

# ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ



## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Τίτλος:** Αξιολόγηση και βελτίωση της γλώσσας  
μοντελοποίησης επιχειρησιακών διαδικασιών IDEF0

Επιμέλεια εργασίας:  
**ΑΝΤΡΕΑΣ ΓΕΝΤΣΟΣ**

Επιβλέπον καθηγητής:  
Λουκάς Τσιρώνης

ΧΑΝΙΑ  
ΙΟΥΛΙΟΣ 2007



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή της πτυχιακής μου εργασίας κ. Λουκά Τσιρώνη, για την βοήθειά του και την συνεργασία που είχαμε, ιδιαίτερα κατά την περίοδο των καταλήψεων που οι συνθήκες ήταν ιδιαίτερα δύσκολες. Επίσης θα ήθελα να ευχαριστήσω τον κ. Θωμά Κοντογιάννη για την βοήθεια του και την τεχνική υποστήριξη που μου παρείχε.

<b>ΚΕΦΑΛΑΙΟ 1 .....</b>	<b>6</b>
<b>ΕΙΣΑΓΩΓΗ .....</b>	<b>6</b>
1.1 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΘΕΜΑΤΟΣ .....	6
1.2 ΣΚΟΠΟΣ ΕΡΓΑΣΙΑΣ.....	8
1.3 ΔΙΑΡΘΡΩΣΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ .....	9
<b>ΚΕΦΑΛΑΙΟ 2 .....</b>	<b>10</b>
<b>ΕΙΣΑΓΩΓΗ ΣΤΟ BPM.....</b>	<b>10</b>
2.1 ΕΙΣΑΓΩΓΗ .....	10
2.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ .....	10
2.3 ΟΡΙΣΜΟΣ ΤΗΣ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ .....	12
2.4 ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ .....	15
2.5 ΟΡΙΣΜΟΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΜΟΝΤΕΛΟΥ .....	16
2.6 ΕΙΔΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΜΟΝΤΕΛΩΝ .....	17
2.7 ΕΙΔΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ .....	18
<b>ΚΕΦΑΛΑΙΟ 3 .....</b>	<b>20</b>
<b>ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ .....</b>	<b>20</b>
3.1 Η ΓΛΩΣΣΑ IDEF0 .....	20
3.1.1 Η ΠΡΟΣΕΓΓΙΣΗ IDEF0 .....	21
3.1.2 ΣΥΝΤΑΞΗ ΜΟΝΤΕΛΟΥ .....	22
3.1.2.1 Κουτιά .....	23
3.1.2.2 Τόξα .....	23
3.1.2.3 Σημασιολογία .....	24
3.1.2.4 Σημασιολογία κουτιών και βελών .....	24
3.1.3 ΤΑ IDEF0 ΔΙΑΓΡΑΜΜΑΤΑ .....	26
3.1.3.1 Τύποι διαγραμμάτων .....	26
3.1.3.2 Διάγραμμα υψηλότερου επιπέδου .....	26
3.1.3.3 Child diagram .....	27
3.1.3.4 Parent diagram .....	28
3.1.3.5 Κείμενο και γλωσσάριο .....	30
3.1.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΓΡΑΦΗΜΑΤΩΝ .....	31
3.1.4.1 Βέλη σαν προϋποθέσεις .....	31
3.1.4.2 Ενεργοποίηση ενός κουτιού .....	31
3.1.4.3 Τόξα με κλάδους .....	31
3.1.4.4 ICOM κωδικοποίηση boundary τόξων .....	32
3.1.4.5 Tunneled βέλη .....	34
3.1.4.6 Call βέλη .....	35
3.1.5 ΚΑΝΟΝΕΣ ΣΥΝΤΑΞΗΣ .....	36
3.2 WORKFLOW MODELER.....	38
3.3 MICROSOFT VISIO .....	40
3.4 ΟΡΟΛΟΓΙΑ .....	40
<b>ΚΕΦΑΛΑΙΟ 4 .....</b>	<b>41</b>
<b>ΑΠΟΤΕΛΕΣΜΑΤΑ ΒΙΒΛΙΟΓΡΑΦΙΚΗΣ ΕΡΕΥΝΑΣ.....</b>	<b>41</b>
4.1 ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΤΗΣ ΓΛΩΣΣΑΣ IDEF0 .....	41
4.2 ΠΩΣ ΠΡΟΚΥΠΤΕΙ Η ΑΝΑΓΚΗ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΗΣ IDEF0 .....	43
4.3 ΠΡΟΗΓΟΥΜΕΝΕΣ ΠΡΟΣΠΑΘΕΙΕΣ ΒΕΛΤΙΩΣΗΣ .....	43
4.3.1 Χρήση IDEF0, IDEF3 και DFD .....	44
4.3.2 Άλλες μέθοδοι .....	45

4.4 Η ΣΗΜΑΣΙΑ ΤΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ .....	46
<b>ΚΕΦΑΛΑΙΟ 5.....</b>	<b>50</b>
<b>ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....</b>	<b>50</b>
5.1 ΤΡΟΠΟΣ ΥΛΟΠΟΙΗΣΗΣ.....	50
5.2 ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΗΣ IDEF0.....	51
5.3 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΜΟΝΤΕΛΟΥ .....	52
Λεπτομερής περιγραφή του μοντέλου .....	53
5.3.1 Διάγραμμα A-0 .....	53
5.3.2 Διάγραμμα A0.....	54
5.3.3 Διάγραμμα A1.....	56
5.3.4 Διάγραμμα A2.....	58
5.3.5 Διάγραμμα A23.....	59
5.3.6 Διάγραμμα A3.....	60
5.4 ΑΛΛΑΓΕΣ ΣΤΟ ΜΟΝΤΕΛΟ .....	61
Αλλαγές στα διαγράμματα.....	63
5.4.1 Διάγραμμα A-0 .....	63
5.4.2 Διάγραμμα A0.....	64
5.4.3 Διάγραμμα A1.....	65
5.4.4 Διάγραμμα A2.....	66
5.4.5 Διάγραμμα A23.....	68
5.4.6 Διάγραμμα A24.....	69
5.4.7 Διάγραμμα A3.....	70
5.5 ΕΞΗΓΗΣΗ ΜΙΚΡΩΝ ΚΩΔΙΚΩΝ ΚΑΙ ΛΕΙΤΟΥΡΓΙΩΝ .....	71
5.6 ΠΕΡΙΓΡΑΦΗ ΣΦΑΛΜΑΤΩΝ, ΚΑΘΥΣΤΕΡΗΣΕΩΝ ΚΑΙ ΠΑΡΑΛΛΗΛΩΝ ΔΙΑΔΙΚΑΣΙΩΝ ΣΤΑ ΔΙΑΓΡΑΜΜΑΤΑ IDEF0 .....	76
5.6.1 Διάγραμμα A1.....	80
5.6.2 Διάγραμμα A2.....	82
5.6.3 Διάγραμμα A23.....	84
5.6.4 Διάγραμμα A24.....	86
5.6.5 Διάγραμμα A24.....	87
5.7 ΕΛΕΓΧΟΣ ΜΕΣΑ ΣΤΑ ΔΙΑΓΡΑΜΜΑΤΑ ΓΙΑ ΠΙΘΑΝΑ ΣΦΑΛΜΑΤΑ.....	88
Παραδείγματα εφαρμογής .....	90
5.8 ΑΡΙΘΜΟΣ ΠΛΗΡΟΦΟΡΙΩΝ Η ΚΟΜΜΑΤΙΩΝ ΑΝΑ ΒΕΛΟΣ .....	95
5.9 ΣΦΑΛΜΑΤΑ ΠΟΥ ΜΠΟΡΕΙ ΝΑ ΠΡΟΚΥΨΟΥΝ ΣΤΟ ΜΟΝΤΕΛΟ.....	97
<b>ΚΕΦΑΛΑΙΟ 6.....</b>	<b>105</b>
<b>ΣΥΜΠΕΡΑΣΜΑΤΑ ΠΕΡΙΟΡΙΣΜΟΙ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ .....</b>	<b>105</b>
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ .....	105
6.2 ΠΕΡΙΟΡΙΣΜΟΙ .....	108
6.3 ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ .....	109
<b>ΠΑΡΑΡΤΗΜΑ .....</b>	<b>111</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>132</b>

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

### 1.1 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΘΕΜΑΤΟΣ

Η διπλωματική εργασία πραγματεύεται την μοντελοποίηση επιχειρησιακών διαδικασιών (Business Process Modeling, BPM). Η επιχειρηματική μοντελοποίηση είναι αναγκαία διαδικασία για κάθε σύγχρονη εταιρία. Ο όρος “Επιχειρησιακή Διαδικασία” χρησιμοποιείται ευρέως τα τελευταία χρόνια στον επιχειρησιακό κόσμο και όχι μόνο, περιγράφοντας τον τρόπο με τον οποίο κάτι επιχειρείται να γίνει μέσα σε ένα οργανισμό ή μια επιχείρηση. Τα χρόνια αυτά που χρησιμοποιείται ο όρος αυτός έχουν δοθεί αρκετοί ορισμοί για το: “Τι ονομάζουμε επιχειρησιακή διαδικασία;”, από διάφορους ερευνητές. Σύμφωνα με τους Hammer και Champy (1993): *“Η επιχειρησιακή διαδικασία είναι ένα σύνολο από δραστηριότητες οι οποίες είναι αναγκαίες για να μετατραπεί το προϊόν από ακατέργαστη ύλη σε προϊόν που θα προσφέρει αξία στον τελικό καταναλωτή”*.

Γενικά, θα μπορούσαμε να πούμε ότι μία επιχειρησιακή διαδικασία είναι μία σειρά από γεγονότα που αποτελούνται από μία ή περισσότερες συνδεδεμένες δραστηριότητες (εργασίες) οι οποίες είναι απαραίτητες για να παραδοθεί ένα προϊόν ή υπηρεσία με μία αντιληπτή αξία στον πελάτη (Vernadat, 1995).

Στο σημείο αυτό θεωρείται χρήσιμο να αναλύσουμε κάθε μια από τις δυο έννοιες: “Επιχείρηση” και “Διαδικασία”. Έτσι, όσον αφορά από τον πρώτο όρο, “Επιχείρηση”: είναι μια οργανωμένη οντότητα που μετασχηματίζει πόρους με σκοπό να παρέχει στους πελάτες της τα επιθυμητά προϊόντα ή υπηρεσίες. Από την άλλη πλευρά, όσον αφορά την δεύτερη έννοια, ως “Διαδικασία ”: ορίζεται μια ομάδα δραστηριοτήτων οι οποίες μετατρέπουν ένα σύνολο από δεδομένα εισόδου σε ένα σύνολο δεδομένων εξόδου για κάποιο πρόσωπο ή άλλη διαδικασία, χρησιμοποιώντας ανθρώπινο δυναμικό, τεχνολογία και πόρους (Laguna and Marklund, 2005).

