατσιλιέρης, Α. Μαμαλης Εκδόσεις Κάτοπτρο, 2000)
3. Murata T. (1989), Petri nets: Properties, Analysis and Applications", *Proceedings of the IEEE*, vol 77, No. 4, APRIL 1989
4. Noss R. & Hoyles C. (1986). Synthesizing mathematical conceptions and their formalization through the construction of a LOGO-based school mathematics curriculum, International Journal of Mathematics Education in Science and Technology
5. Papert S., (1980). Mindstorms, The Harvester Press
6. Prod User's Guide, (1993), Helsinki University of Technology, Digital Systems Laboratory
7. Tanenbaum A., (1992), *Modern Operating Systems*, Prentice Hall Int.

8. Thornton S. (1997), *Karl Popper,* Stanford Encyclopedia of Philosophy
9. Treleaven P.BrownBridge D., Hopkins R.(1982), "Data-Driven and Demand-Driven Computer Architecture", *Computing Surveys of ACM*, Vol. 17, No. 1
10. Tsalgatidou A., Louridas P., Schizas T. and Fesakis G.. (1996). "Multilevel Petri Nets for Modeling and Simulating Organizational Dynamic Behavior", *Simulation & Gaming,* Special Issue on Simulation of Information Systems, Vol. 27, No. 4, Dec. 1996., pp. 484-506.