



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ**

**ICAGENT: Περιβάλλον Ανάπτυξης  
Νοημόνων Πρακτόρων  
για Δυναμικά και Απρόβλεπτα Περιβάλλοντα**

**Ευάγγελος Κουράκος Μαυρομιχάλης**  
Διατριβή για την απόκτηση διδακτορικού διπλώματος

**Πανεπιστήμιο Αιγαίου**  
**Τμήμα Μηχανικών Πληροφοριακών και**  
**Επικοινωνιακών Συστημάτων**

**Μάρτιος 2006**







## ΔΙΑΤΡΙΒΗ

Για την απόκτηση διδακτορικού διπλώματος  
του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων  
του Πανεπιστημίου Αιγαίου

Ευάγγελος Κουράκος Μαυρομιχάλης

### ICAGENT: Περιβάλλον Ανάπτυξης

### Νοημόνων Πρακτόρων

για Δυναμικά και Απρόβλεπτα Περιβάλλοντα

#### Επταμελής Επιτροπή

##### **Επιβλέπων:**

Γεώργιος Βούρος, Καθηγητής  
Τμήμα Πληροφοριακών και Επικοινωνιακών Συστημάτων  
Πανεπιστήμιο Αιγαίου

##### **Μέλη:**

Ιωάννης Βλαχάβας, Καθηγητής  
Τμήμα Πληροφορικής  
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Σωκράτης Κάτσικας, Καθηγητής  
Τμήμα Πληροφοριακών και Επικοινωνιακών Συστημάτων  
Πανεπιστημίο Αιγαίου

Πάνος Τραχανιάς, Καθηγητής  
Τμήμα Επιστήμης Υπολογιστών  
Πανεπιστήμιο Κρήτης

Μανόλης Κουμπάρκης, Αν. Καθηγητής  
Τμήμα Πληροφορικής και Τηλεπικοινωνιών  
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Διομήδης Σπινέλλης, Αν. Καθηγητής  
Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας  
Οικονομικό Πανεπιστήμιο Αθηνών

Κωνσταντίνος Στεργίου, Λέκτορας  
Τμήμα Πληροφοριακών και Επικοινωνιακών Συστημάτων  
Πανεπιστήμιο Αιγαίου









# Abstract

This thesis presents extended research on the development of intelligent collaborative agents for dynamic and unpredictable environments. In complex, resource bounded environments, agents face inherent limitations with respect to their perception, reasoning, performance and cooperation abilities: They cannot be fully aware of the changes that occur in their physical environment; the available time to compute responses is always limited and bounded to the time that they have until the resources get exhausted; they have to plan and to achieve their goals without exceeding environmental and own resources, and finally, they often need to collaborate with others to achieve better results.

The main objective of the thesis is to present a generic framework called ICAGENT (Intelligent Collaborative Agents), for developing intelligent collaborative agents capable to act efficiently in dynamic and unpredictable environments. To overcome the above limitations, ICAGENT provides a framework for the development of resource bounded agents that are able to deliberate, to plan and to act with respect to its own, others', and environment resources, adapting to unforeseen events and to newly detected facts, either individually or in collaboration with others. On the other hand, an agent needs to react to events/facts that provide opportunities for achieving its goals or to events/facts that may seriously affect its mission or missions of their collaborators. In either case, an agent developed in ICAGENT framework may perform individually or in conjunction with other agents balancing between individual reaction and deliberation as well as between social reaction (simple interaction) and social deliberation.

In contrast to other frameworks and agent architectures where agents' behaviour is determined by the mere mapping of domain actions to specific types of behaviours (layered architectures) without agents being able to adapt their behaviour selections to environmental changes, in ICAGENT framework agents' behavioural adaptation is considered as a "tuning" process, with respect to the way several mental actions are performed. Therefore, the type of behaviour that is adopted by an agent, at a specific time

point, is considered as a property that emerges as the agent performs, according to the perceived state of the environment, the occurring events, and to the agent's mental state.

ICAGENT framework (a) realizes a mental view of plan management and introduces a novel approach concerning the way an agent adapts its behaviour to environmental changes. Utilizing a BDI-based reasoning mechanism, mental actions are treated in coordination with the domain actions and their performance affects agents' behaviour according to its beliefs and intentions, (b) enable agents to collaborate with each other establishing shared goals and developing shared plans following the *SharedPlans* model of collaborative activity, and (c) provides facilities for agents to exhibit helpful behavior to application users in a trustworthy manner.

The ICAGENT framework has been realized a prototype and has been used for the development of collaborative agents for the dynamic environment *Tileworld*, as well as for developing intelligent collaborative applications' interface agents.

## Περίληψη

Η παρούσα διατριβή παρουσιάζει εκτενή έρευνα σε θέματα που αφορούν την ανάπτυξη πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα. Σε σύνθετα περιβάλλοντα που η κατάστασή τους αλλάζει συνεχώς και με απρόβλεπτο τρόπο, οι πράκτορες δεν έχουν τη δυνατότητα να γνωρίζουν πλήρως τα γεγονότα που συμβαίνουν σε αυτά αλλά ούτε και τις ενέργειες που πραγματοποιούνται από άλλους πράκτορες, διότι ο χρόνος επεξεργασίας των πληροφοριών που λαμβάνουν από το περιβάλλον τους, καθώς και ο χρόνος που χρειάζονται για να σχεδιάσουν και να υλοποιήσουν τα πλάνα τους, είναι περιορισμένος από το χρόνο εξάντλησης των πόρων τόσο του περιβάλλοντος όσο και δικών τους.

Αντικείμενο της παρούσας διατριβής είναι να παρουσιάσει ένα γενικευμένο πλαίσιο εργασίας (δηλαδή μίας αρχιτεκτονικής και ενός σύνολου δομών και διαδικασιών προγραμματισμού) ανάπτυξης νοημόνων συνεργατικών πρακτόρων (the ICAGENT framework) ικανών να δρουν αποτελεσματικά σε περιβάλλοντα δυναμικά και απρόβλεπτα. Για να μπορεί ένας πράκτορας να δρα αποτελεσματικά σε τέτοιου είδους περιβάλλοντα θα πρέπει να είναι ικανός να προσαρμόζει συνεχώς τη συμπεριφορά του στις αλλαγές του περιβάλλοντος και να συνεργάζεται με άλλους πράκτορες (ή/και με τον χρήστη) τόσο στο επίπεδο του σχεδιασμού των κοινών πλάνων όσο και κατά την υλοποίηση αυτών (εναλλάσσοντας συνεχώς τα δύο αυτά επίπεδα).

Σε αντίθεση με άλλα πλαίσια εργασίας όπου η συμπεριφορά των πρακτόρων είναι αποτέλεσμα μιας απλής αντιστοίχισης ενεργειών σε συγκεκριμένους τύπους συμπεριφοράς χωρίς μετέπειτα δυνατότητα προσαρμογής αυτής στις αλλαγές του περιβάλλοντος, το πλαίσιο εργασίας ICAGENT δίνει τις κατάλληλες δομές και διαδικασίες για την ανάπτυξη πρακτόρων ικανών ν' «ακολουθούν» τις αλλαγές του περιβάλλοντος προσαρμόζοντας το τρόπο με τον οποίο παρατηρούν το περιβάλλον τους, στοχάζονται (είτε μόνοι τους είτε σε συνεργασία με άλλους πράκτορες) και υλοποιούν τα πλάνα τους. Έτσι οι πράκτορες δρουν αποτελεσματικά ανάλογα με τους περιορισμούς και τις δυνατότητες που έχουν είτε ως άτομα ή ως ομάδες.

Η συνεισφορά του πλαισίου εργασίας ICAGENT έγκειται: (α) στο ότι εισάγει μία εναλλακτική ως προς άλλες αρχιτεκτονικές άποψη για το τρόπο με τον οποίο ένας πράκτορας προσαρμόζει τη συμπεριφορά του στις αλλαγές του φυσικού περιβάλλοντός μέσω ενός κοινού μηχανισμού χειρισμού των νοητικών ενεργειών που ελέγχουν το τρόπο με τον οποίο σχεδιάζει και υλοποιεί τα πλάνα του όσο και των ενεργειών που χρησιμοποιεί για την επίτευξη των στόχων του, (β) στη παροχή δυνατότητας ανάπτυξης πρακτόρων ικανών να συνεργάζονται μεταξύ τους αναπτύσσοντας κοινούς στόχους και πλάνα ακολουθώντας το μοντέλο συνεργασίας *SharedPlans* και (γ) στη παροχή δυνατότητας εδραίωσης σχέσεων εμπιστοσύνης μεταξύ συνεργαζόμενων μερών (για την παρούσα διατριβή μεταξύ χρηστών εφαρμογής λογισμικού και πρακτόρων διεπαφής) μέσω διαλόγων (κοινών πλάνων) που αναπτύσσονται από κοινού κατά τη διάρκεια της συνεργασίας.

Το πλαίσιο εργασίας ICAGENT έχει αναπτυχθεί ως ένα πρότυπο σύστημα και έχει χρησιμοποιηθεί για την ανάπτυξη συνεργατικών πρακτόρων για το περιβάλλον εξομίσωσης δυναμικών περιβαλλόντων *Tileworld* καθώς και για τη δημιουργία νοημόνων συνεργατικών πρακτόρων διεπαφής (*collaborative interface agents*) εφαρμογών λογισμικού. Η αρχιτεκτονική η οποία προτείνεται έχει υλοποιηθεί στη δηλωτική γλώσσα προγραμματισμού *SWI-Prolog* (*multi-thread version*), ενώ βασίζεται στη γνωσιακή επιστήμη και πιο συγκεκριμένα στη θεωρία προθετικών πρακτόρων (*BDI Agents*) και στο μοντέλο συνεργασίας *SharedPlans* των Barbara Grosz και Sarit Kraus.

## Ευχαριστίες

Η παρούσα διατριβή είναι αποτέλεσμα έρευνας στην οποία έχουν συνεισφέρει τόσο ερευνητικά όσο και ηθικά αρκετοί άνθρωποι τους οποίους νιώθω την υποχρέωση να ευχαριστήσω.

Πρώτα θα ήθελα να ευχαριστήσω τους γονείς μου και τον αδελφό μου για την υποστήριξη που μου προσέφεραν όλα αυτά τα χρόνια και για τη πίστη τους σε εμένα. Ιδιαίτερα θα ήθελα να ευχαριστήσω τη γυναίκα μου Ευγενία όχι μόνο για την ηθική υποστήριξη που μου προσέφερε αλλά αυτά τα χρόνια, αλλά και για το ιδιαίτερο νόημα που έδωσε στην εργασία μου η ύπαρξή της στη ζωή μου.

Η εργασία αυτή θα ήταν αδύνατον να ολοκληρωθεί χωρίς τη συνεχή υποστήριξη που μου προσέφερε όλα αυτά τα χρόνια ο επιβλέπων καθηγητής μου, Καθηγητής Γεώργιος Βούρος. Η προσφορά του ήταν καθοριστική όχι μόνο στην ολοκλήρωση της διατριβής αλλά και στο να μου δείξει τον δρόμο προς την επιστημονική ολοκλήρωση. Η υπομονή-επιμονή του για ουσιαστική έρευνα και ο τρόπος δουλειάς του που τον διακρίνει από άλλους επίσης άξιους επιστήμονες αποτελούσε και εξακολουθεί να αποτελεί για εμένα παράδειγμα. Μου δίδαξε το τρόπο με τον οποίο πρέπει ν' αναζητώ τη πληροφορία εκεί που οι άλλοι δεν έχουν ψάξει, αλλά και να την αξιοποιώ ανοίγοντας νέους (μικρούς ή μεγάλους) δρόμους στην ερευνητική περιοχή της διατριβής.

Τέλος, θα ήθελα να ευχαριστήσω θερμά όλα τα μέλη της ομάδας InCoSys για την άμεση ή έμμεση συνεισφορά τους στην ολοκλήρωση της διατριβής και ιδιαίτερα τον Ιωάννη Παρτσακουλάκη για τις ιδέες του και τις ατελείωτες συζητήσεις που είχαμε για ερευνητικά θέματα που και οι δύο ενδιαφερόμαστε.

Τη διατριβή αυτή την αφιερώνω στη γυναίκα μου Ευγενία και στο γιο μου Νικολάκη, που μόλις γεννήθηκε...

# Πίνακας Περιεχομένων

<b>Abstract</b> .....	<b>i</b>
<b>Περίληψη</b> .....	<b>iii</b>
<b>Ευχαριστίες</b> .....	<b>v</b>
<b>Πίνακας Περιεχομένων</b> .....	<b>vi</b>
<b>Κεφάλαιο 1ο: Εισαγωγή</b> .....	<b>1</b>
1.1. Ορισμός πράκτορα λογισμικού .....	2
1.2. Σύγκριση πρακτόρων και αντικειμένων .....	5
1.3. Ισχυρή και ασθενής θεώρηση πρακτόρων λογισμικού.....	6
1.4. Βασικές κατηγορίες αρχιτεκτονικών πρακτόρων λογισμικού.....	7
1.5. Νοήμονες πράκτορες.....	8
1.6. Συνεργασία μεταξύ νοημόνων πρακτόρων .....	10
1.7. Δομή της διατριβής .....	14
<b>Κεφάλαιο 2ο: Κίνητρα και Στόχοι της Έρευνας</b> .....	<b>16</b>
2.1. Το πρόβλημα ανάπτυξης πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα .....	19
2.2. Προσέγγιση και συνεισφορά στην ανάπτυξη πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα .....	21
2.3. Το πρόβλημα ανάπτυξης πρακτόρων για περιβάλλοντα επιφανειών διεπαφής.....	24
2.4. Προσέγγιση και συνεισφορά στην ανάπτυξη πρακτόρων επιφανειών διεπαφής	28
2.5. Εφαρμογές υλοποίησης .....	32
2.5.1. <i>Μελέτη περίπτωσης του δυναμικού φυσικού περιβάλλοντος Tileworld</i>	32
2.5.2. <i>Μελέτη περίπτωσης φυσικού περιβάλλοντος επιφάνειας διεπαφής</i> .....	38

<b>Κεφάλαιο 3ο: Σχετική Βιβλιογραφία .....</b>	<b>48</b>
3.1. Το BDI – μοντέλο ανάπτυξης πρακτόρων.....	48
3.2. Το μοντέλο συνεργασίας <i>SharedPlans</i> .....	52
3.2.1. Ορισμοί.....	53
3.2.2. Νοητικές δομές.....	55
3.2.3. Προσωπικά πλάνα.....	59
3.2.4. Κοινά πλάνα.....	59
3.3. Πλαίσια ανάπτυξης έξυπνων πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα.....	63
3.3.1. <i>dMARS</i> .....	63
3.3.2. <i>InteRRaP</i> .....	66
3.3.3. <i>STEAM</i> .....	68
3.3.4. <i>HAC</i> .....	74
3.3.5. <i>CCAF</i> .....	75
3.3.6. Σύγκριση πλαισίων ανάπτυξης έξυπνων πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα.....	77
3.3.7. Το προτεινόμενο πλαίσιο ανάπτυξης πρακτόρων <i>ICAGENT</i> .....	79
3.4. Πλαίσια ανάπτυξης έξυπνων συνεργατικών πρακτόρων επιφανιών διεπαφής.....	79
3.4.1. <i>Collagen</i> .....	81
3.4.2. <i>DIAL</i> .....	83
3.4.3. Σύγκριση πλαισίων ανάπτυξης έξυπνων πρακτόρων επιφανειών διεπαφής βασισμένων στο μοντέλο των <i>SharedPlans</i> .....	85
<b>Κεφάλαιο 4ο: Το Πλαίσιο Ανάπτυξης Νοημόνων Πρακτόρων ICAGENT .....</b>	<b>90</b>
4.1. Η βάση γνώσης.....	92
4.1.1. Νοητικές δομές.....	93
4.1.2. Αναπαράσταση των πλάνων .....	100
4.1.3. Ο μηχανισμός ενημέρωσης και εξαγωγής γνώσης .....	106
4.2. Οι διεργασίες συλλογισμού .....	110
4.2.1. Η διεργασία συλλογισμού της αντίληψης.....	112
4.2.2. Η διεργασία συλλογισμού της αναγνώρισης καταστάσεων.....	116

- 4.2.3. *Η διεργασία συλλογισμού του εντοπισμού και επίλυσης συγκρούσεων* 119  
 4.2.4. *Ανάπτυξη και υλοποίηση πλάνων*..... 126

**ΚΕΦΑΛΑΙΟ 5ο: Η Δραστηριότητα της Συνεργασίας στο Πλαίσιο Εργασίας του ICAGENT**..... 132

- 5.1. Αναγνώριση πιθανών ευκαιριών συνεργασίας..... 133  
 5.2. Σχηματισμός της ομάδας..... 135  
 5.3. Διαπραγμάτευση μεθόδων..... 137  
 5.4. Καταμερισμός εργασιών ..... 139  
 5.5. Υλοποίηση προθέσεων..... 142

**Κεφάλαιο 6ο: Ανάπτυξη Πρακτόρων για Περιβάλλοντα Επιφανειών Διεπαφής**145

- 6.1. Το γενικό πλαίσιο της δραστηριότητας της συνεργασίας του πράκτορα διεπαφής με τον χρήστη ..... 145  
 6.2. Αναγνώριση της ανάγκης για βοήθεια ..... 148  
 6.3. Σχηματισμός της ομάδας..... 150  
 6.4. Διαπραγμάτευση μεθόδων..... 150  
 6.5. Καταμερισμός εργασιών μεταξύ πράκτορα και χρήστη..... 151  
 6.6. Παρατηρώντας τις ενέργειες του χρήστη ..... 152  
 6.7. Κανόνες αναγνώρισης καταστάσεων ..... 153  
 6.8. Ερμηνεία ενεργειών του χρήστη ..... 154  
     6.8.1. *Σχηματίζοντας υποθέσεις για τους πρωταρχικούς στόχους του χρήστη*155  
     6.8.2. *Ερμηνεία ενεργειών χρήστη υπό το πλαίσιο ενεργειών ενός πρωταρχικού στόχου*..... 157  
 6.9. Ανάπτυξη ιεραρχικών πλάνων ..... 159  
 6.10. Παράδειγμα συνεργασίας μεταξύ πράκτορα διεπαφής και χρήστη..... 162

**Κεφάλαιο 7ο: Συμπεράσματα** ..... 171

- 7.1. Μελλοντική εργασία ..... 173

**Κεφάλαιο 8ο: Παραρτήματα**..... 175

- Περιγραφή κατευθύνσεων συμπεριφοράς** ..... 175  
**Μέθοδοι υλοποίησης νοητικών ενεργειών** ..... 177



---

<b>Παραδείγματα μεθόδων υλοποίησης ενεργειών για το περιβάλλον του Tileworld</b> .....	179
<b>Παραδείγματα μεθόδων υλοποίησης ενεργειών για το περιβάλλον διεπαφής της εφαρμογής Δημιουργός Μοντέλων</b> .....	182
<b>Παραδείγματα προδιαγραφών αισθητήρων για το πράκτορα διεπαφής της εφαρμογής Δημιουργός Μοντέλων</b> .....	184
<b>BNF–Περιγραφή των νοητικών δομών και μεθόδων του ICAGENT</b> .....	185
<b>Βιβλιογραφία</b> .....	<b>188</b>
<b>Γλωσσάρι</b> .....	<b>196</b>
<b>Ευρετήριο</b> .....	<b>200</b>







# Κεφάλαιο 1ο:

## Εισαγωγή

Τα τελευταία χρόνια με την αλματώδη ανάπτυξη του αντικειμενοστρεφούς προγραμματισμού, των συστημάτων αλληλεπίδρασης ανθρώπου-μηχανής, του παγκόσμιου διαδικτύου και γενικότερα των κατανεμημένων συστημάτων, δημιουργήθηκε η ανάγκη κατασκευής προγραμμάτων που να ενεργούν αυτόνομα και να εκτελούν εργασίες μέσα σε σύνθετα δυναμικά (και πολλές φορές απρόβλεπτα) περιβάλλοντα για λογαριασμό είτε κάποιου χρήστη είτε κάποιας άλλης οντότητας λογισμικού. Εργασίες όπως η παροχή βοήθειας σ' έναν άπειρο χρήστη, το φιλτράρισμα ηλεκτρονικών μηνυμάτων, η ανάκτηση πληροφορίας από το διαδίκτυο, η διαχείριση ελέγχου εναέριας κυκλοφορίας, η αναγνώριση λαθών σε διαστημικά οχήματα, η εξομοίωση αεροπορικών μαχητικών αποστολών, ο έλεγχος σύνθετων λειτουργιών όπως αυτών ενός robot σε ένα άγνωστο περιβάλλον ή σε μία γραμμή παραγωγής και άλλες παρόμοιες εργασίες [Odell, 2000], προϋποθέτουν προγράμματα που ο αντικειμενικός τους στόχος δεν είναι απλά να υπακούουν στις εντολές του χρήστη (εμφανίζοντας έναν «δουλοπρεπή» χαρακτήρα) αλλά να έχουν την ικανότητα ν' αντιλαμβάνονται τις αλλαγές του περιβάλλοντός τους, να συνεργάζονται με τον χρήστη ή μεταξύ τους και να παίρνουν πρωτοβουλίες επιδεικνύοντας ευφυή συμπεριφορά με σκοπό να ικανοποιήσουν τους στόχους τους. Τα προγράμματα αυτά ονομάζονται Πράκτορες Λογισμικού (Software Agents).

Η παρούσα διατριβή παρουσιάζει μία εκτενή έρευνα σε θέματα που αφορούν την ανάπτυξη πρακτόρων για περιβάλλοντα που η κατάστασή τους αλλάζει συνεχώς και με απρόβλεπτο τρόπο. Αντικείμενο της έρευνάς μας είναι ο σχεδιασμός και υλοποίηση ενός γενικευμένου πλαισίου εργασίας (ICAGENT framework) ανάπτυξης νοημόνων συνεργατικών πρακτόρων ικανών να δρουν αποτελεσματικά σε δυναμικά περιβάλλοντα και να συνεργάζονται τόσο μεταξύ τους όσο και με τους χρήστες.

Οι επιμέρους στόχοι της ερευνάς μας είναι:

- (α) Η ανάδειξη και η περιγραφή των προβλημάτων και περιορισμών που προκύπτουν από την εφαρμογή πλαισίων εργασίας και αρχιτεκτονικών ανάπτυξης πρακτόρων που έχουν προταθεί για δυναμικά περιβάλλοντα καθώς και για επιφάνειες διεπαφής.
- (β) Η ανάπτυξη μιας αρχιτεκτονικής και ενός πλαισίου εργασίας η οποία (εισάγοντας μία νοητική άποψη διαχείρισης των κατάλληλων δομών και διεργασιών) επιτρέπει τη δημιουργία πρακτόρων ικανών να σχεδιάζουν και να πραγματοποιούν πλάνα σε συνεργασία με άλλους πράκτορες και να προσαρμόζουν (adapt) τη συμπεριφορά τους ανάλογα με τις αλλαγές του περιβάλλοντός στο οποίο δρουν αντιδρώντας άλλοτε γρήγορα, χωρίς να σχεδιάζουν προσεκτικά τα πλάνα τους, και άλλοτε στοχαζόμενοι ν' αναπτύσσουν προσεκτικά το πλάνο κάθε ενέργειας που πραγματοποιούν.
- (γ) Η εφαρμογή του πλαισίου εργασίας ICAGENT για την ανάπτυξη νοημόνων πρακτόρων διεπαφής (intelligent interface agents) ικανών ν' αναγνωρίζουν τις προθέσεις και τις ανάγκες του χρήστη της διεπαφής, ν' σχηματίζουν κοινούς στόχους και να συνεργάζονται με τον χρήστη για την επίτευξη των στόχων αυτών κάνοντας χρήση διαλόγων (shared plans).

Για την επίτευξη των παραπάνω στόχων αναπτύχθηκε το πλαίσιο ανάπτυξης νοημόνων συνεργατικών πρακτόρων ICAGENT (Intelligent Collaborative Agent) βασίζεται στη γνωσιακή επιστήμη και πιο συγκεκριμένα στη θεωρία προθετικών πρακτόρων (BDI Agents) και στο μοντέλο συνεργατικών πρακτόρων *SharedPlans* των B. Grosz και S. Kraus, ενώ έχει υλοποιηθεί στη SWI-Prolog (multi-thread version).

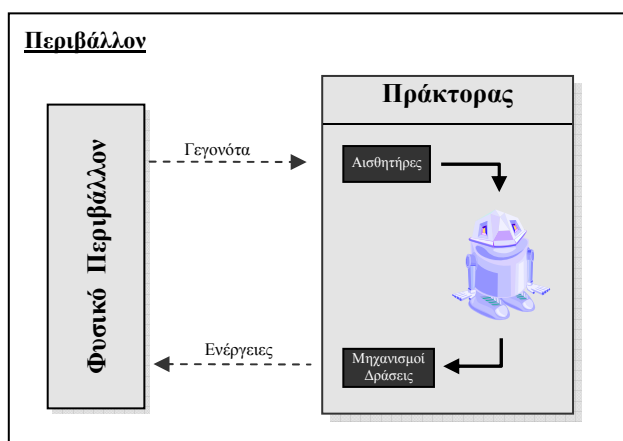
### 1.1. Ορισμός πράκτορα λογισμικού

Λόγω του ότι ο όρος «πράκτορας λογισμικού» έχει χρησιμοποιηθεί εκτενώς τόσο από τη βιομηχανία λογισμικού όσο και από την επιστημονική περιοχή της τεχνητής νοημοσύνης για το σχεδιασμό και τη κατασκευή έξυπνων συστημάτων αλλά και από τη γνωσιακή επιστήμη ως μοντέλο για τη κατανόηση της ανθρώπινης νόησης και συμπεριφοράς έχουν δοθεί διάφοροι ορισμοί για το *τί* είναι ένας πράκτορας. Παρακάτω παραθέτουμε μερικούς από τους πιο γνωστούς και γενικότερα αποδεκτούς ορισμούς στη περιοχή της τεχνητής νοημοσύνης:

**S. Russell – P. Norvig 2003:** «Ένας πράκτορας είναι ένα οποιοδήποτε πρόγραμμα το οποίο μπορεί να θεωρηθεί ότι έχει την ικανότητα να παρατηρεί το περιβάλλον του μέσω αισθητήρων (sensors) και να ενεργεί σε αυτό μέσω μηχανισμών δράσης (actuators)».

**M. Wooldridge 2000:** «Ένας πράκτορας είναι ένα πρόγραμμα ικανό να ενεργεί αυτόνομα σε κάποιο περιβάλλον».

**P. Maes 1995:** «Οι αυτόνομοι πράκτορες είναι υπολογιστικά συστήματα τα οποία «κατοικούν» σε σύνθετα δυναμικά περιβάλλοντα, έχουν την ικανότητα να παρατηρούν το περιβάλλον τους και ενεργούν μέσα σε αυτό αυτόνομα υλοποιώντας στόχους για τους οποίους έχουν σχεδιαστεί».



*Εικόνα 1:* Ένας πράκτορας έχει την ικανότητα να αντιλαμβάνεται το φυσικό περιβάλλον του μέσω των αισθητήρων που διαθέτει και να δρα αυτόνομα μέσα σε αυτό αλλάζοντας τη κατάσταση του μέσω μηχανισμών δράσης.

Παρ' όλο το πλήθος των ορισμών [Nwana, 1996] που έχουν δοθεί για το ποια προγράμματα μπορούν να χαρακτηριστούν ως πράκτορες είναι γενικά αποδεκτό ότι ένας πράκτορας λογισμικού θα πρέπει να έχει τουλάχιστον τις παρακάτω ιδιότητες (βλ. Εικόνα 1):

- (α) να «υπάρχει» σε ένα περιβάλλον,
- (β) να διαθέτει τους κατάλληλους μηχανισμούς με τους οποίους μπορεί να αντιλαμβάνεται τις αλλαγές του περιβάλλοντός του και να ενεργεί μέσα σε αυτό και
- (γ) να μπορεί να δρα αυτόνομα, δηλαδή χωρίς τη συνεχή παρέμβαση του χρήστη ή άλλου λογισμικού.

Ο πράκτορας αντιλαμβάνεται τη κατάσταση του περιβάλλοντός του καθώς και τις ενέργειες που πραγματοποιούνται από άλλους πράκτορες μέσα σε αυτό μέσω διακριτών μονάδων λογισμικού/υλικού οι οποίες υλοποιούν τους «αισθητήρες» του πράκτορα. Αφαιρετικά, ως πράκτορας μπορεί να θεωρηθεί οποιαδήποτε οντότητα λογισμικού η οποία λαμβάνει δεδομένα από τους αισθητήρες, τα επεξεργάζεται και ενεργεί μέσω μηχανισμών δράσης αλλάζοντας τη κατάσταση του φυσικού του περιβάλλοντος.

Γενικά ως *περιβάλλον* (environment) καθορίζεται το υλικό ή/και το λογισμικό εκείνο το οποίο παρέχει τις συνθήκες εκείνες κάτω από τις οποίες ένας πράκτορας ή ένα αντικείμενο υπάρχει. Ενώ ως *φυσικό περιβάλλον* (physical environment) είναι το υλικό ή/και το λογισμικό εκείνο το οποίο καθορίζει τις ιδιότητες και τις λειτουργίες του κόσμου μέσα στον οποίο ένας πράκτορας δρα, επικοινωνεί με τους άλλους πράκτορες και δέχεται δεδομένα από άλλα αντικείμενα ή πράκτορες [Odell, 2002].

<i>Περιβάλλον Εφαρμογής</i>	<i>Παρατηρήσιμο</i>	<i>Αιτιοκρατικό</i>	<i>Στατικό</i>	<i>Περιορισμένων Πόρων</i>	<i>Ανοικτό</i>
<b>Σταυρόλεξο</b>	Πλήρως	Αιτιοκρατικό	Στατικό	Όχι	Όχι
<b>Σκάκι</b>	Πλήρως	Στρατηγικό	Ήμιστατικό	Όχι	Όχι
<b>Γραμμή Παραγωγής</b>	Πλήρως	Αιτιοκρατικό	Δυναμικό	Ναι	Όχι
<b>Επιφάνεια Διεπαφής</b>	Πλήρως	Ήμιαιτιοκρατικό	Ήμιστατικό	Όχι	Όχι
<b>Διαδίκτυο</b>	Μερικώς	Στοχαστικό	Δυναμικό	Ναι	Ναι
<b>Tileworld</b>	Μερικώς	Στοχαστικό	Δυναμικό	Ναι	Ναι
<b>Robo-Soccer</b>	Μερικώς	Στοχαστικό	Δυναμικό	Ναι	Όχι

Πίνακας 1: Κατηγορίες φυσικών περιβαλλόντων.

Ανάλογα με τους νόμους που διέπουν ένα φυσικό περιβάλλον αυτό χαρακτηρίζεται σε [Russell and Norvig 2003]:

- μερικώς ή πλήρως παρατηρήσιμο (full or partial observable) ανάλογα με το αν το περιβάλλον δίνει τη δυνατότητα στο πράκτορα να έχει πρόσβαση μέσω των αισθητήρων του στη πλήρη κατάσταση του σε κάθε χρονική στιγμή ή μόνο σε ένα μέρος αυτού,



- αιτιοκρατικό ή στοχαστικό (deterministic or stochastic) ανάλογα με το αν ο πράκτορας μπορεί να προβλέψει τη κατάστασή του σε μια μελλοντική στιγμή,
- στατικό ή δυναμικό (static or dynamic) ανάλογα με το αν η κατάσταση του αλλάζει κατά τη διάρκεια του χρόνου χωρίς τη παρέμβαση κάποιου πράκτορα,
- περιορισμένων ή μη πόρων (resource bounded or not resource bounded) ανάλογα με το αν οι πόροι του φυσικού περιβάλλοντος είναι περιορισμένοι ή όχι,
- ανοικτό ή κλειστό (opened or closed) ανάλογα με το αν δίνει τη δυνατότητα στους πράκτορες να εισέρχονται και να εξέρχονται απ' αυτό οποιαδήποτε χρονική στιγμή.

Στο Πίνακα 1 δίνονται μερικά παραδείγματα περιβαλλόντων με τις παραπάνω ιδιότητες.

## 1.2. Σύγκριση πρακτόρων και αντικειμένων

Αφαιρετικά ο όρος «Πράκτορας Λογισμικού» μπορεί να ιδωθεί ως μία οντότητα, όπως ακριβώς αυτή του αντικείμενου στον αντικειμενοστρεφή προγραμματισμό [Odell, 2000; Shoham, 1990]. Σε ένα υψηλό επίπεδο αφαιρετισμού ένας πράκτορας μπορεί να περιγραφεί από ένα σύνολο ιδιοτήτων και μεθόδων (διαδικασιών) όπως ακριβώς ένα αντικείμενο. Η διαφορά ενός πράκτορα από ένα αντικείμενο είναι ότι το πρώτο μπορεί να περιγραφεί επιπλέον με όρους που χαρακτηρίζουν τη συμπεριφορά του όπως αυτονομία (autonomy), συνεργασία (cooperation), προσαρμοστικότητα (adaptability), μεταφερσιμότητα (mobility) και άλλων σχετικών συμπεριφορών [Odell, 2000].

Έτσι ο προγραμματισμός προσανατολισμένος σε πράκτορες παρέχει δυνατότητες περιγραφής σύνθετων οντοτήτων ικανών να δρουν σε δυναμικά περιβάλλοντα και να αλληλεπιδρούν είτε με άλλους πράκτορες (συμπεριλαμβανομένου και του χρήστη) είτε με αντικείμενα. Κατά συνέπεια, στη σχεδίαση λογισμικού, η έννοια του πράκτορα δεν έρχεται ν' αντικαταστήσει την έννοια του αντικείμενου. Ένα καλά σχεδιασμένο σύστημα λογισμικού βασισμένο σε πράκτορες χρησιμοποιεί τόσο τα αντικείμενα όσο και τους πράκτορες με τρόπο ισορροπημένο όπως αυτός υπάρχει και στο πραγματικό κόσμο [Odell, 2000].

### 1.3. Ισχυρή και ασθενής θεώρηση πρακτόρων λογισμικού

Στη σύγχρονη βιβλιογραφία της τεχνητής νοημοσύνης (TN) και γενικότερα της γνωστικής επιστήμης είναι γενικά αποδεκτή η ταύτιση της TN με το σχεδιασμό και τη κατασκευή νοημόνων πρακτόρων και πολύ-πρακτορικών συστημάτων [Russell and Norvig 2003; Nilsson 1998; Poole, Mackworth and Goebel 1997]. Όπως υπάρχουν δύο συμπληρωματικές θεωρήσεις για το τί είναι τεχνητή νοημοσύνη, όπου η μία την ορίζει ως μία επιστήμη με σκοπό τη κατασκευή έξυπνων συστημάτων και η δεύτερη ως μία εμπειρική επιστήμη η οποία έχει ως σκοπό την υπολογιστική μοντελοποίηση της ανθρώπινης νόησης και συμπεριφοράς, έτσι με παράλληλο τρόπο μπορεί να οριστεί η έννοια του πράκτορα με βάση την *ασθενή θεώρηση* ενός πράκτορα (weak notion of agency) και την *ισχυρή θεώρηση* ενός πράκτορα (strong notion of agency).

Η ασθενής θεώρηση ορίζει ότι ένας πράκτορας μπορεί να είναι ένα οποιοδήποτε λογισμικό ή υλικό το οποίο [Wooldridge, 2000]: (α) είναι αυτόνομο, (β) μπορεί να επικοινωνεί με άλλους πράκτορες (ή με τον χρήστη) χρησιμοποιώντας μία γλώσσα επικοινωνίας πρακτόρων (agent-communication language), (γ) έχει την ικανότητα να παρατηρεί το περιβάλλον του και ν' αντιδρά στις αλλαγές αυτού και (δ) να μπορεί να προνοεί (pro-activeness) δηλαδή να έχει τη δυνατότητα να λαμβάνει πρωτοβουλίες σύμφωνα με τους στόχους που διαθέτει.

Η ισχυρή θεώρηση ενός πράκτορα, επιπλέον των παραπάνω χαρακτηριστικών, δίνει έμφαση στο ότι το λογισμικό ή το υλικό διαθέτει και την ικανότητα συλλογιστικής ή/και άλλων χαρακτηριστικών που σχετίζονται με τη νοητική κατάσταση και συμπεριφορά των ανθρώπων. Τέτοια χαρακτηριστικά είναι η αναπαράσταση της εσωτερικής κατάστασης του πράκτορα χρησιμοποιώντας νοητικές δομές όπως η πεποίθηση, ο στόχος, η πρόθεση και η υποχρέωση καθώς και η χρήση διεργασιών συλλογισμού οι οποίες (κάνοντας χρήση των νοητικών δομών αυτών) καθιστούν τους πράκτορες ικανούς να παρακολουθούν τη κατάσταση του περιβάλλοντος (environment monitoring), να επιλέγουν τους στόχους που θέλουν να πετύχουν (option selection), ν' αναγνωρίζουν και να επιλύουν συγκρούσεις μεταξύ των στόχων που θέλουν να πετύχουν (reconciliation), ν' αναπτύσσουν πλάνα (plan elaboration) και να τα πραγματοποιούν (plan realization). Οι διεργασίες συλλογισμού καθορίζουν τη συμπεριφορά του πράκτορα όπως η προσαρμο-

στικότητα (adaptability), η ειλικρίνεια (veracity), η συναισθηματικότητα (emotionality) και η μεταφερσιμότητα (mobility)<sup>1</sup>.

#### 1.4. Βασικές κατηγορίες αρχιτεκτονικών πρακτόρων λογισμικού

Οι αρχιτεκτονικές υλοποίησης πρακτόρων (που έχουν προταθεί από τα μέσα της δεκαετίας του 80 μέχρι σήμερα) μπορούν να ομαδοποιηθούν σε δύο κύριες κατηγορίες: στους *πράκτορες που αντιδρούν αντανακλαστικά* (reactive agents) και στους *πράκτορες που σχεδιάζουν τα πλάνα τους* (deliberative agents).

Κύριο χαρακτηριστικό των πρακτόρων που αντιδρούν αντανακλαστικά είναι ότι δεν διαθέτουν κάποια εσωτερική συμβολική αναπαράσταση του κόσμου στον οποίο δρουν [Brooks, 1986]. Πρόκειται περισσότερο για προγράμματα που ακολουθούν την ασθενή θεώρηση του τι είναι ένας πράκτορας, βασίζονται σε κανόνες (rule-based programs) τόσο για να αποφασίσουν το πότε θα δράσουν όσο και για να πετύχουν τους στόχους τους και συμπεριφέρονται αντανακλαστικά στις αλλαγές του περιβάλλοντός τους. Κατασκευαστικά, κύριος στόχος των πρακτόρων αυτών είναι η γρήγορη απόκριση τους στις αλλαγές του φυσικού τους περιβάλλοντος όπως αυτό του RoboCup [Stone and Veloso, 1999] και των περιβαλλόντων εξομοίωσης πεδίων μάχης [Jones, 1999; Atkin et. al., 1999], ενώ χρησιμοποιούνται κυρίως για δουλειές ρουτίνας-μηχανικής φύσεως, όπως είναι η αναζήτηση πληροφοριών από μέσα αποθήκευσης ή από το διαδίκτυο, η απλή παροχή βοήθειας στο χρήστη χρησιμοποιώντας βάσεις δεδομένων, ως διαχειριστές ηλεκτρονικών ταχυδρομείων κ.τ.λ.

Όσο όμως η πολυπλοκότητα των συστημάτων που χρησιμοποιούν πράκτορες αυξάνει τόσο αυξάνονται και οι απαιτήσεις απ' αυτούς. Η χρήση νοητικών δομών για ν' αναπαρασταθεί η εσωτερική κατάσταση των πρακτόρων από τους [Bratman, 1987; Dennett 1989; Shoham 1990] και αργότερα από τους [Rao and Georgeff 1995] είχε ως

---

<sup>1</sup> Δεδομένου ότι το περιβάλλον ανάπτυξης (υλικό-λογισμικό) του πράκτορα επιτρέπει μία τέτοιου είδους συμπεριφορά.

αποτέλεσμα τη δημιουργία των *νοημόνων ή στοχαστικών*<sup>2,3,4</sup> πρακτόρων. Κύριο χαρακτηριστικό των πρακτόρων της κατηγορίας αυτής είναι η αναπαράσταση της εσωτερικής τους κατάστασης με χρήση νοητικών δομών, η ικανότητά τους να συλλογίζονται πρακτικά (practical reasoning) καθώς και η δυνατότητά τους να συνεργάζονται με άλλους πράκτορες όποτε αυτό είναι αναγκαίο ή θεμιτό.

Τέλος, υπάρχουν υβριδικές αρχιτεκτονικές οι οποίες κάνουν χρήση τόσο των νοητικών δομών για την αναπαράσταση της εσωτερικής κατάστασης των πρακτόρων ικανών να συλλογίζονται πρακτικά, όσο και κανόνων οι οποίοι υλοποιούν τους μηχανισμούς με τους οποίους οι πράκτορες επικοινωνούν με το φυσικό περιβάλλον και ενεργούν σε αυτό ανακλαστικά [Müller, 1996; Jung, 1999].

Για να ολοκληρωθεί η απάντηση για το τι είναι ένας νοήμων πράκτορας, θα πρέπει να δώσουμε απάντηση στο ερώτημα: ποιες οντότητες μπορούν να περιγραφούν με νοητικές δομές όπως πεποιθήσεις, δεσμεύσεις κ.τ.λ. καθώς και το τί είναι ο πρακτικός συλλογισμός.

## 1.5. Νοήμονες πράκτορες

Ο D. Dennett (1989) παρατηρεί ότι μπορούμε να προσδώσουμε σ' ένα σύστημα νοητικές ικανότητες, όπως προθέσεις και ελεύθερη βούληση, αν θεωρήσουμε ότι βρίσκεται σε μία «στάση πρόθεσης» (intentional stance). Δηλαδή, κατά το Dennett, το θέμα δεν

<sup>2</sup> Από λεξικό Μ. Τριανταφυλλίδη: **στοχαστικός** <sup>1</sup> -ή -ό [stoxastikós] Ε1 : 1α. που γίνεται με περίσκεψη, που είναι αποτέλεσμα περίσκεψης: *Τα λόγια του ήταν στοχαστικά*, συνετά. β. που φανερώνει περίσκεψη, βαθιά σκέψη και προβληματισμό: *Στοχαστική ματιά. Στοχαστικό βλέμμα*. 2. για πρόσωπο που στοχάζεται, που προσπαθεί με τη βαθιά σκέψη του να δώσει απάντηση στα διάφορα φαινόμενα, προβλήματα και ερωτηματικά. **στοχαστικά** ΕΙΠΡΡ. [αρχ. *στοχαστικός*].

<sup>3</sup> Από λεξικό Γ. Μπαμπινιώτη: **στοχάζομαι** (ρ) 1. σκέφτομαι βαθιά, κάνω συλλογισμούς, συλλογίζομαι, διανοούμαι 2. λαμβάνω υπ' όψιν μου, σκέφτομαι καλά: *πριν το κάνεις, στοχάσου τις συνέπειες (ζύγιζε πάντοτε προσεκτικά τα λόγια και τις πράξεις σου)* ΣΥΝ. υπολογίζω, λογαριάζω, σταθμίζω ζυγίζω. [ΕΤΥΜ. Αρχ., αρχική σημασία «σημαδεύω με το τόξο, στοχεύω», < στόχος. Η λέξη απέκτησε ήδη από την αρχαιότητα τη σημασία «κατευθύνω τις προσπάθειές και τις σκέψεις μου προς συγκεκριμένο στόχο, διανοούμαι»].

<sup>4</sup> Από την εγκυκλοπαίδεια Wikipedia: **Stochastic**, from the Greek "stochos" or "goal", means of, relating to, or characterized by conjecture; conjectural; random. A stochastic process is one whose behavior is non-deterministic in that the next state of the environment is not **fully** determined by the previous state of the environment.

είναι πότε ένα σύστημα διαθέτει ή όχι κάποια ικανότητα νόησης, αλλά πότε εμείς μπορούμε να το δούμε έτσι (βασιζόμενοι σε κάποια λογικά κριτήρια).

Το να αποδώσουμε σε ένα λογισμικό ή σε μία μηχανή, η οποία βρίσκεται σε μια συγκεκριμένη θέση και κατάσταση, νοητικές δομές (οι οποίες αναπαριστούν νοητικές ικανότητες ή κάποιες άλλες ισόμορφες ως προς αυτές έννοιες) είναι πολλές φορές λογικό αλλά όχι αναγκαίο. Γενικά είναι πιο εύκολο και λογικό να αποδώσουμε νοητικές ικανότητες σε μία μηχανή ή σε ένα λογισμικό που η δομή του είναι γνωστή. Από την άλλη μεριά όμως, είναι πιο χρήσιμο να αποδώσουμε νοητικές ικανότητες σε μία μηχανή ή σε ένα λογισμικό που ο τρόπος λειτουργίας του είναι εν μέρη άγνωστος.

Συγκεκριμένα ο J. McCarthy [McCarthy, 1979] διαχωρίζει τη *νομιμότητα* από τη *χρησιμότητα* στην απόδοση νοητικών δομών στις οντότητες: Λέμε ότι είναι νόμιμο να αποδώσουμε πεποιθήσεις, ελεύθερη βούληση, σκοπούς, συνείδηση, δυνατότητες ή επιθυμίες, σ' ένα μηχανήμα ή σ' ένα πρόγραμμα, όταν υπάρχει (ή μπορεί να υπάρξει) κάποιο πρόσωπο το οποίο έχει αυτές τις νοητικές δομές και οι πληροφορίες που προκύπτουν από αυτές, να εκφράζουν τα ίδια πράγματα για το πρόσωπο αυτό όπως και στο μηχανήμα ή πρόγραμμα. Αντίθετα, λέμε ότι είναι χρήσιμο να αποδώσουμε σε μια μηχανή ή πρόγραμμα τις παραπάνω νοητικές ικανότητες αν μας βοηθάει να καταλάβουμε καλύτερα τη δομή, τη λειτουργία, τη μελλοντική ή παρελθοντική συμπεριφορά του ή το πώς θα το βελτιώσουμε ή επισκευάσουμε.

Για παράδειγμα, ένας τυπικός κανόνας στα συστήματα βασισμένα στη γνώση είναι: «αν ο επεξεργαστής A δεν ξέρει αν ο επεξεργαστής B έχει λάβει το μήνυμά του, τότε ο επεξεργαστής A δε θα στείλει το επόμενο μήνυμα του στον B». Αντίθετα στο προγραμματισμό νοημόνων πρακτόρων οι νοητικές δομές εμπλουτίζουν τέτοιες λογικές προτάσεις με τις έννοιες των επιλογών, ικανοτήτων, δεσμεύσεων και πιθανόν άλλες. Έτσι ένας κανόνας παρόμοιος με τον παραπάνω μπορεί τώρα να γίνει: «αν ο πράκτορας A πιστεύει ότι ο πράκτορας B έχει δεσμευθεί να πραγματοποιήσει την ενέργεια P, τότε ο A δεν θα ζητήσει από τον B να αλλάξει την επιλογή του».

Η περιγραφή της εσωτερικής κατάστασης ενός πράκτορα με χρήση νοητικών δομών μας επιτρέπει να υλοποιήσουμε πράκτορες που οι ενέργειές τους είναι αποτέλεσμα συλλογισμού (πρακτικής συλλογιστικής) και όχι απλής εφαρμογής κανόνων (για παράδειγμα χρησιμοποιώντας κάποιο μαθηματικό μοντέλο). Αυτό είναι ιδιαίτερα σημαντικό

σε περιβάλλοντα που η κατάσταση τους αλλάζει συνεχώς και με μη-αιτιοκρατικό τρόπο με αποτέλεσμα η συμπεριφορά του πράκτορα να μην μπορεί να περιγραφεί αποτελεσματικά μέσω κανόνων. Η χρήση κανόνων προϋποθέτει τόσο γνώση των πιθανών καταστάσεων που μπορεί να βρεθεί το περιβάλλον όσο και πλήρη γνώση των αποτελεσμάτων των ενεργειών των πρακτόρων κατά τη διάρκεια του σχεδιασμού των κανόνων. Κάτι τέτοιο όμως σε δυναμικά και μη-αιτιοκρατικά περιβάλλοντα δεν είναι πάντα εφικτό.

Εδώ είναι σημαντικό να τονίσουμε τη διαφορά μεταξύ της *πρακτικής συλλογιστικής* (practical reasoning) [Bratman, 1987] και της *θεωρητικής συλλογιστικής* (theoretical reasoning) [Eliasmith, 2005].

Η θεωρητική συλλογιστική κατευθύνεται αποκλειστικά από τις πεποιθήσεις του πράκτορα. Για παράδειγμα, «αν ο πράκτορας πιστεύει ότι ο Σωκράτης είναι άνθρωπος τότε πιστεύει ότι είναι και θνητός». Αντίθετα η πρακτική συλλογιστική κατευθύνεται όχι μόνο από τις πεποιθήσεις του πράκτορα αλλά και από τις ενέργειες που αυτός θέλει να πραγματοποιήσει. Για παράδειγμα, «ο πράκτορας επιθυμεί να πάει στη τράπεζα αλλά δεν επιλέγει να το κάνει τούτο διότι έχει τη πεποίθηση ότι το αυτοκίνητο του δεν έχει αρκετή βενζίνη». Στο παράδειγμα αυτό η πρακτική συλλογιστική του πράκτορα τον οδηγεί να μην επιλέξει να πραγματοποιήσει μία ενέργεια (να πάει στη τράπεζα) λόγω ενός περιορισμού (η βενζίνη που διαθέτει δεν είναι αρκετή).

Στη γνωσιακή επιστήμη η πρακτική συλλογιστική αποτελείται από δύο διακριτές διεργασίες: Η πρώτη διεργασία αφορά την απόφαση του πράκτορα σχετικά με το *τί* ενέργειες θέλει να πραγματοποιήσει ή *ποιες* καταστάσεις του φυσικού ή νοητικού κόσμου του θέλει να πετύχει. Αυτή η διεργασία ονομάζεται ατομικός στοχασμός (individual deliberation). Η δεύτερη διεργασία αφορά το *πώς* ο πράκτορας μπορεί να πραγματοποιήσει τις ενέργειες αυτές ή το *πώς* μπορεί να πετύχει τις καταστάσεις αυτές και ονομάζεται ανάλυση μέσων και σκοπών (means-end reasoning).

## 1.6. Συνεργασία μεταξύ νοημόνων πρακτόρων

Ένα ακόμα επιθυμητό χαρακτηριστικό των νοημόνων πρακτόρων είναι η δυνατότητα τους να συνεργάζονται με άλλους πράκτορες (ή με το χρήστη). Γενικά, διακρίνουμε τρία μοντέλα συν-εργασίας ανάλογα με τη γνώση που έχει ο κάθε πράκτορας για τη

νοητική κατάσταση των συνεργατών του [Grosz, 1994; Falcone and Castelfranchi, 2000; Parunak, 2004, Reed and Long 1997]:

- Το πρώτο μοντέλο είναι αυτό της απλής ανάθεσης εργασιών (delegation). Στο μοντέλο αυτό ο πράκτορας επικοινωνεί με έναν ή περισσότερους πράκτορες και απλά αναθέτει εργασίες σ' αυτούς. Η απλή αυτή περίπτωση συνεργασίας δεν προϋποθέτει κάποιο κοινό στόχο. Ο πράκτορας που θ' αναλάβει να πραγματοποιήσει την ενέργεια ή το στόχο είναι ο μόνος υπεύθυνος για την υλοποίηση του ενώ ο πράκτορας που αναθέτει την ενέργεια δεν δεσμεύεται ότι θα βοηθήσει. Προϋπόθεση για ν' αναθέσει ένας πράκτορας σε άλλον μία ενέργεια είναι να τον εμπιστευτεί, δηλαδή να γνωρίζει ότι μπορεί να πραγματοποιήσει την ενέργεια και ότι πράγματι θα αναλάβει να την κάνει.
- Το δεύτερο μοντέλο είναι αυτό της συνέργειας ή της σύμπραξης (cooperation). Στο μοντέλο αυτό ο πράκτορας εργάζεται με έναν ή περισσότερους πράκτορες για να πετύχουν έναν κοινό στόχο. Κάθε ένας πράκτορας που δέχεται να συνεργαστεί μπορεί να μοιράζεται με τους άλλους ένα κοινό πλάνο (όχι κατ' ανάγκη πλήρες) χωρίς όμως να δεσμεύεται ότι οι συνεργάτες του θα έχουν πάντα τη δυνατότητα να πραγματοποιήσουν τις ενέργειες που έχουν αναλάβει.
- Το τρίτο μοντέλο είναι αυτό του κοινωνικού στοχασμού (social deliberation). Στο επίπεδο αυτό ο κάθε ένας πράκτορας στοχάζεται (deliberate) μαζί με τους συνεργάτες του όχι μόνο για το ΠΙ θα πραγματοποιήσουν μαζί (δηλαδή ποιους στόχους ή ποιες ενέργειες) αλλά και το ΠΩΣ ενώ ταυτόχρονα δεσμεύονται για την επιτυχή έκβαση των στόχων ή των ενεργειών των συνεργατών τους. Όπως θα διαπιστώσουμε παρακάτω, τέτοιου είδους δεσμεύσεις οδηγούν τους πράκτορες να βοηθούν τους συνεργάτες τους όποτε αυτό είναι αναγκαίο.

Σε αντίθεση με το συνεργατικό μοντέλο ανάθεσης εργασιών και της σύμπραξης δύο ή περισσότερων πρακτόρων για την επίτευξη ενός κοινού στόχου· το μοντέλο του κοινωνικού στοχασμού προϋποθέτει κάτι παραπάνω: γνώση των προθέσεων των συνεργατών και δέσμευση των πρακτόρων όχι μόνο για την επίτευξη του κοινού στόχου αλλά και για το ότι οι συνεργάτες θα πετύχουν τους στόχους τους. Η γνώση αυτή είναι αναγκαία ώστε οι πράκτορες να μπορούν να συνεργάζονται όχι μόνο στο επίπεδο εκτέλεσης ενεργειών αλλά και στο επίπεδο ανάπτυξης του κοινού πλάνου. Η δέσμευση κάθε

πράκτορα ότι οι συνεργάτες του θα πετύχουν τους στόχους τους έχει το εξής αποτέλεσμα σε ότι αφορά τη συμπεριφορά των πρακτόρων: ο κάθε πράκτορας είτε βοηθά τους συνεργάτες του εκτελώντας κάποιες ενέργειες που συμβάλουν στη πραγματοποίηση των στόχων τους είτε απλώς προσέχει το τρόπο με τον οποίο ενεργεί έτσι ώστε να μην εμποδίζει τις ενέργειες των συνεργατών του.

Για να κατανοήσουμε καλύτερα των πρώτο μοντέλο συνεργασίας σε σχέση με το τρίτο ας πάρουμε για παράδειγμα τους δύο πράκτορες της Εικόνα 2 [Grosz, 1994].



Εικόνα 2: Συνεργασία μεταξύ πρακτόρων.

Στην εικόνα αυτή οι δύο πράκτορες συνεργάζονται προκειμένου να βρουν το δρόμο τους προς την έξοδο του λαβυρίνθου. Παρατηρώντας μόνο την εικόνα αυτή είναι αδύνατον να μιλήσουμε για το τύπο της συνεργασίας τους. Θα μπορούσε ο πρώτος (αυτός που βρίσκεται από πάνω) να δίνει οδηγίες στο δεύτερο προκειμένου να τον μεταφέρει προς τα εκεί που επιθυμεί. Σ' αυτή τη περίπτωση δεν υπάρχει ούτε κοινός στόχος ούτε κοινό πλάνο. Αν για παράδειγμα ο ένας πράκτορας ανατραπεί τότε ο δεύτερος δεν θα τον βοηθήσει. Απλά, πάντα ο πρώτος πράκτορας αναθέτει ενέργειες στο δεύτερο χωρίς όμως ο δεύτερος να ενδιαφέρεται για τους στόχους του πρώτου (δηλαδή για τους λόγους τους οποίους του αναθέτει τις ενέργειες).

Αν όμως οι δύο πράκτορες μοιράζονται τον ίδιο στόχο και δεσμεύεται ο καθένας όχι μόνο για την επίτευξη του κοινού στόχου αλλά και για την επιτυχή πραγματοποίηση των ενεργειών του συνεργάτη του τότε η συμπεριφορά τού κάθε πράκτορα αλλάζει σε ότι αφορά τη βοήθεια που παρέχει. Για παράδειγμα αν ο πρώτος πράκτορας θέλει να δει πίσω από έναν τοίχο τότε ο δεύτερος δεν θα τον τοποθετήσει μόνο κοντά στον τοίχο αλλά θα τον πάει σε τέτοιο σημείο που να του επιτρέπει να έχει καλύτερο οπτικό πεδίο.



Για να κατανοήσουμε καλύτερα τη διαφορά του δεύτερου μοντέλου συνεργασίας, δηλαδή αυτού της σύμπραξης, με αυτό του κοινωνικού στοχασμού ας πάρουμε το παράδειγμα δύο πρακτόρων οι οποίοι έχουν ως στόχο να βάνουν ένα σπίτι. Οι δύο πράκτορες μπορούν να εργάζονται μαζί έχοντας τον ίδιο στόχο (να βάνουν το σπίτι) και το ίδιο πλάνο (ο πρώτος πράκτορας να βάνει τον πρώτο όροφο και ο δεύτερος τον δεύτερο όροφο). Αν όμως οι δύο πράκτορες χρησιμοποιούν τους ίδιους πόρους (όπως για παράδειγμα ίδια πινέλα) τότε μπορεί να δημιουργηθεί σύγκρουση μεταξύ των ενεργειών των δύο πρακτόρων. Το πρόβλημα αυτό δημιουργείται λόγω του ότι το περιβάλλον αλλάζει δυναμικά και ο κάθε πράκτορας έχει μερική γνώση (όπως συμβαίνει συνήθως σε πραγματικά προβλήματα) για τα πλάνα του συνεργάτη του.

Όπως τονίσαμε παραπάνω, αναγκαία προϋπόθεση για να συνεργάζονται αποτελεσματικά δύο ή περισσότεροι πράκτορες σε ένα δυναμικό περιβάλλον με περιορισμένους πόρους είναι οι πράκτορες να γνωρίζουν τις προθέσεις των συνεργατών τους και να δεσμεύονται για την επιτυχή έκβασή τους και όχι απλώς να εργάζονται κάτω από ένα κοινό πλάνο (όπως γίνεται στο δεύτερο μοντέλο συνεργασίας).

Πιο συγκεκριμένα, για να είναι επιτυχές ένα μοντέλο συνεργασίας σε ένα δυναμικό και απρόβλεπτο περιβάλλον με περιορισμένους πόρους όπου ο κάθε πράκτορας έχει διαφορετικές πεποιθήσεις, ικανότητες και δυνατότητες<sup>5</sup>, θα πρέπει [Grosz, 1994]:

- (α) Ο κάθε πράκτορας να γνωρίζει τις ικανότητες των συνεργατών του και ν' αναγνωρίζει τις προθέσεις τους. Η επικοινωνία μεταξύ των πρακτόρων είναι αναγκαία για τη δημιουργία του κοινού πλαισίου κάτω από το οποίο πραγματοποιούν τις ενέργειες τους (context of action). Το πλαίσιο αυτό περιλαμβάνει πληροφορίες όχι μόνο για τις ενέργειες που ο καθένας έχει δεσμευτεί να πραγματοποιήσει αλλά και πληροφορίες σχετικά με τους περιορισμούς που οι ενέργειες αυτές θέτουν (context constraints) καθώς και για τις δυνατότητες των πρακτόρων σε ότι αφορά την εκτέλεση των ενεργειών αυτών (applicability conditions).

---

<sup>5</sup> Ως ικανότητα ενός πράκτορα ορίζεται η νοητική δομή εκείνη η οποία αναπαριστά το τι μπορεί να πραγματοποιήσει (δηλαδή ποιο τύπο ενέργειας), ενώ ως δυνατότητα ορίζεται το σύνολο το περιορισμών εκείνων που πρέπει να ισχύουν για να μπορεί ο πράκτορας να πραγματοποιήσει μία ενέργεια.

- (β) Οι πράκτορες θα πρέπει να έχουν την ικανότητα να συνεργάζονται όχι μόνο στο επίπεδο της εκτέλεσης των ενεργειών αλλά και στο επίπεδο σχεδιασμού του κοινού πλάνου.
- (γ) Η δυναμικότητα του περιβάλλοντος καθιστά αναγκαίο οι πράκτορες να είναι ικανοί ν' αναθεωρούν αποτελεσματικά τις πεποιθήσεις τους, έτσι ώστε να μπορούν να σχεδιάζουν και να εκτελούν τα πλάνα τους ταυτόχρονα.
- (δ) Τέλος οι πράκτορες θα πρέπει να έχουν την ικανότητα να σχεδιάζουν και ν' αναπτύσσουν το κοινό πλάνο τους όχι σαν ένωση των επιμέρους προσωπικών πλάνων τους αλλά ως αποτέλεσμα της συνεργασίας τους. Για παράδειγμα θα πρέπει να επικοινωνούν και να συνεννοούνται μεταξύ τους για να καθορίσουν μία σειρά παραμέτρων του κοινού πλάνου όπως ποιοι πράκτορες θα πρέπει να αποτελέσουν την ομάδα συνεργασίας, ποιό αρχικό (ενδεχομένως όχι πλήρως καθορισμένο) πλάνο πρέπει να ακολουθήσουν, πώς πρέπει να καταμεριστούν οι εργασίες μεταξύ τους και πότε αυτές πρέπει να πραγματοποιηθούν [Wooldridge, 1999].

Αποτέλεσμα των παραπάνω χαρακτηριστικών είναι ότι η ικανότητα συνεργασίας ενός πράκτορα δεν μπορεί να επιτευχθεί απλά προσθέτοντας κάποιους επιμέρους μηχανισμούς στην υπάρχουσα αρχιτεκτονική του πράκτορα, αλλά θα πρέπει να σχεδιαστεί από την αρχή έτσι ώστε να διακρίνουμε σε αυτό τις παραπάνω ιδιότητες [Grosz, 1994, 1996].

## 1.7. Δομή της διατριβής

Η οργάνωση της διατριβής είναι η ακόλουθη:

- Στο εισαγωγικό πρώτο κεφάλαιο παραθέσαμε το γενικό πλαίσιο (έννοιες και ιδέες) στο οποίο η διατριβή βασίζεται.
- Στο δεύτερο κεφάλαιο δίνουμε τα κίνητρα, τους στόχους και τις προσεγγίσεις της ερευνάς μας, περιγράφουμε αναλυτικά το πρόβλημα ανάπτυξης πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα και παρουσιάζουμε δύο εφαρμογές υλοποίησης του πλαισίου ανάπτυξης πρακτόρων ICAGENT: το Tileworld και την επιφάνεια διεπαφής της εφαρμογής Δημιουργός Μοντέλων.

- Στο τρίτο κεφάλαιο γίνεται αναφορά στη σχετική βιβλιογραφία για το BDI-μοντέλο ανάπτυξης πρακτόρων, για το μοντέλο συνεργασίας *SharedPlans*, για άλλα πλαίσια ανάπτυξης πρακτόρων για δυναμικά περιβάλλοντα καθώς και για πλαίσια ανάπτυξης πρακτόρων για επιφάνειες διεπαφής.
- Στο τέταρτο κεφάλαιο περιγράφουμε αναλυτικά το πλαίσιο ανάπτυξης νοημόνων πρακτόρων ICAGENT: τη γενική αρχιτεκτονική του, τη βάση γνώσης που χρησιμοποιεί και τις διεργασίες συλλογισμού στις οποίες βασίζεται η συμπεριφορά των πρακτόρων (με παραδείγματα εφαρμογής στο περιβάλλον του Tileworld).
- Στο πέμπτο κεφάλαιο περιγράφεται η δραστηριότητα της συνεργασίας στο πλαίσιο ανάπτυξης πρακτόρων ICAGENT με παράδειγμα εφαρμογής στο Tileworld.
- Στο έκτο κεφάλαιο περιγράφουμε την ανάπτυξη πρακτόρων για επιφάνειες διεπαφής χρησιμοποιώντας το πλαίσιο ανάπτυξης πρακτόρων ICAGENT και δίνουμε ένα παράδειγμα εφαρμογής.
- Τέλος, στο έβδομο κεφάλαιο παραθέτουμε τα συμπεράσματα της ερευνάς μας και προτείνουμε θέματα για μελλοντική εργασία.

## Κεφάλαιο 2ο:

# Κίνητρα και Στόχοι της Έρευνας

Ο σκοπός της παρούσας διατριβής είναι η μελέτη και κατασκευή ενός πλαισίου εργασίας (ονομαζόμενου ICAGENT framework) ανάπτυξης νοημόνων πρακτόρων για δυναμικά και μη-αιτιοκρατικά περιβάλλοντα. Το πλαίσιο εργασίας αποτελείται από μία αρχιτεκτονική και ένα σύνολο νοητικών δομών και διεργασιών με τις οποίες ένας προγραμματιστής μπορεί ν' υλοποιήσει αυτόνομους πράκτορες περιορισμένων πόρων (resource bounded agents) ικανούς να δρουν αποτελεσματικά σε δυναμικά περιβάλλοντα είτε μόνοι τους είτε σε συνεργασία με άλλους πράκτορες συμπεριλαμβανομένου και με ανθρώπους<sup>6</sup>. Η ερευνά μας χωρίζεται σε τρία επίπεδα:

Το πρώτο αφορά τη περιγραφή των προβλημάτων και των περιορισμών που προκύπτουν καθώς οι πράκτορες δρουν σε περιβάλλοντα που η κατάστασή τους αλλάζει συνεχώς και με απρόβλεπτο τρόπο όπου οι πόροι που χρησιμοποιούν εξαντλούνται κατά τη διάρκεια του χρόνου (βλ. ενότητα 2.1). Ένα τέτοιο περιβάλλον είναι το περιβάλλον εξομοίωσης πραγματικών δυναμικών περιβαλλόντων Tileworld το οποίο χρησιμοποιούμε για να κατανοήσουμε τη δυνατότητα της αρχιτεκτονικής του ICAGENT στο να κατασκευάζουμε πολυπρακτορικά συστήματα με δυνατότητες μεταξύ τους συνεργασίας και ευελιξίας συμπεριφοράς σε απρόβλεπτες αλλαγές του περιβάλλοντος (στην υποενότητα 2.5.1 γίνεται μία εκτενής παρουσίαση του περιβάλλοντος αυτού).

Μία ιδιαίτερη κατηγορία φυσικού περιβάλλοντος την οποία επίσης μελετάμε είναι αυτή των επιφανειών διεπαφής εφαρμογών λογισμικού (application's user interfaces). Στη κατηγορία αυτή μελετάμε τα χαρακτηριστικά που πρέπει να έχει μία επιφάνεια

---

<sup>6</sup> Θεωρώντας το χρήστη ενός (πολύ-)πρακτορικού συστήματος ως πράκτορα μας ενδιαφέρει η συνεργασία τόσο μεταξύ πρακτόρων του συστήματος όσο και μεταξύ πράκτορα και χρήστη του συστήματος.

διεπαφής προκειμένου να μπορεί ένας πράκτορας ν' αναγνωρίζει αποτελεσματικά τις προθέσεις του χρήστη και να συνεργάζεται μαζί του (βλ. ενότητα 2.3 και υποενότητα 2.5.2).

Το δεύτερο επίπεδο αφορά την ικανότητα των πρακτόρων να συνεργάζονται μεταξύ τους και να προσαρμόζουν (adapt) τη συμπεριφορά τους ισορροπώντας μεταξύ γρήγορων αντιδράσεων και ενεργειών που προκύπτουν κατόπιν ατομικού ή κοινωνικού στοχασμού (individual or social deliberation). Το πλαίσιο ανάπτυξης πρακτόρων ICAGENT δίνει τη δυνατότητα υλοποίησης πρακτόρων ικανών να δρουν αποτελεσματικά σε περιβάλλοντα που η κατάστασή τους αλλάζει δυναμικά και με τρόπο που δεν είναι εφικτή η λεπτομερή σχεδίαση των πλάνων τους (long term planning) και η πραγματοποίηση αυτών<sup>7</sup> μετά την ολοκλήρωση του σχεδιασμού τους.

Κύρια χαρακτηριστικά της αρχιτεκτονικής του ICAGENT είναι: (α) η χρήση νοητικών δομών τόσο για την αναπαράσταση της εσωτερικής κατάστασης του πράκτορα και των πλάνων του, (β) η χρήση ενός συνόλου διεργασιών συλλογισμού με τις οποίες ο πράκτορας μπορεί ν' αντιλαμβάνεται το περιβάλλον του, να δημιουργεί νέους στόχους, ν' αναπτύσσει τα πλάνα του, να εντοπίζει τυχούσες προβληματικές καταστάσεις και να υλοποιεί τα πλάνα του σταδιακά και κατά τη διάρκεια του σχεδιασμού τους και (γ) η χρησιμοποίηση του μοντέλου συνεργασίας *SharedPlans* [Grosz and Kraus, 1996] βάσει του οποίου ένας πράκτορας μπορεί να συνεργαστεί με μία ομάδα πρακτόρων.

Η χρήση των νοητικών δομών για την αναπαράσταση τόσο της εσωτερικής κατάστασης όσο και των πλάνων του πράκτορα (mental state view of plans) δίνει τη δυνατότητα χειρισμού των διεργασιών συλλογισμού που διαθέτει το πλαίσιο εργασίας ICAGENT μέσω νοητικών ενεργειών. Κατ' αυτόν τον τρόπο ένας πράκτορας μπορεί να σχεδιάζει και να διαχειρίζεται τα πλάνα των νοητικών του ενεργειών με τον ίδιο τρόπο (δηλαδή με τις ίδιες διεργασίες συλλογισμού) όπως ακριβώς χειρίζεται τα πλάνα των ενεργειών τού πεδίου εφαρμογής.

---

<sup>7</sup> Αν για παράδειγμα η κατάσταση του περιβάλλοντος αλλάζει με απρόβλεπτο τρόπο τότε οι πράκτορες κατά τη διάρκεια του σχεδιασμού και εκτέλεσης των πλάνων τους μπορεί να μην γνωρίζουν όλους του δυνατούς παραμέτρους που μπορούν να επηρεάσουν τις αποφάσεις τους.

Σε αντίθεση με άλλα πλαίσια εργασιών που υλοποιούν τις συμπεριφορές του πράκτορα σε επίπεδα (layered architectures) (περιορίζοντας έτσι τους τύπους συμπεριφοράς του πράκτορα στα συγκεκριμένα αυτά επίπεδα), το πλαίσιο εργασίας ICAGENT μέσω του ενιαίου τρόπου χειρισμού των ενεργειών του πεδίου εφαρμογής και των νοητικών ενεργειών έχει ως αποτέλεσμα η συμπεριφορά του πράκτορα να είναι αποτέλεσμα (emerges) του τρόπου με τον οποίο συλλογίζεται ο πράκτορας (δηλαδή του τρόπου με τον οποίο υλοποιεί τις διεργασίες συλλογισμού μέσω νοητικών ενεργειών) και όχι του τρόπου με τον οποίο επιλέγει ένα συγκεκριμένο τύπο συμπεριφοράς από ένα σύνολο προκαθορισμένων συμπεριφορών. Έτσι ο πράκτορας έχει τη δυνατότητα να προσαρμόζει τη συμπεριφορά του με ευέλικτο τρόπο ανάλογα με τις ανάγκες (στόχους) του, τη τρέχουσα κατάσταση του φυσικού περιβάλλοντος στο οποίο δρα καθώς και τη νοητική κατάσταση των συνεργατών του (δηλαδή τις επιθυμίες τους, τις προθέσεις τους και τη γνώση τους) ισορροπώντας μεταξύ γρήγορων αντιδράσεων και ενεργειών τα πλάνα των οποίων προκύπτουν κατόπιν ατομικού ή κοινωνικού στοχασμού.

Το τρίτο επίπεδο αφορά την ικανότητα των πρακτόρων να συνεργάζονται με ανθρώπους (χρήστες επιφανειών διεπαφής) βασιζόμενοι στο μοντέλο του κοινωνικού στοχασμού (social deliberation). Το πλαίσιο ανάπτυξης πρακτόρων ICAGENT δίνει τη δυνατότητα υλοποίησης πρακτόρων επιφανειών διεπαφής (interface agents) ικανών (α) να παρατηρούν τις ενέργειες του χρήστη στην επιφάνεια διεπαφής και ν' αναγνωρίζουν τους πιθανούς στόχους του, (β) να επικοινωνούν με τον χρήστη μέσω διαλόγων κάνοντας γνωστό τόσο τους στόχους τους όσο και τους τρόπους επίτευξης αυτών και (γ) να παρουσιάζουν εξυπηρετική συμπεριφορά τόσο όταν ο χρήστης ζητά τη βοήθεια του πράκτορα όσο και όταν ο χρήστης ενεργεί λανθασμένα στην επιφάνεια διεπαφής. Η επικοινωνία μέσω διαλόγων και η δυνατότητα που παρέχει το πλαίσιο εργασίας ICAGENT στους πράκτορες να εντοπίζουν και να επιλύουν συγκρούσεις μεταξύ ενεργειών, έχουν ως αποτέλεσμα την εδραίωση σχέσεων εμπιστοσύνης μεταξύ χρήστη και πράκτορα.

Στις παρακάτω τέσσερις ενότητες περιγράφουμε το πρόβλημα ανάπτυξης πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα (ενότητα 2.1) και ειδικότερα για επιφάνειες διεπαφής (ενότητα 2.3), καθώς επίσης περιγράφουμε τη προσέγγιση που ακολουθούμε και τη συνεισφορά της ερευνάς μας στα δύο αυτά περιβάλλοντα (ενότητες 2.2 και 2.4). Τέλος παρουσιάζουμε δύο εφαρμογές υλοποίησης (ένα περιβάλλον εξομοίωσης

δυναμικών και απρόβλεπτων περιβαλλόντων και μία επιφάνεια διεπαφής) στις οποίες έχουμε χρησιμοποιήσει το πλαίσιο ανάπτυξης πρακτόρων ICAGENT (ενότητα 2.5).

## **2.1. Το πρόβλημα ανάπτυξης πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα**

Το γεγονός ότι οι πράκτορες δρουν σε δυναμικά και απρόβλεπτα περιβάλλοντα και έχουν περιορισμένους πόρους (resource bounded agents) έχει τέσσερις σημαντικές συνέπειες στη συμπεριφορά τους (ως προς τον τρόπο με τον οποίο αντιλαμβάνονται το περιβάλλον τους, σχεδιάζουν τα πλάνα τους, ενεργούν και συνεργάζονται με άλλους πράκτορες):

- (α) Οι πράκτορες δεν μπορούν να γνωρίζουν πλήρως τη κατάσταση του φυσικού τους περιβάλλοντος (agents cannot be omniscient). Η συνεχής παρακολούθηση του περιβάλλοντος και ο εντοπισμός νέων καταστάσεων που δεν ενδιαφέρουν τον πράκτορα κοστίζει σε χρόνο διότι οδηγεί σε πλεονάζοντα χρόνο επεξεργασίας πληροφοριών και γενικότερα στη σπατάλη πόρων. Για αυτό οι πράκτορες πρέπει να έχουν την ικανότητα να επικεντρώνουν τη προσοχή τους μόνο σε γεγονότα και καταστάσεις που σχετίζονται με τους στόχους τους ή που μπορούν να οδηγήσουν σε ανάληψη εκτέλεσης νέων ενεργειών.
- (β) Ο χρόνος τον οποίο οι πράκτορες διαθέτουν για να επεξεργαστούν τις πληροφορίες από το περιβάλλον τους είναι πάντα περιορισμένος από το χρόνο εξάντλησης των πόρων τους. Η αναθεώρηση της γνώσης κάθε πράκτορα καθώς και η δημιουργία νέας γνώσης πρέπει να γίνεται σε χρόνο που να επιτρέπει την εκπλήρωση των στόχων του πριν τελειώσουν οι πόροι του περιβάλλοντός του.
- (γ) Επιπροσθέτως, ο σχεδιασμός και η υλοποίηση των πλάνων του πράκτορα πρέπει να γίνονται πάντα πριν εξαντληθούν οι πόροι του φυσικού περιβάλλοντος ή οι δικόι του.
- (δ) Οι πράκτορες πολλές φορές πρέπει να συνεργάζονται μεταξύ τους. Στις περιπτώσεις που ένας πράκτορας δεν έχει τη δυνατότητα να πραγματοποιήσει μόνος του μία ενέργεια (ή να υλοποιήσει έναν στόχο) ή πιστεύει ότι είναι καλύτερα να τη

πραγματοποιήσει με τη βοήθεια κάποιου άλλου πράκτορα, τότε πρέπει να έχει τη δυνατότητα να συνεργάζεται με άλλους πράκτορες.

Για ν' αντιμετωπισθούν οι παραπάνω δυσκολίες ένας πράκτορας με περιορισμένους πόρους θα πρέπει να έχει την ικανότητα να αποφασίζει το τί θέλει να πετύχει, να σχεδιάζει πλάνα και να τα υλοποιεί λαμβάνοντας υπόψη του στόχους που έχει ήδη δεσμευθεί να πραγματοποιήσει, τους περιορισμούς που θέτουν οι στόχοι αυτοί, καθώς και τους στόχους και τους πόρους των πρακτόρων με τους οποίους συνεργάζεται. Επιπροσθέτως, ο πράκτορας θα πρέπει ν' αντιδρά γρήγορα στα καινούργια ερεθίσματα από το φυσικό του περιβάλλον προσέχοντας οι καινούργιες ενέργειες που επιλέγει να πραγματοποιήσει να μην έρχονται σε αντίφαση με τους στόχους του και τα πλάνα που ήδη έχει (εκτός αν αυτό είναι αναγκαίο). Για αυτό είτε δρώντας μόνος του ή σε συνεργασία με άλλους πράκτορες είναι αναγκαίο να έχει την ικανότητα να προσαρμόζει (adapt) τη συμπεριφορά του σε κάθε νέα πληροφορία που λαμβάνει και όποτε αυτό είναι αναγκαίο.