Περιγραφόμενες διαδικασίες μπορεί να είναι μια γραμμή παραγωγής, η λειτουργία ενός εργοστασίου, η λειτουργία ενός ή παραπάνω τμημάτων μιας επιχείρησης, η διαδικασία λήψης μιας απόφασης ή ακόμα πιο απλές διαδικασίες όπως η καθημερινή τροφοδοσία μιας οικογένειας.

Η μοντελοποίηση επιχειρησιακών διαδικασιών εμφανίζεται πρώτη φορά το 1866 από τον Henri Fayol ως διοίκηση επιχειρησιακών διαδικασιών και στην συνέχεια από τον πιο διάσημο οικονομολόγο όλων των εποχών, τον Adam Smith (1876). Ακαδημαϊκά ο πρώτος που ασχολήθηκε με τη μοντελοποίηση επιχειρησιακών διαδικασιών ήταν ο Nordsieck (1932 και 1934). Από τότε μέχρι σήμερα έχει γίνει τεράστια πρόοδος και μοντέλα που αναπαριστούν επιχειρησιακές διαδικασίες χρησιμοποιούνται ευρέως στον τομέα της διοίκησης επιχειρήσεων.

Προς την κατεύθυνση αυτή έχουν αναπτυχθεί πολλές γλώσσες μοντελοποίησης επιχειρησιακών διαδικασιών, οι οποίες πλέον δεν στοχεύουν μόνο στην περιγραφή των διαδικασιών αλλά και στην βελτίωσή τους.. Μερικές από αυτές είναι το EPC (εκδόθηκε το 1998), GRAI Integrated Methodology (GIM, εκδόθηκε το 1985), Structured System Analysis Design Methodology (SSADM, εκδόθηκε το 1986), Data Flow Diagrams (DFD, εκδόθηκε το 1979), Structured Systems Analysis (SSA, εκδόθηκε το 1979), Structured Design Analysis Technique (SDT, εκδόθηκε το 1977) και η οικογένεια μεθόδων IDEF που περιλαμβάνει περίπου δεκαπέντε μεθοδολογίες.

Μια από τις πιο σημαντικές γλώσσες μοντελοποίησης επιχειρησιακών διαδικασιών είναι η IDEF0, που εμπεριέχεται στην οικογένεια μεθόδων IDEF. Αυτή είναι και η μέθοδος που θα χρησιμοποιηθεί στην πορεία. Αποτελείται από μια σειρά διαγραμμάτων ροής πληροφοριών, που συνθέτονται από βέλη και κουτιά, με ιεραρχική ανάλυση. Αυτό ουσιαστικά σημαίνει ότι το μοντέλο που προκύπτει, ξεκινάει από ένα γενικό διάγραμμα πρώτου επιπέδου, που περιέχει τις εισόδους και τις εξόδους του μοντέλου και σταδιακά αναλύεται λεπτομερώς από πάνω προς τα κάτω. Δηλαδή το κάθε κουτί κάθε διαγράμματος αναλύεται από ένα νέο διάγραμμα (όταν αυτό απαιτείται). Τα κουτιά αναπαριστούν λειτουργίες και τα βέλη

πληροφορίες ή κομμάτια. Σημαντικό είναι να γίνει κατανοητό ότι τα βέλη δεν συμβολίζουν διαδοχή λειτουργιών αλλά μεταφορά πληροφοριών ή κομματιών.

Όπως κάθε μια από τις μεθόδους που αναφέραμε παραπάνω, έτσι και η IDEF0 έχει κάποια πλεονεκτήματα και κάποια μειονεκτήματα. Υπάρχουν πράγματα που είτε δεν μπορούν να μοντελοποιηθούν μέσω της IDEF0 είτε δεν γίνονται ευδιάκριτα με αποτέλεσμα να δημιουργούνται προβλήματα. Σκοπός είναι η αντιμετώπιση κάποιων από αυτά τα προβλήματα, με εισαγωγή νέων στοιχείων. Κάποια από τα προβλήματα που συναντάμε κατά την χρήση της IDEF0 είναι η αναπαράσταση και η χρήση(μέσω προσομοίωσης) χαμένων χρόνων, η αναπαράσταση και η χρήση των λαθών που μπορεί να γίνουν σε μία επιχειρησιακή διαδικασία, η αποσαφήνιση και αναπαράσταση των παράλληλων διαδικασιών και το πόσα κομμάτια περιέχει κάθε στιγμή ένα βέλος.

## 1.2 ΣΚΟΠΟΣ ΕΡΓΑΣΙΑΣ

Όπως αναφέρθηκε και παραπάνω στόχος των μεθόδων μοντελοποίησης επιχειρησιακών διαδικασιών είναι πλέον, εκτός από την περιγραφή (αναπαράσταση) της διαδικασίας και η βελτίωσή της. Η βελτίωση μια διαδικασίας μπορεί να γίνει εφικτή μέσω της χρήση τεχνικών προσομοίωσης. Δηλαδή μετά την κατασκευή του μοντέλου με μία από τις μεθόδους, αυτό προσομοιώνεται δίνει κάποια αποτελέσματα. Στην συνέχεια με τις εξαγόμενες πληροφορίες γίνονται προσπάθειες βελτίωσης της διαδικασίας. Άρα με γνώμονα την περιγραφή και βελτίωση μιας επιχειρησιακής διαδικασίας, θα κινηθούμε σε δύο άξονες, την γραφική αναπαράσταση και την προσομοίωση. Οι προσπάθειες βελτίωσης που θα γίνουν θα έχουν κάνουν με την βελτίωση της μεθόδου IDEF0 σε επίπεδο γραφικής απεικόνισης και λογισμικής βελτίωσης.

Αξίζει να σημειωθεί ότι η IDEF0 είναι μια στατική μέθοδος. Στατικό λέγεται ένα μοντέλο το οποίο δεν αλλάζει με την πάροδο του χρόνου. Αυτός ακριβώς αυτός είναι και ο λόγος που χρησιμοποιούμε εργαλεία προσομοίωσης σε συνεργασία με την IDEF0. Η IDEF0 παρέχει την αποτελεσματική και αυστηρή μοντελοποίηση μιας διαδικασίας και η προσομοίωση την εισαγωγή χρονικής μεταβολής στο μοντέλο. Για τον σκοπό αυτό χρησιμοποιήθηκε το πακέτο της MetaSoftware, Workflow Modeler



και η έκδοσή του για την χρήση προσομοίωσης Workflow Simulator. Αρχικά η διαδικασία που επεξεργάζεται στην συνέχεια αναπαραστάθηκε στο Workflow Modeler και στην συνέχεια έγινε προσομοίωση του στο Workflow Simulator.

### 1.3 ΔΙΑΡΘΡΩΣΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Στο πρώτο κεφάλαιο υπάρχει μια εισαγωγή στο θέμα που πραγματεύεται η παρούσα εργασία και ο σκοπός της. Στο δεύτερο κεφάλαιο παρέχονται στον αναγνώστη γενικότερες πληροφορίες γύρω από την μοντελοποίηση επιχειρηματικών διαδικασιών. Στο τρίτο κεφάλαιο παραθέτουμε το θεωρητικό υπόβαθρο της εργασίας, που έχει να κάνει κατά κύριο λόγο με την IDEF0 και λιγότερο με το Workflow Modeler και κάποια ορολογία. Ακολουθεί το τέταρτο κεφάλαιο στο οποίο παρουσιάζονται τα αποτελέσματα της βιβλιογραφικής έρευνας, όσο αναφορά την σημαντικότητα της IDEF0, τα πλεονεκτήματά της και τα μειονεκτήματά της. Το πέμπτο κεφάλαιο αποτελεί την κυρίως ανάπτυξη της εργασίας βήμα προς βήμα. Στο έκτο κεφάλαιο αναπτύσσονται τα συμπεράσματα της διπλωματικής εργασίας και οι περιορισμοί που συναντήθηκαν, επίσης γίνονται κάποιες προτάσεις για περαιτέρω έρευνα. Τέλος η εργασία ολοκληρώνεται με το παράρτημα στο οποίο παραθέτουμε όλα τα σχήματα που χρησιμοποιήθηκαν ώστε να είναι πιο ευανάγνωστα και την βιβλιογραφία.

## **ΚΕΦΑΛΑΙΟ 2**

### **ΕΙΣΑΓΩΓΗ ΣΤΟ BPM**

#### **2.1 ΕΙΣΑΓΩΓΗ**

Στο κεφάλαιο αυτό θα προσδιορίσουμε τις έννοιες του επιχειρηματικού μοντέλου και της επιχειρηματικής διαδικασίας. Θα παρουσιαστεί μια ιστορική αναδρομή που έχει να κάνει με την μοντελοποίηση επιχειρησιακών διαδικασιών. Θα δούμε επίσης τα είδη επιχειρηματικής διαδικασίας.

Σκοπός του κεφαλαίου αυτού είναι η εισαγωγή του αναγνώστη στο πρωτογενές θεωρητικό επίπεδο της επιχειρηματικής μοντελοποίησης με την περιγραφή βασικών εννοιών, έτσι ώστε να μπορέσει να κατανοήσει το αντικείμενο της εργασίας αυτής, όσο γίνεται καλύτερα.

#### **2.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ**

Στην ουσία ο όρος Business Process Management, έχει να κάνει με τον συντονισμό των επιχειρησιακών διαδικασιών εντός μιας εταιρίας ή ανάμεσα σε εταιρίες. Κατά συνέπεια μπορεί να σχετιστεί με αναφορές σε οικονομικά θέματα αλλά και σε θέματα διοίκησης επιχειρήσεων. Ένας από τους πρωτοπόρους της θεωρίας μοντέρνας οργάνωσης ο Henri Fayol (1866) συνιστά την υποδιαίρεση καθηκόντων με σκοπό την αύξηση της παραγωγικότητας. Ο Adam Smith (1876) έδειξε τα πλεονεκτήματα της υποδιαίρεσης καθηκόντων αναλύοντας τα δεδομένα της παραγωγής (pin production). Ενώ σαν μειονέκτημα αναφέρεται η ανάγκη συντονισμού μεταξύ υποκαθηκόντων. Το Business Process Management είναι υπεύθυνο για την δημιουργία συντονιστικών μηχανισμών, με σκοπό την αύξηση της αποδοτικότητας συστημάτων που παράγουν αγαθά ή υπηρεσίες, εφαρμόζοντας υποδιαίρεση καθηκόντων. Ο Frederic Taylor (1911) υποστήριξε την δημιουργία ενός ιδεατού περιβάλλοντος εργασίας, βασισμένο σε επιστημονικές μεθόδους, βρίσκοντας τον πιο αποδοτικό τρόπο εκτέλεσης ανεξάρτητων εργασιακών βημάτων. Η αποδοτικότητα της οργάνωσης επιχειρησιακών διαδικασιών, έγινε ακόμα πιο

εμφανής με την ιδέα της γραμμής παραγωγής. Που σκέφτηκε και εφάρμοσε πρώτος ο Henry Ford (1926), δημιουργώντας έναν κύκλο παραγωγής ογδόντα ενός ωρών, πρωτοφανή για εκείνη την εποχή.

Ακαδημαϊκά ένας από τους πρώτους που ξεχώρισε την δομική από την οργάνωση διαδικασιών (process organization) ήταν ο Nordsieck (1932 και 1934). Περιέγραψε διάφορους τύπους διαγραμμάτων ροής εργασιών, για υποδιαίρεση και διανομή καθηκόντων, διαδοχή δραστηριοτήτων και εκχώρησης εργασιών (1932). Σε αυτό το πλαίσιο, ο Nordsieck αναγνώρισε την σειρά των βημάτων εργασίας και την χρονολογική σειρά καθηκόντων, σαν θέμα της οργάνωσης διαδικασιών (process organization), της οποίας ρόλος είναι η ενοποίηση αυτών των βημάτων (1934).

Της δεκαετίας που ακολούθησαν τον δεύτερο παγκόσμιο πόλεμο, τουλάχιστον στις γερμανόφωνες χώρες, δόθηκε μεγαλύτερη προσοχή στη δομική οργάνωση από ότι στην οργάνωση διαδικασιών (process organization). Στις αρχές της δεκαετίας του εβδομήντα, έγινε εμφανές ότι τα πληροφοριακά συστήματα είχαν την δυνατότητα να δώσουν μια νέα διάσταση. Αλλά ακόμη και τότε ο προσανατολισμός παρέμεινε στη δομική προσέγγιση της οργάνωσης. Εκείνη την περίοδο η λογική των επιχειρησιακών διαδικασιών, ήταν κώδικας σε εφαρμογές αυτοματοποιημένων συστημάτων γραμμής παραγωγής και για αυτό λόγο ήταν δύσκολο να αλλάξει. Η χρήση της τεχνολογίας αυτοματοποίησης, στα τέλη της δεκαετίας του εβδομήντα, ήταν η απαρχή για ένα πιο σαφή έλεγχο της ροής πληροφοριών και του συντονισμού δραστηριοτήτων. Η δουλειά των Ellis και Nutts (1979-1980) και του Zisman (1977-1978), έδωσαν μοντέλα επιχειρησιακών διαδικασιών βασισμένα στα δίκτυα Petri. Επίσης εκείνη την περίοδο αναπτύχθηκε και η SADT που αποτέλεσε προπομπό της IDEF0, που ακολούθησε στις αρχές της δεκαετίας του ογδόντα.

Παρόλο που η επιχειρησιακή αξία των διαδικασιών τράβηξε την προσοχή κατά την δεκαετία του ογδόντα και κάποιες καινοτομίες είχαν γίνει με την υποστήριξη των πληροφοριακών συστημάτων, ήταν στις αρχές της δεκαετίας του ενενήντα που το work flow management επικράτησε σαν νέα τεχνολογία για να υποστηρίξει τις επιχειρησιακές διαδικασίες. Ένας αυξανόμενος αριθμός εταιριών παροχής εμπορικών πακέτων επωφελήθηκαν από τις νέες ιδέες στη διοίκηση επιχειρήσεων, όπως τα προγράμματα Process Innovation και Process Reengineering.

Τα προαναφερθέντα επιχειρησιακά προγράμματα βασίζονται στη τεχνολογία των πληροφοριακών συστημάτων, και αναφέρονται σε συγκεκριμένα work flow συστήματα, με σκοπό τη βελτίωση της αποδοτικότητας τους. Μέχρι το τέλος της δεκαετίας του ενενήντα οι διεπιχειρησιακές διαδικασίες ήταν το πεδίο στο οποίο εστιαζόταν η διοίκηση επιχειρησιακών διαδικασιών (Business Process Management).

Φτάνοντας στην δεκαετία που διανύουμε, με την ύπαρξη του eXtended Markup Language (XML) και των διαδικτυακών υπηρεσιών, η εφαρμογή σεναρίων για πλήρη περιγραφή και προσδιορισμό των επιχειρησιακών διαδικασιών έχει γίνει πολύ πιο εύκολη εκτελεστεί.

Σήμερα, η διοίκηση επιχειρησιακών διαδικασιών είναι μια σημαντική ερευνητική περιοχή που συνδυάζει γνώση από διοίκηση επιχειρήσεων, θεωρία οργάνωσης, επιστήμη υπολογιστών και προγράμματα που υποστηρίζονται από υπολογιστές.. Επίσης είναι μια μεγάλη αγορά για τους πωλητές λογισμικού, για παροχή υπηρεσιών IT καθώς και για τους σύμβουλους επιχειρήσεων.

### 2.3 ΟΡΙΣΜΟΣ ΤΗΣ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ

Από το ξεκίνημα της θεωρίας οργάνωσης, πολύ ορισμοί επιχειρησιακών διαδικασιών έχουν προταθεί. Στο ξεκίνημα της δεκαετίας του τριάντα ο Nordsieck περιέγραψε μια επιχειρησιακή διαδικασία σαν μια αλληλουχία ενεργειών που παρέχουν μία έξοδο. Συνεχίζοντας με το ίδιο σκεπτικό, προσδιόρισε μια ενέργεια ως τη μικρότερη ξεχωριστή μονάδα εργασίας που εκτελείται από ένα υποκείμενο.

Οι Becker και Kugeler (2003) πρότειναν τον παρακάτω ορισμό:

«Μια διαδικασία είναι μια τελείως κλειστή, χρονικά και λογικά αλληλουχία ενεργειών οι οποίες είναι απαραίτητες για να δουλέψεις πάνω σε ένα προσανατολισμένο στη διαδικασία επιχειρησιακό αντικείμενο. Ένα τέτοιο αντικείμενο μπορεί να είναι, για παράδειγμα, ένα τιμολόγιο, η παραλαβή παραγγελίας ή ένα δείγμα. Μια επιχειρησιακή διαδικασία είναι μια ιδιαίτερη διαδικασία που καθοδηγείται από τους επιχειρησιακούς στόχους μιας εταιρίας και από το επιχειρησιακό περιβάλλον. Βασικό χαρακτηριστικό μιας επιχειρησιακής διαδικασίας η αλληλεπίδραση με τους επιχειρησιακούς συνεργάτες της εταιρίας (όπως οι πελάτες και οι προμηθευτές).»

Η διοίκηση επιχειρησιακών διαδικασιών μπορεί οριστεί σαν το σετ των διοικητικών ενεργειών που σχετίζονται με την επιχειρησιακή διαδικασία. Στην ουσία, οι διοικητικές ενέργειες που σχετίζονται τις επιχειρησιακές διαδικασίες ιδεαλιστικά οργανωθεί σε έναν κύκλο ζωής. Ένας κύκλος ζωής αποτελείται από την ανάλυση δραστηριοτήτων (activities analysis), τον σχεδιασμό (design), την εγκατάσταση (implementation), την περιγραφή (enactment), τον έλεγχο (monitoring) και την αξιολόγηση (evaluation). Παρακάτω τα μέρη ενός κύκλου ζωής επεξηγούνται ένα προς ένα.

- Ο κύκλος ζωής της διοίκησης επιχειρησιακών διαδικασιών ξεκινάει με την ανάλυση δραστηριοτήτων. Αυτή η ανάλυση περιλαμβάνει και το περιβάλλον της διαδικασίας και την οργανωτική δομή. Η έξοδος αυτού του βήματος είναι ένα σετ από απαιτήσεις για την επιχειρησιακή διαδικασία.
- Αυτές οι απαιτήσεις οδηγούν σε έναν επακόλουθο σχεδιασμό δραστηριοτήτων. Συγκεκριμένα ο σχεδιασμός περιέχει την αναγνώριση των διαδικαστικών δραστηριοτήτων, τον ορισμό της σειράς τους, την αντιστοίχιση των πηγών στις δραστηριότητες και έναν ορισμό της οργανωτικής δομής. Αυτές οι διαφορετικές όψεις της διαδικασίας σχεδιασμού

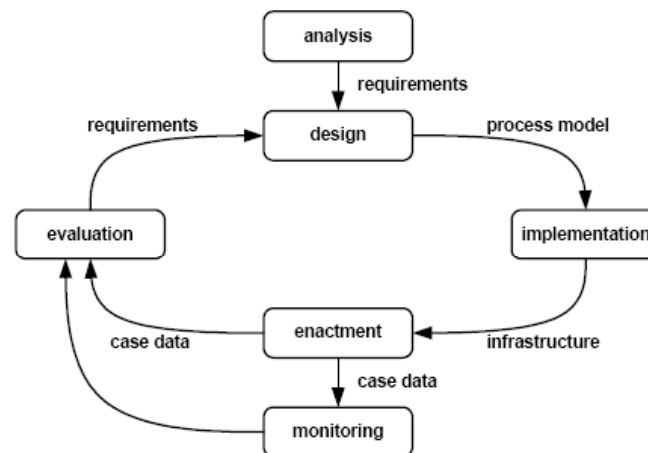
τυποποιούνται ως μοντέλο επιχειρησιακής διαδικασίας (Business Process Model). Το μοντέλο που προκύπτει μπορεί να ελεγχθεί αν πληροί τις προϋποθέσεις σχεδιασμού μέσω της χρήσης προσομοίωσης.

- Το μοντέλο της διαδικασίας εισέρχεται σαν είσοδος για την εγκατάσταση. Σε αυτή τη φάση η υποδομή για την επιχειρησιακή διαδικασία είναι έτοιμη. Αυτό περιέχει ανάμεσα στα άλλα την εκπαίδευση του προσωπικού. Αν η εκτέλεση της διαδικασίας πρέπει να υποστηρίζεται από πληροφοριακά συστήματα, το μοντέλο της διαδικασίας χρησιμοποιείται σαν σχέδιο για την εγκατάσταση.
- Όταν η διαδικασία της εγκατάστασης ολοκληρώνεται, η ακριβής περιγραφή της διαδικασίας μπορεί να ξεκινήσει. Σε αυτή τη φάση μια συγκεκριμένη υποδομή χρησιμοποιείται για να ελέγξει ξεχωριστές περιπτώσεις καλυμμένες από την επιχειρησιακή διαδικασία. Η περιγραφή παράγει πληροφορίες όπως κατανάλωση χρόνου, πηγές, υλικά και άλλα για κάθε ελεγχόμενη περίπτωση. Αυτό αποτελεί την είσοδο για τον έλεγχο και την αξιολόγηση.
- Ο έλεγχος είναι μια συνεχής ενέργεια και εξαρτάται από τις προϋποθέσεις της διαδικασίας (π.χ. ο μέγιστος χρόνος αναμονής για μια συγκεκριμένη διαδικασία). Ο έλεγχος επιφέρει συνέπειες αν αποδειχθεί ότι υφίσταται μια προβληματική κατάσταση.
- Η αξιολόγηση από την άλλη, μελετά περιπτώσεις πληροφοριών σε ένα αθροιστικό επίπεδο. Τα αποτελέσματα της απόδοσης συγκρίνονται με τις αρχικές απαιτήσεις και αναζητούνται πιθανοί τρόποι βελτίωσης (αν χρειάζεται). Με αυτό τον τρόπο η αξιολόγηση οδηγεί σε νέες απαιτήσεις και αποτελεί την είσοδο για την επόμενη στροφή του κύκλου ζωής επιχειρησιακής διαδικασίας.