Στη σύγχρονη βιβλιογραφία έχουν προταθεί αρκετά πλαίσια εργασίας και αρχιτεκτονικές ανάπτυξης έξυπνων πρακτόρων και πολύ-πρακτορικών συστημάτων ικανών να δρουν σε δυναμικά περιβάλλοντα είτε μόνοι τους είτε σε συνεργασία με άλλους πράκτορες (εκτενή αναφορά στα πιο αντιπροσωπευτικά αυτών παρουσιάζεται στο Κεφάλαιο 3). Οι αρχιτεκτονικές αυτές καθιστούν τους πράκτορες ικανούς να ενεργούν με τρεις τρόπους συμπεριφοράς: αυτής της *γρήγορης/αντανακλαστικής συμπεριφοράς* (reactive behaviour) όπου οι πράκτορες αντιδρούν αντανακλαστικά (reactively) στις αλλαγές του περιβάλλοντος χωρίς να σχεδιάζουν τις ενέργειες τους, αυτής της *σχεδιασής ενεργειών μέσω ατομικού στοχασμού* (deliberative behaviour) όπου οι πράκτορες αντιδρούν εκτελώντας νοητικές διεργασίες όπως αυτές της επιλογής στόχου (option selection), του φιλτραρίσματος επιθυμιών (desires filtering) και της ανάλυσης μέσωσ και σκοπών (means-end reasoning) και τέλος αυτής της *κοινωνικής συμπεριφοράς* (social behaviour) όπου οι πράκτορες σχεδιάζουν ή/και εκτελούν τα πλάνα τους σε συνεργασία με άλλους πράκτορες.

Η δυνατότητα αλλαγής της συμπεριφοράς ενός πράκτορα μεταξύ αυτών των διακριτών τύπων συμπεριφοράς συναντάται κυρίως σε αρχιτεκτονικές επιπέδων (layered architectures) όπου το κάθε επίπεδο υλοποιεί μία από τις παραπάνω συμπεριφορές. Το μειονέκτημα των αρχιτεκτονικών αυτών είναι ότι επιτρέπουν στους πράκτορες ν' αντι-



στοιχούν σε κάθε ενέργεια ένα από τους παραπάνω τρεις τύπους συμπεριφοράς χωρίς όμως να δίνουν τη δυνατότητα αλλαγής-προσαρμογής της συμπεριφοράς αυτής ανάλογα με τη κατάσταση του περιβάλλοντος στο οποίο δρουν.

## **2.2. Προσέγγιση και συνεισφορά στην ανάπτυξη πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα**

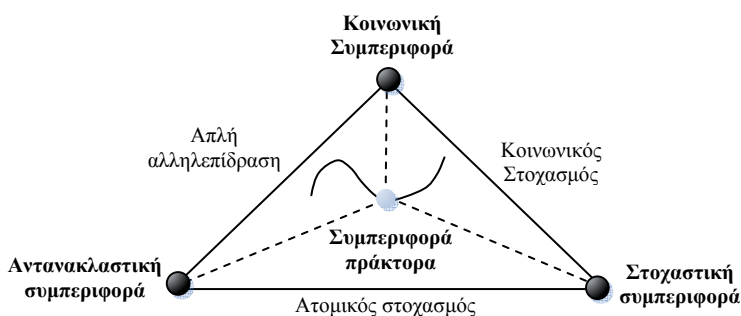
Όπως αναφέρεται στις εργασίες [Weiss, 2001; Kourakos and Vouros, 2001; Kourakos and Vouros, 2006] για να μπορεί ένας πράκτορας να δρα αποτελεσματικά σε ένα περιβάλλον που η κατάσταση του αλλάζει συνεχώς και με απρόβλεπτο τρόπο θα πρέπει η συμπεριφορά του να μην είναι αποτέλεσμα μιας απλής αντιστοίχισης ενεργειών σε συγκεκριμένους τύπους συμπεριφοράς, χωρίς καμία μετέπειτα δυνατότητα προσαρμογής αυτής στις αλλαγές του περιβάλλοντος. Η συμπεριφορά του πράκτορα θα πρέπει να «ακολουθεί» τις αλλαγές του περιβάλλοντος προσαρμόζοντας τον τρόπο με τον οποίο παρατηρεί το περιβάλλον του, στοχάζεται και υλοποιεί τα πλάνα του· τόσο στις ανάγκες του όσο και στους περιορισμούς που θέτει το φυσικό του περιβάλλον.

Ειδικότερα, αναγκαία προϋπόθεση για να δρα αποτελεσματικά ένας πράκτορας σε ένα δυναμικό περιβάλλον είναι να έχει προηγμένες ικανότητες διαχείρισης των πλάνων του [Pollack and Horthy 1999]. Στο πλαίσιο εργασίας ICAGENT ο πράκτορας διαχειρίζεται τα πλάνα κάνοντας χρήση των ακόλουθων διεργασιών συλλογισμού (reasoning tasks):

- (α) Παρακολούθηση περιβάλλοντος (environment monitoring). Ο πράκτορας επιλέγει ποιος τύπος γεγονότων και ενεργειών πρέπει να παρατηρεί στο φυσικό του περιβάλλον.
- (β) Αναγνώριση καταστάσεων (situation recognition). Ο πράκτορας αποφασίζει σε ποιες καταστάσεις θα πρέπει να λάβει τη πρωτοβουλία να δράσει.
- (γ) Αναγνώριση και επίλυση συγκρούσεων (reconciliation). Ο πράκτορας αποφασίζει για το αν θα προσπαθήσει ν' αναγνωρίσει και να επιλύσει τυχόν συγκρούσεις μεταξύ ενεργειών ή μεταξύ των προϋποθέσεων εκτέλεσης των ενεργειών (preconditions) και της τρέχουσας νοητικής του κατάστασης.

- (δ) Ανάπτυξη πλάνων (plan elaboration). Με αυτή τη διεργασία συλλογισμού ο πράκτορας αναπτύσσει τα πλάνα του είτε μόνος του είτε σε συνεργασία με άλλους πράκτορες.
- (ε) Πραγματοποίηση προθέσεων (intention realization). Με αυτή τη διεργασία συλλογισμού ο πράκτορας αποφασίζει το πότε θα πραγματοποιήσει τα πλάνα του.

Για την υλοποίηση των παραπάνω διεργασιών συλλογισμού, κάνουμε χρήση νοητικών δομών όπως αυτή της πεποίθησης, της επιθυμίας και της πρόθεσης (BDI mental structures) καθώς και μίας ομάδας νοητικών ενεργειών οι οποίες ορίζουν τη νοητική-όψη της χειρισμού των πλάνων ενός πράκτορα (the mental-state view of plan management) (οι νοητικές δομές και ενέργειες περιγράφονται αναλυτικά στο Κεφάλαιο 4).



*Εικόνα 3:* Η συμπεριφορά του πράκτορα προκύπτει από τον τρόπο με τον οποίο αυτός υλοποιεί τις διεργασίες συλλογισμού μέσω νοητικών ενεργειών. Η κάθε μία νοητική ενέργεια μπορεί να πραγματοποιηθεί με αντανακλαστικό τρόπο, με στοχαστικό τρόπο ή σε συνεργασία με άλλους πράκτορες. Η συνολική συμπεριφορά δεν αναπαριστάται από ένα σύνολο διακριτών συμπεριφορών αλλά από μία συνεχή γραμμή περιγεγραμμένη από τις τρεις βασικές συμπεριφορές την αντανακλαστική (reactive behaviour), τη στοχαστική (deliberative behaviour) και τη κοινωνική (social behaviour) έτσι όπως αυτές προκύπτουν από τον τρόπο με τον οποίο πραγματοποιούνται οι διεργασίες συλλογισμού [Kourakos-Mavromichalis and Vouros, 2006; Kourakos-Mavromichalis and Vouros, 2001].

Η συνεισφορά του πλαισίου εργασίας ICAGENT είναι ότι εισάγει μία εναλλακτική ως προς τις αρχιτεκτονικές επιπέδων άποψη για τον τρόπο με τον οποίο ένας πράκτορας προσαρμόζει τη συμπεριφορά του στις αλλαγές του περιβάλλοντός του. Σε αντίθεση με τις αρχιτεκτονικές επιπέδων που υλοποιούν κάθε τύπο συμπεριφοράς (όπως αυτό της αντανακλαστικής αντίδρασης και της αντίδρασης που προκύπτει κατόπιν ατομικού ή κοινωνικού στοχασμού) σε μία ξεχωριστή μονάδα (module) της αρχιτεκτονικής, το πλαίσιο εργασίας ICAGENT εισάγει τη «ρυθμιστική» (tuning) προσέγγιση (βλ. Εικόνα 3): οι εργασίες συλλογισμού (reasoning tasks) με τις οποίες ο πράκτορας αναπτύσσει τα

πλάνα του υλοποιούνται μέσω νοητικών ενεργειών (mental actions) τις οποίες ο πράκτορας χειρίζεται όπως τις ενέργειες της εφαρμογής (domain actions) κάτω από το ίδιο μηχανισμό διαχείρισης πλάνων, βασισμένο στις ίδιες νοητικές δομές. Όπως τονίσαμε παραπάνω, ο κοινός χειρισμός των ενεργειών αυτών έχει ως αποτέλεσμα ο πράκτορας να μπορεί να προσαρμόζει τη συμπεριφορά του ανάλογα με τις ανάγκες του, τη τρέχουσα κατάσταση του φυσικού περιβάλλοντος στο οποίο δρα καθώς και τη νοητική κατάσταση των συνεργατών του.

Μεταξύ όλων των τύπων συμπεριφοράς, στο πλαίσιο εργασίας ICAGENT ξεχωρίζουμε τρεις κύριους τύπους: την αμιγώς *αντανακλαστική*, τη αμιγώς *στοχαστική* και αυτήν της *κοινωνικής* συμπεριφοράς.

Λέμε ότι ένας πράκτορας ενεργεί αμιγώς αντανακλαστικά όταν αναπτύσσει/προσαρμόζει/αναθεωρεί τα πλάνα του μόνο ως αντίδραση στις αλλαγές του φυσικού του περιβάλλοντος χωρίς να λαμβάνει υπόψη του δικούς του στόχους ή αυτούς των άλλων πρακτόρων. Ακολουθώντας τη συμπεριφορά αυτή ο πράκτορας δεν συνεργάζεται με άλλους πράκτορες και η μόνη πληροφορία που ενδεχομένως να μοιράζεται με άλλους αφορά τη κατάσταση του φυσικού του περιβάλλοντος.

Αντίθετα, ένας πράκτορας δρα αμιγώς στοχαστικά όταν αναπτύσσει/προσαρμόζει/αναθεωρεί τα πλάνα του ως αποτέλεσμα συλλογισμού και όχι κάποιας απλής αντανακλαστικής αντίδρασης. Όπως και στην αμιγώς αντανακλαστική συμπεριφορά έτσι και στη στοχαστική ο πράκτορας δεν συνεργάζεται με άλλους και η μόνη πληροφορία η οποία μπορεί να μοιράζεται με άλλους αφορά το φυσικό του περιβάλλον.

Τέλος λέμε ότι ένας πράκτορας συμπεριφέρεται κοινωνικά όταν υλοποιεί τις ενέργειές του συνεργατικά με άλλους πράκτορες με στοχαστικό ή αντανακλαστικό τρόπο. Η πληροφορία που μπορεί να μοιράζονται οι πράκτορες κατά τη διάρκεια της συνεργασίας αφορά στόχους, μεθόδους υλοποίησης ενεργειών, νοητικές καταστάσεις καθώς επίσης και πληροφορία για την κατάσταση του φυσικού περιβάλλοντος.

Στο πλαίσιο εργασίας ICAGENT, η συνολική συμπεριφορά όμως ενός πράκτορα δεν καθορίζεται αποκλειστικά απ' αυτούς τους τρεις τύπους συμπεριφοράς, αλλά από τους τρόπους με τους οποίους οι παραπάνω διεργασίες συλλογισμού υλοποιούνται. Για παράδειγμα, αν ένας πράκτορας (α) δεν παρατηρεί το περιβάλλον του για εμφάνιση

νέων συμβάντων παρά μόνο δέχεται πληροφορίες από άλλους πράκτορες, (β) δεν ελέγχει για τυχόν συγκρούσεις μεταξύ των ενεργειών που δεσμεύεται να πραγματοποιήσει και (γ) δεν λαμβάνει υπόψη τους στόχους και τους περιορισμούς των άλλων πρακτόρων τότε η συμπεριφορά του χαρακτηρίζεται λιγότερο στοχαστική (ή περισσότερο αντανακλαστική) από εκείνη ενός πράκτορα που λαμβάνει υπ' όψη όλα τα παραπάνω.

Η αρχιτεκτονική η οποία χρησιμοποιείται για την ανάπτυξη του ICAGENT βασίζεται στο μοντέλο προθετικών πρακτόρων (BDI Agents) και στο μοντέλο συνεργατικών πρακτόρων *SharedPlans* της Barbara Grosz και Sarit Kraus. Το μοντέλο των *SharedPlans* είναι ένα γενικευμένο μοντέλο το οποίο περιγράφει τις νοητικές δομές και προϋποθέσεις που πρέπει να έχει μία ομάδα πρακτόρων για να συνεργάζονται μεταξύ τους. Σε δυναμικά όμως περιβάλλοντα το μοντέλο αυτό χρειάζεται περαιτέρω υποστήριξη σε θέματα που αφορούν τον «στοχασμό» και την «ανάλυση μέσων και σκοπών». Για αυτό το λόγο, κάνουμε χρήση του μοντέλου BDI έτσι ώστε να υποστηριχθεί η πρακτική συλλογιστική που απαιτείται από τους πράκτορες. Όπως ήδη έχουμε αναφέρει με το μοντέλο αυτό ο πράκτορας μπορεί να «στοχάζεται» (δηλαδή να αποφασίζει ποιες καταστάσεις πραγμάτων πρέπει να πετύχει) και να «σχεδιάζει» τα πλάνα του (δηλαδή να αποφασίζει πως θα επιτύχει αυτές τις καταστάσεις) όταν το περιβάλλον του αλλάζει με δυναμικό και μη-αιτιοκρατικό τρόπο. Εκτενή αναφορά στο μοντέλο του BDI και *SharedPlans* παρουσιάζουμε στο Κεφάλαιο 3.

### **2.3. Το πρόβλημα ανάπτυξης πρακτόρων για περιβάλλοντα επιφανειών διεπαφής**

Ως περιβάλλον διεπαφής ονομάζεται το λογισμικό εκείνο με το οποίο ο χρήστης ελέγχει ή/και επικοινωνεί με μία εφαρμογή. Η επικοινωνία συνήθως γίνεται είτε μέσω μίας γραμμής εντολών (όπως για παράδειγμα στον επεξεργαστή κειμένου vi του Unix) είτε μέσω ενός γραφικού περιβάλλοντος αποτελούμενου από εικονίδια, menus, αναδιπλούμενες λίστες κτλ.

Το πιο επιτυχημένο μοντέλο γραφικού περιβάλλοντος διεπαφής είναι αυτό του «άμεσου χειρισμού» (direct manipulated interfaces) που έχει προταθεί από τον Ben Shneiderman [Shneiderman, 1983; Shneiderman, 1997]. Πιο συγκεκριμένα, για να χαρακτηριστεί ένα γραφικό περιβάλλον διεπαφής ως «άμεσου χειρισμού» θα πρέπει να

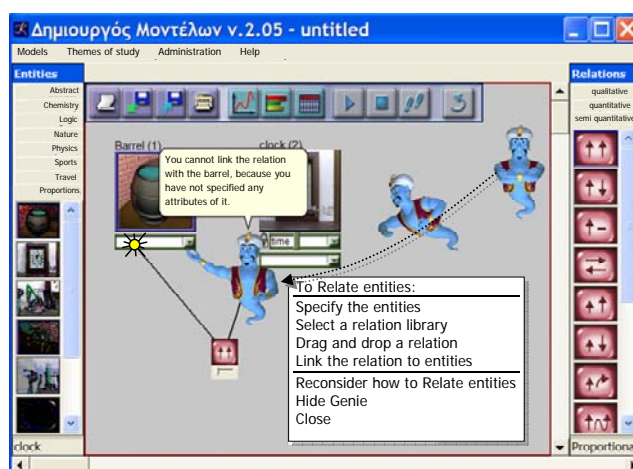
έχει τις ακόλουθες ιδιότητες [Shneiderman, 1997; Hutchins et. al, 1986]: (α) οι ενέργειες του χρήστη θα πρέπει να πραγματοποιούνται με φυσικό τρόπο (αντί της χρήσης κάποιου σύνθετου συντακτικού) ή ν' αναπαριστώνται με αντικείμενα (π.χ. με πλήκτρα) τα οποία να χαρακτηρίζουν μοναδικά τις ενέργειες αυτές, (β) τα αποτελέσματα τόσο των ενεργειών του χρήστη όσο και του συστήματος θα πρέπει να είναι οπτικώς εμφανή, (γ) η εφαρμογή θα πρέπει ν' αντιδρά άμεσα και συνεχώς σε κάθε ενέργεια του χρήστη, (δ) οι ενέργειες του χρήστη θα πρέπει να είναι αντιστρέψιμες έτσι ώστε σε περίπτωση λάθους να μπορεί να τις αναιρεί και να επανασχεδιάζει τους στόχους του, (ε) ο χρήστης θα πρέπει να είναι ο μόνος υπεύθυνος για τις ενέργειες που πραγματοποιεί, δηλαδή δεν θα πρέπει να παρεμβάλλεται κάποιο άλλο λογισμικό μεταξύ του χρήστη και της επιφάνειας διεπαφής.

Η επιτυχία των επιφανειών διεπαφής «άμεσου χειρισμού» έγκειται στο γεγονός ότι οι ενέργειες (domain tasks) που επιθυμεί να πραγματοποιήσει ο χρήστης αναλύονται σε άλλες απλούστερες ενέργειες της επιφάνειας διεπαφής οι οποίες είτε αναπαριστώνται με αντικείμενα είτε πραγματοποιούνται με φυσικό τρόπο (όπως το σύρσιμο και το άφημα ενός αντικειμένου). Η εγγύτητα μεταξύ των ενεργειών του χρήστη στην επιφάνεια διεπαφής και των πραγματικών στόχων του χρήστη (δηλαδή το γεγονός ότι το βάθος της ανάλυσης ενός στόχου του χρήστη μέχρι τις ενέργειες που πραγματοποιεί στην επιφάνεια διεπαφής είναι μικρός) έχει ως αποτέλεσμα την απλοποίηση των εργασιών του και τον εύκολο χειρισμό της εφαρμογής [Shneiderman, 1997].

Παρ' όλα αυτά, είναι γενικά αποδεκτό ότι για να μάθει ένας άπειρος χρήστης τις λειτουργίες ενός σύνθετου περιβάλλοντος διεπαφής «άμεσου χειρισμού» θα πρέπει είτε να εκπαιδευτεί από έναν έμπειρο χρήστη είτε να διαβάσει το εγχειρίδιο χρήσης της εφαρμογής [Shneiderman, 1997]. Επειδή όμως τα εγχειρίδια χρήσης, σε οποιαδήποτε μορφή και αν βρίσκονται (έντυπα, πολυμέσων κτλ.), έχουν σχεδιαστεί με βάση τα χαρακτηριστικά ενός τυπικού χρήστη δεν έχουν την ικανότητα να προσαρμόζονται στις ιδιαιτερότητες και τις γνώσεις του κάθε χρήστη ξεχωριστά. Για αυτόν το λόγο οι χρήστες σπάνια διαβάζουν τα εγχειρίδια χρήσης και πολύ συχνά χρειάζονται τη βοήθεια ενός πιο έμπειρου χρήστη.

Όπως τονίζεται στις εργασίες των [Shneiderman and Maes, 1997; Delisle, 2002] «η αλληλεπίδραση του χρήστη με τον υπολογιστή θα πρέπει να ιδωθεί ως μία συνεργασία

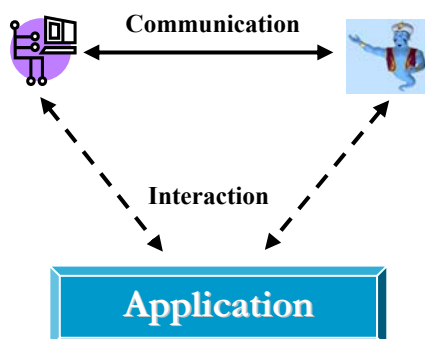
κατά τη διάρκεια της οποίας ο υπολογιστής υποστηρίζει τις ενέργειες του χρήστη με έξυπνο τρόπο, ιδιαίτερα μάλιστα όταν συναντά δυσκολίες”. Με στόχο την υλοποίηση της ιδέας αυτής, προτείνουμε ένα νοήμονα συνεργατικό πράκτορα διεπαφής, ο οποίος δρώντας παράλληλα με τις επιφάνειες διεπαφής «άμεσου χειρισμού» αναπτύσσει πλάνα (διαλόγους) με τον χρήστη με στόχο την από κοινού πραγματοποίηση ενεργειών (βλ. Εικόνα 4).



Εικόνα 4: Παράδειγμα πράκτορα επιφάνειας διεπαφής βασισμένο στο πλαίσιο εργασίας ICAGENT.

Πιο συγκεκριμένα, χρησιμοποιούμε το παράδειγμα συνεργατικού πράκτορα επιφάνειας διεπαφής της Pattie Maes [Maes, 1994] (βλ. Εικόνα 5). Σ’ αυτό ο πράκτορας δρα ανεξάρτητα και παράλληλα της επιφάνειας διεπαφής, επικοινωνεί με τον χρήστη, παρατηρεί τις ενέργειές του και αναλαμβάνει τη πρωτοβουλία να τον βοηθήσει όποτε αυτό είναι αναγκαίο.

Ένα από τα πιο σημαντικά θέματα των πρακτόρων διεπαφής, το οποίο μάλιστα έχει δεχθεί και τις περισσότερες κριτικές, είναι αυτό της *εμπιστοσύνης* (trust) [Castelfranchi and Falcone, 1998; Horvitz, 1999]. Όταν αναθέτει ο χρήστης σε ένα πράκτορα τη πραγματοποίηση μίας εργασίας πρέπει να πιστεύει ότι ο πράκτορας έχει την ικανότητα να τη πραγματοποιήσει, έχει κατανοήσει τί πραγματικά θέλει να πετύχει ο χρήστης και ότι θα τον βοηθήσει χωρίς οι ενέργειές του να έρχονται σε σύγκρουση με τους στόχους του χρήστη.



Εικόνα 5: Για να συμπεριφέρεται αποτελεσματικά ένας συνεργατικός πράκτορας θα πρέπει να δρα παράλληλα με τον χρήστη να παρατηρεί τις ενέργειες του και να τον βοηθά μόνο όταν χρειάζεται.

Ιδιαίτερο ρόλο στην ανάπτυξη της εμπιστοσύνης μεταξύ του χρήστη και του πράκτορα παίζει η επικοινωνία μεταξύ αυτών. Ο πράκτορας θα πρέπει να επικοινωνεί τις επιθυμίες του, τους στόχους του, να κάνει υποθέσεις για το τι ο χρήστης προσπαθεί να πραγματοποιήσει και να επικοινωνεί τη πρόθεση του για βοήθεια όποτε πιστεύει ότι ο χρήστης χρειάζεται. Από την άλλη πλευρά ο χρήστης θα πρέπει να νιώθει ότι έχει τον έλεγχο της εφαρμογής και της συνεργασίας δίνοντας του τη δυνατότητα να καθορίζει μόνος του τους στόχους του και να τους επικοινωνεί στο πράκτορα όποτε χρειάζεται βοήθεια. Η συνεχής επικοινωνία είναι σημαντική για τα συνεργαζόμενα μέλη τόσο για τη δημιουργία κοινών στόχων όσο και για τη δημιουργία κοινών πλάνων ώστε ο χρήστης να εμπιστεύεται στον πράκτορα την ανάθεση εργασιών και ο πράκτορας να είναι ικανός να εντοπίζει αποτελεσματικά τις ανάγκες του χρήστη και να τον βοηθά με αποτελεσματικό τρόπο.

Η προϋπόθεση ύπαρξης επιφάνειας διεπαφής άμεσου χειρισμού παρέχει στο πράκτορα τις κατάλληλες συνθήκες για μία αποτελεσματική συνεργασία με τον χρήστη. Η κύρια συνθήκη είναι η εγγύτητα (που περιγράψαμε παραπάνω) μεταξύ των ενεργειών που μπορεί να πραγματοποιήσει ο χρήστης στην επιφάνεια διεπαφής και τους πραγματικούς του στόχους. Λόγω του ότι κάθε ενέργεια του χρήστη συνδέεται άμεσα με ένα αντικείμενο της επιφάνειας διεπαφής, ο πράκτορας μπορεί να αναγνωρίζει ευκολότερα τους κύριους στόχους του χρήστη (η αναγνώριση αυτή περιγράφεται στην ενότητα 6.8). Επίσης, το γεγονός ότι τα αποτελέσματα κάθε ενέργειας του πράκτορα γίνονται άμεσα ορατά από το χρήστη έχει ως αποτέλεσμα την ευκολότερη κατανόηση της συμπεριφοράς του πράκτορα από τον χρήστη και την αύξηση της εμπιστοσύνης του προς αυτόν.

## 2.4. Προσέγγιση και συνεισφορά στην ανάπτυξη πρακτόρων επιφανειών διεπαφής

Δεδομένων των πλεονεκτημάτων και των προϋποθέσεων των επιφανειών διεπαφής «άμεσου χειρισμού» που περιγράψαμε στη προηγούμενη ενότητα και των επικοινωνιακών απαιτήσεων μεταξύ πράκτορα και χρήστη έτσι ώστε να υφίσταται μία σχέση εμπιστοσύνης μεταξύ τους, οδηγούμαστε στη κατασκευή έξυπνων συνεργατικών πρακτόρων διεπαφής με τα ακόλουθα χαρακτηριστικά:

*Αναγνώριση των επιθυμιών του χρήστη:* Ο πράκτορας διεπαφής θα πρέπει να παρακολουθεί το χρήστη διακριτικά (δηλαδή χωρίς να του αποσπά τη προσοχή του με περιττές ενέργειες) και ν' αναγνωρίζει τί προσπαθεί να πετύχει, αποδίδοντας σε αυτόν επιθυμίες. Ο πράκτορας θα πρέπει να έχει την ικανότητα να αναγνωρίζει και να «ξετυλίγει» το συνολικό πλαίσιο της δραστηριότητας του χρήστη, τις προτιμήσεις του και τα κίνητρα του. Αυτό είναι σημαντικό για την επακόλουθη ερμηνεία των ενεργειών του χρήστη και κατά επέκταση για τη συνεργασία.

*Ικανότητα σχεδιασμού και εκτέλεσης πλάνων σε συνεργασία με τον χρήστη:* Για κάθε κοινό στόχο που ο πράκτορας και ο χρήστης συμφωνούν να πετύχουν μαζί, ο πράκτορας θα πρέπει να έχει την ικανότητα να σχεδιάζει και να πραγματοποιεί πλάνα από κοινού με τον χρήστη προς επίτευξη των στόχων αυτών.

*Εντοπισμός και επίλυση συγκρούσεων μεταξύ των ενεργειών του χρήστη:* Ο πράκτορας θα πρέπει να μπορεί να εντοπίζει τυχόν συγκρούσεις που συμβαίνουν μεταξύ των ενεργειών του χρήστη και του κοινού πλάνου που αναπτύσσουν ή του εκτιμώμενου πλαισίου ενεργειών του χρήστη (στη περίπτωση που ο χρήστης δεν έχει συμφωνήσει ακόμα να συνεργαστεί με τον πράκτορα). Για την επίλυση αυτών των συγκρούσεων, ο πράκτορας παρέχει χρήσιμες πληροφορίες και ενεργεί με τρόπο που να επιλύει τις συγκρούσεις αυτές μόνο στη περίπτωση που ο χρήστης έχει συμφωνήσει να συνεργαστεί με τον πράκτορα προς την επίτευξη ενός κοινού στόχου.

*Εκμετάλλευση διδακτικών στρατηγικών:* Ένας πράκτορας πρέπει να προσφέρει συμβουλές και να ενεργεί με τέτοιο τρόπο έτσι ώστε να καταστεί τον χρήστη ικανό να επιτυγχάνει τους τρέχοντες και μελλοντικούς του στόχους επιτυχώς.



*Προσαρμογή της συμπεριφοράς του πράκτορα ανάλογα με την εμπειρία του χρήστη:* Ο πράκτορας πρέπει να προσαρμόζει τη συμπεριφορά του (σε ότι αφορά τον τρόπο με τον οποίο ενεργεί στην επιφάνεια διεπαφής και παρουσιάζει πληροφορία στο χρήστη) βασισμένος σε γνώση και πληροφορία για την εμπειρία, τις προτιμήσεις και τις ανάγκες του χρήστη.

Τα παραπάνω σε συνδυασμό με την απαίτηση ότι ο χρήστης πρέπει να έχει πάντα τον έλεγχο της εφαρμογής είναι βασικά για να παρουσιάζει ο πράκτορας διεπαφής εξυπηρετική συμπεριφορά προς τον χρήστη και να γίνεται αποδεκτή. Εντούτοις, πρέπει να σημειωθεί ότι στη παρούσα διατριβή δεν έχει δοθεί έμφαση στις εκπαιδευτικές στρατηγικές που μπορούν να χρησιμοποιηθούν και στη προσαρμογή της συμπεριφοράς των πρακτόρων στο προφίλ του χρήστη, παρά μόνο στη προσαρμογή της συμπεριφοράς του πράκτορα στο κοινό πλαίσιο ενεργειών (διάλογο) που αναπτύσσεται μεταξύ του πράκτορα και του χρήστη.

Ως εξυπηρετική συμπεριφορά ορίζουμε την ικανότητα του πράκτορα να βοηθά τον χρήστη όταν χρειάζεται, δηλαδή όταν (α) ο χρήστης ζητά μία πληροφορία από τον πράκτορα ή του αναθέτει την πραγματοποίηση μίας ενέργειας, (β) όταν ο χρήστης δεν μπορεί να πραγματοποιήσει μία ενέργεια και (γ) όταν χρήστης ενεργεί λανθασμένα στην επιφάνεια διεπαφής. Στη πρώτη περίπτωση, ο πράκτορας χρησιμοποιώντας γνώση την οποία έχει για την επιφάνεια διεπαφής πραγματοποιεί ενέργειες (domain actions) για λογαριασμό του χρήστη ή παρέχει σ' αυτόν πληροφορίες. Ενώ στη δεύτερη και τρίτη περίπτωση ο πράκτορας και ο χρήστης συνεργάζονται προκειμένου να επιλύσουν από κοινού το πρόβλημα (της αδυναμίας πραγματοποίησης μίας ενέργειας ή της πραγματοποίησης μίας λανθασμένης ενέργειας).

Όπως ήδη έχουμε αναφέρει ένα από τα πιο σημαντικά θέματα των πρακτόρων διεπαφής είναι η εμπιστοσύνη (trust) του χρήστη προς τον πράκτορα. Η έλλειψη εμπιστοσύνης καταλήγει στην απροθυμία των χρηστών να αναθέσουν εργασίες στους πράκτορες. Για ν' αναθέσει ένας χρήστης μία εργασία σε ένα πράκτορα διεπαφής, θα πρέπει πρώτον να νιώθει ότι έχει τον έλεγχο, δεύτερον να γνωρίζει ότι ο πράκτορας μπορεί να πραγματοποιήσει την εργασία αυτή, τρίτον ότι ο πράκτορας θα δεσμευτεί να τη πραγματοποιήσει, και τέταρτον ότι η εργασία θα πραγματοποιηθεί υπό τους περιορισμούς που θέτει το τρέχον πλαίσιο ενεργειών του χρήστη, δηλαδή ότι η εργασία θα πραγματο-

ποιηθεί σύμφωνα με τις επιθυμίες του χρήστη και δεν θα πραγματοποιηθούν ενέργειες που θα εμποδίσουν το χρήστη να επιτύχει τους στόχους του.

Προϋπόθεση για την εδραίωση σχέσης εμπιστοσύνης μεταξύ του χρήστη και του πράκτορα διεπαφής είναι η ικανότητα του πράκτορα ν' αναπτύσσει διαλόγους με το χρήστη. Ο διάλογος δίνει τη δυνατότητα στο πράκτορα να αναγνωρίζει τις επιθυμίες, τα κίνητρα, τις ανάγκες και το επίπεδο γνώσεων του χρήστη [Ortiz and Grosz, 2002; Ragnemalm, 1996], να μοιράζεται τους στόχους και τα πλάνα του με τον χρήστη έτσι ώστε να μπορεί να ενεργεί επιτυχώς είτε για λογαριασμό του χρήστη είτε από κοινού όποτε αυτό είναι αναγκαίο. Με ανάλογο τρόπο, ο διάλογος μπορεί να αξιοποιηθεί από τον χρήστη για να επιθεωρήσει και να καθοδηγήσει περαιτέρω τις υποθέσεις του πράκτορα καθώς και για να τον ενημερώσει για τις προθέσεις του και τα κίνητρα του.

Όπως αναφέρεται στην εργασία [Grosz, 1994], «η επικοινωνία χρειάζεται όχι μόνο όταν πρέπει να ληφθούν αποφάσεις για την επίλυση προβλημάτων που αφορούν την σωστή διαχείριση των πόρων (που απαιτούνται για την επίτευξη ενός στόχου) αλλά και όταν κάτι πάει λάθος στην εκτέλεση μίας από κοινού δραστηριότητας». Γι' αυτό κάθε φορά που ο πράκτορας εντοπίζει ότι κάτι πάει λάθος στην επίτευξη ενός κοινού στόχου, τότε αυτός θα πρέπει να επικοινωνήσει το πρόβλημα στο χρήστη και να τον παροτρύνει να το επιλύσει είτε μόνος ή σε συνεργασία με αυτόν εκδηλώνοντας εξυπηρετική συμπεριφορά.

Συνοψίζοντας τα παραπάνω, η κύρια συνεισφορά του πλαισίου εργασίας ICAGENT σε ότι αφορά την ανάπτυξη πρακτόρων διεπαφής, έγκειται στην δυνατότητα που παρέχει εδραίωσης σχέσεων εμπιστοσύνης μεταξύ πράκτορα και χρήστη μέσω διαλόγων (κοινών πλάνων) που αναπτύσσονται κατά τη διάρκεια της συνεργασίας τους. Κατά τη διάρκεια των διαλόγων, ο πράκτορας διεπαφής επικοινωνεί με τον χρήστη όταν:

1. Χρειάζεται να συγκεκριμενοποιήσει τις υποθέσεις του όσο αφορά τις επιθυμίες του χρήστη, παρέχοντας σε αυτόν επιλογές που τον βοηθούν να διευκρινίσει τον επιθυμητό στόχο και να αρχίσει τη συνεργασία μαζί του.
2. Αναγνωρίζει ότι ο χρήστης ενεργεί λανθασμένα τόσο κατά τη διάρκεια της συνεργασίας όσο και πριν από αυτή. Στη πρώτη περίπτωση ο πράκτορας πληροφορεί τον χρήστη για τυχόν προβλήματα που εντοπίζει στις ενέργειες του χρήστη, ενώ στη

δεύτερη απλώς τον παρακινεί να συνεργαστεί μαζί του για να λύσουν το πρόβλημα από κοινού χωρίς όμως να τον διακόπτει από τις τρέχουσες ενέργειές του.

3. Αναπτύσσει από κοινού με τον χρήστη ένα πλάνο για να πετύχουν ένα κοινό στόχο. Κατά τη διάρκεια της ανάπτυξης του κοινού πλάνου ο πράκτορας επικοινωνεί (μοιράζεται) τόσο τις μεθόδους επίτευξης των στόχων αυτών όσο και τους λόγους πραγματοποίησης αυτών.

Αντίθετα, ο χρήστης κατά τη διάρκεια των διαλόγων επικοινωνεί με τον πράκτορα διεπαφής για: να αποδεχθεί ή ν' απορρίψει μία μέθοδο επίτευξης ενός στόχου, να ενημερώσει τον πράκτορα για τις τρέχουσες επιθυμίες του και προθέσεις του, καθώς και για ν' αναθέσει στο πράκτορα τη πραγματοποίηση μίας ενέργειας.

Όπως αναφέραμε στην ενότητα 2.2, για την παροχή ικανότητας συνεργασίας μεταξύ πρακτόρων σε δυναμικά περιβάλλοντα (όπως είναι και οι επιφάνειες διεπαφής εφαρμογών λογισμικού) η υλοποίησή του πλαισίου εργασίας ICAGENT βασίζεται στο μοντέλο συνεργασίας *SharedPlans* καθώς και στο BDI μοντέλο ανάπτυξης πρακτόρων.

Το μοντέλο *SharedPlans*, σε ότι αφορά την ανάπτυξη πρακτόρων διεπαφής, προσδιορίζει τις νοητικές δομές και προϋποθέσεις που απαιτούνται για να δρα ένας πράκτορας σε συνεργασία με τον χρήστη. Ειδικότερα, περιγράφει τη διαδικασία ανάπτυξης του κοινού πλάνου (διαλόγου) μεταξύ του πράκτορα και του χρήστη: Ο πράκτορας σχηματίζει *προθέσεις να (intentions to)* ενεργήσει και *προθέσεις ότι (intentions that)* ο χρήστης θα πετύχει τον κοινό στόχο. Οι τελευταίου τύπου προθέσεις δεν επιτρέπουν στο πράκτορα διεπαφής να δεσμευτεί για ενέργειες που έρχονται σε σύγκρουση τους στόχους του χρήστη, ενώ τον οδηγεί να εκδηλώσει εξυπηρετική συμπεριφορά όταν ο χρήστης τη χρειάζεται. Με αυτό τον τρόπο, ο πράκτορας και ο χρήστης συνεργάζονται προς τη κατασκευή ενός ολοκληρωμένου κοινού πλάνου (διαλόγου) αρχίζοντας συνήθως από ένα μερικά ανεπτυγμένο κοινό πλάνο (μέθοδο) το οποίο ο πράκτορας ή ο χρήστης προτείνει.

Επίσης, πρέπει να τονίσουμε πως ο συνεργατικός σχεδιασμός των πλάνων και η από κοινού δράση του πράκτορα παρέχονται στο πλαίσιο του ICAGENT ως ενσωματωμένες λειτουργίες (build-in functionalities) και όχι ως επιπρόσθετες μονάδες (add-on modules). Μία τέτοια εγγενής προσέγγιση επιτρέπει στον πράκτορα να αντιλαμβάνεται

τη νοητική κατάσταση του χρήστη στο σύνολό της (επιθυμίες, προθέσεις, πεποιθήσεις), συμβιβάζοντας τις δικές του επιθυμίες, προθέσεις και πεποιθήσεις με αυτές του χρήστη κάτω από ένα κοινό πλαίσιο δράσης (context of action), εντοπίζοντας έτσι τις περιπτώσεις όπου ο χρήστης χρειάζεται βοήθεια. Όπως θα παρουσιάσουμε στις ακόλουθες ενότητες, αυτή η συμπεριφορά επιτυγχάνεται μέσω του *μηχανισμό ελέγχου της BDI-Πρακτικής Συλλογιστικής* (BDI-Control of Practical Reasoning) του ICAGENT.

Προσεγγίσεις ολοκληρωμένης υλοποίησης συνεργατικών πρακτόρων που χρησιμοποιούν την θεωρία *SharedPlans* συμπεριλαμβάνουν το Collagen [Rich, Sidner and Lesh, 2001; Rickel et. al., 2001] και το DIAL [Ortiz and Grosz, 2000]. Μία εκτενής παρουσίαση των συστημάτων αυτών γίνεται στις υποενότητες 3.4.1 και 3.4.2, ενώ μία συζήτηση των διαφορών μεταξύ αυτών των προσεγγίσεων και του ICAGENT δίνεται στην υποενότητα 3.4.3.

## 2.5. Εφαρμογές υλοποίησης

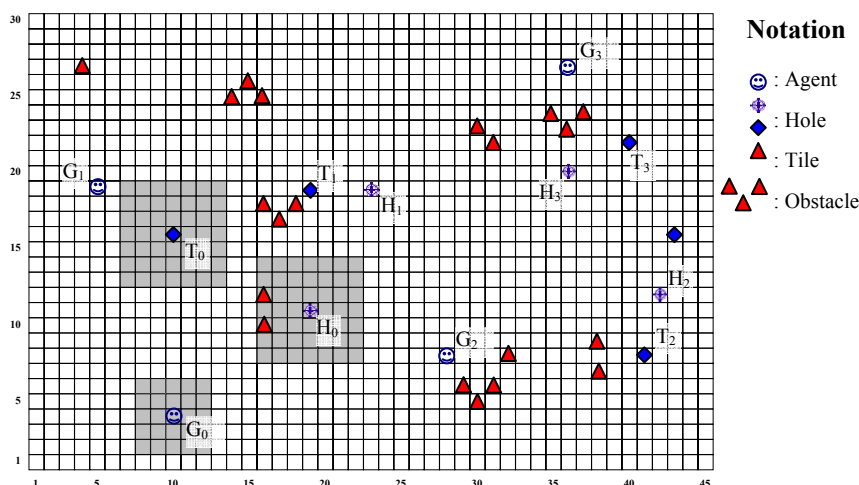
Το πλαίσιο εργασίας ICAGENT έχει χρησιμοποιηθεί για την ανάπτυξη συνεργατικών πολύ-πρακτορικών συστημάτων για το ανοικτό περιβάλλον Tileworld, καθώς και για τη δημιουργία έξυπνων συνεργατικών πρακτόρων περιβαλλόντων διεπαφής. Στα επόμενα δύο υποενότητες παρουσιάζουμε αναλυτικά τους δύο αυτούς τύπους φυσικών περιβαλλόντων.

### 2.5.1. Μελέτη περίπτωσης του δυναμικού φυσικού περιβάλλοντος Tileworld

Το Tileworld είναι ένα δυναμικό και απρόβλεπτο περιβάλλον εξομοίωσης με δυνατότητες παραμετροποίησης που προτάθηκε από τους M. Pollack και M. Ringuette (1990) για την αξιολόγηση αρχιτεκτονικών νοημόνων (βασισμένων στο μοντέλο του BDI) πρακτόρων. Παρόλο που το πλαίσιο εργασίας ICAGENT έχει χρησιμοποιηθεί για την κατασκευή πρακτόρων ικανών να συνεργάζονται με τους ανθρώπους σε απρόβλεπτα περιβάλλοντα όπως τα περιβάλλοντα διεπαφής [Kourakos and Vouros, 2006], χρησιμοποιώντας το Tileworld ως φυσικό περιβάλλον στοχεύουμε στο να κατανοήσουμε τη δυνατότητα της αρχιτεκτονικής του ICAGENT στο να κατασκευάζουμε ανοικτές ομάδες πρακτόρων με δυνατότητες μεταξύ τους συνεργασίας και ευελιξίας συμπεριφοράς σε απρόβλεπτες αλλαγές του φυσικού περιβάλλοντος.

Το Tileworld αποτελείται από μία σκακιέρα με  $M \times N$  τετράγωνα η οποία περιλαμβάνει τέσσερα είδη οντοτήτων: εμπόδια, κύβους, τρύπες και πράκτορες. Ανά σταθερά χρονικά διαστήματα μερικών δευτερολέπτων, η κατάσταση του φυσικού αυτού περιβάλλοντος μπορεί να αλλάξει: νέοι κύβοι και εμπόδια μπορεί να εμφανιστούν, παλιά μπορεί να εξαφανιστούν ενώ μερικά από αυτά μπορεί να μετακινούνται κατά ένα τετράγωνο προς τυχαία κατεύθυνση. Οι τρύπες εμφανίζονται τυχαία αλλά εξαφανίζονται μετά από ένα δεδομένο χρονικό διάστημα μερικών δευτερολέπτων (TTL), ενώ ομάδες εμποδίων μπορούν να σχηματίζουν αδιέξοδα για τους πράκτορες. Νέοι πράκτορες μπορεί να εισέρχονται στη σκακιέρα σε τυχαίες θέσεις. Οι πράκτορες μπορούν να αντιλαμβάνονται μέρος της κατάστασης του φυσικού περιβάλλοντος και να γνωρίζουν τα δεδομένα χρονικά διαστήματα που εξαφανίζονται οι τρύπες.

Το Tileworld είναι ένα δυναμικό περιβάλλον επειδή η κατάσταση του αλλάζει συνεχώς χωρίς τη παρέμβαση των πρακτόρων, ενώ είναι απρόβλεπτο λόγω του ότι η επόμενη κατάστασή του δεν μπορεί να καθοριστεί από τη τρέχουσα και τις τρέχουσες ενέργειες των πρακτόρων. Εάν θεωρήσουμε τους πράκτορες στο Tileworld ως ένα σύστημα, τότε αυτό είναι ανοικτό, διότι ο πληθυσμός των πρακτόρων αλλάζει καθώς ο χρόνος περνά.



Εικόνα 6: Μία διάταξη της σκακιέρας του Tileworld.

Όλοι οι πράκτορες στο Tileworld είναι ικανοί να αντιλαμβάνονται τα γεγονότα και τις ενέργειες των άλλων πρακτόρων που λαμβάνουν χώρο στο φυσικό περιβάλλον σε

δεδομένη ακτίνα από τις τρέχουσες θέσεις τους, καθώς και από τις θέσεις άλλων αντικείμενων ενδιαφέροντος (π.χ. κύβους, τρύπες και άλλους πράκτορες). Η τελευταία ικανότητα αντανακλά την ικανότητα των πρακτόρων να εστιάζουν τη προσοχή τους σε συγκεκριμένες περιοχές ενδιαφέροντος. Το πεδίο αντίληψης εξαρτάται από τις ικανότητες των πρακτόρων και μπορεί να ποικίλει από πράκτορα σε πράκτορα. Η Εικόνα 6 δείχνει μία διάταξη σκακιέρας 45x30.

Ο πρωταρχικός στόχος κάθε πράκτορα είναι να γεμίσει όσο το δυνατόν περισσότερες τρύπες με κυβάρια. Κάθε πράκτορας είναι ικανός να εκτελέσει έναν αριθμό ενεργειών του πεδίου εφαρμογής (domain actions): (α) μετακίνηση του εαυτού του κατά μία θέση αριστερά, δεξιά και πάνω (αλλά όχι προς τα πίσω), (β) μεταφορά ενός εμποδίου που βρίσκεται μπροστά του κατά μία θέση αριστερά ή δεξιά του (αλλά όχι πίσω του ή διαγώνια), (γ) ανασήκωση και μεταφορά ενός κύβου (υποθέτουμε ότι ένας πράκτορας δεν μπορεί να μεταφέρει παραπάνω από ένα κύβο τη φορά) (δ) τοποθέτηση του κύβου που μεταφέρει σε τρύπα, και (ε) επικοινωνία με άλλους πράκτορες.

Επιπλέον, κάθε εμπόδιο έχει ένα συγκεκριμένο βάρος. Έτσι κάθε πράκτορας έχει τη δυνατότητα να μετακινεί εμπόδια μέχρι ενός συγκεκριμένου βάρους. Επίσης υποθέτουμε ότι η κίνηση κάθε πράκτορα από ένα τετράγωνο στο αμέσως επόμενο χρειάζεται συγκεκριμένο χρόνο ο οποίος διαφέρει από πράκτορα σε πράκτορα. Έτσι έχουμε πράκτορες οι οποίοι μετακινούνται πιο γρήγορα από κάποιους άλλους.

Πρέπει να τονίσουμε ότι κάθε οντότητα και χαρακτηριστικό του Tileworld αναπαριστά συγκεκριμένες οντότητες και χαρακτηριστικά πραγματικών δυναμικών περιβαλλόντων. Για παράδειγμα,

- οι τρύπες αναπαριστούν τους στόχους των πρακτόρων,
- τα εμπόδια και τα αδιέξοδα αναπαριστούν τους περιορισμούς που επιβάλλει το περιβάλλον,
- ο χρόνος ζωής (TTL) των τρυπών και ο αριθμός των κύβων είναι πόροι του φυσικού περιβάλλοντος που οι πράκτορες μοιράζονται μεταξύ τους,
- η συχνότητα με την οποία τα εμπόδια, οι κύβοι και οι τρύπες εμφανίζονται και εξαφανίζονται από το φυσικό περιβάλλον καθορίζουν την δυναμική φύση του περιβάλλοντος,

- ο αριθμός των εμποδίων και των αδιεξόδων που ο κάθε πράκτορας μπορεί να συναντήσει αναπαριστά την εχθρότητα του περιβάλλοντος,
- ο αριθμός των πρακτόρων που μπορούν να μοιραστούν πληροφορία ή να συνεργαστούν αναπαριστά την φιλικότητα του περιβάλλοντος,
- το μέρος του φυσικού περιβάλλοντος που κάθε πράκτορας επιθεωρεί αναπαριστά το πεδίο αντίληψης που έχει ο κάθε πράκτορας για το περιβάλλον του.

Στις ακόλουθες παραγράφους, παρουσιάζουμε ένα σενάριο στο περιβάλλον του Tileworld στο οποίο δείχνουμε τον τρόπο με τον οποίο ένας πράκτορας προσαρμόζει τη συμπεριφορά του ανάλογα με τις αλλαγές που συντελούνται στο περιβάλλον του.

Θεωρούμε ότι η αρχική διάταξη της σκακιέρας είναι αυτή της Εικόνα 6. Στο σχήμα αυτό υπάρχουν τρεις πράκτορες στη σκακιέρα: ο  $G_1$ , ο  $G_2$  και ο  $G_3$  στα τετράγωνα (5,19), (28,8) και (36,27) αντίστοιχα. Ο πράκτορας  $G_1$  έχει δεσμευτεί να γεμίσει τη τρύπα  $H_1$  στο τετράγωνο (23,19) με το κύβο  $T_1$  από το τετράγωνο (19,19). Οι πράκτορες  $G_2$  και  $G_3$  έχουν δεσμευτεί να γεμίσουν τις τρύπες  $H_2$  στο (42,12) και  $H_3$  στο (36,20) με τους κύβους  $T_2$  (41,8) και  $T_3$  (40,22), αντίστοιχα.

Ας υποθέσουμε ότι τοποθετούμε έναν ακόμη πράκτορα, ας πούμε τον  $G_0$  στο τετράγωνο (10,4). Ο νέος πράκτορας αντιλαμβάνεται ότι η πιο κοντινή τρύπα βρίσκεται στο τετράγωνο (19,11) ενώ ο πιο κοντινός κύβος βρίσκεται στη θέση (10,16). Αναγνωρίζοντας τη κατάσταση αυτή ο πράκτορας δεσμεύεται να γεμίσει τη τρύπα αυτή με το κύβο. Για να πετύχει το στόχο αυτό βρίσκει ένα μονοπάτι προς το κύβο και μετά προς τη τρύπα. Ο πράκτορας ξεκινά να διασχίσει το μονοπάτι αυτό με αντανακλαστικό τρόπο μέχρι να εντοπίσει ένα από τα ακόλουθα γεγονότα: (α) ένα αδιέξοδο, (β) άλλους πράκτορες που μπορεί να χρησιμοποιούν του ίδιους πόρους (δηλαδή να σκοπεύουν να γεμίσουν την ίδια τρύπα ή να θέλουν να μεταφέρουν το ίδιο κύβο), και (γ) άλλους πράκτορες που μπορούν να βοηθήσουν είτε στο να μεταφέρει πιο γρήγορα το κύβο ή στο να μετακινήσει ένα εμπόδιο το οποίο βρίσκεται στο δρόμο του. Στη τελευταία περίπτωση ο πράκτορας αναγνωρίζει τη δυνατότητα συνεργασίας με άλλους πράκτορες.

Ο πράκτορας ακολουθώντας το μονοπάτι αποφεύγει τα εμπόδια που ξαφνικά βρίσκονται μπροστά του αντανακλαστικά μετακινώντας τα κατά ένα τετράγωνο. Σε αυτές τις περιπτώσεις, ο πράκτορας δεν χρειάζεται να εξετάσει άλλες εναλλακτικές λύσεις, να αναθεωρήσει τις δεσμεύσεις του ή να εγκαταλείψει κάποιες από αυτές. Ο τρόπος συ-

μπεριφοράς τού πράκτορα όσο αναφορά την εκτέλεση της σύνθετης ενέργειας «γέμισε την τρύπα» δεν αλλάζει μέχρι ότου εντοπίσει είτε ένα νέο πράκτορα που να είναι κοντά στο κύβο, είτε ένα εμπόδιο που συνιστά αδιέξοδο στο μονοπάτι που διασχίζει.

Στη περίπτωση που ο πράκτορας εντοπίζει ένα νέο πράκτορα κοντά στο κύβο, η συμπεριφορά του αλλάζει από αντανακλαστική σε κοινωνικά στοχαστική (social deliberative). Οι πράκτορες επικοινωνούν προσπαθώντας να αναγνωρίσουν τη δυνατότητα για συνεργασία<sup>8</sup>, να διαπραγματευτούν για τη μέθοδο που θα ακολουθήσουν στο γέμισμα της τρύπας από κοινού, και για να συμφωνήσουν στην ανάθεση εργασιών μεταξύ τους. Είναι σημαντικό να σημειώσουμε ότι ενώ οι πράκτορες συνεργάζονται στο γέμισμα της τρύπας, ο κάθε πράκτορας μπορεί να έχει διαφορετικές, ατελείς και ασύμβατες γνώμες για τον κόσμο και τις νοητικές καταστάσεις των άλλων πρακτόρων. Έτσι, για να δράσουν από κοινού, απαιτείται επικοινωνία για να αποφευχθεί ο λανθασμένος συντονισμός.

Επιστέφοντας στο παράδειγμα μας, αναγνωρίζοντας τη δυνατότητα για συνεργασία, ο πράκτορας  $G_0$  ζητά από τον πράκτορα  $G_1$  να δουλέψουν από κοινού για να γεμίσουν την τρύπα  $H_0$ . Ο  $G_1$ , όντως αρχικά καλοπροαίρετος (αυτή η υπόθεση ισχύει για όλους τους πράκτορες στη σκακιέρα), δέχεται το αίτημα του  $G_0$  και δεσμεύεται να εκτελέσει την ενέργεια από κοινού με τον  $G_0$ . Κάνοντας αυτό, ο  $G_1$  προσπαθεί να συμβιβάσει (reconcile) την μέθοδο που προτείνει ο  $G_0$  με τη νοητική του κατάσταση καθώς και με άλλες προθέσεις που ο ίδιος έχει στο πλαίσιο δράσης του. Αναγνωρίζοντας ότι δεν λαμβάνει χώρο καμία σύγκρουση (επειδή πιστεύει ότι ο χρόνος ζωής της τρύπας  $H_1$  που θέλει να γεμίσει δεν θα εξαντληθεί εάν βοηθήσει τον  $G_0$  να γεμίσει τη τρύπα  $H_0$ ), ο  $G_1$  πληροφορεί τον  $G_0$  ότι δέχεται τη προτεινόμενη μέθοδο, η οποία στη συνέχεια γίνεται κοινή.

---

<sup>8</sup> Ο πράκτορας  $G_0$  ζητά πληροφορία από τον  $G_1$  για τους στόχους και τις δυνατότητες του. Ο  $G_1$  επιστρέφει ένα πλαίσιο (template) με τη θέση της τρύπας που προσπαθεί να γεμίσει, τη θέση του κύβου, την ταχύτητα του (τετράγωνο ανά λεπτό) και το μέγιστο βάρος των εμποδίων που μπορεί να μετακινήσει. Από αυτό το πλαίσιο, μπορεί ο πράκτορας  $G_0$  να αναγνωρίσει εάν ο  $G_1$  είναι ταχύτερος ή πιο κοντά στο κύβο  $T_0$  και επομένως, εάν μπορεί να βοηθήσει στο να πάρει το κύβο γρηγορότερα (π.χ. φέρνοντας το κύβο  $T_0$  πλησιέστερα στον  $G_0$ ).



Η μέθοδος (recipre) αυτή περιλαμβάνει τρεις ενέργειες: «βρες το μονοπάτι προς το κύβο και μετά προς τη τρύπα», «πάρε το κύβο» και «ρίξε το κύβο στη τρύπα». Σύμφωνα με τη κοινή μέθοδο, οι πράκτορες συμφωνούν να δράσουν ως εξής: (α) κάθε πράκτορας να εκτελέσει προσωπικά τη πρωταρχική ενέργεια «βρες το μονοπάτι προς το κύβο και μετά προς τη τρύπα», (β) να εκτελέσουν σε συνεργασία τη σύνθετη ενέργεια «πάρε το κύβο» και τέλος, (γ) να εκτελέσει ο πράκτορας  $G_0$  τη σύνθετη ενέργεια «ρίξε το κύβο στη τρύπα». Κάθε πράκτορας δεσμεύεται ότι κάθε μία από τις ενέργειες αυτές θα εκτελεσθεί επιτυχώς είτε την εκτελέσει ο ίδιος είτε ο άλλος πράκτορας είτε και οι δύο μαζί. Αυτό έχει ως αποτέλεσμα ο κάθε πράκτορας να δεσμεύεται να βοηθήσει τον άλλο να πετύχει το στόχου του (ή τους στόχους τους) εκτελώντας ενέργειες για λογαριασμό του άλλου, είτε απλά προσέχοντας οι ενέργειες που εκτελούν να μην έρχονται σε σύγκρουση με τους στόχους του συνεργάτη του.

Αυτού του είδους οι δεσμεύσεις έχουν τις ακόλουθες συνέπειες στη συμπεριφορά των πρακτόρων:

(α) Το τετράγωνο (και κατά επέκταση το μονοπάτι που θα βρει προς αυτό) στο οποίο ο πράκτορας θα μετακινήσει το κύβο πρέπει να είναι στο μονοπάτι του  $G_0$  και όσο πιο κοντά μπορεί σε αυτόν.

(β) Στη περίπτωση που ο πράκτορας  $G_0$  αλλάξει το μονοπάτι του και εάν το τετράγωνο που ο  $G_1$  θα μεταφέρει το κύβο σ' αυτό δεν είναι στο νέο μονοπάτι του  $G_0$ , η δέσμευση του  $G_0$  ότι η ομάδα ( $G_0, G_1$ ) θα εκτελέσει την ενέργεια «γέμισε τη τρύπα» επιτυχώς τον οδηγεί στο να επικοινωνήσει αυτήν την αλλαγή στον  $G_1$ .

(γ) Εάν ο  $G_0$  ή ο  $G_1$  εντοπίσει ένα εμπόδιο στον δρόμο του, τότε πρέπει να μετακινήσει το εμπόδιο σε ένα μέρος που δεν θα εμποδίζει το δρόμο στον άλλον.

Είναι σημαντικό να παρατηρήσουμε ότι έχοντας αρχίσει μία περίοδο συνεργασίας, οι πράκτορες διασχίζουν το μονοπάτι τους προς το κύβο με αντανακλαστικό τρόπο. Οι πράκτορες ακολουθούν τη σχεδιασμένη διαδρομή και αντιδρούν στις αλλαγές του φυσικού περιβάλλοντος μέχρις ότου εντοπισθεί κάποια αδιέξοδο.

Κάθε πράκτορας μπορεί να συνεργαστεί με περισσότερους από έναν πράκτορες, όποτε αυτό χρειάζεται. Για παράδειγμα, κατά τη διάρκεια της περιόδου συνεργασίας με τον  $G_1$ , εάν ο πράκτορας  $G_0$  εντοπίσει ένα αδιέξοδο στο μονοπάτι του προς την  $H_0$ , τότε

πριν φτάσει εκεί, αλλάζει τη συμπεριφορά του από ανταντακλαστική σε κοινωνικώς ή ατομικώς στοχαστική (social or individual deliberative). Στη πρώτη περίπτωση ο πράκτορας εισάγει στην αρχική ομάδα συνεργασίας ( $G_0, G_1$ ) τον πράκτορα  $G_2$  που έχει την ικανότητα να μετακινήσει το εμπόδιο που συνιστά το αδιέξοδο. Στην περίπτωση του κοινωνικού στοχασμού, και στην περίπτωση που οι πράκτορες  $G_0, G_1, G_2$  πετυχαίνουν να σχηματίσουν μία ομάδα συνεργατών, θα μοιραστούν το κοινό στόχο «πάρε το κύβο» καθώς και τον λόγο για τον οποίο η ενέργεια αυτή πραγματοποιείται δηλαδή την «γέμισε τη τρύπα».

Στη περίπτωση του ατομικού στοχασμού, ο πράκτορας προσπαθεί να βρει ένα εναλλακτικό πλάνο για να γεμίσει τη τρύπα, συμβιβάζοντας (reconciled) το πλάνο αυτό με τη τρέχουσα νοητική του κατάσταση και τις άλλες δεσμεύσεις που μπορεί να έχει. Αφού αντιμετωπίσει το πρόβλημα, ο πράκτορας συνεχίζει να διασχίζει το μονοπάτι με ανταντακλαστικό τρόπο.

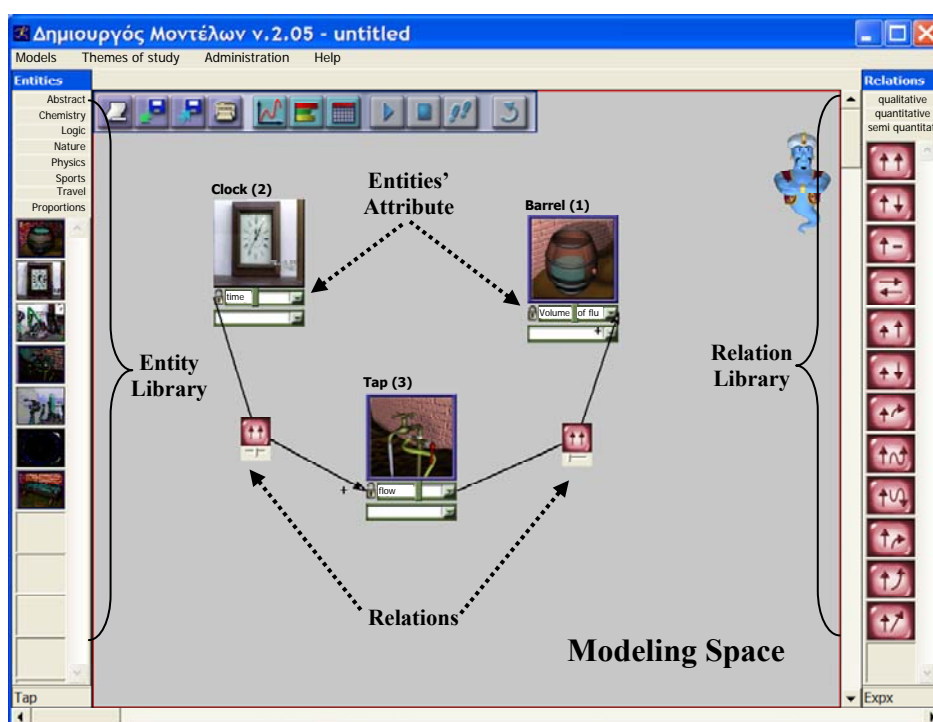
Συνοψίζοντας, για να μπορεί να δρα ένας πράκτορας στο περιβάλλον του Tileworld θα πρέπει να είναι ικανός να παρακολουθεί τις αλλαγές που συμβαίνουν στο περιβάλλον του, να αναγνωρίζει γρήγορα τις καταστάσεις στις οποίες πρέπει να πάρει τη πρωτοβουλία να δράσει, καθώς και τις προβληματικές καταστάσεις, και να ενεργεί είτε μόνος του είτε σε συνεργασία με άλλους πράκτορες. Λόγω των περιορισμένων και συνεχώς εξαντλούμενων πόρων του φυσικού περιβάλλοντος ο πράκτορας για να μπορεί να δρα αποτελεσματικά θα πρέπει να έχει την ικανότητα να επιλέγει όχι μόνο το κατάλληλο τύπο συμπεριφοράς για κάθε εργασία του πεδίου εφαρμογής, αλλά θα πρέπει να μπορεί να προσαρμόζει τη συμπεριφορά του σύμφωνα με τις πεποιθήσεις και τις προθέσεις που έχει καθώς και με τη νοητική κατάσταση των συνεργατών του.

### 2.5.2. Μελέτη περίπτωσης φυσικού περιβάλλοντος επιφάνειας διεπαφής

Ως μελέτη περίπτωσης φυσικού περιβάλλοντος επιφάνειας διεπαφής χρησιμοποιούμε τη διεπαφή της εφαρμογής Δημιουργός Μοντέλων (Models Creator) (βλ. Εικόνα 7). Η συγκεκριμένη εφαρμογή παρέχει ένα περιβάλλον ανάπτυξης μοντέλων του πραγματικού κόσμου για παιδιά από 10 έως 14 ετών [Dimitracopoulou et. al., 1999].

Τα αντικείμενα που συνθέτουν το περιβάλλον του μοντέλου χωρίζονται σε οντότητες και σχέσεις. Κάθε οντότητα έχει ιδιότητες που αναπαριστούν τα χαρακτηριστικά της

οντότητας. Για παράδειγμα, η οντότητα «βαρέλι» έχει την ιδιότητα «χωρητικότητα» και την ιδιότητα «όγκος του υγρού». Η οντότητα «βρύση» έχει την ιδιότητα «ροή» και η οντότητα «ρολόι» έχει την ιδιότητα «χρόνος». Σχέσεις μπορούν να υπάρξουν μόνο μεταξύ ιδιοτήτων των οντοτήτων. Η Εικόνα 7 μας δείχνει ένα απλό μοντέλο το οποίο αποτελείται από τρεις οντότητες (ένα ρολόι, ένα βαρέλι και μία βρύση) και δύο σχέσεις μεταξύ αυτών. Η ιδιότητα «χρόνος» της οντότητας «ρολόι» σχετίζεται με την ιδιότητα «ροή» της οντότητας «βρύση», και η τελευταία ιδιότητα σχετίζεται με την ιδιότητα «όγκο του υγρού» της οντότητας «βαρέλι». Η σχέση μεταξύ κάθε ζευγαριού αυτών των ιδιοτήτων προσδιορίζεται ως ευθεία αναλογία, δηλαδή όσο αυξάνεται η τιμή μίας ιδιότητας ανάλογα αυξάνεται και η άλλη (για παράδειγμα όσο περνάει ο χρόνος τόσο αυξάνεται η ροή της βρύσης και άρα τόσο αυξάνεται ο όγκος του υγρού στο βαρέλι).



Εικόνα 7: Η εφαρμογή διεπαφής του Δημιουργού Μοντέλων: Ο μαθητής μπορεί να δημιουργήσει μοντέλα πραγματικών καταστάσεων φτιάχνοντας σχέσεις μεταξύ οντοτήτων. Τις σχέσεις και τις οντότητες τις επιλέγει από τις αντίστοιχες βιβλιοθήκες. Τέτοιες σχέσεις μπορεί να είναι ποσοτικές ή ποιοτικές. Οι μαθητές μπορούν να πειραματιστούν με διάφορους σχηματισμούς μέχρις ότου προσεγγίσουν έναν που μοντελοποιεί το δεδομένο πρόβλημα. Ο σχηματισμός σε αυτό το σχήμα μοντελοποιεί τη σχέση μεταξύ του χρόνου, της ροής μίας βρύσης και του όγκου ενός βαρελιού.

Για να συσχετιστούν δύο οντότητες, ο χρήστης πρέπει να εκτελέσει τις ακόλουθες

σύνθετες και βασικές ενέργειες:

- (1) καθορισμός των οντοτήτων που θα συσχετιστούν (συμπεριλαμβανομένου και των ιδιοτήτων αυτών),
- (2) επιλογή μίας βιβλιοθήκης σχέσεων,
- (3) σύρσιμο και απόθεση μιας σχέσης από τη βιβλιοθήκη σχέσεων στο χώρο δημιουργίας μοντέλων,
- (4) σύνδεση των ιδιοτήτων των οντοτήτων με τη σχέση.



Εικόνα 8: Συσχετίζοντας δύο οντότητες με μία σχέση στο Δημιουργό Μοντέλων.

Για παράδειγμα, το αποτέλεσμα συσχετισμού των ιδιοτήτων «όγκος υγρού» και «χρόνος» των οντοτήτων «βαρέλι» και «ρολόι» φαίνεται στην Εικόνα 8(α).

Ένα συνηθισμένο λάθος που κάνουν τα παιδιά είναι να προσπαθούν να συσχετίσουν δύο οντότητες χωρίς να έχουν προσδιορίσει εκ των προτέρων τις ιδιότητες τους. Σε αυτή τη περίπτωση, τα παιδιά σύρουν τη σχέση στο χώρο δημιουργίας μοντέλων και προσπαθούν να συσχετίσουν τις οντότητες όπως στην Εικόνα 8(β). Το σύστημα αντιδρά σε αυτή τη συμπεριφορά διαγράφοντας τη σχέση και περιμένοντας την επόμενη ενέργεια του χρήστη.

Παρόλο που η επιφάνεια διεπαφής αντιδρά με ορθό τρόπο, ούτε εξηγεί ούτε παρέχει κάποια βοηθητική υπόδειξη στον χρήστη ως προς το τί πρέπει να κάνει για να δημιουργήσει μία σχέση μεταξύ οντοτήτων. Αυτή είναι μια τυπική επιφάνεια διεπαφής η οποία δεν συνεργάζεται με τον χρήστη αλλά αλληλεπιδρά με αυτόν χωρίς να κατανοεί τις επιθυμίες και τους στόχους του με αποτέλεσμα να μην επιδεικνύει εξυπηρετική συμπεριφορά ως προς την επίτευξη των στόχων του χρήστη.

Η διεπαφή του Δημιουργού Μοντέλων διευκολύνει τη σταδιακή εκμάθηση των παρεχόμενων λειτουργιών επιτρέποντας στους χρήστες να πειραματιστούν και να παρατηρήσουν τα αποτελέσματα των ενεργειών τους. Αυτή είναι μία διεπαφή άμεσου χειρισμού λόγω των ακολούθων [Shneiderman, 1997]: (α) Οι πράξεις που μπορούν να εκτελέσουν οι χρήστες είναι ορατές μέσω των αντικειμένων της διεπαφής. Για παράδειγμα, όταν ο χρήστης σύρει μία σχέση στο χώρο δημιουργίας μοντέλων, τότε δημιουργείται μία γραμμή (από τη σχέση προς το σημείο κατάδειξης (mouse pointer)) που δείχνει ότι το σύστημα αναμένει από τον χρήστη να συνδέσει τη σχέση. (β) Τα αποτελέσματα των ενεργειών αναπαριστώνται οπτικά. Για παράδειγμα, όταν ένας χρήστης επιλέγει μία ιδιότητα μιας οντότητας, τότε μία άκρη δημιουργείται μεταξύ αυτής της ιδιότητας και της σχέσης. (γ) Η ανταπόκριση που λαμβάνουν οι χρήστες είναι άμεση και συνεχής. Για παράδειγμα, όπως έχουμε ήδη δείξει, σε περίπτωση που ο χρήστης προσπαθεί να συσχετίσει τις οντότητες μεταξύ τους αντί τις ιδιότητες τους, το σύστημα αυτόματα διαγράφει τη σχέση από το χώρο δημιουργίας μοντέλων. (δ) Οι πράξεις είναι αντιστρέψιμες. (ε) Ο χρήστης πάντα αναλαμβάνει την ευθύνη των πράξεων του και δεν υπάρχει άλλο λογισμικό μεταξύ του χρήστη και της διεπαφής της εφαρμογής.

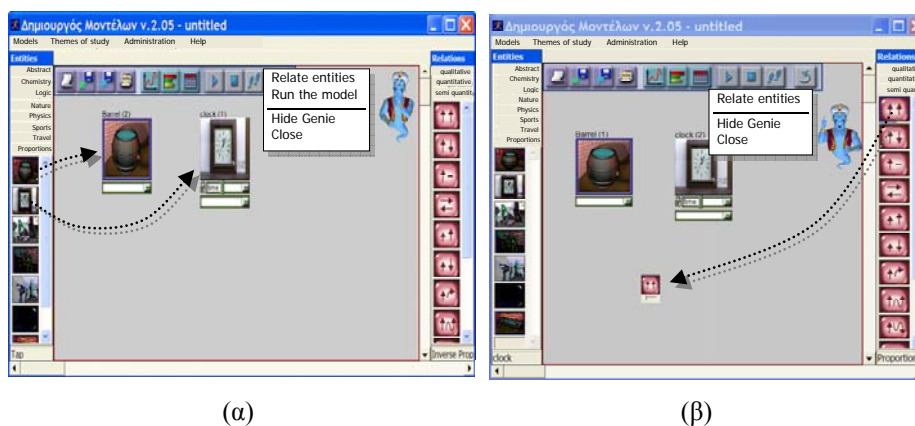
Ένας πράκτορας ο οποίος συνεργάζεται με τον χρήστη σε μία επιφάνεια διεπαφής άμεσου χειρισμού όπως αυτή του Δημιουργού Μοντέλων, πρέπει να αντιλαμβάνεται τα γεγονότα που συμβαίνουν στην επιφάνεια διεπαφής, να αναγνωρίζει τις επιθυμίες του χρήστη, να είναι σε συνεχή επικοινωνία μαζί του και να εντοπίζει την ανάγκη για συνεργασία, παρέχοντας εξυπηρετική συμπεριφορά.

Ας εξετάσουμε ένα συγκεκριμένο σενάριο συνεργασίας που αποκαλύπτει τους πιο σημαντικούς παράγοντες που αφορούν τον πράκτορα συνεργατικής διεπαφής του Δημιουργού Μοντέλων. Το σενάριο παρουσιάζεται σε μορφή διαλόγου μεταξύ του χρήστη και του πράκτορα και δείχνει πως ο πράκτορας:

- σχηματίζει υποθέσεις για τις επιθυμίες του χρήστη,
- επικοινωνεί με τον χρήστη,
- αρχίζει τη συνεργασία μαζί του (initiates collaboration) και
- εκτελεί εργασίες από κοινού με τον χρήστη.

*Χρήστης:* Όπως δείχνει η Εικόνα 9(α), ο χρήστης εισάγει την οντότητα «ρολόι» και την οντότητα «βαρέλι» στον (αρχικά άδειο) χώρο δημιουργίας μοντέλων, αλλά προσδιορίζει μόνο την ιδιότητα «χρόνο» του «ρολογιού».

*Πράκτορας:* Ο πράκτορας, αντιλαμβανόμενος τα γεγονότα που αντιστοιχούν σε αυτές τις ενέργειες, σχηματίζει εναλλακτικές υποθέσεις που αφορούν τις επιθυμίες του χρήστη και παρουσιάζει αυτές τις υποθέσεις στον χρήστη: Ο πράκτορας υποθέτει ότι ο χρήστης επιθυμεί είτε να συσχετίσει τις οντότητες ("Relate entities") είτε να τρέξει το μοντέλο ("Run the model"). Όπως δείχνει η Εικόνα 9(α), ο πράκτορας παρουσιάζει αυτές τις υποθέσεις χρησιμοποιώντας ένα πλαίσιο διαλόγου. Ο χρήστης μπορεί να ζητήσει από τον πράκτορα βοήθεια επιλέγοντας μία από αυτές τις υποθέσεις. Ενεργώντας έτσι, ο χρήστης προσδιορίζει την επιθυμία του να εκτελέσει την αντίστοιχη ενέργεια και αρχίζει (initiates) τη μεταξύ τους συνεργασία.



Εικόνα 9: (α) Ο χρήστης εισάγει τις οντότητες «βαρέλι» και «ρολόι», αλλά δεν ορίζει μία ιδιότητα για το βαρέλι. (β) Ο χρήστης σύρει και αφήνει μία σχέση στο χώρο δημιουργίας μοντέλων. Ο πράκτορας αναγνωρίζει ένα πιθανό πρωταρχικό στόχο του χρήστη και δίνει την επιλογή σ' αυτόν να ξεκαθαρίσει την επιθυμία του εμφανίζοντας ένα πλαίσιο διαλόγου με το στόχο αυτό.

*Χρήστης:* Ας υποθέσουμε ότι ο χρήστης αγνοεί τις υποθέσεις του πράκτορα και μεταφέρει μία σχέση από την βιβλιοθήκη σχέσεων στον χώρο δημιουργίας μοντέλων (Εικόνα 9(β)).

*Πράκτορας:* Ο πράκτορας αντιλαμβάνεται το καινούργιο αυτό γεγονός και ελαττώνει το σύνολο των αρχικών του υποθέσεων σε μία. Ο χρήστης προσπαθεί να εκτελέσει την ενέργεια "Relate entities". Αυτή η υπόθεση σχηματίζεται λόγω του ότι η ενέργεια της

μεταφοράς της σχέσης στο χώρο δημιουργίας μοντέλων δε συνεισφέρει στην εκτέλεση της ενέργειας "Run the model".

Ο πράκτορας όντας σε συνεχή επικοινωνία με τον χρήστη και για να εξηγήσει τις υποθέσεις του, χρησιμοποιεί ένα πλαίσιο διαλόγου για να παρουσιάσει τις υποθέσεις που αφορούν τις επιθυμίες του χρήστη. Αυτό επιτρέπει στον χρήστη να ξεκαθαρίσει τις επιθυμίες του και να αρχίσει τη συνεργασία με τον πράκτορα.

Έχοντας μία μοναδική υπόθεση για τις επιθυμίες του χρήστη<sup>9</sup>, ο πράκτορας εστιάζεται στην εκτέλεση αυτής της ενέργειας και ερμηνεύει τις ενέργειες του χρήστη υπό το πλαίσιο αυτής της ενέργειας.

Ερμηνεύοντας τις ενέργειες του χρήστη στο πλαίσιο μίας μόνο υπόθεσης επιτρέπει στο πράκτορα ν' αναγνωρίσει προβληματικές καταστάσεις που συμβαίνουν στην πορεία της ενέργειας και να παρέχει βοήθεια ως προς την επίλυση αυτών των προβλημάτων αυτών.

*Χρήστης:* Ο χρήστης συνεχίζει την προσωπική του πορεία ενεργειών κάνοντας κλικ σε μία ιδιότητα της οντότητας «ρολόι» για να συνδέσει τη σχέση που έχει επιλέξει με το «ρολόι» (Εικόνα 10(α)).

*Πράκτορας:* Ο πράκτορας ερμηνεύει αυτή την ενέργεια ως μία βοηθητική ενέργεια της σύνθετης ενέργειας "Link relation to entities", που με την σειρά της αναγνωρίζεται από τον πράκτορα ως μία βοηθητική ενέργεια της ενέργειας "Relate entities".

*Χρήστης:* Ο χρήστης κάνει κλικ στην οντότητα «βαρέλι» για να ολοκληρώσει την σύνδεση της σχέσης με τις αντίστοιχες οντότητες (Εικόνα 10(β)).

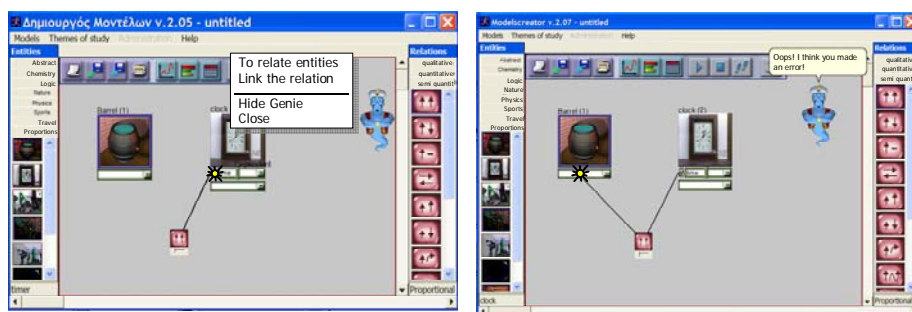
Η απόκριση της διεπαφής του Δημιουργού Μοντέλων είναι να διαγράψει την σχέση, διότι ο χρήστης δεν έχει επιλέξει καμία ιδιότητα του βαρελιού.

*Πράκτορας:* Σε αυτή τη περίπτωση ο πράκτορας αναγνωρίζει μία σύγκρουση: Οι περιορισμοί για την εκτέλεση της ενέργειας "Link relation to entities" δεν ισχύουν. Όπως

---

<sup>9</sup> Αυτό μπορεί να αλλάξει σε ένα μεταγενέστερο χρονικό σημείο. Η εκτέλεση νέων ενεργειών μπορεί να οδηγήσει τον πράκτορα να σχηματίσει παραπάνω από μία υποθέσεις όσον αναφορά τις επιθυμίες του χρήστη.

δείχνει η Εικόνα 10(β), ο πράκτορας δείχνει ότι κάτι πήγε λάθος, παρακινώντας το χρήστη να συνεργαστεί μαζί του.



(α)

(β)

Εικόνα 10: (α) Ο χρήστης συνδέει την σχέση με μία ιδιότητα του «ρολογιού» επιτυχώς ενώ ο πράκτορας παρουσιάζει τις νέες υποθέσεις για τις επιθυμίες του χρήστη: για να συσχετίσεις τις οντότητες (To relate entities) σύνδεσε τη σχέση με τα αντικείμενα (Link the relation). (β) Ο χρήστης προσπαθεί να συνδέσει την σχέση με την οντότητα «βαρέλι». Ο πράκτορας αναγνωρίζει ότι κάτι πάει λάθος και παρακινεί το χρήστη να συνεργαστεί.

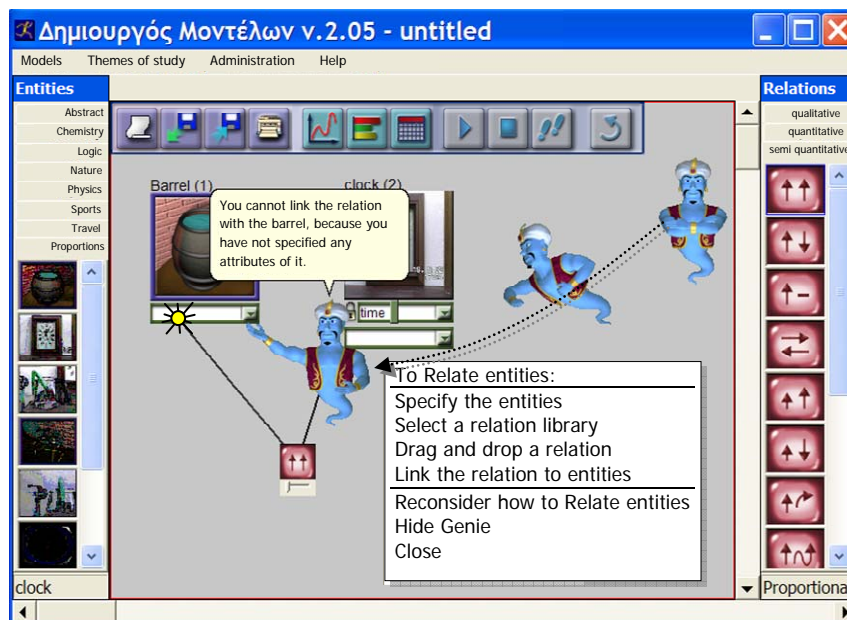
*Χρήστης:* Αν ο χρήστης κάνει κλικ στον πράκτορα ζητώντας βοήθεια, αρχίζει η συνεργασία μεταξύ τους.

*Πράκτορας:* Σε αυτήν την περίπτωση ο πράκτορας εκτελεί τις ακόλουθες ενέργειες: (α) αναγνωρίζει ότι η ενέργεια "Specify the entities" μπορεί να επιλύσει την σύγκρουση, (β) εξηγεί τη σύγκρουση που εντόπισε μετακινώντας τον εαυτό του προς το βαρέλι και λέγοντας ότι «Δεν μπορείς να συνδέσεις την σχέση με το βαρέλι, γιατί δεν έχεις προσδιορίσει καμία ιδιότητα για αυτό», (Εικόνα 11), (γ) δεσμεύεται να συνεργαστεί με το χρήστη για την ενέργεια "Relate entities" (στην εκτέλεση της οποίας συνεισφέρουν τόσο η λανθασμένη ενέργεια όσο και η ενέργεια που επιλύει τη σύγκρουση).

Όπως δείχνει η Εικόνα 11, ο πράκτορας επικοινωνεί τις ενέργειες που πρέπει να εκτελεστούν και παρακινεί το χρήστη για την εκτέλεση τους. Αυτό επιτυγχάνεται χρησιμοποιώντας ένα πλαίσιο διαλόγου τριών πεδίων. Στο πρώτο πεδίο ο πράκτορας παρουσιάζει το στόχο που πρέπει να επιτευχθεί (στο παράδειγμα μας ο στόχος αυτός είναι "Relate entities"). Στο δεύτερο πλαίσιο ο πράκτορας προτείνει μία μέθοδο (recipe) η οποία αν ακολουθηθεί από το χρήστη ή/και το πράκτορα επιτυγχάνεται ο στόχος (στη περίπτωση μας αυτό περιλαμβάνει τις βοηθητικές ενέργειες "Specify the entities", "Select a relation library", "Drag and drop a relation" and "Link the relation to entities"). Στο τρίτο πλαίσιο ο πράκτορας παρέχει επιλογές όπως "Reconsider how to



Relate entities", "Hide Genie", and "Close".

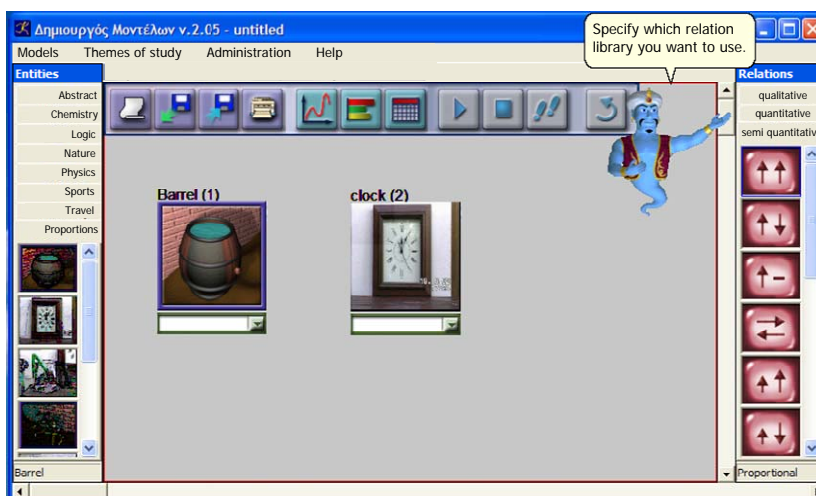


Εικόνα 11: Σε περίπτωση που ο χρήστης κάνει κλικ στο πράκτορα, τότε ο πράκτορας παρουσιάζει τη σύγκρουση που έχει αναγνωρίσει και παρακινεί τον χρήστη να συνεργαστεί.

Γενικά, ο χρήστης και ο πράκτορας αρχίζουν να συνεργάζονται μεταξύ τους για την εκτέλεση μίας ενέργειας στις ακόλουθες περιπτώσεις:

- (α) όταν η ενέργεια έχει παρουσιαστεί ως υπόθεση που αφορά τις επιθυμίες του χρήστη (π.χ. στην Εικόνα 9(α) και (β) και στην Εικόνα 10(α)) και ο χρήστης κάνει κλικ σε αυτή την ενέργεια, ή
- (β) όταν ο πράκτορας εντοπίζει μία προβληματική κατάσταση και ο χρήστης ζητά την βοήθεια του πράκτορα κάνοντας κλικ στον πράκτορα (π.χ. στην Εικόνα 10(β)).

Όντας σε συνεργασία με τον πράκτορα, ο χρήστης μπορεί να εκτελέσει οποιαδήποτε από τις ενέργειες που υποδεικνύονται από τον πράκτορα μέσω του πλαισίου διαλόγου, ή μπορεί να κάνει κλικ σε οποιαδήποτε από τις ενέργειες που υποδεικνύει το πλαίσιο αυτό. Εάν ο χρήστης κάνει κλικ σε μία ενέργεια στο δεύτερο πεδίο του πλαισίου διαλόγου τότε ζητά από τον πράκτορα να εκτελέσει τη πράξη αυτή.



Εικόνα 12: Σε περίπτωση που ο χρήστης ζητήσει από τον πράκτορα να εκτελέσει την ενέργεια «Επέλεξε μία βιβλιοθήκη σχέσεων» (κάνοντας κλικ στην αντίστοιχη επιλογή του πλαισίου ενεργειών) ο πράκτορας αναγνωρίζει ότι δεν μπορεί να αποφασίσει ποια συγκεκριμένη βιβλιοθήκη σχέσεων πρέπει να επιλέξει. Για αυτό το λόγο ζητάει από το χρήστη να καθορίσει την επιθυμητή βιβλιοθήκη.

*Χρήστης:* Ο χρήστης κάνει κλικ στην ενέργεια "Select a relation library" στο δεύτερο πεδίο του μενού επιλογών.

*Πράκτορας:* Ο πράκτορας προσπαθεί να εκτελέσει την ενέργεια αυτή εξυπηρετώντας το αίτημα του χρήστη. Εντούτοις, ο πράκτορας γρήγορα αναγνωρίζει ότι δεν μπορεί να αποφασίσει ποια βιβλιοθήκη σχέσεων να επιλέξει. Όπως δείχνει η Εικόνα 12, ο πράκτορας όντας σε συνεργασία με τον χρήστη, δείχνει προς τις βιβλιοθήκες σχέσεων και ζητά από τον χρήστη να επιλέξει μία.

Συνοψίζοντας, ο πράκτορας παρατηρώντας τις ενέργειες του χρήστη στην επιφάνεια διεπαφής, αρχικά σχηματίζει υποθέσεις για τους βασικούς του στόχους του χρήστη και τους εμφανίζει χρησιμοποιώντας ένα πλαίσιο διαλόγου. Όσο ο χρήστης ενεργεί στην επιφάνεια εργασίας προσπαθώντας να πετύχει ένα στόχο τόσο οι υποθέσεις του πράκτορα περιορίζονται προσεγγίζοντας το στόχο αυτό. Όταν οι υποθέσεις του πράκτορα περιορίζονται σε μία, τότε ερμηνεύοντας τις ενέργειες του χρήστη κάτω από το πλαίσιο ενεργειών που πετυχαίνουν το στόχο αυτό ο πράκτορας εντοπίζει πιθανές λανθασμένες ενέργειες του πράκτορα προτρέποντάς αυτόν να συνεργαστεί μαζί του.

Αν ο χρήστης αποδεχθεί την προτροπή του πράκτορα για συνεργασία ή αν επιλέξει έναν από τους πιθανούς στόχους που του παρουσιάζει, τότε η συνεργασία μεταξύ αυτών αρχίζει για την επίτευξη ενός κοινού στόχου. Ο πράκτορας σε κάθε επιλογή που κάνει σχετικά με τις μεθόδους που επιλέγει τις παρουσιάζει στο χρήστη καθώς και τους λόγους (στόχους) για τους οποίους επιλέγει τις συγκεκριμένες αυτές μεθόδους. Με αυτό τον τρόπο ο χρήστης εμπιστεύεται τις επιλογές και ενέργειες του πράκτορα διότι καθ' όλη τη διάρκεια της συνεργασίας τους ο χρήστης έχει κοινή γνώση με τον πράκτορα σχετικά με το τι προσπαθούν να πετύχουν και με ποιο τρόπο, ενώ ο χρήστης έχει τον πλήρη έλεγχο της συνεργασίας και των ενεργειών στη διεπαφή.

## Κεφάλαιο 3ο:

### Σχετική Βιβλιογραφία

Για την ανάπτυξη του πλαισίου εργασίας ICAGENT έχουν χρησιμοποιηθεί δύο μοντέλα: το BDI μοντέλο ανάπτυξης πρακτόρων και το μοντέλο συνεργατικών πρακτόρων *SharedPlans*. Το κεφάλαιο αυτό παρουσιάζει τα δύο αυτά μοντέλα, σχετικές αρχιτεκτονικές ανάπτυξης πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα καθώς και αρχιτεκτονικές ανάπτυξης έξυπνων συνεργατικών πρακτόρων για περιβάλλοντα διεπαφής.

#### 3.1. Το BDI – μοντέλο ανάπτυξης πρακτόρων

Όπως αναφέραμε στην εισαγωγή, το BDI μοντέλο ανάπτυξης πρακτόρων έχει τις ρίζες του στη κατανόηση της ανθρώπινης πρακτικής συλλογιστικής. Ως σκοπό έχει την υλοποίηση της θεωρίας του M. Bratman δηλαδή τη συσχέτιση των νοητικών δομών της πεποίθησης, της επιθυμίας και της πρόθεσης ενός πράκτορα με τις δύο θεμελιώδεις ενέργειες της πρακτικής συλλογιστικής, αυτής του στοχασμού (deliberation) και της ανάλυσης μέσων και σκοπών (means-end reasoning) [Bratman, 1987; Bratman, 1999].

Το μοντέλο BDI βασίζεται σε τρεις νοητικές δομές: αυτή της πεποίθησης (Belief), της επιθυμίας (Desire) και της πρόθεσης (Intention). Οι δομές αυτές περιγράφουν τη νοητική κατάσταση του πράκτορα σε ότι αφορά τις πληροφορίες (information states) που διαθέτει για το περιβάλλον του, τα κίνητρά του (motivational states), και τους στοχασμούς του (deliberative states) αντίστοιχα [Rao and Georgeff, 1995]. Πιο συγκεκριμένα:

- Η νοητική δομή της πεποίθησης αναπαριστά την άποψη (view) του πράκτορα για γεγονότα που συμβαίνουν στο φυσικό του περιβάλλον (συμπεριλαμβανομένου της νοητικής κατάσταση άλλων πρακτόρων). Η γνώση αυτή, λόγω του ότι συνήθως είναι αποτέλεσμα παρατήρησης του περιβάλλοντός του ή επικοινωνίας με άλλους πρά-

κτορες, μπορεί να μην είναι πλήρης ή ακόμα να είναι και λανθασμένη (αν για παράδειγμα ο πράκτορας παρατηρεί το περιβάλλον του όχι συνεχώς αλλά ανά τακτά χρονικά διαστήματα τότε η γνώση που μπορεί να έχει γ' αυτό μπορεί να μην περιλαμβάνει γεγονότα τα οποία έχουν συμβεί σε ενδιάμεσα χρονικά διαστήματα).

- Η νοητική δομή της επιθυμίας αναπαριστά τις ενέργειες του πράκτορα που έχει επιλέξει να πραγματοποιήσει. Οι επιθυμίες αφορούν ενέργειες που ο πράκτορας δεν έχει εξετάσει αν μπορούν να πραγματοποιηθούν ή αν υπάρχει σύγκρουση με άλλες ενέργειες που έχει επιλέξει να πραγματοποιήσει.
- Σε αντίθεση με τις επιθυμίες, οι προθέσεις ενός πράκτορα αναπαριστούν ενέργειες που όχι μόνο έχει επιλέξει αλλά έχει δεσμευθεί να πραγματοποιήσει διότι έχει ελέγξει αν είναι εφικτό να πραγματοποιηθούν. Οι προθέσεις ενός πράκτορα αποτελούν πάντα υποσύνολο των επιθυμιών του.

Το μοντέλο BDI δίνει τη δυνατότητα σε έναν πράκτορα που δρα σε ένα φυσικό περιβάλλον να αποφασίζει *τί* θέλει να πετύχει και με *ποιο* τρόπο βασιζόμενος πάντα στη γνώση που έχει για αυτό το περιβάλλον, τις δυνατότητές του και τα πλάνα του. Οι αποφάσεις αυτές όμως συνήθως έχουν χρονικό κόστος. Η καθυστέρηση στη λήψη αυτών των αποφάσεων μπορεί να αποτελέσει πρόβλημα σε περιβάλλοντα που αλλάζουν συχνά και υπάρχουν πόροι (του πράκτορα ή του περιβάλλοντος) που κατά τη διάρκεια του χρόνου εξαντλούνται.

Σε φυσικά περιβάλλοντα που η κατάστασή τους αλλάζει με αιτιοκρατικό τρόπο οι δύο θεμελιώδεις διεργασίες συλλογισμού της πρακτικής συλλογιστικής συνήθως μπορούν να περιγραφούν υπολογιστικά χρησιμοποιώντας μαθηματικά μοντέλα. Ακόμα και αν οι πόροι του πράκτορα ή του φυσικού περιβάλλοντος είναι περιορισμένοι συγκεκριμένες υπολογιστικές διαδικασίες μπορούν να χρησιμοποιηθούν για να περιγράψουν πλήρως τη συμπεριφορά του πράκτορα έτσι ώστε να πετυχαίνει αποτελεσματικά τους στόχους του.

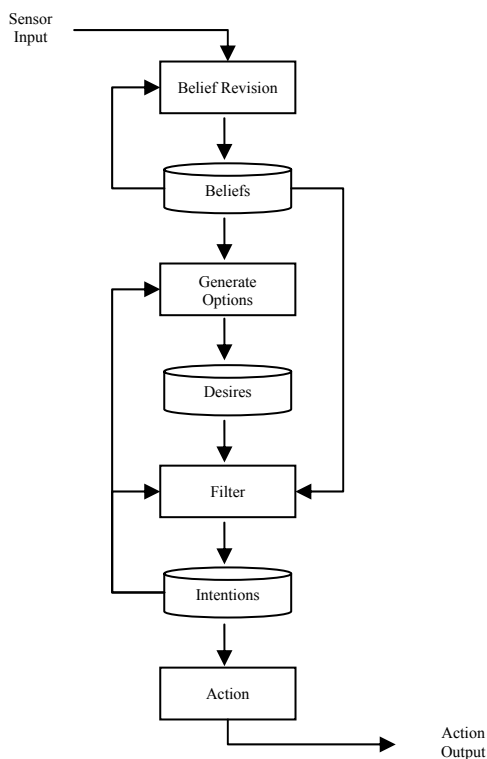
Σε περιβάλλοντα όμως πραγματικού χρόνου (real-time environment) όπου η κατάσταση τους αλλάζει δυναμικά και με μη αιτιοκρατικό τρόπο, ο χρόνος που αφιερώνουν οι πράκτορες για τον υπολογισμό των ενεργειών τους μπορεί να μειώσει το βαθμό απόκρισής τους λόγω του ότι στα χρονικά διαστήματα αυτά το περιβάλλον μπορεί ν'

αλλάζει. Οι περιορισμένοι πόροι μπορεί να καταστήσουν αδύνατη τη χρήση υπολογιστικών διαδικασιών που να περιγράφουν πλήρως και αποτελεσματικά τη συμπεριφορά των πρακτόρων [Musliner, 1999].

Για να είναι αποτελεσματικός ένας πράκτορας σ' ένα περιβάλλον πραγματικού χρόνου θα πρέπει να πετυχαίνει τους στόχους του (όχι κατ' ανάγκη με βέλτιστο τρόπο) πριν οι πόροι εξαντληθούν. Ο ρόλος των προθέσεων στους BDI πράκτορες είναι καθοριστικός στο να αποφασίζουν και να δρουν αποτελεσματικά σε τέτοιου τύπου περιβάλλοντα. Ακολουθώντας τον Bratman [Bratman, 1987] οι Cohen και Levesque [Cohen and Levesque, 1990] παρατήρησαν επτά ιδιότητες για τις προθέσεις:

1. Οι προθέσεις θέτουν νέους στόχους στους πράκτορες οι οποίοι θα πρέπει να βρουν τρόπους για να τους πετύχουν.
2. Οι προθέσεις παρέχουν ένα είδος «φίλτρου» για την αποδοχή νέων προθέσεων.
3. Οι πράκτορες εμμένουν στις προθέσεις τους. Συνεχώς ελέγχουν αν έχουν επιτευχθεί και αν όχι τότε προσπαθούν να τις πετύχουν.
4. Οι πράκτορες πάντα έχουν τη πεποίθηση ότι οι προθέσεις τους μπορούν να επιτευχθούν.
5. Οι πράκτορες δεν πιστεύουν ότι δεν έχουν την ικανότητα να πραγματοποιήσουν τις προθέσεις τους.
6. Κάτω από συγκεκριμένες καταστάσεις του περιβάλλοντός τους πιστεύουν ότι έχουν την ικανότητα να πετύχουν τις προθέσεις τους.
7. Οι πράκτορες μπορεί να μην επιθυμούν τα παράπλευρα αποτελέσματα των προθέσεών τους.

Η πρακτική συλλογιστική ενός BDI-πράκτορα παρουσιάζεται στις ακόλουθες εικόνες [Wooldridge, 1999]. Η Εικόνα 13 παρουσιάζει μία γενικευμένη αρχιτεκτονική ενός BDI-πράκτορα καθώς η Εικόνα 14 παρουσιάζει το βασικό αλγόριθμο που υλοποιεί τη πρακτική συλλογιστική ενός BDI-πράκτορα.



Εικόνα 13: Βασικός αλγόριθμος υλοποίησης BDI-πράκτορα.

Algorithm Agent Control Loop.

```

1. while true
2.   observe the world;
3.   update internal mental state;
4.   deliberate about what intention to achieve next;
5.   use means-end reasoning to get a plan for the intentions;
6.   execute the plan
7. end while
  
```

Εικόνα 14: Βασικός αλγόριθμος υλοποίησης BDI-πράκτορα.

Οι βασικές διεργασίες που υλοποιούν έναν BDI-πράκτορα είναι οι παρακάτω:

- (α) Η διαδικασία αναθεώρησης των πεποιθήσεων του πράκτορα (revise beliefs). Δεδομένου των νέων πληροφοριών που δέχεται ο πράκτορας από το φυσικό του περιβάλλον είτε δημιουργεί νέες πεποιθήσεις είτε επανεξετάζει τις ήδη υπάρχουσες αν ισχύουν.

- (β) Η διαδικασία δημιουργίας νέων πιθανών επιλογών για δράση (option generator). Η αναθεώρηση των πεποιθήσεων ή η προσπάθεια του πράκτορα να υλοποιήσει ενέργειες που ήδη έχει δεσμευθεί να πραγματοποιήσει μπορούν να οδηγήσουν σε νέες επιθυμίες.
- (γ) Η διαδικασία φιλτραρίσματος των επιλογών που έχει ο πράκτορας. Κάθε επιθυμία του πράκτορα υπόκειται σε έλεγχο του κατά πόσο είναι εφικτό να πραγματοποιηθεί. Η διαδικασία αυτή αναπαριστά το στοχασμό (deliberation) του πράκτορα για το τι πρέπει να πραγματοποιήσει. Αποτέλεσμα της νοητικής αυτής διεργασίας είναι η αλλαγή της νοητικής του κατάστασης: κάθε επιθυμία του πράκτορα που περνάει επιτυχώς τη διαδικασία αυτή μετατρέπεται σε πρόθεση.
- (δ) Η διαδικασία επιλογής ενεργειών για πραγματοποίηση. Δεδομένου ενός συνόλου προθέσεων, ο πράκτορας με τη διαδικασία αυτή επιλέγει τις ενέργειες που πρέπει να πραγματοποιήσει άμεσα και τις πραγματοποιεί.

Εφαρμογές του BDI μοντέλου παρουσιάζουμε στην ενότητα 3.3.

### 3.2. Το μοντέλο συνεργασίας *SharedPlans*

Όπως τονίσαμε στην ενότητα 1.6 ένα από τα χαρακτηριστικά των νοημόνων πρακτόρων είναι η δυνατότητα επικοινωνίας με άλλους πράκτορες ή με τον χρήστη και η συνεργασία μεταξύ τους προκειμένου να πετύχουν έναν κοινό σκοπό ή μία κοινή ενέργεια.

Το μοντέλο συνεργασίας *SharedPlans* [Grosz and Kraus, 1996] καθορίζει τις νοητικές δομές και καταστάσεις που απαιτούνται από μία ομάδα πρακτόρων για να ενεργήσουν συνεργατικά. Πιο συγκεκριμένα, δεδομένου ενός κοινού στόχου, το μοντέλο καθιστά τους πράκτορες ικανούς να συνεργάζονται για την από κοινού επίτευξη του στόχου αυτού αναπτύσσοντας αρχικά ένα *μερικώς ολοκληρωμένο κοινό πλάνο* (partial shared plan). Το πλάνο αυτό σταδιακά το μετατρέπουν σε *πλήρως ολοκληρωμένο κοινό πλάνο* (full shared plan) χωρίς όμως να είναι αναγκαίο ο κάθε πράκτορας να γνωρίζει πλήρως τις ενέργειες των άλλων πρακτόρων.

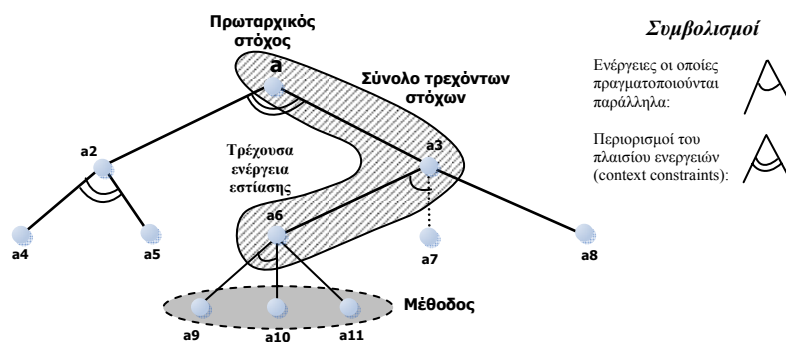
Πριν παρουσιάσουμε αναλυτικότερα το μοντέλο συνεργασίας *SharedPlans* θα αναφερθούμε πρώτα σε ορισμένους όρους και δομές που χρησιμοποιούνται σ' αυτό.



### 3.2.1. Ορισμοί

Κάθε ενέργεια που πραγματοποιεί ένας πράκτορας μπορεί να είναι είτε *βασική* (basic level action) είτε *σύνθετη* (complex level action). Μία βασική ενέργεια εκτελείται απευθείας από τον πράκτορα αλλάζοντας τη κατάσταση του φυσικού του περιβάλλοντος ή/και τη νοητική του κατάσταση. Αντίθετα, μία σύνθετη ενέργεια αναλύεται σε υποενέργειες που μπορεί να είναι και αυτές είτε σύνθετες είτε απλές.

Κάθε ένας πράκτορας έχει μία βιβλιοθήκη με μεθόδους πραγματοποίησης ενεργειών. Μια *μέθοδος πραγματοποίησης ενέργειας* (recipe of action) αποτελείται από μία ομάδα υπό-ενεργειών, ένα σύνολο περιορισμών εφαρμογής (applicability conditions) και ένα σύνολο γεγονότων που αναπαριστούν τα αποτελέσματα της ενέργειας (effects). Κάνοντας χρήση των μεθόδων αυτών για κάθε υπό-ενέργεια ο πράκτορας αναπτύσσει ιεραρχικά πλάνα όπως αυτό της Εικόνας 15. Το δένδρο αυτό ονομάζεται *δένδρο πραγματοποίησης ενέργειας* (recipe tree) ή πλαίσιο ενεργειών της ενέργειας. Κάθε κόμβος του δένδρου αυτού αναπαριστά μία σύνθετη ενέργεια ενώ κάθε φύλλο του μία βασική ενέργεια. Η τρέχουσα ενέργεια που προσπαθεί ο πράκτορας να πραγματοποιήσει ονομάζεται *τρέχουσα ενέργεια εστίασης* (focal action). Επίσης, οι σύνθετες ενέργειες που περιέχουν την τρέχουσα ενέργεια εστίασης ως υπό-ενέργεια ορίζουν το *σύνολο των τρεχόντων στόχων του πράκτορα* (the agent's attentional state). Οι στόχοι αυτοί εκφράζουν τους λόγους για τους οποίους ο πράκτορας πραγματοποιεί την τρέχουσα ενέργεια.



Εικόνα 15: Το πλαίσιο ενεργειών (context of action) μίας ενέργειας *a*.

Αν οι υπό-ενέργειες μίας σύνθετης ενέργειας πραγματοποιούνται από παραπάνω από έναν πράκτορα τότε η ενέργεια αυτή ονομάζεται *πολύ-πρακτορική* (multi-agent action) ενώ αν όλες οι υπό-ενέργειες πραγματοποιούνται από έναν και μόνο πράκτορα τότε ονομάζεται *προσωπική ενέργεια* (individual action). Αντίστοιχα διαχωρίζουμε τα πλάνα

σε πολύ-πρακτορικά και προσωπικά αναλόγως αν το πλάνο αποτελείται από προσωπικές μόνο ενέργειες ή υπάρχουν και πολύ-πρακτορικές.

Δεδομένης μίας ενέργειας  $a$ , διαχωρίζουμε την *ικανότητα* (ability) από την *δυνατότητα* (capability) ενός πράκτορα να πραγματοποιήσει την ενέργεια αυτή. Συγκεκριμένα, αν η ενέργεια  $a$  είναι βασική<sup>10</sup> λέμε ότι ένας πράκτορας έχει την ικανότητα να την πραγματοποιήσει (the agent is able to perform the action  $a$ ) αν το σώμα του πράκτορα διαθέτει τουλάχιστον έναν κατάλληλο μηχανισμό δράσης (actuator) που να υλοποιεί την ενέργεια αυτή.

Ενώ λέμε ότι ο πράκτορας έχει την *δυνατότητα να πραγματοποιήσει μία ενέργεια  $a$*  (the agent is capable to perform an action  $a$  or the agent can bring about an action  $a$ ) στις ακόλουθες περιπτώσεις:

- (α) Αν η ενέργεια  $a$  είναι βασική, τότε θα πρέπει να έχει την ικανότητα να τη πραγματοποιήσει, και να ισχύουν οι περιορισμοί εφαρμογής της μεθόδου (applicability conditions).
- (β) Αν η ενέργεια  $a$  είναι σύνθετη, τότε θα πρέπει να ισχύουν οι περιορισμοί εφαρμογής (applicability conditions) της μεθόδου της ενέργειας αυτής και για κάθε υπό-ενέργεια  $\beta_i$  της  $a$  ο πράκτορας θα πρέπει να γνωρίζει τουλάχιστον μία σχετική μέθοδο  $\gamma'$  αυτή και (αναδρομικά) να έχει τη δυνατότητα να τη πραγματοποιήσει.

Αν οι περιορισμοί εφαρμογής μίας σύνθετης ενέργειας πρέπει να διατηρούνται καθ' όλη τη διάρκεια πραγματοποίησης αυτής τότε οι περιορισμοί αυτοί ονομάζονται *περιορισμοί του πλαισίου ενεργειών* της ενέργειας αυτής (context constraints of action).

Κατά τη διάρκεια του σχεδιασμού ενός πλάνου πραγματοποίησης μίας ενέργειας (recipe tree) μπορούν να δημιουργηθούν συγκρούσεις μεταξύ των μεθόδων των ενεργειών (για παράδειγμα μεταξύ των αποτελεσμάτων της μεθόδου της μίας ενέργειας και των περιορισμών εφαρμογής της άλλης) ή/και μεταξύ των περιορισμών των ενεργειών και της τρέχουσας νοητικής κατάστασης του πράκτορα (για παράδειγμα οι πεποιθήσεις

για τη τρέχουσα κατάσταση του φυσικού περιβάλλοντός του πράκτορα μπορεί να έρχονται σε αντίφαση με τους περιορισμούς εφαρμογής μίας μεθόδου). Αναλυτική περιγραφή των συγκρούσεων που ένας πράκτορας μπορεί ν' αναγνωρίσει στο πλαίσιο ανάπτυξης πρακτόρων ICAGENT δίνουμε στην υποενότητα 4.2.3.

Τέλος διαχωρίζουμε τις βασικές ενέργειες σε ενέργειες πεδίου εφαρμογής (domain actions) και σε νοητικές ενέργειες (mental actions). Κάθε βασική ενέργεια πεδίου εφαρμογής έχει ως αποτέλεσμα ν' αλλάζει τη κατάσταση του φυσικού περιβάλλοντος του πράκτορα, ενώ κάθε νοητική ενέργεια αλλάζει τη νοητική του κατάσταση.

Οι τρεις κυριότερες νοητικές ενέργειες που χρησιμοποιούνται στο μοντέλο *SharedPlans* είναι αυτή της επιλογής μεθόδου πραγματοποίησης ενέργειας (recipe selection), αυτής της ανάπτυξης πλάνου (plan elaboration) και αυτής της αναγνώρισης και επίλυσης διενέξεων (reconciliation). Με τη πρώτη ο κάθε πράκτορας μπορεί να επιλέξει (από τη βιβλιοθήκη μεθόδων) μεθόδους πραγματοποίησης ενεργειών για ενέργειες που επιθυμεί να πραγματοποιήσει. Με τη δεύτερη μπορεί ν' αναπτύσσει περαιτέρω τις μεθόδους πραγματοποίησης ενεργειών βρίσκοντας μεθόδους για τις υπό-ενέργειές τους και τέλος, με τη τρίτη νοητική ενέργεια ελέγχει αν η μέθοδος που έχει επιλέξει είναι εφαρμόσιμη και στη περίπτωση που δεν είναι προσπαθεί να επιλύσει τις τυχόν συγκρούσεις.

Στις παρακάτω τρεις υποενότητες παρουσιάζουμε τις βασικές νοητικές δομές που χρησιμοποιούνται στο μοντέλο των *SharedPlans* καθώς και τους ορισμούς του πλήρους προσωπικού πλάνου, του μερικώς αναπτυγμένου πλάνου, του πλήρως ανεπτυγμένου κοινού πλάνου και του μερικώς ανεπτυγμένου κοινού πλάνου.

### 3.2.2. Νοητικές δομές

Για την αναπαράσταση της νοητικής κατάστασης του πράκτορα, στο μοντέλο των *SharedPlans*, χρησιμοποιούνται πέντε νοητικές δομές: η πρώτη είναι αυτή της πεποίθησης, η δεύτερη της αμοιβαίας πεποίθησης, η τρίτη της επιθυμίας, η τέταρτη της πρόθεσης πραγματοποίησης μίας ενέργειας και τέλος η πέμπτη είναι αυτή της πρόθεσης να ισχύσει μία κατάσταση.

---

<sup>10</sup> Για τις σύνθετες ενέργειες δεν ορίζεται η έννοια της ικανότητας.

Η πεποίθηση αναπαριστά τη γνώση του πράκτορα για γεγονότα που συμβαίνουν στο φυσικό του περιβάλλον (συμπεριλαμβανομένου και της εσωτερικής (νοητικής) κατάστασης των άλλων πρακτόρων). Η αμοιβαία πεποίθηση αναπαριστά τη γνώση που έχει ένας πράκτορας ότι ένα γεγονός ισχύει καθώς και το ότι οι άλλοι πράκτορες γνωρίζουν το γεγονός αυτό και το ότι οι άλλοι γνωρίζουν ότι ο πράκτορας γνωρίζει ότι γνωρίζουν... και συνεχίζοντας έτσι επ' άπειρο [Wooldridge, 2000].

Λέμε ότι ένας πράκτορας έχει την *επιθυμία* (potential intention to) να πραγματοποιήσει μία ενέργεια όταν θέλει να τη πραγματοποιήσει αλλά δεν γνωρίζει αν η μέθοδος που έχει επιλέξει είναι εφαρμόσιμη, δηλαδή δεν έχει εξετάσει ακόμα για πιθανές συγκρούσεις μεταξύ της ενέργειας αυτής και των προθέσεων του ή/ και της νοητικής του κατάστασης.

Λέμε ότι ένας πράκτορας έχει τη *πρόθεση* να πραγματοποιήσει μία ενέργεια "*intention to do an action*" αν έχει εξετάσει επιτυχώς τη μέθοδο της ενέργειας για τυχόν συγκρούσεις και:

1. Αν η ενέργεια είναι βασική τότε ο πράκτορας θα πρέπει να έχει τη δυνατότητα να πραγματοποιήσει αυτή και να έχει δεσμευθεί να κάνει αυτό.
2. Αν η ενέργεια είναι σύνθετη τότε ο πράκτορας θα πρέπει να έχει βρει είτε ένα πλήρες πλάνο γ' αυτή είτε:
  - (α) ένα μερικό πλάνο για αυτή,
  - (β) ένα πλήρες πλάνο για το πώς θα το αναπτύξει το μερικό πλάνο σε πλήρες και
  - (γ) να έχει δεσμευτεί να αναπτύξει το μερικό πλάνο. Δηλαδή, ένας πράκτορας μπορεί να έχει τη πρόθεση να πραγματοποιήσει μία ενέργεια ακόμα και αν δεν γνωρίζει πλήρως το πλάνο αυτής.

Κάθε κοινό πλάνο περιγράφεται ως συνάρτηση δύο νοητικών δομών: Αυτή της *πρόθεσης πραγματοποίησης* μιας ενέργειας "*intention to do an action*" και αυτή της *πρόθεσης ισχύος* μιας κατάστασης "*intention that a state holds*". Οι ομοιότητες των δύο αυτών νοητικών δομών είναι ότι (α) δεσμεύουν το πράκτορα να μην υιοθετήσει ενέργειες οι οποίες να έρχονται σε σύγκρουση μεταξύ τους και (β) ότι οδηγούν τον πράκτορα να επανασχεδιάσει τα πλάνα του σε περίπτωση αποτυχίας. Οι διαφορές αυτών δεν έγκειται

τόσο στο τύπο των αντικειμένων που δεσμεύονται να πραγματοποιήσουν ή να πετύχουν αντίστοιχα αλλά στο ότι η πρώτη νοητική δομή οδηγεί τον πράκτορα πάντα σε ανάλυση μέσων και σκοπών (προκειμένου να πραγματοποιήσει την ενέργεια), ενώ η δεύτερη νοητική δομή αποτρέπει τον πράκτορα να εκτελέσει ενέργειες που έρχονται σε αντίφαση με ενέργειες των συνεργατών του καθώς επίσης τον οδηγεί να πραγματοποιήσει ενέργειες που βοηθούν την επίτευξη των στόχων των συνεργατών του (όποτε αυτό είναι αναγκαίο).

Η νοητική δομή *intent to* οδηγεί πάντα τον πράκτορα σε ανάλυση μέσων και σκοπών προκειμένου να σχεδιάσει τον τρόπο με τον οποίο θα πραγματοποιήσει την ενέργεια που έχει δεσμευθεί να κάνει, καθώς επίσης και να τη πραγματοποιήσει. Αντίθετα, η νοητική δομή *intention that* δεν παρακινεί τον πράκτορα άμεσα σε μια τέτοια συμπεριφορά, αλλά παρέχει στο πράκτορα τη βάση να συνδυάζει (mesh) και να συντονίζει (coordinate) τα πλάνα του με αυτά των άλλων συνεργατών του παρέχοντας βοήθεια σε αυτούς όποτε χρειάζεται. Για αυτό το λόγο, οι προθέσεις της μορφής *intention that* μπορούν να οδηγήσουν τους πράκτορες σε υιοθέτηση προθέσεων της μορφής *intent to* (και περαιτέρω σε ανάλυση μέσων και σκοπών) προκειμένου να πετύχουν ή να διατηρήσουν τη κατάσταση.

Οι πιο σημαντικές λειτουργίες της νοητικής δομής *intention that* στο μοντέλο των *SharedPlans* είναι ο ρόλος της στη λήψη αποφάσεων του πράκτορα κατά την ανάπτυξη των πλάνων του καθώς και στη λήψη πρωτοβουλιών είτε για να βοηθήσει κάποιον άλλο πράκτορα είτε για να καταστήσει τον εαυτό του ικανό στο να πετύχει ένα στόχο<sup>11</sup> (προσωπικό ή της ομάδας) [Grosz and Kraus, 1999]. Πιο συγκεκριμένα η δομή *intention that* μπορεί να χρησιμοποιηθεί για τους ακόλουθους λόγους:

- *Δέσμευση ομάδας για κοινή δράση* (Group activity commitment): Η πρόθεση *intention that* μπορεί να χρησιμοποιηθεί για να αποτρέψει τους πράκτορες μίας ομάδας στο να δεσμευθούν να πραγματοποιήσουν ενέργειες που μπορεί να έρθουν σε σύγκρουση με το κοινό στόχο της ομάδας. Επίσης, μπορεί να οδηγήσει τους πράκτορες στη κατασκευή κοινών πλάνων (για να πετύχουν τους κοινούς στόχους τους) και

να τους κατευθύνει στο να ενεργούν με τέτοιο τρόπο ώστε ο ένας να βοηθά τον άλλον είτε εκτελώντας ενέργειες για λογαριασμό της ομάδας είτε πραγματοποιώντας ενέργειες οι οποίες καθιστούν τους συνεργάτες ικανούς να πραγματοποιήσουν τις ενέργειες που έχουν δεσμευθεί να κάνουν.

- *Καθορισμός παραμέτρων (Parameter identification)*: Η νοητική δομή *intention that* μπορεί να οδηγήσει τους πράκτορες στη πραγματοποίηση ενεργειών οι οποίες καθορίζουν παραμέτρους των ενεργειών που επιθυμούν να πραγματοποιήσουν. Για παράδειγμα, μπορεί να καθορίζονται παράμετροι που αφορούν τις υπό-ενέργειες της μεθόδου ή τα αποτελέσματα (effects) αυτής ή παραμέτρους των περιορισμών εφαρμογής αυτής.
- *Ικανοποίηση περιορισμών (Constraints satisfaction)*: Η νοητική δομή *intention that* οδηγεί τους συνεργαζόμενους πράκτορες να εξετάσουν την ικανότητα εκτέλεσης βασικών ενεργειών (basic level actions), τους περιορισμούς εφαρμογής των μεθόδων σύνθετων ενεργειών (applicability conditions) και τους περιορισμούς που πρέπει να διατηρούνται κατά τη διάρκεια πραγματοποίησης της σύνθετης ενέργειας (context constraints).
- *Βασική πρόθεση (Core case)*: Ο κύριος λόγος ύπαρξης της νοητικής δομής *intention that* είναι να περιορίζει τις επιλογές των πρακτόρων έτσι ώστε να μπορούν να χειρίζονται κοινούς πόρους χωρίς οι ενέργειες τους να έρχονται σε σύγκρουση με τις ανάγκες των συνεργατών τους. Όπως τονίζουμε στις επόμενες παραγράφους οι βασικές προθέσεις (core case commitments) έχουν ως στόχο να καθιστούν τους πράκτορες ικανούς να συντονίζουν τα προσωπικά τους πλάνα με τα κοινά πλάνα της ομάδας.
- *Ικανοποίηση κοινών πεποιθήσεων (Mutual belief satisfaction)*: Η νοητική δομή *intention that* προτρέπει επίσης τους πράκτορες να επικοινωνήσουν τις ικανότητές τους, τις δυνατότητές τους και γενικότερα τη γνώση που έχουν για τις μεθόδους που έχουν επιλέξει για τις κοινές ενέργειες τους προκειμένου να δημιουργήσουν τις κοινές πεποιθήσεις που απαιτούνται για την ανάπτυξη των κοινών πλάνων.

---

<sup>11</sup> Ως στόχο εδώ θεωρούμε τη κατάσταση *State* της πρόθεσης "*intention that a state holds*".

### 3.2.3. Προσωπικά πλάνα

Ο ορισμός του πλήρους ανεπτυγμένου προσωπικού πλάνου καθορίζει τις συνθήκες εκείνες που πρέπει να ισχύουν για να έχει ένας πράκτορας  $G$  ένα πλάνο  $P$  για μια ενέργεια  $a$  χρησιμοποιώντας μία μέθοδο  $R_a$  υπό τις συνθήκες που θέτουν οι περιορισμοί  $C_a$  της μεθόδου  $R_a$ .

Ένας πράκτορας έχει ένα πλήρες προσωπικό πλάνο για μία ενέργεια  $a$  αν:

- (1) γνωρίζει μία μέθοδο  $R_a$  πραγματοποίησης της  $a$  και
- (2) για κάθε υπό-ενέργεια  $\beta_i$  της μεθόδου  $R_a$  έχει δεσμευθεί να τη πραγματοποιήσει υπό τους περιορισμούς  $C_a$  της μεθόδου  $R_a$ . Αν η ενέργεια  $a$  είναι σύνθετη τότε θα πρέπει να γνωρίζει ένα πλήρες πλάνο  $\gamma'$  αυτής.

Ένας πράκτορας έχει ένα μερικώς ανεπτυγμένο προσωπικό πλάνο για μία ενέργεια  $a$  όταν γνωρίζει ότι υπάρχει μία μέθοδος πραγματοποίησης  $R_a$  αλλά η μέθοδος αυτή δεν είναι πλήρως γνωστή (δηλαδή είναι γνωστές μερικές ή καμία από τις ενέργειες που την αποτελούν). Σε αυτή τη περίπτωση θα πρέπει να έχει τη πρόθεση ν' αναπτύξει τη μέθοδο αυτή και να έχει βρει ένα πλήρες πλάνο για τη νοητική ενέργεια αυτή (δηλαδή της ανάπτυξης πλάνου). Επίσης για κάθε μία ενέργεια  $\beta_i$  της μεθόδου  $R_a$  θα πρέπει να ισχύει μία από τις παρακάτω περιπτώσεις:

- (1) Είτε ο πράκτορας θα πρέπει να έχει τη πρόθεση να πραγματοποιήσει την ενέργεια  $\beta_i$  μόνος του γνωρίζοντας ένα (ίσως μερικώς ανεπτυγμένο) πλάνο  $\gamma'$  αυτής.
- (2) Είτε να μην έχει ολοκληρώσει την διεργασία ελέγχου συγκρούσεων μεταξύ της μεθόδου της  $\beta_i$  με τις μεθόδους άλλων ενεργειών που έχει δεσμευθεί να πραγματοποιήσει, καθώς και μεταξύ των περιορισμών εφαρμογής της μεθόδου της ενέργειας με τη τρέχουσα νοητική του κατάσταση.

### 3.2.4. Κοινά πλάνα

Ο ορισμός ενός κοινού πλάνου (shared plan) καθορίζει τις συνθήκες εκείνες που πρέπει να ισχύουν για να έχει από μία ομάδα πρακτόρων  $GR$  ένα κοινό πλάνο  $P$  για μια ενέργεια  $a$  υπό το σύνολο περιορισμών  $C_a$ .

Μία ομάδα πρακτόρων  $GR$  έχουν ένα κοινό πλάνο  $P$  για μία ενέργεια  $a$  αν:

- (1) η ομάδα έχει αναπτύξει ένα πλήρες πλάνο για την  $a$  ή
- (2) αν η ομάδα έχει βρει ένα μερικό πλάνο για την  $a$  και ένα πλήρες πλάνο για το πώς θα το αναπτύξει σε πλήρη μορφή.

Μία βασική αρχή των κοινών πλάνων είναι ότι κανένας από τους πράκτορες δεν είναι αναγκαίο να γνωρίζει πλήρως το κοινό πλάνο. Ακόμα και αν το κοινό πλάνο έχει συμπληρωθεί πλήρως, μπορεί κανένας από τους πράκτορες να μην γνωρίζει όλες τις βασικές ενέργειες του πλάνου. Ο στόχος του κοινού πλάνου είναι να ορίσει την ελάχιστη γνώση που πρέπει να μοιράζονται μεταξύ τους οι πράκτορες για να πραγματοποιήσουν από κοινού μία ενέργεια.

#### 3.2.4.1. Πλήρως αναπτυγμένα κοινά πλάνα

Μία ομάδα πρακτόρων  $GR$  έχει ένα πλήρως ανεπτυγμένο πλάνο (full shared plan)  $P$  για μία ενέργεια  $a$  υπό τους περιορισμούς που θέτουν οι συνθήκες  $C_a$  αν:

- (1) η ομάδα πιστεύει ότι ο κάθε πράκτορας έχει τη πρόθεση να πετύχει η ομάδα τον στόχο της (δηλαδή να πραγματοποιήσουν από κοινού την ενέργεια  $a$ ),
- (2) ο κάθε πράκτορας της ομάδας γνωρίζει μία μέθοδο πραγματοποίησης  $R_a$  για την ενέργεια  $a$  και αμοιβαία πιστεύει ότι και οι άλλοι συνεργάτες γνωρίζουν την ίδια.
- (3) Για κάθε υπο-ενέργεια  $\beta_i$  της μεθόδου  $R_a$  ισχύουν ένα από τα παρακάτω: είτε υπάρχει ένας πράκτορας που έχει δεσμευθεί ότι θα τη πραγματοποιήσει (αν η  $\beta_i$  είναι προσωπική ενέργεια), είτε υπάρχει μία υπο-ομάδα της αρχικής ομάδας  $GR$  η οποία έχει δεσμευθεί ότι θα πραγματοποιήσει την ενέργεια  $\beta_i$  (στη περίπτωση που η  $\beta_i$  είναι πολύ-πρακτορική ενέργεια).

Λόγω της αμοιβαίας πεποίθησης ότι όλοι οι πράκτορες της ομάδας έχουν δεσμευθεί να πετύχουν τον κοινό στόχο (της συνθήκης (1)) προκύπτουν δύο σημαντικά χαρακτηριστικά των κοινών πλάνων: το πρώτο είναι ότι αποτρέπει τους πράκτορες να υιοθετήσουν μελλοντικές προθέσεις που να έρχονται σε αντίφαση με το κοινό σκοπό της ομάδας και κατά δεύτερον προτρέπει τους πράκτορες να επικοινωνήσουν οποτεδήποτε υπάρχει πρόβλημα είτε κατά το σχεδιασμό είτε κατά την εκτέλεση των πλάνων τους.



Για κάθε υπό-ενέργεια  $\beta_i$  της μεθόδου  $R_a$  η οποία πραγματοποιείται από έναν και μόνο πράκτορα  $G_k$  θα πρέπει:

- (1) ο πράκτορας  $G_k$  να έχει τη πρόθεση να πραγματοποιήσει την υπο-ενέργεια αυτή και η ομάδα  $GR$  να πιστεύει ότι ο  $G_k$  έχει τη πρόθεση αυτή (δηλαδή ότι ο πράκτορας  $G_k$  έχει τη πρόθεση να τη πραγματοποιήσει),
- (2) αν η υπο-ενέργεια  $\beta_i$  είναι βασική τότε η ομάδα  $GR$  θα πρέπει να πιστεύει ότι ο πράκτορας  $G_k$  έχει την ικανότητα να την αναλάβει υπό την ένωση των περιορισμών που θέτουν οι ενέργειες  $a$  και  $\beta_i$ .
- (3) αν η υπο-ενέργεια  $\beta_i$  δεν είναι βασική τότε ο  $G_k$  θα πρέπει να έχει ένα πλήρες πλάνο για την υπο-ενέργεια αυτή και η ομάδα θα πρέπει να πιστεύει ότι ο πράκτορας είναι ικανός να τη πράξει υπό τους περιορισμούς που θέτουν οι ενέργειες  $a$  και  $\beta_i$ .
- (4) η ομάδα να πιστεύει ότι ο κάθε πράκτορας έχει δεσμευθεί στο να έχει ο πράκτορας  $G_k$  τη δυνατότητα να πραγματοποιήσει την ενέργεια (δηλαδή, ο κάθε πράκτορας δεσμεύεται να .

Για κάθε υπό-ενέργεια  $\beta_i$  της μεθόδου  $R_a$  η οποία πραγματοποιείται από μία υπό-ομάδα  $GR_k$  θα πρέπει:

- (1) να υπάρχει μία μέθοδος  $R_{\beta_i}$  για την υπο-ενέργεια  $\beta_i$  τέτοια ώστε να υπάρχει ένα πλήρες πλάνο για την ενέργεια αυτή,
- (2) η ομάδα  $GR$  να πιστεύει ότι υπάρχει η μέθοδος αυτή,
- (3) η ομάδα να έχει την ικανότητα να πραγματοποιήσει την υπο-ενέργεια  $\beta_i$  χρησιμοποιώντας τη μέθοδο  $R_{\beta_i}$  υπό τους περιορισμούς των ενεργειών  $a$  και  $\beta_i$ .
- (4) η ομάδα  $GR$  να πιστεύει ότι ο κάθε πράκτορας της υπό-ομάδας  $GR_k$  έχει δεσμευθεί ότι η υπό-ομάδα είναι ικανή να πραγματοποιήσει την ενέργεια αυτή υπό τους περιορισμούς που θέτουν οι ενέργειες  $a$  και  $\beta_i$ .

### 3.2.4.2. Μερικώς αναπτυγμένα κοινά πλάνα

Μία ομάδα πρακτόρων  $G_R$  έχει ένα μερικώς ανεπτυγμένο πλάνο (partial shared plan)  $P$  για μία ενέργεια  $a$  υπό τους περιορισμούς  $C_a$  αν:

- (1) η ομάδα  $G_R$  πιστεύει ότι όλα τα μέλη της έχουν δεσμευτεί στο να πετύχει η ομάδα το σκοπό της, δηλαδή ότι θα πραγματοποιήσει την ενέργεια  $a$  υπό τους περιορισμούς  $C_a$ .
- (2) η ομάδα  $G_R$  πιστεύει ότι υπάρχει μία μέθοδος  $R_a$  πραγματοποίησης της ενέργειας  $a$  η οποία είναι μερικώς ολοκληρωμένη (δηλαδή οι ενέργειες που την αποτελούν δεν έχουν καθοριστεί πλήρως) αλλά η ομάδα έχει ένα πλήρες κοινό πλάνο για το πώς θα την ολοκληρώσει.

Για κάθε υπό-ενέργεια  $\beta_i$  της κοινής μερικώς ολοκληρωμένης μεθόδου  $R_a$  ισχύουν ένα από τα παρακάτω:

- (α) είτε υπάρχει ένας πράκτορας  $G_k$  που έχει τη πρόθεση να πραγματοποιήσει την υπό-ενέργεια αυτή αλλά έχει βρει ένα μερικώς ανεπτυγμένο πλάνο (single-agent subaction),
- (β) είτε υπάρχει μία υπό-ομάδα  $G_{R_k}$  πρακτόρων που έχει ένα κοινό πλάνο για τη πραγματοποίηση αυτής αλλά δεν είναι ακόμα πλήρες (multi-agent subaction),
- (γ) είτε η ομάδα δεν έχει ακόμα αποφασίσει (στοχαστεί) για το πώς θα πραγματοποιηθεί (unreconciled case).

Στις περιπτώσεις (α) και (β) η ομάδα  $G_R$  δεσμεύεται ότι ο πράκτορας  $G_k$  ή η υπό-ομάδα πρακτόρων  $G_{R_k}$  αντίστοιχα θα μπορέσουν να πραγματοποιήσουν την υπό-ενέργεια  $\beta_i$ . Όπως έχουμε αναφέρει παραπάνω τέτοιου είδους δεσμεύσεις έχουν ως αποτέλεσμα οι πράκτορες να βοηθούν τους συνεργάτες τους είτε έμμεσα είτε άμεσα. Τους υποχρεώνει να μην ενεργούν ενάντια στους σκοπούς των συνεργατών τους και να επικοινωνούν μεταξύ τους όποτε αυτό είναι αναγκαίο με στόχο να βοηθήσουν στη πραγματοποίηση των ενεργειών των συνεργατών τους.

Όπως τονίσαμε στην αρχή του υποκεφαλαίου το μοντέλο των *SharedPlans* περιγράφει τις νοητικές καταστάσεις των πρακτόρων κατά τη διάρκεια της συνεργασίας. Επιπλέον, για να μπορούν δύο ή περισσότεροι πράκτορες να συνεργάζονται αναπτύσσοντας κοινά πλάνα, θα πρέπει ο κάθε πράκτορας να διαθέτει τις κατάλληλες νοητικές

ενέργειες και μηχανισμούς με τους οποίους να μπορεί να αναγνωρίζει την ανάγκη για συνεργασία, να βρίσκει τους κατάλληλους συνεργάτες συμφωνώντας για τους κοινούς στόχους της ομάδας, να διαπραγματεύεται τη μέθοδο με την οποία θα επιτευχθεί ο κοινός στόχος, να διαπραγματεύεται το καταμερισμό των ενεργειών που υλοποιούν το στόχο στους συνεργάτες του και να υλοποιεί τις δεσμεύσεις του. Οι νοητικές δομές και οι νοητικές ενέργειες που υλοποιούν το μοντέλο συνεργασίας SharedPlans υποστηρίζονται από το πλαίσιο εργασίας ICAGENT και παρουσιάζονται στα κεφάλαια 3 και 4.

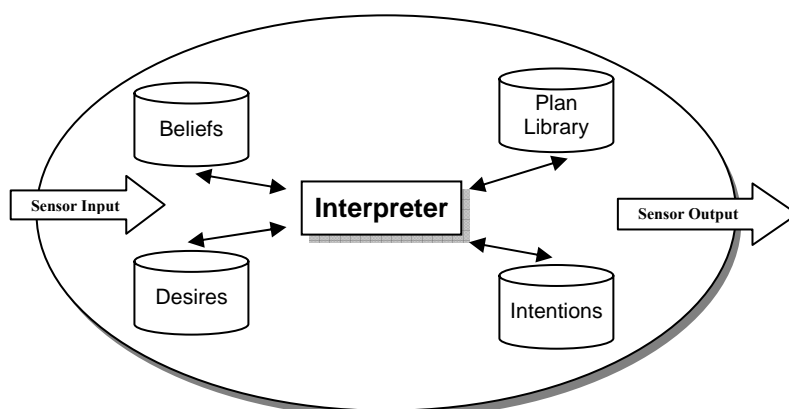
### **3.3. Πλαίσια ανάπτυξης έξυπνων πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα**

Παρ' όλο το μεγάλο πλήθος των πλαισίων εργασίας που έχουν προταθεί για ανάπτυξη πρακτόρων και πολύ-πρακτορικών συστημάτων [Luck et. al., 2003; Mangina, 2002] σ' αυτό το κεφάλαιο θ' αναφερθούμε στα κυριότερα πιο αντιπροσωπευτικά<sup>12</sup> πλαίσια εργασίας που καθιστούν ικανούς τους πράκτορες να δρουν αποτελεσματικά σε περιβάλλοντα δυναμικά και απρόβλεπτα ενεργώντας είτε μόνοι τους είτε σε συνεργασία με άλλους πράκτορες. Πιο συγκεκριμένα θ' αναφερθούμε στις αρχιτεκτονικές των: dMARS [d'Inverno, et. al., 1997; d'Inverno, et. al., 2004], InteRRaP [Müller, 1996], STEAM [Tambe, 1997], HAC [Atkin et. al., 1999] και CCAF [Weiss, 2001].

#### **3.3.1. dMARS**

Η αρχιτεκτονική του dMARS βασίζεται στο διαδικασιακό συλλογιστικό σύστημα PRS (Procedural Reasoning System) των Georgeff και Lansky [Georgeff and Lansky, 1987]. Το PRS είναι μία BDI αρχιτεκτονική, όπως αυτή που παρουσιάσαμε στην ενότητα 3.1, με τη μόνη διαφορά ότι κάνει χρήση μίας βιβλιοθήκης με έτοιμα πλάνα. Πιο συγκεκριμένα η BDI αρχιτεκτονική αποτελείται από τέσσερις δομές δεδομένων (βλ. *Εικόνα 16*): τις πεποιθήσεις του πράκτορα, τις επιθυμίες του, τις προθέσεις και μία βάση με τα πλάνα του. Οι δομές αυτές καθορίζουν σε κάθε χρονική στιγμή την εσωτερική κατάσταση του πράκτορα.

Οι πεποιθήσεις του πράκτορα αντιστοιχούν στη γνώση που έχει για τα γεγονότα που συμβαίνουν στο περιβάλλον του και αναπαριστώνται με χρήση συμβόλων [Georgeff and Lansky, 1987]. Οι επιθυμίες του πράκτορα αντιστοιχούν σε ενέργειες που θέλει να πραγματοποιήσει αλλά δεν έχει ελέγξει ακόμα αν μπορεί. Οι προθέσεις του πράκτορα αντιστοιχούν στο υποσύνολο εκείνων των επιθυμιών που πιστεύει ότι μπορεί να τις πραγματοποιήσει. Οι προθέσεις αναπαριστώνται με πλάνα που ο πράκτορας έχει επιλέξει από τη βιβλιοθήκη πλάνων.



Εικόνα 16: Η BDI αρχιτεκτονική του dMARS βασισμένη στο PRS.

Η διαφορά του PRS και του dMARS από το γενικευμένο μοντέλο BDI που παρουσιάσαμε στην ενότητα 3.1 είναι η χρήση μίας βιβλιοθήκης με έτοιμα πλάνα. Κάθε πλάνο αναπαριστά τη διαδικαστική γνώση του πράκτορα για το πώς μπορεί να πετύχει μία επιθυμία. Τα πλάνα αυτά είναι πλήρη, δηλαδή ο πράκτορας δεν τα σχεδιάζει αλλά τα παίρνει έτοιμα από τη βιβλιοθήκη.

Κάθε πλάνο αποτελείται από μία συνθήκη ενεργοποίησης (triggering condition), μία συνθήκη εφαρμογής (pre-conditions), μία συνθήκη διατήρησης της ισχύος του πλάνου (maintenance condition) και το σώμα (body) του πλάνου. Η συνθήκη ενεργοποίησης καθορίζει το πότε (δηλαδή σε ποιες καταστάσεις του περιβάλλοντος) ένα πλάνο θεωρείται κατάλληλο για τη πραγματοποίηση μίας προθέσεως. Η συνθήκη εφαρμογής

<sup>12</sup> Οι αρχιτεκτονικές των περισσότερων πλαισίων εργασίας ανάπτυξης νοημόνων πρακτόρων βασίζονται στις ιδέες των συστημάτων αυτών.

καθορίζει το πότε ο πράκτορας μπορεί να πραγματοποιήσει το πλάνο, ενώ η συνθήκη διατήρησης της ισχύος του πλάνου καθορίζει τις συνθήκες εκείνες που πρέπει να ισχύουν συνεχώς κατά τη διάρκεια εκτέλεσης του πλάνου. Τέλος, το σώμα του πλάνου αναπαριστά το σύνολο ενεργειών που πρέπει να πραγματοποιήσει ο πράκτορας για να υλοποιηθεί το πλάνο.

Κάθε πράκτορας υλοποιημένος στο dMARS παρακολουθεί συνεχώς τόσο τα γεγονότα που συμβαίνουν στο περιβάλλον του όσο και την εσωτερική του κατάσταση. Κάθε καινούργιο γεγονός που αντιλαμβάνεται το τοποθετεί σε μία ουρά (event queue). Ο διερμηνέας της Εικόνα 16 είναι υπεύθυνος να διαχειρίζεται τη συνολική συμπεριφορά του πράκτορα εκτελώντας κυκλικά τα παρακάτω βήματα:

1. Ο πράκτορας παρατηρεί το περιβάλλον του (τόσο το φυσικό του περιβάλλον όσο και την εσωτερική του κατάσταση) και αναθεωρεί την ουρά γεγονότων τοποθετώντας σ' αυτή τα πιο πρόσφατα γεγονότα.
2. Δημιουργεί νέες επιθυμίες βρίσκοντας πλάνα που η συνθήκη ενεργοποίησής τους αληθεύει από τα γεγονότα της ουράς.
3. Από το σύνολο των ενεργοποιημένων πλάνων επιλέγει ένα για να το πραγματοποιήσει. Η επιλογή αυτή εξαρτάται από το αν έχει τη δυνατότητα να το πραγματοποιήσει δηλαδή από το αν ισχύει η συνθήκη εφαρμογής ή όχι.
4. Το πλάνο αυτό που επέλεξε το τοποθετεί στο πάνω μέρος μιας νέας στοίβας ή σε μία ήδη υπάρχουσα αναλόγως αν το πλάνο είναι υλοποίηση ενός καινούργιου στόχου ή ενός ήδη υπάρχοντος αντίστοιχα.
5. Από τις υπάρχουσες στοίβες (δηλαδή από τα υπάρχοντα πλάνα που έχει δεσμευθεί να πραγματοποιήσει) επιλέγει μία και εκτελεί τη πρόθεση (δηλαδή το πλάνο) που βρίσκεται στο πιο πάνω μέρος της στοίβας. Αν η πρόθεση αφορά ενέργεια τότε ο πράκτορας την εκτελεί, ενώ αν αφορά υπο-στόχο τότε τον τοποθετεί στην ουρά γεγονότων.

Με αυτό τον τρόπο κατά τη διάρκεια εκτέλεσης ενός πλάνου τοποθετεί στην ουρά γεγονότων όλους τους νέους υπο-στόχους του και αυτοί με τη σειρά τους δημιουργούν νέα πλάνα τα οποία τοποθετούνται στις στοίβες ως νέες προθέσεις του πράκτορα.

Το μειονέκτημα της αρχιτεκτονικής του dMARS είναι ότι δεν είναι εφικτό να διαχωριστεί με καθαρό τρόπο το πότε ένας πράκτορας δρα στοχαστικά και πότε αντανακλαστικά κρατώντας ισορροπία μεταξύ αυτών των δύο συμπεριφορών. Η συμπεριφορά του πράκτορα είναι πάντοτε προκαθορισμένη και εξαρτάται από την αντιστοίχιση<sup>13</sup> των (σχετικών) μεθόδων που μπορεί να επιλέξει ο πράκτορας σε συγκεκριμένες καταστάσεις του περιβάλλοντός του. Η αρχιτεκτονική του dMARS δεν προϋποθέτει κάποιο μηχανισμό ελέγχου ο οποίος να προσδιορίζει το πότε θα πρέπει να ελέγχονται οι συνθήκες ισχύος των μεθόδων. Έτσι δεδομένου μίας σχετικής μεθόδου που έχει επιλέξει ο πράκτορας δεν υπάρχει τρόπος προσδιορισμού της συμπεριφοράς του ανάλογα με τις συνθήκες του περιβάλλοντός του.

Επίσης καμία αναφορά δεν γίνεται για το πώς δύο ή περισσότεροι πράκτορες μπορούν να συνεργαστούν σε ένα πολύ-πρακτορικό περιβάλλον. Οι δομές δεδομένων που χρησιμοποιούνται αφορούν μόνο τα εσωτερικά πλάνα των πρακτόρων και όχι τα κοινά πλάνα ή τους κοινούς στόχους που μπορεί να έχουν.

### 3.3.2. InteRRaP

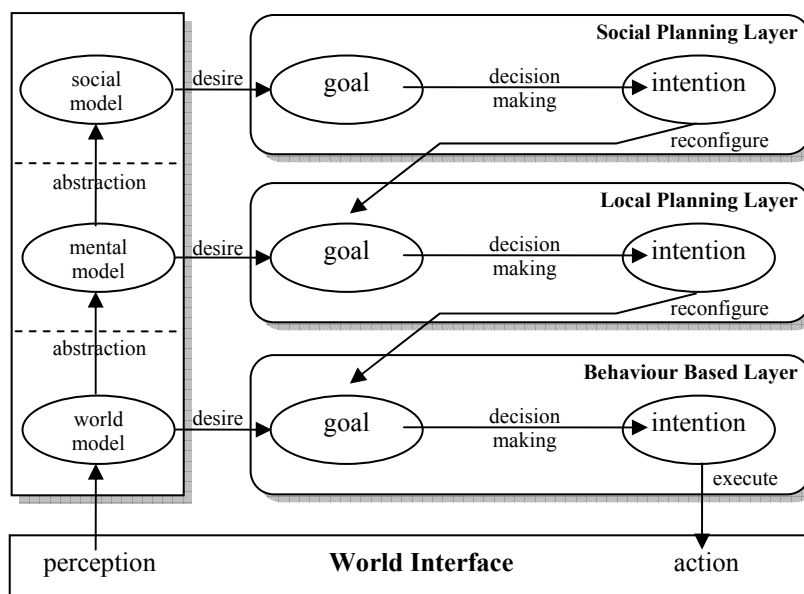
Το πλαίσιο εργασίας InteRRaP αναπτύχθηκε από τους Jorg Müller και Klaus Fischer [Müller, 1996; Fischer, 1995; Müller, 1993] και έχει εφαρμοστεί σε χώρους αυτόματης φόρτωσης (automated loading dock) [Müller, 1994] καθώς και στο RoboCup-98 [Jung, 1999]. Το InteRRaP βασίζεται σε μία υβριδική αρχιτεκτονική τριών επιπέδων κάθετου τύπου (Εικόνα 17).

Η αρχιτεκτονική του InteRRaP μοντελοποιεί μία ομαλή σειριακή μετάβαση του ελέγχου των ενεργειών του πράκτορα από το επίπεδο χειρισμού των πλάνων με αντανακλαστικό τρόπο (behaviour based layer) στο επίπεδο του σχεδιασμού προσωπικών πλάνων (local planning layer) και μετά στο επίπεδο σχεδιασμού πλάνων σε συνεργασία με άλλους πράκτορες (cooperative planning layer).

---

<sup>13</sup> Η αντιστοίχιση αυτή πραγματοποιείται από τον προγραμματιστή κατά τη διάρκεια σχεδιασμού των μεθόδων.

Το κάθε επίπεδο επικοινωνεί με μία ξεχωριστή βάση γνώσης: η πρώτη (αρχίζοντας από κάτω προς τα πάνω) παρέχει πληροφορίες για τη κατάσταση του φυσικού κόσμου, η δεύτερη παρέχει πληροφορίες για την εσωτερική νοητική κατάσταση του πράκτορα ενώ η τρίτη παρέχει πληροφορίες για τη κατάσταση της κοινωνίας των πρακτόρων.



Εικόνα 17: Η BDI αρχιτεκτονική του InteRRaP [Jung and Fischer, 1999].

Κάθε ένα επίπεδο ελέγχου των ενεργειών του πράκτορα περιλαμβάνει δύο διαδικασίες: η πρώτη δημιουργεί καινούργιους στόχους (επιθυμίες) προς επίτευξη και η δεύτερη βρίσκει πλάνα (τα οποία αναπαριστούν τις προθέσεις του πράκτορα) για να υλοποιήσει τους στόχους αυτούς.

Το πρώτο επίπεδο χρησιμοποιώντας τους μηχανισμούς αντίληψης (perception module) του πράκτορα αναθεωρεί τη γνώση που έχει για το κόσμο του και δημιουργεί επιθυμίες οι οποίες εκτελούνται άμεσα με χρήση έτοιμων πλάνων (διαδικασιών χαμηλού επιπέδου). Με αυτό τον τρόπο ο πράκτορας αντιδρά άμεσα στα ερεθίσματα του περιβάλλοντος χωρίς να σχεδιάζει πλάνα για τις ενέργειες που επιθυμεί να εκτελέσει.

Στη περίπτωση που μία επιθυμία δεν μπορεί (ή δεν είναι θεμιτό) να υλοποιηθεί άμεσα από το πρώτο επίπεδο τότε ωθείται στο αμέσως ανώτερο επίπεδο, δηλαδή σ' αυτό του σχεδιασμού προσωπικών πλάνων. Στο επίπεδο αυτό ο πράκτορας για κάθε στόχο που θέλει να πραγματοποιήσει σχεδιάζει ένα πλάνο (planning). Οι ενέργειες του πλάνου

αυτού είτε πραγματοποιούνται από το κατώτερο επίπεδο (όποτε αυτό είναι εφικτό) ή ωθούνται στο ανώτερο επίπεδο προκειμένου να πραγματοποιηθούν σε συνεργασία με άλλους πράκτορες.

Ο διαχωρισμός των τρόπων συμπεριφοράς ενός πράκτορα σε τρία διακριτά επίπεδα (δηλαδή σε αυτό της αντανακλαστικής, του σχεδιασμού πλάνων και της συνεργατικής συμπεριφοράς) καθώς και η σειριακή εναλλαγή τους κατά το σχεδιασμό και την υλοποίηση των πλάνων του πράκτορα έχει σαν αποτέλεσμα τη μη αποτελεσματική προσαρμογή της συμπεριφοράς του πράκτορα στις αλλαγές του περιβάλλοντος [Weiss, 2001]. Κάθε φορά που ο πράκτορας επιλέγει ένα τρόπο συμπεριφοράς για μία ενέργεια αυτός παραμένει σταθερός καθ' όλη τη διάρκεια εκτέλεσης του πλάνου της ενέργειας αυτής. Επίσης λόγω της σειριακής εναλλαγής των επιπέδων (για κάθε ενέργεια ο πράκτορας πρώτα προσπαθεί να την υλοποιήσει στο πρώτο επίπεδο αν δεν μπορέσει προσπαθεί στο επόμενο κτλ.) ο πράκτορας αναγκάζεται να κάνει περιττούς ελέγχους κάνοντας αναποτελεσματική τη συμπεριφορά του όταν πρέπει να πάρει άμεσα μία απόφαση. Τέλος, προγραμματιστικά απαιτείται αρκετός κώδικας διότι κάθε ενέργεια στην οποία αντιστοιχεί παραπάνω από μία συμπεριφορά θα πρέπει να υλοποιηθεί με διαφορετικούς τρόπους (πλάνα) που ο κάθε ένας ν' αντιστοιχεί στη κατάλληλη συμπεριφορά.

### 3.3.3. STEAM

Το STEAM [Tambe, 1997] είναι ένα γενικευμένο μοντέλο ομαδικής εργασίας (teamwork) το οποίο μπορεί να χρησιμοποιηθεί από υπάρχουσες αρχιτεκτονικές, όπως αυτές των Soar και PRS, για ανάπτυξη πρακτόρων ικανών να συνεργάζονται μεταξύ τους. Το μοντέλο αυτό βασίζεται στο μοντέλο συνεργασίας *συντονισμένων προθέσεων* (joint intentions theory) των P. Cohen και C. Levesque (1990) ενώ παράλληλα δανείζεται ιδέες και έννοιες από το μοντέλο συνεργασίας *SharedPlans* της B. Grosz και S. Kraus (1996).

Το μοντέλο συνεργασίας των *συντονισμένων προθέσεων* έχει χρησιμοποιηθεί ως το βασικό δομικό στοιχείο του STEAM το οποίο επιτρέπει στους πράκτορες να συντονίζουν τις ενέργειές τους με αυτές της ομάδας, ν' εντοπίζουν και να αποκαθιστούν τυχόν προβλήματα που ανακύπτουν κατά τη διάρκεια της συνεργασίας τους και να επικοινωνούν αποτελεσματικά μεταξύ τους.



Το STEAM έχει υλοποιηθεί χρησιμοποιώντας το Soar κάνοντας χρήση τριακοσίων ανεξάρτητων του πεδίου εφαρμογής κανόνων του (domain-independed rules). Έχει χρησιμοποιηθεί σε ένα αρκετά ευρύ σύνολο εφαρμογών από πολεμικές αεροπορικές αποστολές [Hill et. al., 1997] μέχρι ανάπτυξη ποδοσφαιρικών ομάδων για το RoboSoccer [Kitano et. al., 1997] και από υποστήριξη ανθρωπίνων οργανισμών [Pynadath, 2003] μέχρι σε προσομοιώσεις διάσωσης [Scerri et. al., 2003]. Στις παρακάτω δύο υπό-παραγράφους δίνουμε δύο σύντομες αναφορές για το Soar καθώς και για το μοντέλο συνεργασίας *συντονισμένων προθέσεων*, καθώς και για τον τρόπο με τον οποίο δύο ή περισσότεροι πράκτορες συνεργάζονται κάνοντας χρήση του μοντέλου συνεργασίας STEAM.

### 3.3.3.1. Soar

Η τεχνολογία του Soar [Laird 1987; Jones 1999] έχει τις ρίζες της στη γνωσιακή ψυχολογία και στην τεχνητή νοημοσύνη και βασίζεται στην θεώρηση ότι μία γνωσιακή αρχιτεκτονική (cognitive architecture) θα πρέπει να περιγράφει τη συμπεριφορά ενός πράκτορα σε τρία διακριτά επίπεδα: σε αυτό της γνώσης, των συμβόλων και σε αυτό της αρχιτεκτονικής [Newell, 1990].

Στο επίπεδο της γνώσης, όλες οι εργασίες που μπορεί να πραγματοποιήσει ένας πράκτορας περιγράφονται μέσω τελεστών (operators), δηλαδή μέσω κανόνων της μορφής «υπόθεση-ενέργεια» οι οποίοι αναλύονται ιεραρχικά δομώντας τα πλάνα του πράκτορα. Αυτοί οι τελεστές περιγράφουν τον τρόπο με τον οποίο ο πράκτορας σκέπτεται (deliberate) προσπαθώντας να υλοποιήσει μια ενέργεια ή να πετύχει έναν στόχο. Με αυτό τον τρόπο υλοποιούνται οι βασικές νοητικές διεργασίες του συστήματος. Κάθε μία ενέργεια μπορεί να είναι απλή αλλάζοντας είτε την εσωτερική κατάσταση του πράκτορα είτε τη κατάσταση του φυσικού περιβάλλοντος μέσα στο οποίο δρα ή να είναι σύνθετη ενέργεια αποτελούμενη από άλλες ενέργειες επίσης σύνθετες ή απλές.