Ο κύκλος ζωής της διοίκησης επιχειρησιακών διαδικασιών αποκαλύπτει ότι τα μοντέλα επιχειρησιακών διαδικασιών παίζουν σημαντικό ρόλο στον σχεδιασμό, στην εγκατάσταση και στη φάση της περιγραφής, ειδικά όταν τα πληροφοριακά συστήματα υποστηρίζουν την περιγραφή της διαδικασίας. Καθώς είναι μια αξιολόγηση

πηγή για την συνεχή βελτίωση μιας διαδικασίας, τη διοίκηση ολικής ποιότητας, το Knowledge Management, τα ERP συστήματα και εγκαταστάσεις λογισμικού.

Στο παρακάτω σχήμα φαίνονται τα βήματα ενός κύκλου ζωής που αναλύθηκαν παραπάνω.



Σχήμα 2.1: Κύκλος ζωής επιχειρησιακής διαδικασίας

## 2.4 ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ

Η μοντελοποίηση των διαδικασιών μιας επιχείρησης, με στόχο την αναδιοργάνωση, κερδίζει ολοένα και περισσότερο έδαφος στις σύγχρονες επιχειρήσεις. Η μοντελοποίηση αυτή, στοχεύει όχι μόνο στην περιγραφή των διαδικασιών αλλά και στην ανάλυση και βελτιστοποίηση τους, μέσω χρονικών αναλύσεων και αναλύσεων κόστους.

Η μοντελοποίηση μιας επιχείρησης αποτελεί ένα βασικό εργαλείο για την κατανόηση, την ανάλυση, την σχεδίαση και την βελτίωση της επιχείρησης, μειώνοντας συγχρόνως την σχετική πολυπλοκότητα.

Στόχος μιας προσπάθειας μοντελοποίησης ενός οργανισμού δεν είναι η μοντελοποίηση ολόκληρου του οργανισμού με κάθε λεπτομέρεια, αλλά η ανάλυση

και η μοντελοποίηση εκείνων των διαδικασιών των οποίων η εκτέλεση και ο έλεγχος θα μπορούν να αυτοματοποιηθούν.

Λόγω της πολυπλοκότητας και του μεγέθους μιας επιχείρησης το επιχειρηματικό μοντέλο συνήθως αποτελείται από ομάδες μοντέλων. Επιπλέον υπάρχουν πολλές διαφορετικές οπτικές γωνίες από τις οποίες μπορεί κανείς να αναλύσει μία επιχείρηση, με αποτέλεσμα να υπάρχει ποικιλία μοντέλων για διαφορετικούς σκοπούς.

## 2.5 ΟΡΙΣΜΟΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΜΟΝΤΕΛΟΥ

Το “Μοντέλο” αποτελεί μία θεωρητική αναπαράσταση της πραγματικότητας. Ο δημιουργός ενός μοντέλου ορίζει τις όψεις εκείνες του πραγματικού συστήματος οι οποίες τον ενδιαφέρουν και τα στοιχεία του υπό εξέταση συστήματος τα οποία θα μοντελοποιήσει. Η αξία ενός μοντέλου προκύπτει από την ικανότητά του να παρέχει μία απλοποιημένη απεικόνιση του πραγματικού συστήματος και να προβλέπει συγκεκριμένα γεγονότα σχετικά με το σύστημα αυτό. Η ζωή των μοντέλων επιμηκύνεται όταν αυτά χρησιμοποιούνται σε καθημερινή βάση και έχουν σχεδιαστεί έτσι ώστε να ανταποκρίνονται στη μακροπρόθεσμη στρατηγική της επιχείρησης.

Το “Επιχειρηματικό Μοντέλο” ορίζεται ως μία συμβολική αναπαράσταση της επιχείρησης και των θεμάτων τα οποία την αφορούν και αποτελείται από συμπληρωματικά μεταξύ τους μοντέλα των επιμέρους όψεων της επιχείρησης. Περιέχει αναπαραστάσεις μεμονωμένων γεγονότων, αντικειμένων, σχέσεων και συναλλαγών της επιχείρησης. Είναι σημαντικό το επιχειρηματικό μοντέλο να περιέχει όχι μόνο στατικές αλλά και δυναμικές όψεις της επιχείρησης.



## 2.6 ΕΙΔΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΜΟΝΤΕΛΩΝ

Το επιχειρηματικό μοντέλο αποτελεί μία συμβολική αναπαράσταση της επιχείρησης η οποία προκύπτει με την αφαιρετική μέθοδο. Συνήθως ένα επιχειρηματικό μοντέλο αποτελείται (χωρίς να περιορίζεται σε αυτά μόνο) από:

1. Οργανωσιακά μοντέλα, τα οποία τεκμηριώνουν την οργανωσιακή δομή και τις υπευθυνότητες και δικαιοδοσίες των μελών της.
2. Μοντέλα βελτιστοποίησης και λήψης αποφάσεων, τα οποία χρησιμοποιούνται από υποστηρικτικά συστήματα λήψης αποφάσεων (DSSs).
3. Μοντέλα δραστηριοτήτων, τα οποία υπαγορεύουν τις λειτουργίες και ενέργειες που πρέπει να εκτελεστούν στις επιχειρηματικές δραστηριότητες
4. μοντέλα πόρων, τα οποία περιγράφουν τα χαρακτηριστικά, τις πολιτικές διαχείρισης των πόρων, και τις ενέργειες που γίνονται για τη διεκπεραίωση διαφόρων δραστηριοτήτων.
5. Μοντέλα πληροφορίας, τα οποία περιγράφουν τη δομή και τις αλληλοσυσχετίσεις των δεδομένων και των πληροφοριακών στοιχείων του επιχειρηματικού πληροφοριακού συστήματος.
6. Οικονομικά μοντέλα, τα οποία παρέχουν μία αναλυτική όψη για τα διάφορα κόστη της επιχείρησης.
7. Μοντέλα προϊόντων, για την αναπαράσταση των γεωμετρικών και μη-γεωμετρικών χαρακτηριστικών του προϊόντος καθώς και των λεπτομερειών της σχεδίασης του προϊόντος και των τμημάτων που το απαρτίζουν σε όλη τη διάρκεια του κύκλου ζωής του.

## 2.7 ΕΙΔΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ

Οι διαδικασίες ανάλογα με το πόσο αυστηρά δομημένες ή ευέλικτες είναι, διακρίνονται σε :

1. Ασθενώς-δομημένες, των οποίων ούτε το τελικό αποτέλεσμα, ούτε η αλληλουχία των δραστηριοτήτων είναι απολύτως γνωστά. Αυτές αφορούν:
  - Ad-hoc workflows, όπου η συγκέντρωση και δρομολόγηση της πληροφορίας δεν ακολουθεί κάποια καθορισμένη πορεία αλλά είναι αυθόρμητη.
  - Cooperative team workflows, των οποίων η δομή προκύπτει από τη συνεργασία μιας ομάδας ατόμων.
2. Ημι-δομημένες, των οποίων το αναμενόμενο αποτέλεσμα είναι γνωστό, αλλά η διαδοχή των δραστηριοτήτων γίνεται γνωστή μόνο κατά την ώρα της εκτέλεσης. Αυτές αποτελούνται από:
  - Αλυσιδωτές δραστηριότητες (chained activity)
  - Ολοκληρωμένες δραστηριότητες μιας ομάδας ανθρώπων με συγκεκριμένη δομή (integrated team activity).
3. Καλά δομημένες, των οποίων το αναμενόμενο αποτέλεσμα είναι γνωστό και η διαδοχή των δραστηριοτήτων καλά ορισμένη (ντετερμινιστική). Αυτές αφορούν standard workflow, δηλ. προτυποποιημένα, με συγκεκριμένη δομή τα οποία αυτοματοποιούν συνήθως επαναλαμβανόμενες δραστηριότητες. Στις καλά δομημένες διαδικασίες όποτε συμβεί κάποια εξαίρεση (ad-hoc exception), τότε ο τελικός χρήστης μπορεί να αποκλίνει από το στάνταρτ ορισμό της διαδικασίας.

Οι δραστηριότητες (activities) διακρίνονται και αυτές σε :

- δομημένες, των οποίων η συμπεριφορά είναι προβλέψιμη.
- μη-δομημένες, των οποίων η συμπεριφορά δεν είναι προβλέψιμη.

## **ΚΕΦΑΛΑΙΟ 3**

### **ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ**

#### **3.1 Η ΓΛΩΣΣΑ IDEF0**

Κατά τη διάρκεια της δεκαετίας του 70, με το πρόγραμμα ICAM (Integrated Computer Aided Manufacturing), η πολεμική αεροπορία των Η.Π.Α. επιχείρησε να αυξήσει την κατασκευαστική παραγωγικότητα μέσω της συστηματοποιημένης εφαρμογής της τεχνολογίας των υπολογιστών. Το παραπάνω πρόγραμμα διαπίστωσε την ανάγκη για καλύτερη ανάλυση και ανάπτυξη επικοινωνιακών τεχνικών, για τους ανθρώπους που συμμετέχουν στην βελτίωση της κατασκευαστικής παραγωγικότητας. Σαν αποτέλεσμα των παραπάνω, το πρόγραμμα ICAM ανέπτυξε μια σειρά από μεθόδους, γνωστές ως IDEF (ICAM Definition). Κάποιες από αυτές είναι οι ακόλουθες:

- IDEF0: Χρησιμοποιείται για να κατασκευαστεί μια μοντελοποίηση λειτουργιών. Ένα τέτοιο μοντέλο είναι μια δομημένη αναπαράσταση λειτουργιών, δραστηριοτήτων ή διαδικασιών ενός συστήματος.
- IDEF1: Χρησιμοποιείται για τη δημιουργία ενός πληροφοριακού μοντέλου. Ένα πληροφοριακό μοντέλο αναπαριστά τη δομή και τη σημασιολογία των πληροφοριών σε ένα σύστημα.
- IDEF1X: Το 1983 αναπτύχθηκε μια βελτιωμένη έκδοση της μεθοδολογίας IDEF1, η IDEF1X (IDEF Extended), που όπως και η IDEF1 χρησιμοποιείται για την ανάπτυξη ενός πληροφοριακού μοντέλου.
- IDEF2: Χρησιμοποιείται κατά την κατασκευή ενός δυναμικού μοντέλου. Ένα δυναμικό μοντέλο αναπαριστά μια συμπεριφορά χαρακτηριστικών που αλλάζουν κατά το πέρασμα του χρόνου σε ένα σύστημα.