Στο επίπεδο της αρχιτεκτονικής, το Soar αποτελείται από έναν κύκλο με τον οποίο πράκτορας διαρκώς δημιουργεί καινούργιους στόχους (δηλαδή ενέργειες που πρέπει να πραγματοποιήσει), επιλέγει τελεστές και τους εφαρμόζει αλλάζοντας κάθε φορά τη κατάσταση του περιβάλλοντός του. Η επιλογή του τελεστή για κάθε μία ενέργεια και η εφαρμογή αυτού γίνεται πάντα δυναμικά και εξαρτώνται από τη γνώση που έχει ο πρά-

κτορας για τη κατάσταση του φυσικού περιβάλλοντος καθώς και από τη γνώση που έχει για το πεδίο εφαρμογής (domain knowledge). Το Soar δεν διαθέτει κάποιο μηχανισμό εύρεσης και επίλυσης διενέξεων μεταξύ των ενεργειών που επιλέγει. Για αυτόν τον λόγο η γνώση σχετικά με το πεδίο εφαρμογής θα πρέπει να είναι πάντα αρκετή έτσι ώστε η επιλογή των τελεστών να μην είναι προβληματική. Αυτό το γεγονός περιορίζει τη συμπεριφορά του πράκτορα σε δυναμικά περιβάλλοντα διότι μπορεί να μην διαθέτει τον κατάλληλο χρόνο για να υπολογίσει όλες τις δυνατές επιλογές για τις ενέργειες που θέλει να πραγματοποιήσει.

### 3.3.3.2. Το μοντέλο συνεργασίας συντονισμένων ενεργειών (Joint Intentions)

Το μοντέλο των *συντονισμένων προθέσεων* (ή συντονισμένων ενεργειών) μίας ομάδας πρακτόρων [Cohen and Levesque, 1991; Cohen and Levesque, 1990] βασίζεται στην ιδέα ότι οι πράκτορες πρέπει να συντονίζουν τη νοητική τους κατάσταση, και πιο συγκεκριμένα τις προθέσεις τους, προκειμένου να έχουν την ικανότητα να συνεργάζονται μεταξύ τους (και όχι απλά να ενεργούν μαζί βάση ενός προκαθορισμένου πλάνου). Το ρόλο του «συντονιστεί» των προθέσεων ενός πράκτορα με τις προθέσεις των συνεργατών του, παίζει μια επιπλέον νοητική δομή αυτή της συντονισμένης πρόθεσης (joint intention).

Στο μοντέλο ορίζεται ότι μία ομάδα  $\Theta$  έχει τη πρόθεση να πραγματοποιήσει συντονισμένα μία ενέργεια  $a$  αν ο κάθε πράκτορας της ομάδας ( $\alpha$ ) αμοιβαία πιστεύει ότι η ομάδα θα πραγματοποιήσει την ενέργεια  $a$  και ( $\beta$ ) έχει τη πρόθεση να τη πραγματοποιήσει. Κατ' επέκταση του ορισμού αυτού, ορίζεται ο *διαρκής συντονισμένος στόχος* (Joint Persistent Goal (JPG)) μίας ομάδας: λέμε ότι μία ομάδα  $\Theta$  διαρκώς και συντονισμένα στοχεύει να πετύχει τη κατάσταση  $p$  (όπου  $p$  είναι μία πρόταση η οποία αναπαριστά το αποτέλεσμα μιας ομαδικής ενέργειας  $a$ ) με τη προϋπόθεση ότι ισχύει η πρόταση  $q$ , και συμβολίζουμε  $JPG(\Theta, p, q)$ , αν:

1. Όλα τα μέλη της ομάδας αμοιβαία πιστεύουν ότι η  $p$  δεν ισχύει στο τρέχοντα χρόνο.
2. Όλα τα μέλη της ομάδας έχουν ως στόχο να πετύχουν τη  $p$  και αμοιβαία γνωρίζουν ότι θέλουν να πετύχουν το στόχο αυτό, δηλαδή θέλουν τελικώς να ισχύσει η  $p$  ( $p$  to be eventually true).

3. Όλα τα μέλη της ομάδας αμοιβαία πιστεύουν ότι μέχρι αμοιβαία  $v$ ' αναγνωρίζουν ότι η πρόταση  $p$  έχει επιτευχθεί ή ότι είναι αδύνατον να επιτευχθεί ή ότι δεν είναι σχετική (irrelevant) ο κάθε πράκτορας έχει τον ασθενή στόχο (weak goal) να πετύχει τη  $p$ . Ένας πράκτορας  $\mu$  μίας ομάδας  $\Theta$  έχει τον ασθενή στόχο να πετύχει τη  $p$  και συμβολίζουμε  $WAG(\mu, p, \Theta, q)$ , όταν:

- ο πράκτορας  $\mu$  έχει τη πεποίθηση ότι στο τρέχοντα χρόνο η πρόταση  $p$  είναι ψευδής και θέλει να την κάνει αληθής, ή όταν

- ο πράκτορας έχει αναγνωρίσει ότι η πρόταση  $p$  έχει ήδη επιτευχθεί ή δεν μπορεί να επιτευχθεί ή ότι δεν είναι σχετική, και έχει δεσμευθεί να ενημερώσει τους άλλους πράκτορες της ομάδας για το γεγονός αυτό (δημιουργώντας έτσι μία αμοιβαία πεποίθηση).

Από τα παραπάνω είναι φανερό ότι για να δημιουργηθεί η συντονισμένη πρόθεση  $JPG(\Theta, p, q)$  όλοι οι πράκτορες πρέπει να δημιουργήσουν τις κατάλληλες αμοιβαίες πεποιθήσεις και δεσμεύσεις. Για το λόγο αυτό η ύπαρξη ενός πρωτοκόλλου επικοινωνίας που θα επιτρέπει τις κατάλληλες αιτήσεις-επιβεβαιώσεις (request-inform speech acts) είναι απαραίτητη [Tambe, 1997] και ονομάζεται πρωτόκολλο εδραίωσης δεσμεύσεων (establish commitment protocol). Για τη δημιουργία του πρωτοκόλλου αυτού χρησιμοποιείται μία επιπλέον νοητική δομή: η  $PWAG(v_i, p, \Theta)$ . Η δομή  $PWAG(v_i, p, \Theta)$  αναπαριστά τη δέσμευση ενός πράκτορα  $v_i$  της ομάδας  $\Theta$  να δημιουργήσει μία συντονισμένη πρόθεση για τη  $p$ . Αν  $v_i$  είναι ο πράκτορας που πρώτος αρχίζει την επικοινωνία με την ομάδα  $\Theta$  τότε σταδιακά επικοινωνεί με τους πράκτορες για να εδραιώσει (establish) τη κοινή δέσμευση. Αν όλοι οι πράκτορες συμφωνήσουν τότε αμοιβαία όλοι έχουν τη πεποίθηση ότι ισχύει η  $JPG(T, p, q)$ . Αν τουλάχιστον ένας πράκτορας δεν συμφωνήσει τότε δεν εδραιώνεται η κοινή πρόθεση. Στη τελευταία περίπτωση μπορεί να υπάρξει διαπραγμάτευση μεταξύ του πράκτορα  $\mu$  που πρώτος αιτείται την κοινή δέσμευση και αυτού που δεν αποδέχεται να συνεργαστεί, αλλά η περίπτωση αυτή παραμένει ανοικτή για το μοντέλο συνεργασίας STEAM.

Η σημασία ενός JPG κατά τη διάρκεια της συνεργασίας ενός πράκτορα με μία ομάδα πρακτόρων είναι ότι τον δεσμεύει να προσπαθεί συνεχώς να πετύχει το στόχο της ομαδικής ενέργειας μέχρι είτε να πιστέψει ότι ο στόχος αυτός έχει επιτευχθεί είτε ότι είναι αδύνατο να επιτευχθεί είτε ότι δεν είναι σχετικός. Αν ο στόχος επιτευχθεί ή αν ο πρά-

κτορας αποφασίζει ότι πρέπει να εγκαταλείψει τη προσπάθεια προς την επίτευξη του στόχου τότε δεσμεύεται να κάνει τη γνώση που έχει για τη κατάσταση του στόχου αμοιβαία γνωστή. Η δέσμευση του πράκτορα να σχηματίσει την αμοιβαία αυτή πεποίθηση έχει ως αποτέλεσμα να επικοινωνεί με τους άλλους πράκτορες της ομάδας και να τους ενημερώσει για τη κατάσταση της ομαδικής ενέργειας. Η δέσμευση για τη δημιουργία της αμοιβαίας πεποίθησης έχει ιδιαίτερη σημασία κατά τη διάρκεια της συνεργασίας διότι έτσι εξασφαλίζεται ότι οι πράκτορες της ομάδας παραμένουν ενημερωμένοι για την κατάσταση του κοινού τους στόχου. Αποτέλεσμα της αμοιβαίας αυτής πεποίθησης είναι ότι αποτρέπει τους πράκτορες να συνεχίσουν να πραγματοποιούν ενέργειες όταν ο στόχος δεν είναι εφικτός ή δεν είναι σχετικός για τουλάχιστον έναν πράκτορα ή έχει ήδη επιτευχθεί.

### 3.3.3.3. Η σημασία του μοντέλου συνεργασίας *Joint Intentions* στο STEAM

Η σημασία του μοντέλου συνεργασίας *συντονισμένων προθέσεων* στο STEAM είναι ότι επιτρέπει στους πράκτορες:

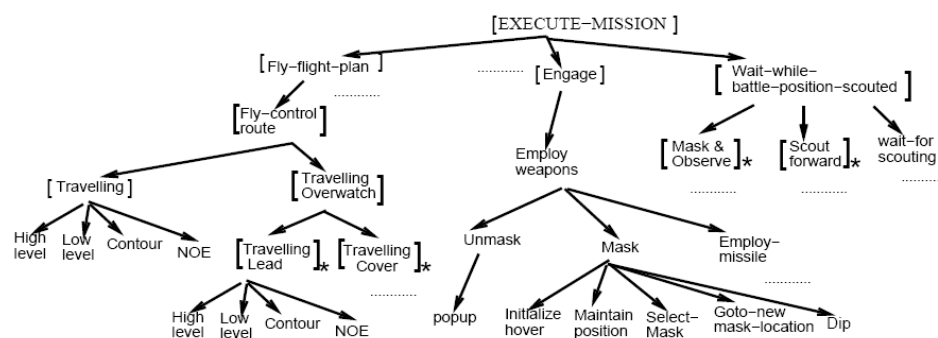
(α) να επικοινωνούν αποτελεσματικά και να συντονίζουν τις ενέργειες που πραγματοποιούν μεταξύ τους. Αν ένας πράκτορας έχει ένα  $WAG(\mu, p, \theta, \alpha)$ , δηλαδή επιθυμεί να πετύχει ένα στόχο  $p$  σε συνεργασία με μία ομάδα πρακτόρων  $\theta$  υπό τις συνθήκες  $\alpha$ , τότε κάνοντας χρήση του πρωτοκόλλου επικοινωνίας που αναφέραμε παραπάνω παρακινεί τους συνεργάτες του να δεσμευθούν να πετύχουν από κοινού την  $p$ , δηλαδή προσπαθεί να δημιουργήσει μία συντονισμένη διαρκείς πρόθεση  $JPG(\theta, p, \alpha)$ .

(β) να παρακολουθούν τις συντονισμένες προθέσεις τους και να ενημερώνουν τους συνεργάτες τους όταν πιστεύουν ότι ο στόχος έχει επιτευχθεί (δηλαδή, ότι η  $p$  ισχύει) ή όταν πιστεύουν ότι δεν μπορεί να επιτευχθεί (δηλαδή, ότι η  $p$  είναι αδύνατη) ή ότι ο στόχος δεν είναι πλέον σχετικός (δηλαδή, όταν  $\alpha$  είναι ψευδής).

(γ) να αναπαριστούν τους κοινούς στόχους της ομάδας και έτσι να μπορούν να εξάγουν συμπεράσματα (*reason about*) για τη κατάστασή της συνεργασίας (βλ. επόμενη παράγραφο).

Κάνοντας χρήση του Soar, το μοντέλο συνεργασίας STEAM δίνει τη δυνατότητα στους πράκτορες να πετυχαίνουν τις συντονισμένες προθέσεις τους αναπτύσσοντας ιεραρχικά πλάνα (βλ. Εικόνα 18).

Το βασικό δομικό στοιχείο των πλάνων αυτών είναι οι ατομικοί τελεστές (individual operators) που περιγράψαμε στην υποενότητα 3.3.3.1 και οι ομαδικοί τελεστές (team operators) οι οποίοι αναπαριστούν ομαδικά πλάνα με τα οποία οι πράκτορες υλοποιούν τις συντονισμένες προθέσεις τους (στην Εικόνα 18 οι ομαδικοί τελεστές συμβολίζονται με []). Ο κάθε ομαδικός τελεστής αποτελείται από (α) ένα σύνολο κανόνων που αναπαριστούν τις προϋποθέσεις που πρέπει να ισχύουν για να επιλέξει μία ομάδα πρακτόρων τον συγκεκριμένο τελεστή, (β) ένα σύνολο προϋποθέσεων εφαρμογής που πρέπει να ισχύουν κατά τη διάρκεια εκτέλεσης του τελεστή, και (γ) ένα σύνολο κανόνων που καθορίζουν το τερματισμό (επίτευξη) του συγκεκριμένου τελεστή.



Εικόνα 18: Ιεραρχικό πλάνο ενός πράκτορα στο πεδίο εφαρμογής εξομοίωσης αποστολών μη επανδρωμένων ελικοφόρων οχημάτων [Tambe, 1997].

Κάθε υπό-ενέργεια ενός ομαδικού τελεστή μπορεί να πραγματοποιηθεί είτε από έναν μόνο πράκτορα είτε από ένα υποσύνολο πρακτόρων της αρχικής ομάδας του τελεστή (στην Εικόνα 18 οι τελεστές αυτοί συμβολίζονται με ένα \*) ή ακόμα και από ολόκληρη την ομάδα. Για το αν ένας τελεστής θα είναι ομαδικός ή ατομικός καθορίζεται δυναμικά καθώς το πλάνο αναπτύσσεται από τους πράκτορες. Για παράδειγμα, σύμφωνα με το πλάνο της Εικόνα 18, στην αρχή η ομάδα των πρακτόρων που αναλαμβάνουν να εκτελέσουν την αποστολή ([EXECUTE-MISSION]) αναλαμβάνουν να καθορίσουν το σχέδιο πτήσης τους, μετά ο καθένας ξεχωριστά να επιλέξει τα όπλα που πιθανώς να χρησιμοποιήσει και τέλος όλοι μαζί κατευθυνόμενοι προς το πεδίο μάχης μία υπό-ομάδα ανα-

λαμβάνει να κάνει παρατήρηση και απόκρυψη, ενώ μία άλλη υπό-ομάδα αναλαμβάνει να κάνει αναγνώριση.

Για να υποστηριχθεί το μοντέλο συνεργασίας STEAM από μία αρχιτεκτονική όπως αυτή του Soar και dMARS θα πρέπει: (α) να υπάρχει η δυνατότητα αναπαράστασης των ομαδικών τελεστών (έτοιμων πλάνων), (β) να υπάρχει η δυνατότητα αναπαράστασης της κατάστασης τόσο της εσωτερικής των πρακτόρων (προσωπικές πεποιθήσεις, προθέσεις κτλ.) όσο και της ομάδας (αμοιβαίες πεποιθήσεις, JPGs κτλ.), (γ) δυνατότητα ανάπτυξης των πλάνων κάνοντας χρήση των έτοιμων τελεστών, (δ) δυνατότητα εντοπισμού συγκρούσεων μεταξύ των τελεστών έτσι ώστε οι πράκτορες να μπορούν να αναγνωρίζουν προβληματικές καταστάσεις είτε εντός των προσωπικών τους πλάνων είτε μεταξύ ενεργειών των ομαδικών πλάνων, (ε) δυνατότητα επίλυσης συγκρούσεων μεταξύ των τελεστών, και (στ) δυνατότητα διαπραγμάτευσης τόσο των τελεστών που θα χρησιμοποιηθούν για τη πραγματοποίηση των ομαδικών ενεργειών όσο και για την επίλυση των συγκρούσεων.

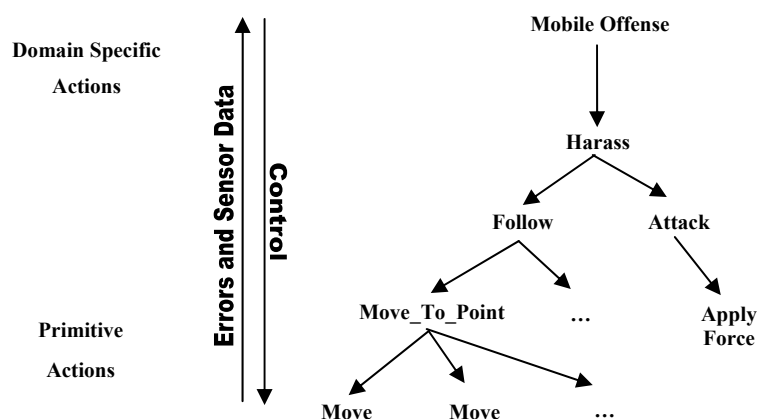
#### 3.3.4. HAC

Το HAC [Atkin et. al., 1999] είναι ένα πλαίσιο εργασίας ανάπτυξης πρακτόρων ικανών να δρουν σε δυναμικά περιβάλλοντα. Το κύριο χαρακτηριστικό του πλαισίου αυτού είναι η δυνατότητα που δίνει στους πράκτορες να σχεδιάζουν και να εκτελούν ιεραρχικά πλάνα ταυτόχρονα. Κάθε μία σύνθετη ενέργεια του πλάνου αναλύεται σε άλλες απλούστερες και αυτές με τη σειρά τους σε άλλες πιο απλές μέχρι τις βασικές οι οποίες ενεργοποιούν τους μηχανισμούς δράσης του πράκτορα (actuators). Οι βασικές ενέργειες είτε αλλάζουν τη κατάσταση του φυσικού περιβάλλοντος είτε περνούν μηνύματα του περιβάλλοντος (για παράδειγμα άλλων πρακτόρων) στις σύνθετες ενέργειες που τις περιέχουν.

Για κάθε μία ενέργεια του ιεραρχικού πλάνου ο πράκτορας εκτελεί τις ακόλουθες ενέργειες: (α) αντιδρά στα μηνύματα που προέρχονται από τα παιδιά της ενέργειας (δηλαδή από τις υποενέργειες), (β) ενημερώνει (update) τη εσωτερική κατάσταση του πράκτορα, (γ) σχεδιάζει τις επόμενες υπο-ενέργειες που πρέπει να πραγματοποιήσει και (δ) περνάει τυχόν μηνύματα στις ενέργειες πάνω απ' αυτήν.

Με αυτό τον τρόπο οι ενέργειες σε υψηλότερα επίπεδα θεωρούνται συλλογιστικές ενέργειες (cognitive actions) διότι σχεδιάζουν και ελέγχουν τις ενέργειες των κατώτερων επιπέδων. Αντίθετα, οι ενέργειες των κατώτερων επιπέδων θεωρούνται αντιδραστικές ενέργειες (reactive actions) διότι είτε αλλάζουν τη κατάσταση του φυσικού περιβάλλοντος είτε ελέγχουν τη ροή πληροφορίας απ' αυτές προς τις πιο σύνθετες ενέργειες των ανωτέρων επιπέδων (βλ. Εικόνα 19).

Από τα παραπάνω γίνεται εμφανές ότι η συμπεριφορά ενός πράκτορα, υλοποιημένου στο HAC, δεν μπορεί να ελεγχθεί με χρήση νοητικών διεργασιών με αποτέλεσμα η συμπεριφορά του πράκτορα να εξαρτάται αποκλειστικά από τον τρόπο με τον οποίο υλοποιούνται οι ενέργειες του πεδίου εφαρμογής (domain actions) και όχι από τον τρόπο με τον οποίο σχεδιάζει τις ενέργειες αυτές.

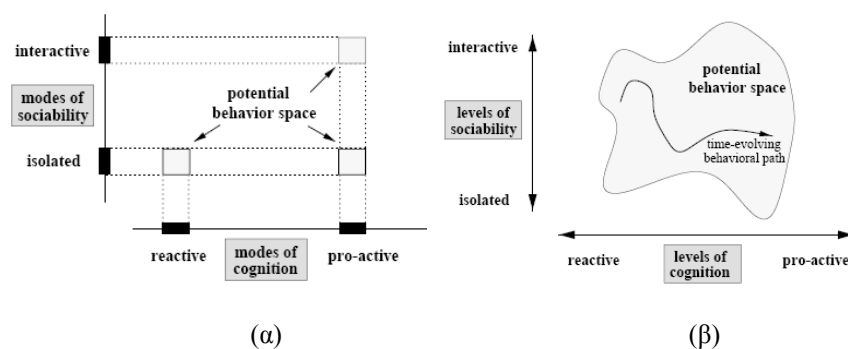


Εικόνα 19: Στα ιεραρχικά πλάνα του HAC οι σύνθετες ενέργειες ελέγχουν πάντα τις υποενέργειες των κατώτερων επιπέδων ενώ οι τελευταίες τροφοδοτούν τις πρώτες με μηνύματα που αφορούν τη κατάσταση του περιβάλλοντος στο οποίο δρουν.

### 3.3.5. CCAF

Το πλαίσιο εργασίας CCAF (constraint-centered architectural framework) βασίζεται σε μία αρχιτεκτονική προσανατολισμένη στο χειρισμό περιορισμών που θέτει είτε το εξωτερικό περιβάλλον είτε η εσωτερική κατάσταση του πράκτορα. Η βασική ιδέα στην οποία στηρίζεται η αρχιτεκτονική CCAF είναι ότι η συμπεριφορά ενός πράκτορα θα πρέπει να προκύπτει (emerge) από τον τρόπο με τον οποίο χειρίζεται τα πλάνα του και όχι μέσω ξεχωριστών διαδικασιών (modules της αρχιτεκτονικής του) που υλοποιούν τις συμπεριφορές του (όπως για παράδειγμα γίνεται στις αρχιτεκτονικές επιπέδων). Με

αυτό τον τρόπο η συμπεριφορά ενός πράκτορα μπορεί να προσδιοριστεί μέσω ενός συνεχούς χώρου συμπεριφορών (Εικόνα 20(β)) και όχι από έναν διακριτό (Εικόνα 20(α)).



Εικόνα 20: Διακριτός και συνεχής χώρος συμπεριφοράς.

Ο G. Weiss θέτει τέσσερις προϋποθέσεις για να μπορεί η συμπεριφορά ενός πράκτορα να προκύψει από έναν συνεχή χώρο συμπεριφορών διαμέσου του χειρισμού περιορισμών (constraint handling).

Η πρώτη προϋπόθεση είναι η άμεση σύνδεση των περιορισμών με τις προσωπικές ενέργειες του πράκτορα: (α) ο πράκτορας θα πρέπει να έχει την ικανότητα να αλλάζει και να τροποποιεί καταλλήλως το σύνολο των περιορισμών καθ' όσο ενεργεί ελέγχοντας πάντα τη μεταξύ τους συμβατότητα. (β) Ο πράκτορας θα πρέπει να μπορεί να εκτελεί ενέργειες υπό περιορισμούς χωρίς αυτοί να παραβιάζονται από τα αποτελέσματα των ενεργειών αυτών.

Η δεύτερη προϋπόθεση είναι ότι ο πράκτορας θα πρέπει να μπορεί να εργάζεται από κοινού με άλλους πράκτορες όταν οι περιορισμοί απαιτούν κάτι τέτοιο είτε για να πετύχει ένα στόχο τον οποίο από μόνος του δεν μπορεί είτε διότι έτσι μπορεί να πετύχει καλύτερα αποτελέσματα. Η συνεργασία μπορεί να υπάρξει είτε δίνοντας μία εργασία σε άλλον πράκτορα (delegation) είτε εργαζόμενος από κοινού για την επίτευξη ενός κοινού στόχου (cooperation).

Η τρίτη προϋπόθεση είναι ότι οι πράκτορες θα πρέπει να μπορούν να μετά-συλλογίζονται, δηλαδή να αποφασίζουν για τους περιορισμούς που θα χρησιμοποιήσουν και όποτε είναι αναγκαίο να συνεργάζονται για το σκοπό αυτό με άλλους πράκτορες. Ο τύπος του μετά-συλλογισμού θα πρέπει να είναι περισσότερο ποσοτικός παρά



ποιοτικός διότι ο ποιοτικός προσδιορισμός (όπως ο ποιοτικός προσδιορισμός χρονικών περιορισμών) μπορεί να είναι αρκετά σύνθετος για να πραγματοποιηθεί από τον πράκτορα.

Η τέταρτη προϋπόθεση αφορά τον τρόπο με τον οποίο μία αρχιτεκτονική πρέπει να υλοποιηθεί για να μπορεί ένας πράκτορας να χειρίζεται αποτελεσματικά περιορισμούς. Γενικά, λόγω του ότι οι περιορισμοί: (α) ισχύουν ή παραβιάζονται δυναμικά ως αποτέλεσμα των ενεργειών του πράκτορα ή/και γενικότερα των αλλαγών που συμβαίνουν στο περιβάλλον, (β) κάθε περιορισμός μπορεί να έχει εφαρμογή σε παραπάνω από μία ενέργεια σε ένα πλαίσιο ενεργειών (context constraints), (γ) ο μηχανισμός χειρισμού των περιορισμών θα πρέπει να μην παραβιάζει τους ίδιους τους περιορισμούς (όπως για παράδειγμα τους χρονικούς περιορισμούς των ενεργειών του πράκτορα που ενδεχομένως να υπάρχουν), η υλοποίηση του μηχανισμού ελέγχου των περιορισμών θα πρέπει να είναι κεντρικοποιημένη. Στην αντίθετη περίπτωση καταναμημένου χειρισμού των περιορισμών θα πρέπει οι πράκτορες να διαθέτουν τις κατάλληλες εσωτερικές ενέργειες διαπραγμάτευσης (negotiation) και χειρισμού εξαιρέσεων (exception handling) περιορισμών οι οποίες θα εγγυώνται την εγκυρότητα αυτών.

### **3.3.6. Σύγκριση πλαισίων ανάπτυξης έξυπνων πρακτόρων για δυναμικά και απρόβλεπτα περιβάλλοντα**

Τα παραπάνω αντιπροσωπευτικά πλαίσια εργασίας ανάπτυξης πρακτόρων για δυναμικά περιβάλλοντα μπορούν να κατηγοριοποιηθούν σε δύο κύριες κατηγορίες:

(α) σε εκείνες που η αρχιτεκτονική τους βασίζεται στη φιλοσοφία της πρακτικής συλλογιστικής και ακολουθούν το μοντέλο BDI (όπως οι αρχιτεκτονικές των dMARS και InteRRaP) και

(β) στις γνωσιακές αρχιτεκτονικές οι οποίες περιγράφουν τη συμπεριφορά των πρακτόρων χρησιμοποιώντας γνώση (περιορισμούς και κανόνες του τύπου αν-τότε) για το πώς πρέπει να δράσουν σε συγκεκριμένες καταστάσεις του περιβάλλοντος (όπως οι αρχιτεκτονικές των Soar, HAC και CCAF).

Οι τρόποι συμπεριφοράς των πρακτόρων που βασίζονται στις αρχιτεκτονικές των dMARS, Soar και HAC δεν ξεχωρίζουν με διακριτό τρόπο. Δηλαδή δεν είναι διακριτό το πότε οι πράκτορες συμπεριφέρονται αντανακλαστικά ή στοχαστικά ή συνεργάζονται

με άλλους πράκτορες. Οι πράκτορες αναπτύσσουν ιεραρχικά πλάνα στα οποία οι ενέργειες των υψηλότερων επιπέδων αναπαριστούν στόχους που πρέπει να επιτευχθούν από ενέργειες των κατωτέρων επιπέδων. Οι ενέργειες αυτές των ανωτέρων επιπέδων εκφράζουν το στοχαστικό τρόπο συμπεριφοράς των πρακτόρων ενώ οι ενέργειες των κατωτέρων επιπέδων που είτε παρέχουν πληροφορίες στους στόχους ή αλλάζουν τη κατάσταση του περιβάλλοντος όπου δρα ο πράκτορας εκφράζουν τον αντανακλαστικό τρόπο συμπεριφοράς. Αποτέλεσμα αυτού του γεγονότος είναι ότι η συμπεριφορά του πράκτορα σε ότι αφορά τους μηχανισμούς συλλογισμού και δράσης είναι συγκεκριμένοι χωρίς να δίνεται η δυνατότητα ελέγχου αυτών των μηχανισμών και ακόμα περισσότερο προσαρμογής αυτών στις αλλαγές τους περιβάλλοντός του.

Αντίθετα, οι τρόποι συμπεριφοράς πρακτόρων που βασίζονται σε αρχιτεκτονικές επιπέδων όπως αυτή του InteRRaP διακρίνονται μεταξύ τους και υλοποιούνται από διακριτά επίπεδα λογισμικού: το αντανακλαστικό, το στοχαστικό και το κοινωνικό. Η διάκριση των τρόπων συμπεριφοράς του πράκτορα μεταξύ αυτών των τριών επιπέδων περιορίζει τη συνολική συμπεριφορά του διότι τον αναγκάζει να εναλλάσσει τη συμπεριφορά του μόνο μεταξύ αυτών των τριών διακριτών επιπέδων θεωρώντας ότι δεν υπάρχει τίποτα μεταξύ αυτών. Ανάλογα με τη κατάσταση του περιβάλλοντος κάθε ένας στόχος του πράκτορα αντιστοιχίζεται σε έναν και μόνο τρόπο συμπεριφοράς, γεγονός που δεν του επιτρέπει να προσαρμόζει το πόσο θα «σκεφτεί» για το συγκεκριμένο στόχο ή αν θα ενεργήσει γρήγορα (με αντανακλαστικό τρόπο) επιλέγοντας ένα πλάνο το οποίο μπορεί να μην είναι βέλτιστο αλλά όμως να είναι ικανοποιητικό δεδομένων των περιορισμένων πόρων που διαθέτει.

Συνοψίζοντας τα παραπάνω διαπιστώνουμε ότι οι αρχιτεκτονικές αυτές είτε δεν διακρίνουν με ακριβή και ελέγξιμο τρόπο τον τρόπο με τον οποίο συμπεριφέρεται ένας πράκτορας είτε όταν το κάνουν περιορίζουν τη συμπεριφορά του μεταξύ συγκεκριμένων τρόπων. Η μόνη αρχιτεκτονική η οποία είναι κοντά στους αντικειμενικούς στόχους του προτεινόμενου πλαισίου ανάπτυξης ICAGENT είναι αυτή του CCAF, η οποία βασίζεται στη θεώρηση ότι η συμπεριφορά του πράκτορα δεν πρέπει να περιορίζεται μεταξύ διακριτών τρόπων αλλά πρέπει να υπάρχουν μονάδες λογισμικού που να ελέγχουν τον τρόπο συμπεριφοράς του πράκτορα για κάθε μία ενέργεια ξεχωριστά. Η βασική ιδέα με την οποία έχει υλοποιηθεί η συγκεκριμένη αρχιτεκτονική είναι ότι η συμπεριφορά ενός πράκτορα θα πρέπει να προκύπτει (emerges) από τον τρόπο με τον οποίο ο πράκτορας

χειρίζεται τους περιορισμούς των ενεργειών του. Κάτω από αυτή τη θεώρηση θέτει τις τέσσερις προϋποθέσεις που περιγράψαμε στην υποενότητα 3.3.5.

### 3.3.7. Το προτεινόμενο πλαίσιο ανάπτυξης πρακτόρων ICAGENT

Όπως τονίσαμε και στην ενότητα 2.2, σε αντίθεση με τα παραπάνω πλαίσια ανάπτυξης πρακτόρων, η συμπεριφορά ενός πράκτορα στο πλαίσιο ανάπτυξης έξυπνων συνεργατικών πρακτόρων ICAGENT είναι αποτέλεσμα (emerges) των συλλογιστικών εργασιών (reasoning tasks) που αυτός διαθέτει. Δηλαδή, η συμπεριφορά του πράκτορα εξαρτάται από τον τρόπο με τον οποίο αντιλαμβάνεται το περιβάλλον του (environment monitoring), αναγνωρίζει καταστάσεις στις οποίες πρέπει να δράσει (situation recognition), από τον τρόπο με τον οποίο αναγνωρίζει και επιλύει τυχόν συγκρούσεις μεταξύ των ενεργειών που προτίθεται να εκτελέσει (reconciliation), καθώς και από τον τρόπο με τον οποίο αναπτύσσει και εκτελεί τα πλάνα του (plan elaboration and realization) είτε μόνος του είτε σε συνεργασία με άλλους πράκτορες.

Υλοποιώντας τις παραπάνω συλλογιστικές εργασίες μέσω νοητικών ενεργειών και χειριζόμενος αυτές κάτω από το ίδιο πλαίσιο με τις άλλες ενέργειες εφαρμογής (domain actions), ο πράκτορας έχει τη δυνατότητα να «ρυθμίζει» τη συμπεριφορά του επιλέγοντας το πια γεγονός πρέπει ν' αντιλαμβάνεται στο περιβάλλον του, το πόσο σχολαστικός πρέπει να είναι σε ότι αφορά τυχόν συγκρούσεις μεταξύ ενεργειών στο πλάνο του και να αποφασίζει τον τρόπο με τον οποίο θα αναπτύξει και θα πραγματοποιήσει τα πλάνα του (προσωπικά ή συνεργατικά). Στο Κεφάλαιο 4 δίνουμε αναλυτικότερα το πώς ο πράκτορας χειρίζεται τις ενέργειες αυτές προκειμένου να πραγματοποιήσει τις συλλογιστικές του διεργασίες.

## 3.4. Πλαίσια ανάπτυξης έξυπνων συνεργατικών πρακτόρων επιφανίων διεπαφής

Όπως αναφέραμε στο Κεφάλαιο 2 ο δεύτερος τομέας εφαρμογής του πλαισίου ανάπτυξης έξυπνων συνεργατικών πρακτόρων ICAGENT είναι αυτός των πρακτόρων επιφανείων διεπαφής. Παρ' όλο του ότι δεν μπορούμε να κατηγοριοποιήσουμε τους πράκτορες επιφανείων διεπαφής *άμεσου χειρισμού* με ρητό (explicit) τρόπο μπορούμε να διακρίνουμε δύο κύριες κατηγορίες: τους συμβουλευτικούς πράκτορες (advisor-style

agents) και τους επικουρικούς πράκτορες (assistant-style agents) [Kourakos-Mavromichalis and Vouros, 2006].

Οι **συμβουλευτικού-τύπου πράκτορες** [Selker, 1994; de Rosis, 1999] στοχεύουν στο να διδάξουν τους χρήστες να εκτελούν σωστά τις εργασίες του χώρου εφαρμογής (domain tasks) παρέχοντας τους βοηθητικές πληροφορίες. Η παρεχόμενη πληροφορία μπορεί να προσαρμόζεται στα χαρακτηριστικά του χρήστη και στη γνώση που ήδη έχει ενώ μπορεί να παρουσιάζεται με παραπάνω από μία μορφές (forms) και τύπους (styles). Οι συμβουλές μπορεί να παρέχονται είτε κατόπιν αίτησης του χρήστη [de Rosis, 1999] είτε ευκαιριακά παρατηρώντας τις ενέργειες του χρήστη στην επιφάνεια διεπαφής [Selker, 1994].

Οι **επικουρικού-τύπου πράκτορες** [Maes, 1994; Shneiderman and Maes, 1997; Lieberman, 1997] είναι συμπληρωματικοί των επιφανειών διεπαφής «άμεσου χειρισμού» και ακολουθούν το μοντέλο του «έξυπνου μπάτλερ» που πρώτος ο N. Negroponte (1970) εισήγαγε: οι πράκτορες αυτοί αναλαμβάνουν τη πρωτοβουλία να δράσουν πραγματοποιώντας ενέργειες στην επιφάνεια διεπαφής για λογαριασμό του χρήστη. Ο στόχος αυτών των πρακτόρων δεν είναι να πληροφορήσουν το χρήστη για το πώς πρέπει να χειριστεί την επιφάνεια διεπαφής αλλά να τον απελευθερώσουν από την εκτέλεση πολύπλοκων ενεργειών ή ενεργειών ρουτίνας.

Για να παρέχει ένας πράκτορας σε έναν χρήστη τη βοήθεια που πραγματικά χρειάζεται χωρίς να του τη ζητήσει με ρητό τρόπο, αναγκαία προϋπόθεση είναι να έχει την ικανότητα ν' αναγνωρίζει τους στόχους του (από τις ενέργειες που ο χρήστης πραγματοποιεί στην επιφάνεια διεπαφής), να διακρίνει τυχόν λάθη του και μετά ν' αποφασίζει τον τρόπο με τον οποίο θα τον βοηθήσει. Οι συμβουλές που παρέχουν οι παραπάνω πράκτορες ή οι ενέργειες που πραγματοποιούν εκ μέρους του χρήστη δεν βασίζονται στο πλαίσιο του διαλόγου (dialogue context) που αναπτύσσεται μεταξύ του χρήστη και του πράκτορα. Αποτέλεσμα αυτού του γεγονότος είναι ότι είτε περιορίζεται η παρεχόμενη βοήθεια του πράκτορα προς το χρήστη σε απλές συμβουλές πάνω στις τρέχουσες ενέργειες, είτε πραγματοποιεί μόνο ενέργειες που του αναθέτει ο χρήστης, είτε αναλαμβάνει τη πρωτοβουλία να πραγματοποιήσει ενέργειες βασισμένος σε ποσοτικούς προσδιορισμούς οι οποίοι δεν εγγυώνται πάντα ότι βρίσκουν το τι πραγματικά θέλει να κάνει ο χρήστης (παράδειγμα τέτοιου προσδιορισμού είναι: αν ο πράκτορας έχει παρα-

τηρήσει ότι μέχρι  $a$  φορές έχει πραγματοποιήσει ο χρήστης την ενέργεια  $b$  τότε την  $a+1$  φορά ο πράκτορας θα πραγματοποιήσει για αυτόν την ενέργεια αυτή).

Στο υπόλοιπο του κεφαλαίου αυτού θ' αναφερθούμε μόνο στα πλαίσια ανάπτυξης πρακτόρων επιφανειών διεπαφής τα οποία κάνουν χρήση πλαισίων διαλόγων και ειδικότερα σε εκείνα που χρησιμοποιούν το ίδιο μοντέλο συνεργασίας με αυτό του ICAGENT, δηλαδή αυτό των *SharedPlans*. Το μοντέλο αυτό έχει εφαρμοστεί για την ανάπτυξη δύο τεχνολογιών για πράκτορες επιφανειών διεπαφής: αυτό του Collagen [Rich, Sidner and Lesh, 2001; Rickel et. al., 2001] και του Dial [Ortiz and Grosz, 2000].

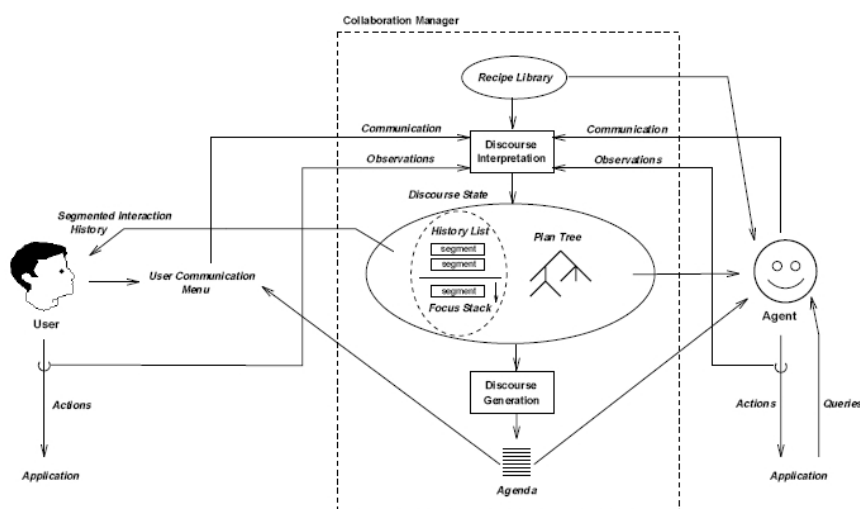
#### 3.4.1. Collagen

Το Collagen παρέχει μία ανεξαρτήτου εφαρμογής τεχνολογία, η οποία μετατρέπει απλές επιφάνειες διεπαφής σε συνεργατικές διαμέσου συμβουλευτικών ή επικουρικών πρακτόρων διεπαφής. Πιο συγκεκριμένα, το Collagen χρησιμοποιεί τη θεωρία διαλόγων συνεργατικών πρακτόρων του μοντέλου *SharedPlans* (*SharedPlans theory of discourse*) [Grosz and Kraus, 1996] για την ανάπτυξη ενός ενδιάμεσου λογισμικού το οποίο υποστηρίζει τη συνεργασία του χρήστη με τον πράκτορα της επιφάνειας διεπαφής. Η βασική ιδέα στην οποία στηρίζεται το Collagen είναι ότι χρησιμοποιώντας έναν πράκτορα παράλληλα με μία επιφάνεια διεπαφής' ο οποίος επικοινωνεί με τον χρήστη βασιζόμενος στους συνηθισμένους κανόνες διαλόγων μεταξύ των ανθρώπων θα είναι ευκολότερη τόσο η εκμάθηση της επιφάνειας διεπαφής όσο και ο χειρισμός της από έναν άπειρο χρήστη.

Κλειδί στην ανάπτυξη του Collagen είναι η καταγραφή της κατάστασης του διαλόγου (*discourse state*) μεταξύ του χρήστη και του πράκτορα μέσω ενός λογισμικού το οποίο ερμηνεύει τις ενέργειες τους στην επιφάνεια διεπαφής αναγνωρίζοντας τις προθέσεις τόσο του πράκτορα όσο και του χρήστη. Η κατάσταση του διαλόγου χρησιμοποιείται για την αναπαράσταση και συντήρηση (*maintenance*) της κατάστασης της συνεργασίας μεταξύ του χρήστη και του πράκτορα.

Το Collagen έχει χρησιμοποιηθεί για την ανάπτυξη εκπαιδευτών πρακτόρων (*tutor agents*) ικανών να εκπαιδεύουν χρήστες στο να χειρίζονται μηχανές (PACO) [Rickel, Lesh et. al., 2001] και αεροπορικών πρακτόρων οι οποίοι βοηθούν τους χρήστες να κλείνουν αεροπορικά εισιτήρια (Triton) [Davies et. al., 2001].

Το παράδειγμα συνεργατικού πράκτορα επιφάνειας διεπαφής το οποίο ακολουθείται στο Collagen είναι αυτό της Pattie Maes [Maes, 1994] που περιγράψαμε στην ενότητα 2.3. Όπως έχουμε τονίσει, σ' αυτό ο πράκτορας δρα ανεξάρτητα της επιφάνειας διεπαφής, επικοινωνεί με τον χρήστη, παρατηρεί τις ενέργειές του και αναλαμβάνει τη πρωτοβουλία να τον βοηθήσει όποτε αυτό είναι αναγκαίο.



Εικόνα 21: Η αρχιτεκτονική του Collagen.

Η αρχιτεκτονική του Collagen απεικονίζεται στην Εικόνα 23. Ακολουθώντας τη δουλειά των Grosz και Sidner [Grosz and Sidner, 1986] το Collagen διαχωρίζει τη κατάσταση του διαλόγου μεταξύ του πράκτορα και του χρήστη σε τρία επίπεδα: σε αυτό της γλωσσικής δομής (linguistic structure), το σύνολο των τρεχόντων στόχων (attentional state) και στο πλάνο των προθέσεων τους (intentional structure).

Η γλωσσική δομή υλοποιείται ως μία στοίβα από τμήματα κειμένου που ο χρήστης ή ο πράκτορας εκφωνούν<sup>14</sup> κατά τη διάρκεια της συνεργασίας. Κάθε τμήμα κειμένου αποτελείται από μία συνεχή σειρά από ενέργειες ή λεκτικά που συνεισφέρουν στην επίτευξη ενός στόχου. Το πλάνο των προθέσεων είναι ένα ιεραρχικό πλάνο το οποίο αναπαριστά τις κοινές προθέσεις μεταξύ του πράκτορα και του χρήστη (έτσι όπως το

περιγράψαμε στην υποενότητα 3.2.4). Τέλος, το σύνολο των τρεχόντων στόχων περιλαμβάνει την τρέχουσα ενέργεια που προσπαθεί να πραγματοποιήσει ο πράκτορας καθώς και όλους τους υπερ-στόχους που τη περιλαμβάνουν.

Κάθε ενέργεια που εκτελεί ο χρήστης ή ο πράκτορας στη διεπιφάνεια της εφαρμογής ερμηνεύεται (διαμέσου του μηχανισμού ερμηνείας (discourse interpretation)) από το Collagen κάτω από το κοινό πλαίσιο των υπο-ενεργειών του κοινού στόχου ενημερώνοντας έτσι τη κατάσταση του διαλόγου (discourse state). Η ενημέρωση της κατάστασης του διαλόγου έχει ως αποτέλεσμα τη αναγνώριση νέων ενεργειών ή προτάσεων (μέσω της διαδικασίας discourse generation) που πρέπει να πραγματοποιήσει ο πράκτορας ή να παρουσιαστούν στο χρήστη.

#### 3.4.2. DIAL

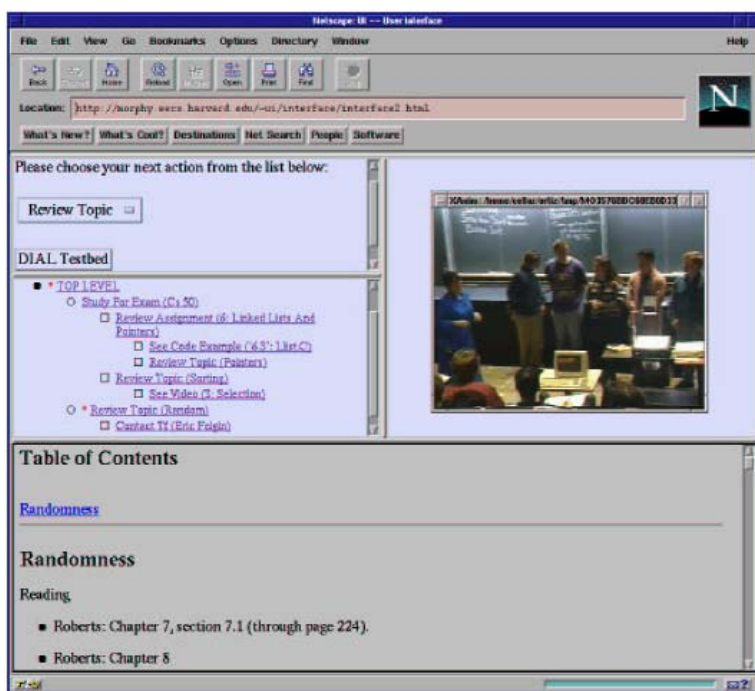
Το DIAL βασίζεται στην ιδέα ότι το μοντέλο ανάθεσης εργασιών master/slave δεν είναι το κατάλληλο για σχεδιασμό επιφανειών διεπαφής συστημάτων εξ' αποστάσεως εκπαίδευσης ή απλής παροχής πληροφοριών. Το σύστημα θα πρέπει να δίνει στο χρήστη τη δυνατότητα να επικοινωνεί τους στόχους του και όχι να απαιτεί απ' αυτόν να περιγράψει τα ακριβή βήματα που πρέπει να εκτελέσει το σύστημα προκειμένου να λάβει τη πληροφορία που χρειάζεται. Επιπλέον για να έχει το σύστημα τη δυνατότητα να συνεργάζεται με τον χρήστη θα πρέπει η επιφάνεια διεπαφής του να παρέχει στο χρήστη τη δυνατότητα να περιγράφει όχι μόνο το τί θέλει να πετύχει αλλά και για ποιό λόγο έτσι ώστε η πληροφορία που του παρουσιάζει να μην είναι έξω από τα ενδιαφέροντα του.

Το DIAL έχει χρησιμοποιηθεί για την ανάπτυξη ενός συνεργατικού περιβάλλοντος εξ' αποστάσεως εκπαίδευσης το οποίο παρέχει πληροφορία στο χρήστη ερμηνεύοντας τα αιτήματά του κάτω από ένα κοινό πλαίσιο ενεργειών (context of action). Το μοντέλο συνεργασίας στο οποίο βασίζεται το DIAL είναι αυτό των *SharedPlans* που περιγράψαμε στην ενότητα 3.2.

---

<sup>14</sup> Η εκφώνηση μίας ενέργειας συνήθως γίνεται είτε κάνοντας κλικ σε μία επιλογή ενός μενού είτε εκφωνώντας μία λέξη αν προστεθεί στο Collagen δυνατότητα αναγνώρισης φωνής.

Ο σκοπός του DIAL δεν είναι να υλοποιήσει το μοντέλο αυτό αλλά να χρησιμοποιήσει τις δομές που παρέχει για ν' αναπαραστήσει τόσο τις προθέσεις του χρήστη όσο του συστήματος έτσι όπως προκύπτουν από το διάλογο που έχουν (the intentional structure of the dialogue) καθώς και το σύνολο των τρεχόντων στόχων τους (the attentional state). Όπως αναφέραμε και στο προηγούμενο κεφάλαιο αυτές οι δύο δομές παίζουν πρωταρχικό ρόλο στην επιτυχή συνεργασία μεταξύ πρακτόρων.



Εικόνα 22: Το πλάνο εντοπισμού πληροφορίας αναπαρίσταται με ένα ιεραρχικό πλάνο το οποίο απεικονίζει το διάλογο του χρήστη με το σύστημα.

Κεντρικό ρόλο στην αρχιτεκτονική του DIAL παίζει το «πλάνο εντοπισμού πληροφορίας» (information-locating plan) το οποίο αναπαρίσταται στην επιφάνεια διεπαφής με ένα ιεραρχικό πλάνο από links (βλ. Εικόνα 22). Για να εντοπίζει αποτελεσματικότερα τη πληροφορία που επιθυμεί ο χρήστης το DIAL απεικονίζει τόσο τις προθέσεις του χρήστη όσο και τη κατάσταση των ενεργειών τρέχουσας εστίασης στο πλάνο εντοπισμού πληροφορίας. Με αυτό τον τρόπο γίνεται γνωστό τόσο στο σύστημα όσο και στο χρήστη: (α) οι κοινές προθέσεις τους, (β) οι λόγοι για τους οποίους ο χρήστης αναζητά μία πληροφορία, και τέλος (γ) η τρέχουσα ενέργεια εστίασης του χρήστη για την οποία



το σύστημα δίνει τη δυνατότητα στο χρήστη να την αλλάξει όποτε αυτός επιθυμεί να εστιάσει τη προσοχή του σε άλλη πληροφορία.

### 3.4.3. Σύγκριση πλαισίων ανάπτυξης έξυπνων πρακτόρων επιφανειών διεπαφής βασισμένων στο μοντέλο των *SharedPlans*

Παρ' όλο του ότι έχουν προταθεί και άλλα μοντέλα συνεργασίας μεταξύ πρακτόρων [Cohen and Levesque, 1991; Sonenberg et al., 1992] το μοντέλο των *SharedPlans* έχει αναπτυχθεί για να μοντελοποιήσει το διάλογο που προκύπτει από τη συνεργασία δύο ή περισσότερων πρακτόρων (collaborative discourse) και για αυτό θεωρείται ως το πιο κατάλληλο μοντέλο για την ανάπτυξη πρακτόρων επιφανειών διεπαφής ικανών να συνεργάζονται με το χρήστη [Ortiz and Grosz, 2002]. Η υποστήριξη των μερικώς ανεπτυγμένων πλάνων, η δυνατότητα καταγραφής των ενεργειών των πρακτόρων σε ένα κοινό πλαίσιο ενεργειών και σε συνδυασμό με τη δυνατότητα που παρέχει το μοντέλο *SharedPlans* να καθιστά τους πράκτορες ικανούς να εμφανίζουν εξυπηρετική συμπεριφορά (helpful behaviour) στους συνεργάτες τους είναι τα κύρια χαρακτηριστικά που κάνει το μοντέλο αυτό να διαφέρει από άλλα που έχουν προταθεί. Η βασική ιδέα στην οποία βασίζεται τόσο το πλαίσιο ανάπτυξης πρακτόρων ICAGENT όσο το DIAL και το Collagen είναι ότι μία επιφάνεια διεπαφής μίας εφαρμογής η οποία βασίζεται στους συνηθισμένους κανόνες και υποθέσεις που διέπουν τους διαλόγους μεταξύ ανθρώπων είναι πιο αποτελεσματική και κατανοητή από το χρήστη από μία που δεν βασίζεται σε αυτούς.

Σε ότι αφορά το DIAL, ο στόχος του είναι να καταστήσει το χρήστη ικανό να βρίσκει τη πληροφορία που χρειάζεται με τη βοήθεια του συστήματος, χωρίς να καθορίζει τον ακριβή τρόπο με τον οποίο το σύστημα θα πρέπει να βρει τη πληροφορία αυτή. Για να επιτευχθεί αυτό, το DIAL χρησιμοποιώντας το μοντέλο των *SharedPlans* δίνει τη δυνατότητα στο χρήστη και στο σύστημα ν' αναπτύσσουν διαλόγους (μέσω ιεραρχικών πλάνων) με τους οποίους κάνουν γνωστές τόσο τις προθέσεις τους (intentional structure of the dialogue) όσο και τους στόχους για τους οποίους οι προθέσεις αυτές γίνονται (the attentional state of the dialogue). Στο πλαίσιο εργασίας του ICAGENT (όσο και στο Collagen) γίνεται χρήση προκαθορισμένων μεθόδων (recipes) με τις οποίες ο πράκτορας μπορεί όχι μόνο να παρουσιάζει τη κατάσταση του διαλόγου, αλλά και να ενεργεί στην επιφάνεια διεπαφής καθώς και να εξηγεί (κάνοντας χρήση εκπαιδευτικών στρατη-

γικών) τι κάνει. Επιπλέον, το πλαίσιο ανάπτυξης του ICAGENT διαθέτει τους κατάλληλους μηχανισμούς (διεργασίες συλλογισμού) με τις οποίες ο πράκτορας μπορεί ν' αντιλαμβάνεται τις λανθασμένες ενέργειες του χρήστη και να τον βοηθά να τις επιλύσει αναπτύσσοντας κοινά πλάνα. Αντίθετα το DIAL λόγω της έλλειψης τέτοιων μηχανισμών δεν έχει την ικανότητα να βοηθά το χρήστη όταν αυτός ενεργεί λανθασμένα.

Σε ότι αφορά τις διαφορές του ICAGENT από το Collagen αυτές διακρίνονται πρώτον στη βασική θεώρηση του μοντέλου που το κάθε πλαίσιο χρησιμοποιεί για την ανάπτυξη του πράκτορα διεπαφής και δεύτερον, στη διάκριση και τη σχέση που υπάρχει μεταξύ της αναγνώρισης των πρωταρχικών στόχων του χρήστη, των κοινών στόχων και της συνεργασίας.

Το πλαίσιο ανάπτυξης πρακτόρων ICAGENT υλοποιώντας το μοντέλο συνεργασίας *SharedPlans* δεν θεωρεί το πράκτορα διεπαφής ως μία ακόμα εναλλακτική επιλογή των επιφανειών διεπαφής αλλά ως οντότητες λογισμικού οι οποίες συμπληρώνουν τις επιφάνειες αυτές δίνοντας τους ικανότητες συνεργασίας με το χρήστη μέσω διαλόγων. Το Collagen υποστηρίζει τη βασική θεώρηση αυτή αλλά την υλοποιεί μέσω ενός ενδιάμεσου του πράκτορα και της εφαρμογής λογισμικού το οποίο καταγράφει το διάλογο του χρήστη με τον πράκτορα (collaboration manager) (βλ. Εικόνα 21). Αποτέλεσμα αυτού του γεγονότος είναι ότι για κάθε εφαρμογή στην οποία χρησιμοποιείται το Collagen χρειάζεται να υλοποιηθούν επιπλέον οι μηχανισμοί του πράκτορα που αφορούν την επικοινωνία του με τον χρήστη καθώς και ο μηχανισμός εντοπισμού εσφαλμένων ενεργειών του χρήστη στην επιφάνεια διεπαφής [Garland et. al. 2003]. Η τελευταία ικανότητα είναι σημαντική για να μπορεί ο πράκτορας να βοηθά το χρήστη όταν χρειάζεται.

Επιπλέον, το παράδειγμα διαλόγου που χρησιμοποιεί το Collagen είναι αυτό της σύνθεσης τμημάτων κειμένου τα οποία μπορούν να οδηγήσουν σε μακρούς διαλόγους με το χρήστη αποσπώντας τη προσοχή του όταν αυτοί δεν είναι αναγκαίοι. Αντίθετα, στο πράκτορα διεπαφής του ICAGENT για την αναπαράσταση του διαλόγου χρησιμοποιούμε πλαίσια διαλόγου (βλ. Εικόνα 10(α) και Εικόνα 11) με τα οποία ο πράκτορας παρουσιάζει τις υποθέσεις του για τους τρέχοντες πιθανούς στόχους του χρήστη (πριν ο χρήστης συμφωνήσει να συνεργαστεί με τον πράκτορα), τους κοινούς στόχους που ο χρήστης συμφωνεί να πετύχει σε συνεργασία με τον πράκτορα και τη τρέχουσα μέθοδο που από κοινού συμφωνούν να χρησιμοποιήσουν για να πετύχουν τους στόχους αυτούς.

Ο χρήστης από τη πλευρά του, μπορεί είτε να αγνοήσει τις παροτρύνσεις του πράκτορα για συνεργασία και γενικότερα την ύπαρξη των πλαισίων διαλόγου και να συνεχίσει να ενεργεί στην επιφάνεια διεπαφής, είτε να χρησιμοποιήσει τα πλαίσια αυτά για να δηλώσει τους πρωταρχικούς του στόχους (αρχίζοντας έτσι τη συνεργασία με τον πράκτορα), ν' αλλάξει τον τρέχοντα στόχο, ν' αναθέσει στο πράκτορα τη πραγματοποίηση μίας ενέργειας ή ν' απορρίψει τη μέθοδο που ο πράκτορας του προτείνει για τον τρέχοντα κοινό στόχο τους.

Η τρίτη σημαντική διαφορά μεταξύ του ICAGENT και του Collagen είναι η διάκριση και η σχέση που υπάρχει μεταξύ της αναγνώρισης των επιθυμιών και προθέσεων του χρήστη πριν και κατά τη διάρκεια της συνεργασίας. Σε αντίθεση με το Collagen που η ερμηνεία των ενεργειών του χρήστη γίνεται μόνο κάτω από ένα κοινό στόχο που ο χρήστης έχει ενημερώσει το πράκτορα ότι θέλει να πετύχει στο πλαίσιο εργασίας του ICAGENT ξεχωρίζουμε δύο καταστάσεις στη συμπεριφορά του πράκτορα σε ότι αφορά τις ερμηνείες των ενεργειών του χρήστη: τη συμπεριφορά του πράκτορα πριν τη συνεργασία και τη συμπεριφορά του πράκτορα κατά τη διάρκεια της συνεργασίας. Αυτή η διάκριση σε συνδυασμό με το σαφή διαχωρισμό των επιθυμιών του χρήστη και των προθέσεων, επιτρέπει στον πράκτορα να χειρίζεται αποτελεσματικά τη συμπεριφορά του χρήστη τόσο όταν είναι συγκεντρωμένος στην επίτευξη ενός στόχου (focused user) όσο και όταν δεν είναι (unfocused user).

Στη περίπτωση που ο πράκτορας και ο χρήστης δεν έχουν ακόμα συμφωνήσει να συνεργαστούν, τότε ο πράκτορας παρατηρώντας τις ενέργειες του χρήστη στην επιφάνεια διεπαφής προσπαθεί να αναγνωρίσει τον πρωταρχικό στόχο (ή τους πρωταρχικούς στόχους) του χρήστη. Λόγω του ότι ο χρήστης μπορεί να μην έχει αποφασίσει τι ακριβώς θέλει να κάνει ή να ενεργεί παράλληλα για την επίτευξη παραπάνω τους ενός στόχου ο πράκτορας προσπαθεί ν' αναγνωρίσει όλους τους πιθανούς πρωταρχικούς στόχους του χρήστη ερμηνεύοντας τις ενέργειές του κάτω από διαφορετικά πλαίσια ενεργειών.

Στη περίπτωση της συνεργασίας, ο χρήστης έχει συμφωνήσει με τον πράκτορα να πραγματοποιήσουν από κοινού ένα στόχο και άρα η προσοχή του είναι επικεντρωμένη σε έναν μόνο στόχο. Έτσι ο πράκτορας ερμηνεύει τις ενέργειες του χρήστη κάτω από το πλαίσιο ενεργειών ενός και μόνου κοινού στόχου εντοπίζοντας εύκολα το πότε ο χρήστης ενεργεί λανθασμένα. Παρ' όλα αυτά αν ο χρήστης πραγματοποιήσει μία ενέργεια

που δεν μπορεί να ερμηνευτεί κάτω από το τρέχον πλαίσιο ενεργειών και δεν έρχεται σε σύγκρουση με κάποιον από τους περιορισμούς του πλαισίου ενεργειών ο πράκτορας υποθέτει ότι ο κοινός στόχος εξακολουθεί να ισχύει μέχρι ο χρήστης να εξακολουθήσει να πραγματοποιεί έναν αριθμό ενεργειών που δεν μπορούν ερμηνευτούν στο τρέχον πλαίσιο ενεργειών. Στη τελευταία περίπτωση, ο πράκτορας θεωρεί ότι δεν υπάρχει πλέον κοινός στόχος και προσπαθεί πάλι να αναγνωρίσει το πρωταρχικό στόχο του χρήστη.

Στις παραπάνω δύο περιπτώσεις συμπεριφοράς του πράκτορα πριν και κατά τη διάρκεια της συνεργασίας, υπάρχει ένα άνω επίπεδο ανοχής στον αριθμό των ενεργειών του χρήστη που δεν συνεισφέρουν στην επίτευξη κάποιου πρωταρχικού (πιθανού ή κοινού) στόχου. Ο μέγιστος αυτός αριθμός μπορεί να μεταβληθεί ανάλογα με το πεδίο εφαρμογής και τον τύπο του χρήστη. Στη περίπτωση που ο αριθμός αυτός καθοριστεί να είναι ίσος με το μηδέν (δηλαδή όταν δεν επιτρέπεται ο χρήστης να πραγματοποιεί ενέργειες που δεν συνεισφέρουν σε κάποιο στόχο) τότε και στις δύο παραπάνω περιπτώσεις ο πράκτορας αναπτύσσει ένα και μοναδικό πλαίσιο ενεργειών. Αν ο χρήστης πραγματοποιήσει μία ενέργεια που δεν συνεισφέρει στην επίτευξη του τρέχοντα πρωταρχικού πιθανού στόχου τότε ο στόχος αυτός και το αντίστοιχο πεδίο ενεργειών του αλλάζει. Ενώ στη περίπτωση που ο πράκτορας και ο χρήστης έχουν ένα κοινό στόχο τότε η συνεργασία εγκαταλείπεται και ο πράκτορας προσπαθεί πάλι ν' αναγνωρίσει το πρωταρχικό στόχο του χρήστη.

Καθώς ο αριθμός των μέγιστων επιτρεπτών ενεργειών που δεν συνεισφέρουν στην επίτευξη ενός πιθανού ή κοινού στόχου μεγαλώνει τόσο αυξάνεται και η ανοχή του πράκτορα στις ενέργειες του χρήστη που δεν μπορεί να ερμηνεύσει (δηλαδή δεν μπορεί να καταλάβει το λόγο για τον οποίο ο χρήστης τις πραγματοποιεί). Αυτό έχει ως αποτέλεσμα οι ενέργειες που πραγματοποιεί ο χρήστης και που δεν μπορούν να ερμηνευτούν από τον πράκτορα, να μην θεωρούνται ως λάθη, όπως γίνεται στο πλαίσιο του Collagen [Garland et. al., 2003]. Έτσι στο πλαίσιο εργασίας του ICAGENT ο πράκτορας μπορεί να ερμηνεύει τις ενέργειες του χρήστη κάτω από εναλλακτικά πλαίσια ενεργειών προσπαθώντας να κάνει υποθέσεις για του βασικούς στόχους του χρήστη, χωρίς ο τελευταίος να χρειάζεται να τους δηλώσει με ρητό τρόπο (*exclusively*), ενώ στη περίπτωση που ο χρήστης και ο πράκτορας συμφωνήσουν να πετύχουν από κοινού ένα στόχο, τότε ο πράκτορας μπορεί να εξακολουθεί να μοιράζεται το κοινό στόχο χωρίς να τον ενοχλεί

με άσκοπα μηνύματα λάθους ακόμα και αν ο χρήστης δεν είναι απόλυτα επικεντρωμένος (focused) στην επίτευξη του στόχου αυτού.

## Κεφάλαιο 4ο:

# Το Πλαίσιο Ανάπτυξης Νοημόνων Πρακτόρων ICAGENT

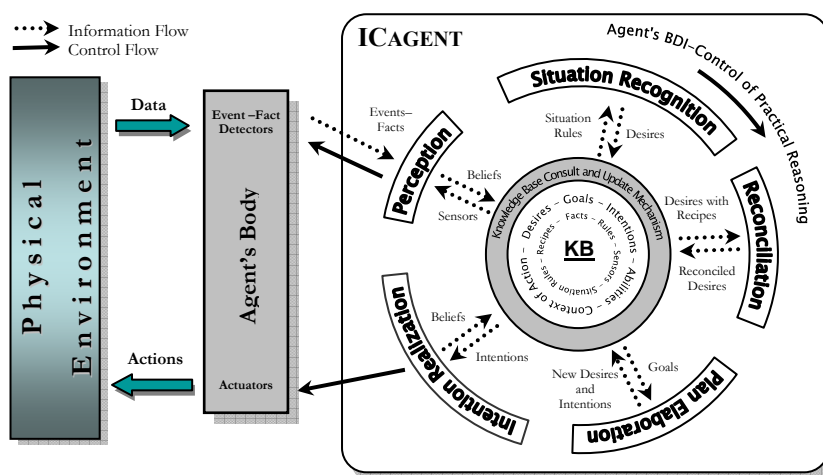
Το πλαίσιο ανάπτυξης νοημόνων πρακτόρων ICAGENT παρέχει τη δυνατότητα κατασκευής πρακτόρων ικανών να συνεργάζονται μεταξύ τους και να προσαρμόζουν τη συμπεριφορά τους ανάλογα με τη κατάσταση του φυσικού τους περιβάλλοντος και τις πληροφορίες που λαμβάνουν από τους άλλους πράκτορες. Η γενική (overall) αρχιτεκτονική του ICAGENT η οποία καθορίζει τη συμπεριφορά του πράκτορα αναπαριστάται στην Εικόνα 23. Αυτή αποτελείται από τρία κύρια μέρη:

- Το Σώμα (Body) του πράκτορα μέσω του οποίου επικοινωνεί με το φυσικό περιβάλλον και δρα σ' αυτό.
- Τη Βάση Γνώσης (Knowledge Base).
- Το μηχανισμό ελέγχου της *BDI-Πρακτικής Συλλογιστικής* (BDI-Control of Practical Reasoning).

Το Σώμα υλοποιεί τους μηχανισμούς δράσης και αντίληψης του πράκτορα με τον έξω κόσμο. Με τους μηχανισμούς αυτούς ο πράκτορας μπορεί ν' αντιλαμβάνεται τις αλλαγές που τελούνται στο φυσικό του περιβάλλον και να υλοποιεί τις προθέσεις του. Λόγω όμως του ότι οι μηχανισμοί αυτοί (που συνθέτουν το σώμα του πράκτορα) εξαρτώνται από την εκάστοτε εφαρμογή, το πλαίσιο εργασίας ICAGENT δεν υποστηρίζει αυτούς τους μηχανισμούς (όπως επίσης και τους μηχανισμούς με τους οποίους το φυσικό περιβάλλον αλλάζει).

Ανεξάρτητα του πεδίου εφαρμογής το σώμα περιλαμβάνει: (α) αισθητήρες εντοπισμού γεγονότων και συμβάντων (fact – event detectors) με τους οποίους ο πράκτορας μπορεί να εντοπίζει συγκεκριμένου τύπου γεγονότα και ενέργειες άλλων πρακτόρων που συμβαίνουν στο φυσικό του περιβάλλον, και (β) μηχανισμούς δράσης (actuators) οι

οποίοι υλοποιούν τις βασικές ενέργειες (basic level actions) που ο πράκτορας προτίθεται να πραγματοποιήσει. Οι αισθητήρες καθώς και οι μηχανισμοί δράσης μπορούν να υλοποιηθούν σε οποιαδήποτε γλώσσα προγραμματισμού με την προϋπόθεση ύπαρξης μίας διεπαφής μεταξύ της διεργασίας της αντίληψης (Perception module) του ICAGENT και των αισθητήρων του καθώς και μεταξύ της διεργασίας υλοποίησης των προθέσεων (Intention Realization module) και των μηχανισμών δράσης.



Εικόνα 23: Η γενική αρχιτεκτονική του ICAGENT.

Ο μηχανισμός ελέγχου της *BDI-Πρακτικής Συλλογιστικής* (BDI-CPR) του ICAGENT είναι υλοποιημένος στη δηλωτική πολυνηματική γλώσσα προγραμματισμού SWI-Prolog αποτελείται από πέντε μονάδες (modules) όπου η κάθε μία από αυτές υλοποιεί μία διεργασία συλλογισμού (reasoning task). Οι διεργασίες αυτές τρέχουν παράλληλα και δεν επικοινωνούν μεταξύ τους παρά μόνο με τη βάση γνώσης. Πιο συγκεκριμένα αυτές είναι: (α) η διεργασία της *αντίληψης* νέων δεδομένων από το φυσικό περιβάλλον του πράκτορα (perception), (β) η *αναγνώριση νέων νοητικών καταστάσεων* οι οποίες μπορούν να οδηγήσουν τον πράκτορα σε πρωτοβουλίες δράσης (situation recognition), (γ) η *αναγνώριση και επίλυση συγκρούσεων* μεταξύ ενεργειών του πράκτορα ή/και μεταξύ των προϋποθέσεων εφαρμογής των ενεργειών και της νοητικής κατάστασης του πράκτορα (reconciliation), (δ) η διεργασία συλλογισμού *εύρεσης αρχικού πλάνου για μία ενέργεια και η περαιτέρω ανάπτυξη του πλάνου αυτής* (plan elaboration) και τέλος (ε) η διεργασία συλλογισμού *πραγματοποίησης των προθέσεων* του πράκτορα, δηλαδή της

υλοποίησης των ενεργειών που ο πράκτορας έχει τη πρόθεση να πραγματοποιήσει (intention realization).

#### 4.1. Η βάση γνώσης

Η βάση γνώσης περιλαμβάνει τη νοητική κατάσταση του πράκτορα, δηλαδή τις τρέχουσες επιθυμίες του, τους στόχους του, τις προθέσεις του και τις πεποιθήσεις του πράκτορα. Οι πεποιθήσεις αφορούν τη γνώση του πράκτορα σχετικά με τη κατάσταση του φυσικού του περιβάλλοντος (environmental facts), τις ενέργειες που πραγματοποιούνται σε αυτό (events), τις μεθόδους εκτέλεσης ενεργειών (recipes), τους κανόνες με τους οποίους ο πράκτορας ενημερώνει τη βάση γνώσης του και εξάγει νέα γνώση (update and consult rules), καθώς επίσης και τους κανόνες αναγνώρισης νέων καταστάσεων για δράση (situation recognition rules). Η γνώση του πράκτορα ενημερώνεται και χρησιμοποιείται από τις διεργασίες συλλογισμού μέσω του μηχανισμού *ενημέρωσης και εξαγωγής γνώσης* που εμπεριέχεται στη βάση γνώσης (Knowledge Base Consult and Update Mechanism).

Νοητική Δομή	Σύντομη Περιγραφή
<code>&lt;bel&gt; ::= bel(State)</code>	Μία πεποίθηση αναπαριστά τη γνώση του πράκτορα για τη κατάσταση του φυσικού του περιβάλλοντος όσο και για την εσωτερική του κατάσταση.
<code>&lt;desire_to&gt; ::= desire_to(Agent, Action)</code>	Η δομή μίας επιθυμίας αναπαριστά την επιλογή του πράκτορα να πραγματοποιήσει μία ενέργεια (ή να πετύχει μία κατάσταση). Ο πράκτορας μπορεί να έχει επιθυμίες οι οποίες να βρίσκονται σε σύγκρουση ή γενικότερα να μην μπορούν να πραγματοποιηθούν στο τρέχον πλαίσιο ενεργειών.
<code>&lt;goal_to&gt; ::= goal_to(Agent, Action)</code>	Ένας στόχος αναπαριστά τη νοητική κατάσταση όπου ο πράκτορας γνωρίζει τουλάχιστον ένα μερικός ανεπτυγμένο πλάνο για την ενέργεια που επιθυμεί να πραγματοποιήσει και πιστεύει ότι αυτό είναι πραγματοποιήσιμο (εφαρμόσιμο) χωρίς αυτό να έρχεται σε σύγκρουση με άλλους στόχους του ή προθέσεις.
<code>&lt;int_to&gt; ::= int_to(Agent, Action)</code>	Η δομή <code>&lt;int_to&gt;</code> αναπαριστά τη δέσμευση του πράκτορα να πραγματοποιήσει μία ενέργεια. Για να έχει ο πράκτορας το τύπο της πρόθεσης αυτής θα πρέπει να γνωρίζει τουλάχιστον ένα μερικός ανεπτυγμένο και πραγματοποιήσιμο πλάνο για την ενέργεια αυτή, να έχει δεσμευτεί να το αναπτύξει περαιτέρω και να γνωρίζει ένα πλήρες πλάνο για το πώς θα το κάνει αυτό [Grosz and Kraus, 1996].
<code>&lt;int_that_do&gt; ::= int_that(Group, Action, Recipe)</code>	Η νοητική <code>&lt;int_that_do&gt;</code> αναπαριστά τη πρόθεση το πράκτορα να πραγματοποιήσει την ενέργεια Action σε συνεργασία με μία ομάδα πρακτόρων Group ακολουθώντας τη κοινή μέθοδο Recipe.
<code>&lt;int_that_cba&gt; ::=</code>	Η νοητική <code>&lt;int_that_cba&gt;</code> αναπαριστά τη δέσμευση του πράκτορα να πραγματοποιηθεί επιτυχώς η ενέργεια Action είτε από τον ίδιο το



<code>int_that (Agent, cba (Group, Action, Recipe))</code>	πράκτορα (αν το <code>Group</code> περιλαμβάνει μόνο τον εαυτό του) είτε από μία ομάδα πρακτόρων <code>Group</code> ακολουθώντας από κοινού τη μέθοδο <code>Recipe</code> .
--	---

Πίνακας 2: Οι νοητικές δομές στο πλαίσιο ανάπτυξης πρακτόρων ICAGENT.

Στις επόμενες τρεις υπό-παραγράφους παρουσιάζουμε τη σημασία των νοητικών δομών που συνθέτουν τη νοητική κατάσταση του πράκτορα (βλ. Πίνακας 2), τη δομή των μεθόδων (`recipes`) που ο πράκτορας χρησιμοποιεί για να πραγματοποιήσει μόνος του ή σε συνεργασία με άλλους πράκτορες τις ενέργειες που θέλει να κάνει, καθώς επίσης και τον τρόπο με τον οποίο η βάση γνώσης ενημερώνεται και χρησιμοποιείται από τον πράκτορα.

#### 4.1.1. Νοητικές δομές

Στο μοντέλο των προθετικών πρακτόρων (BDI model of agency) η νοητική κατάσταση ενός πράκτορα περιγράφεται μέσω τεσσάρων νοητικών δομών: της πεποίθησης, της επιθυμίας, του στόχου, της πρόθεσης να πραγματοποιήσει μία ενέργεια (`intent to`) και της πρόθεσης να ισχύσει μία κατάσταση (`intent that`).

Η νοητική δομή της πεποίθησης (`belief`) αναπαριστά τη γνώση του πράκτορα για τη κατάσταση του φυσικού περιβάλλοντός του ή τη γνώση που έχει για τη νοητική του κατάσταση (αυτογνωσία) ή για το τί πιστεύει κάποιος άλλος πράκτορας. Στο πλαίσιο ανάπτυξης ICAGENT η νοητική δομή της πεποίθησης αναπαρίσταται με γεγονότα της Prolog ως εξής:

$$\langle \text{bel} \rangle ::= \text{bel}(\text{Agent}, \text{State}, C)$$

Στο παραπάνω γεγονός, `Agent` είναι το `id` του πράκτορα που έχει τη πεποίθηση (για απλούστευση, από εδώ και πέρα όποτε παραλείπουμε το `id` του πράκτορα θα θεωρούμε ότι η πεποίθηση αναφέρεται σε ένα γεγονός που ο ίδιος ο πράκτορας πιστεύει). Το όρισμα `c` είναι ένας δυαδικός αριθμός ο οποίος αναπαριστά τη γνώση (`c=1`) ή την άγνοια (`c=0`) του πράκτορα σχετικά με την ύπαρξη της κατάστασης `State`. Διακρίνουμε μεταξύ τριών πιθανών τύπων πεποιθήσεων που μπορεί να κατέχει ένας πράκτορας: (α) ο πράκτορας έχει τη πεποίθηση ότι η κατάσταση `State` ισχύει, δηλαδή `bel(State, 1)`, (β) ο πράκτορας έχει τη πεποίθηση ότι η κατάσταση δεν ισχύει, δηλαδή `bel(not State, 1)` και (γ) ο πράκτορας έχει τη πεποίθηση ότι η κατάσταση ούτε ισχύει ούτε δεν ισχύει (δηλαδή έχει άγνοια γ' αυτή), δηλαδή `bel(State, 0)`. Το τελευταίο είναι ισοδύ-

ναμο με το να μην έχει ο πράκτορας καμία πεποίθηση σχετικά με την ύπαρξη ή όχι της κατάστασης  $State$ <sup>15</sup>.

Ο υπολογισμός της τιμής αληθείας μίας πεποίθησης ανάλογα με τη τιμή του  $c$  βασίζεται στη 3-τιμή λογική του Kleene [Turner, 1984] και περιγράφεται στην υπό-παράγραφο 4.1.3. Για απλούστευση, στο υπόλοιπο της διατριβής όταν η μεταβλητή  $C$  παίρνει την τιμή 1 θα γράφουμε  $bel(State)$  αντί του  $bel(State, 1)$ . Σε κάθε άλλη περίπτωση, όταν αναφερόμαστε στην άγνοια του πράκτορα για την ύπαρξη μίας κατάστασης θα γράφουμε  $bel(State, 0)$ .

Για παράδειγμα, στο περιβάλλον του *Tileworld* αν ο πράκτορας έχει τη πεποίθηση ότι στη θέση (19,11) υπάρχει μία τρύπα τότε το γεγονός  $bel(hole(19, 11))$  αληθεύει. Αν η πεποίθηση  $bel(hole(25, 28), 0)$  αληθεύει τότε αυτό σημαίνει ότι ο πράκτορας δεν έχει καμία πληροφορία για το αν υπάρχει ή όχι τρύπα στη θέση (25,28). Αντίθετα, αν έχει στη βάση γνώσης του τις πεποιθήσεις  $bel(hole(25, 28))$  και  $bel(not\ empty(hole(25, 28)))$  τότε ο πράκτορας πιστεύει ότι υπάρχει τρύπα στη θέση (25,28) και ότι δεν είναι άδεια.

Αναγκαία προϋπόθεση για μία επιτυχή συνεργασία μεταξύ πρακτόρων μίας ομάδας βασιζόμενων στο μοντέλο των *SharedPlans* είναι η ύπαρξη της αμοιβαίας πεποίθησης [Grosz and Kraus, 1999]. Η δομή της αμοιβαίας πεποίθησης αναπαριστά τη γνώση του πράκτορα για ένα γεγονός το οποίο πιστεύει ότι ισχύει και ότι πιστεύει ότι οι άλλοι πράκτορες της ομάδας πιστεύουν ότι ισχύει καθώς και ότι πιστεύει ότι οι άλλοι πράκτορες πιστεύουν ότι ο πράκτορας πιστεύει ότι ισχύει και συνεχίζοντας έτσι επ' άπειρο οι πράκτορες μοιράζονται τη γνώση ότι μία κατάσταση ισχύει [Wooldridge, 2000]. Μία πεποίθηση της μορφής αυτής δημιουργείται όταν ένας πράκτορας επικοινωνεί την ύπαρξη ενός γεγονότος σε μία ομάδα πρακτόρων. Η αμοιβαία πεποίθηση (*mutual belief*) αναπαρίσταται ως ακολούθως:

$$\langle mb \rangle ::= mb(Group, State)$$

<sup>15</sup> Δηλαδή αν στη Prolog τα  $not(bel(State, 1))$  και  $not(bel(not\ State, 1))$  είναι ταυτόχρονα ψευδοί.

όπου `Group` είναι η ομάδα, τα μέλη της οποίας αμοιβαία πιστεύουν ότι η κατάσταση `State` ισχύει.

Στο πλαίσιο εργασίας ICAGENT αμοιβαίες πεποιθήσεις δημιουργούνται κατά τη διάρκεια της συνεργασίας των πρακτόρων και πιο συγκεκριμένα (όπως θα δούμε στο Κεφάλαιο 5), κατά τη διάρκεια του καθορισμού της ομάδας (`group formation`), της συνεννόησης για τη μέθοδο εκτέλεσης μιας ενέργειας (`recipe negotiation`) καθώς και κατά τη διάρκεια της ανάθεσης των ενεργειών στους συνεργάτες (`task allocation`).

Όπως τονίσαμε παραπάνω, η δεύτερη νοητική δομή που συνθέτει τη νοητική κατάσταση ενός πράκτορα είναι αυτή της επιθυμίας (`desire`). Μία επιθυμία αναπαριστά την επιλογή του πράκτορα να πραγματοποιήσει μία ενέργεια (ή να πετύχει μία κατάσταση)<sup>16</sup>. Ο πράκτορας μπορεί να έχει επιθυμίες οι οποίες να βρίσκονται σε σύγκρουση ή γενικότερα να μην μπορούν να πραγματοποιηθούν στο τρέχον πλαίσιο ενεργειών (για παράδειγμα να μην έχει ο πράκτορας τη δυνατότητα να τις πραγματοποιήσει). Η επιθυμία ενός πράκτορα να πραγματοποιήσει μία ενέργεια αναπαριστάται με την ακόλουθη δομή:

```
<desire_to> ::= desire_to(Agent, Action)
```

Η τρίτη νοητική δομή αφορά τους στόχους του πράκτορα. Ακολουθώντας τη θεωρία των Cohen-Levesque (1990), οι *στόχοι* ενός πράκτορα ορίζονται ως το σύνολο των επιθυμιών του που έχουν περάσει από τη διεργασία αναγνώρισης και επίλυσης συγκρούσεων. Ένας στόχος (`goal`) αναπαριστά τη νοητική κατάσταση του πράκτορα στην οποία: (α) έχει βρει μία σχετική μέθοδο (μερικώς ανεπτυγμένο πλάνο) για την ενέργεια που επιθυμεί να πραγματοποιήσει και (β) πιστεύει ότι η πραγματοποίησή της ενέργειας είναι εφικτή, δηλαδή ότι η μέθοδος εκτέλεσης της ενέργειας δεν έρχεται σε σύγκρουση με τις άλλες ενέργειες που έχει δεσμευθεί να πραγματοποιήσει (μόνος του ή σε συνεργασία με άλλους πράκτορες) και επιπλέον πιστεύει ότι οι νοητικές συνθήκες που εκφράζουν τις προϋποθέσεις εφαρμογής της μεθόδου (`applicability mental conditions`) ισχύουν. Ο παραπάνω ορισμός ισχύει μόνο όταν ο πράκτορας στοχάζεται ατομικά ή

κοινωνικά (individually or socially deliberatively).

Όπως θα διαπιστώσουμε στις ακόλουθες παραγράφους η (β) προϋπόθεση δεν ισχύει όταν ο πράκτορας συμπεριφέρεται αντανακλαστικά (reactively). Σε αυτή τη περίπτωση ο στόχος μπορεί να βρίσκεται σε σύγκρουση με άλλους στόχους ή προθέσεις που μπορεί να έχει εκείνη τη στιγμή ο πράκτορας. Ένας στόχος αναπαριστάται με την ακόλουθη δομή:

$$\langle \text{goal\_to} \rangle ::= \text{goal\_to}(\text{Agent}, \text{Action})$$

Η τέταρτη νοητική δομή του πλαισίου αναφέρεται στις προθέσεις του πράκτορα. Σύμφωνα με το μοντέλο συνεργασίας *SharedPlans* της Grosz (1996), για να έχει ένας πράκτορας τη *πρόθεση* (intention to) να πραγματοποιήσει μία ενέργεια, πρέπει να γνωρίζει τουλάχιστον ένα μερικό πλάνο γ' αυτή, πρέπει να έχει δεσμευθεί ότι θ' αναπτύξει περαιτέρω το πλάνο αυτής και να γνωρίζει έναν τρόπο για να το αναπτύξει. Όταν ο πράκτορας συμπεριφέρεται στοχαστικά, οι προθέσεις του είναι πάντοτε εφαρμόσιμες διότι έχει ελέγξει για το αν μπορούν να πραγματοποιηθούν. Για τις ενέργειες όμως που έχει «επιλέξει» να πραγματοποιήσει αντανακλαστικά δεν ισχύει κάτι τέτοιο. Η πρόθεση ενός πράκτορα να πραγματοποιήσει μία ενέργεια αναπαρίσταται με τη νοητική δομή:

$$\langle \text{int\_to} \rangle ::= \text{int\_to}(\text{Agent}, \text{Action})$$

Τέλος, η πρόθεση ενός πράκτορα να πετύχει ή να διατηρήσει (αν ήδη ισχύει) μία κατάσταση (intent that) υπό το πλαίσιο ενεργειών  $\text{IC}$ <sup>17</sup> αναπαρίσταται με την ακόλουθη νοητική δομή:

$$\langle \text{int\_that} \rangle ::= \text{int\_that}(\text{Agent}, \text{State}, \text{IC})$$

Όπως αναφέραμε στην ενότητα 3.2, τόσο η νοητική δομή  $\langle \text{int\_to} \rangle$  όσο και η  $\langle \text{int\_that} \rangle$  δεσμεύουν τον πράκτορα ως προς τη μη υιοθέτηση αντικρουόμενων ενεργειών (και γενικότερα προθέσεων που δεν μπορούν να πραγματοποιηθούν ταυτόχρονα)

<sup>16</sup> Θεωρούμε ότι η επίτευξη μίας κατάστασης  $s$  μπορεί να επιτευχθεί από μία ενέργεια της μορφής *achieve*( $s$ ) [Grosz, 1996].

<sup>17</sup> Το όρισμα  $\text{IC}$  αναπαριστά το  $\text{id}$  του πλαισίου ενεργειών κάτω από το οποίο ο πράκτορας έχει τη πρόθεση να πετύχει ή να διατηρήσει τη κατάσταση *State*.

και οδηγούν τον πράκτορα σε επανασχεδιασμό των πλάνων του σε περίπτωση αποτυχίας. Η ειδοποιός διαφορά τους έγκειται όχι στο τύπο των αντικειμένων που η κάθε μία σχετίζεται<sup>18</sup>, αλλά στο τρόπο ανάλυσης μέσων και σκοπών που η κάθε μία οδηγεί τον πράκτορα<sup>19</sup> καθώς και στις διαφορετικές προϋποθέσεις που πρέπει να ισχύουν για να έχει ένας πράκτορας τις δομές αυτές.

Για να έχει ένας πράκτορας τη πρόθεση να πετύχει ή να διατηρήσει μία κατάσταση πρέπει να ισχύουν τουλάχιστον δύο προϋποθέσεις (αξιώματα) [Grosz and Kraus, 1999]: (α) Ο πράκτορας θα πρέπει να μην πιστεύει ότι η κατάσταση δεν είναι δυνατό να επιτευχθεί (δηλαδή ότι ισχύει  $\text{not}(\text{bel}(\text{not State}, 1))$ )<sup>20</sup> και (β) ότι αν ο πράκτορας πιστεύει ότι η προτιθέμενη κατάσταση είναι δυνατή αλλά δεν έχει γνώση για το αν ισχύει  $(\text{bel}(\text{State}, 0))$  και γνωρίζει μία μέθοδο που μπορεί να τη κάνει να ισχύσει τότε ο πράκτορας πρέπει να προσπαθήσει να πραγματοποιήσει τη μέθοδο αυτή. Τα παραπάνω αξιώματα παρουσιάζονται και αναλύονται περισσότερο στην υποενότητα 4.2.3.

Στο πλαίσιο εργασίας ICAGENT χρησιμοποιούμε τους παρακάτω τύπους  $\langle \text{int\_that} \rangle$ :

- Δέσμευση ομάδας για κοινή δράση (group activity commitment):

$\langle \text{int\_that\_do} \rangle ::= \text{int\_that}(\text{Agent}, \text{do}(\text{Group}, \text{Action}, \text{Recipe}))$

- Βασική πρόθεση (core case commitment):

$\langle \text{int\_that\_cba} \rangle ::= \text{int\_that}(\text{Agent}, \text{cba}(\text{Group}, \text{Action}, \text{Recipe}))$

Στους παραπάνω δύο τύπους η παράμετρος  $\text{Agent}$  αναπαριστά τον πράκτορα ο οποίος δεσμεύεται ότι η ομάδα πρακτόρων ( $\text{Group}$ ) θα πραγματοποιήσει την ενέργεια  $\text{Action}$  ακολουθώντας τη μέθοδο  $\text{Recipe}$  υπό τις συνθήκες του πλαισίου της ενέργειας  $\text{Action}$ . Οι διαφορές των δύο προθέσεων έγκειται στο ρόλο που αυτές παίζουν κατά το σχεδιασμό των κοινών πλάνων που πραγματοποιούν οι πράκτορες.

<sup>18</sup> Η πρώτη σχετίζεται με ενέργειες ενώ η δεύτερη με καταστάσεις.

<sup>19</sup> Δηλαδή, στο πλαίσιο εργασίας ICAGENT με τη διεργασία συλλογισμού εντοπισμού και επίλυσης συγκρούσεων.

<sup>20</sup> Είτε να το συνάγει μέσω κάποιου κανόνα είτε διότι η γνώση αυτή δεν υπάρχει στη βάση γνώσης.

Η συνεργασία μεταξύ πρακτόρων για μία ενέργεια αρχίζει (initiated) όταν ένας πράκτορας σχηματίζει την δέσμευση να πραγματοποιήσει την ενέργεια αυτή συνεργατικά, δηλαδή όταν ένας πράκτορας σχηματίζει μία δέσμευση κοινής δράσης (<int\_that\_do>). Μία πρόθεση αυτού του τύπου σχηματίζεται σε τρεις περιπτώσεις: (α) Όταν ο πράκτορας πιστεύει ότι είναι επιθυμητό να συνεργαστεί με άλλους πράκτορες προκειμένου να υλοποιήσει τη μέθοδο που έχει επιλέξει για την ενέργεια αυτή (για παράδειγμα μπορεί να πιστεύει ότι η συνεργασία μπορεί να επιταχύνει την εκτέλεση της ενέργειας), (β) κατά τη διάρκεια επίλυσης μίας σύγκρουσης (για παράδειγμα αν κατά τη διάρκεια της συλλογιστικής διεργασίας εντοπισμού και επίλυσης συγκρούσεων αναγνωρίσει ότι δεν μπορεί να πραγματοποιήσει μία ενέργεια μόνος του παρά μόνο σε συνεργασία με άλλους πράκτορες) και τέλος (γ) όταν δέχεται μία αίτηση από άλλον πράκτορα (ή άλλους πράκτορες) για συνεργασία σε μία ενέργεια.

Στο Κεφάλαιο 5 θα δούμε αναλυτικότερα πως η χρήση αυτού του τύπου δέσμευσης επηρεάζει τις αποφάσεις των πρακτόρων κατά τη διάρκεια του σχεδιασμού και εκτέλεσης των κοινών πλάνων τους καθώς επίσης και πως τους οδηγεί σε συνεργατική συμπεριφορά.

Η πρόθεση ενός πράκτορα ότι η ομάδα ή ο ίδιος μπορεί να πραγματοποιήσει μία ενέργεια αναπαρίσταται από τη νοητική δομή <int\_that\_cba>. Η δέσμευση αυτή σχηματίζεται σε δύο περιπτώσεις: (α) όταν συμφωνεί για το ποιοι πράκτορες θ' αποτελέσουν την ομάδα και (β) όταν αποφασίζει να μην ενεργήσει αντανακλαστικά. Στη πρώτη περίπτωση ένας πράκτορας της ομάδας σχηματίζει τη δέσμευση <int\_that\_cba> για να «προστατέψει» τις ενέργειες των συνεργατών του από ενέργειες που πιθανώς ο ίδιος πραγματοποιήσει. Στη δεύτερη περίπτωση ο πράκτορας δημιουργεί τη δέσμευση αυτή για να «προστατέψει» τις ενέργειες που προτίθεται να πραγματοποιήσει ο ίδιος είτε μόνος του είτε σε συνεργασία με άλλους πράκτορες. Σε κάθε περίπτωση οι δεσμεύσεις της μορφής <int\_that\_cba> έχουν ως αποτέλεσμα την ενεργοποίηση της συλλογιστικής διεργασίας εντοπισμού και επίλυσης συγκρούσεων.

Στο πλαίσιο ανάπτυξης πρακτόρων ICAGENT οι υπόλοιποι από τους τύπους προθέσεων <int\_that> που παρουσιάσαμε στην υποενότητα 3.2.2 δεν αναπαριστώνται με αποκλειστικό τρόπο αλλά υλοποιούνται μέσω συλλογιστικών διεργασιών και νοητικών ενεργειών (όπως θα δούμε και στην ενότητα 4.2).

Πιο συγκεκριμένα, ο καθορισμός των παραμέτρων (parameter identification) μίας μεθόδου υλοποιείται από τις νοητικές ενέργειες επιλογής σχετικής μεθόδου (relevant recipe mental action) και ανάπτυξης μεθόδου (plan elaboration mental action) (βλ. υποενότητα 4.2.4). Κατά τη διάρκεια των δύο αυτών νοητικών ενεργειών καθορίζονται παράμετροι που αφορούν τις υπό-ενέργειες και τα αποτελέσματα (effects) της μεθόδου, τα μέλη της ομάδας που πιθανόν ο πράκτορας θα συνεργαστεί (στη περίπτωση που η ενέργεια πρέπει να πραγματοποιηθεί σε συνεργασία με άλλους πράκτορες), καθώς και ο καταμερισμός των υπό-ενεργειών σε αυτούς<sup>21</sup>.

Η ικανοποίηση των περιορισμών (constraint satisfaction) μίας μεθόδου υλοποιείται μέσω της συλλογιστικής διεργασίας εντοπισμού και επίλυσης διενέξεων. Κάθε πράκτορας αφού επιλέξει μία σχετική μέθοδο για τη κοινή ενέργεια (και μόνο αν έχει αποφασίσει να συμπεριφερθεί στοχαστικά) εξετάζει αν έχει την δυνατότητα να τη πραγματοποιήσει. Παρά το γεγονός ότι η μέθοδος μπορεί να είναι κοινή για όλους τους πράκτορες της ομάδας (δηλαδή οι υπό-ενέργειες που αποτελούν τη μέθοδο) οι περιορισμοί που τη συνοδεύουν μπορεί να είναι διαφορετικοί για κάθε πράκτορα. Για παράδειγμα, στο Tileworld όταν οι πράκτορες συνεργάζονται για να γεμίσουν μία τρύπα (και πιο συγκεκριμένα στη περίπτωση που ο ένας πράκτορας προσπαθεί να σηκώσει τον κύβο, ενώ ο δεύτερος προσπαθεί να τον βοηθήσει φέρνοντας τον κύβο πιο κοντά σε αυτόν), δηλαδή όταν η κοινή τους ενέργεια είναι η `fill_hole`, τότε οι συνθήκες εφαρμογής της κοινής μεθόδου που επιλέγουν διαφέρουν μεταξύ τους. Ο πρώτος πράκτορας που προσπαθεί να σηκώσει τον κύβο για τον ρίξει στη τρύπα «ενδιαφέρεται» να υπάρχει ένα μονοπάτι προς τον κύβο, να υπάρχει ο κύβος καθώς και η τρύπα που έχει επιλέξει να είναι συνεχώς κενή, ενώ ο δεύτερος πράκτορας «ενδιαφέρεται» μόνο να υπάρχει ο κύβος και ένα μονοπάτι προς αυτόν.

Τέλος, κοινές πεποιθήσεις δημιουργούνται όταν οι πράκτορες επικοινωνούν τη δυνατότητά τους να συνεργαστούν, όταν προσπαθούν να σχηματίσουν ομάδα, όταν διαπραγματεύονται τη μέθοδο που θ' ακολουθήσουν καθώς και όταν διαπραγματεύονται

---

<sup>21</sup> Στην υποενότητα που ακολουθεί περιγράφουμε όλες τις παραμέτρους που έχει μία μέθοδο.

το πώς θα μοιραστούν οι ενέργειες μεταξύ τους. Αυτές οι νοητικές ενέργειες παρουσιάζονται και αναλύονται περαιτέρω στο Κεφάλαιο 5.

#### 4.1.2. Αναπαράσταση των πλάνων

Μία μέθοδος πραγματοποίησης μίας ενέργειας ή επίτευξης μίας κατάστασης (recipe) αναπαριστά τη γνώση του πράκτορα για τον τρόπο με τον οποίο μπορεί να πραγματοποιήσει μία ενέργεια ή να πετύχει μία κατάσταση μόνος του ή σε συνεργασία με άλλους πράκτορες. Το σύνολο όλων των μεθόδων που γνωρίζει ένας πράκτορας συνθέτει τη βιβλιοθήκη μεθόδων εκτέλεσης ενεργειών (recipe library). Η δομή μίας μεθόδου είναι η ακόλουθη (η BNF-περιγραφή της παρουσιάζεται στο Κεφάλαιο 8):

```
<recipe_lib> ::= recipe( Action, Recipe_Goal,
                        RecipeId,
                        Relevance_Mental_Condition,
                        Behaviour_Directives,
                        Collaboration_Mental_Condition,
                        Task_Allocation,
                        Applicability_Mental_Condition,
                        Action_List, Effects )
```

Το αναγνωριστικό της μεθόδου (RecipeId) είναι ένας αριθμός (id) ο οποίος χαρακτηρίζει μοναδικά τη μέθοδο.

Ο στόχος της μεθόδου (Recipe\_Goal) αναπαριστά τη κατάσταση του φυσικού περιβάλλοντος ή τη νοητική κατάσταση που ο πράκτορας μπορεί να πετύχει χρησιμοποιώντας τη μέθοδο αυτή. Κάθε μέθοδος συνδέεται με ακριβώς έναν στόχο ο οποίος επιτυγχάνεται όταν ο πράκτορας εκτελέσει επιτυχώς τις υπό-ενέργειες της μεθόδου της Action<sup>22</sup>. Ο στόχος της μεθόδου οδηγεί το πράκτορα σε μία συμπεριφορά προσανατολισμένη στις καταστάσεις που θέλει να πετύχει (goal directed behaviour). Αυτό αποτρέπει τον πράκτορα από το να πραγματοποιεί ενέργειες οι οποίες δεν συνεισφέρουν στους

---

<sup>22</sup> Η επιτυχής πραγματοποίηση των υποενεργειών μίας μεθόδου είναι ικανή αλλά όχι αναγκαία για την επίτευξη του στόχου μίας μεθόδου. Διαδικασίες που συμβαίνουν στο φυσικό περιβάλλον του πράκτορα (π.χ. ενέργειες άλλων πρακτόρων) μπορεί να επισπεύσουν την επίτευξη του στόχου πριν ο πράκτορας πραγματοποιήσει όλες τις υπό-ενέργειες της μεθόδου.



στόχους του (over-acting) (δηλαδή να πραγματοποιεί ενέργειες που τα αποτελέσματά τους (effects) να μην είναι θεμιτά) και που μπορεί να περιορίζουν περαιτέρω τους μελλοντικούς του στόχους ή να μειώνουν την ανταπόκρισή του στις αλλαγές του περιβάλλοντος πραγματοποιώντας ενέργειες οι οποίες δεν είναι απαραίτητες. Η συμπεριφορά προσανατολισμένη στους στόχους επιτυγχάνεται με δύο τρόπους. Ο πρώτος είναι χρησιμοποιώντας την ενέργεια `achieve(State)`. Για την ενέργεια αυτή ο πράκτορας προσπαθεί να βρει μία σχετική μέθοδο για τη κατάσταση (`State`) που προσπαθεί να πετύχει (επιλέγοντας τη μέθοδο που έχει ως στόχο το `State` αυτό) και όχι για την ενέργεια `achieve(State)`. Ο δεύτερος τρόπος είναι μέσω των κατευθύνσεων συμπεριφοράς οι οποίες επιτρέπουν κατά τη διάρκεια πραγματοποίησης μία ενέργειας ο πράκτορας να ελέγχει αν ο στόχος της μεθόδου έχει πραγματοποιηθεί. Με αυτό τον τρόπο ο πράκτορας μπορεί να σταματήσει τη πραγματοποίηση μίας ενέργειας πριν πραγματοποιήσει όλες τις υπό-ενέργειες της μεθόδου. Σε αντίθετη περίπτωση ο πράκτορας πραγματοποιεί ενέργειες οι οποίες δεν συνεισφέρουν στο στόχο (overacting) αφού αυτός έχει ήδη επιτευχθεί.

*Η νοητική συνθήκη σχετικότητας της μεθόδου* (`Relevance_Mental_Condition`) είναι μία λογική πρόταση η οποία σχηματίζεται από οποιονδήποτε συνδυασμό πεποιθήσεων του πράκτορα χωρισμένες με λογικούς τελεστές της μορφής "not", "and" και "or"<sup>23</sup>. Η συνθήκη αυτή καθορίζει τη σχετικότητα της μεθόδου στο τρέχον πλαίσιο ενεργειών στο οποίο αυτή εκτελείται. Όπως αναφέραμε στην υποενότητα 4.1.1, όποτε ο πράκτορας ελέγχει τη συνθήκη αυτή, έχει τη δυνατότητα να καθορίσει τις παραμέτρους (parameter identification) των άλλων ορισμάτων της μεθόδου (μέσω ενοποίησης μεταβλητών). Πιο συγκεκριμένα μπορεί να καθορίσει τις παραμέτρους που αφορούν τις κατευθύνσεις εντοπισμού και επίλυσης συγκρούσεων της μεθόδου, τις παραμέτρους που αφορούν τη γνώση του για το αν θα πρέπει να πραγματοποιήσει την ενέργεια σε συνεργασία με άλλους πράκτορες ή μόνος του, τις παραμέτρους που αφορούν τη γνώση του πράκτορα σχετικά με το ποιους πράκτορες πρέπει να συνεργαστεί (στη περίπτωση που αποφασί-

---

<sup>23</sup> Από το σημείο αυτό και στο υπόλοιπο της εργασίας, οποτεδήποτε αναφέρουμε μία πρόταση ως νοητική συνθήκη θα εννοούμε μία λογική πρόταση η οποία σχηματίζεται από οποιονδήποτε συνδυασμό πεποιθήσεων χρησιμοποιώντας του τελεστές "not", "and" και "or".

σει να συνεργαστεί), τις παραμέτρους που αφορούν την εκτέλεση της μεθόδου (για παράδειγμα να καθορίσει κάποιες από τις υπο-ενέργειες που συνθέτουν τη μέθοδο ή κάποιες από τις μεταβλητές αυτών) και τέλος μπορεί να καθορίσει τα αποτελέσματα (effects) της ενέργειας ή κάποιες παραμέτρους (μεταβλητές που δεν έχουν πάρει τιμή) αυτών.

Οι κατευθύνσεις συμπεριφοράς (Behaviour\_Directives) είναι μία λίστα με εντολές οι οποίες «καθοδηγούν» τον πράκτορα κατά τη διάρκεια εκτέλεσης της συλλογιστικής διεργασίας του εντοπισμού και επίλυσης συγκρούσεων (reconciliation). Οι κατευθύνσεις συμπεριφοράς κάθε μεθόδου μπορεί είτε να είναι προκαθορισμένες από τον προγραμματιστή κατά τη διάρκεια σχεδιασμού των μεθόδων είτε να καθορίζονται τη στιγμή που ο πράκτορας επιλέγει τη μέθοδο και ελέγχει τη νοητική συνθήκη `Relevance_Mental_Condition` (parameter identification). Στην τελευταία περίπτωση, οι κατευθύνσεις συμπεριφοράς εξαρτώνται από τις πεποιθήσεις που έχει εκείνη τη στιγμή ο πράκτορας σχετικά με την κατάσταση του περιβάλλοντός του (φυσικού και νοητικού). Στο πλαίσιο εργασίας ICAGENT, στοχεύοντας να υποστηρίξουμε τη δυνατότητα των πρακτόρων να προσαρμόζουν τη συμπεριφορά τους ανάλογα με τη κατάσταση του περιβάλλοντός τους, παρέχουμε έναν μεγάλο αριθμό τέτοιου είδους κατευθύνσεων (βλ. Παράρτημα Α). Κάθε κατεύθυνση συμπεριφοράς αναπαρίσταται με έναν κατηγορημα της μορφής: `detectiondir(ResolveList)`, όπου το όνομα του κατηγορήματος `detectiondir` καθορίζει τον τύπο της σύγκρουσης που η διεργασία συλλογισμού του εντοπισμού και επίλυσης συγκρούσεων θα προσπαθήσει να εντοπίσει για την συγκεκριμένη μέθοδο και το όρισμα `ResolveList` καθορίζει τον τρόπο με τον οποίο θα επιλύσει τη σύγκρουση (το κατηγορημα `detectiondir(ResolveList)` παρουσιάζεται και αναλύεται περαιτέρω σε σχέση με τις νοητικές ενέργειες εντοπισμού και επίλυσης συγκρούσεων στην υποενότητα 4.2.3).

Για παράδειγμα, στο περιβάλλον του `Tileworld`, καθ' όσον ο πράκτορας κινείται προς το κύβο (κατά ένα τετράγωνο κάθε φορά), ενεργεί αντανακλαστικά ακολουθώντας τις κατευθύνσεις συμπεριφοράς `[react([], goal([repeat])]` (σύμφωνα με τη μέθοδο (α) του Πίνακα 10 του Παραρτήματος Γ). Αυτές οι κατευθύνσεις συμπεριφοράς οδηγούν τον πράκτορα να κινείται αντανακλαστικά, δηλαδή χωρίς να ελέγχει για τυχόν συγκρούσεις που μπορεί να υπάρχουν στο πλαίσιο των ενεργειών του (λόγω της κατεύ-

θυσης συμπεριφοράς `react([])`). Η συμπεριφορά του όμως δεν είναι απολύτως αντανακλαστική, διότι ταυτόχρονα, όπως δείχνει η δεύτερη κατεύθυνση (`goal([repeat])`), αφού έχει εκτελέσει την ενέργεια ελέγχει κάθε φορά αν επιτεύχθηκε ο στόχος της μεθόδου (που είναι η κατάσταση «ο πράκτορας βρίσκεται στο κύβο»). Όποτε δεν επιτυγχάνεται ο στόχος, ο πράκτορας επιλύει τη «σύγκρουση» αυτή απορρίπτοντας τη μέθοδο που έχει επιλέξει και επιλέγοντας πάλι την ίδια. Επαναλαμβάνοντας συνεχώς τις ενέργειες – επιλογή σχετικής μεθόδου για την ενέργεια «μετακινήσου κατά ένα τετράγωνο», εκτέλεση της ενέργειας αυτής, έλεγχος επίτευξης στόχου και απόρριψής μεθόδου λόγω μη επίτευξης του στόχου – ο πράκτορας φτάνει στο κύβο. Η συμπεριφορά αυτή του πράκτορα αλλάζει όταν αντιλαμβάνεται είτε κάποιον άλλο πράκτορα κοντά στο στόχο του είτε ένα εμπόδιο στη πορεία του. Στην πρώτη περίπτωση, ο πράκτορας αλλάζει τη συμπεριφορά του ελέγχοντας τις νοητικές συνθήκες συνεργασίας (`collaborate mental conditions`) των γονικών ενεργειών της τρέχουσας ενέργειας στο πλαίσιο ενεργειών. Στη δεύτερη περίπτωση, αλλάζει τη συμπεριφορά του ελέγχοντας τις νοητικές συνθήκες οι οποίες εκφράζουν τη δυνατότητα του πράκτορα για το αν μπορεί να συνεχίσει τη πορεία του.

Η *νοητική συνθήκη συνεργασίας* (`Collaboration_Mental_Condition`) καθιστά τον πράκτορα ικανό να αποφασίζει για το αν πρέπει να σχεδιάσει και να εκτελέσει την ενέργεια μόνος του ή να συνεργαστεί με άλλους πράκτορες. Στη περίπτωση που η νοητική συνθήκη ισχύει τότε ο πράκτορας δεσμεύεται να πραγματοποιήσει την ενέργεια συνεργατικά (δηλαδή σχηματίζει μία πρόθεση της μορφής `<int_that_do>`). Σε αντίθετη περίπτωση, ο πράκτορας αναπτύσσει περαιτέρω τη μέθοδο και εκτελεί το πλάνο του προσωπικά. Επιπλέον, ελέγχοντας τη νοητική αυτή συνθήκη (και μόνο όταν ισχύει) ο πράκτορας μπορεί να παίρνει αποφάσεις για το σχήμα της ομάδας συνεργασίας. Πιο συγκεκριμένα, ελέγχοντας τη συνθήκη αυτή ο πράκτορας μπορεί να καθορίσει παραμέτρους της λίστας καταμερισμού εργασιών `Task_Allocation` (βλ. επόμενη παράγραφο). Για να αποφασίσει ποιοι πράκτορες θα αποτελούν την ομάδα ο πράκτορας χρησιμοποιεί τη γνώση που διαθέτει για το σκοπό αυτό καθώς επίσης επικοινωνεί με τους άλλους πράκτορες πριν την τελική του απόφαση. Κατά την επικοινωνία αυτή οι πράκτορες ανταλλάσσουν πληροφορίες σχετικά με τις ικανότητές τους και τις δυνατότητές τους και στη περίπτωση που συμφωνήσουν να συνεργαστούν σχηματίζουν τη

κοινή πεποίθηση ότι η ομάδα δεσμεύεται να πραγματοποιήσει την ενέργεια συνεργατικά (κάτι το οποίο είναι αναγκαίο στο μοντέλο συνεργασίας *SharedPlans*).

Για παράδειγμα, στο περιβάλλον του *Tileworld*, όταν ένας πράκτορας δεν έχει την δυνατότητα να μετακινήσει ένα εμπόδιο και έχει εντοπίσει ένα ή περισσότερους πράκτορες κοντά σ' αυτό, τότε ελέγχοντας τη συνθήκη συνεργασίας ο πράκτορας επικοινωνεί μαζί τους και εντοπίζει ποιος είναι πιο κοντά και έχει την δυνατότητα να μετακινήσει το εμπόδιο. Αν η συνθήκη αυτή ισχύσει τότε ο πράκτορας αρχίζει τη συνεργασία σχηματίζοντας μία πρόθεση της μορφής `<int_that_do>`.

Η λίστα καταμερισμού εργασιών (*Task\_Allocation*) επιτρέπει στο πράκτορα να γνωρίζει ποιοι είναι οι συνεργάτες του και να κατανέμει τις υπό-ενέργειες της μεθόδου (οι οποίες περιέχονται στην *ActionList*) σε αυτούς. Αν η λίστα αυτή είναι κενή τότε η μέθοδος θεωρείται ατομική και πραγματοποιείται αποκλειστικά από τον πράκτορα. Τα στοιχεία της λίστας αυτής καθορίζονται είτε από τον προγραμματιστή (κατά την διάρκεια του σχεδιασμού των μεθόδων) ή όταν ο πράκτορας ελέγχει τη νοητική συνθήκη της συνεργασίας. Η λίστα περιέχει όρους της μορφής: `do(Subgroup, ActionId, CooperationType)`, όπου *Subgroup* είναι μία λίστα με τα *ids* των πρακτόρων που θα πραγματοποιήσουν την υπό-ενέργεια με *id* *ActionId* (παρατηρήστε ότι το *Subgroup* μπορεί να περιέχει έναν ή περισσότερους πράκτορες της ομάδας). Η παράμετρος *CooperationType* δηλώνει τον τύπο της συνεργασίας που ο πράκτορας θα έχει κατά τη διάρκεια σχεδιασμού και εκτέλεσης της υπό-ενέργειας. Αυτή μπορεί να πάρει δύο τιμές: *delegate* ή *collaborate*. Αν οι πράκτορες της υπό-ομάδας αποδεχθούν την έκλυση για συνεργασία τότε και στους δύο τύπους συνεργασίας ο πράκτορας δεσμεύεται να μην εμποδίσει τις ενέργειες της υπό-ομάδας ή ακόμα και να τους βοηθήσει αν αυτό είναι αναγκαίο, δηλαδή σχηματίζει μία πρόθεση της μορφής `<int_that_cba>`.

Η τιμή *delegate* δηλώνει ότι ο πράκτορας πρέπει ν' αναθέσει την υπό-ενέργεια στο πράκτορα (ή στην υπό-ομάδα πρακτόρων). Ο πράκτορας ή οι πράκτορες που αποδέχονται την ανάθεση εργασίας δεν συμμετέχουν στην ανάπτυξη της μεθόδου της ενέργειας *Action*.

Στη περίπτωση που η παράμετρος *CooperationType* πάρει την τιμή *collaborate* τότε οι πράκτορες του *Subgroup* συμμετέχουν στο σχεδιασμό όχι μόνο της υπό-ενέργειας αλλά και γενικότερα της ενέργειας *Action*, διότι θεωρούνται μέλη της ομά-

δας της *Action*. Αυτό έχει ως αποτέλεσμα ο πράκτορας να συνεργάζεται με αυτά τα μέλη της ομάδας όχι μόνο κατά την εκτέλεση της μεθόδου αλλά και κατά τη διάρκεια του σχεδιασμού αυτής (αναλυτικότερα περιγράφουμε τη συνεργασία το πρακτόρων στο Κεφάλαιο 5).

*Η νοητική συνθήκη εφαρμογής της μεθόδου* (*Applicability\_Mental\_Condition*) καθιστά τον πράκτορα ικανό να γνωρίζει το πότε η μέθοδος είναι εφαρμόσιμη ή όχι (δηλαδή το πότε ο πράκτορας έχει τη δυνατότητα να πραγματοποιήσει τη συγκεκριμένη μέθοδο ή όχι). Στη περίπτωση που η λίστα κατευθύνσεων συμπεριφοράς περιέχει τη κατεύθυνση *cconstr* (βλ. Παράρτημα A), η συνθήκη αυτή μπορεί να χρησιμοποιηθεί και ως *περιορισμός του πλαισίου ενεργειών* (*context constraints*). Στη περίπτωση αυτή καθ' όσον ο πράκτορας σχεδιάζει και εκτελεί τις υπό-ενέργειες της μεθόδου οι περιορισμοί αυτοί θα πρέπει να μην παραβιάζονται.

Η λίστα υπό-ενεργειών (*Action\_List*) καθορίζει τις υποενέργειες που πρέπει να πραγματοποιηθούν από τον πράκτορα ή την ομάδα πρακτόρων προκειμένου να υλοποιηθεί η ενέργεια *Action* και να επιτευχθεί ο στόχος της μεθόδου *Recipe\_Goal*. Αν η λίστα είναι κενή τότε η ενέργεια θεωρείται βασική (*basic level action*), σε αντίθετη περίπτωση θεωρείται σύνθετη (*complex level action*). Κάθε ενέργεια που περιλαμβάνεται στη λίστα μπορεί να πραγματοποιηθεί είτε σε σειρά με τις υπόλοιπες ενέργειες είτε παράλληλα ανάλογα με το αν χωρίζονται με τον τελεστή *par* ή όχι (στην Εικόνα 24 παρουσιάζουμε τη BNF περιγραφή της λίστας υπό-ενεργειών).

```

<actionlist> ::= <actions_list> OR <null_list>
<actions_list> ::= [<gaction> | <actions_list>]
<gaction> ::= <action> OR <parallel_actions>
<parallel_actions> ::= par(<actions_list>, <actions_list>)
<null_list> ::= []
<action> ::= action_name(ActionId, ...)

```

Εικόνα 24: Η BNF περιγραφή της *ActionList*.

Τέλος, τα αποτελέσματα *Effects* μίας ενέργειας αναπαριστώνται με λίστα καταστάσεων φυσικού περιβάλλοντος ή νοητικών καταστάσεων που επιτυγχάνονται αμέσως μετά την εκτέλεση των ενεργειών της *Action\_List*. Ο πράκτορας ενημερώνει τη βάση

γνώσης με τα αποτελέσματα της ενέργειας όταν όλες οι υπό-ενέργειες της μεθόδου έχουν εκτελεστεί επιτυχώς.

#### 4.1.3. Ο μηχανισμός ενημέρωσης και εξαγωγής γνώσης

Μέσω του μηχανισμού ενημέρωσης και εξαγωγής γνώσης (Knowledge Base Consult and Update Mechanism) (KB-CUB) όλες οι διεργασίες συλλογισμού μπορούν να ενημερώνουν και να χρησιμοποιούν τη βάση γνώσης (βλ. Εικόνα 23) χρησιμοποιώντας τρεις νοητικές ενέργειες: τη `<learn>` με την οποία ο πράκτορας μπορεί να ενημερώνει τη βάση, τη `<bel>` με την οποία μπορεί να εξάγει γνώση από την ήδη υπάρχουσα και τη `<forget>` με την οποία μπορεί διαγράφει γεγονότα τα οποία ήδη γνωρίζει.

Η διατήρηση της συνέπειας της βάσης γνώσης είναι σημαντική για τον πράκτορα όχι μόνο για να μπορεί να παίρνει σωστές αποφάσεις κατά τη διάρκεια σχεδιασμού των πλάνων του, αλλά και για να μπορούν να λειτουργούν παράλληλα και συντονισμένα οι μηχανισμοί ελέγχου της BDI-Πρακτικής Συλλογιστικής του ICAGENT. Για παράδειγμα, ο πράκτορας δεν μπορεί να σχηματίσει την επιθυμία να πετύχει μία κατάσταση και ταυτόχρονα να έχει τη πρόθεση να τη πετύχει, διότι η μεν επιθυμία θα ενεργοποιήσει το μηχανισμό εντοπισμού και επίλυσης συγκρούσεων η δε πρόθεση το μηχανισμό υλοποίησης ενεργειών.

##### 4.1.3.1. Ενημέρωση της βάσης γνώσης

Ο πράκτορας είναι ικανός να μαθαίνει νέα γεγονότα εκτελώντας την ακόλουθη νοητική ενέργεια:

$$\langle \text{learn} \rangle ::= \text{learn}(\text{Fact}, C)$$

όπου `Fact` αναπαριστά ένα γεγονός (κατάσταση) του φυσικού περιβάλλοντος ή μία νοητική κατάσταση το ιδίου ή άλλου πράκτορα και `c` ένας δυαδικός αριθμός έτσι όπως τον ορίσαμε στην υποενότητα 4.1.1.

Κάθε φορά που ο πράκτορας εκτελεί τη νοητική ενέργεια `<learn>` ο μηχανισμός KB-CUB, πριν εισάγει την αντίστοιχη πεποίθηση `bel(Fact, C)` στη βάση γνώσης προσπαθεί να εντοπίσει πιθανές συγκρούσεις (conflicts) μεταξύ της πεποίθησης αυτής και των άλλων πεποιθήσεων που ήδη έχει.

Η γνώση του πράκτορα σχετικά με τον τρόπο με τον οποίο ελέγχει τη συνέπεια της βάσης γνώσης και ενημερώνει αυτή κατά την εισαγωγή μιας καινούργιας πεποιθήσης, αναπαρίσταται με κανόνες της μορφής:

```
<update_rule> ::= update_rule(belief(Fact,C), MntlCond, ActionList)
```

Το γεγονός αυτό της Prolog αναπαριστά για τον πράκτορα ένα κανόνα ενημέρωσης και έχει την ακόλουθη σημασία: Η πεποίθηση `belief(Fact, C)` είναι συνεπής με τις άλλες πεποιθήσεις που ο πράκτορας διατηρεί στη βάση γνώσης του μόνο αν η νοητική συνθήκη `MntlCond` ισχύει. Σ' αυτή τη περίπτωση ο μηχανισμός KB-CUB ενημερώνει τη βάση εκτελώντας σειριακά τις (νοητικές) ενέργειες της `ActionList` και μετά εισάγει τη πεποίθηση `belief(Fact, C)`. Αν η νοητική συνθήκη `MntlCond` δεν ισχύει τότε ο πράκτορας πιστεύει ότι δεν μπορεί να ενημερώσει τη βάση γνώσης εκτελώντας τις ενέργειες της `ActionList` και για αυτό το λόγο αγνοεί το συγκεκριμένο κανόνα `<update_rule>`.

Πιο συγκεκριμένα, οποτεδήποτε εκτελείται η νοητική ενέργεια `<learn>` ο μηχανισμός KB-CUB ενεργεί όπως παρακάτω:

Πρώτα δημιουργεί μία λίστα με όλους τους σχετικούς κανόνες `<update_rule>` για το συγκεκριμένο γεγονός (`Fact`)<sup>24</sup>. Ένας κανόνας είναι σχετικός αν και μόνο αν οι όροι `Fact` και `C` μπορούν να ενοποιηθούν με τους αντίστοιχους όρους της `<learn>`. Για τους κανόνες ενημέρωσης ισχύει η υπόθεση του κλειστού κόσμου (`closed world assumption`): Αν δεν υπάρχει κανένας σχετικός κανόνας, τότε ο πράκτορας πιστεύει ότι η συγκεκριμένη πεποίθηση δεν έρχεται σε αντίφαση με κάποια που ήδη έχει και άρα την εισάγει στη βάση γνώσης (διότι δεν υπάρχει καμία ένδειξη για το αντίθετο).

Δεύτερον, για κάθε σχετικό κανόνα (ξεκινώντας από την αρχή της λίστας) ελέγχει αν η νοητική συνθήκη `MntlCond` που περιλαμβάνει ισχύει. Αν πράγματι ισχύει τότε εκτελεί τις ενέργειες που περιέχει η αντίστοιχη `ActionList` του κανόνα και τέλος εισάγει το `bel(Fact, C)` στη βάση γνώσης. Αν δεν ισχύει η `MntlCond`, τότε αγνοεί τον κανόνα και περνάει στον επόμενο.

---

<sup>24</sup> Η σειρά με την οποία η κανόνες εισάγονται στη λίστα είναι η ίδια με αυτή που έχουν εισαχθεί στη βάση γνώσης του πράκτορα.

Αν η νοητική συνθήκη  $\text{MntlCond}$  όλων των κανόνων της λίστας δεν ισχύουν τότε ο μηχανισμός KB-CUB θεωρεί ότι η πεποίθηση  $\text{belief}(\text{Fact}, \text{C})$  δεν είναι συμβατή με τη γνώση που ήδη έχει στη βάση (και ούτε μπορεί να γίνει μέσω των νοητικών ενεργειών κάποιας  $\text{ActionList}$ ) και για αυτό αγνοεί τη νοητική ενέργεια  $\langle \text{learn} \rangle$  και δεν εισάγει την αντίστοιχη πεποίθηση στη βάση γνώσης.

Για να είναι έγκυρη (valid) η παραπάνω διαδικασία κάνουμε χρήση μίας ισχυρής υπόθεσης: η γνώση του πράκτορα σχετικά με το  $\text{Fact}$ , μπορεί ν' αλλάξει από την άγνοια στη γνώση και το αντίστροφο (δηλαδή η λογική που χρησιμοποιούμε είναι μη-μονότονη). Αυτό σημαίνει ότι αν ο πράκτορας στο τρέχοντα χρόνο δεν γνωρίζει τίποτα για την αλήθεια ή το ψεύδος ενός γεγονότος (δηλαδή  $\text{C}=0$ ), τότε την επόμενη χρονική στιγμή μπορεί να μάθει είτε ότι αληθεύει είτε ότι δεν ισχύει. Το αντίστροφο αυτής της πρότασης επίσης ισχύει, δηλαδή αν ο πράκτορας πιστεύει ότι ισχύει το συγκεκριμένο γεγονός ή ότι δεν ισχύει, τότε μπορεί να περάσει στην άγνοια σχετικά για το γεγονός αυτό. Επίσης θεωρούμε ότι αν ο πράκτορας δεν έχει καμία πεποίθηση κατάφασης ή άρνησης για το γεγονός στη βάση γνώσης του, τότε ο πράκτορας πιστεύει ότι έχει άγνοια για την ύπαρξη του γεγονότος αυτού (δηλαδή ότι ισχύει  $\text{bel}(\text{Fact}, 0)$ ).

Τέλος, μία σημαντική ιδιότητα των πεποιθήσεων είναι αυτή της διατήρησης (persistence) κατά τη διάρκεια του χρόνου. Δηλαδή, αν σε οποιαδήποτε χρονική στιγμή ο πράκτορας πιστεύει ότι ένα γεγονός ισχύει ( $\text{belief}(\text{Fact}, 1)$ ) ή ότι δεν ισχύει ( $\text{belief}(\text{not Fact}, 1)$ ) ή ότι έχει άγνοια για αυτό (δηλαδή είτε  $\text{belief}(\text{Fact}, 0)$  είτε  $\text{belief}(\text{not Fact}, 0)$  είτε δεν υπάρχει στη βάση γνώσης) τότε στις επόμενες χρονικές στιγμές θα εξακολουθήσει να έχει τη πεποίθηση αυτή μέχρι να μάθει κάτι διαφορετικό από αυτό που ήδη γνωρίζει.

#### 4.1.3.2. Εξαγωγή γνώσης

Ο πράκτορας έχει την ικανότητα να εξάγει γνώση από τη βάση γνώσης του χρησιμοποιώντας την ακόλουθη νοητική ενέργεια:

$\langle \text{bel} \rangle ::= \text{bel}(\text{Fact}, \text{C})$



Όπου  $Fact$  είναι το γεγονός (η κατάσταση) που θέλει να ελέγξει αν ισχύει και  $c$  είναι μία δυαδική μεταβλητή η οποία μπορεί να πάρει τη τιμή 1 ή 0 ή να μην έχει ακόμα καθοριστεί (un-instantiated variable).

Αν η μεταβλητή  $c$  έχει την τιμή 1 τότε η πεποίθηση  $\langle bel \rangle$  είναι αληθής (true) αν και μόνο αν ο πράκτορας γνωρίζει το γεγονός αυτό (δηλαδή, αν η βάση γνώσης του περιέχει το γεγονός  $belief(Fact1, 1)$  και το  $Fact1$  μπορεί να ενοποιηθεί με το  $Fact$ ) ή αν μπορεί να το συνάγει μέσω κάποιου κανόνα  $\langle consult\_rule \rangle$ . Πιο συγκεκριμένα, αν ο πράκτορας αποτύχει να εντοπίσει μία πεποίθηση  $\langle belief \rangle$  που να ενοποιείται με το  $Fact$  τότε προσπαθεί να το συνάγει χρησιμοποιώντας κανόνες της μορφής:

$$\langle consult\_rule \rangle ::= consult\_rule(Mnt1Cond, Fact)$$

Αν η νοητική συνθήκη ενός τουλάχιστον κανόνα ισχύσει τότε ο πράκτορας συμπεραίνει ότι το γεγονός ισχύει.

Αν η μεταβλητή  $c$  έχει την τιμή 0 τότε η πεποίθηση  $\langle bel \rangle$  είναι αληθής αν και μόνο αν ο πράκτορας έχει άγνοια για το γεγονός  $Fact$ , δηλαδή όταν ισχύει ένα από τα παρακάτω:

1. Ο πράκτορας έχει στη KB το γεγονός  $belief(Fact1, 0)$  και το  $Fact1$  μπορεί να ενοποιηθεί με το  $Fact$  της  $\langle bel \rangle$ .
2. Ο πράκτορας έχει στη KB το γεγονός  $belief(not\ Fact1, 0)$  και το  $Fact1$  μπορεί να ενοποιηθεί με το  $Fact$  της  $\langle bel \rangle$ .
3. Ο πράκτορας δεν έχει κανένα σχετικό  $\langle belief \rangle$  ούτε για το  $Fact$  ούτε για το  $not(Fact)$  στη KB.

Τέλος, αν η μεταβλητή  $C$  δεν έχει ακόμα καθοριστεί τότε ο πράκτορας ρωτά τη KB προκειμένου να μάθει αν πράγματι πιστεύει ότι ισχύει ή έχει άγνοια για αυτό.

#### 4.1.3.3. Διαγραφή γεγονότων από τη βάση γνώσης

Για να σταματήσει να πιστεύει ο πράκτορας ένα γεγονός θα πρέπει είτε να μάθει την άρνηση αυτού, είτε να αλλάξει τη πεποίθησή του εισάγοντας το γεγονός  $belief(Fact, 0)$ , είτε να το διαγράψει από τη βάση.

Για να διαγραφεί ένα `<belief>` από τη βάση γνώσης χρησιμοποιούμε τη νοητική ενέργεια `<forget>`:

```
<forget> ::= forget(Fact)
```

Οποτεδήποτε ο πράκτορας προσπαθεί να διαγράψει μία πεποίθηση από τη βάση γνώσης, ο μηχανισμός KB-CUM ενεργεί ως ακολούθως: (α) δημιουργεί μία λίστα με όλους τους σχετικούς κανόνες `<update_rule>` με  $C=0$ , (β) για κάθε έναν από αυτούς ελέγχει την αντίστοιχη `MntlCond`. Για κάθε ένα σχετικό κανόνα που η αντίστοιχη `MntlCond` είναι αληθής εκτελεί όλες τις ενέργειες που περιέχει η αντίστοιχη `ActionList` και αφαιρεί τη πεποίθηση από τη βάση γνώσης. Σε αντίθετη περίπτωση αν όλες οι νοητικές συνθήκες `MntlCond` δεν αληθεύουν τότε αγνοεί την `<forget>`. Τέλος αν δεν υπάρχει κανένας σχετικός κανόνας με  $C=0$  τότε αφαιρεί τη πεποίθηση από τη βάση γνώσης χωρίς να την ενημερώσει περαιτέρω.

## 4.2. Οι διεργασίες συλλογισμού

Η υπό-ενότητα αυτή περιγράφει τις διεργασίες συλλογισμού οι οποίες υλοποιούν τον BDI-μηχανισμό ελέγχου της πρακτικής συλλογιστικής του ICAGENT. Επίσης εστιάζουμε το ενδιαφέρον μας στο πώς επηρεάζεται η συμπεριφορά του πράκτορα από τις συλλογιστικές αυτές διεργασίες κατά τη διάρκεια του σχεδιασμού και της εκτέλεσης ατομικών ή ομαδικών ενεργειών σε δυναμικά περιβάλλοντα με περιορισμένους πόρους.

Νοητική Κατάσταση Ενεργοποίησης Διεργασίας Συλλογισμού	Διεργασία Συλλογισμού	Αποτέλεσμα Διεργασίας Συλλογισμού
Γνώση χειρισμού των αισθητήρων του σώματος του πράκτορα:  <code>bel(sensor(   Sensor_Id, Fact,   Sensor_Time_Period,   Learn_Mental_Condition,   Forget_Mental_Condition ))</code>	Αντίληψης	Ενημέρωση της βάσης γνώσης για νέα γεγονότα ή ενέργειες που συμβαίνουν στο περιβάλλον του πράκτορα:  <code>learn(Fact) or forget(Fact)</code>
Γνώση για καταστάσεις που ο πράκτορας πρέπει να πάρει τη πρωτοβουλία να δράση:  <code>bel(situation(Id,   TimePeriod,   MentalCondition, Action))</code>	Αναγνώρισης Καταστάσεων	Δημιουργία νέων επιθυμιών πραγματοποίησης ενεργειών:  <code>desire_to(Agent,Action)</code>
Ενέργειες που ο πράκτορας επιθυμεί να πραγματοποιήσει και έχει δεσμευθεί να τις πραγματοποιήσει επιτυχώς (μόνος	Εντοπισμού και Επίλυσης Συγκρούσεων	Δημιουργία επιθυμιών για έλεγχο συγκρούσεων:  <code>desire_to(Agent,</code>

του ή σε συνεργασία με άλλους πράκτορες):  <pre> desire_to (Agent, Action) and int_that (Agent,           cba (Group, Action, Recipe)) </pre>		<pre> detect_conflict (Action) </pre>
Πεποίθηση ότι μία ενέργεια βρίσκεται σε σύγκρουση:  <pre> bel (conflict (Action,               ConflictAction,               Type_of_Conflict) ) </pre>	Εντοπισμού και Επίλυσης Συγκρούσεων	Δημιουργία επιθυμίας επίλυσης σύγκρουσης:  <pre> desire_to (Agent,           resolve_conflict (Action)) </pre>
Πεποίθηση ότι μία ενέργεια δεν βρίσκεται σε σύγκρουση:  <pre> bel (not_conflict (Action, _, _)) </pre>	Εντοπισμού και Επίλυσης Συγκρούσεων	Ο πράκτορας στοχεύει να πραγματοποιήσει την ενέργεια:  <pre> goal_to (Agent, Action) </pre>
Ενέργεια που ο πράκτορας επιθυμεί να πραγματοποιήσει ή έχει δεσμευτεί να βοηθήσει μία ομάδα πρακτόρων να τη πραγματοποιήσει:  <pre> desire_to (Agent, Action) or int_that (Agent,           cba (Group, Action, Recipe)) </pre>	Ανάπτυξης Πλάνων	Δημιουργία επιθυμίας να επιλέξει μία σχετική μέθοδο:  <pre> desire_to (Agent,           select_recipe (Action)) </pre>
Ενέργεια που ο πράκτορας έχει στόχο να τη πραγματοποιήσει:  <pre> goal_to (Agent, Action) </pre>	Ανάπτυξης Πλάνων	Δημιουργία επιθυμίας ν' αναπτύξει τη σχετική και εφαρμόσιμη μέθοδο της ενέργειας:  <pre> desire_to (Agent,           elaborate_recipe (Action)) </pre>
Ενέργεια που ο πράκτορας έχει δεσμευτεί να πραγματοποιήσει:  <pre> int_to (Agent, Action) </pre>	Υλοποίησης Προθέσεων	Ενημέρωση της βάσης γνώσης για τα αποτελέσματα της πραγματοποίησης της ενέργειας:  <pre> learn (Fact) or forget (Fact) </pre>

*Πίνακας 3:* Κάθε διεργασία συλλογισμού ενεργοποιείται από μία ή περισσότερες νοητικές καταστάσεις και έχει ως αποτέλεσμα την αλλαγή της εσωτερικής κατάστασης του πράκτορα. Οι νέες νοητικές καταστάσεις του πράκτορα οδηγούν σε περαιτέρω συλλογισμό ενεργοποιώντας άλλες νοητικές διεργασίες.

Ο BDI-μηχανισμός ελέγχου της πρακτικής συλλογιστικής του ICAGENT (BDI-CPR) υλοποιείται μέσω συλλογιστικών διεργασιών (βλ. Πίνακας 3) με τους οποίους ο πράκτορας μπορεί να αντιλαμβάνεται το φυσικό του περιβάλλον, να αξιοποιεί πληροφορίες σχετικά με το πότε πρέπει να δράσει, να ελέγχει τις νοητικές ενέργειες οι οποίες του επιτρέπουν να εντοπίζει και να επιλύει συγκρούσεις μεταξύ ενεργειών, ν' αναπτύσσει τα πλάνα του και να τα υλοποιεί. Στις υπό-ενότητες που ακολουθούν παρουσιάζουμε με λεπτομέρεια τις διεργασίες αυτές καθώς και το πώς επηρεάζουν τη συμπεριφορά του πράκτορα.

#### 4.2.1. Η διεργασία συλλογισμού της αντίληψης

Όπως ήδη έχουμε αναφέρει, το ενδιάμεσο λογισμικό (μεταξύ του φυσικού περιβάλλοντος και του BDI-CPR) με το οποίο ο πράκτορας μπορεί να υλοποιεί τις προθέσεις του, να αντιλαμβάνεται τις αλλαγές του περιβάλλοντός του και να επικοινωνεί με άλλους πράκτορες ονομάζεται «σώμα» του πράκτορα.

Τα δύο κύρια δομικά στοιχεία του σώματος είναι οι αισθητήρες εντοπισμού γεγονότων-ενεργειών (Event-Fact Detectors) και οι μηχανισμοί δράσης (Actuators). Οι αισθητήρες εντοπισμού γεγονότων και ενεργειών χρησιμοποιούνται από τον πράκτορα για να αντιλαμβάνεται τα γεγονότα που συμβαίνουν στο περιβάλλον του, καθώς και για να ερμηνεύει τις ενέργειες των άλλων πρακτόρων κάτω από πλαίσια ενεργειών.

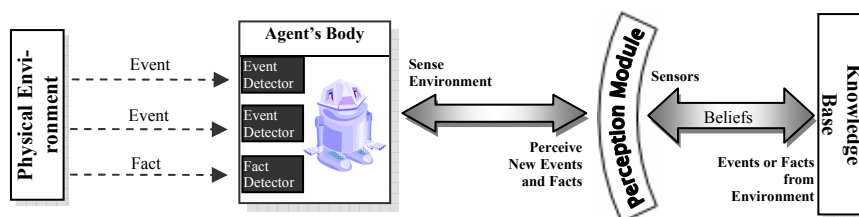
Η μονάδα αντίληψης (Perception Module) του BDI-CPR υλοποιεί τη διεργασία συλλογισμού της αντίληψης<sup>25</sup> (βλ. Εικόνα 25). Αυτή καθιστά ικανό τον πράκτορα να χειρίζεται τους αισθητήρες εντοπισμού γεγονότων και ενεργειών έτσι ώστε να μπορεί να αντιλαμβάνεται τα γεγονότα που συμβαίνουν στο φυσικό περιβάλλον του, τις ενέργειες των άλλων πρακτόρων<sup>26</sup>, καθώς επίσης και να ελέγχει την ύπαρξη καταστάσεων στο φυσικό του περιβάλλον.

Η τελευταία ικανότητά του πράκτορα είναι σημαντική, διότι δεν χρειάζεται να γνωρίζει ακριβώς όλους τους μηχανισμούς λειτουργίας του φυσικού περιβάλλοντος. Η λειτουργία του φυσικού περιβάλλοντος βασίζεται στο δικό της μοντέλο και έχει το ίδιο την ευθύνη να αλλάζει τη κατάστασή του σύμφωνα με το μοντέλο αυτό. Μέσω της συλλογιστικής διεργασίας της αντίληψης ο πράκτορας έχει την ικανότητα να αντιλαμβάνεται την κατάσταση μέρους ή ολόκληρου του φυσικού περιβάλλοντος. Αυτό έχει ως αποτέλεσμα λιγότερη ανάγκη (άρα και προσπάθεια από πλευράς του προγραμματιστή) αναπαράστασης γνώσης σχετικής με το φυσικό περιβάλλον του πράκτορα [Maes, 1994].

---

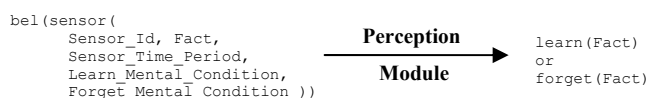
<sup>25</sup> Από λεξικό Γ. Μπαμπινιώτη: **αντίληψη** (η) **1.** η ικανότητα να αντιλαμβανόμαστε μέσω των αισθήσεων και το αποτέλεσμα αυτής **2.** η ικανότητα να καταλαβαίνει κανείς τι συμβαίνει γύρω του, να συλλαμβάνει στοιχεία της πραγματικότητας και να διαμορφώνει σαφή εικόνα για πρόσωπα και καταστάσεις.

<sup>26</sup> Δηλαδή ν' αντιλαμβάνεται τόσο τα αποτελέσματα των ενεργειών τους (facts) όσο και το τύπο των ενεργειών αυτών(events).



Εικόνα 25: Το σώμα και ο μηχανισμός αντίληψης του ICAGENT.

Η διεργασία συλλογισμού εντοπισμού γεγονότων–ενεργειών λειτουργεί ως ένα είδος φίλτρου πληροφοριών από το φυσικό περιβάλλον προς τη βάση γνώσης του πράκτορα. Το φίλτρο αυτό βελτιστοποιεί τη συμπεριφορά του πράκτορα διότι δεν εισάγονται στη βάση γνώσης του γεγονότα (ή ενέργειες) τα οποία συμβαίνουν στο περιβάλλον του αλλά δεν είναι σχετικά με τους στόχους του (για παράδειγμα, στο Tileworld εμπόδια τα οποία δεν εμποδίζουν τη κίνηση του πράκτορα αγνοούνται).



Εικόνα 26: Ο πράκτορας «φιλτράρει» τα γεγονότα και τις ενέργειες που συμβαίνουν στο περιβάλλον του κάνοντας χρήση της γνώσης του για το τι πρέπει να προσέχει και τότε.

Στο πλαίσιο εργασίας του ICAGENT η γνώση για το *πότε* θα χειριστεί ο πράκτορας έναν αισθητήρα και για *ποιο* τύπο γεγονότων ή ενεργειών θα τον χρησιμοποιήσει εκφράζεται μέσω της προδιαγραφής (ή προσδιορισμού λειτουργίας) του αισθητήρα (sensor's specification knowledge). Μία προδιαγραφή αισθητήρα αναπαρίσταται με την ακόλουθη δομή:

```
<sensor> ::= sensor( Sensor_Id, Fact, Sensor_Time_Period,
                    Learn_Mental_Condition, Forget_Mental_Condition )
```

Όπου *SensorId* αναπαριστά το αναγνωριστικό (id) της προδιαγραφής του αισθητήρα (ένας αισθητήρας μπορεί να χρησιμοποιείται για παραπάνω από έναν λόγους και για αυτό μπορεί να υπάρχουν παραπάνω από μία προδιαγραφές), το *Fact* καθορίζει τον τύπο του γεγονότος (ή της ενέργειας) που παρακολουθεί, η *Learn\_Mental\_Condition* είναι μία νοητική συνθήκη η οποία καθορίζει το *πότε* ο πράκτορας πιστεύει ότι το συγκεκριμένο γεγονός ισχύει, και τέλος η *Forget\_Mental\_Condition* είναι μία νοητική συνθήκη η οποία καθορίζει το *πότε* ο πράκτορας παύει να πιστεύει την ύπαρξη του συγκεκριμένου γεγονότος.

Για κάθε προδιαγραφή ενός αισθητήρα (που περιλαμβάνεται στη βάση γνώσης του πράκτορα) η μονάδα αντίληψης (perception module) δημιουργεί ένα νήμα. Κάθε νήμα τρέχει μία διαδικασία η οποία περιοδικά (ο χρόνος της περιόδου εξαρτάται από την τιμή που παίρνει η παράμετρος `Sensor_Time_Period`) ελέγχει τις νοητικές συνθήκες `Learn_Mental_Condition` και `Forget_Mental_Condition` προκειμένου να αναγνωρίσει νέα γεγονότα που συμβαίνουν στο περιβάλλον του πράκτορα ή να διαγράψει παλαιά από τη βάση γνώσης τα οποία δεν ισχύουν. Με αυτό τον τρόπο ο πράκτορας μπορεί να χρησιμοποιεί ταυτόχρονα παραπάνω από έναν αισθητήρα και να παρακολουθεί το περιβάλλον του ανεξάρτητα από τις άλλες διεργασίες συλλογισμού.

Η ανεξαρτησία αυτή από τις άλλες διεργασίες του ICAGENT δεν σημαίνει ότι η πρακτική συλλογιστική του πράκτορα δεν επηρεάζει τον τρόπο με τον οποίο παρακολουθεί το περιβάλλον του. Κάθε σύνθετη ενέργεια του πράκτορα μπορεί να περιλαμβάνει υπό-ενέργειες (της μορφής `<learn>` και `<forget>`) οι οποίες να ενημερώνουν τη βάση γνώσης είτε για καινούργιες προδιαγραφές αισθητήρων είτε να διαγράφουν ήδη υπάρχουσες. Για παράδειγμα, στο περιβάλλον του `Tileworld`, οποτεδήποτε ο πράκτορας γεμίζει μία τρύπα με ένα κύβο, η ενέργεια `<fill_hole>` έχει ως αποτέλεσμα να εισάγει μία προδιαγραφή αισθητήρα με την οποία ο πράκτορας μπορεί να εντοπίσει νέες τρύπες και κύβους. Επίσης, οποτεδήποτε ο πράκτορας εντοπίσει ένα αντικείμενο που τον ενδιαφέρει (όπως για παράδειγμα μία τρύπα ή ένα κύβο) τότε μέσω της συλλογιστικής διεργασίας ενημέρωσης της βάσης γνώσης δημιουργεί μία νέα προδιαγραφή αισθητήρα με την οποία ελέγχει (monitors) την περιοχή γύρω από το αντικείμενο.

Λόγω του ότι οι προδιαγραφές των αισθητήρων αλλάζουν η μονάδα αντίληψης περιοδικά ελέγχει αν τα αντίστοιχα νήματα αυτών που εκτελούνται είναι έγκυρα (valid), δηλαδή αν για κάθε νήμα ο πράκτορας πιστεύει ότι η προδιαγραφή του ισχύει (`bel(<sensor>)`). Αν όχι τότε η μονάδα της αντίληψης καταστρέφει το νήμα.

Κάθε νήμα μπορεί να θεωρηθεί ως νοητική ενέργεια<sup>27</sup> η οποία ενημερώνει τις πεποιθήσεις του πράκτορα όποτε ισχύσει μία από τις δύο νοητικές συνθήκες (`Learn_Mental_Condition` ή `Forget_Mental_Condition`) της αντίστοιχης προδιαγρα-

φής. Η εκτέλεση αυτού του τύπου νοητικών ενεργειών, καθώς επίσης και ο τρόπος με τον οποίο οι προδιαγραφές δημιουργούνται στη βάση γνώσης, επηρεάζουν τη συμπεριφορά του πράκτορα.

Πιο συγκεκριμένα, η συμπεριφορά του πράκτορα, όσον αφορά τη διεργασία συλλογισμού της αντίληψης, επηρεάζεται: (α) από το σχεδιασμό των πλάνων που αφορούν τις προδιαγραφές των αισθητήρων (perception planning), δηλαδή από τη συμπεριφορά του πράκτορα κατά τη διάρκεια του σχεδιασμού των πλάνων των ενεργειών που διαμορφώνουν το τί και το πώς θα παρατηρεί στο περιβάλλον του (των ενεργειών δηλαδή που διαμορφώνουν τις προδιαγραφές των αισθητήρων <sensors>) και (β) από τη συλλογιστική (reasoning) του πράκτορα κατά τη διάρκεια ενημέρωσης της βάσης γνώσης λόγω νέων πληροφοριών από το φυσικό του περιβάλλον.

Για παράδειγμα, ο πράκτορας αφού γεμίσει μία τρύπα για να βρει την επόμενη τρύπα θα πρέπει να ψάξει είτε μέσα στο οπτικό του πεδίο είτε να πάει σε περιοχές που πιστεύει ότι είναι πιο πιθανό να βρει μία άδεια τρύπα (σε αυτό μπορούν να βοηθήσουν και οι άλλοι πράκτορες που ενδεχομένως να καλύπτουν περιοχές που ο πράκτορας δεν έχει πρόσβαση). Αν ο πράκτορας πρέπει να σχεδιάσει τα πλάνα των ενεργειών που διαμορφώνουν τις προδιαγραφές των αισθητήρων (perception planning) (για παράδειγμα κάνοντας χρήση της συλλογιστικής διεργασίας του εντοπισμού και επίλυσης συγκρούσεων) τότε η συμπεριφορά του πράκτορα γίνεται πιο στοχαστική. Εντούτοις αν ο πράκτορας γνωρίζει εκ των προτέρων πως πρέπει να χειριστεί τους αισθητήρες χρησιμοποιώντας προδιαγραφές οι οποίες έχουν προκαθοριστεί από τον προγραμματιστή, τότε η συμπεριφορά του πράκτορα θεωρείται πιο αντανakλαστική.

Επιπλέον, η δημιουργία νέων πεποιθήσεων για γεγονότα τα οποία συμβαίνουν στο περιβάλλον του πράκτορα μπορεί να επηρεάσουν τη συμπεριφορά του. Για παράδειγμα, νέες πεποιθήσεις μπορεί να οδηγήσουν τον πράκτορα να σταματήσει να δρα αντανakλαστικά και να αρχίσει έλεγχο τυχόν συγκρούσεων μεταξύ των περιορισμών του πλαισίου ενεργειών (context constraints) και της τρέχουσας νοητικής του κατάστασης (όπως για παράδειγμα στο Tileworld, κατά τη διάρκεια διάσχισης του μονοπατιού ο

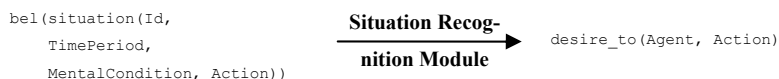
---

<sup>27</sup> Δηλαδή ως ενέργεια η οποία αλλάζει μόνο τη νοητική κατάσταση του πράκτορα.

εντοπισμός νέων εμποδίων οδηγεί τον πράκτορα στον έλεγχο αν ο δρόμος είναι προσβάσιμος). Αυτό έχει ως αποτέλεσμα είτε να επανασχεδιάζει τα πλάνα του (πχ. να βρει κάποιο άλλο μονοπάτι) είτε να πράττει με τέτοιο τρόπο που να επιλύει τις συγκρούσεις (πχ. να συνεργαστεί με κάποιον άλλο πράκτορα που να μπορεί να μετακινήσει το εμπόδιο).

#### 4.2.2. Η διεργασία συλλογισμού της αναγνώρισης καταστάσεων

Η μονάδα αναγνώρισης καταστάσεων (Situation Recognition Module) καθιστά το πράκτορα ικανό ν' αναγνωρίζει τις καταστάσεις (νοητικές ή/και του φυσικού περιβάλλοντος) εκείνες για τις οποίες πρέπει να πάρει τη πρωτοβουλία να δράσει.



*Εικόνα 27:* Η μονάδα αναγνώρισης καταστάσεων χρησιμοποιεί γνώση του πράκτορα (κανόνες) για το πότε πρέπει να πάρει τη πρωτοβουλία να δράσει.

Η μονάδα αυτή υλοποιεί τη διεργασία συλλογισμού αναγνώρισης καταστάσεων αξιοποιώντας κανόνες οι οποίοι περιγράφουν το πότε πρέπει ο πράκτορας να πάρει τη πρωτοβουλία να δράσει και ποια ενέργεια να πραγματοποιήσει (ή ποιο στόχο να πετύχει)<sup>28</sup>. Κάθε κανόνας αναγνώρισης καταστάσεων αναπαρίσταται από μία νοητική συνθήκη και μία ενέργεια:

```
<situation> ::= situation(Id, TimePeriod, MentalCondition, Action)
```

Όπου `Id` είναι ένας χαρακτηριστικός μοναδικός αριθμός για κάθε κανόνα, `TimePeriod` είναι ένας ακέραιος ο οποίος καθορίζει τη περίοδο όπου η μονάδα αναγνώρισης καταστάσεων ελέγχει αν ισχύει η νοητική συνθήκη `MentalCond`. Η νοητική συνθήκη `MentalCond` καθορίζει τη νοητική κατάσταση του πράκτορα η οποία τον οδηγεί στην επιθυμία να πραγματοποιήσει την ενέργεια `Action`.

<sup>28</sup> Θεωρούμε ότι η επίτευξη ενός στόχου (κατάστασης) μπορεί να πραγματοποιηθεί με την ενέργεια `achieve(State)` και σε ότι αφορά το σχεδιασμό και την υλοποίησή της δεν έχει καμία διαφορά από οποιαδήποτε άλλη ενέργεια εκτός του τρόπου με τον οποίο επιλέγει τη σχετική μέθοδο (βλ. υποενότητα 4.2.4).



Όπως η μονάδα της αντίληψης, έτσι και η μονάδα της αναγνώρισης νοητικών καταστάσεων εκτελείται παράλληλα με τις άλλες μονάδες (διεργασίες συλλογισμού) και είναι υπεύθυνη για τη δημιουργία ή την καταστροφή νημάτων μέσω των οποίων ο πράκτορας αναγνωρίζει τις καταστάσεις. Η μονάδα περιοδικά αναζητά από τη βάση γνώσης κανόνες αναγνώρισης καταστάσεων και για κάθε κανόνα που δεν τρέχει ήδη ένα νήμα, δημιουργεί ένα. Οποτεδήποτε ο πράκτορας πιστεύει ότι ένας τέτοιος κανόνας δεν είναι έγκυρος (δηλαδή ότι το γεγονός  $bel(situation(Id, TimePeriod, MentalCond, Action))$  δεν ισχύει) τότε η μονάδα αναγνώρισης καταστάσεων καταστρέφει το αντίστοιχο νήμα.

Κάθε νήμα αναγνώρισης καταστάσεων ελέγχει τη νοητική συνθήκη  $MntlCond$  του αντίστοιχου κανόνα ανά έναν αριθμό χιλιοστών του δευτερολέπτου που το όρισμα  $TimePeriod$  του κανόνα καθορίζει. Όταν η νοητική αυτή συνθήκη ισχύει, το νήμα σχηματίζει μία επιθυμία για τον πράκτορα να πραγματοποιήσει την αντίστοιχη ενέργεια.

Το σύνολο των κανόνων αναγνώρισης καταστάσεων που περιέχονται στη βάση γνώσης του πράκτορα μπορεί να ενημερώνεται είτε κατόπιν κάποιου συλλογισμού (δηλαδή μέσω κάποιου κανόνα ενημέρωσης  $\langle update\_rule \rangle$ ) είτε κατόπιν εκτέλεσης κάποιας μεθόδου όπου στη λίστα αποτελεσμάτων της ( $action's\ effect\ list$ ) περιέχεται ένας τέτοιος κανόνας.

Για παράδειγμα, στο *Tileworld* οποτεδήποτε ένας πράκτορας αντιλαμβάνεται την ύπαρξη μίας τρύπας κοντά σε αυτόν τότε μέσω ενός κανόνα ενημέρωσης  $\langle update\_rule \rangle$  εισάγει ένα νέο κανόνα αναγνώρισης κατάστασης  $\langle situation \rangle$ . Η ύπαρξη αυτού του κανόνα στη βάση γνώσης του πράκτορα έχει ως αποτέλεσμα η μονάδα αναγνώρισης καταστάσεων να δημιουργεί ένα νήμα το οποίο ανά κάποια εκατοστά του δευτερολέπτου να προσπαθεί να εντοπίσει την ακριβή θέση τη συγκεκριμένης τρύπας. Αν δεν υπάρχει κάποια άλλη πιο κοντά σε αυτόν τότε σχηματίζει μία επιθυμία να γεμίσει ο πράκτορας την τρύπα.

Αν ο πράκτορας δεσμευθεί να γεμίσει τη τρύπα ή αν αντιληφθεί ότι η τρύπα δεν υπάρχει πλέον στο περιβάλλον, τότε ο πράκτορας πιστεύει ότι ο κανόνας αυτός αναγνώρισης κατάστασης δεν είναι πια έγκυρος. Αυτό έχει ως αποτέλεσμα η μονάδα αναγνώ-

ρισης καταστάσεων να διαγράψει από τη βάση γνώσης το συγκεκριμένο κανόνα και να σταματήσει την εκτέλεση του αντίστοιχου νήματος.

Κάθε νήμα αναγνώρισης κατάστασης μπορεί να θεωρηθεί ως μία νοητική ενέργεια, η οποία αλλάζει τη νοητική κατάσταση του πράκτορα εισάγοντας νέες επιθυμίες. Η εκτέλεση αυτών των ενεργειών καθώς και ο έλεγχος που πραγματοποιούν οι κανόνες αναγνώρισης καταστάσεων επηρεάζουν τη συμπεριφορά του πράκτορα.

Πιο συγκεκριμένα, η διεργασία συλλογισμού της αναγνώρισης καταστάσεων επηρεάζει τη συμπεριφορά του πράκτορα με τους ακόλουθους τρόπους:

Πρώτον, οποτεδήποτε ο πράκτορας σχεδιάζει τα πλάνα του για να αποφασίσει *ποιοι* κανόνες είναι έγκυροι στο τρέχοντα χρόνο τότε η συμπεριφορά του χαρακτηρίζεται ως στοχαστική (*deliberative*). Σε αντίθετη περίπτωση όπου ο πράκτορας δεν σχεδιάζει πλάνα για τους κανόνες αναγνώρισης καταστάσεων παρά μόνο ελέγχει τις νοητικές συνθήκες έτοιμων κανόνων τότε η συμπεριφορά του χαρακτηρίζεται αντανακλαστική (πάντα σε ότι αφορά την αναγνώριση καταστάσεων).