- Εντός του πλαισίου IDEF συμπεριλαμβάνονται συνολικά 15 μεθοδολογίες μοντελοποίησης, κάθε μια από τις οποίες σχηματίζει ένα ανεξάρτητο μοντέλο.

Οι μεθοδολογίες IDEF0 και IDEF1X είναι οι ευρύτερα χρησιμοποιούμενες στην κυβέρνηση, εργοστασιακών και εμπορικών τομέων, υποστηρίζοντας προσπάθειες μοντελοποίησης για ποικίλες εταιρίες και πεδία εφαρμογών.

Στην πορεία της διπλωματικής εργασίας θα χρησιμοποιηθεί η μεθοδολογία μοντελοποίησης IDEF0, η οποία αναλύεται παρακάτω.

### 3.1.1 Η ΠΡΟΣΕΓΓΙΣΗ IDEF0

Η μεθοδολογία μοντελοποίησης IDEF0, βασίζεται στη μεθοδολογία SADT (Structured Analysis and Design Technique) που αναπτύχθηκε από τον Douglas T. Ross και την SofTech Inc. Η μεθοδολογία SADT αναπτύχθηκε στο τέλος της δεκαετίας του εξήντα (ξεκίνησε το 1969 και η πρώτη της μεγάλη εφαρμογή έγινε το 1973), κατά τη διάρκεια της επανάστασης του δομημένου προγραμματισμού. Η IDEF0 στην αυθεντική της μορφή, συμπεριλαμβάνει και ορισμούς της γραφικής γλώσσας μοντελοποίησης και περιγραφή της μεθοδολογίας για τα αναπτυσσόμενα μοντέλα.

Η IDEF0 μπορεί να χρησιμοποιηθεί για μια μεγάλη ποικιλία αυτοματοποιημένων ή μη συστημάτων. Για καινούρια συστήματα, η IDEF0 μπορεί να χρησιμοποιηθεί πρώτα για να οριστούν οι απαιτήσεις και να καθοριστούν οι λειτουργίες και μετά να σχεδιαστεί η υλοποίηση που πληρεί τις προϋποθέσεις και υλοποιεί τις λειτουργίες. Για ήδη υπάρχοντα συστήματα, η IDEF0 χρησιμοποιείται για να αναλυθούν οι λειτουργίες του συστήματος και να καταγραφούν οι μηχανισμοί (μέσα) μέσω των οποίων αυτές εκτελούνται.

Το αποτέλεσμα της εφαρμογής της μεθόδου IDEF0 σε ένα σύστημα είναι ένα μοντέλο που αποτελείται από μια ιεραρχική σειρά διαγραμμάτων, κειμένου και ειδικό λεξιλόγιο που τα συνδέει. Τα δύο κυριότερα στοιχεία που συνθέτουν την μέθοδο, είναι οι λειτουργίες (που αναπαριστούνται στο διάγραμμα με κουτιά) και τις

πληροφορίες και αντικείμενα που συνδέουν τις λειτουργίες (αναπαριστούνται στο διάγραμμα με βέλη).

Γενικότερα σαν γλώσσα μοντελοποίησης λειτουργιών, η IDEF0 έχει τα ακόλουθα χαρακτηριστικά:

- Είναι περιεκτική και εκφραστική, ικανή να αναπαραστήσει γραφικά μία μεγάλη ποικιλία επιχειρησιακών, παραγωγικών και άλλου τύπου λειτουργιών ενός οργανισμού σε οποιονδήποτε βαθμό λεπτομέρειας.
- Είναι μία κατανοητή και απλή γλώσσα, με το να παρέχει τη δυνατότητα ακριβούς και λογικά συνεπούς έκφρασης και με το να προάγει συνέπεια στη χρήση και την επεξήγηση της.
- Προάγει την επικοινωνία μεταξύ αναλυτών, σχεδιαστών συστημάτων και χρηστών με το να είναι εύκολη προς εκμάθηση και με το να δίνει έμφαση στην ιεραρχική αποκάλυψη λεπτομέρειας.
- Είναι αποδεδειγμένα δοκιμασμένα, μετά από πολλά χρόνια αξιοποίησης της σε κυβερνητικά αναπτυξιακά προγράμματα και στην ιδιωτική βιομηχανία.
- Είναι δυνατό να χρησιμοποιηθεί από ποικιλία σχεδιαστικών εργαλείων Η/Υ. Πολλά εμπορικά προϊόντα υποστηρίζουν ειδικά την ανάπτυξη και ανάλυση διαγραμμάτων και μοντέλων IDEF0.

### 3.1.2 ΣΥΝΤΑΞΗ ΜΟΝΤΕΛΟΥ

Οι δομικές συνιστώσες και τα χαρακτηριστικά μιας γλώσσας καθώς και οι κανόνες που ορίζουν τις μεταξύ τους σχέσεις, αναφέρονται σαν γλώσσα σύνταξης. Τα κύρια στοιχεία της σύνταξης της IDEF0 είναι, τα κουτιά και τα τόξα που αποτελούν τα διαγράμματα. Τα κουτιά αναπαριστούν λειτουργίες, που μπορεί να είναι ενέργειες, διαδικασίες ή μετασχηματισμοί. Τα τόξα αναπαριστούν πληροφορίες ή αντικείμενα που σχετίζονται με τις λειτουργίες. Οι κανόνες ορίζουν πως τα στοιχεία

χρησιμοποιούνται και τα διαγράμματα παρέχουν ένα σχέδιο για απεικόνιση του μοντέλου και λεκτικά και γραφικά.

#### 3.1.2.1 Κουτιά

Ένα κουτί παρέχει μια περιγραφή για το τι συμβαίνει σε μια συγκεκριμένη λειτουργία. Το κουτί στα όρια του περιέχει το όνομά του και το νούμερό του. Το όνομα πρέπει να είναι μια λέξη ή μια πρόταση που να περιγράφει την λειτουργία. Επίσης κάθε κουτί πρέπει στη κάτω δεξιά του γωνία να έχει ένα νούμερο. Το νούμερο αυτό χρησιμεύει στην αναγνώριση των λειτουργιών κατά την ιεραρχική ανάλυση. Ένα κουτί πρέπει να είναι ορθογώνιο παραλληλόγραμμο και να είναι αρκετά μεγάλο ώστε να χωράει η ονομασία του.

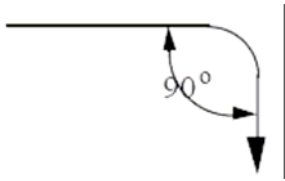
#### 3.1.2.2 Τόξα

Ένα τόξο σχηματίζεται από ένα ή περισσότερα ευθύγραμμα τμήματα, με μια κεφαλή τόξου στο τέλος. Τα τόξα μπορεί να είναι ευθείες γραμμές ή και να περιέχουν γωνία (πάντα ενενήντα μοιρών). Επίσης τα τόξα μπορούν διακλαδίζονται (είτε να ενώνονται, είτε να χωρίζονται). Τα τόξα δεν αντιπροσωπεύουν ροή ή συνέχεια όπως στα μοντέλα διαγραμμάτων ροής, αλλά μεταφέρουν πληροφορίες ή αντικείμενα σχετικά με τις λειτουργίες που συνδέουν.

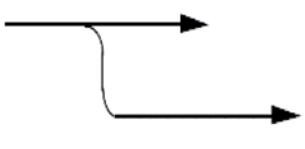
Τα τόξα πρέπει να έχουν οριζόντιες ή κάθετες γραμμές και όχι διαγώνιες, πρέπει να ξεκινάνε ή να τερματίζουν επάνω στην περίμετρο των κουτιών και να μην εισέρχονται μέσα σε αυτά. Τέλος τα τόξα πρέπει να ακουμπάνε στις πλευρές των κουτιών και όχι στις γωνίες.



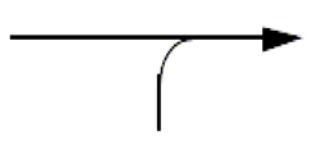
Τόξο.



Τόξο που σχηματίζει γωνία.



Τόξο που διακλαδίζεται.



Τόξο με κλάδους που ενώνονται.

### 3.1.2.3 Σημασιολογία

Η σημασιολογία αναφέρεται στο νόημα των συντακτικών στοιχείων της γλώσσας και συντελούν στην ορθότητα της ερμηνείας. Ερμηνεία που έχει να κάνει με την σημειογραφία κουτιών και βελών και την αλληλεπίδραση των λειτουργικών σχέσεων.

### 3.1.2.4 Σημασιολογία κουτιών και βελών

Εφόσον η IDEF0 υποστηρίζει μοντέλα λειτουργιών, η ονομασία των κουτιών πρέπει να γίνεται με ρήματα ή με φράσεις, έτσι ώστε να γίνεται κατανοητή η λειτουργία κάθε κουτιού. Το αποφασιστικό βήμα για την ονομασία των κουτιών είναι η ενσωμάτωση των βελών (ανάλογα και με τον προσανατολισμό τους στις πλευρές των κουτιών) που συμπληρώνουν και ολοκληρώνουν την δυναμική της έκφρασης της IDEF0.



Μια τυποποιημένη ορολογία πρέπει να υιοθετηθεί, ώστε να διασφαλιστεί μια ακρίβεια στην ανάγνωση των μοντέλων. Μια λειτουργία (κουτί) μπορεί να αναλυθεί σε περισσότερες, σε ένα νέο διάγραμμα κατώτερου επιπέδου που δείχνει τις διαδικασίες που ενεργούνται για να περατωθεί. Τα ονόματα των βελών πρέπει να είναι ουσιαστικά ή φράση χωρίς όμως ρήμα. Επίσης τμήματα βελών είτε πριν από ένωση είτε μετά από διακλάδωση, έχουν πάντα ξεχωριστά ονόματα που αναφέρονται στην πληροφορία που φέρει κάθε ξεχωριστός κλάδος.