Το να παίρνουν οι πράκτορες σωστές αποφάσεις σχετικά με το *ποιοι* κανόνες αναγνώρισης καταστάσεων πρέπει να χρησιμοποιηθούν στο τρέχον πλαίσιο του περιβάλλοντος (φυσικού και νοητικού) είναι σημαντικό, διότι οι πράκτορες δεν πρέπει να σπαταλούν το χρόνο τους σε άσκοπους ελέγχους για το *ποιοι* κανόνες είναι έγκυροι στη τρέχουσα κατάσταση του περιβάλλοντός τους.

Ο δεύτερος τρόπος με τον οποίο η διεργασία συλλογισμού της αναγνώρισης καταστάσεων επηρεάζει τη συμπεριφορά του πράκτορα είναι όταν ελέγχει το *πότε* ο πράκτορας πρέπει να πάρει τη πρωτοβουλία να δράσει (δηλαδή όταν τα νήματα ελέγχουν τις νοητικές συνθήκες των κανόνων αναγνώρισης καταστάσεων). Ορθές αποφάσεις για το *πότε* ένας πράκτορας πρέπει να παίρνει πρωτοβουλίες δράσης και με *ποιο* τρόπο είναι σημαντικές διότι ο πράκτορας δεν πρέπει να δημιουργεί επιθυμίες που αργότερα είτε δεν μπορούν να πραγματοποιηθούν (π.χ. διότι δεν έχει τις κατάλληλες δυνατότητες) είτε μπορεί να έρθουν σε σύγκρουση με άλλες προθέσεις που μπορεί να έχει. Για παράδειγμα στο *Tileworld* κάθε πράκτορας σχηματίζει μία επιθυμία να γεμίσει μία τρύπα μόνο αν πιστεύει ότι αυτή δεν περιστοιχίζεται από εμπόδια που δεν μπορεί να μετακινήσει.

### 4.2.3. Η διεργασία συλλογισμού του εντοπισμού και επίλυσης συγκρούσεων

Όταν η συμπεριφορά του πράκτορα είναι αποτέλεσμα ατομικού στοχασμού (deliberative behaviour) τότε για κάθε σχετική μέθοδο (recipe) που επιλέγει, ελέγχει αν αυτή είναι εφαρμόσιμη (applicable). Εναλλακτικά, αν αποφασίσει να δράσει με αντανακλαστικό τρόπο (reactive behaviour), ο πράκτορας αναπτύσσει το πλάνο της ενέργειας χωρίς να ελέγχει αν οι σχετικοί μέθοδοι που επιλέγει είναι εφαρμόσιμες ή όχι. Μεταξύ αυτών των δύο ακραίων συμπεριφορών, η συμπεριφορά του πράκτορα (σε ότι αφορά τη διεργασία συλλογισμού του εντοπισμού και επίλυσης συγκρούσεων) εξαρτάται από τον τρόπο (ή τους τρόπους) που θα επιλέξει για να εντοπίσει και επιλύσει τυχόν συγκρούσεις.

Το πλαίσιο ανάπτυξης έξυπνων πρακτόρων ICAGENT παρέχει τη δυνατότητα εντοπισμού και επίλυσης συγκρούσεων τόσο μεταξύ ενεργειών όσο και μεταξύ των νοητικών συνθηκών εφαρμογής των μεθόδων (applicability mental conditions) των ενεργειών και της νοητικής κατάστασης του πράκτορα. Αναλυτικά τους τύπους των συγκρούσεων τους περιγράφουμε στην επόμενη υποενότητα.

Η υλοποίηση της μονάδας του εντοπισμού και επίλυσης συγκρούσεων (Reconciliation Module) βασίζεται στα δύο αξιώματα [Grosz and Kraus, 1999] που αναφέραμε στην υποενότητα 4.1.1 και τα οποία συνθέτουν τις ελάχιστες προϋποθέσεις για έναν πράκτορα να έχει μία πρόθεση της μορφής `<int_that>`:

Το πρώτο αξίωμα ορίζει ότι αν ο πράκτορας έχει τη πρόθεση να πετύχει μία κατάσταση (`<int_that>`), τότε θα πρέπει να πιστεύει ότι η κατάσταση αυτή είναι δυνατόν να επιτευχθεί. Αυτό το αξίωμα οδηγεί τους πράκτορες μίας ομάδας να στοχάζονται (deliberate) ελέγχοντας αν έχουν τη δυνατότητα να πετύχουν τη προτιθέμενη κατάσταση. Σύμφωνα με το πλαίσιο εργασίας ICAGENT, οι δεσμεύσεις μίας ομάδας για κοινή δράση (group activity commitments) οδηγούν τους πράκτορες να συνεργάζονται και να σχηματίζουν βασικές δεσμεύσεις (core case commitments) για τις υπό-ενέργειες που συνθέτουν το κοινό τους πλάνο. Οι τελευταίες δεσμεύσεις έχουν ως αποτέλεσμα οι πράκτορες να ελέγχουν αν οι ίδιοι ή οι συνεργάτες τους έχουν τη δυνατότητα να τις πραγματοποιήσουν. Πιο συγκεκριμένα οι βασικές δεσμεύσεις οδηγούν τους πράκτορες στο να ελέγξουν την εφαρμογή των μεθόδων που έχουν επιλέξει ως σχετικές για τις ενέργειες που προτίθενται να πραγματοποιήσουν (δηλαδή να ελέγξουν τις συνθήκες

των μεθόδων που εκφράζουν τις δυνατότητες των πρακτόρων να πραγματοποιήσουν τις ενέργειες). Στη περίπτωση που εντοπίσουν μία σύγκρουση τότε προσπαθούν να την επιλύσουν είτε μόνοι τους (αρχικώς) είτε επικοινωνώντας με τους άλλους πράκτορες της ομάδας για να αρχίσουν μία νέα συνεργασία αυτή τη φορά για την επίλυση της σύγκρουσης.

Το δεύτερο αξίωμα αναφέρει ότι (α) αν ο πράκτορας πιστεύει ότι η κατάσταση που προτίθεται να πετύχει είναι επιτεύξιμη, και (β) δεν πιστεύει ότι ισχύει στο τρέχοντα χρόνο, τότε αν γνωρίζει κάποια ενέργεια που μπορεί να την κάνει να ισχύσει τότε πρέπει ο πράκτορας να προσπαθήσει να πραγματοποιήσει την ενέργεια αυτή. Όπως τονίζεται στην εργασία [Grosz and Kraus, 1999], αυτό το αξίωμα συνδέει έμμεσα τις προθέσεις του πράκτορα της μορφής `<int_that>` με ενέργειες που πρέπει να πραγματοποιήσει προκειμένου να τηρήσει αυτές τις προθέσεις του.

Πιο συγκεκριμένα, η μονάδα εντοπισμού και επίλυσης συγκρούσεων του BDI-CPR υλοποιείται μέσω των νοητικών ενεργειών `<detect_conflict>` και `<resolve_conflict>` (οι ενέργειες αυτές παρουσιάζονται αναλυτικότερα στις επόμενες δύο υποενότητες). Η μονάδα αυτή ελέγχει διαρκώς την ύπαρξη τριών νοητικών καταστάσεων:

A. Αν ο πράκτορας

(α) έχει την επιθυμία να πραγματοποιήσει μία ενέργεια για την οποία έχει βρει μία σχετική μέθοδο και

(β) έχει αποφασίσει να μην αντιδράσει αντανακλαστικά (δηλαδή η κατεύθυνση συμπεριφοράς `react` δεν περιλαμβάνεται στη λίστα κατευθύνσεων συμπεριφοράς της μεθόδου),

τότε ο πράκτορας σχηματίζει τη βασική δέσμευση (core case commitment) ότι θα έχει τη δυνατότητα να πραγματοποιήσει την ενέργεια μόνος του (intention that can bring about the action). Σύμφωνα με το πρώτο από τα παραπάνω αξιώματα, μία πρόθεση του τύπου `<int_that_cba>` οδηγεί τον πράκτορα σε μία στοχαστική συμπεριφορά δημιουργώντας την επιθυμία να εντοπίσει τυχόν συγκρούσεις της μεθόδου ακολουθώντας τις κατευθύνσεις εντοπισμού συγκρούσεων της μεθόδου.

B. Στη περίπτωση που τα ακόλουθα ισχύουν:

(α) ο πράκτορας έχει την επιθυμία να πραγματοποιήσει μία ενέργεια για την οποία έχει βρει μία σχετική μέθοδο,

(β) δεν έχει αποφασίσει να αντιδράσει αντανακλαστικά (δηλαδή η κατεύθυνση συμπεριφοράς `react` δεν περιλαμβάνεται στη λίστα κατευθύνσεων συμπεριφοράς της μεθόδου) και

(γ) έχει δεσμευθεί να πραγματοποιήσει την ενέργεια συνεργατικά (group activity commitment), δηλαδή έχει σχηματίσει τη πρόθεση `int_that(Agent, do(Group, Action, Recipe))`,

τότε ο πράκτορας προτίθεται να βοηθήσει την ομάδα στο να έχει τη δυνατότητα πραγματοποίησης της ενέργειας συνεργατικά. Αυτό καθορίζεται από τη βασική δέσμευση `<int_that_cba>` που ο πράκτορας σχηματίζει για την κοινή ενέργεια. Αυτή η δέσμευση οδηγεί το πράκτορα στο να εντοπίσει τυχόν συγκρούσεις και να ενημερώσει τους άλλους πράκτορες στη περίπτωση που εντοπίσει κάποιο πρόβλημα.

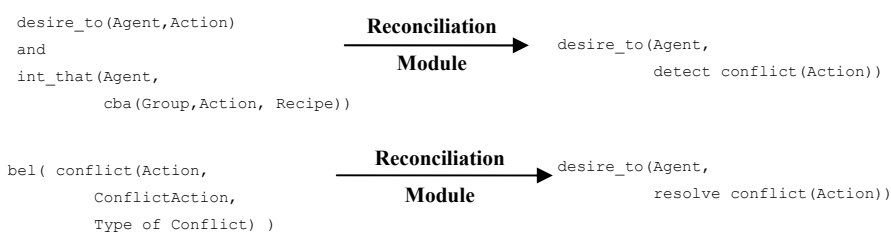
Γ. Αν ο πράκτορας επιθυμεί να πραγματοποιήσει μία ενέργεια με αντανακλαστικό τρόπο ή αν πιστεύει ότι η ενέργεια δεν έρχεται σε σύγκρουση με άλλες προθέσεις του πράκτορα ή των συνεργατών του, τότε η διεργασία συλλογισμού του εντοπισμού και επίλυσης διενέξεων αλλάζει τη νοητική κατάσταση του πράκτορα μεταβάλλοντας την επιθυμία `desire_to(Agent, Action)` του σε στόχο `goal_to(Agent, Action)`. Όπως θα περιγράψουμε στην υπό-ενότητα 4.2.4 αυτός ο στόχος έχει ως αποτέλεσμα να ενεργοποιήσει τη διεργασία συλλογισμού ανάπτυξης των πλάνων του πράκτορα.

Στη περίπτωση που ο πράκτορας έχει σχηματίσει τη πρόθεση `<int_that_cba>` και πιστεύει ότι η σχετική μέθοδος που έχει επιλέξει (ή έχουν επιλέξει) δεν είναι εφαρμόσιμη λόγω κάποιας σύγκρουσης, τότε ο πράκτορας σχηματίζει την επιθυμία να επιλύσει τη σύγκρουση πραγματοποιώντας την ενέργεια `<resolve_conflict>`. Αυτό έρχεται ως αποτέλεσμα της εφαρμογής των δύο παραπάνω αξιωμάτων:

Σύμφωνα με το πρώτο αξίωμα, ο πράκτορας προσπαθεί να αναγνωρίσει αν ο ίδιος ή η ομάδα έχει τη δυνατότητα να πραγματοποιήσει τη προτιθέμενη ενέργεια πραγματοποιώντας τη διεργασία συλλογισμού του εντοπισμού και επίλυσης συγκρούσεων. Στη περίπτωση που εντοπίσει κάποια σύγκρουση δεν σημαίνει ότι δεν υπάρχει κανένας τρόπος να αποκτήσει τη δυνατότητα πραγματοποίησης αυτής. Αντιθέτως, η διεργασία

συλλογισμού εντοπισμού και επίλυσης συγκρούσεων οδηγεί τον πράκτορα στο να σχηματίζει μία πρόθεση επίλυσης της σύγκρουσης κάτι το οποίο έπεται από το πρώτο αξίωμα.

Σύμφωνα με το δεύτερο αξίωμα και λόγω του ότι ο πράκτορας πάντα γνωρίζει μία σχετική μέθοδο για τη νοητική ενέργεια `<resolve_conflict>` σχηματίζει μία επιθυμία να τη πραγματοποιήσει ακολουθώντας τις κατευθύνσεις συμπεριφοράς της μεθόδου που έχει επιλέξει.



*Εικόνα 28:* Όταν ο πράκτορας επιθυμεί να πραγματοποιήσει μία ενέργεια και έχει δεσμευθεί να τη πραγματοποιήσει επιτυχώς (είτε ενεργώντας ατομικά είτε σε συνεργασία με άλλους πράκτορες) τότε συμπεριφέρεται με στοχαστικό τρόπο ελέγχοντας αν η μέθοδος της ενέργειας αυτής είναι εφαρμόσιμη και επιλύοντας τυχόν συγκρούσεις.

#### 4.2.3.1. Η νοητική ενέργεια εύρεσης σύγκρουσης

Όπως αναφέραμε παραπάνω, όποτε ένας πράκτορας αποφασίζει να ελέγξει αν μία μέθοδος είναι εφαρμόσιμη (applicable), η διεργασία εντοπισμού και επίλυσης συγκρούσεων δημιουργεί στο πράκτορα την επιθυμία να πραγματοποιήσει τη νοητική ενέργεια `<detect_conflict>`. Η ενέργεια αυτή έχει την ακόλουθη δομή:

```
<detect_conflict> ::= detect_conflict(Action)
```

Αρχικά (by default), η ενέργεια αυτή εκτελείται ως βασική (basic level action) και χρησιμοποιώντας τις κατευθύνσεις συμπεριφοράς της αντίστοιχης μεθόδου για την Action εντοπίζει πιθανές συγκρούσεις στο τρέχον πλαίσιο ενεργειών.

Σύμφωνα με τις κατευθύνσεις συμπεριφοράς του Πίνακα 4 του Παραρτήματος Α, ο πράκτορας μπορεί να ελέγξει:

- (1) τη νοητική συνθήκη εφαρμογής της μεθόδου (applicability mental condition) η οποία εκφράζει τη δυνατότητα του πράκτορα να πραγματοποιήσει τη συγκεκριμένη μέθοδο.
- (2) τις νοητικές συνθήκες εφαρμογής των μεθόδων των (γονικών) ενεργειών στο πλαίσιο ενεργειών για τις οποίες ο πράκτορας έχει υποθέσει ότι πρέπει να διατηρούνται (context constraints).
- (3) την επίτευξη του στόχου της μεθόδου (recipe goal) μετά τη πραγματοποίηση της ενέργειας (της μεθόδου αυτής) καθώς και κατά τη διάρκεια εκτέλεσης αυτής.
- (4) συγκρούσεις μεταξύ των αποτελεσμάτων (effects) μίας μεθόδου και των νοητικών συνθηκών εφαρμογής των ενεργειών που ακολουθούν και δεν έχουν πραγματοποιηθεί ακόμα.
- (5) συγκρούσεις μεταξύ των αποτελεσμάτων (effects) μίας μεθόδου και των στόχων των μεθόδων των γονικών ενεργειών.

Μετά την εκτέλεση της νοητικής αυτής ενέργειας, ο πράκτορας είτε πιστεύει ότι δεν υπάρχει σύγκρουση είτε ότι υπάρχει τουλάχιστον μία. Το τελευταίο γεγονός αναπαριστάται ως:

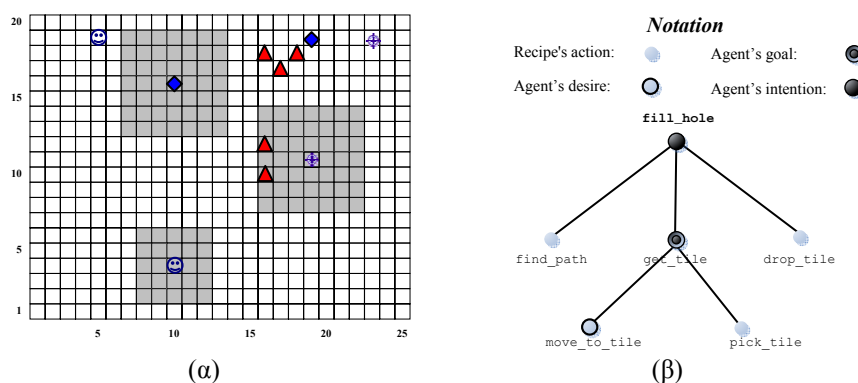
```
<conflict> ::= conflict(Action, ConflictAction, Type_of_Conflict)
```

και έχει τη σημασία ότι η μέθοδος της ενέργειας *Action* βρίσκεται σε σύγκρουση με τη μέθοδο της ενέργειας *ConflictAction*. Το όρισμα *Type\_of\_Conflict* αναπαριστά τον τύπο της σύγκρουσης και μπορεί να πάρει μία από τις τιμές του Πίνακα 4 του Παραρτήματος Α. Επιπλέον, λόγω του ότι το γεγονός *<conflict>* αναπαριστά και συγκρούσεις μεταξύ της μεθόδου της ενέργειας και της τρέχουσας νοητικής κατάστασης του πράκτορα το όρισμα *ConflictAction* μπορεί να πάρει και τη τιμή *null*.

Θεωρώντας το παράδειγμα του *Tileworld*, στην Εικόνα 29(β) απεικονίζεται το πλαίσιο των προθέσεων (intentional context) του πράκτορα  $G_0$  σε μία χρονική στιγμή όπου αυτός μετακινείται προς το κύβο  $T_0(10,16)$  προκειμένου να το σηκώσει και μετά να γεμίσει με αυτό την τρύπα  $H_0(19,11)$ .

Οι τρεις υποενέργειες πραγματοποιούνται υπό τις ακόλουθες νοητικές συνθήκες (βλ. μέθοδο (α) του Πίνακα 8 του Παραρτήματος Γ): Η τρύπα και ο κύβος πρέπει να υπάρ-

χουν στο περιβάλλον, η τρύπα πρέπει να είναι άδεια και ο κύβος όπως και η τρύπα πρέπει να είναι προσβάσιμα από τον πράκτορα. Οι περιορισμοί αυτοί εκφράζουν τις νοητικές εκείνες συνθήκες που πρέπει να ισχύουν καθ' όσον ο πράκτορας πραγματοποιεί την ενέργεια  $\langle fill\_hole \rangle$ . Για τον λόγο αυτό οι συνθήκες αυτές θεωρούνται περιορισμοί του πλαισίου ενεργειών (context constraints) της  $\langle fill\_hole \rangle$  (βλ. τις Applicability Mental Conditions της μεθόδου (α) του Πίνακα 8 του Παραρτήματος Γ) και ελέγχονται καθ' όλη τη διάρκεια εκτέλεσής της. Στη περίπτωση που μία από αυτές τις συνθήκες παύει να ισχύσει, τότε ακολουθώντας τις κατευθύνσεις συμπεριφοράς (Behaviour Directives) της μεθόδου ο πράκτορας αναγνωρίζει τη σύγκρουση και αναθεωρεί τη μέθοδο διαλέγοντας μία άλλη.



Εικόνα 29: Μία αποτύπωση της κατάστασης της σκακιέρας, καθώς ο πράκτορας  $G_0$  κινείται προς τον κύβο και το αντίστοιχα πλαίσιο των προθέσεών του.

Είναι σημαντικό να παρατηρήσουμε ότι ο έλεγχος της προσβασιμότητας τόσο προς τον κύβο όσο και προς τη τρύπα κοστίζει σε χρόνο διότι ο πράκτορας πρέπει συνεχώς να σχεδιάζει πλάνα για να βρίσκει μονοπάτια προς αυτά. Για αυτόν το λόγο οι περιορισμοί του πλαισίου των ενεργειών της  $\langle fill\_hole \rangle$  πρέπει να ελέγχονται μόνο όταν είναι απαραίτητο. Λαμβάνοντας υπ' όψη την ανάγκη αυτή, ο πράκτορας κινείται προς το κύβο (ή προς τη τρύπα) με αντανακλαστικό τρόπο. Δηλαδή, καθ' όσον εκτελεί την ενέργεια  $\langle move\_to\_tile \rangle$  δεν λαμβάνει υπ' όψη τους παραπάνω περιορισμούς της  $\langle fill\_hole \rangle$ . Η συμπεριφορά αυτή αλλάζει μόνο όταν αντιληφθεί είτε ένα νέο εμπόδιο στο δρόμο του είτε έναν άλλο πράκτορα ο οποίος είναι κοντά στο κύβο.

Ο πράκτορας συμπεριφέρεται με αυτό τον τρόπο διότι οι κατευθύνσεις συμπεριφοράς της μεθόδου της  $\langle move\_to\_tile \rangle$  τον οδηγούν να μην εντοπίζει πιθανές συγκρού-



σεις μεταξύ των περιορισμών του πλαισίου ενεργειών της `<fill_hole>` και της τρέχουσας νοητικής του κατάστασης. Στη περίπτωση όπου ο πράκτορας αντιληφθεί μία αλλαγή στο περιβάλλον του τότε οι κατευθύνσεις συμπεριφοράς της μεθόδου της `<move_to_tile>` αλλάζουν μετατρέποντας τη συμπεριφορά του πράκτορα από αντανάκλαστική σε στοχαστική ελέγχοντας του περιορισμούς που θέτουν οι ενέργειες των σύνθετων ενεργειών που περιλαμβάνουν την ενέργεια `<move_to_tile>` (δηλαδή, τους περιορισμούς των ενεργειών `<get_tile>` και `<fill_hole>`).

#### 4.2.3.2. Η νοητική ενέργεια επίλυσης συγκρούσεων

Οποτεδήποτε ο πράκτορας εντοπίσει μία σύγκρουση μέσω της νοητικής ενέργειας `<detect_conflict>`, η διεργασία συλλογισμού εντοπισμού και επίλυσης συγκρούσεων δημιουργεί στο πράκτορα την επιθυμία να επιλύσει τη σύγκρουση αυτή εκτελώντας τη νοητική ενέργεια `<resolve_conflict>`:

```
<resolve_conflict> ::= resolve_conflict(Action)
```

Η νοητική αυτή ενέργεια αξιοποιεί τη γνώση του πράκτορα για το τρέχον πλαίσιο ενεργειών της `Action`, τον τύπο της σύγκρουσης, τις κατευθύνσεις επίλυσης συγκρούσεων της μεθόδου που έχει επιλέξει για την ενέργεια αυτή καθώς και τη γνώση επίλυσης συγκρούσεων οι οποίες εξαρτώνται από την εφαρμογή που ο πράκτορας χρησιμοποιείται (domain depended knowledge about resolve conflicts).

Η νοητική ενέργεια επίλυσης συγκρούσεων οδηγεί τον πράκτορα είτε σε προσωπικό στοχασμό είτε σε κοινωνικό. Αν η κατεύθυνση επίλυσης συγκρούσεως είναι μία από τις `resolve`, `reconsider`, `backtrack` ή `repeat`, τότε ο πράκτορας είτε προσπαθεί να επιλύσει τη σύγκρουση χρησιμοποιώντας γνώση που εξαρτάται από το πεδίο εφαρμογής (`resolve`), είτε προσπαθεί να επιλέξει μία εναλλακτική μέθοδο για την `Action` (`reconsider`), είτε προσπαθεί να βρει μία εναλλακτική μέθοδο για την αμέσως γονική ενέργεια στο πλαίσιο ενεργειών που περιλαμβάνει την `Action` (`backtrack`), είτε εγκαταλείπει τη μέθοδο και επαναεπιλέγει την ίδια θεωρώντας ότι αν ξαναπροσπαθήσει η σύγκρουση θα σταματήσει να υπάρχει (`repeat`).

Για παράδειγμα, στο `Tileworld` οποτεδήποτε ο πράκτορας εκτελεί την βασική ενέργεια `<move_to_tile>` και δεν έχει φτάσει στον κύβο, τότε σύμφωνα με τις κατευθύνσεις

συμπεριφοράς της αντίστοιχης μεθόδου (που περιλαμβάνει τη κατεύθυνση `goal(repeat)`) ο πράκτορας αντιλαμβάνεται μία σύγκρουση όταν ο στόχος της ενέργειας δεν έχει επιτευχθεί (ο πράκτορας δεν έχει φτάσει στον κύβο). Σε αυτή την περίπτωση η κατεύθυνση συμπεριφοράς `repeat` οδηγεί τον πράκτορα να επαναεπιλέξει τη μέθοδο κάνοντας τον πράκτορα να κινείται συνεχώς προς τα εμπρός (ακολουθώντας πάντα το μονοπάτι) μέχρι να φτάσει στο κύβο.

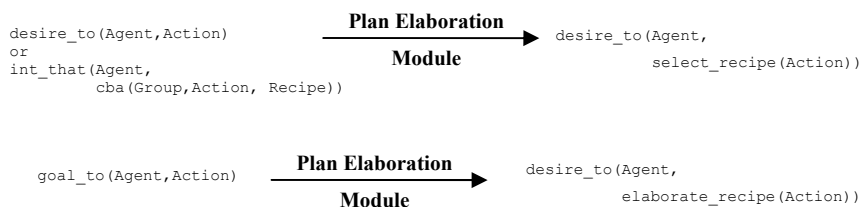
Αν ο πράκτορας αποτύχει να επιλύσει τη σύγκρουση ακολουθώντας τις παραπάνω κατευθύνσεις και δεν θέλει να αγνοήσει τη σύγκρουση (`override`), ή να εγκαταλείψει το πλάνο ολόκληρο (`drop_plan`), ή να ακυρώσει την εκτέλεση της ενέργειας (`cancel`), ή να αναβάλλει για αργότερα την εκτέλεση της μεθόδου της ενέργειας μέχρι να επιλυθεί η σύγκρουση (`postpone`) τότε ο πράκτορας μπορεί να αποφασίσει να συνεργαστεί με άλλους πράκτορες προκειμένου να επιλύσει τη σύγκρουση (`collaborate`). Στη τελευταία περίπτωση, ο πράκτορας δεσμεύεται να πραγματοποιήσει την ενέργεια σε συνεργασία με άλλους πράκτορες σχηματίζοντας μία δέσμευση για κοινή δράση (`group activity commitment`). Αυτό έχει ως αποτέλεσμα να αναπτύξει το πλάνο της ενέργειας σε συνεργασία με τους άλλους πράκτορες (βλ. Κεφάλαιο 5).

#### 4.2.4. Ανάπτυξη και υλοποίηση πλάνων

Η νοητική διεργασία της ανάπτυξης των πλάνων ενός πράκτορα ελέγχει τις νοητικές ενέργειες οι οποίες αναλύουν τις σύνθετες ενέργειες σε άλλες απλούστερες αλλάζοντας έτσι το πλαίσιο ενεργειών (`context of action`) του πράκτορα. Όταν ένας πράκτορας αναπτύσσει το πλάνο μίας ομαδικής ενέργειας τότε η διεργασία συλλογισμού ανάπτυξης πλάνων τον οδηγεί να επικοινωνήσει με άλλους πράκτορες προκειμένου να καθορίσει τα μέλη της ομάδας, τη μέθοδο που θα ακολουθήσουν από κοινού, καθώς και για να διαπραγματευτεί το καταμερισμό των υπό-ενεργειών της μεθόδου μεταξύ των μελών της ομάδας.

Πιο συγκεκριμένα, η μονάδα ανάπτυξης πλάνων (`Plan Elaboration Module`) ελέγχει τις νοητικές ενέργειες επιλογής `<select_recipe>` και ανάπτυξης μεθόδου `<elaborate_recipe>`. Η πρώτη καθιστά τον πράκτορα ικανό να επιλέγει σχετικές μεθόδους (από τη βιβλιοθήκη μεθόδων) για ενέργειες που τον ενδιαφέρει να πραγματοποιηθούν, ενώ η δεύτερη νοητική ενέργεια παρέχει στο πράκτορα τρόπους με τους οποίους μπορεί

να αναπτύξει μερικώς αναπτυγμένα (προσωπικά ή κοινά) πλάνα σε πλήρως αναπτυγμέ-



*Εικόνα 30:* Αν ο πράκτορας έχει την επιθυμία να πραγματοποιήσει μία ενέργεια ή έχει δεσμευθεί να βοηθήσει μία ομάδα πρακτόρων να τη πραγματοποιήσει, τότε δημιουργεί την επιθυμία να επιλέξει μία σχετική μέθοδο για αυτή. Αν έχει ήδη επιλέξει μία σχετική μέθοδο και πιστεύει ότι είναι εφαρμόσιμη (δηλαδή όταν έχει ένα στόχο να πραγματοποιήσει μία ενέργεια), τότε δημιουργεί την επιθυμία να αναπτύξει τη μέθοδο αυτή περαιτέρω.

Οποτεδήποτε ένας πράκτορας (α) επιθυμεί να πραγματοποιήσει μία ενέργεια ή (β) έχει δεσμευθεί για την επιτυχή πραγματοποίηση αυτής από κάποιο άλλο πράκτορα (ή κάποιους άλλους πράκτορες) (core case commitment) τότε ο πράκτορας χρειάζεται να βρει μία σχετική μέθοδο για την ενέργεια αυτή.

Η νοητική ενέργεια επιλογής μίας μεθόδου (recipe) για μία ενέργεια καθιστά τον πράκτορα ικανό να αποφασίζει τον τρόπο με τον οποίο θα πραγματοποιήσει την ενέργεια αυτή ενεργώντας είτε ατομικά είτε σε συνεργασία με άλλους πράκτορες. Η νοητική αυτή ενέργεια έχει την ακόλουθη δομή:

`<select_recipe> ::= select_recipe(Action)`

Η αρχική (default) μέθοδος της `<select_recipe>` θεωρεί τη νοητική ενέργεια αυτή ως βασική και υλοποιείται από τα ακόλουθα βήματα:

1. Από όλες τις μεθόδους στη βιβλιοθήκη μεθόδων επιλέγει μία όπου το όρισμα της ενέργειας ταυτοποιείται με αυτό της `Action`.
2. Ελέγχει αν η μέθοδος αυτή είναι σχετική, δηλαδή αν η νοητική συνθήκη `Relevance_Mental_Condition` αληθεύει ή όχι. Αν αληθεύει τότε η μέθοδος αυτή χαρακτηρίζεται ως σχετική και ενοποιούνται οι παράμετροι της μεθόδου με αυτές της `Relevance_Mental_Condition`. Εισάγοντας τη καινούργια μέθοδο με τις ενοποιημένες παραμέτρους στη βάση γνώσης αυτόματα ενημερώνεται το τρέχον πλαίσιο ενεργειών (στο οποίο η `Action` ανήκει) με τις υπό-ενέργειες που η μέθοδος πε-

ριέχει. Αν η μέθοδος αυτή δεν είναι σχετική, τότε ο αλγόριθμος επαναεκτελείται από το πρώτο βήμα διαλέγοντας μία εναλλακτική μέθοδο.

Αν ο πράκτορας θεωρήσει τη `<select_recipe>` ως σύνθετη ενέργεια τότε για να σχεδιάσει και εκτελέσει το πλάνο της χρησιμοποιεί τις ίδιες διεργασίες συλλογισμού (του μηχανισμού ελέγχου της BDI-πρακτικής συλλογιστικής) που χρησιμοποιεί και για κάθε άλλη ενέργεια του πεδίου εφαρμογής<sup>29</sup>.

Αυτό έχει μία σημαντική συνέπεια στη συμπεριφορά του πράκτορα: τον καθιστά ικανό να επιλέξει διαφορετικές μεθόδους εκτέλεσης για τη `<select_recipe>` δεδομένης της γνώσης που έχει για τη τρέχουσα κατάσταση του φυσικού του περιβάλλοντος και του πλαισίου ενεργειών. Για παράδειγμα, η μέθοδος που ο πράκτορας επιλέγει για να πραγματοποιήσει τη `<select_recipe>` όταν συνεργάζεται με άλλους πράκτορες μπορεί να είναι διαφορετική από τη μέθοδο που επιλέγει όταν ενεργεί ατομικά (βλ. Πίνακας 7).

Όπως επισημάναμε στη υπό-παράγραφο 4.1.1, για να δεσμευθεί ο πράκτορας να εκτελέσει μία ενέργεια (δηλαδή να σχηματίσει τη νοητική δομή `<int_to>` για την `Action`) θα πρέπει πρώτα να έχει επιλέξει μία σχετική και εφαρμόσιμη μέθοδο για αυτή και να γνωρίζει τουλάχιστον το πώς μπορεί να την αναπτύξει περαιτέρω, δηλαδή να έχει βρει τουλάχιστον ένα πλήρως αναπτυγμένο πλάνο για τη νοητική ενέργεια `<elaborate_recipe>`.

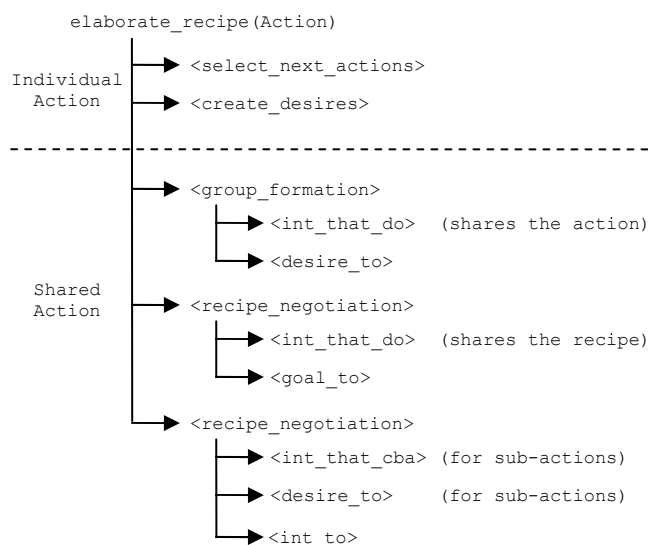
Για αυτό το λόγο, οποτεδήποτε ο πράκτορας έχει επιλέξει (είτε μόνος του είτε σε συνεργασία με άλλους πράκτορες) μία σχετική και εφαρμόσιμη μέθοδο για μία ενέργεια, σχηματίζει την επιθυμία να την αναπτύξει περαιτέρω. Η νοητική ενέργεια της ανάπτυξης μίας μεθόδου έχει την ακόλουθη δομή:

```
<elaborate_recipe> ::= elaborate_recipe(Action)
```

---

<sup>29</sup> Στη περίπτωση που η `Action` της `<select_recipe>` για την οποία ο πράκτορας επιλέγει μία μέθοδο είναι επίσης μία `<select_recipe>` και προκειμένου να αποφύγει ατέρμονους βρόγχους του τύπου `select_recipe(Id, select_recipe(...))` θεωρούμε πάντα τη δεύτερη `<select_recipe>` ως βασική (basic level action) η οποία εκτελείται χωρίς περαιτέρω ανάλυση.

Η εφαρμογή της `<elaborate_recipe>` εξαρτάται από τον τύπο της ενέργειας (ατομική ή κοινή) και από τη γνώση που έχει ο πράκτορας σχετικά με την ανάπτυξη των μεθόδων (recipes) στο συγκεκριμένο πεδίο εφαρμογής.

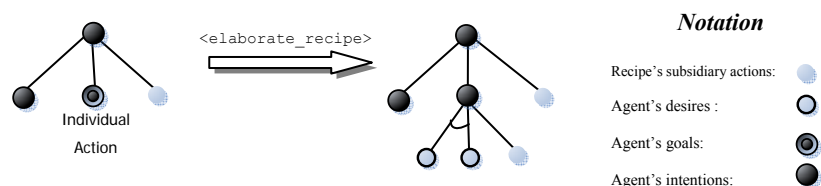


Όπως με τη `<select_recipe>` έτσι και με τη `<elaborate_recipe>` οι μέθοδοι που ο πράκτορας μπορεί να χρησιμοποιήσει για να αναπτύξει τις μεθόδους των ενεργειών (ενός συγκεκριμένου πεδίου εφαρμογής) μπορεί να διαφέρουν μεταξύ τους. Αυτοί οι μέθοδοι υλοποιούν την `<elaborate_recipe>` είτε ως σύνθετη ενέργεια είτε ως βασική, καθώς επίσης είτε ως ατομική (individual) είτε ως κοινή (shared). Ανεξαρτήτως της μεθόδου που ο πράκτορας θα επιλέξει για την `<elaborate_recipe>`, η λίστα των αποτελεσμάτων της (effect list) οδηγούν τον πράκτορα να δημιουργεί επιθυμίες για να πραγματοποιήσει τις υποενέργειες που περιλαμβάνει η επιλεγμένη μέθοδος της Action.

Ανάλογα με το πώς ο πράκτορας αναπτύσσει και υλοποιεί τις νοητικές ενέργειες `<select_recipe>` και `<elaborate_recipe>` (που προκύπτουν από κάθε μία επιθυμία) χρησιμοποιώντας το μηχανισμό ελέγχου της BDI-Πρακτικής Συλλογιστικής η συμπεριφορά του πράκτορα μεταβάλλεται από αντανakλαστική σε στοχαστική (και αντιστρόφως) και από ατομική σε κοινωνική (και αντιστρόφως).

Γενικά, ξεχωρίζουμε δύο διαφορετικές περιπτώσεις ανάπτυξης της μεθόδου μίας ενέργειας (και άρα δύο διαφορετικές μεθόδους υλοποίησης της `<elaborate_recipe>`) ανάλογα με το αν η ενέργεια είναι ατομική (individual) ή κοινή (shared).

Αν η ενέργεια είναι ατομική τότε ο πράκτορας θεωρεί την `<elaborate_recipe>` ως βασική ενέργεια (βλ. τη μέθοδο (α) του Πίνακα 6 του Παραρτήματος Β) και ενεργεί ως ακολούθως: Σύμφωνα με τις υπό-ενέργειες που το όρισμα `ActionList` της μεθόδου της ενέργειας περιέχει, επιλέγει την επόμενη υπό-ενέργεια ή τις υπό-ενέργειες που πρέπει να πραγματοποιήσει. Για τις ενέργειες αυτές σχηματίζει από μία επιθυμία να τις πραγματοποιήσει. Αποτέλεσμα του σχηματισμού των επιθυμιών αυτών είναι η ενεργοποίηση του μηχανισμού ελέγχου της BDI-πρακτικής συλλογιστικής του πράκτορα. Στη περίπτωση που ο πράκτορας έχει αναπτύξει τα πλάνα όλων των υπό-ενεργειών της μεθόδου τότε πιστεύει ότι έχει βρει ένα πλήρες πλάνο για την ενέργεια αυτή.



Εικόνα 31: Ανάπτυξη μίας προσωπικής ενέργειας A.

Στη δεύτερη περίπτωση όπου ο πράκτορας έχει δεσμευθεί να πραγματοποιήσει την ενέργεια σε συνεργασία με άλλους πράκτορες (βλ. τη μέθοδο (β) Πίνακα 6 του Παραρτήματος Β), τότε η νοητική ενέργεια `<elaborate_recipe>` θεωρείται σύνθετη ενέργεια η οποία υλοποιείται από τις νοητικές υπό-ενέργειες: `<group_formation>`, `<recipe_negotiation>`, και `<task_allocation>`. Αυτές επεκτείνουν το μερικώς αναπτυγμένο κοινό πλάνο της `Action` σε πλήρες. Οι νοητικές αυτές ενέργειες καθιστούν τον πράκτορα ικανό να σχηματίζει την ομάδα συνεργασίας, να διαπραγματεύεται για τη μέθοδο που όλοι μαζί θα ακολουθήσουν, καθώς και τον τρόπο με τον οποίο θα μοιράσουν μεταξύ τους τις υπό-ενέργειες της `Action`.

Η πραγματοποίηση από τον πράκτορα των τριών αυτών υπό-ενεργειών της `<elaborate_recipe>` έχουν ως αποτέλεσμα την αλλαγή της νοητικής κατάστασης τόσο του ίδιου του πράκτορα όσο και των συνεργατών του. Μετά την επιτυχή εκτέλεση της νοητικής ενέργειας `<group_formation>` όλοι οι πράκτορες σχηματίζουν τη πρόθεση να πραγματοποιήσουν την ενέργεια συνεργατικά (commitment to group activity). Έχοντας αποφασίσει τη μέθοδο που θα ακολουθήσουν από κοινού, δηλαδή μετά την επιτυχή εκτέλεση της `<recipe_negotiation>`, όλοι οι πράκτορες μοιράζονται το κοινό στόχο έχοντας σχηματίσει ο καθένας από μία δομή `<goal_to>` για την ενέργεια. Τέλος, έχο-

ντας αποφασίζει τη κατανομή των ενεργειών στα μέλη της ομάδας, δηλαδή έχοντας πραγματοποιήσει επιτυχώς τη νοητική ενέργεια `<task_allocation>`, ο κάθε πράκτορας της ομάδας μοιράζεται το (ίσως μερικώς αναπτυγμένο) πλάνο της `Action`. Δηλαδή, όλοι οι πράκτορες έχοντας εκτελέσει επιτυχώς όλες τις υποενέργειες της `<elaborate_recipe>` μοιράζονται τη κοινή πρόθεση `<int_to>` να πραγματοποιήσουν την `Action`. χρησιμοποιώντας την κοινή μέθοδο.

Στην ενότητα που ακολουθεί, παρουσιάζουμε τις τρεις αυτές νοητικές ενέργειες ως μέρος της συνολικής συνεργατικής συμπεριφοράς του πράκτορα στο πλαίσιο εργασίας ICAGENT.

## Κεφάλαιο 5ο:

# Η Δραστηριότητα της Συνεργασίας στο Πλαίσιο Εργασίας του ICAGENT

Η Εικόνα 32 συνοψίζει τα τέσσερα στάδια της συνεργασίας μεταξύ πρακτόρων [Wooldridge, 1999]. Τα στάδια αυτά είναι: (α) η αναγνώριση της ανάγκης για συνεργασία (recognition of the potential for collaboration), (β) ο σχηματισμός της ομάδας των συνεργατών (group formation), (γ) η διαπραγμάτευση της μεθόδου που θα ακολουθήσουν από κοινού με τους άλλους πράκτορες της ομάδας (recipe negotiation), (δ) ο καταμερισμός των εργασιών της μεθόδου στους πράκτορες της ομάδας (task allocation) και (ε) η υλοποίηση των δεσμεύσεων του πράκτορα (intention realization). Τα στάδια αυτά υλοποιούνται από κάθε πράκτορα ξεχωριστά και συντονισμένα μεταξύ τους μέσω συλλογιστικών διεργασιών και νοητικών ενεργειών που θα παρουσιάσουμε στις επόμενες ενότητες.



Εικόνα 32: Τα τέσσερα στάδια της συνεργασίας μεταξύ πρακτόρων.

Στο πλαίσιο εργασίας του ICAGENT τα στάδια αυτά δεν πραγματοποιούνται πάντα σειριακά. Για παράδειγμα, αν δύο (ή περισσότεροι) πράκτορες αναγνωρίσουν την ανάγκη για συνεργασία προκειμένου να πραγματοποιήσουν από κοινού μία ενέργεια  $a$  τότε αρχικά ο κάθε πράκτορας (και παράλληλα ο ένας με τον άλλο) ακολουθεί σειριακά τα βήματα αυτά. Αν όμως ένας από τους δύο δεν μπορέσει να πραγματοποιήσει μία υπό-ενέργεια της κοινής ενέργειας  $a$  και χρειαστεί ν' αναθεωρήσει τη μέθοδο που έχει επιλέξει για αυτήν τότε τα βήματα (α) και (β) παραλείπονται (διότι θεωρεί ότι έχουν ήδη πραγματοποιηθεί) και προσπαθεί πάλι να διαπραγματευτεί μόνο τη νέα μέθοδο (και μετά αν ο δεύτερος πράκτορας συμφωνήσει να εγκαταλείψουν την παλιά μέθοδο, διαπραγματεύονται πάλι τη κατανομή των υπό-ενεργειών μεταξύ τους κτλ.).

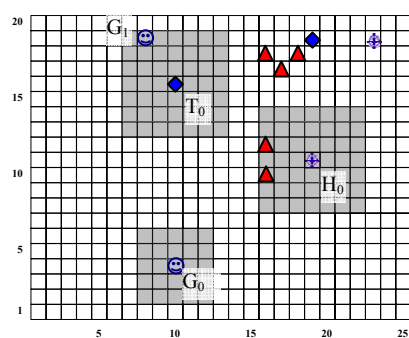


### 5.1. Αναγνώριση πιθανών ευκαιριών συνεργασίας

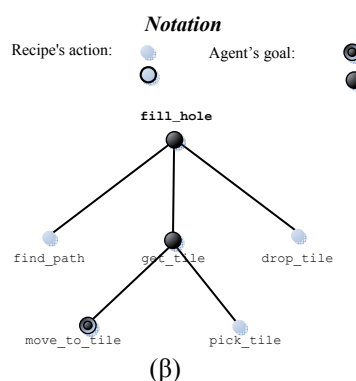
Σε αυτό το στάδιο της συνεργασίας οι πράκτορες αναγνωρίζουν το ενδεχόμενο να συνεργαστούν για να πραγματοποιήσουν από κοινού μία ενέργεια. Ένας πράκτορας αποφασίζει να πραγματοποιήσει μία ενέργεια σε συνεργασία με έναν ή περισσότερους πράκτορες σε δύο περιπτώσεις:

Η πρώτη περίπτωση προκύπτει όταν ο πράκτορας αναγνωρίζει ότι η σχετική μέθοδος που έχει επιλέξει για την ενέργεια πρέπει να πραγματοποιηθεί συνεργατικά διότι πιστεύει ότι κάτι τέτοιο θα εξυπηρετήσει καλύτερα του σκοπούς του. Η περίπτωση συνεργασίας αυτή προκύπτει όταν η συνθήκη συνεργασίας *Collaboration\_Mental\_Condition* της αντίστοιχης μεθόδου αληθεύει.

Για παράδειγμα, στο *Tileworld*, καθ' όσον ο πράκτορας μετακινείται στο κύβο  $T_0$  και πριν φτάσει σ' αυτόν αντιλαμβάνεται (μέσω των αισθητήρων εντοπισμού γεγονότων) ότι ένας άλλος πράκτορας  $G_1$  βρίσκεται κοντά στο κύβο (βλ. Εικόνα 33(α)). Ο εντοπισμός του καινούργιου πράκτορα έχει ως αποτέλεσμα να αλλάξουν οι κατευθύνσεις συμπεριφοράς της βασικής ενέργειας *<move\_to\_tile>*. Οι νέες κατευθύνσεις οδηγούν το πράκτορα να εντοπίσει την ανάγκη για συνεργασία για την ενέργεια *<move\_to\_tile>* ή για οποιαδήποτε από τις ενέργειες που περιλαμβάνουν την υπό-ενέργεια αυτή (γονικές ενέργειες στο πλαίσιο ενεργειών).



(α)



(β)

Εικόνα 33: Ο πράκτορας  $G_0$  συνεργάζεται με το πράκτορα  $G_1$  για να γεμίσει τη τρύπα  $H_0$ .

Έτσι ελέγχοντας τις νοητικές συνθήκες *Collaboration\_Mental\_Condition* της μεθόδου της ενέργειας *<fill\_hole>*, ο πράκτορας  $G_0$  αντιλαμβάνεται την ανάγκη να συνεργαστεί για να πετύχει πιο γρήγορα το βασικό του στόχο. Ελέγχοντας τις συνθήκες

αυτές ο πράκτορας επικοινωνεί με τον πράκτορα  $G_1$  για να λάβει τις αναγκαίες πληροφορίες για τις δυνατότητές και τους στόχους του. Αν ο πράκτορας  $G_0$  αναγνωρίσει ότι ο  $G_1$  μπορεί να βοηθήσει στη πραγματοποίηση της ενέργειας `<fill_hole>` τότε δημιουργεί μία δέσμευση να πραγματοποιήσει την ενέργεια αυτή σε συνεργασία με αυτόν (δηλαδή, σχηματίζει τη δέσμευση της μορφής `int_that(Agent, do(Null, Action, Null))`)<sup>30</sup>.

Η δεύτερη περίπτωση προκύπτει όταν ο πράκτορας εντοπίσει μία σύγκρουση στο πλαίσιο ενεργειών του (context of actions) την οποία δεν μπορεί να επιλύσει μόνος του. Σ' αυτή τη περίπτωση η δέσμευση να πραγματοποιήσει μία ενέργεια σε συνεργασία με άλλους πράκτορες προκύπτει από τη νοητική ενέργεια `<resolve_conflict>` καθοδηγούμενη από τη κατεύθυνση συμπεριφοράς επίλυσης συγκρούσεων "collaborate". Για παράδειγμα, στο Tileworld, μία τέτοια περίπτωση εμφανίζεται όταν ο πράκτορας αντιλαμβάνεται ότι δεν μπορεί να μετακινήσει ένα εμπόδιο (δηλαδή δεν μπορεί να πραγματοποιήσει την ενέργεια `<get_tile>`). Αν ο πράκτορας εντοπίσει κάποιον άλλον πράκτορα κοντά στο εμπόδιο και αναγνωρίσει ότι αυτός έχει τη δυνατότητα να το μετακινήσει, τότε ο πρώτος σχηματίζει τη δέσμευση να πραγματοποιήσει την ενέργεια `<get_tile>` συνεργατικά με τον δεύτερο πράκτορα.

Και στις δύο παραπάνω περιπτώσεις, μετά την αναγνώριση της ανάγκης για συνεργασία ο πράκτορας δεσμεύεται να συνεργαστεί με έναν ή περισσότερους πράκτορες, δηλαδή σχηματίζει τη δέσμευση `int_that(Agent, do(Null, Action, Null))`. Η δέσμευση αυτή αναπαριστά την ανάγκη του πράκτορα για συνεργασία και επηρεάζει τον τρόπο με τον οποίο αναπτύσσει τα πλάνα του (όπως αναφέραμε στην υποενότητα 4.2.4). Πιο συγκεκριμένα, έχοντας δεσμευθεί να πραγματοποιήσει μία ενέργεια συνεργατικά, ο πράκτορας δεν προσπαθεί ν' αναπτύξει τη μέθοδο της ενέργειας μόνος του, αλλά εκτελώντας τις νοητικές υπό-ενέργειες `<group_formation>`, `<recipe_negotiation>` και `<task_allocation>` της `<elaborate_recipe>` προσπαθεί να

---

<sup>30</sup> Οι παράμετροι `Group` και `Recipe` της δέσμευσης `<int_that_do>` δεν έχουν πάρει κάποια τιμή σε αυτό το στάδιο της συνεργασίας και γ' αυτό έχουν τη τιμή `Null`. Όπως θα δούμε αργότερα, οι μεταβλητές αυτές καθορίζονται μετά την εκτέλεση των νοητικών ενεργειών `<group_formation>` και `<recipe_negotiation>`.

βρει τους πράκτορες που θα συνεργαστεί, να διαπραγματευτεί για τη μέθοδο που θα ακολουθήσουν και τέλος για τη κατανομή των ενεργειών μεταξύ τους.

## 5.2. Σχηματισμός της ομάδας

Ο σχηματισμός της ομάδας `<group_formation>` είναι μία (νοητική) υπό-ενέργεια της `<elaborate_recipe>` που ο πράκτορας πραγματοποιεί όταν έχει δεσμευθεί να πραγματοποιήσει την ενέργεια συνεργατικά (δηλαδή όταν έχει σχηματίσει τη δέσμευση `int_that(Agent, do(Null, Action, Null))`). Με την ενέργεια αυτή επικοινωνεί με έναν ή περισσότερους πράκτορες προκειμένου να καθορίσει την ομάδα (Group) συνεργασίας.

Αρχικώς (by default) η νοητική ενέργεια `<group_formation>` θεωρείται ως ατομική η οποία χρησιμοποιεί γνώση από τη λίστα καταμερισμού εργασιών `Task_Allocation` της μεθόδου της ενέργειας (βλ. υποενότητα 4.1.2). Γνωρίζοντας ο πράκτορας τα ids των πιθανών μελών της ομάδας τους πληροφορεί για τη ανάγκη να συνεργαστούν μαζί του προκειμένου να πραγματοποιήσουν μαζί την ενέργεια (Action) και περιμένει για την απάντησή τους.

Κάθε πράκτορας (ας τον ονομάσουμε `GrAgent`) που αποδέχεται την αίτηση από τον πράκτορα (Agent) για συνεργασία θεωρεί την ενέργεια (Action) ως κοινή (shared) ενέργεια με αποτέλεσμα να αλλάζει τη νοητική του κατάσταση ως ακολούθως: (α) Σχηματίζει την επιθυμία να πραγματοποιήσει την Action (`desire_to(GrAgent, Action)`) και (β) δεσμεύεται να κάνει αυτό συνεργατικά (`int_that(GrAgent, do(Null, Action, Null))`).

Η επιθυμία έχει ως αποτέλεσμα κάθε πράκτορας να ενεργοποιεί το BDI-CPR μηχανισμό με τον οποίο διαλέγει μία σχετική και εφαρμόσιμη μέθοδο (ίσως διαφορετική από αυτή που έχει επιλέξει ο Agent) για την Action. Επιπλέον, η νοητική δομή `<int_that_do>` καθοδηγεί τη συνεργατική συμπεριφορά των πρακτόρων οδηγώντας κάθε πράκτορα να πληροφορεί τους συνεργάτες του για κάθε αλλαγή που συμβαίνει στην ομάδα. Έτσι στη περίπτωση που ένας πράκτορας εισέρχεται ή φεύγει από την

ομάδα οι λοιποί πράκτορες της ομάδας επανασχεδιάζουν το πλάνο της ενέργειας πραγματοποιώντας πάλι τις νοητικές ενέργειες<sup>31</sup> `<group_formation>`, `<recipe_negotiation>` και `<task_allocation>`. Αναπτύσσοντας ξανά τη μέθοδο για την Action οι πράκτορες επικοινωνούν (μόνο αν είναι αναγκαίο) για να επανα-σχηματίσουν την ομάδα, για να επανα-διαπραγματευτούν τη μέθοδο που ο καθένας έχει επιλέξει καθώς και για να επανα-κατανεύμουν τις υπό-ενέργειες της Action. Έτσι η ομάδα των πρακτόρων μπορεί να αλλάζει δυναμικά κατά τη διάρκεια της συνεργασίας των πρακτόρων.

Κάθε πράκτορας της ομάδας, μετά την επιτυχή εκτέλεση της νοητικής ενέργειας `<group_formation>` σχηματίζει την αμοιβαία πεποίθηση `mb(Group, int_that(Agent, do(Group, Action, Null)))`. Η νοητική κατάσταση αυτή είναι ο στόχος της μεθόδου της ενέργειας `<group_formation>`. Στη περίπτωση που μετά την εκτέλεση της νοητικής ενέργειας αυτής ο στόχος της μεθόδου της δεν ισχύσει, τότε η συνεργασία του πράκτορα με τους άλλους πράκτορες αποτυγχάνει με αποτέλεσμα ο πράκτορας να προσπαθήσει να επιλύσει τη σύγκρουση που προκύπτει.

Στο πεδίο εφαρμογής του Tileworld, ο πράκτορας  $G_0$  αιτείται από τον  $G_1$  να συνεργαστούν προκειμένου να πραγματοποιήσουν μαζί την ενέργεια `<fill_hole>`. Μετά την αποδοχή της αίτησης από τον  $G_1$  και την επιτυχή εκτέλεση της `<group_formation>` και από τους δύο πράκτορες η ενέργεια `<fill_hole>` θεωρείται κοινή διότι και οι δύο πράκτορες έχουν την αμοιβαία πεποίθηση ότι έχουν δεσμευθεί να πραγματοποιήσουν την ενέργεια συνεργατικά.

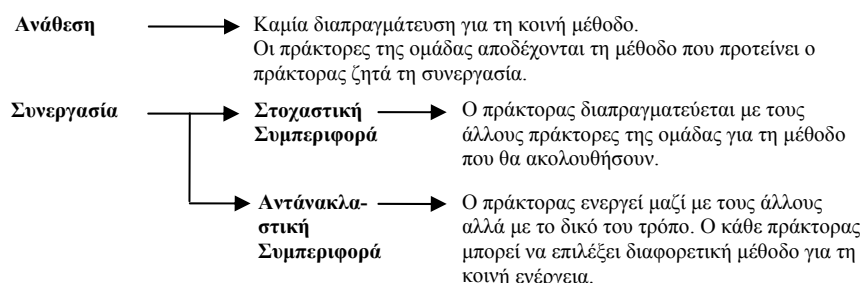
---

<sup>31</sup> Παρατηρείστε ότι οι νοητικές ενέργειες `<recipe_negotiation>` και `<task_allocation>` πραγματοποιούνται από τους πράκτορες μόνο όταν αυτό είναι αναγκαίο, δηλαδή όταν οι πράκτορες πιστεύουν ότι κάτι έχει αλλάξει στη μέθοδο που ακολουθούν ή στη κατανομή των εργασιών μεταξύ τους.

### 5.3. Διαπραγμάτευση μεθόδων

Αφού καθοριστεί η ομάδα συνεργασίας, ο κάθε πράκτορας της ομάδας πραγματοποιεί τη νοητική υπό-ενέργεια (της `<elaborate_recipe>`) `<recipe_negotiation>` προκειμένου να καθοριστεί η κοινή μέθοδος που θ' ακολουθήσει η ομάδα για να πραγματοποιήσει τη κοινή ενέργεια. Μετά τη διαπραγμάτευση της κοινής μεθόδου ο κάθε πράκτορας έχει ως στόχο τη πραγματοποίηση της κοινής ενέργειας (δηλαδή ο κάθε πράκτορας `Agent` σχηματίζει το νοητική δομή `goal_to(Agent, Action)`) και αμοιβαία πιστεύει ότι έχει δεσμευθεί να πραγματοποιήσει την κοινή ενέργεια συνεργατικά με τα λοιπά μέλη της ομάδας ακολουθώντας τη κοινή αυτή μέθοδο (δηλαδή για κάθε πράκτορα ισχύει `mb(Group, int_that(Agent, do(Group, Action, Recipe)))`).

Η υλοποίηση της νοητικής ενέργειας της διαπραγμάτευσης μεθόδων εξαρτάται από τον τύπο της αλληλεπίδρασης που ο κάθε πράκτορας έχει επιλέξει να έχει με τους άλλους πράκτορες της ομάδας προκειμένου να πραγματοποιήσει τις υπό-ενέργειες της μεθόδου της `Action` (αυτό καθορίζεται από το όρισμα `Task_Allocation` της μεθόδου της `Action`) καθώς επίσης και από τον τύπο της συμπεριφοράς που ο πράκτορας έχει επιλέξει για να ελέγξει την εφαρμοσιμότητα της `Action` (βλ. Εικόνα 34).



*Εικόνα 34:* Η συμπεριφορά του πράκτορα εξαρτάται από: (α) την απόφασή του για το αν θα αναθέσει τις υποενέργειες της μεθόδου στους άλλους πράκτορες ή αν θα συνεργαστεί μαζί τους, και (β) από την απόφασή του για το αν θα συμπεριφερθεί στοχαστικά ή αντανάκλαστικά για τη συγκεκριμένη ενέργεια.

Πιο συγκεκριμένα, στη περίπτωση που ο πράκτορας έχει αποφασίσει να *αναθέσει* (delegate) μία υπό-ενέργεια σε έναν πράκτορα ή σε μία υπό-ομάδα πρακτόρων (Sub-Group) τότε ο πράκτορας αιτείται από αυτούς να αποδεχθούν τη μέθοδο που έχει ήδη επιλέξει για την `Action` χωρίς καμία διαπραγμάτευση. Κάθε πράκτορας του `SubGroup` είτε αποδέχεται τη μέθοδο που ο πρώτος πράκτορας προτείνει είτε αποχωρεί από την

ομάδα (`Group`) ενημερώνοντας τους υπόλοιπους για την επιλογή του αυτή. Η υπό-ομάδα (`SubGroup`) είναι υπεύθυνη να βρει τη κατάλληλη σχετική και εφαρμόσιμη μέθοδο για την υπό-ενέργεια που έχει αναλάβει να πραγματοποιήσει.

Αντίθετα, αν ο αρχικός πράκτορας έχει αποφασίσει να *συνεργαστεί* (`collaborate`) με ένα υποσύνολο (`SubGroup`) πρακτόρων της ομάδας για να πραγματοποιήσει από κοινού μία υπό-ενέργεια, τότε ο πράκτορας δεν αναθέτει τις υπό-ενέργειες της μεθόδου που έχει επιλέξει στους άλλους πράκτορες αλλά απλώς ενημερώνει-προτείνει τη μέθοδο που έχει επιλέξει για την `Action`. Ο τύπος της συμπεριφοράς που ο κάθε πράκτορας θα ακολουθήσει για την κοινή ενέργεια `Action` εξαρτάται από τις κατευθύνσεις συμπεριφοράς που θα επιλέξει για τη συγκεκριμένη μέθοδο της `Action`.

Στη περίπτωση που ένας πράκτορας (`GrAgent`) της ομάδας (`Group`) έχει αποφασίσει να συμπεριφερθεί στοχαστικά<sup>32</sup> (`deliberatively`), τότε δεσμεύεται να βοηθήσει την ομάδα στο να πραγματοποιήσει την ενέργεια (`int_that(GrAgent, cba(Group, Action, Recipe))`) και διαπραγματεύεται τόσο τη κοινή μέθοδο που θα ακολουθηθεί όσο και τη κατανομή των εργασιών μεταξύ τους (`social deliberation`). Διαφορετικά, ο πράκτορας ενεργεί αντανακλαστικά χωρίς να δεσμεύεται ότι θα βοηθήσει τους άλλους σε ότι αφορά τη δυνατότητα τους να πραγματοποιήσουν την ενέργεια αυτή. Στη τελευταία περίπτωση, ο πράκτορας δρα μαζί με τους άλλους πράκτορες της ομάδας για να πραγματοποιήσουν από κοινού την ενέργεια αλλά ακολουθώντας το «δικό του τρόπο» (δηλαδή, τη δικιά του μέθοδο). Παρ' όλο μία τέτοια αντανακλαστικού τύπου συμπεριφορά μέσα σε μία ομάδα είναι επικίνδυνη διότι μπορεί να οδηγήσει σε προβληματικό συντονισμό μεταξύ των πρακτόρων, μπορεί να είναι ωφέλιμη σε περιπτώσεις που οι πόροι του συστήματος είναι περιορισμένοι και κατά τη διάρκεια του χρόνου εξαντλούνται.

Στη περίπτωση όπου ο πράκτορας επιλέξει να στοχαστεί για τη μέθοδο που έχει επιλέξει, όπως παρατηρήσαμε στην υπό-ενότητα 4.2.3, η νοητική δομή `<int_that_cba>` τον αναγκάζει να ελέγξει την εφαρμοσιμότητα της μεθόδου (`reconcile the recipe`) και

---

<sup>32</sup> Δηλαδή, οι κατευθύνσεις συμπεριφοράς της μεθόδου που έχει επιλέξει δεν περιλαμβάνουν τη κατεύθυνση `react`.

(στη περίπτωση του κοινωνικού στοχασμού) να διαπραγματευτεί τη μέθοδο που έχει επιλέξει με τους άλλους πράκτορες της ομάδας.

Αν ο πράκτορας δεν εντοπίσει κάποια σύγκρουση τότε η νοητική δομή `<int_that_cba>` οδηγεί τους πράκτορες να διαπραγματευτούν (μέσω της νοητικής ενέργειας `<recipe_negotiation>`) για μία κοινή μέθοδο για την `Action`.

Ένας πράκτορας συμφωνεί ν' ακολουθήσει μία μέθοδο που προτείνει ενός άλλος πράκτορας (π.χ. τη μέθοδο του πράκτορα που αρχίζει τη συνεργασία) αν και μόνο αν ο στόχος, η λίστα ενεργειών και η λίστα των αποτελεσμάτων της μεθόδου αυτής μπορούν να ενοποιηθούν με τα αντίστοιχα της μεθόδου που έχει επιλέξει ο ίδιος. Επιπλέον, η ένωση των περιορισμών των πλαισίων ενεργειών όλων των συνεργατών πρέπει να είναι συνεπής με τους περιορισμούς εφαρμογής της μεθόδου. Κάτω από αυτό το κοινό πλαίσιο ενεργειών ο πράκτορας πρέπει να πραγματοποιήσει τις υπό-ενέργειες (που αντιστοιχούν σε αυτόν) της `Action`.

Στη περίπτωση που δημιουργηθεί μία σύγκρουση και ο πράκτορας δεν μπορεί να την επιλύσει ή στη περίπτωση που οι πράκτορες μεταξύ τους δεν μπορούν να συμφωνήσουν σε μία κοινή μέθοδο τότε η συνεργασία μεταξύ των πρακτόρων διακόπτεται και ο πράκτορας που άρχισε τη συνεργασία προσπαθεί να επιλέξει μία εναλλακτική μέθοδο. Η καινούργια μέθοδος μπορεί να οδηγήσει τους πράκτορες σε οποιοδήποτε τύπο συμπεριφοράς.

Στο περιβάλλον του `Tileworld`, μετά το σχηματισμό της ομάδας, οι πράκτορες  $G_0$  και  $G_1$  διαπραγματεύονται για τη μέθοδο που θα ακολουθήσουν προκειμένου να πραγματοποιήσουν τη κοινή ενέργεια `fill_hole`. Στη περίπτωση που ο πράκτορας  $G_1$  δεν συμφωνήσει με τη μέθοδο που ο  $G_0$  προτείνει (για παράδειγμα διότι πιστεύει ότι ο πράκτορας  $G_1$  δεν έχει κάποιο μονοπάτι προς το κύβο) τότε η συνεργασία μεταξύ τους τερματίζεται και ο  $G_1$  συνεχίζει να ενεργεί ανεξάρτητα από το πράκτορα  $G_0$ .

#### 5.4. Καταμερισμός εργασιών

Οι πράκτορες της ομάδας, ενεργώντας συνεργατικά, κατανέμουν τις εργασίες της κοινής μεθόδου μεταξύ τους. Η νοητική ενέργεια `<task_allocation>` καθιστά ικανούς τους πράκτορες ν' αποφασίζουν για τον τρόπο με τον οποίο θα γίνει η κατανομή αυτή.

Όπως αναφέραμε παραπάνω η `<task_allocation>` είναι μία υπό-ενέργεια της σύνθετης νοητικής ενέργειας `<elaborate_recipe>` και εκτελείται από κάθε πράκτορα αφού έχει συμφωνηθεί πρώτα η κοινή μέθοδος που θα ακολουθήσουν (δηλαδή, όταν κάθε πράκτορας έχει την αμοιβαία πεποίθηση `mb(Group, int_that(Agent, do(Group, Action, Recipe))` όπου τα ορίσματα `Group` και `Recipe` είναι ορισμένα).

Για να πραγματοποιήσει ένας πράκτορας τη νοητική ενέργεια `<task_allocation>` εκμεταλλεύεται τη πληροφορία του ορίσματος `Task_Allocation` της σχετικής και εφαρμόσιμης μεθόδου που έχει επιλέξει για την ενέργεια (`Action`). Επιπλέον, η πραγματοποίηση της νοητικής αυτής ενέργειας εξαρτάται από τον τύπο της αλληλεπίδρασης (με τους άλλους πράκτορες της ομάδας) που έχει επιλέξει για κάθε μία υπό-ενέργεια της μεθόδου (βλ. τη δομή του ορίσματος `Task_Allocation` στην υποενότητα 4.1.2).

Οι πράκτορες στους οποίους **αναθέτει** τη πραγματοποίηση μίας ή περισσότερες υπό-ενέργειες της μεθόδου (*delegated agents*) δεν συμμετέχουν στη διαπραγμάτευση της κατανομής των εργασιών της κοινής ενέργειας. Οι πράκτορες αυτοί είτε αποδέχονται τις εργασίες (υπό-ενέργειες) που τους έχουν ανατεθεί ή εγκαταλείπουν την ομάδα ενημερώνοντας τους άλλους πράκτορες για την απόφασή τους αυτή.

Στη περίπτωση της **συνεργασίας**, ο τύπος της συμπεριφοράς που ο κάθε πράκτορας `GrAgent` ακολουθεί (σε ότι αφορά τη κατανομή των εργασιών), εξαρτάται από τις κατευθύνσεις συμπεριφοράς (`Behaviour_Directives`) της μεθόδου που έχει επιλέξει για την `Action`.

Αν ο πράκτορας `GrAgent` επιλέξει να συμπεριφερθεί με αντανακλαστικό τρόπο τότε δεν συμμετέχει στη διαπραγμάτευση της κατανομής των υπό-ενεργειών της `Action` με τους πράκτορες της ομάδας, παρά μόνο σχηματίζει μία επιθυμία για την επόμενη υπό-ενέργεια που πρέπει να πραγματοποιήσει και μία δέσμευση ότι θα συνεργαστεί με τους άλλους πράκτορες για να πραγματοποιήσει την ενέργεια αυτή.

Από την άλλη πλευρά, αν ο πράκτορας `GrAgent` αποφασίσει να συμπεριφερθεί στοχαστικά, τότε διαπραγματεύεται τη κατανομή των εργασιών με τους άλλους πράκτορες της ομάδας (*social deliberation*). Όταν όλοι οι πράκτορες συμφωνήσουν σε μία κοινή κατανομή τότε ο πράκτορας σχηματίζει την επιθυμία να πραγματοποιήσει την επόμενη ενέργεια που πρέπει να πραγματοποιήσει. Επίσης, για κάθε μία υπό-ενέργεια της μεθό-



δου της `Action` (που είτε συμμετέχει στη πραγματοποίηση αυτής είτε όχι) δημιουργεί μία δέσμευση του τύπου `<int_that_cba>` έτσι ώστε να έχει τη δυνατότητα να βοηθήσει τους συνεργάτες του όποτε παραστεί ανάγκη (δηλαδή σχηματίζει τη δέσμευση `int_that( GrAgent, cba(SubGroup, SubAction, Null))`). Ο σχηματισμός αυτών των δεσμεύσεων εξαρτάται από τις κατευθύνσεις συμπεριφοράς που ο πράκτορας έχει επιλέξει για τη συγκεκριμένη μέθοδο.

Ο τελευταίος τύπος δέσμευσης οδηγεί τον πράκτορα να παρέχει βοήθεια στους συνεργάτες του με άμεσο ή έμμεσο τρόπο ακόμα και όταν αυτός δεν συμμετέχει στην υλοποίηση της υπό-ενέργειας. Για παράδειγμα, αν ο πράκτορας αντιληφθεί ότι ο συνεργάτης του δεν μπορεί να πετύχει μία εργασία της κοινής μεθόδου και γνωρίζει μία ενέργεια που μπορεί να τον καταστήσει ικανό τότε δημιουργεί μία επιθυμία να πραγματοποιήσει την ενέργεια αυτή (άμεσος τρόπος βοήθειας). Αντίθετα, ένας έμμεσος τρόπος βοήθειας είναι να μην πραγματοποιήσει ο πράκτορας κάποια ενέργεια που να έρθει σε σύγκρουση με την υπό-ενέργεια που προσπαθεί να πραγματοποιήσει ο συνεργάτης του.

Η επιτυχής εκτέλεση της νοητικής ενέργειας του καταμερισμού των υπό-ενεργειών μίας μεθόδου ολοκληρώνει την υλοποίηση της νοητικής ενέργειας `<elaborate_recipe>`. Έτσι, το κάθε μέλος της ομάδας αμοιβαία πιστεύει ότι η ομάδα έχει δεσμευθεί να πραγματοποιήσει τη κοινή ενέργεια (`mb(Group, int_to(Agent, Action))`).

Επιστρέφοντας στο παράδειγμα του `Tileworld`, οι πράκτορες  $G_0$  και  $G_1$  πραγματοποιούν τις υπό-ενέργειες της `<fill_hole>` ως εξής (βλ. Εικόνα 33(β)): η υπό-ενέργεια `<find_path>` πραγματοποιείται από το καθένα ταυτόχρονα και ανεξάρτητα ο ένας από τον άλλον, η υπό-ενέργεια `<get_tile>` πραγματοποιείται συνεργατικά, ενώ τελικώς ο πράκτορας  $G_0$  πραγματοποιεί μόνος του την ενέργεια `<drop_tile>`.

Και οι δύο πράκτορες δεσμεύονται να παρέχουν βοήθεια ο ένας στον άλλον για όλες τις υπό-ενέργειες της `ActionList` της μεθόδου `<fill_hole>`. Δηλαδή, οι πράκτορες σχηματίζουν μία δέσμευση της μορφής `<int_that_cba>` για κάθε μία από τις υπό-ενέργειες αυτές.

Τη στιγμή που η συνεργασία μεταξύ των δύο αρχίζει, λόγω του ότι ο πράκτορας  $G_0$  έχει ήδη βρει ένα μονοπάτι προς το κύβο· ο πράκτορας δεν ξανά-εκτελεί την ενέργεια

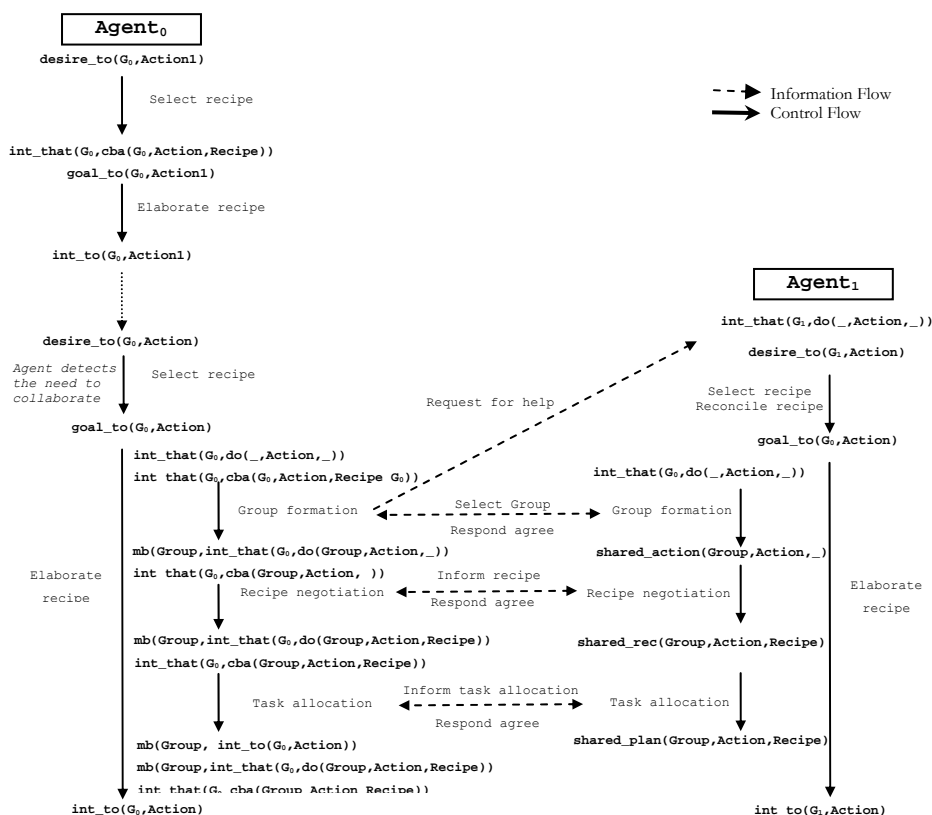
`<find_path>` (διότι ο στόχος της μεθόδου αυτής έχει πραγματοποιηθεί). Από την άλλη πλευρά ο πράκτορας  $G_1$  σχηματίζει μία επιθυμία να πραγματοποιήσει την υπό-ενέργεια αυτή. Λόγω του ότι έχει ήδη δεσμευθεί να βοηθήσει τον  $G_0$  στη πραγματοποίηση της `<get_tile>` προσπαθεί να βρει το συντομότερο μονοπάτι πρώτα προς το κύβο και μετά προς το  $G_0$ .

Για τη κοινή ενέργεια `<get_tile>` ο κάθε πράκτορας (ανεξάρτητα ο ένας από τον άλλο) επιλέγει τη μέθοδο ( $\beta$ ) του Πίνακα 9. Η μέθοδος αυτή είναι σχετική μόνο στη περίπτωση που και οι δύο πράκτορες έχουν δεσμευθεί να πραγματοποιήσουν την ενέργεια σε συνεργασία. Η `ActionList` της μεθόδου περιέχει δύο ζεύγη ενεργειών όπου πραγματοποιούνται παράλληλα από τους δύο πράκτορες: οι πρώτες δύο ενέργειες `<move_to>` και `<pick_tile>` πραγματοποιούνται σειριακά από τον  $G_0$ , ενώ οι `<move_to>` και `<push_tile>` πραγματοποιούνται σειριακά από τον  $G_1$ . Έτσι καθώς ο πράκτορας  $G_0$  κινείται προς το κύβο, ο πράκτορας  $G_1$  προσπαθεί να μετακινηθεί προς το κύβο και μετά να το σπρώξει προς τον  $G_0$ .

## 5.5. Υλοποίηση προθέσεων

Συνοψίζοντας τα παραπάνω βήματα της δραστηριότητας της συνεργασίας μεταξύ πρακτόρων, ένας πράκτορας μπορεί είτε ν' αναθέσει μία ενέργεια σ' έναν άλλο πράκτορα, είτε να συνεργαστεί με άλλους πράκτορες στοχεζόμενος κοινωνικά για ενέργειες που πρέπει να τις πραγματοποιήσει μαζί με άλλους πράκτορες, είτε απλά να δράσει αντανακλαστικά προσπαθώντας να πραγματοποιήσει ενέργειες μαζί με άλλους πράκτορες χωρίς όμως να συμμετέχει στο σχεδιασμό των πλάνων.

Όπως δείχνει η Εικόνα 35, στη περίπτωση που ένας πράκτορας ενεργεί προσωπικά, χρησιμοποιεί το BDI-CPR μηχανισμό για να επιλέξει τις σχετικές μεθόδους των ενεργειών που επιθυμεί να πραγματοποιήσει, ελέγχει την εφαρμοσιμότητα των μεθόδων εντοπίζοντας και επιλύοντας πιθανές σύγκρουσης της μεθόδου με μεθόδους άλλων ενεργειών (ακολουθώντας πάντα τις κατευθύνσεις συμπεριφοράς που προκύπτουν κατά τη διάρκεια σχεδιασμού και εκτέλεσης του πλάνου της ενέργειας) και αναπτύσσει τη μέθοδο σχηματίζοντας νέες δεσμεύσεις.



Εικόνα 35: Η νοητικές καταστάσεις των πρακτόρων  $G_0$  και  $G_1$  κατά τη διάρκεια της συνεργασίας.

Στη περίπτωση που ο πράκτορας εντοπίσει την ανάγκη να συνεργαστεί με μία ομάδα πρακτόρων τότε σχηματίζει μία δέσμευση για κοινή δράση (group activity commitment). Αυτή η δέσμευση υποχρεώνει το πράκτορα να στοχαστεί μαζί με τους πιθανούς συνεργάτες του τον τρόπο με τον οποίο θα αναπτύξει τη σχετική και εφαρμόσιμη μέθοδο που έχει επιλέξει. Ο κοινωνικός αυτός στοχασμός πραγματοποιείται εκτελώντας τις τρεις νοητικές ενέργειες <group\_formation>, <recipe\_negotiation> και <task\_allocation>. Τελικώς, ο πράκτορας δεσμεύεται να πραγματοποιήσει τη μέθοδο που έχει επιλέξει μόνο όταν επιτύχει να εκτελέσει και τις τρεις παραπάνω νοητικές υπό-ενέργειες της <elaborate\_recipe>.

Με τη νοητική ενέργεια <group\_formation> ο πράκτορας επικοινωνεί με τους άλλους πράκτορες προκειμένου να ζητήσει βοήθεια. Στη περίπτωση που οι πράκτορες αποδεχθούν το αίτημα του για βοήθεια τότε ταυτόχρονα τα μέλη της ομάδας σχηματίζονται.

ζουν μία δέσμευση ότι η ομάδα θα πετύχει τον κοινό στόχο και από μία επιθυμία ο κάθε πράκτορας να πραγματοποιήσει την ενέργεια αυτή. Αποτέλεσμα της συμπεριφοράς αυτής είναι ο κάθε πράκτορας της ομάδας να βρει μία σχετική και εφαρμόσιμη μέθοδο για τη κοινή ενέργεια (παίρνοντας υπ' όψη τη μέθοδο που διάλεξε ο πρώτος πράκτορας).

Έχοντας καθοριστεί τα μέλη της ομάδας, ο κάθε πράκτορας διαπραγματεύεται τη μέθοδο που έχει επιλέξει προκειμένου να συμφωνήσουν σε μία μέθοδο που από κοινού θ' ακολουθήσουν. Η διαπραγμάτευση επιτυγχάνεται εκτελώντας (ο κάθε πράκτορας της ομάδας παράλληλα από τον άλλο) τη νοητική ενέργεια `<recipe_negotiation>`. Αν οι πράκτορες συμφωνήσουν για το κοινό στόχο της μεθόδου, τις υπό-ενέργειες, τα αποτελέσματα και τις συνθήκες εφαρμογής της μεθόδου τότε όλοι οι πράκτορες πιστεύουν ότι μοιράζονται μία κοινή μέθοδο για την ενέργεια που έχουν δεσμευθεί ότι θα τη πραγματοποιήσουν από κοινού.

Για να ολοκληρώσουν την ανάπτυξη της κοινής μεθόδου, ο κάθε πράκτορας διαπραγματεύεται τον τρόπο με τον οποίο θα καταμεριστούν οι υπό-ενέργειες της μεθόδου μεταξύ τους. Αυτό επιτυγχάνεται εκτελώντας ο κάθε πράκτορας τη νοητική ενέργεια `<task_allocation>`. Στη περίπτωση που όλοι οι πράκτορες συμφωνήσουν σε μία κοινή κατανομή τότε η νοητική ενέργεια `<elaborate_recipe>` ολοκληρώνεται επιτυχώς με αποτέλεσμα ο κάθε πράκτορας της ομάδας να σχηματίζει μία δέσμευση να πραγματοποιήσει την ενέργεια (`int_to(Agent, Action)`).

Στη περίπτωση που ένας πράκτορας αποτύχει σε κάποιο από τα παραπάνω στάδια της συνεργασίας τότε δημιουργείται μία σύγκρουση. Ο πράκτορας αυτός αναγκάζεται (από τη δέσμευση που έχει να πραγματοποιήσει την ενέργεια ομαδικά) να επικοινωνήσει τη σύγκρουση αυτή στα άλλα μέλη της ομάδας με αποτέλεσμα να επαναληφθούν τα βήματα της συνεργασίας και μόνο αυτά που χρειάζονται<sup>33</sup>.

---

<sup>33</sup> Όπως ήδη έχουμε τονίσει, αν για παράδειγμα κανένας πράκτορας δεν εγκαταλείψει την ομάδα τότε οι πράκτορες δεν διαπραγματεύονται ξανά για το ποιοι πράκτορες θα αποτελούν την ομάδα. Δηλαδή, η νοητική ενέργεια `<group_formation>` δεν επαναεκτελείται. Το ίδιο ισχύει και για τις `<recipe_negotiation>` και `<task_allocation>`.

## Κεφάλαιο 6ο:

# Ανάπτυξη Πρακτόρων για Περιβάλλοντα Επιφανειών Διεπαφής

Στο κεφάλαιο αυτό παρουσιάζουμε τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί το πλαίσιο ανάπτυξης πρακτόρων ICAGENT στην υλοποίηση συνεργατικών πρακτόρων επιφανειών διεπαφής. Ως μελέτη περίπτωσης παρουσιάζουμε ένα πράκτορα διεπαφής για την εφαρμογή «Δημιουργός Μοντέλων» που παρουσιάσαμε στην υποενότητα 2.5.2. Στις ενότητες που ακολουθούν θα περιγράψουμε το γενικό πλαίσιο της δραστηριότητας της συνεργασίας του πράκτορα διεπαφής με το χρήστη, τον τρόπο με τον οποίο ο πράκτορας αναγνωρίζει τους πρωταρχικούς στόχους του χρήστη και ερμηνεύει τις ενέργειες του υπό το πλαίσιο ενεργειών των στόχων αυτών, την αναγνώριση της ανάγκης για συνεργασία και τον τρόπο με τον οποίο ο πράκτορας στοχάζεται με τον χρήστη (social deliberate) για ν' αναπτύξουν και να πραγματοποιήσουν τα πλάνα τους από κοινού.

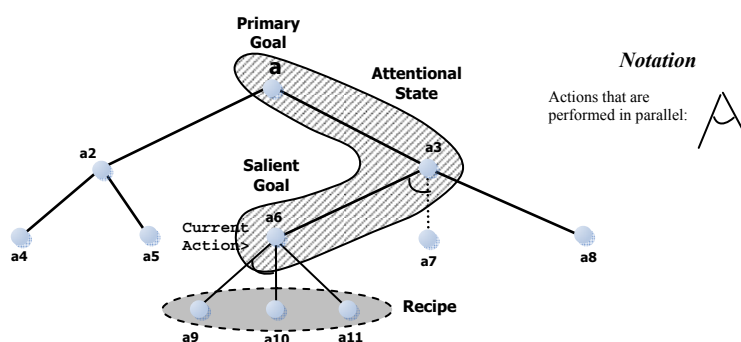
### 6.1. Το γενικό πλαίσιο της δραστηριότητας της συνεργασίας του πράκτορα διεπαφής με τον χρήστη

Όπως ήδη έχουμε αναφέρει στα Κεφάλαια 3 και 4 η συνεργασία στο πλαίσιο ανάπτυξης πρακτόρων ICAGENT βασίζεται στο μοντέλο συνεργασίας των Grosz και Kraus (1996) και στα τέσσερα στάδια της συνεργασίας μεταξύ πρακτόρων του Wooldridge (1999) (βλ. Εικόνα 32).

Πριν παρουσιάσουμε τη δραστηριότητα της συνεργασίας μεταξύ του πράκτορα και του χρήστη στο περιβάλλον του Δημιουργού Μοντέλων θα δώσουμε μερικούς συμπληρωματικούς ορισμούς οι οποίοι είναι αναγκαίοι για τη περιγραφή της συνεργασίας, καθώς επίσης θα αναφερθούμε στον τρόπο επικοινωνίας του πράκτορα με τον χρήστη η

οποία όπως τονίσαμε, είναι αναγκαία για μία αποτελεσματική και έμπιστη συνεργασία (βλ. προηγούμενα ενότητες 2.3 και 2.4).

Κάθε μία ενέργεια του χρήστη ή του πράκτορα στην επιφάνεια διεπαφής μπορεί να είναι είτε σύνθετη είτε βασική και υλοποιείται από τουλάχιστον μία μέθοδο (recipe) έτσι όπως την ορίσαμε στην υποενότητα 4.1.2. Η ανάπτυξη μίας σύνθετης ενέργειας σε άλλες απλούστερες έχει ως αποτέλεσμα τη δημιουργία ενός πλαισίου ενεργειών (βλ. υποενότητα 3.2.1) το οποίο μπορεί να είναι μερικώς γνωστό τόσο στο χρήστη όσο και στον πράκτορα.



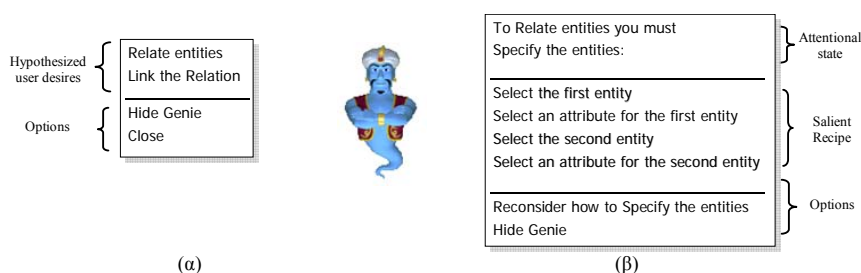
Εικόνα 36: Το πλαίσιο ενεργειών μίας ενέργειας *a*.

Για την ανάπτυξη του πράκτορα διεπαφής ξεχωρίζουμε τρία είδη πλάνων: τα προσωπικά, τα κοινά και τα πλάνα πιθανών πρωταρχικών στόχων CPGs (candidate primary goals) [Kourakos and Vouros, 2006].

- Ένα προσωπικό πλάνο πραγματοποιείται αποκλειστικά από τον πράκτορα ή από τον χρήστη.
- Ένα κοινό πλάνο αντιστοιχεί σε ένα κοινό στόχο που ο πράκτορας και ο χρήστης έχουν συμφωνήσει να πραγματοποιήσουν μαζί. Το πλάνο αυτό αναπτύσσεται από κοινού και οι ενέργειες αυτού μπορούν να πραγματοποιηθούν μόνο από τον πράκτορα ή από τον χρήστη ή από κοινού (μόνο στη περίπτωση που αυτές είναι σύνθετες).
- Ένας πιθανός πρωταρχικός στόχος CPG αναπαριστά ένα πιθανό βασικό στόχο του χρήστη που ο πράκτορας πιστεύει ότι ο χρήστης έχει. Σε αντίθεση με τους προσωπικούς ή τους κοινούς στόχους, ο πράκτορας δεν σχεδιάζει πλάνα για αυτούς τους

στόχους αλλά τους χρησιμοποιεί για να ερμηνεύει τις ενέργειες του χρήστη στην επιφάνεια διεπαφής και ν' αναγνωρίζει το πότε πιθανόν να χρειάζεται βοήθεια. Ο πράκτορας δεν δεσμεύεται να πραγματοποιήσει ένα πιθανό πρωταρχικό στόχο μέχρι να συμφωνήσουν με το χρήστη για την από κοινού πραγματοποίηση του καθώς και για τη κοινή μέθοδο που θα χρησιμοποιήσουν.

Για να υλοποιήσουμε το πράκτορα διεπαφής για το Δημιουργό Μοντέλων χρησιμοποιούμε ως «σώμα» τον MSAgent της Microsoft®. Το λογισμικό αυτό υλοποιεί τη πολύ-μεσική διαδραστική επικοινωνία του πράκτορα διεπαφής με τον χρήστη. Χρησιμοποιώντας τον MSAgent ο χρήστης μπορεί να συμμετέχει σε διαλόγους με τον πράκτορα μέσω πλαισίων διαλόγου ή μέσω μηχανικής αναγνώρισης φωνής (speech recognition engine). Από την άλλη πλευρά, ο πράκτορας διεπαφής είναι ικανός να επικοινωνεί με το χρήστη χρησιμοποιώντας μπαλόνια κειμένου, συνθετικής φωνής και χειρονομιών (βλ Εικόνα 11).



Εικόνα 37: Τύποι πλαισίων διαλόγου για (α) παρουσίαση υποθέσεων που αφορούν τους πιθανούς πρωταρχικούς στόχους του χρήστη και (β) για παρουσίαση των τρεχόντων στόχων (attentional state) και της τρέχουσας μεθόδου υλοποίησης (recipe of focal action).

Ο τύπος του μέσου επικοινωνίας (μπαλόνια κειμένου, συνθετική φωνή και χειρονομίες) εξαρτάται από τους εκπαιδευτικούς στόχους που ο πράκτορας έχει επιλέξει και από τους στόχους του χρήστη. Για τις ανάγκες της συνεργασίας μεταξύ του πράκτορα και του χρήστη κάνουμε χρήση δύο τύπων πλαισίων διαλόγου (βλ. Εικόνα 37):

- (α) Ο πρώτος τύπος χρησιμοποιείται πριν τη συνεργασία για ν' παρουσιάσει ο πράκτορας τις υποθέσεις που έχει σχετικά με τις επιθυμίες του χρήστη (CPGs) καθώς και ως εργαλείο με το οποίο ο χρήστης μπορεί να δηλώσει στο πράκτορα τι πραγματικά θέλει να κάνει. Για παράδειγμα, στην Εικόνα 37(α) απεικονίζουμε ένα πλαίσιο διαλόγου δύο πεδίων όπου στο πρώτο πεδίο ο πράκτορας παρουσιάζει τις υποθέσεις του για τους πιθανούς πρωταρχικούς στόχους του χρήστη, ενώ στο δεύ-

τερο πεδίο εμφανίζει δύο επιλογές με τις οποίες ο χρήστης μπορεί να κρύψει το MSAgent ή να τον απενεργοποιήσει. Ο χρήστης κάνοντας κλικ στις επιλογές του πρώτου πεδίου μπορεί να ενημερώσει τον πράκτορα τι ακριβώς θέλει να κάνει.

- (β) Ο δεύτερος τύπος χρησιμοποιείται κατά τη διάρκεια της συνεργασίας, για να παρουσιάσει ο πράκτορας τη τρέχουσα κατάσταση του κοινού πλάνου (διαλόγου). Πιο συγκεκριμένα, για να παρουσιάσει τους τρέχοντες στόχους (attentional state) της κοινής ενέργειας, καθώς και τη μέθοδο που υλοποιεί την ενέργεια αυτή. Για παράδειγμα, στην Εικόνα 37 απεικονίζεται ένα πλαίσιο διαλόγου με τρία πεδία: στο πρώτο παρουσιάζεται ο κοινός στόχος του πράκτορα και του χρήστη που είναι να καθορίσουν τις οντότητες (Specify the entities) έτσι ώστε να μπορούν να τις συσχετίσουν (To relate the entities). Στο δεύτερο πλαίσιο, παρουσιάζεται ο τρόπος (δηλαδή η μέθοδος) με τον οποίο μπορούν να καθοριστούν οι οντότητες. Τέλος, στο τρίτο πεδίο παρουσιάζεται μία λίστα από επιλογές σύμφωνα με τις οποίες ο χρήστης μπορεί να προτείνει στο πράκτορα είτε να αναθεωρήσει τη μέθοδο που έχει επιλέξει είτε να τελειώσει τη συνεργασία κρύβοντας τον πράκτορα.

Κάνοντας χρήση τέτοιων πλαισίων διαλόγου, ο χρήστης έχει πάντοτε τον έλεγχο του διαλόγου και άρα της συνεργασίας, διότι μπορεί να επιβεβαιώσει τις προθέσεις του, ν' αναθεωρήσει τους κοινούς στόχους καθώς και τους τρόπους που αυτοί πραγματοποιούνται από κοινού.

## 6.2. Αναγνώριση της ανάγκης για βοήθεια

Σε αυτό το στάδιο της συνεργασίας ο πράκτορας ερμηνεύει τις ενέργειες του χρήστη είτε υπό το πλαίσιο μίας κοινής ενέργειας που έχουν συμφωνήσει να πραγματοποιήσουν μαζί είτε υπό το πλαίσιο των ενεργειών που ορίζουν οι πιθανοί πρωταρχικοί στόχοι που έχει εντοπίσει ο πράκτορας προσπαθώντας ν' αναγνωρίσει πιθανές περιπτώσεις συνεργασίας.

Ο πράκτορας αρχίζει τη συνεργασία με τον χρήστη για τη πραγματοποίηση μίας ενέργειας σε δύο περιπτώσεις:

Στη πρώτη περίπτωση αναγνωρίζει ότι η συνεργασία μπορεί να είναι αναγκαία είτε για εκπαιδευτικούς λόγους είτε γιατί γνωρίζει εκ των προτέρων ότι ο χρήστης δεν μπο-



ρεί να πραγματοποιήσει την ενέργεια από μόνος του. Και οι δύο αυτές περιπτώσεις προκύπτουν όταν οι συνθήκες συνεργασίας (*Collaboration\_Mental\_Condition*) της μεθόδου που έχει επιλέξει για την ενέργεια ισχύουν<sup>34</sup>. Για παράδειγμα, στο περιβάλλον του Δημιουργού Μοντέλων όταν ο πράκτορας αναλαμβάνει να καθορίσει (μετά από απαίτηση του χρήστη) μία οντότητα ή μία σχέση (για τη δημιουργία μίας σχέσης) από τη βιβλιοθήκη οντοτήτων ή σχέσεων αντίστοιχα τότε θεωρεί ότι η ενέργεια αυτή πρέπει να γίνει σε συνεργασία με τον χρήστη διότι δεν γνωρίζει ποια ακριβώς οντότητα ή σχέση θέλει ο χρήστης. Σε αυτή τη περίπτωση ο πράκτορας ζητά από τον χρήστη να επιλέξει πρώτα τη βιβλιοθήκη οντοτήτων (ή σχέσεων) που χρειάζεται (βλ. Εικόνα 12 υποκεφαλαίου 2.5.2), μετά να επιλέξει την οντότητα (ή τη σχέση) και τέλος ο πράκτορας αναλαμβάνει να μεταφέρει την οντότητα (ή τη σχέση) από τη βιβλιοθήκη στο χώρο δημιουργίας μοντέλων.

Η δεύτερη περίπτωση προκύπτει όταν ο πράκτορας αναγνωρίζει ότι η σχετική μέθοδος που έχει επιλέξει για μία (προσωπική ή κοινή) ενέργεια δεν είναι εφαρμόσιμη. Εδώ διακρίνουμε δύο περιπτώσεις ανάλογα με το αν η μέθοδος είναι ενός πιθανού πρωταρχικού στόχου ή μίας κοινής ενέργειας:

Αν η μέθοδος είναι ενός πιθανού πρωταρχικού στόχου CPG τότε ο πράκτορας δημιουργεί μία δέσμευση να πραγματοποιήσει το CPG σε συνεργασία με το χρήστη (<int\_that\_do>) και τον παρακινεί ν' αποδεχθεί τη συνεργασία αυτή. Για παράδειγμα αν ο χρήστης έχει εισάγει δύο οντότητες και προσπαθεί να δημιουργήσει μία σχέση χωρίς πρώτα να έχει προσδιορίσει τις ιδιότητές τους τότε ο πράκτορας αντιλαμβάνεται ότι οι συνθήκες εφαρμογής της ενέργειας «σύνδεσης της σχέσεις με τις οντότητες» δεν ισχύουν και προτρέπει τον χρήστη να συνεργαστεί, χωρίς όμως να τον παρεμποδίζει από τις τρέχουσες ενέργειες του (βλ. Εικόνα 10 υποκεφαλαίου 2.5.2).

Αν η σύγκρουση είναι μεταξύ μίας κοινής ενέργειας και μίας ενέργειας του χρήστη και γνωρίζει μία ενέργεια που επιλύει τη σύγκρουση αυτή τότε ο πράκτορας δημιουργεί μία δέσμευση για συνεργασία για τον αμέσως προηγούμενο στόχο (γονική ενέργεια)

---

<sup>34</sup> Λόγω του ότι δεν έχουμε συμπεριλάβει εκπαιδευτικές στρατηγικές στην ανάπτυξη του πράκτορα διεπαφής για το Δημιουργό Μοντέλων, τη πρώτη περίπτωση δεν την αναλύουμε περαιτέρω.

που περιέχει τη ενέργεια αυτή και τη παρουσιάζει κάνοντας χρήστη ενός πλαισίου διαλόγου.

Στη περίπτωση που ο πράκτορας έχει δεσμευτεί να βοηθήσει τον χρήστη για μία ενέργεια, τότε για να αρχίσει η συνεργασία θα πρέπει ο χρήστης να αποδεχθεί τη παρακίνηση του πράκτορα για συνεργασία είτε κάνοντας κλικ στο πράκτορα (αν η ενέργεια αυτή είναι ένα CPG) είτε επιλέγοντας μία ενέργεια από το πλαίσιο διαλόγου.

### 6.3. Σχηματισμός της ομάδας

Οποτεδήποτε ο πράκτορας πιστεύει ότι πρέπει να συνεργαστεί με το χρήστη για τη πραγματοποίηση από κοινού μίας ενέργειας, επικοινωνεί την επιθυμία του αυτή στο χρήστη καθώς και το λόγο για τον οποίο πρέπει να συνεργαστεί. Ο στόχος είναι ο σχηματισμός της ομάδας. Για παράδειγμα, όταν ο πράκτορας εντοπίζει μία σύγκρουση τότε παρουσιάζει τη σύγκρουση αυτή στον χρήστη και τον παρακινεί να συνεργαστεί προκειμένου να πραγματοποιήσουν από κοινού την ενέργεια που επιλύει τη σύγκρουση (βλ. Εικόνα 11 υποκεφαλαίου 2.5.2).

Στην υλοποίηση του πράκτορα διεπαφής η ομάδα είναι πάντα συγκεκριμένη αποτελούμενη από το χρήστη και τον πράκτορα. Η ομάδα σχηματίζεται όταν ο χρήστης αποδέχεται τη συνεργασία. Αποτέλεσμα της νοητικής ενέργειας του σχηματισμού της ομάδας είναι η δημιουργία της αμοιβαίας πεποίθησης ότι και οι δύο έχουν δεσμευθεί να πραγματοποιήσουν την ενέργεια συνεργατικά (`mb(Group, int_that(Agent, do(Group, Action, Null)))`).

### 6.4. Διαπραγμάτευση μεθόδων

Στο στάδιο αυτό, ο πράκτορας συμφωνεί με τον χρήστη για τη μέθοδο που θα ακολουθήσουν για να πραγματοποιήσουν τη κοινή τους ενέργεια. Με το τέλος της νοητικής αυτής ενέργειας ο πράκτορας (όσο και ο χρήστης) γνωρίζει μία σχετική και εφαρμόσιμη μέθοδο για τη κοινή ενέργεια και δεσμεύεται να τη πραγματοποιήσει συνεργατικά ακολουθώντας τη μέθοδο αυτή (δηλαδή όπως τονίσαμε στην ενότητα 5.3 ισχύει η αμοιβαία πεποίθηση `mb(Group, int_that(Agent, do(Group, Action, Recipe)))`).

Ο πράκτορας για να ενημερώσει το χρήστη σχετικά με τη μέθοδο που έχει επιλέξει ως σχετική και εφαρμόσιμη καθώς και για το λόγο για τον οποίο την έχει επιλέξει χρησιμοποιεί ένα πλαίσιο επιλογών (context menu) τριών πεδίων (βλ. Εικόνα 50 υποκεφαλαίου 6.10). Στο πρώτο πεδίο παρουσιάζει τους λόγους για τους οποίους έχει επιλέξει τη συγκεκριμένη μέθοδο (δηλαδή παρουσιάζει τους τρέχοντες στόχους (attentional state)), στο δεύτερο πεδίο παρουσιάζει τα βήματα (υπό-ενέργειες) που υλοποιούν τη συγκεκριμένη μέθοδο και τέλος στο τρίτο πεδίο παρουσιάζει μία σειρά επιλογών με τις οποίες ο χρήστης μπορεί να μη δεχθεί τη συγκεκριμένη μέθοδο, να κρύψει τον πράκτορα κτλ.

Παρουσιάζοντας τη μέθοδο (δηλαδή τις υπό-ενέργειες που τη συνθέτουν) στο χρήστη ο πράκτορας θεωρεί ότι ο χρήστης συμφωνεί να τη πραγματοποιήσουν από κοινού. Σε αντίθετη περίπτωση, ο χρήστης μπορεί να την απορρίψει επιλέγοντας την αντίστοιχη επιλογή από το τρίτο πεδίο του πλαισίου διαλόγου (δηλαδή επιλέγοντας την επιλογή "Reconsider how to ...").

### **6.5. Καταμερισμός εργασιών μεταξύ πράκτορα και χρήστη**

Μετά τη παρουσίαση της μεθόδου στο χρήστη και την αποδοχή αυτής ως κοινής η επόμενη νοητική ενέργεια που πραγματοποιεί ο πράκτορας είναι αυτή του καταμερισμού των υπό-ενεργειών της μεθόδου μεταξύ τους. Αν κατά την υλοποίηση του πράκτορα έχουν ληφθεί υπόψη εκπαιδευτικές στρατηγικές τότε μέσω των συνθηκών συνεργασίας (Collaboration\_Mental\_Condition) της μεθόδου ο πράκτορας μπορεί ν' αποφασίσει για το ποιες ενέργειες θα πραγματοποιήσει ο ίδιος και ποιες θα παρακινήσει το χρήστη να τις κάνει. Λόγω του ότι στη συγκεκριμένη ανάπτυξη του πράκτορα διεπαφής για το Δημιουργό Μοντέλων δεν έχουν ληφθεί υπόψη εκπαιδευτικές στρατηγικές η επιλογή του καταμερισμού των εργασιών γίνεται αποκλειστικά από τον χρήστη.

Οποτεδήποτε ο πράκτορας παρουσιάζει στο χρήστη μία μέθοδο, ο τελευταίος μπορεί να ζητήσει από το πράκτορα να πραγματοποιήσει μία υπό-ενέργεια επιλέγοντας αυτή από το δεύτερο πεδίο του πλαισίου διαλόγου. Αν ο χρήστης αποφασίσει να πραγματοποιήσει μία ή περισσότερες ενέργειες τις μεθόδου προσωπικά τότε ο πράκτορας ερμηνεύει τις ενέργειες του χρήστη κάτω από το πλαίσιο ενεργειών των κοινών στόχων ενημερώνοντας έτσι το κοινό πλάνο.

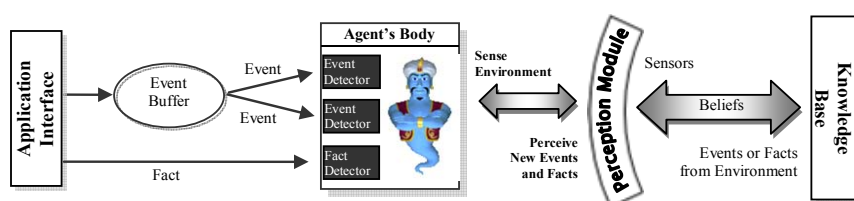
Συνοψίζοντας τις παραπάνω νοητικές ενέργειες που αφορούν τη συνεργασία του πράκτορα με το χρήστη, ο πράκτορας της διεπαφής παρατηρεί τις ενέργειες του χρήστη και ερμηνεύει αυτές προκειμένου να σχηματίσει υποθέσεις σχετικά με τις επιθυμίες του χρήστη (CPGs) και για να ενημερώσει το τρέχον κοινό πλαίσιο ενεργειών. Οι ερμηνείες των ενεργειών αυτών οδηγούν το πράκτορα σε αναγνώριση των στόχων του χρήστη και κατ' επέκταση (μέσω της διεργασίας συλλογισμού «ανάπτυξης πλάνων» και του «εντοπισμού και επίλυσης συγκρούσεων») σε αναγνώριση των λανθασμένων ενεργειών του χρήστη στην επιφάνεια διεπαφής (βλ. ενότητες 6.8 και 6.9).

Στις επόμενες ενότητες παρουσιάζουμε τον τρόπο με τον οποίο ο πράκτορας αντιλαμβάνεται τις ενέργειες του χρήστη, τις ερμηνεύει κάτω από το τρέχον πλαίσιο ενεργειών που έχει σχηματίσει, καθώς επίσης και τον τρόπο με τον οποίο αναπτύσσει τα πλάνα του σε συνεργασία με το χρήστη.

## 6.6. Παρατηρώντας τις ενέργειες του χρήστη

Το πλαίσιο εργασίας του ICAGENT παρέχει τους κατάλληλους μηχανισμούς με τους οποίους ο πράκτορας μπορεί να αντιλαμβάνεται τη (μερική ή ολική) κατάσταση της επιφάνειας διεπαφής και τις ενέργειες που ο χρήστης πραγματοποιεί πάνω σε αυτή.

Για να μπορεί να επικοινωνεί η επιφάνεια διεπαφής με το σώμα του πράκτορα, θεωρούμε ότι κάθε ενέργεια του χρήστη καταγράφεται σε ένα αρχείο (event buffer). Το σώμα του πράκτορα (και επομένως η μονάδα αντίληψης) έχει πρόσβαση στο αρχείο αυτό μέσω διαδικασιών οι οποίες υλοποιούν τους αισθητήρες εντοπισμού ενεργειών (event detectors) (βλ. Εικόνα 38).



Εικόνα 38: Το σώμα και ο μηχανισμός αντίληψης του πράκτορα διεπαφής.

Οι προδιαγραφές των αισθητήρων <sensor> (που παρουσιάσαμε στην υποενότητα 4.2.1) καθιστούν τον πράκτορα ικανό να χειρίζεται τους αισθητήρες και να αντιλαμβάνεται τις ενέργειες που ο χρήστης πραγματοποιεί στην επιφάνεια διεπαφής. Επίσης η

γνώση που αυτές οι προδιαγραφές εμπεριέχουν σχετικά με το πότε και ποια γεγονότα και ενέργειες ο πράκτορας πρέπει να παρατηρεί και σε συνδυασμό με τους κανόνες ενημέρωσης της βάσης γνώσης `<update_rule>` λειτουργούν ως ένα είδος φίλτρου για το τι ο πράκτορας πρέπει να γνωρίζει σχετικά με τη κατάσταση της επιφάνειας διεπαφής και τις προθέσεις του χρήστη. Έτσι ενέργειες του χρήστη οι οποίες δεν σχετίζονται με κάποια ενέργεια της εφαρμογής (όπως για παράδειγμα η μετακίνηση μίας οντότητας από το ένα μέρος του χώρου δημιουργίας μοντέλων σε κάποιο άλλο) ή δεν συνεισφέρουν σε κάποιο κοινό στόχο<sup>35</sup> δεν λαμβάνονται υπόψη.

Στον Πίνακα 13 του Παραρτήματος Ε παρουσιάζουμε δύο παραδείγματα προδιαγραφών αισθητήρων. Με τη πρώτη προδιαγραφή ο πράκτορας αναγνωρίζει ενέργειες του χρήστη του τύπου `current_user_action(UserAction)`. Στη περίπτωση που η ενέργεια του χρήστη (`UserAction`) συνεισφέρει σε ένα πιθανό πρωταρχικό στόχο CPG ή σε ένα κοινό στόχο τότε ο πράκτορας ενημερώνει τη βάση γνώσης (δηλαδή το πλαίσιο ενεργειών) ότι πραγματοποιήθηκε η αντίστοιχη ενέργεια. Με τη δεύτερη προδιαγραφή ο πράκτορας μετρά τις ενέργειες που πραγματοποιεί ο χρήστης οι οποίες δεν συνεισφέρουν στο κοινό στόχο. Αν ο αριθμός αυτών των ενεργειών γίνει μεγαλύτερος ή ίσος ενός προκαθορισμένου αριθμού (`threshold`) (στο συγκεκριμένο παράδειγμα ίσος με 3) τότε ο πράκτορας απορρίπτει το κοινό στόχο και προσπαθεί πάλι ν' αναγνωρίσει το πιθανό πρωταρχικό στόχο του χρήστη.

## 6.7. Κανόνες αναγνώρισης καταστάσεων

Ο πράκτορας διεπαφής του Δημιουργού Μοντέλων χρησιμοποιεί δύο κανόνες αναγνώρισης καταστάσεων `<situation>` που περιγράψαμε στην υπό-παράγραφο 4.2.2:

Ο πρώτος κανόνας αφορά την αναγνώριση της κατάστασης όπου ο πράκτορας έχει ακριβώς μία υπόθεση για το πρωταρχικό στόχο του χρήστη. Σε αυτή τη περίπτωση ο αντίστοιχος κανόνας δημιουργεί την επιθυμία να πραγματοποιήσει ο πράκτορας το

---

<sup>35</sup> Όπως θα δούμε αργότερα μετά από κάποιο αριθμό ενεργειών που δεν συνεισφέρουν στο κοινό στόχο ο πράκτορας σταματάει τη συνεργασία με το χρήστη και προσπαθεί πάλι ν' αναγνωρίσει το πρωταρχικό στόχο του χρήστη (CPG).

πιθανό πρωταρχικό στόχο. Η επιθυμία αυτή οδηγεί το πράκτορα να επιλέξει μία σχετική και εφαρμόσιμη μέθοδο για το στόχο αυτό. Όμως, όπως θα δούμε στην επόμενη ενότητα, ο πράκτορας δεν αναπτύσσει περαιτέρω τη μέθοδο αυτή αλλά τη χρησιμοποιεί για να δημιουργήσει ένα πλαίσιο ενεργειών κάτω από το οποίο θα ερμηνεύσει τις επόμενες ενέργειες του χρήστη.

Ο δεύτερος κανόνας αφορά την κατάσταση όπου ο πράκτορας διεπαφής έχει βρει παραπάνω από έναν πιθανούς πρωταρχικούς στόχους του χρήστη. Στη περίπτωση αυτή, ο αντίστοιχος κανόνας δημιουργεί την επιθυμία να παρουσιάσει ο πράκτορας στο χρήστη τους στόχους αυτούς (μέσω μίας επικοινωνιακής ενέργειας) χρησιμοποιώντας ένα πλαίσιο διαλόγου (dialogue box), ζητώντας από το χρήστη να δηλώσει ποιον ακριβώς στόχο προσπαθεί να πετύχει.

## 6.8. Ερμηνεία ενεργειών του χρήστη

Ο πράκτορας παρατηρεί τις ενέργειες του χρήστη στην επιφάνεια διεπαφής για να (α) σχηματίσει υποθέσεις για τους βασικούς στόχους του χρήστη (CPG) και (β) για να ερμηνεύσει τις ενέργειες του χρήστη κάτω από ένα κοινό πλαίσιο ενεργειών (κοινό στόχο). Ο σχηματισμός των υποθέσεων αυτών και η ερμηνεία των ενεργειών του χρήστη υλοποιείται μέσω μίας ενσωματωμένης διαδικασίας (build-in task) που ο πράκτορας ενεργοποιεί μέσω της διεργασίας συλλογισμού της αντίληψης (Perception module). Οι υποθέσεις και οι ερμηνείες αυτές, μέσω της διεργασίας συλλογισμού του εντοπισμού και επίλυσης συγκρούσεων οδηγούν το πράκτορα στο να ελέγχει την εφαρμοσιμότητα (applicability) των μεθόδων που ο πράκτορας επιλέγει για να ερμηνεύσει τις ενέργειες του χρήστη και άρα στο ν' αναγνωρίζει την ανάγκη του χρήστη για βοήθεια.

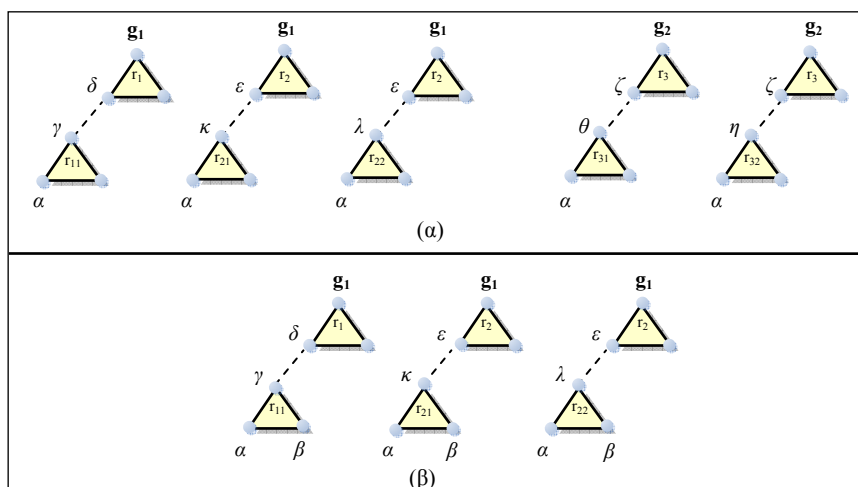
Στις επόμενες δύο υποενότητες περιγράφουμε τη διεργασία αναγνώρισης των πιθανών πρωταρχικών στόχων του χρήστη CPGs καθώς και τη διεργασία με την οποία ο πράκτορας ερμηνεύει τις ενέργειες του χρήστη κάτω από το πλαίσιο ενεργειών ενός (πιθανού ή κοινού) πρωταρχικού στόχου.

### 6.8.1. Σχηματίζοντας υποθέσεις για τους πρωταρχικούς στόχους του χρήστη

Έστω ότι ο πράκτορας και ο χρήστης δεν έχουν ακόμα αποφασίσει να συνεργαστούν για να πραγματοποιήσουν από κοινού μία ενέργεια και ότι ο χρήστης πραγματοποιεί μία ενέργεια στην επιφάνεια διεπαφής. Ο πράκτορας αφού αντιληφθεί την ενέργεια αυτή προσπαθεί να εντοπίσει όλους τους δυνατούς λόγους (attentional states) για τους οποίους ο πράκτορας μπορεί να κάνει την ενέργεια αυτή. Οι πιθανοί στόχοι που περιλαμβάνουν την ενέργεια του χρήστη σχηματίζονται βρίσκοντας όλες τις σχετικές μεθόδους που περιλαμβάνουν την ενέργεια αυτή ως υπό-ενέργεια. Για παράδειγμα, όπως φαίνεται στην Εικόνα 39, η πραγματοποίηση από το χρήστη της ενέργειας  $a$  έχει ως αποτέλεσμα ο πράκτορας να εντοπίσει δύο πιθανούς πρωταρχικούς στόχους του χρήστη: τους  $g_1$  και  $g_2$ . Για το  $g_1$  έχει βρει τρεις υποθέσεις που αφορούν το attentional state του χρήστη, ενώ για το  $g_2$  έχει βρει δύο.

Για να σχηματίσει αυτές τις υποθέσεις, ο πράκτορας βρίσκει μεθόδους οι οποίες περιέχουν την ενέργεια  $a$  ως υπό-ενέργεια (στη λίστα ενεργειών `Action_List` της μεθόδου). Για κάθε τέτοια μέθοδο, ελέγχει τη σχετικότητα της (δηλαδή το νοητική συνθήκη `Relevance_Mental_Condition` της μεθόδου) στο τρέχον πλαίσιο ενεργειών και αν είναι πράγματι σχετική τότε ο πράκτορας συνεχίζει βρίσκοντας μία σχετική μέθοδο για την ενέργεια αυτή (γονική της  $a$ ) και συνεχίζοντας έτσι ο πράκτορας φθάνει μέχρι τη ρίζα του δένδρου η οποία αναπαριστά ένα πιθανό πρωταρχικό στόχο του χρήστη.

Αν ο πράκτορας αναγνωρίσει περισσότερους από έναν πρωταρχικούς στόχους τότε οδηγούμενος από τον αντίστοιχο κανόνα αναγνώρισης κατάστασης (`<situation>`) που περιγράψαμε στην ενότητα 6.7, αιτείται από το χρήστη να αποσαφηνίσει τι θέλει να κάνει παρουσιάζοντάς του σε ένα πλαίσιο διαλόγου όλους τους πιθανούς πρωταρχικούς στόχους που έχει αναγνωρίσει. Αν ο χρήστης επιλέξει έναν από τους εμφανιζόμενους στόχους τότε ο πράκτορας σχηματίζει την επιθυμία να πραγματοποιήσει την αντίστοιχη ενέργεια και τη πρόθεση να κάνει αυτό σε συνεργασία με τον χρήστη (`<int_that_do>`). Ελέγχοντας την εφαρμοσιμότητα της μεθόδου (`Applicability_Mental_Condition`) ο πράκτορας σχηματίζει ένα στόχο (`<goal_to>`) τον οποίο δε μοιράζεται ακόμα με τον χρήστη μέχρι να τον πληροφορήσει για τη μέθοδο που έχει επιλέξει (δηλαδή, μέχρι να εκτελέσει την `<elaborate_recipe>`).



Εικόνα 39: (α) Για κάθε ενέργεια  $a$  του χρήστη ο πράκτορας βρίσκει σχετικές μεθόδους μέχρι να εντοπίσει ένα πιθανό πρωταρχικό στόχο  $g$ . (β) Όσο ο πράκτορας συνεχίζει να ενεργεί οι υποθέσεις αυτές ελαττώνονται.

Δεδομένου μίας ενέργειας του χρήστη, ο χώρος αναζήτησης για να εξηγήσει την ενέργεια αυτή φράζεται από τον αριθμό  $(R \times S)^L$ , όπου  $R$  είναι ο μέγιστος αριθμός σχετικών μεθόδων που μπορούν να θεωρηθούν,  $S$  είναι ο μέγιστος αριθμός των βημάτων (υπό-ενεργειών) σε μία μέθοδο και  $L$  είναι το μέγιστο μήκος του attentional state (μία παρ' όμοια μέθοδο χρησιμοποιείται στην εργασία [Lesh, 1999]). Σε μία επιφάνεια άμεσου χειρισμού το  $L$  είναι πάντα μικρό, διότι το μήκος κάθε σύνθετης ενέργειας από τις βασικές είναι μικρό (συνήθως  $\leq 3$ ).

Επιστρέφοντας στο παράδειγμά μας, αν ο χρήστης πραγματοποιήσει μία ακόμα ενέργεια  $\beta$  η οποία συμβάλει μόνο στην επίτευξη του στόχου  $g_1$ , τότε ο πράκτορας αναθεωρεί τη γνώση του για τους πιθανούς πρωταρχικούς στόχους του χρήστη σε ένα και τις υποθέσεις του για τα πιθανούς τρέχοντες στόχους του (attentional states) σε δύο (βλ. Εικόνα 39(β)). Για τις άλλες υποθέσεις ο πράκτορας πιστεύει ότι δεν είναι έγκυρες διότι δεν μπορεί να βρει μία ερμηνεία (attentional state) που να εξηγεί την ενέργεια  $\beta$  στα συγκεκριμένο πλαίσιο ενεργειών του πιθανού πρωταρχικού στόχου  $g_2$ .

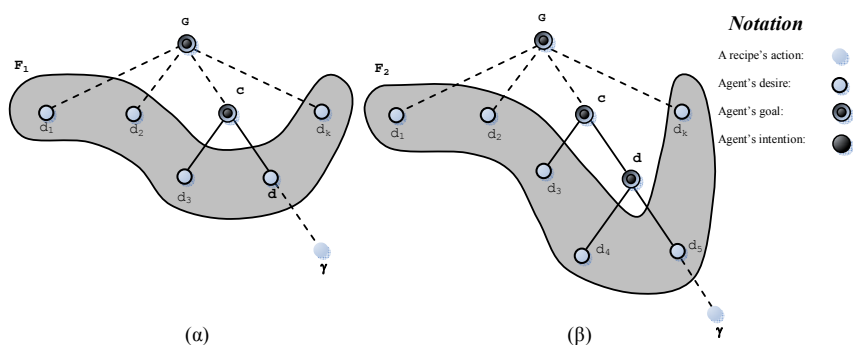
Έχοντας αναγνωρίσει έναν μοναδικό πρωταρχικό στόχο  $g_1$ , ο πράκτορας (μέσω του αντίστοιχου κανόνα αναγνώρισης καταστάσεων που περιγράψαμε στην ενότητα 6.7) σχηματίζει μία επιθυμία να πραγματοποιήσει το  $g_1$ . Αποτέλεσμα της επιθυμίας αυτής είναι να επιλέξει μία σχετική και εφαρμόσιμη μέθοδο. Στη περίπτωση που αναγνωρίσει



ότι δεν υπάρχει εφαρμόσιμη μέθοδος, τότε παρακινεί το χρήστη για συνεργασία χωρίς ν' αποσπά τη προσοχή του (βλ. Εικόνα 10(α)).

### 6.8.2. Ερμηνεία ενεργειών χρήστη υπό το πλαίσιο ενεργειών ενός πρωταρχικού στόχου

Ας υποθέσουμε ότι είτε ο πράκτορας έχει αναγνωρίσει ένα μοναδικό πιθανό πρωταρχικό στόχο του χρήστη, είτε ότι ο χρήστης έχει αρχίσει τη συνεργασία με τον πράκτορα και μοιράζονται ένα κοινό στόχο, ας πούμε  $G$ . Και στις δύο αυτές περιπτώσεις ο πράκτορας έχει τη πρόθεση να βοηθήσει το χρήστη να πραγματοποιήσει επιτυχώς το στόχο  $G$  (δηλαδή, έχει σχηματίσει μία πρόθεση της μορφής  $\langle \text{int\_that\_cba} \rangle$ ). Επίσης, ας υποθέσουμε ότι ο χρήστης πραγματοποιεί στην επιφάνεια διεπαφής την ενέργεια  $\gamma$ . Όπως τονίσαμε στις παραπάνω υπό-ενότητες, ο πράκτορας αντιλαμβανόμενος την ενέργεια αυτή την ερμηνεύει κάτω από το πλαίσιο ενεργειών της  $G$  σχηματίζοντας υποθέσεις (attentional states) για τη συνεισφορά αυτής στην επίτευξη του στόχου.



Εικόνα 40: (α) Το σύνολο των ενεργειών  $F_1$  αποτελεί τη περιοχή παρυφής του στόχου  $G$  πριν ο χρήστης πραγματοποιήσει την ενέργεια  $\gamma$ . (β) Το σύνολο των ενεργειών  $F_2$  αποτελεί τη περιοχή παρυφής του στόχου  $G$  μετά την εκτέλεση της ενέργειας  $\gamma$ .

Το σύνολο των ενεργειών (στο πλαίσιο ενεργειών του  $G$ ) που ο πράκτορας επιθυμεί να πραγματοποιήσει αλλά δεν έχει περάσει τις αντίστοιχες μεθόδους τους από τη διαδικασία του εντοπισμού και επίλυσης συγκρούσεων ονομάζεται περιοχή παρυφής (fringe) του στόχου  $G$  (βλ. Εικόνα 40(α)). Η περιοχή παρυφής ενός στόχου  $G$  δηλώνει το ποιες ενέργειες πρέπει να ληφθούν υπόψη ως πιθανές ερμηνείες των μελλοντικών ενεργειών του χρήστη και επομένως ο πράκτορας για αυτές θα πρέπει να ελέγξει αν οι μέθοδοι αυτών των ενεργειών είναι εφαρμόσιμες. Αρχικώς η περιοχή παρυφής περιέχει μόνο το

G και ενημερώνεται κάθε φορά που ο πράκτορας ελέγχει την εφαρμοσιμότητα μίας ενέργειας που η περιοχή αυτή περιλαμβάνει.

Πιο συγκεκριμένα, όπως δείχνει η Εικόνα 40, ο πράκτορας ενημερώνει τη περιοχή παρυφής μίας ενέργειας G και αναγνωρίζει την ανάγκη για συνεργασία όπως παρακάτω:

UPDATE\_FRINGE (G,  $\gamma$ )

Έστω ότι  $C_G = \{C_G^1, C_G^2, \dots, C_G^n\}$  είναι το σύνολο όλων των πεδίων ενεργειών που μπορούν να υλοποιήσουν την ενέργεια G και  $C \in C_G$  το τρέχον πεδίο ενεργειών που έχει ήδη σχηματιστεί.

Έστω  $F = \{d_1, d_2, \dots, d, \dots, d_k\}$  είναι η τρέχουσα περιοχή παρυφής της G στο πλαίσιο ενεργειών C και

$I_C^\gamma = \{I_1, I_2, \dots, I_m\}$  να είναι το σύνολο άλλων των τρεχόντων στόχων (attentional states) που ερμηνεύουν τη  $\gamma$  στο πεδίο ενεργειών C.

**Αν** το σύνολο  $I_C^\gamma$  δεν είναι κενό τότε

**Αν**  $I_C^\gamma$  περιέχει μόνο ένα σύνολο τρεχόντων στόχων (attentional state)

$I = (G, \dots, c, d, \dots, \gamma)$  και  $d \in F$  τότε

– Η ενέργεια  $d$  ορίζεται ως εκείνη όπου οι τρέχοντες στόχοι συναντούν τη περιοχή παρυφής της G και ονομάζεται τρέχουσα ενέργεια εστίασης της προσοχής του πράκτορα ή *ενέργεια βασικού ενδιαφέροντος* (focal action) του πλαισίου ενεργειών  $c$  της G. Το μονοπάτι  $(G, \dots, c)$  του συνόλου των τρεχόντων στόχων της  $\gamma$  από την G ως τη περιοχή παρυφής περιέχει όλες εκείνες τις ενέργειες που η  $\gamma$  συνεισφέρει και ο πράκτορας έχει ήδη βρει για αυτές από μία εφαρμοσιμη μέθοδο. Το σύνολο αυτών των ενεργειών ονομάζεται *τρέχοντες εφαρμοσιμοί στόχοι της  $\gamma$*  (reconciled attentional state of  $\gamma$ ) στο πλαίσιο ενεργειών C της G. Κάθε ενέργεια του συνόλου αυτού θεωρείται ένα πιθανός στόχος (candidate goal) του χρήστη.

– Ο πράκτορας δεσμεύεται να πραγματοποιήσει επιτυχώς την ενέργεια βασικού ενδιαφέροντος (δηλαδή σχηματίζει μία δέσμευση της μορφής  $\langle \text{int\_that\_cba} \rangle$  για τη  $d$ ).

**Αν** η μέθοδος της  $d$  είναι εφαρμοσιμη τότε:

– Έστω  $F_1 = \{d_{01}, \dots, d_{0m}\}$  να είναι το σύνολο όλων των υπό-ενεργειών της εφαρμοσιμης μεθόδου της  $d$ .

– Ανέπτυξε τη μέθοδο σχηματίζοντας νέες επιθυμίες για όλες τις ενέργειες της  $F_1$ .

– Στη περίπτωση που η ενέργεια G είναι κοινή, τότε σχημάτισε μία δέσμευση πραγματοποίησης της  $d$  από κοινού με τον χρήστη  $\langle \text{int\_that\_do} \rangle$  και παρουσίασε τις υποενέργειες  $F_1$  στο χρήστη καθώς επίσης τους τρέχοντες στόχους που έχει ελέγξει την εφαρμοσιμότητα των μεθόδων τους (reconciled attentional state).

– Η νέα περιοχή παρυφής  $F_2$  της G στο πλαίσιο ενεργειών C είναι η  $F \cup F_1 - \{d\}$ .

– Σταμάτα την εκτέλεση της διαδικασίας (break the loop).

**αλλιώς**

– Ο πράκτορας προχωρά στην επίλυση της σύγκρουσης χρησιμοποιώντας τις κατευθύνσεις επίλυσης συγκρούσεων της μεθόδου.

#### αλλιώς

Αν το σύνολο των τρεχόντων στόχων  $I_C^V$  περιέχει περισσότερα από ένα σύνολα τρεχόντων στόχων (λόγω της ύπαρξης περισσότερες από μίας σχετικής μεθόδου για τουλάχιστον μία ενέργεια), τότε για κάθε τέτοιο σύνολο ο πράκτορας θεωρεί ότι υπάρχει ένα εναλλακτικό πεδίο ενεργειών για τη  $G$ .

Πιο συγκεκριμένα, αν  $C_{G^1}, C_{G^2}, \dots, C_{G^r}$  είναι τα νέα πλαίσια ενεργειών της  $G$  τότε το νέο σύνολο με όλα τα πλαίσια ενεργειών της  $G$  είναι:  $C_G = \{C_G^1, C_G^2, \dots, C_G^n\} \cup \{C_{G^1}, C_{G^2}, \dots, C_{G^r}\}$ .

#### τέλος.

Σύμφωνα με τα παραπάνω, για κάθε βασική ενέργεια που ο χρήστης πραγματοποιεί ερμηνεύεται κάτω από το πλαίσιο ενεργειών της  $G$ . Με αυτό τον τρόπο, όσο ο χρήστης ενεργεί στην επιφάνεια διεπαφής, ο πράκτορας εργάζεται με δύο τρόπους:

- Από τους κύριους στόχους (primary goals) στις βασικές ενέργειες (basic level actions), ο πράκτορας αναπτύσσει στόχους και παρέχει βοήθεια στο χρήστη ελέγχοντας την εφαρμοσιμότητα των μεθόδων.
- Από τις βασικές ενέργειες στους στόχους, ο πράκτορας προσπαθεί να ερμηνεύσει τις ενέργειες του χρήστη υπό τη περιοχή παρυφής της  $G$ .

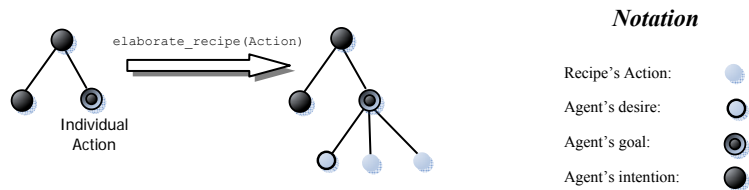
## 6.9. Ανάπτυξη ιεραρχικών πλάνων

Όπως παρουσιάσαμε στην υποενότητα 4.2.4, η διεργασία συλλογισμού ανάπτυξης πλάνων καθιστά τον πράκτορα ικανό να ελέγχει τις νοητικές ενέργειες `<select_recipe>` και `<elaborate_recipe>` με τις οποίες βρίσκει σχετικές μεθόδους για τις ενέργειες που επιθυμεί να πραγματοποιήσει και τις αναπτύσσει είτε μόνος του είτε σε συνεργασία με άλλους πράκτορες.

Στη περίπτωση του πράκτορα διεπαφής, όταν ο πράκτορας έχει ως στόχο να πραγματοποιήσει μία ενέργεια (`goal_to(Agent, Action)`), ανάλογα με το αν η ενέργεια είναι ατομική ή κοινή ή αντιστοιχεί σε ένα πιθανό πρωταρχικό στόχο του χρήστη, τότε ο πράκτορας επιλέγει μία από τις παρακάτω τρεις μεθόδους για τη νοητική ενέργεια `elaborate_recipe(Action)`:

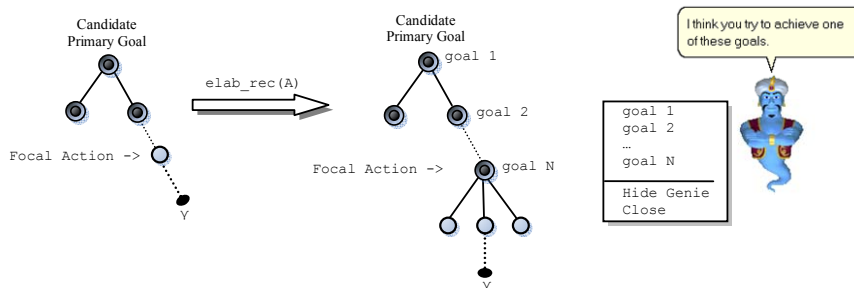
- (α) Αν η ενέργεια `Action` είναι ατομική τότε ο πράκτορας θεωρεί τη νοητική ενέργεια `elaborate_recipe(Action)` ως βασική (basic level action) και ενεργεί όπως πα-

ρακάτω: Από τις υπό-ενέργειες της μεθόδου της `Action` που δεν έχει ακόμα αναπτύξει βρίσκει την επόμενη (ή τις επόμενες) και σχηματίζει την επιθυμία να τη πραγματοποιήσει (ή τις επιθυμίες να τις πραγματοποιήσει). Αν δεν υπάρχει καμία τότε πιστεύει ότι έχει ένα πλήρες πλάνο για την `Action` και προσπαθεί να αναπτύξει τη μέθοδο της επόμενης ενέργειας της μεθόδου που η `Action` ανήκει.



Εικόνα 41: Ανάπτυξη μεθόδου ατομικής ενέργειας.

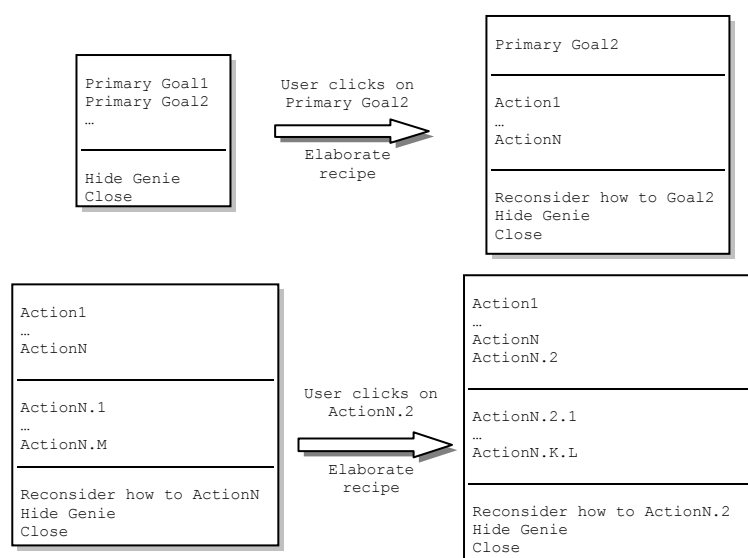
- (β) Αν ο πράκτορας έχει αναγνωρίσει ακριβώς ένα πιθανό πρωταρχικό στόχο μετά τη πραγματοποίηση από το χρήστη της ενέργειας  $\gamma$  και η `Action` είναι ενέργεια βασικού ενδιαφέροντος (focal action) τότε ενεργεί όπως παρακάτω: (i) δημιουργεί από μία επιθυμία για κάθε υπό-ενέργεια που περιέχει η μέθοδος της `Action` (ενημέρωση της περιοχής παρυφής του πρωταρχικού στόχου) και (ii) σχηματίζει την επιθυμία να παρουσιάσει στο χρήστη την υπόθεση που έχει κάνει σχετικά με το σύνολο των τρεχόντων στόχων του χρήστη για τους οποίους πιστεύει ότι είναι εφαρμόσιμοι (reconciled attentional state).



Εικόνα 42: Ανάπτυξη μεθόδου για ένα πιθανό στόχο του χρήστη.

- (γ) Αν ο πράκτορας μοιράζεται με το χρήστη την ενέργεια `Action` τότε πραγματοποιώντας την `elaborate_recipe(Action)` ενημερώνει το χρήστη για τη σχετική και εφαρμόσιμη μέθοδο που έχει επιλέξει καθώς και για τους λόγους για τους οποίους η μέθοδος αυτή πρέπει να πραγματοποιηθεί. Στη περίπτωση αυτή η μέθοδος της `<elaborate_recipe>` περιέχει τις ακόλουθες ενέργειες: παρουσιάζει ένα πλαίσιο

διαλόγου τριών πεδίων όπου στο πρώτο εμφανίζει τους τρέχοντες εφαρμόσιμους κοινούς στόχους (reconciled attentional state) για τους οποίους γίνεται η κοινή ενέργεια βασικού ενδιαφέροντος, στο δεύτερο πεδίο παρουσιάζεται η σχετική και εφαρμόσιμη μέθοδος που ο πράκτορας έχει επιλέξει για την Action και στο τρίτο ένα σύνολο επιλογών με τις οποίες ο χρήστης μπορεί να απορρίψει τη μέθοδο ή να κρύψει το πράκτορα τερματίζοντας τη συνεργασία μαζί του για αυτό το κύριο στόχο ή να κλείσει το πράκτορα απενεργοποιώντας τον.



Εικόνα 43: Αναπτύσσοντας μία σχετική και εφαρμόσιμη μέθοδο για μία κοινή ενέργεια.

Η γνώση ενός πλήρους πλάνου για τη νοητική ενέργεια `<elaborate_recipe>` έχει ως αποτέλεσμα το σχηματισμό προθέσεων της μορφής `<int_to>` για τις ενέργειες που ο πράκτορας στοχεύει να πραγματοποιήσει είτε προσωπικά είτε σε συνεργασία με το χρήστη. Αν όμως η μέθοδος που προσπαθεί ν' αναπτύξει ανήκει σε πιθανό στόχο του χρήστη (βλ. τη πρώτη από τις τρεις παραπάνω περιπτώσεις) τότε ο πράκτορας δεν σχηματίζει τη πρόθεση να πραγματοποιήσει την ενέργεια που υλοποιεί η μέθοδος αυτή διότι δεν γνωρίζει ακόμα αν πράγματι ο χρήστης στοχεύει να πραγματοποιήσει την ενέργεια αυτή. Για να δεσμευτεί ο πράκτορας να πραγματοποιήσει την ενέργεια αυτή θα πρέπει ο χρήστης να δηλώσει κατηγορηματικά στο πράκτορα ότι πράγματι είναι ο στόχος του επιλέγοντας την από το πλαίσιο διαλόγου και αρχίζοντας έτσι τη συνεργασία με το πράκτορα.

Συνοψίζοντας, οποτεδήποτε ο πράκτορας διεπαφής στοχεύει να πραγματοποιήσει μία ενέργεια η μονάδα ανάπτυξης πλάνων δημιουργεί την επιθυμία να πραγματοποιήσει τη νοητική ενέργεια <elaborate\_recipe>. Η νοητική ενέργεια αυτή οδηγεί τον πράκτορα είτε (α) να παρουσιάσει στο χρήστη τους πιθανούς τρέχοντες στόχους που ο πράκτορας πιστεύει ότι ο χρήστης έχει (αν η συνεργασία με το χρήστη δεν έχει ακόμα αρχίσει), είτε (β) να παρουσιάσει τα βήματα των ενεργειών που πρέπει να γίνουν από τον πράκτορα ή από τον χρήστη προκειμένου να επιτευχθεί ένας κοινός στόχος που και οι δύο έχουν συμφωνήσει να πραγματοποιήσουν, είτε (γ) να αναπτύξει τη μέθοδο περαιτέρω δημιουργώντας επιθυμίες για τις υπό-ενέργειες που πρέπει να πραγματοποιήσει προσωπικά.

## **6.10. Παράδειγμα συνεργασίας μεταξύ πράκτορα διεπαφής και χρήστη**

Όπως παρουσιάσαμε παραπάνω, ο πράκτορας παρατηρεί την επιφάνεια διεπαφής, αντιλαμβάνεται τις ενέργειες που πραγματοποιεί ο χρήστης και επεξεργάζεται μόνο αυτές που ερμηνεύονται είτε κάτω από ένα κοινό στόχο είτε ως υπό-ενέργειες ενός πιθανού πρωταρχικού στόχου.

Ο πράκτορας παρουσιάζει τις υποθέσεις του σχετικά με τους στόχους του χρήστη παρακινώντας αυτόν για συνεργασία όταν πιστεύει ότι κάτι κάνει λάθος. Στη περίπτωση που ο χρήστης αποδεχθεί το αίτημα για συνεργασία τότε και οι δύο μοιράζονται ένα στόχο καθώς και μία μέθοδο επίτευξης του στόχου αυτού. Επιπλέον ο πράκτορας δεσμεύεται να μπορεί ο χρήστης να πραγματοποιήσει τις υπό-ενέργειες που η κοινή μέθοδος περιλαμβάνει. Αυτό έχει ως αποτέλεσμα ο πράκτορας να βοηθά τον χρήστη όταν αντιλαμβάνεται ότι ενεργεί λανθασμένα.

Οι παράγραφοι που ακολουθούν εξηγούν τη συμπεριφορά του πράκτορα διεπαφής στο περιβάλλον του Δημιουργού Μοντέλων και πιο συγκεκριμένα στη περίπτωση συνεργασίας που παρουσιάσαμε στην υποενότητα 2.5.2. Το παράδειγμα περιγράφει τον τρόπο με τον οποίο ο πράκτορας διεπαφής αναγνωρίζει τις προθέσεις του χρήστη, αντιλαμβάνεται το πότε ο χρήστης έχει την ανάγκη για βοήθεια, τον τρόπο με τον οποίο συμφωνούν και αναπτύσσουν το κοινό τους πλάνο καθώς και τον τρόπο με τον οποίο μοιράζονται τις εργασίες μεταξύ τους.

Για να γίνει πιο κατανοητός ο τρόπος με το οποίο ο πράκτορας χειρίζεται τις διεργασίες συλλογισμού παρουσιάζουμε το παράδειγμα σε δύο στήλες. Στην αριστερή στήλη, εμφανίζουμε τις διεργασίες συλλογισμού που ενεργοποιούνται, ενώ στη δεξιά παρουσιάζουμε τις ενέργειες του χρήστη και τη συμπεριφορά του πράκτορα.

**Αρχική κατάσταση της επιφάνειας διεπαφής:** Ο χρήστης έχει ήδη τοποθετήσει στην επιφάνεια δημιουργίας μοντέλων ένα *βαρέλι* και ένα *ρολόι*. Στο ρολόι έχει επιλέξει την ιδιότητα *χρόνος*, ενώ στο *βαρέλι* δεν έχει επιλέξει καμία ιδιότητα.

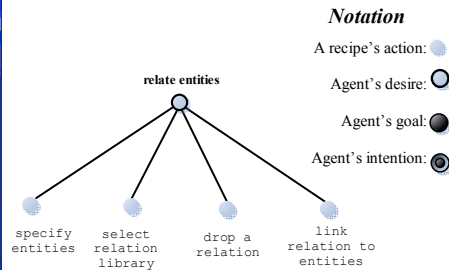
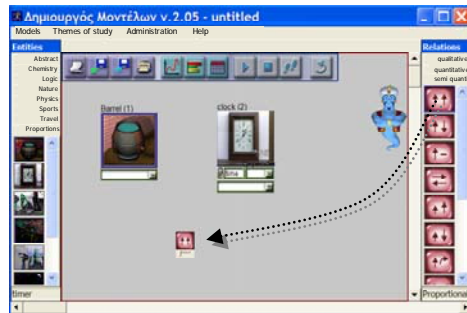
**Αρχική κατάσταση της νοητικής κατάστασης του πράκτορα διεπαφής:** Ο πράκτορας έχει αναγνωρίσει δύο πιθανούς πρωταρχικούς στόχους του χρήστη. Ο πρώτος είναι η συσχέτιση δύο οντοτήτων "Relate entities", ενώ ο δεύτερος είναι το τρέξιμο του μοντέλου "Run the model".

Perception Module      Ας υποθέσουμε ότι ο χρήστης εισάγει μία σχέση από τη βιβλιοθήκη σχέσεων μέσα στο χώρο δημιουργίας μοντέλων. Ο πράκτορας αντιλαμβάνεται την ενέργεια αυτή, μειώνει τον αριθμό των υποθέσεων των πιθανών πρωταρχικών στόχων του χρήστη σε έναν, αυτό της δημιουργίας σχέσης "Relate entities". Η υπόθεση αυτή σχηματίζεται διότι η ενέργεια της εισαγωγής σχέσης "Drop relation" που ο χρήστης πραγματοποιεί συνεισφέρει μόνο στην ενέργεια "Relate entities".

Situation Recognition Module      Έχοντας μία μοναδική υπόθεση για το πρωταρχικό στόχο του χρήστη, ο πράκτορας μέσω του αντίστοιχου κανόνα αναγνώρισης κατάστασης σχηματίζει την επιθυμία να πραγματοποιήσει την ενέργεια που αντιστοιχεί στο πρωταρχικό στόχο του χρήστη.

Plan Elaboration Module      Οποτεδήποτε ο πράκτορας σχηματίζει μία επιθυμία να πραγματοποιήσει μία ενέργεια, βρίσκει επίσης μία σχετική μέθοδο πραγματοποίησης αυτής. Για την ενέργεια "Relate entities" ο πράκτορας επιλέγει τη μέθοδο του Πίνακα 11(α). Ο στόχος της μεθόδου αυτής είναι να συσχετίσει δύο οντότητες χρησιμοποιώντας μία σχέση. Η λίστα ενεργειών της μεθόδου εμπεριέχει τις ακόλουθες τέσσερις ενέργειες: (α) καθόρισε τις οντότητες "Specify entities", (β) επέλεξε μία βιβλιοθήκη σχέσεων "Select relation library", (γ) εισήγαγε μία σχέση "Drop a relation" και (δ) συνέδεσε τις δύο οντότητες με τη σχέση "Link relation to entities".

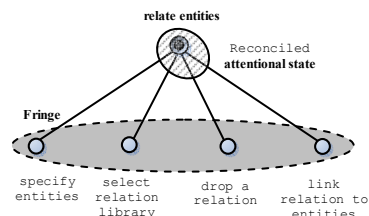
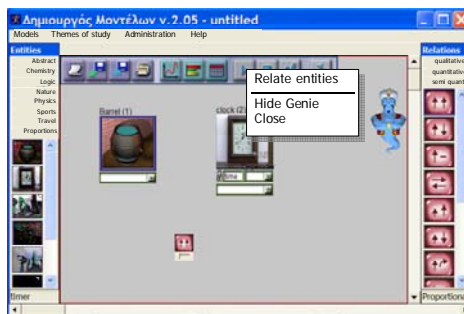
Σύμφωνα με τις κατευθύνσεις εντοπισμού συγκρούσεων (Find Conflicts Directives (FCD)) της μεθόδου, οι συνθήκες εφαρμογής της μεθόδου θεωρούνται συνθήκες του πλαισίου ενεργειών και άρα θα πρέπει να διατηρούνται κατά τη διάρκεια σχεδιασμού και εκτέλεσης της μεθόδου. Πιο συγκεκριμένα, σύμφωνα με τις συνθήκες αυτές θα πρέπει να υπάρχουν τουλάχιστον δύο οντότητες μέσα στο χώρο δημιουργίας μοντέλων οι οποίες να μπορούν να συσχετιστούν με τη συγκεκριμένη σχέση και επίσης να είναι ορατές (δηλαδή να μην επικαλύπτονται από άλλες οντότητες ή σχέσεις).



Εικόνα 44: Ο χρήστης εισάγει μία σχέση μέσα στο χώρο δημιουργίας μοντέλων. Ο πράκτορας αναγνωρίζει την ενέργεια "Relate entities" ως μοναδικό πιθανό πρωταρχικό στόχο του χρήστη και επιλέγει μία σχετική μέθοδο.

**Reconciliation Module** Έχοντας βρει μία σχετική μέθοδο για την ενέργεια "Relate entities", ο πράκτορας δεσμεύεται να πραγματοποιηθεί ο πρωταρχικός στόχος του χρήστη επιτυχώς (σηματίζοντας μία δέσμευση της μορφής  $\langle \text{int\_that\_cba} \rangle$ ). Ως αποτέλεσμα αυτής της δέσμευσης, ο πράκτορας ελέγχει αν η μέθοδος είναι εφαρμόσιμη (μέσω του Reconciliation Module). Παρατηρώντας ότι η μέθοδος είναι πράγματι εφαρμόσιμη (ελέγχοντας τη συνθήκη *Applicability Mental Condition* της μεθόδου του Πίνακα  $I1(\alpha)$ ) ο πράκτορας στοχεύει να πραγματοποιήσει την ενέργεια "Relate entities".

**Plan Elaboration Module** Έχοντας ως στόχο να πραγματοποιήσει την ενέργεια αυτή, ο πράκτορας προχωρά στην ανάπτυξη της εφαρμόσιμης μεθόδου μέσω της νοητικής ενέργειας ενημέρωσης περιοχής παρυφής (*update fringe*) ως εξής: (α) ενημερώνει τη περιοχή παρυφής της ενέργειας "Relate entities" σχηματίζοντας νέες επιθυμίες για τις υπό-ενέργειες της και (β) δημιουργεί την επιθυμία να παρουσιάσει στο χρήστη τους λόγους για τους οποίους πιστεύει ότι γίνεται η ενέργεια "Drop a relation" (δηλαδή να παρουσιάσει το reconciled attentional state της ενέργειας "Drop a relation").





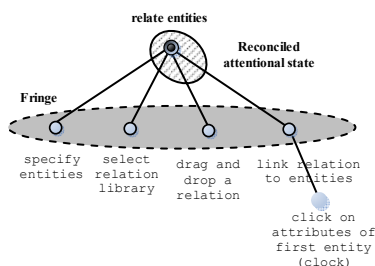
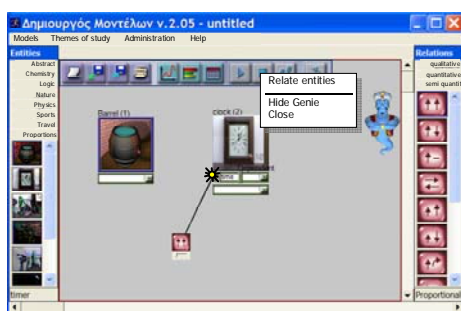
*Εικόνα 45:* Ο πράκτορας ελέγχει την εφαρμοσιμότητα της σχετικής μεθόδου της ενέργειας "Relate entities" στο τρέχον πλαίσιο ενεργειών. Αναγνωρίζοντας ότι είναι πράγματι εφαρμόσιμη παρουσιάζει την ενέργεια αυτή στο χρήστη.

Ο πράκτορας πραγματοποιεί τη πρώτη από τις δύο παραπάνω υπό-ενέργειες της ενημέρωσης περιοχής παρυφής προκειμένου να μπορεί να ερμηνεύσει τις επόμενες ενέργειες του χρήστη στο τρέχον πλαίσιο ενεργειών. Η δεύτερη νοητική υπό-ενέργεια πραγματοποιείται προκειμένου ο πράκτορας να παρουσιάσει τις υποθέσεις του στο χρήστη. Το πλαίσιο διαλόγου που ο πράκτορας παρουσιάζει επιτρέπει στο χρήστη να δηλώσει τι ακριβώς επιθυμεί να κάνει και να αρχίσει τη συνεργασία με τον πράκτορα. Το τρέχον σύνολο εφαρμόσιμων στόχων περιλαμβάνει μόνο την ενέργεια "Relate entities" ενώ η περιοχή παρυφής της ενέργειας αυτής περιλαμβάνει τις ενέργειες "Specify entities", "Select relation library", "Drop a relation" and "Link relation to entities".

Perception  
Module

Ας υποθέσουμε ότι ο χρήστης αγνοεί τη πρόταση του πράκτορα για συνεργασία και συνεχίζει να ενεργεί κάνοντας κλικ σε μία από τις ιδιότητες της οντότητας «ρολόι» προκειμένου να τη συνδέσει με τη σχέση. Ο πράκτορας επιλέγοντας τη μέθοδο (α) του Πίνακα 12 ερμηνεύει τη νέα ενέργεια του χρήστη ως υπό-ενέργεια της ενέργειας "Link relation to entities".

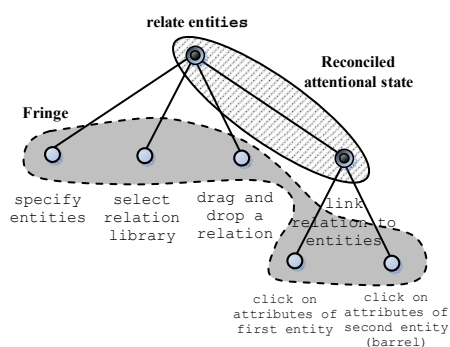
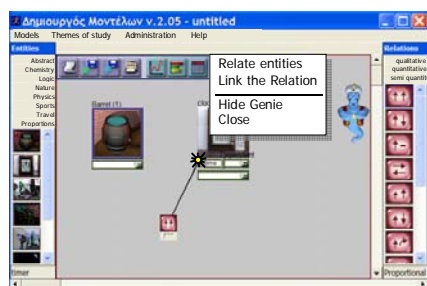
Στο σημείο αυτό θα πρέπει να σημειώσουμε ότι ο πράκτορας (όπως είναι φυσικό) δεν γνωρίζει ποιες ακριβώς οντότητες ο χρήστης θέλει να συσχετίσει. Για το λόγο αυτό οι μεταβλητές της μεθόδου του Πίνακα 11(α) που αντιστοιχούν στις οντότητες αυτές δεν έχουν πάρει κάποια τιμή. Η επιλογή του χρήστη μίας ιδιότητας της οντότητας ρολόι έχει ως αποτέλεσμα τη μετάδοση (propagation) της πληροφορίας ότι μία από τις οντότητες που θέλει να συσχετίσει ο χρήστης είναι το «ρολόι» στη μέθοδο της ενέργειας "Specify entities". Η πληροφορία αυτή μεταφέρεται μέσω της ενοποίησης μεταβλητών στη γονική ενέργεια "Link relation to entities" και μέσω της ενέργειας "Relate entities" μεταδίδεται στη "Specify entities".



*Εικόνα 46:* Ο χρήστης επιλέγει τη πρώτη οντότητα κάνοντας κλικ στην ιδιότητά της αγνοώντας τη προτροπή του πράκτορα για συνεργασία.

**Reconciliation Module** Βασίζόμενοι στη κατάσταση των τρεχόντων στόχων του πλαισίου ενεργειών και έχοντας ενοποιήσει όλες τις μεταβλητές που σχετίζονται με την επιλογή της οντότητας «ρολόι», ο πράκτορας θεωρεί τη σύνθετη ενέργεια "Link relation to entities" ως ενέργεια βασικού ενδιαφέροντος στο τρέχον πλαίσιο ενεργειών. Ο πράκτορας σχηματίζει τη πρόθεση (<intention\_that\_cba>) ότι η ενέργεια "Link relation to entities" πρέπει να πραγματοποιηθεί επιτυχώς χρησιμοποιώντας τη μέθοδο του (α) του Πίνακα 12. Ελέγχοντας την εφαρμοσιμότητα της μεθόδου αυτής στο τρέχον πλαίσιο ενεργειών, ο πράκτορας δεν εντοπίζει κάποια σύγκρουση (παρατηρήστε ότι μέχρι τώρα μόνο οι μεταβλητές που αφορούν τη πρώτη οντότητα έχουν πάρει τιμή). Αυτό οδηγεί το πράκτορα να (α) έχει ως στόχο τη πραγματοποίηση της ενέργειας βασικού ενδιαφέροντος, (β) σχηματίζει από μία επιθυμία για κάθε υπό-ενέργεια της μεθόδου και (γ) να παρουσιάσει στο χρήστη τους εφαρμόσιμους στόχους που έχει αναγνωρίσει μέχρι τη στιγμή αυτή: "Relate entities" και "Link relation to entities".