Κάθε πλευρά του κουτιού έχει μια ιδιαίτερη σημασία στην σχέση κουτιού – βέλους. Η λογική είναι ότι η λειτουργία του κουτιού μετατρέπει τις εισόδους σε εξόδους. Πιο συγκεκριμένα τα βέλη που μπαίνουν από την αριστερή πλευρά του κουτιού είναι οι εισοδοί (inputs, τα βέλη πάντα τερματίζουν σε αυτή την πλευρά και ποτέ δεν αναχωρούν), ενώ από την δεξιά πλευρά βγαίνουν οι εξοδοί (outputs, τα βέλη πάντα αναχωρούν από αυτή την πλευρά και ποτέ δεν τερματίζουν). Τα βέλη που εισέρχονται από την πάνω πλευρά του κουτιού, είναι τα βέλη έλεγχου (controls, πάντα τερματίζουν και ποτέ δεν αναχωρούν βέλη από αυτή την πλευρά). Από την κάτω πλευρά έχουμε τα βέλη που ονομάζονται μηχανισμοί (mechanisms), τα βέλη αυτά μπορεί είτε να εισέρχονται είτε να εξέρχονται από το κουτί. Τα βέλη που εισέρχονται στην κάτω πλευρά του κουτιού είναι οι μηχανισμοί που υποστηρίζουν την λειτουργία. Αντίθετα τα βέλη που εξέρχονται (ονομάζονται call arrows) καθιστούν ικανό το μοίρασμα των λεπτομερειών μεταξύ μοντέλων ή κομματιών του ίδιου μοντέλου. Τα controls βέλη δείχνουν κάτω από ποιες συνθήκες η μετατροπή των εισόδων σε εξόδους διεκπεραιώνεται, ενώ τα mechanism βέλη περιγράφουν πως η λειτουργία διεκπεραιώνεται.

Αυτοί οι τέσσερις διαφορετικοί τύποι βελών μας δίνουν τον λεγόμενο μηχανισμό ICOM (Input, Control, Output, Mechanism). Οι ICOMs συμβάσεις είναι ένα από τα πιο ισχυρά σημεία της μεθόδου IDEF0, καθώς οδηγούν σε ευκολία επικοινωνίας.

### 3.1.3 ΤΑ IDEF0 ΔΙΑΓΡΑΜΜΑΤΑ

#### 3.1.3.1 Τύποι διαγραμμάτων

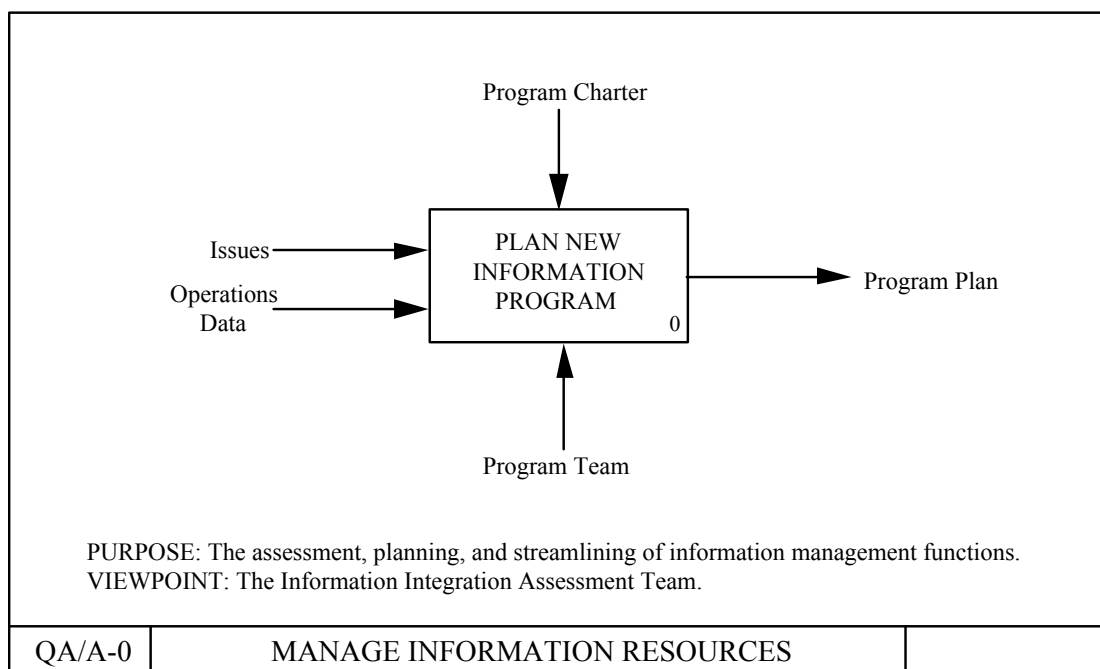
Τα IDEF0 μοντέλα αποτελούνται από τρεις διαφορετικούς τύπους πληροφοριών: γραφικά διαγράμματα, κείμενο και γλωσσάριο. Αυτοί οι τρεις τύποι υπάρχουν για να παρέχουν μια διασταυρωτική αναφορά μεταξύ των τριών ώστε το διάγραμμα να γίνεται κατανοητό. Το βασικό στοιχείο του μοντέλου βέβαια είναι τα γραφικά διαγράμματα, που αποτελούνται κουτιά και τα βέλη. Όπως προείπαμε τα κουτιά συμβολίζουν λειτουργίες, οι οποίες μπορούν να αναλύονται σε άλλα πιο λεπτομερειακά διαγράμματα, μέχρι η συγκεκριμένη λειτουργία να έχει αναλυθεί πλήρως σύμφωνα με τους στόχους του θέματος. Το διάγραμμα του υψηλότερου επιπέδου στο μοντέλο παρέχει τη γενικότερη περιγραφή του θέματος που αναπαριστάται από το μοντέλο. Αυτό το διάγραμμα ακολουθείται από μια σειρά "διαγράμματα παιδιά" (child diagrams) παρέχοντας μια πιο λεπτομερειακή ανάλυση του θέματος.

#### 3.1.3.2 Διάγραμμα υψηλότερου επιπέδου

Κάθε μοντέλο πρέπει να έχει ένα διάγραμμα υψηλότερου επιπέδου, που αποτελείται από ένα μόνο κουτί και τα βέλη του και καλείται A-0 διάγραμμα. Από την στιγμή που το A-0 διάγραμμα αποτελείται από ένα μόνο κουτί, το όνομα που αυτό παίρνει αντιπροσωπεύει την γενικότερη λειτουργία του μοντέλου. Επίσης τα τόξα αντιπροσωπεύουν τις γενικότερες ροές του μοντέλου. Το A-0 δείχνει τον γενικότερο αντικείμενο του μοντέλου και τον προσανατολισμό του.

Στο πρώτο διάγραμμα που αντιστοιχεί σε αυτό του ανώτερου επιπέδου, πρέπει να υπάρχει μια σύντομη περιγραφή του σκοπού (purpose) και της σκοπιάς (viewpoint) του μοντέλου. Οι δήλωση της σκοπιάς εξαρτάται από το κοινό στο οποίο απευθύνεται το μοντέλο. Η δήλωση του σκοπού εκφράζει γιατί δημιουργήθηκε το μοντέλο και καθορίζει την μορφή του. Όπως προαναφέρθηκε το A-0 διάγραμμα αναλύεται σε υπολειτουργίες και αυτές με την σειρά τους επίσης, μέχρι η όλη λογική της σκοπιάς (viewpoint) να έχει να έχει αναπαρασταθεί πλήρως. Κάθε υπολειτουργία

μοντελοποιείται ξεχωριστά από ένα κουτί, με τα κουτιά πατέρες (parent boxes), να αναλύονται από τα διαγράμματα παιδιά (child diagrams) στο επόμενο χαμηλότερο επίπεδο. Όλα τα child diagrams πρέπει να ακολουθούν τη λογική του διαγράμματος του πρώτου επιπέδου.



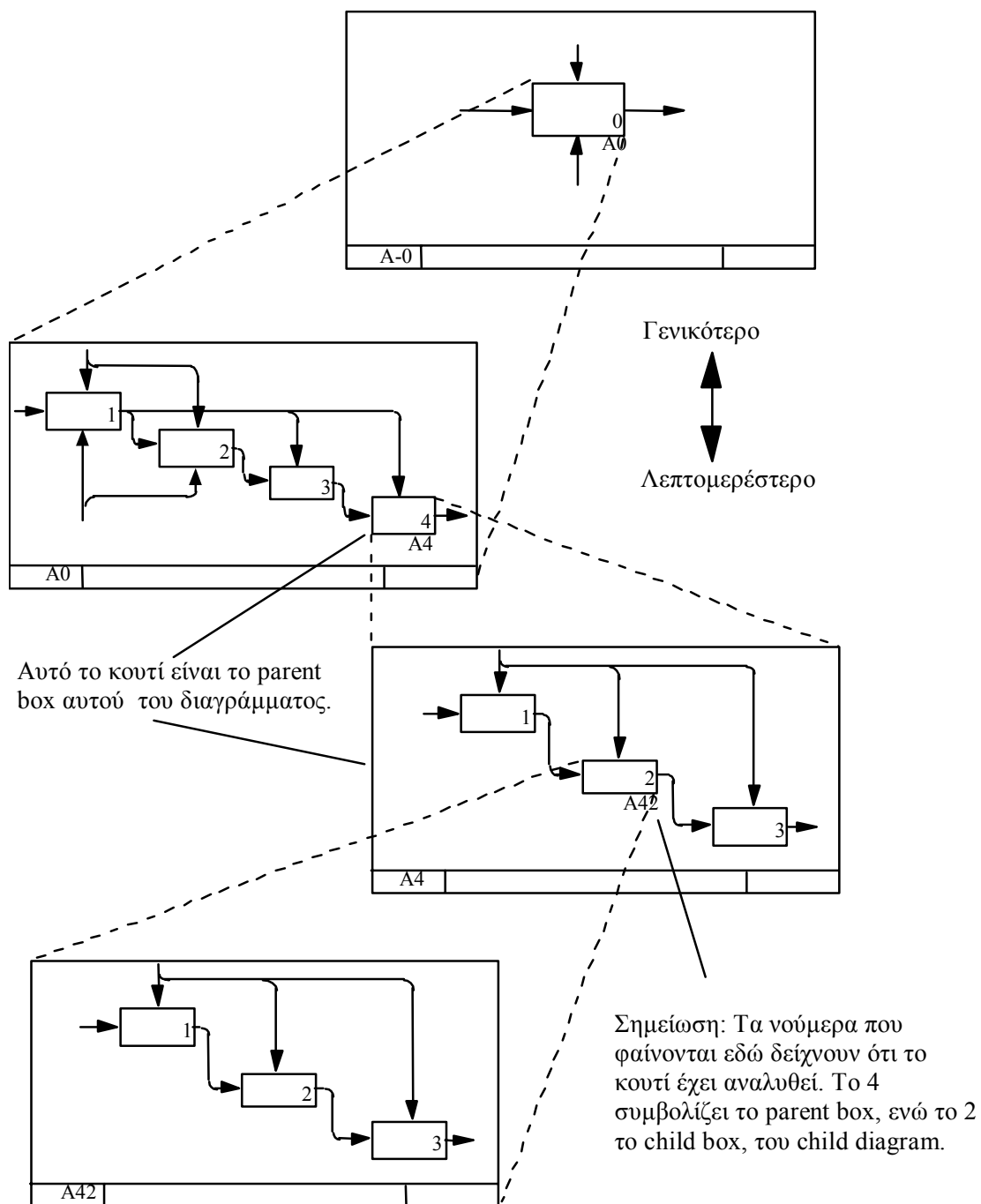
Σχήμα 3.1: Παράδειγμα διαγράμματος επιπέδου A-0.