**Update Fringe**



**Εικόνα 47:** Ο πράκτορας ερμηνεύει την ενέργεια του χρήστη ως υπό-ενέργεια της σύνθετης ενέργειας "Link relation to entities" και βρίσκει μία σχετική και εφαρμόσιμη μέθοδο για αυτή. Επίσης, ενημερώνει τη περιοχή παρυφής και τους τρέχοντες εφαρμόσιμους στόχους (reconciled attentional state) και ενημερώνει το χρήστη για τις υποθέσεις του σχετικά για τους πιθανούς πρωταρχικούς στόχους του.

**Perception Module** Ο χρήστης κάνει κλικ στην οντότητα «βαρέλι» προσπαθώντας να ενώσει τη σχέση με το βαρέλι. Η ενέργεια αυτή είναι λάθος αφού ο χρήστης προσπαθεί να δημιουργήσει μία σχέση μεταξύ δύο οντοτήτων χωρίς πρώτα να έχει καθορίσει τις ιδιότητες και των δύο οντοτήτων. Η επιφάνεια διεπαφής του Δημιουργού Μοντέλων διαγράφει τη σχέση αυτή ως λάθος.

**Reconciliation Module** Ερμηνεύοντας τη νέα αυτή ενέργεια του χρήστη ως υπό-ενέργεια της ενέργειας "Click on the attribute of the 2<sup>nd</sup> entity" στο τρέχον πλαίσιο ενεργειών, το σύνολο των τρεχόντων στόχων περιλαμβάνει τις ενέργειες "Relate entities" και "Link relation to entities", ενώ η τρέχουσα ενέργεια βασικού ενδιαφέροντος είναι η "Click on the attribute of the 2<sup>nd</sup> entity".

Μεταδίδοντας τη πληροφορία της επιλογής της οντότητας «βαρέλι» στο τρέχον πλαίσιο ενεργειών και σχηματίζοντας τη πρόθεση να πραγματοποιηθεί επιτυχώς η ενέργεια "Click on the attribute of the 2<sup>nd</sup> entity" ο πρά-

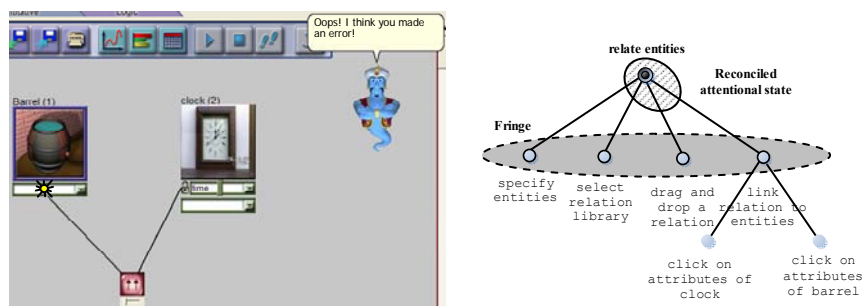
κτορας αναγνωρίζει ότι οι περιορισμοί του πλαισίου ενεργειών της "Link relation to entities" δεν ισχύουν λόγω του γεγονότος ότι ο χρήστης δεν έχει καθορίσει καμία ιδιότητα της οντότητας "βαρέλι".

Έχοντας εντοπίσει τη σύγκρουση, ο πράκτορας σχηματίζει την επιθυμία να επιλύσει αυτή χρησιμοποιώντας τις κατευθύνσεις συμπεριφοράς [reconsider, backtrack, resolve] της μεθόδου της ενέργειας "Link relation to entities".

Δεδομένου ότι δεν υπάρχουν εναλλακτικές μέθοδοι για την ενέργεια "Link relation to entities" ούτε εναλλακτικά πλαίσια ενεργειών ο πράκτορας απορρίπτει το στόχο που ήδη έχει για την ενέργεια αυτή (καθώς επίσης και τις επιθυμίες που ήδη έχει σχηματίσει για τις δύο υπό-ενέργειές της), σχηματίζει μία νέα επιθυμία για την ενέργεια αυτή και πληροφορεί το χρήστη ότι κάτι πάει λάθος.

Η τρέχουσα κατάσταση των στόχων του πράκτορα περιλαμβάνει μόνο το πιθανό τρέχοντα στόχο "Relate entities", η περιοχή παρυφής περιλαμβάνει τις ενέργειες "Specify entities", "Select relation library", "Drag and drop a relation" και "Link relation to entities". Τέλος η τρέχουσα ενέργεια βασικού ενδιαφέροντος είναι η "Link relation to objects".

Ο χρήστης έχει δύο επιλογές: είτε να αγνοήσει τη προτροπή του πράκτορα για συνεργασία είτε να κάνει κλικ πάνω αρχίζοντας έτσι τη συνεργασία μαζί του.



Εικόνα 48: Ο χρήστης πραγματοποιεί μία λανθασμένη ενέργεια: προσπαθεί να συνδέσει της σχέση με μία οντότητα που δεν έχει καθορίσει τουλάχιστον μία ιδιότητά της.

#### Reconciliation Module

Στο παράδειγμά μας, η ενέργεια που δημιουργεί τη σύγκρουση είναι η "Link relation to objects" (αφού οι περιορισμοί πλαισίου ενεργειών της μεθόδου δεν ισχύουν). Αν ο χρήστης κάνει κλικ στο πράκτορα τότε ζητώντας τη βοήθειά του ο πράκτορας: (α) αναγνωρίζει ότι η ενέργεια "Specify entities" επιλύει τη σύγκρουση, (β) εξηγεί τη σύγκρουση στο χρήστη μετακινούμενος στην οντότητα «βαρέλι» και δείχνοντας στις ιδιότητές του εμφανίζει το ακόλουθο μπαλόνι κειμένου "You cannot link the relation with the barrel, because you have not specified any attributes of it." και (γ)

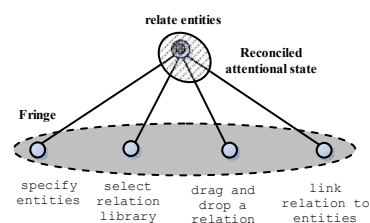
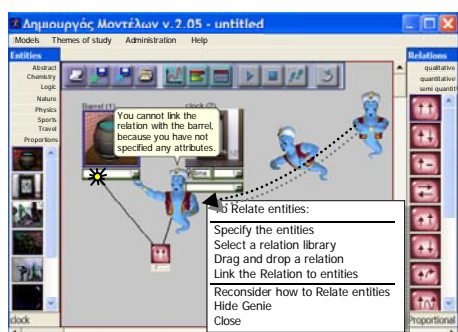
αρχίζει τη συνεργασία με το χρήστη για την ενέργεια "Relate entities" η οποία είναι γόνος τόσο της ενέργειας "Link relation to entities" όσο και της "Specify entities".

Όπως έχουμε ήδη αναφέρει, η συνεργασία μπορεί να αρχίσει από το χρήστη όταν αυτός επιλέξει έναν από τους πιθανούς πρωταρχικούς στόχους που παρουσιάζει ο πράκτορας ή όταν ο πράκτορας αναγνωρίζει μία λάθος ενέργεια του χρήστη και ο χρήστης ζητά τη βοήθεια του πράκτορα κάνοντας κλικ σε αυτόν. Και στις δύο περιπτώσεις ο πράκτορας σχηματίζει τη πρόθεση να συνεργαστεί με το χρήστη προκειμένου να πραγματοποιήσουν από κοινού την επιθυμητή ενέργεια.

Συνεργαζόμενος με το χρήστη ο πράκτορας παρουσιάζει στο χρήστη την μέθοδο που έχει επιλέξει για τη κοινή ενέργεια "Relate entities".

Plan Elaboration Module

Ο πράκτορας χρησιμοποιώντας ένα πλαίσιο διαλόγου τριών πεδίων παρουσιάζει τη κοινή ενέργεια και τη σχετική και εφαρμόσιμη μέθοδο που έχει επιλέξει. Με αυτό τον τρόπο ο χρήστης γνωρίζει τόσο το κοινό στόχο όσο και τη μέθοδο που πρέπει να ακολουθήσει από κοινού με το πράκτορα. Έτσι ο χρήστης γνωρίζει ότι προκειμένου να συσχετίσει οντότητες "Relate entities" πρέπει να τις καθορίσει "Specify the entities", να επιλέξει μία βιβλιοθήκη σχέσεων "Select a relation library", να μεταφέρει μία σχέση στο χώρο δημιουργίας μοντέλων "Drag and drop a relation" και τέλος να συνδέσει τη σχέση με τις οντότητες "Link the relation with the entities".



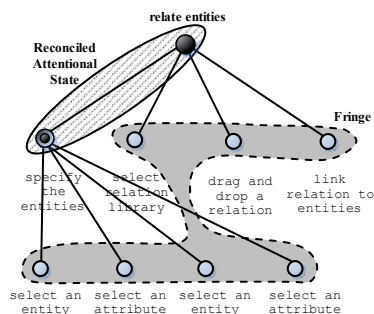
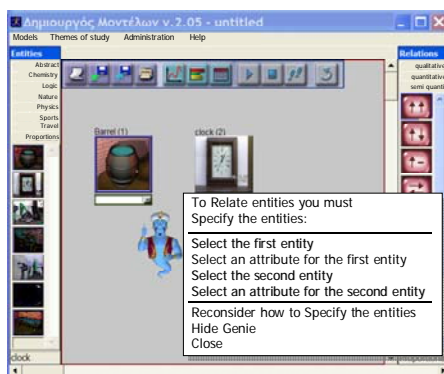
Εικόνα 49: Ο πράκτορας εξηγεί στο χρήστη το λάθος του αφού πρώτα ο χρήστης άρχισε τη συνεργασία με το πράκτορα κάνοντας κλικ πάνω σε αυτόν.

Perception

Στη περίπτωση που ο χρήστης επιλέξει μία ενέργεια της μεθόδου από το δεύτερο πεδίο του πλαισίου ενεργειών τότε αναθέτει στο πράκτορα τη πραγματοποίηση της ενέργειας αυτής ως υπό-ενέργεια του κοινού στόχου "Relate entities". Ο πράκτορας σχηματίζει την επιθυμία να επιλέξει μία σχετική μέθοδο για την υπό-ενέργεια αυτή και βρίσκοντας μία σχετική και εφαρμόσιμη μέθοδο (εξετάζοντας τις συνθήκες εφαρμογής αυτής) ο πράκτορας στοχεύει στη πραγματοποίηση αυτής. Ενημερώνοντας τη περιοχή παρυφής θεωρεί την υπό-ενέργεια ως ενέργεια βασικού ενδιαφέροντος και δημιουργεί μία επιθυμία για κάθε υπό-ενέργεια της. Τέλος σε αυτό το

Plan Elaboration Module

στάδιο σχηματίζει τη πρόθεση να τη πραγματοποιήσει σε συνεργασία με το χρήστη.



Εικόνα 50: Στη περίπτωση που ο χρήστης επιλέξει μία ενέργεια από το πλαίσιο διαλόγου τότε ο πράκτορας θεωρεί την ενέργεια αυτή ως κοινή και παρουσιάζει στο χρήστη τη σχετική και εφαρμόσιμη μέθοδο που έχει επιλέξει.

Perception  
Reconciliation  
Plan Elaboration Module

Για παράδειγμα, αν ο χρήστης επιλέξει την ενέργεια "Specify the entities" τότε ο πράκτορας σχηματίζει την επιθυμία να επιλέξει μία μέθοδο για την ενέργεια αυτή. Έτσι, μετά την επιλογή μίας σχετικής και εφαρμόσιμης μεθόδου για την υπό-ενέργεια "Specify the entities" ο πράκτορας προχωρά στην ανάπτυξη και πραγματοποίηση της μεθόδου σε συνεργασία με το χρήστη παρουσιάζοντας τη μέθοδο χρησιμοποιώντας πάλι το πλαίσιο ενεργειών τριών πεδίων.

Στη περίπτωση που ο χρήστης εξακολουθήσει να ενεργεί στην επιφάνεια διεπαφής του Δημιουργού Μοντέλων χωρίς να λαμβάνει υπ' όψη του τις υποθέσεις και παρατηρήσεις του πράκτορα, ο πράκτορας συνεχίζει να παρατηρεί τις ενέργειες του χρήστη ερμηνεύοντας αυτές υπό το τρέχον πλαίσιο ενεργειών κάνοντας κατανοητό στο χρήστη τους τρέχοντες στόχους και τη μέθοδο που επιλέγει για το στόχο βασικού ενδιαφέροντος.

Συνοψίζοντας, ο πράκτορας αρχικά παρατηρώντας τις ενέργειες του χρήστη αναγνωρίζει τους πρωταρχικούς του στόχους. Αν οι πιθανοί πρωταρχικοί στόχοι είναι παραπάνω από ένας τότε τους παρουσιάζει στο χρήστη για ν' αποσαφηνίσει τι ακριβώς θέλει να πετύχει. Αν εντοπίσει ακριβώς έναν τότε ο πράκτορας δεσμεύεται για την επιτυχή πραγματοποίηση του στόχου αυτού. Η δεσμευση αυτή έχει ως αποτέλεσμα οι επόμενες ενέργειες του χρήστη όχι μόνο να ερμηνεύονται κάτω από αυτόν το πιθανό πρωταρχικό στόχο, αλλά ταυτόχρονα να προσπαθεί να εντοπίζει τυχών συγκρούσεις μεταξύ των ενεργειών αυτών και των προϋποθέσεων εφαρμογής του στόχου.

Η συνεργασία μεταξύ χρήστη και πράκτορα αρχίζει όταν ο πρώτος είτε κάνει κλικ πάνω στον πράκτορα (στην περίπτωση που ο πράκτορας έχει εντοπίσει έναν κοινό στόχο και μία ενέργεια του χρήστη που δεν συνεισφέρει στον κοινό στόχο) είτε όταν ο χρήστης διασαφηνίζει το πρωταρχικό του στόχο επιλέγοντας μία από τις ενέργειες που του παρουσιάζει ο πράκτορας στο πλαίσιο διαλόγου. Και στις δύο αυτές περιπτώσεις ο πράκτορας δεσμεύεται να πραγματοποιήσει από κοινού το στόχο του χρήστη και να τον βοηθήσει όποτε χρειάζεται.

Κατά τη διάρκεια της συνεργασίας ο πράκτορας αναπτύσσει το κοινό στόχο κάνοντας χρήση μίας μεθόδου με την οποία εμφανίζει στον χρήστη (μέσω πλαισίων διαλόγου) όχι μόνο το τι πρέπει να κάνει (δηλαδή ποια μέθοδο πρέπει ν' ακολουθήσει) αλλά και τους λόγους για τους οποίους πρέπει να πραγματοποιηθούν οι ενέργειες της μεθόδου (δηλαδή το attentional state). Η δέσμευση του πράκτορα για την επιτυχή πραγματοποίηση της κοινής μεθόδου και η επικοινωνία μέσω πλαισίων διαλόγου έχει ως αποτέλεσμα την ανάπτυξη σχέσεων εμπιστοσύνης μεταξύ χρήστη και πράκτορα. Ο πράκτορας παρατηρώντας τις ενέργειες του χρήστη εντοπίζει τυχόν συγκρούσεις και ενημερώνει τον χρήστη τόσο για την σύγκρουση όσο και για τον τρόπο με τον οποίο θα την επιλύσει, ενώ από τη μεριά του ο χρήστης έχει την ικανότητα να ενημερώνει τον πράκτορα τι θέλει να κάνει και με ποιό τρόπο προσαρμόζοντας το κοινό πλαίσιο ενεργειών στις ανάγκες του.

# Κεφάλαιο 7ο:

## Συμπεράσματα

Στη παρούσα διατριβή παρουσιάσαμε ένα πλαίσιο ανάπτυξης νοημόνων πρακτόρων (ICAGENT) ικανών να δρουν αποτελεσματικά μόνοι τους ή σε συνεργασία με άλλους πράκτορες σε περιβάλλοντα με περιορισμένους πόρους που η κατάστασή τους αλλάζει συνεχώς και με απρόβλεπτο τρόπο. Το πλαίσιο βασίζεται στο BDI-μοντέλο ανάπτυξης πρακτόρων και στο μοντέλο συνεργασίας *SharedPlans*.

Το BDI-μοντέλο ανάπτυξης πρακτόρων προδιαγράφει/καθορίζει τις βασικές νοητικές δομές και διεργασίες για τη κατασκευή νοημόνων πρακτόρων ικανών να δημιουργούν στόχους και ν' αναπτύσσουν τα πλάνα τους με αποτελεσματικό τρόπο όταν το φυσικό τους περιβάλλον αλλάζει δυναμικά και οι πόροι (δικοί τους ή του φυσικού περιβάλλοντος) εξαντλούνται κατά τη διάρκεια του χρόνου. Σε συνδυασμό με το μοντέλο συνεργασίας *SharedPlans*, το οποίο καθορίζει τις νοητικές δομές και τις προϋποθέσεις που πρέπει να ισχύουν για να μπορούν δύο ή περισσότεροι πράκτορες να συνεργάζονται μεταξύ τους, η αρχιτεκτονική του ICAGENT υλοποιεί τα δύο αυτά μοντέλα μέσω διεργασιών συλλογισμού (reasoning tasks). Αυτές επιτρέπουν στο πράκτορα να συλλογίζεται πρακτικά, να σχηματίζει κοινούς στόχους, ν' αναπτύσσει κοινά πλάνα με άλλους πράκτορες και να προσαρμόζει τη συμπεριφορά του ανάλογα με: (α) τη γνώση που έχει για τη κατάσταση του φυσικού του περιβάλλοντος, (β) τη γνώση που έχει για την νοητική κατάσταση των συνεργατών του, (γ) τη γνώση που έχει για τους τρόπους με τους οποίους μπορεί να πετύχει τους (προσωπικούς ή κοινούς) στόχους του και (δ) τις «αποκρίσεις» (πληροφορίες) που λαμβάνει από το περιβάλλον του.

Η προσαρμογή της συμπεριφοράς εξαρτάται από τον τρόπο με τον οποίο ο πράκτορας χρησιμοποιεί τη γνώση αυτή (μέσω νοητικών ενεργειών) για να πραγματοποιήσει τις διεργασίες συλλογισμού και όχι (όπως συμβαίνει σε άλλες προσεγγίσεις) προκαθορίζοντας τη συμπεριφορά του πράκτορα αντιστοιχίζοντας κάθε μία ενέργεια πεδίου σε

ένα συγκεκριμένο τύπο συμπεριφοράς (αντανακλαστικό, στοχαστικό κτλ.) ανάλογα με τις πιθανές καταστάσεις του περιβάλλοντός του. Αυτό έχει ως αποτέλεσμα ο πράκτορας να δρα πιο αποτελεσματικά αφού η συμπεριφορά του είναι αποτέλεσμα του τρόπου με τον οποίο συλλογίζεται και όχι κάποιας προκαθορισμένης αντιστοίχισης συγκεκριμένων καταστάσεων σε συμπεριφορές.

Το πλαίσιο ανάπτυξης πρακτόρων ICAGENT μέσω του μηχανισμού ελέγχου της BDI-πρακτικής συλλογιστικής (BDI-Control of Practical Reasoning) που διαθέτει δίνει τη δυνατότητα χειρισμού των πλάνων τόσο των ενεργειών εφαρμογής (domain actions) όσο και των νοητικών ενεργειών (mental actions) που υλοποιούν τις διεργασίες συλλογισμού, με ενιαίο τρόπο. Αποτέλεσμα αυτού του «ενιαίου» τρόπου χειρισμού των πλάνων είναι ότι ο πράκτορας μπορεί να αλλάζει τη συμπεριφορά του δυναμικά ανάλογα με τις ανάγκες του (δηλαδή ανάλογα με τους στόχους του και τους περιορισμούς που θέτουν οι στόχοι αυτοί) και τους περιορισμούς που θέτει το περιβάλλον του, αντιδρώντας έτσι άλλοτε γρήγορα χωρίς να σχεδιάζει προσεκτικά τα πλάνα του και άλλοτε προσεχτικά ελέγχοντας τις προϋποθέσεις και τα αποτελέσματα των ενεργειών των πλάνων που αναπτύσσει.

Στη περίπτωση που ο πράκτορας αναγνωρίσει την ανάγκη συνεργασίας τότε το πλαίσιο εργασίας ICAGENT δίνει τη δυνατότητα να σχεδιάσει και να υλοποιήσει τα πλάνα του σε συνεργασία με άλλους πράκτορες (social deliberation). Κατά τη διάρκεια της συνεργασίας ο πράκτορας μπορεί να σχηματίζει κοινούς στόχους και να διαπραγματεύεται για το ποιοι πράκτορες θα αποτελέσουν την ομάδα συνεργασίας, τις μεθόδους που από κοινού θα ακολουθήσουν προκειμένου να πετύχουν τους κοινούς τους στόχους, καθώς και τον τρόπο με τον οποίο θα μοιραστούν οι εργασίες των μεθόδων μεταξύ τους.

Σε ότι αφορά τη συνεργασία πράκτορα-χρήστη το πλαίσιο εργασίας ICAGENT δίνει τη δυνατότητα ανάπτυξης πρακτόρων διεπαφής ικανών να αντιλαμβάνονται τις ενέργειες του χρήστη και να τις ερμηνεύουν κάτω από πιθανά πλαίσια ενεργειών αναγνωρίζοντας τόσο τους πρωταρχικούς στόχους του χρήστη όσο και τους υπό-στόχους του. Η δυνατότητα ερμηνείας των ενεργειών του χρήστη υπό το πλαίσιο ενεργειών ενός (πιθανού ή καθορισμένου κοινού) πρωταρχικού στόχου σε συνδυασμό με τη δυνατότητα εντοπισμού και επίλυσης συγκρούσεων καθιστούν το πράκτορα ικανό να εντοπίζει τις



λανθασμένες ενέργειες του χρήστη και να τον βοηθά όποτε χρειάζεται αρχίζοντας συνεργασία μαζί του.

Η βοήθεια του πράκτορα διεπαφής προς το χρήστη πραγματοποιείται μέσω διαλόγων (που αναπαριστώνται με χρήση πλαισίων κειμένου) με τους οποίους ο πράκτορας παρουσιάζει αρχικά τις υποθέσεις του για τους πιθανούς πρωταρχικούς στόχους του χρήστη και μετέπειτα, κατά τη διάρκεια της συνεργασίας, για να μοιράζεται με το χρήστη το κοινό πλαίσιο ενεργειών το οποίο περιλαμβάνει τόσο τις τρέχουσες προθέσεις του χρήστη και του πράκτορα (intentional state) όσο τους λόγους για τους οποίους αυτοί πραγματοποιούνται (attentional state). Με το πλαίσιο διαλόγου αυτό αναπτύσσονται σχέσεις εμπιστοσύνης μεταξύ χρήστη – πράκτορα διότι από τη πλευρά του πράκτορα δίνει τη δυνατότητα να παρουσιάζει/εξηγεί στο χρήστη ποιες ενέργειες προσπαθεί να πραγματοποιήσει καθώς και για ποιο λόγο, ενώ από τη πλευρά του ο χρήστης νιώθει ότι έχει συνεχώς τον έλεγχο της εφαρμογής (και του πράκτορα) δίνοντας του τη δυνατότητα να καθορίζει μόνος του τους στόχους του και να τους διευκρινίζει/καθορίζει στο πράκτορα όποτε χρειάζεται βοήθεια.

### 7.1. Μελλοντική εργασία

Η μελλοντική εργασία στην έρευνα μας αφορά τη περαιτέρω ανάπτυξη της προτεινόμενης αρχιτεκτονικής για τη καλύτερη υποστήριξη των ικανοτήτων του πράκτορα τόσο στο να προσαρμόζει αποτελεσματικότερα τη συμπεριφορά του σε δυναμικά περιβάλλοντα όσο και στο να συνεργάζεται καλύτερα με άλλους πράκτορες καθώς και με τον άνθρωπο.

Σε ότι αφορά τις ικανότητες του πράκτορα στο να προσαρμόζει αποτελεσματικότερα τη συμπεριφορά του σε δυναμικά περιβάλλοντα η έρευνά μας μπορεί να επεκταθεί στη μελέτη και ενσωμάτωση μεθόδων για τις νοητικές ενέργειες οι οποίες ελέγχουν-υλοποιούν τις διεργασίες συλλογισμού του πράκτορα και δίνουν σ' αυτόν την ικανότητα μετά-συλλογισμού (meta-reasoning). Συγκεκριμένα, ο πράκτορας για να ελέγχει αποτελεσματικότερα τη συμπεριφορά του θα πρέπει να διαθέτει: (α) Μεθόδους (recipes) για τις νοητικές ενέργειες οι οποίες καθορίζουν τις προδιαγραφές των αισθητήρων του (<sensor>). Οι μέθοδοι αυτοί δίνουν στο πράκτορα τη δυνατότητα να σχεδιάζει πλάνα (perception planning) με τα οποία ελέγχει τους αισθητήρες του εστιάζοντας έτσι τη

προσοχή του μόνο σε γεγονότα και συμβάντα που σχετίζονται με τους στόχους του και τις ανάγκες του. (β) Νοητικές ενέργειες και μεθόδους υλοποίησης αυτών για το προσδιορισμό των καταστάσεων (<situation>) στις οποίες πρέπει να παίρνει τη πρωτοβουλία να δράσει. (γ) Περαιτέρω μελέτη και ανάπτυξη των μεθόδων για τις νοητικές ενέργειες <detect\_conflict> και <resolve\_conflict> έτσι ώστε ο πράκτορας να μπορεί να εντοπίζει και επιλύει συγκρούσεις σε συνεργασία με άλλους πράκτορες. (δ) Μεθόδους για τις νοητικές ενέργειες <select\_recipe> και <elaborate\_recipe> έτσι ώστε να μπορεί να επιλέγει σχετικές μεθόδους και ν' αναπτύσσει τα πλάνα του ανάλογα με τους πόρους που διαθέτει.

Σε ότι αφορά τη συνεργασία του πράκτορα με άλλους πράκτορες, περαιτέρω έρευνα και ανάπτυξη μπορεί να υπάρξει για τις μεθόδους των νοητικών ενεργειών με τις οποίες ο πράκτορας επιλέγει τους συνεργάτες του (δηλαδή, σχηματίζει την ομάδα συνεργασίας), διαπραγματεύεται τις μεθόδους με τους άλλους πράκτορες της ομάδας και μοιράζεται τις υπό-ενέργειες των κοινών ενεργειών. Ιδιαίτερα μας ενδιαφέρει η χρήση του πλαισίου εργασίας ICAGENT για την ανάπτυξη πρακτόρων ικανών να δρουν σε οργανισμούς πρακτόρων [Partsakoulakis, Kourakos and Vouros, 2004]. Περαιτέρω έρευνα και ανάπτυξη χρειάζεται η χρήση των μεθόδων που παρέχει το πλαίσιο εργασίας ICAGENT για τη δημιουργία ρόλων οι οποίοι καθορίζουν τη συμπεριφορά του πράκτορα στον οργανισμό.

Τέλος, σε ότι αφορά την ανάπτυξη πρακτόρων διεπαφής, περαιτέρω έρευνα μπορεί να υπάρξει στη χρήση των μεθόδων που παρέχει το πλαίσιο εργασίας ICAGENT για την ανάπτυξη εκπαιδευτικών στρατηγικών καθώς και στη προσαρμογή της συμπεριφοράς του πράκτορα ανάλογα με το προφίλ του χρήστη.

# Κεφάλαιο 8ο:

## Παραρτήματα

### Παράρτημα Α΄

#### Περιγραφή κατευθύνσεων συμπεριφοράς

Κατεύθυνση	Περιγραφή
react/0	Κατευθύνει τον πράκτορα να αγνοήσει οποιαδήποτε άλλη κατεύθυνση συμπεριφοράς που βρίσκεται στη λίστα κατευθύνσεων συμπεριφοράς μετά από αυτή την κατεύθυνση.
precond/1	Κατευθύνει τον πράκτορα να ελέγξει τη νοητική συνθήκη εφαρμογής της μεθόδου (applicability mental condition). Επίσης ελέγχει τις νοητικές συνθήκες εφαρμογής των γονικών ενεργειών που έχουν χαρακτηριστεί ως περιορισμοί πλαισίου ενεργειών (context constraints) (βλ. επόμενη κατεύθυνση συμπεριφοράς).
cconstr/1	Κατευθύνει τον πράκτορα να θεωρήσει τη νοητική συνθήκη εφαρμογής της μεθόδου ως περιορισμό του πλαισίου ενεργειών αυτής (context constraints). Η νοητική συνθήκη αυτή θα πρέπει να διατηρείται όσο ο πράκτορας πραγματοποιεί την ενέργεια αυτή.
stop_check_cconstr/0	Κατευθύνει τον πράκτορα να σταματήσει τον έλεγχο των νοητικών συνθηκών εφαρμογής των μεθόδων των γονικών ενεργειών (δηλαδή να αγνοήσει τη κατεύθυνση cconstr/1 όλων των γονικών ενεργειών).
goal/1	Κατευθύνει τον πράκτορα να ελέγξει αν ο στόχος της μεθόδου έχει επιτευχθεί μετά την πραγματοποίησή της μεθόδου.
gcontext/1	Κατευθύνει τον πράκτορα να ελέγξει αν ο στόχος της μεθόδου έχει επιτευχθεί κατά τη διάρκεια εκτέλεσης της ενέργειας.
stop_check_gcontext/1	Κατευθύνει τον πράκτορα να σταματήσει τον έλεγχο επιτυχίας των στόχων των γονικών ενεργειών.
effects_goals/1	Κατευθύνει το πράκτορα να ελέγξει για συγκρούσεις μεταξύ των αποτελεσμάτων (effects) της μεθόδου και των στόχων των μεθόδων των γονικών ενεργειών στο τρέχον πλαίσιο ενεργειών.
effects_future mentalcondition/1	Κατευθύνει τον πράκτορα να ελέγξει για συγκρούσεις μεταξύ των αποτελεσμάτων της μεθόδου και των νοητικών συνθηκών εφαρμογής των μελλοντικών ενεργειών στο τρέχον πλαίσιο ενεργειών.
not_collaborate/0	Κατευθύνει τον πράκτορα να μην ελέγξει τις νοητικές συνθήκες που καθορίζουν τη συνεργατική συμπεριφορά του πράκτορα.
collab_cconstr/0	Κατευθύνει τον πράκτορα να ελέγξει τις νοητικές συνθήκες που καθορίζουν τη συνεργατική συμπεριφορά του πράκτορα καθ' όλη τη διάρκεια εκτέλεσης της ενέργειας. Οποτεδήποτε αυτές οι συνθήκες ισχύουν, ο πράκτορας αρχίζει τη συνεργασία για τη συγκεκριμένη ενέργεια σχηματίζοντας τη δέσμευση ότι η

	αντίστοιχη ενέργεια θα πραγματοποιηθεί σε συνεργασία με άλλους πράκτορες.
stop_check_collab_cconst/0	Κατευθύνει τον πράκτορα να αγνοήσει τις νοητικές συνθήκες συνεργασίας των μεθόδων όλων των γονικών ενεργειών.
interact/0	Κατευθύνει τον πράκτορα ν' αγνοήσει όλες τις νοητικές συνθήκες των γονικών ενεργειών που εκφράζουν περιορισμούς του πλαισίου ενεργειών και να ελέγξει μόνο τις νοητικές συνθήκες συνεργασίας αυτών.

Πίνακας 4: Κατευθύνσεις συμπεριφοράς εύρεσης συγκρούσεων και ελέγχου της δυνατότητας συνεργασίας.

Κατεύθυνση	Περιγραφή
resolve	Κατευθύνει τον πράκτορα να επιλύσει τη σύγκρουση κάνοντας χρήση γνώσης για το πεδίο εφαρμογής.
override	Κατευθύνει τον πράκτορα ν' αγνοήσει τη σύγκρουση που έχει εντοπίσει και να συνεχίσει την ανάπτυξη του πλάνου της ενέργειας.
reconsider	Κατευθύνει τον πράκτορα να βρει μία εναλλακτική μέθοδο για την ενέργεια. Αν δεν μπορεί να βρει άλλη σχετική μέθοδο τότε εφαρμόζει την επόμενη κατεύθυνση επίλυσης σύγκρουσης. Αν δεν υπάρχει άλλη κατεύθυνση τότε εφαρμόζει (by default) τη κατεύθυνση <code>backtrack</code> .
repeat	Κατευθύνει τον πράκτορα να επαναεπιλέξει τη σχετική μέθοδο που έχει ήδη επιλέξει ή κάποια άλλη (αν αυτή δεν είναι σχετική) για την ενέργεια. Αν δεν υπάρχει άλλη κατεύθυνση τότε εφαρμόζει (by default) τη κατεύθυνση <code>backtrack</code> .
backtrack	Κατευθύνει τον πράκτορα να απορρίψει τη σχετική μέθοδο που έχει επιλέξει για τη γονική ενέργεια και να επιλέξει μία άλλη. Αν δεν βρει άλλη τότε κάνει πάλι <code>backtrack</code> για τη γονική της γονικής ενέργειας.
drop_plan	Κατευθύνει τον πράκτορα ν' απορρίψει το όλο πλάνο (δηλαδή το πλαίσιο ενεργειών) που η ενέργεια ανήκει.
postpone	Κατευθύνει τον πράκτορα να παύσει την εκτέλεση της μεθόδου της ενέργειας μέχρι η σύγκρουση να επιλυθεί.
cancel	Κατευθύνει τον πράκτορα ν' ακυρώσει την ανάπτυξη ή εκτέλεση της μεθόδου της ενέργειας και να προχωρήσει στην επόμενη ενέργεια του πλαισίου ενεργειών της γονικής ενέργειας.
collaborate	Κατευθύνει τον πράκτορα να συνεργαστεί με άλλους πράκτορες προκειμένου να πραγματοποιήσει την ενέργεια συνεργατικά με άλλους πράκτορες (η κατεύθυνση αυτή χρησιμοποιείται όταν ο πράκτορας πιστεύει ότι ενεργώντας συνεργατικά μπορεί να επιλύσει την σύγκρουση).
drop_collaboration	Στη περίπτωση που ο πράκτορας έχει αποφασίσει να πραγματοποιήσει την ενέργεια μαζί με άλλους πράκτορες και αποτύχει να αναπτύξει το πλάνο αυτής τότε ο πράκτορας σταματάει τη συνεργασία με τους άλλους πράκτορες και προσπαθεί να πραγματοποιήσει την ενέργεια μόνος του.

Πίνακας 5: Κατευθύνσεις συμπεριφοράς επίλυσης συγκρούσεων.

## Παράρτημα Β'

### Μέθοδοι υλοποίησης νοητικών ενεργειών

<pre> rec_lib( /*Mental Action*/elaborate_recipe(Action), /*Recipe_Goal*/ null, /*RecipeId*/    elaborate_recipe1, /*Relevance_MentalCondition*/   not bel(int_that(Agent,do(_,Action,_))), /*Behaviour Directives*/ [react], /*Collaborate_MentalCondition*/ false, /*Task_Allocation*/ [], /*Applicability_MentalCondition*/ true, /*ActionList*/ [], /*Effects*/    _Effects ). </pre>	<pre> rec_lib( /*Reasoning Task*/elaborate_recipe(Action), /*Recipe_Goal*/ null, /*RecipeId*/    elaborate_recipe2, /*Relevance_MentalCondition*/   bel(int_that(Agent, do(Group,     Action, GrRecipe))), /*Behaviour Directives*/ [react], /*Collaborate_MentalCondition*/ false, /*Task_Allocation*/ [], /*Applicability_MentalCondition*/true, /*ActionList*/ [   group_formation(_, Action, Group),   recipe_negotiation(_,Action,Group,Recipe),   task_allocation(_, Action, Group, Recipe) ], /*Effects*/ [   int_that(Agent,do(Group,Action,GrRecipe)) ] ). </pre>
(a) The recipe for the mental action <elaborate_recipe> when the agent has not committed to perform the Action jointly with other agents.	(b) The recipe for the mental action <elaborate_recipe> when the agent has committed to perform the Action jointly with other agents.

Πίνακας 6: Μέθοδοι για τη νοητική ενέργεια <elaborate\_recipe>.

<pre> rec_lib( /*Mental Action*/select_recipe(Action), /*Recipe_Goal*/ null, /*RecipeId*/    select_recipe1, /*Relevance_MentalCondition*/   not bel(int_that(Agent,do(_,Action,_))), /*Behaviour Directives*/ [react], /*Collaborate_MentalCondition*/ false, /*Task_Allocation*/ [], /*Applicability_MentalCondition*/ true, /*ActionList*/ [], /*Effects*/    [] ). </pre>	<pre> rec_lib( /*Reasoning Task*/ select_recipe(Action), /*Recipe_Goal*/ null, /*RecipeId*/    select_recipe2, /*Relevance_MentalCondition*/   bel(int_that(Agent, do(Group,     Action, _))), /*Behaviour Directives*/ [react], /*Collaborate_MentalCondition*/ false, /*Task_Allocation*/ [], /*Applicability_MentalCondition*/true, /*ActionList*/ [   request_recipes(_,Action, Group, Recipes),   select_recipe(_,Action, Recipes), ], /*Effects*/ [] ). </pre>
(a) The recipe for the mental action <select_recipe> when the agent has not committed to perform the Action jointly with other agents.	(b) The recipe for the mental action <select_recipe> when the agent has committed to perform the Action jointly with other agents.

Πίνακας 7: Μέθοδοι για τη νοητική ενέργεια <elaborate\_recipe>.

## Παράρτημα Γ'

### Παραδείγματα μεθόδων υλοποίησης ενεργειών για το περιβάλλον του Tileworld

<pre> rec_lib( /*Domain Action*/ fill_hole(Hole, Tile), /*Recipe Goal*/   not empty(Hole), /*Recipe Id*/     fill_hole,  /*Relevance Mental Condition*/   bel(exist(empty(Hole)))   and   bel(exist(Tile)),  /*Behaviour Directives*/   cconstr([reconsider]) and   collab_cconstr,  /*Collaborate Mental Condition*/   bel(exist(agents(Group)))   and   bel(can_help(Group,SubGroup1,ActId2))   and   bel(can_help(Group,SubGroup2,ActId3)),  /*Task Allocation*/   [ do([VarAgent], ActId1),     do([agent SubGroup1], ActId2),     do([agent SubGroup2], ActId3) ],  /*Applicability Mental Condition*/   bel(exist(empty(Hole)))   and   bel(exist(Tile))   and   bel(exist(path(agent, Tile, Hole))),  /*Action List*/   [ find_path(ActId1, agent, Tile, Hole),     get_tile(ActId2, Tile),     drop_tile(ActId3,Tile,Hole) ],  /*Effects*/       [not empty(Hole)] ). </pre>	<pre> rec_lib( /*Domain Action*/ fill_hole(Hole, Tile), /*Recipe Goal*/   not empty(Hole), /*Recipe Id*/     fill_hole_idle,  /*Relevance Mental Condition*/   ( not bel(exist(empty(Hole)))     then       BehaviourDirectives=goal([drop])   )   or   ( not bel(exist(Tile))     then       BehaviourDirectives=goal([repeat])   )   or   ( not bel(exist(path(agent, Tile, Hole)))     then       BehaviourDirectives=goal([repeat])   ),  /*Behaviour Directives*/   BehaviourDirectives,  /*Collaborate Mental Condition*/   false,  /*Task Allocation*/ [],  /*Applicability Mental Condition*/   true,  /*Action List*/     [],  /*Effects*/         [] ). </pre>
(a) The recipe for the domain task <fill_hole> when the agent has recognized any empty hole and a tile.	(b) The recipe for the domain task <fill_hole> when the agent has not recognized any empty hole.

Πίνακας 8: Μέθοδοι για την ενέργεια <fill\_hole> του πεδίου εφαρμογής Tileworld.

<pre> rec_lib( /*Domain Action*/ get_tile(Tile), /*Recipe Goal*/   hold(Tile), /*Recipe Id*/     get_the_tile1,  /*Relevance Mental Condition*/ not bel(int_that(Agent, do(Group, get_tile(Id,Tile), _Recipe))),  /*Behaviour Directives*/ [cconstr([reconsider])],  /*Collaborate Mental Condition*/ false,  /*Task Allocation*/ [],  /*Applicability Mental Condition*/ not bel(int_that(Agent, do(Group, get_tile(Id,Tile), _Recipe))),  /*Action List*/ [ move_to(ActId1, Tile),   pick_tile(ActId2, Tile) ],  /*Effects*/      [] ). </pre>	<pre> rec_lib( /*Domain Action*/ get_tile(Tile), /*Recipe Goal*/   hold(Tile), /*Recipe Id*/     get_the_tile1,  /*Relevance Mental Condition*/ bel(int_that(Agent, do(Group, get_tile(Id, Tile), _Recipe))),  /*Behaviour Directives*/ [],  /*Collaborate Mental Condition*/ bel(exist(obstacle(Path, Obstacle))) and bel(allocated_tasks(Id, AllocatedTasks)),  /*Task Allocation*/ AllocatedTasks,  /*Applicability Mental Condition*/ true,  /*Action List*/ [ par( [ move_to(ActId1, Tile),         pick_tile(ActId2, Tile)],       [ move_to(ActId3, Tile),         push(ActId4,Tile) ]     ) ],  /*Effects*/      [] ). </pre>
(a) The recipe for the domain task <get_tile> when the agent has decided to perform it individually.	(b) The recipe for the domain task <get_tile> when the agent performs it jointly with another agent.

Πίνακας 9: Μέθοδοι για την ενέργεια <get\_tile> του πεδίου εφαρμογής Tileworld.



<pre> rec_lib( /*Domain Action*/ move_to(Tile), /*Recipe Goal*/  agent_on(Tile), /*RecipeId*/  move_to_tile_by_one_square,  /*Relevance Mental Condition*/ bel(path(agent, Tile, Hole, Path)) and %case1: ( not bel(detect_new(Agent)) AND not bel(detect_new(Obstacle, Path)) then BDirectives=[react([]), goal([repeat])] ) or %case2: ( bel(detect_new(Obstacle, Path)) then BDirectives=[ precondition([reconsider]), goal([repeat]) ] ) or %case3: ( bel(detect_new(Agent)) then BDirectives=[ stop_check_cconstr, goal([repeat]) ] ),  /*Behaviour Directives*/ BDirectives,  /*Collaborate Mental Conditions*/ false,  /*Task Allocation*/ [],  /*Applicability Mental Conditions*/ bel(can_move(Obstacle, Path)),  /*Action List*/  [], /*Effects*/      [] ). </pre>	<pre> rec_lib( /*Domain Action*/ move_to(Tile), /*Recipe Goal*/  agent_on(Tile), /*RecipeId*/ obstacle_in_neighbour_square,  /*Relevance Mental Condition*/ bel(path(agent, Tile, Hole, Path)) and bel(exist(obstacle(Path, Obstacle))) and not bel(can_move(Obstacle, Path)),  /*Behaviour Directives*/ [goal([reconsider])],  /*Collaborate Mental Conditions*/ false,  /*Task Allocation*/ [],  /*Applicability Mental Conditions*/ null,  /*Action List*/ [find_path(_, agent, Tile, Hole)],  /*Effects*/      [] ). </pre>
(a) The recipe for the domain task <move_to> when the agent has not detected any obstacle or when the obstacle is not heavy for it.	(b) The recipe for the domain task <move_to> when the agent has detected an obstacle that cannot move.

Πίνακας 10: Μέθοδοι για την ενέργεια &lt;move\_tile&gt; του πεδίου εφαρμογής Tileworld.

## Παράρτημα Δ'

### Παραδείγματα μεθόδων υλοποίησης ενεργειών για το περιβάλλον διεπαφής της εφαρμογής Δημιουργός Μοντέλων

<pre> rec( /*Domain Task*/ relate_entities(Id,Relation,Entity1,Entity2),  /*Recipe Goal*/ related(Relation, Entity1, Entity2),  /*RecipeId*/ relate_two_entities1,  /*Relevance Mental Condition*/ RelationLibrary=obj(Id1,relation_library,Name) and ModelingSpace = obj(Id2, modeling_space,                     modeling_space),  /*Behaviour Directives*/ [cconstr([reconsider,resolve]),  /*Collaborate Mental Condition*/ true,  /*Task Allocation*/ [ do(Group, ActId1),   do([user], ActId2),   do(Group, ActId3),   do(Group, ActId4) ],  /*Applicability Mental Condition*/ bel(number_of_entities_on_mspace(EntitiesNo)) and EntitiesNo&gt;=2 and bel(can_be_related(Relation,Entity1,Entity2)) and not bel(overlapped(ModelingSpace)),  /*ActionList*/ [ specify_entities(ActId1, Entity1, Entity2),   select(ActId2,RelationLibrary),   drag_and_drop(ActId3, Relation,                 RelationLibrary, modeling_space),   link_relation_to_entities(ActId4, Relation,                             Entity1, Entity2) ], /*Effects*/ [] ). </pre>	<pre> rec( /*Domain Task*/ specify_entities(Id, Entity1, Entity2),  /*RecipeGoal*/ specified_entities( Entity1, Attribute1,                     Entity2, Attribute2),  /*RecipeId*/ specify_related_entities1,  /*Relevance Mental Condition*/ not bel(related(Relation, Entity1, Entity2)),  /*Behaviour Directives*/ [cconstr([ reconsider,resolve,           backtrack])],  /*Collaborate Mental Condition*/ true,  /*Task Allocation*/ [ do([user], ActId1),   do([user], ActId2),   do([user], ActId3),   do([user], ActId4) ],  /*Applicability Mental Conditions*/ not bel(overlapped(Entity1)) and not bel(overlapped(Entity2)),  /*ActionList*/ [ select_entity(ActId1, Entity1 ),   select_attribute(ActId2, Entity1, Attribute1),    select_entity(Actid3, Entity2 ),   select_attribute(ActId4, Entity2,Attribute2) ],  /*Effects*/ [specified(Entity1),              specified(Entity2)] ). </pre>
(a) Recipe for the domain task relate_entities.	(b) Recipe for the domain task specify_entities.

Πίνακας 11: Μέθοδοι για τις ενέργειες <relate\_entities> και <specify\_entities>.

<pre> rec( /*Domain Task*/ link_relation_to_entities(Id, Relation,                         Entity1, Entity2),  /*Recipe Goal*/ linked_entities(Relation, Entity1, Entity2),  /*Recipe_Id*/ link_a_relation_with_entities,  /*Relevance Mental Condition*/ bel(on_mspace(Relation)) and bel(unrelated(Relation)),  /*Behaviour Directives*/ [cconstr([reconsider,resolve,backtrack] )],  /*Collaborate Mental Condition*/ false,  /*Task Allocation*/ [ do([user], ActId1),   do([user], ActId2) ],  /*Applicability Mental Conditions*/ ( bel(var(Entity1)) or   bel(obj(_, Entity1, attribute(specified,                                 Name1, Value1))) ) and ( bel(var(Entity2)) or   bel(obj(_, Entity2, attribute(specified,                                 Name2, Value2))) ),  /*ActionList*/ [ click_on(ActId1, obj(_, Entity1, attribute(_,  Name1, Value1))),    click_on(ActId2, obj(_, Entity2, attribute(_,  Name2, Value2))) ],  /*Effects*/ [] ). </pre>	<pre> rec( /*Domain Task*/ select(Id, Object),  /*Recipe Goal*/ clicked_on(Object),  /*Recipe_Id*/ click_on_object,  /*Relevance Mental Condition*/ not bel(var(Object)),  /*Behaviour Directives*/ [],  /*Collaborate Mental Condition*/ false,  /*Task Allocation*/ [],  /*Applicability Mental Conditions*/ true,  /*ActionList*/ [],  /*Effects*/ [selected(Object)] ). </pre>
(a) Recipe for the domain task link_relation_to_entities.	(b) Recipe for the domain task select(Entity) from an entity library.

Πίνακας 12: Μέθοδοι για τις ενέργειες <link\_relation\_to\_entities> και <select\_entity>.

## Παράρτημα Ε΄

### Παραδείγματα προδιαγραφών αισθητήρων για το πράκτορα διεπαφής της εφαρμογής Δημιουργός Μοντέλων

<pre> sensor( /*Sensor Id*/   sensor1,  /*Event type*/   current_user_action(UserAction),  /*Time Period*/   0.01,  /*Learn Mental Condition*/   bel(cpg(Action))   or   bel(do(shared,Action))   and  bel(event_detector(event_occur(UserAction)))   and   bel(contributes_to(UserAction, Action)),  /*Forget Mental Condition*/   bel(cpg(_))   or   bel(do(shared,_))   and   event_detector(event_occur(     CurrentUserAction))    and   bel(current_user_action(UserAction))   and   bel(CurrentUserAction \= UserAction) ). </pre>	<pre> sensor( /*Sensor Id*/   sensor2,  /*Event Type*/   unrecognized(current_user_action(UserAction)),  /*Time Period*/   0.01,  /*Learn Mental Condition*/   bel(do(shared, Action))   and   bel(event_detector(event_occur(UserAction)))   and   not bel(contributes_to(UserAction, Action))   and   bel(unrecognized_actions(Action, Counter))   and   bel(Counter&gt;3),  /*Forget Mental Condition*/   bel(cpg(Action))   or   bel(do(shared,Action))   and   bel(event_detector(event_occur(UserAction)))   and   bel(contributes_to(UserAction, Action)) ). </pre>
<p>(a) A Sensor that enables the agent to perceive the performance of a basic level action that contributes to a candidate or shared action.</p>	<p>(b) A Sensor that enables the agent to perceive the user's actions that do not contribute to the current shared goal.</p>

Πίνακας 13: Προδιαγραφές αισθητήρων.

## Παράρτημα ΣΤ΄

### BNF–Περιγραφή των νοητικών δομών και μεθόδων του ICAGENT

Νοητική δομή	BNF Περιγραφή	Κεφάλαιο
Πεποίθηση	<pre>&lt;bel&gt; ::= bel(Agent, State, C) &lt;Agent&gt; ::= prolog fact &lt;State&gt; ::= prolog fact &lt;C&gt; ::= 0 1</pre>	4.1.1.
Αμοιβαία πεποίθηση	<pre>&lt;mb&gt; ::= mb(Group, State) &lt;Group&gt; ::= prolog list</pre>	4.1.1.
Επιθυμία	<pre>&lt;desire_to&gt; ::= desire_to(Agent, Action) &lt;Action&gt; ::= prolog fact</pre>	4.1.1.
Στόχος	<pre>&lt;goal_to&gt; ::= goal_to(Agent, Action)</pre>	4.1.1.
Πρόθεση	<pre>&lt;int_to&gt; ::= int_to(Agent, Action)</pre>	4.1.1.
Πρόθεση να πετύχει ή να διατηρήσει μία κατάσταση	<pre>&lt;int_that&gt; ::= int_that(Agent, State, IC) &lt;IC&gt; ::= integer</pre>	4.1.1.
Δέσμευση ομάδας για κοινή δράση	<pre>&lt;int_that_do&gt; ::= int_that(Agent, do(Group, Action, Recipe))</pre>	4.1.1, 5.2, 6.2, 6.8.1
Δέσμευση πράκτορα για επιτυχή πραγματοποίηση κοινής ενέργειας	<pre>&lt;int_that_cba&gt; ::= int_that(Agent, cba(Group, Action, Recipe))</pre>	4.1.1, 4.2, 5.3, 5.4, 6.8.2

Πίνακας 14: Νοητικές δομές.



```
<resolve_list> ::= [<resolvedir> | <resolve_list>]
<resolvedir> ::= resolve || override || reconsider || repeat || backtrack ||
drop_plan || postpone || cancel || collaborate || drop_collaboration

<task_allocation> ::= [] || [do(<subgroup>, <action_id>, <cooperation_Type>)
| <task_allocation>]
<subgroup> ::= [<agent_id> | <subgroup>]
<cooperation_type> ::= delegate || collaborate

<action_list> ::= [] || <actions_list>
<actions_list> ::= [<gaction> | <actions_list>]
<gaction> ::= <recipe_action> || <parallel_actions>
<parallel_actions> ::= par(<actions_list>, <actions_list>)
<recipe_action> ::= action_name(<action_id>, ...)

<effects> ::= [<prolog_fact> | <effects>]
```

# Βιβλιογραφία

- [Atkin et. al., 1999] M. Atkin, D. Westbrook and P. Cohen: Capture the Flag: Military Simulation Meets Computer Games. AAAI Spring Symposium on AI and Computer Games, 1999.
- [Atkin et. al., 1998] M. Atkin, D. Westbrook and P. Cohen, G. Jorstad: AFS and HAC. In proceedings of AAAI-98 Workshop of Software Tools for Developing Agents, 1998.
- [Bratman, 1987] M. Bratman: Intentions, Plans, and Practical Reason. Stanford University Press, 1987.
- [Bratman, 1999] M. Bratman: Faces of Intentions. Cambridge University Press, 1999.
- [Brooks, 1986] R. Brooks: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, 2(1): 14-23, 1986.
- [Castelfranchi and Falcone, 1998] C. Castelfranchi and R. Falcone: Principles of Trust for MAS: Cognitive Anatomy, Social Importance, And Quantification. In Proceedings of the Third International Conference on Multi Agent Systems, IEEE, July 1998.
- [Cohen and Levesque, 1991] P. Cohen, H. Levesque: Teamwork. Nous vol. 25, pp. 487-512, 1991.
- [Cohen and Levesque, 1990] P. Cohen and H. Levesque: Intention is choice with commitment. Artificial Intelligence, 42(3), 1990.
- [d'Inverno et. al., 2004] M. d'Inverno, M. Luck, M. Georgeff, D. Kinny, M. Wooldridge: The dMARS Architecture: A Specification of the Distributed Multi-Agent Reasoning System. Journal of Autonomous Agents and Multi-Agent Systems, 9, 5-53, 2004.
- [d'Inverno et. al., 1997] M. d'Inverno, D. Kinny, M. Luck, M. Wooldridge: A formal specification of dMARS. Australian Artificial Intelligence Institute, Technical Note 72, 1997.
- [Davies et. al., 2001] J. Davies, A. Gertner, N. Lesh, C. Rich, J. Rickel, C. Sidner: Incorporating Tutorial Strategies into an Intelligent Assistant. In Proceedings of Intelligent User Interfaces, January 2001.
- [De Rosi, 1999] F. de Rosi, B. de Carolis, S. Pizzutilo: Software documentation with Animated Agents. In 5th ERCIM Workshop on 'User Interfaces For All'. Dagstuhl, 1999.



- [Delisle, 2002] S. Delisle and B. Moulin: User Interfaces and Help Systems: From Helplessness to Intelligent Assistance. In *Artificial Intelligence Review*, Kluwer Academic Publishers, pp117-157, 2002.
- [Dennett, 1989] D. Dennett: *The Intentional Stance*. The MIT Press, 1989.
- [Dimitracopoulou et. al., 1999] A. Dimitracopoulou, V. Komis, P. Apostolopoulos, P. Politis: Design Principles of a New Modeling Environment Supporting Various Types of Reasoning and Interdisciplinary Approaches. In *Proceedings of 9th International Conference of Artificial Intelligence in Education: Open Learning Environments - New Computational Technologies to Support Learning, Exploration and Collaboration*, IOS Press Ohmsha., pp. 109-112, 1999.
- [Eliasmith, 2005] C. Eliasmith (Ed.): *Dictionary of the philosophy*. In: <http://philosophy.uwaterloo.ca/MindDict/>, 2005.
- [Falcone and Castelfranchi, 2000] R. Falcone and C. Castelfranchi. Levels of Delegation and Levels of Adoption as the basis for Adjustable Autonomy. In E. Lamma and P. Mello (Eds): *AI\*IA, LNAI 1792*, Springer-Verlag, pp.273-284, 2000.
- [Fischer, 1995] K. Fischer, J.P. Muller and M. Pisschel. Unifying control in a layered agent architecture. *IJCAI95, Agent Theory, Architecture and Language Workshop 95:240-252*, Montreal, August 1995.
- [Garland et. al., 2003] A. Garland, N. Lesh, C. Rich: Responding to and Recovering from Mistakes during Collaboration. In *Proceedings of IJCAI-03 workshop on Mixed-Initiative Intelligent Systems*, 2003
- [Georgeff and Lansky, 1987] M. Georgeff and A. Lansky: Reactive Reasoning and Planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, Seattle, WA, pp. 677-682, 1987.
- [Georgeff et. al., 1999] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge. The Belief-Desire-Intention Model of Agency. In J. P. Muller, M. Singh, and A. Rao, editors *Intelligent Agents V Springer-Verlag Lecture Notes in AI Volume 1365*, March 1999. (The importance of BDI – from Georgeff JAAMAS)
- [Grosz, 1994] B. Grosz: Collaborative Systems. In: *1994 AAAI Presidential Address*. 2(17), pp. 67 - 85, 1994.

- [Grosz and Kraus, 2000] B. Grosz and S. Kraus: Intention Reconciliation by Collaborative Agents. In: Proceedings of the Fourth International Conference on Multi-Agent Systems, 2000.
- [Grosz and Kraus, 1999] B. Grosz and S. Kraus: The Evolution of SharedPlans. In: Foundations and Theories of Rational Agencies, A. Rao and M. Wooldridge, eds. pp. 227-262, 1999.
- [Grosz and Kraus, 1996] B. Grosz and S. Kraus: Collaborative plans for complex group action. In: Artificial Intelligence, 86(2):269-357, 1996.
- [Grosz and Sidner, 1986] B. Grosz and C. Sidner: Attention, Intentions, and the Structure of Discourse. In: Computational Linguistics, 12(3):175-204, 1986.
- [Hill et. al., 1997] R. Hill, J. Chen, J. Gratch, P. Rosenbloom, M. Tambe: Intelligent agents for the synthetic battlefield: A company of rotary wing aircraft. Innovative Applications of Artificial Intelligence (IAAI-97).
- [Horvitz, 1999] Horvitz, E.: Principles of Mixed-Initiative User Interfaces. In: Computer Human Interaction, pp. 15-20, May, 1999.
- [Hutchins et. al, 1986] E. Hutchins, J. Hollan, D. Norman: Direct Manipulation Interfaces. In D. Norman, S. Draper (Eds): User Centered System Design. Lawrence Erlbaum Associates, 1986.
- [Jones, 1999] R. Jones, J. Laird, P. Nielsen, K. Coulter, P. Kenny, F. Koss: Automated Intelligent Pilots for Combat Flight Simulation. In: AI Magazine, Spring, 1999.
- [Jung, 1999] R. Jung: Layered and Resource-Adapting Agents in the RoboCup Simulation. In: M. Asada and H. Kitano (Eds.): RoboCup-98, LNAI 1604, pp. 207-220, 1999. Springer-Verlag Heidelberg Berlin, 1999.
- [Jung and Fischer, 1999] R. Jung, K. Fischer: A Layered Agent Calculus with Concurrent, Continuous Processes. In: M. P. Singh, A. S. Rao, and M. Wooldridge (eds.): Intelligent Agents IV, LNAI 1365:245-258, Springer-Verlag 1998.
- [Kitano et. al., 1997] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, H. Matsubara.: RoboCup: A challenge problem for AI. AI Magazine, 18 (1), 73-85, 1997.

- [Kourakos-Mavromichalis and Vouros, 2006] V. Kourakos-Mavromichalis and G. Vouros: Building Intelligent Collaborative Interface Agents with the ICAGENT Framework. In: Journal of Autonomous Agents and Multi-Agents Systems, Vol 13, Springer, 2006.
- [Kourakos-Mavromichalis and Vouros, 2006] V. Kourakos-Mavromichalis and G. Vouros: Behaviour Flexibility in Dynamic and Unpredictable Environments: The ICAGENT Approach. In: Proceedings of SETN'06 (full paper), LNAI Volume 3955, Springer, 2006.
- [Partsakoulakis, Kourakos-Mavromichalis and Vouros, 2004] I. Partsakoulakis, V. Kourakos-Mavromichalis and G. Vouros: "Socially Deliberating Agents for Human-Centered Knowledge Management". In Proceedings of International Conference on Systems, Man and Cybernetics, IEEE SMC, Hague, Netherlands, 2004.
- [Kourakos-Mavromichalis and Vouros, 2001] V. Kourakos-Mavromichalis and G. Vouros: Balancing Between Reactivity and Deliberation in the ICAGENT Framework, In: Balancing Between Reactivity and Social Deliberation in Multi-agent Systems, LNAI Volume 2103, Springer-Verlag, 2001.
- [Vouros, Partsakoulakis and Kourakos-Mavromichalis, 2003] G. Vouros, I. Partsakoulakis, V. Kourakos-Mavromichalis: Realizing Human Centered Systems via Socially Deliberating Agents, In: Proceedings of 10th International Conference on Human - Computer Interaction, Crete, Greece, June 2003.
- [Kourakos-Mavromichalis and Vouros, 2001] V. Kourakos-Mavromichalis and G. Vouros: Building Intelligent Collaborative Interface Agents with the ICAGENT Development Framework, In: Proceedings of 8th Panhellenic Conference on Informatics, 2001.
- [Vouros and Kourakos-Mavromichalis, 2001] G. Vouros and V. Kourakos-Mavromichalis: Towards a Generic Framework for Building Intelligent Collaborative Interface Agents, In: ERCIM NEWS, Special Theme: Human Computer Interaction, Number 46, July 2001.
- [Kourakos-Mavromichalis and Vouros, 2000] V. Kourakos-Mavromichalis and G. Vouros: ICAGENT: Balancing between reactivity and deliberation. In: Proceedings of "Balancing Reactivity and Social Deliberation in Multi-Agent Systems" Workshop held in European Conference of Artificial Intelligent (ECAI), Berlin, 2000.
- [Laird, 1987] J. Laird, A. Newell, P. Rosenbloom: SOAR: An Architecture for General Intelligence. In: Artificial Intelligence, 33:1-64, 1987.

- [Lesh, 1999] N. Lesh, C. Rich, C. Sidner: Using Plan Recognition in Human-Computer Collaboration. In Proceedings of the Seventh International Conference on User Modeling, pages 23--32, 1999.
- [Lieberman, 1997] H. Lieberman: Autonomous Interface Agents. In Proceedings of CHI, ACM, (pg. 67-74), 1997.
- [Luck et. al., 2003] M. Luck, P. McBurney, C. Preist: Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent-Based Computing). In: Report of AgentLink, A European Co-ordination Action for Agent-based Computing, 2003.
- [Maes, 1995] P. Maes: Modeling Adaptive Autonomous Agents. In Artificial Life Journal, edited by C. Langton, Vol. 1, No. 1 & 2, pp. 135-162, MIT Press, 1994.
- [Maes, 1994] P. Maes: Agents that Reduce Work and Information Overload. In Communications of the ACM, Vol. 37, No. 7, pp. 31-41, July 1994.
- [Mangina, 2002] E. Mangina: Review of Software Products for Multi-Agent Systems. In Report of AgentLink, A European Co-ordination Action for Agent-based Computing, 2002.
- [McCarthy, 1979] J. McCarthy: Ascribing mental qualities to machines. In: Technical Report Memo 326, Stanford AI Lab, 1979.
- [Müller, 1996] J. Müller: The Design of Intelligent Agents, Lecture Notes in Computer Science; Vol. 1177, Springer, 1996.
- [Müller, 1994] J. Müller and M. Pischel: Integrating Agent Interaction into a Planner-Reactor Architecture., In: M. Klein (ed.), Proc. Of the 13th International Workshop on Distributed Artificial Intelligence , Seattle, WA, USA, July 1994.
- [Müller, 1993] P. Muller and M. Pischel. The Agent Architecture InteRRaP: Concept and Application. Technical Report RR-93-26, DFKI Saarbrucken, 1993.
- [Musliner, 1999] D. Musliner, R. Goldman, M. Pelican and K. Krebsbach. Self-Adaptive Software for Hard Real-Time Environments. In IEEE Intelligent Systems, Vol. 14, No4, July/August, 1999.
- [Negroponte, 1970] N. Negroponte.: The Architecture Machine; Towards a more Human Environment. MIT Press, 1970.
- [Newell, 1990] A. Newell: Unified Theories of Cognition. Cambridge, MA: Harvard University Press, 1990.

- [Nwana, 1996] H. Nwana: Software Agents: An Overview. In Knowledge Engineering Review, Vol. 11, No 3, pp.1-40, Cambridge University Press, 1996.
- [Odell and Parunak, 2002] J. Odell, H. Parunak, M. Fleischer and S. Brueckner: Modeling Agents and their Environment. In proceedings of AOSE 2002, Bologna, Italy, July, 2002.
- [Odell, 2000] J. Odell: Agents: Technology and Usage (Part 1). In Distributed Computing Architecture/E-Business Advisory Service, Executive Report. Cutter Consortium, Vol.3, No.4, 2000.
- [Ortiz and Grosz, 2000] C. Ortiz, B. Grosz: Interpreting Information Requests in Context: A Collaborative Web Interface for Distance Learning. In Journal of Autonomous Agents and Multi-Agent Systems, 5, 429-465, Kluwer Academic Publishers, 2002.
- [Parunak, 2002] J. H. Parunak, S. Brueckner, S. Fleischer and J. Odell: A Design Taxonomy of Multi-agent Interactions. In Proceedings of AOSE 2003, Springer, LNCS 2935, pp. 123-137, 2004.
- [Pynadath, 2003] D. Pynadath, M. Tambe: An automated teamwork infrastructure for heterogeneous software agents and humans. Journal of Autonomous Agents and Multi-Agent Systems, Special Issue on Infrastructure and Requirements for Building Research Grade Multi-Agent Systems, 7:71-100, 2003.
- [Pollack and Ringuette, 1999] M. Pollack, M. Ringuette: Introducing the Tileworld: Experimentally Evaluating Agent Architectures. In: Proceedings of the Eighth National Conference on Artificial Intelligence, 183-189. Menlo Park, Calif.: American Association for Artificial Intelligence, 1990.
- [Pollack and Horty, 1999] M. Pollack and J. Horty: BDI Agents: There's More to Life than Making Plans: Plan Management in Dynamic, Multi-agent Environments. In: AAAI Magazine Vol. 20, No.4, 1999.
- [Ragnemalm, 1996] E. Ragnemalm: Collaborative Dialogue with a Learning Companion as Source of Information on Student Reasoning. In Proceedings of ITS, Montreal, pp. 12-14, Springer-Verlag, 1996.
- [Rao and Georgeff, 1995] A. Rao and M. Georgeff: BDI Agents: From Theory to Practice. In Technical Report 56, Australian Artificial Intelligence Institute, Melbourne, Australia, April 1995.
- [Reed and Long, 1997] C. Reed and D. Long: Collaboration, Cooperation and Dialogue Classification. In: Proceedings of IJCAI-1997.

- [Rich, Sidner and Lesh, 2001] C. Rich, C. Sidner, N. Lesh: COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. In AI Magazine, Special Issue on Intelligent User Interfaces, 2001.
- [Rickel, Lesh et. al., 2001] J. Rickel, N. Lesh, N. Sidner and C. Getner: Using a Model of Collaborative Dialogue to Teach Procedural Tasks. In Proceedings of AIED-2001, Workshop on Tutorial Dialog Systems, 2001.
- [Rickel et. al., 2001] J. Rickel, N. Lesh, N. Sidner and C. Getner: Building a Bridge between Intelligent Tutoring and Collaborative Dialogue Systems. In Proceedings of Tenth International Conference on AI in Education, 2001.
- [Selker, 1994] T. Selker: Coach: A Teaching Agent that Learns. In Communications of the ACM, Vol.37, No. 7, July 1994.
- [Scerri et. al., 2003] P. Scerri, D. Pynadath, L. Johnson, N. Schurr, M. Tambe: A prototype infrastructure for distributed robot-agent-person teams. The Second International Joint Conference on Autonomous Agents and Multiagent Systems, 2003.
- [Shneiderman, 1983] B. Shneiderman: Direct manipulation: A step beyond programming languages, IEEE Computer 16, 8, 57-69. 1983.
- [Shneiderman, 1997] B. Shneiderman: Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. In proceedings of Intelligent User Interfaces (ACM) 1997.
- [Shneiderman and Maes, 1997] B. Shneiderman and P. Maes: Direct Manipulation vs. Interface Agents, excerpts from the debates at UI 97 and CHI 97. In Interactions, ACM, November + December, volume IV.6, pp42-61, 1997.
- [Shoham, 1990] Y. Shoham: Agent Oriented Programming. Technical Report STAN-CS-1335-90. Stanford University, 1990.
- [Sonenberg et. al., 1992] Sonenberg, E., Tidhar, G., Werner, E., Kinny, D., Ljungberg, M., and Rao, A.: Planned team activity. In Artificial Social Systems, Lecture Notes in Artificial Intelligence (LNAI-830). Springer Verlag, 227-256, 1992.
- [Stone and Veloso, 1999] P. Stone, M. Veloso: Task Decomposition and Dynamic Role Assignment for Real-Time Strategic Teamwork. In: Proceedings of ATAL'98, LNAI 1555, pp. 293-308, 1999.
- [Tambe, 1997] M. Tambe: Towards flexible teamwork. In: Journal of Artificial Intelligence Research, 7,83-124, 1997.

- 
- [Turner, 1984] Turner, R.: Logics for Artificial Intelligence. Ellis Horwood Series Artificial Intelligence, 1984.
- [Weiss, 2001] G. Weiss. Cognition, Sociability, and Constraints. In: Balancing Between Reactivity and Social Deliberation in Multi-agent Systems, LNAI Volume 2103, Springer, 2001.
- [Wooldridge and Jennings, 1999] M. Wooldridge and N. Jennings: The Cooperative Problem Solving Process. In Journal of Logic and Computation, 9(4):563-592, 1999.
- [Wooldridge, 1999] M. Wooldridge: Intelligent Agents. In Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence. Editor G. Weiss. MIT Press, 1999.
- [Wooldridge, 2000] M. Wooldridge: Reasoning About Rational Agents. MIT Press, 2000.

# Γλωσσάρι

## A

**Ability** Ικανότητα

**Actuators** Μηχανισμοί δράσης

### Action

**Basic level** Βασική ενέργεια

**Complex level** Σύνθετη ενέργεια

**Multi-agent** Πολύ-πρακτορική Ε-  
νέργεια

**Single-agent or individual** Προσω-  
πική ή ατομική ενέργεια

**Adaptability** Προσαρμοστικότητα

**Advisor-style agent** Συμβουλευτικού  
τύπου πράκτορας

**Attentional state** Σύνολο τρεχόντων  
στόχων

**Applicability conditions** Προϋποθέσεις  
εφαρμογής

**Assistant-style agent** Επικουρικού τύ-  
που πράκτορας

## B

### Behaviour

**Reactive** Αντανακλαστική συμπερι-  
φορά

**Deliberative** Στοχαστική συμπεριφο-  
ρά

**Behavioural** Συμπεριφορικό ή Συμπερι-  
φερσιολογικό

### BDI–Control of Practical Reasoning

#### Mechanism

Μηχανισμός ελέγχου της BDI–  
Πρακτικής Συλλογιστικής

## C

**Capability** Δυνατότητα

**Cognitive** Γνωστικό ή Γνωσιακό

**Cooperation** Συνεργασία – Σύμπραξη

**Collaboration** Συνεργασία

**Constraints Satisfaction** Ικανοποίηση  
περιορισμών

**Context of action** Πλαίσιο ενεργειών

**Context constraints** Περιορισμοί του  
πλαισίου ενεργειών



**Core case commitments** Βασικές προθέσεις

## D

### Deliberation

**Individual** Ατομικός στοχασμός

**Social** Κοινωνικός στοχασμός

**Deliberative agent** Στοχαστικός πράκτορας

**Deliberative layer** Επίπεδο στοχασμού

**Delegation** Ανάθεση εργασιών

**Detectors** αισθητήρες

**facts–events** αισθητήρες εντοπισμού γεγονότων και ενεργειών

**Direct manipulated interfaces** Επιφάνειες διεπαφής άμεσου χειρισμού

**Domain Actions** Ενέργειες εφαρμογής

## E

**Environment monitoring** Ικανότητα παρακολούθησης του περιβάλλοντος

**Effect of action** Αποτέλεσμα ενέργειας

## F

**Focal action** Ενέργεια εστίασης ενδιαφέροντος του πράκτορα ή ενέργεια βασικού ενδιαφέροντος

**Fringe** Περιοχή παρυφής

## G

**Group activity commitment** Δέσμευση ομάδας για κοινή δράση

## I

**ICAGENT framework** (Intelligent Collaborative Agent Framework) Πλαίσιο Ανάπτυξης Έξυπνων Συνεργατικών Πρακτόρων

**Intentional stance** Στάση πρόθεσης

**Interface (of application)** Επιφάνεια διεπαφής (της εφαρμογής)

**Interface agent** Πράκτορας διεπαφής

**Intentional Context** πλαίσιο προθέσεων

## J

**Joint Intentions** Συντονισμένες ενέργειες

**Joint persistent goal** Διαρκείς συντονισμένος στόχος

## K

**Knowledge Base Consult and Update**

**Mechanism** Μηχανισμός ενημέρωσης και εξαγωγής γνώσης

## L

**Layered architectures** Αρχιτεκτονικές

Επιπέδων

## M

**Means-end reasoning** Ανάλυση μέσων και σκοπών

**Mental state view of plan management** Νοητική όψη χειρισμού των πλάνων

**Mutual beliefs** Αμοιβαίες πεποιθήσεις

**Mutual belief satisfaction** Ικανοποίηση κοινών περιορισμών

## N

**Notion o agency**  
**weak** Ασθενής θεώρηση ενός πράκτορα.  
**strong** Ισχυρή θεώρηση ενός πράκτορα.

## O

**Observable Physical Environment**  
 Παρατηρήσιμο φυσικό περιβάλλον

## P

**Parameter identification** Καθορισμός παραμέτρων

**Practical reasoning** Πρακτική συλλογιστική

**Procedural reasoning system** Διαδικασιακό συλλογιστικό σύστημα

**Perception module** Μηχανισμός Αντίληψης ή Μονάδα αντίληψης ή συλλογιστική διαδικασία αντίληψης

**Physical Environment** Φυσικό Περιβάλλον

**Plan elaboration module** Μονάδα ανάπτυξης πλάνων

## R

**Rational behaviour** Ορθολογική Συμπεριφορά

**Reaction** Αντίδραση

**Reactive agent** Πράκτορας ικανός ν' αντιδρά αντανακλαστικά

**Reactive layer** Επίπεδο αντανακλαστικής αντίδρασης

**Real time environments** Περιβάλλοντα πραγματικού χρόνου

**Reasoning task** Διεργασία συλλογισμού

**Recipe** Μέθοδος

**Recipe tree** Δένδρο πραγματοποίησης ενέργειας

**Reconciliation** Αναγνώριση και επίλυση συγκρούσεων

**Resource Bounded** Περιορισμένων πόρων

**S**

**Sensor** Αισθητήρας

**Shared Action** Κοινή ενέργεια

**Shared plans**

**full** Πλήρως ανεπτυγμένα κοινά πλάνα

**Partial** Μερικώς ανεπτυγμένα κοινά πλάνα

**Situation recognition module** Μονάδα αναγνώρισης νοητικών καταστάσεων ή διεργασία συλλογισμού αναγνώρισης καταστάσεων

**Software Agents** Πράκτορες Λογισμικού

**T**

**Theoretical Reasoning** Θεωρητική Συλλογιστική

**Trust** Εμπιστοσύνη

**Tutor Agents** Εκπαιδευτικοί πράκτορες

# Ευρετήριο

— <i>A</i> —	
ability .....	54
actuators .....	3, 74, 90
adapt .....	2, 17, 20
advisor-style agents .....	80
applicability conditions ..	13, 53, 54, 58, 105, 119, 123, 154, 155, 175
assistant-style agents .....	80
attentional state .....	53, 82, 84, 85, 147, 148, 151, 155, 156, 158, 160, 161, 164, 166, 173
— <i>B</i> —	
basic level action .....	53, 105, 122, 128, 159, 184
BDI-Control of Practical Reasoning	32, 90, 172
belief .....	93
— <i>C</i> —	
capability .....	54
cognitive .....	69, 75
collaboration .....	41, 86, 176
complex level action .....	53, 105
Constraints satisfaction .....	58
context constraints .	13, 54, 58, 77, 105, 115, 123, 124, 175
context of action .....	13, 32, 83, 126
cooperation .....	5, 11, 76
core case commitment	58, 97, 119, 120, 127
— <i>D</i> —	
delegation .....	11, 76
deliberative .	7, 20, 36, 38, 48, 118, 119
desire .....	95
detectors .....	90, 152
deterministic .....	5
direct manipulated interfaces .....	24
dynamic .....	5
— <i>E</i> —	
environment .....	4
— <i>F</i> —	
fringe .....	157, 164
full shared plan .....	60
— <i>G</i> —	
goal .....	95
group activity commitment .....	97
Group activity commitment .....	57, 197

- group formation ..... 132
- *I*—
- ICAGENT ..... 2, 90
- individual deliberation ..... 17
- int\_that\_cba ..... 97
- int\_that\_do ..... 97
- intent that ..... 96
- Intention Realization ..... 132
- intentional stance ..... 8
- interface agents ..... 2
- *J*—
- joint intention ..... 70
- Joint Persistent Goal ..... 70
- *K*—
- Knowledge Base Consult and Update  
Mechanism ..... 92, 106, 197
- *L*—
- layered architectures ..... 20
- *M*—
- mental state view of plans ..... 17
- mental-state view of plan management  
..... 22
- mutual belief ..... 94
- *P*—
- Parameter identification ..... 58, 198
- partial shared plan ..... 62
- perception ..... 67, 91, 114, 115, 173
- Perception Module ..... 112
- physical environment ..... 4
- Plan Elaboration Module ..... 126
- practical reasoning ..... 8, 10
- Procedural Reasoning System ..... 63
- *R*—
- reactive ..... 7, 20, 75, 119
- recipe ..... 100
- recipe library ..... 100
- Recipe Negotiation ..... 132
- reconciliation ..... 6, 21, 55, 79, 91, 102
- Reconciliation Module ..... 119
- resource bounded ..... 5
- resource bounded agents ..... 16, 19
- *S*—
- sensor's specification knowledge ..... 113
- SharedPlans* ... 2, 17, 24, 52, 68, 81, 85,  
96
- Situation Recognition Module ..... 116
- social deliberation 11, 18, 138, 140, 172
- stochastic ..... 5
- strong notion of agency ..... 6
- *T*—
- Task Allocation ..... 132
- trust ..... 26, 29
- tutor agents ..... 81
- *W*—
- weak notion of agency ..... 6