### 3.1.3.3 Child diagram

Όπως είπαμε παραπάνω, το αρχικό διάγραμμα της γενικότερης λειτουργίας του συστήματος αναλύεται σε υπολειτουργίες δημιουργώντας ένα child diagram. Αυτές οι υπολειτουργίες μπορούν να αναλυθούν με τη σειρά τους, δημιουργώντας κι άλλα child diagrams. Σε ένα διάγραμμα, κάποιες από τις λειτουργίες, καμία από τις λειτουργίες ή όλες οι λειτουργίες μπορούν να αναλύονται περαιτέρω. Κάθε child diagram περιέχει child boxes (κουτιά) και βέλη, παρέχοντας μια πρόσθετες πληροφορίες για το parent box. Το child diagram έχει την ίδια οπτική που έχει και το parent box που αναλύεται. Γι' αυτό μπορούμε να φανταστούμε ότι ένα child diagram βρίσκεται μέσα σε ένα parent box.

#### 3.1.3.4 Parent diagram

Ένα parent diagram είναι αυτό που περιέχει ένα ή περισσότερα parent boxes. Κάθε διάγραμμα, εκτός από το A-0, είναι ένα child diagram αφού αναλύει ένα parent box. Έτσι ένα διάγραμμα μπορεί να είναι και parent diagram (περιέχοντας parent boxes) και child diagram (αναλύοντας ένα συγκεκριμένο parent box). Όμοια, ένα κουτί μπορεί να είναι και parent box (αναλυόμενο από ένα child diagram) και child box (εμφανιζόμενο σε child diagram). Η πρώτη ιεραρχικά σχέση είναι ανάμεσα σε ένα parent box και σε ένα child diagram, που το αναλύει.



Σχήμα 3.2: Δομή ανάλυσης. Parent box/diagram και child box/diagram.

Το γεγονός ότι ένα parent box είναι ταυτόχρονα και child box, γίνεται αντιληπτό από την παρουσία του αριθμού του κουτιού στην κάτω δεξιά γωνία (που λέγεται Detail Reference Expression, DRE). Είναι ο κωδικός του parent box που εμφανίζεται και στην αρίθμηση του child diagram.

Το DRE μπορεί να έχει μια από τις παρακάτω μορφές:

1. Ένα νούμερο χρονικής δημιουργίας που ονομάζεται "C-number" και προσδιορίζει μια συγκεκριμένη εκδοχή του child diagram.
2. Το νούμερο της σελίδας ενός child diagram, στο δημοσιευμένο έγγραφο στο οποίο το μοντέλο εμφανίζεται (page number).
3. Το νούμερο του κουτιού αναφέρεται μόνο στο child diagram. Αν το διάγραμμα έχει πολλαπλά επίπεδα, τότε γίνεται δύσκολος ο προσδιορισμός των λειτουργιών και των υπολειτουργιών (node number).
4. Ένα μοντέλο νούμερα σημείωσης που παραπέμπει σε ένα κείμενο που καθορίζει τις συνθήκες επιλογής μιας συγκεκριμένης εκδοχής child (note number).

#### 3.1.3.5 Κείμενο και γλωσσάριο

Ένα διάγραμμα μπορεί να συνδέεται με ένα κείμενο, το οποίο χρησιμοποιείται για να δώσει μια λιτή και περιεκτική περίληψη του διαγράμματος. Το κείμενο πρέπει να χρησιμοποιείται για να τονίζει τα σημαντικά θέματα, τις ροές και συνδέσεις των κουτιών, για να ξεκαθαριστούν τα παραπάνω.

Το γλωσσάριο πρέπει να χρησιμοποιείται για να ορίζονται τα ακρώνυμα, οι λέξεις κλειδιά και οι προτάσεις που συνδέονται με το διάγραμμα.

### 3.1.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΓΡΑΦΗΜΑΤΩΝ

#### 3.1.4.1 Βέλη σαν προϋποθέσεις

Σε ένα IDEF0 διάγραμμα τα βέλη μπορούν να αναπαριστούν αποκλειστικά πληροφορίες ή αντικείμενα. Μόνο σε χαμηλά επίπεδα ανάλυσης μπορούν να αναπαριστούν ροή ή συνέχεια λειτουργιών, όταν το θέμα έχει αναλυθεί επαρκώς. Συνδέοντας το output ενός κουτιού στο input, στο control, ή στο mechanism ενός άλλου κουτιού, σημαίνει ότι η συγκεκριμένη λειτουργία προϋποθέτει την παρουσία ενός output από προηγούμενο κουτί.

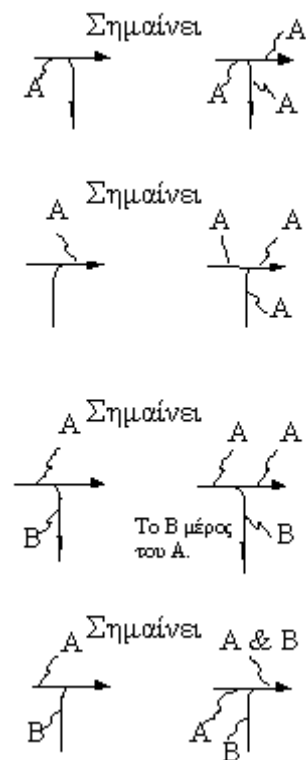
#### 3.1.4.2 Ενεργοποίηση ενός κουτιού

Ένα κουτί μπορεί να εκτελεί διάφορες λειτουργίες χρησιμοποιώντας συνδυασμούς input και control, και παράγοντας διαφορετικά outputs. Αυτές οι διαφορετικές λειτουργίες ονομάζονται, διαφορετικές ενεργοποιήσεις κουτιών (activation rules).

#### 3.1.4.3 Τόξα με κλάδους

Ένα τόξο μπορεί να έχει κλάδους, που είτε να διακλαδίζονται είτε να ενώνονται. Αυτό σημαίνει ότι το ίδιο είδος πληροφορίας ή αντικειμένου, μπορεί να είναι αναγκαίο (διακλάδωση) ή να είναι κατασκευασμένο (ένωση) από παραπάνω από μια λειτουργία. Μια διακλάδωση μπορεί να σημαίνει ότι περιέχει την ίδια πληροφορία ή αντικείμενα, ή διαφορετικά μέρη ή ποσοστά της ίδιας πληροφορίας ή αντικειμένου. Στην ίδια λογική ένα τόξο που διακλαδίζεται μπορεί να σχηματίζει κλάδους πιο λεπτομερείς (από μια γενική πληροφορία σε δύο ή περισσότερες πιο ιδιές).

#### Διακλάδωση Μετάφραση

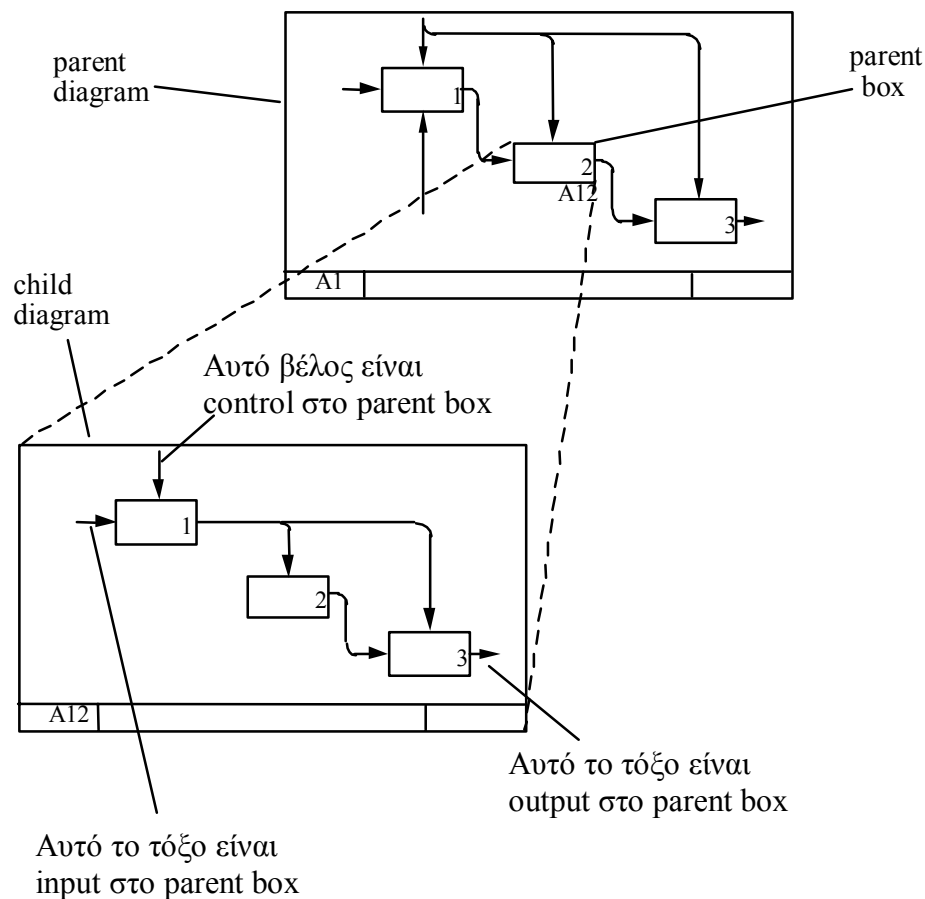


Σχήμα 3.3: Παραδείγματα διακλαδώσεων.

#### 3.1.4.4 ICOM κωδικοποίηση boundary τόξων

Boundary τόξα, είναι τα τόξα του διαγράμματος (child) που προέρχονται από το parent box. Δηλαδή αφορά διαγράμματα χαμηλότερου του A-0 διαγράμματος και μπορούν να είναι τόξα input, control, output ή mechanism.

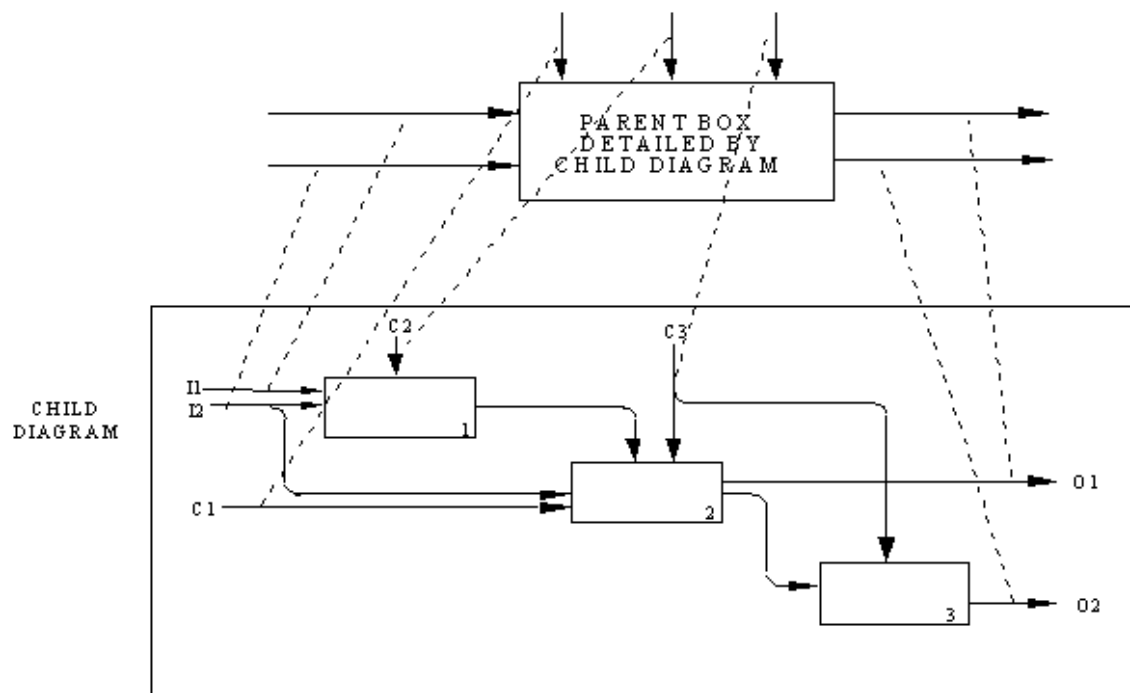




Σχήμα 3.4: Παράδειγμα boundary βελών.

Η κωδικοποίηση ICOM συνδέει τα boundary τόξα στο child diagram, με τα τόξα στο parent diagram. Η κωδικοποίηση ICOM είναι μια ειδική σημειογραφία, που συνδέει τα γράμματα I (input), C (control), O (output), M (mechanism), που εμφανίζονται στο child diagram, με τα αντίστοιχα βέλη του parent box. Πιο συγκεκριμένα τα input βέλη από το parent box, εμφανίζονται στο child diagram με κωδική ονομασία I και ένα νούμερο (π.χ. I2), κατά αύξουσα σειρά από πάνω προς τα κάτω (αύξουσα σειρά από αριστερά προς δεξιά αν πρόκειται για control ή mechanism βέλη).

Έτσι η κωδικοποίηση συνδέει το child diagram με το ακριβώς προηγούμενο parent box. Αν τα κουτιά στο child diagram αναλυθούν περαιτέρω, σε νέα child διαγράμματα, νέα κωδικοποίηση ICOM προσδιορίζεται σε κάθε νέο child diagram, συσχετίζοντας τα boundary βέλη του διαγράμματος με τα βέλη στο αμέσως προηγούμενο parent box.



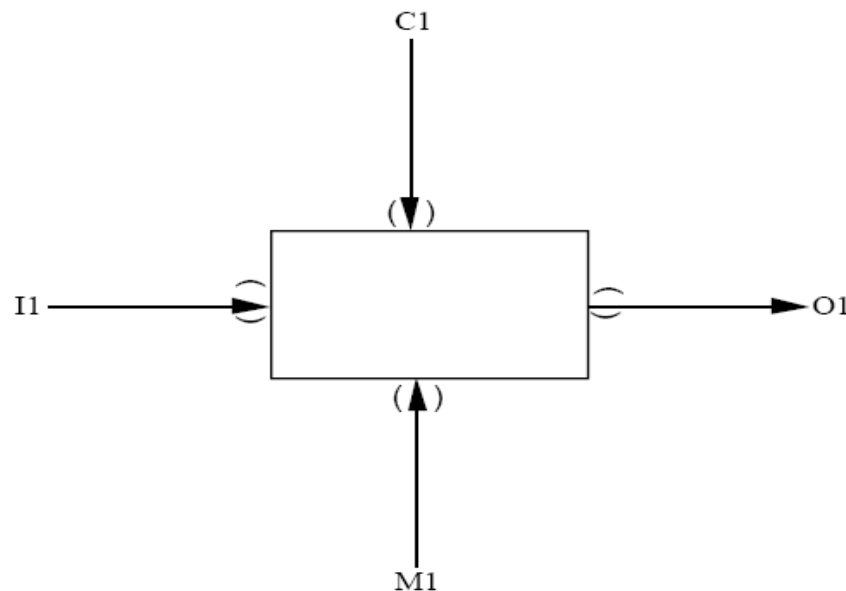
Σχήμα 3.5: Το σχήμα δείχνει πως η κωδικοποίηση ICOM συνδέει τα βέλη του parent box με τα βέλη του child diagram.

#### 3.1.4.5 Tunneled βέλη

Ένα tunneled βέλος χρησιμοποιείται για να δώσει μια πληροφορία, σε ένα συγκεκριμένο επίπεδο της ανάλυσης, που δεν απαιτείται για να γίνουν κατανοητά τα άλλα επίπεδα. Ένα βέλος μπορεί να είναι tunneled σε οποιοδήποτε επίπεδο ανάλυσης.

Χρησιμοποιώντας τον συμβολισμό της παρένθεσης στη κεφαλή ενός tunneled βέλους, δείχνουμε ότι το βέλος αυτό δεν είναι απαραίτητο για να γίνει κατανοητό το child diagram του κουτιού (άρα δεν εμφανίζεται). Ενώ το προαναφερθέν βέλος μπορεί να προέρχεται από ένα parent box. Επίσης ο συμβολισμός της παρένθεσης

στην ουρά ενός βέλους, σημαίνει ότι το συγκεκριμένο βέλος δεν είναι απαραίτητο για να γίνουν κατανοητά τα προηγούμενα επίπεδα και έτσι εμφανίζεται πρώτη φορά στο συγκεκριμένο διάγραμμα. Τα tunneled βέλη παρόλα αυτά, επειδή είναι απαραίτητα στο επίπεδο που εμφανίζονται έχουν κωδικοποίηση ICOM.



Σχήμα 3.6: Παράδειγμα tunneled βελών.

#### 3.1.4.6 Call βέλη

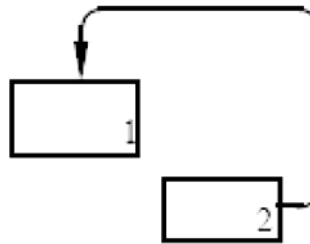
Ένα call βέλος, είναι μια ειδική περίπτωση mechanism βέλους και δηλώνει ότι το αντίστοιχο κουτί (caller box) δεν έχει child diagram. Όμως μπορεί να αναλύεται από ένα εξολοκλήρου από ένα άλλο κουτί, στο ίδιο ή σε άλλο μοντέλο. Διάφορα caller boxes, μπορούν να καλούν (call) το ίδιο κουτί.

Το βέλος αυτό παίρνει σαν όνομα το νούμερο του διαγράμματος στο οποίο βρίσκεται το called box (το κουτί που θα κληθεί) μαζί με το νούμερο του κουτιού called. Ένα caller box μπορεί να καλεί μόνο ένα κουτί κάθε φορά που ενεργοποιείται. Όμως, ανάλογα και με τις συνθήκες του μοντέλου που σημειώνεται πάνω στο call βέλος, μπορεί ένα caller box να διαλέξει ανάμεσα από κάποια called boxes. Σε αυτή την περίπτωση το call βέλος πρέπει να περιέχει την λίστα των δυνατών call boxes.

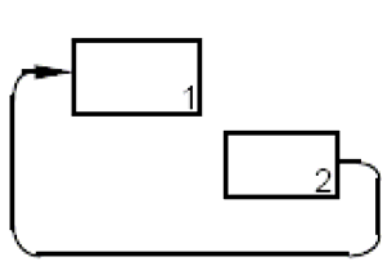
### 3.1.5 ΚΑΝΟΝΕΣ ΣΥΝΤΑΞΗΣ

1. α διαγράμματα πρέπει να έχουν αρίθμηση του τύπου A-n, όπου το n πρέπει να είναι μεγαλύτερο ή ίσο του μηδέν.
2. Το διάγραμμα πρέπει να περιέχει ένα A-0 διάγραμμα πρώτου επιπέδου, που να περιέχει μόνο ένα κουτί.
3. Το νούμερο του κουτιού στο A-0 επίπεδο πρέπει να είναι μηδέν.
4. Ένα διάγραμμα χαμηλότερου του A-0 επιπέδου, πρέπει να περιέχει το λιγότερο τρία και το περισσότερο έξι κουτιά.
5. Κάθε κουτί σε κάθε διάγραμμα πρέπει να έχει μια αρίθμηση, στην κάτω δεξιά γωνία, από μηδέν ως έξι, με αύξουσα σειρά από το πάνω αριστερά στο κάτω δεξιά.
6. Κάθε κουτί που έχει αναλυθεί πρέπει να έχει την detailed reference expression (DRE) στο child diagram, κάτω από την δεξιά γωνία κάθε τετραγώνου. (Σημείωση: Στο work flow modeler που χρησιμοποιούμε στην διπλωματική η τυποποίηση είναι διαφορετική. Το νούμερο του τετραγώνου είναι το νούμερο του διαγράμματος (π.χ. A2) και το νούμερο του κουτιού στο διάγραμμα (π.χ. 3, άρα το κουτί θα είναι το A23)).
7. Τα βέλη πρέπει να αποτελούνται από οριζόντια και κάθετα ευθύγραμμα τμήμα. Διαγώνιες γραμμές δεν πρέπει να χρησιμοποιούνται ποτέ.
8. Κάθε κουτί πρέπει να έχει τουλάχιστον ένα control και ένα output βέλος.
9. Κάθε κουτί πρέπει να έχει μηδέν ή παραπάνω input βέλη.

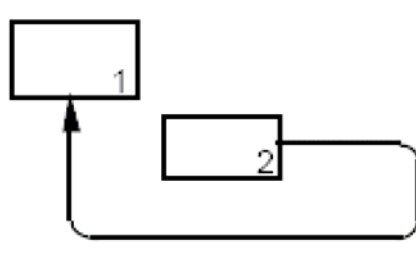
10. Κάθε κουτί πρέπει να έχει μηδέν ή παραπάνω μη-call, mechanism βέλος.
11. Κάθε κουτί πρέπει να έχει μηδέν ή ένα call βέλος.
12. Ένα control βέλος ανατροφοδότησης πρέπει να πρέπει να σχεδιάζεται ως ακολούθως:



13. Βέλη ανατροφοδότησης input πρέπει να σχεδιάζονται ως ακολούθως:

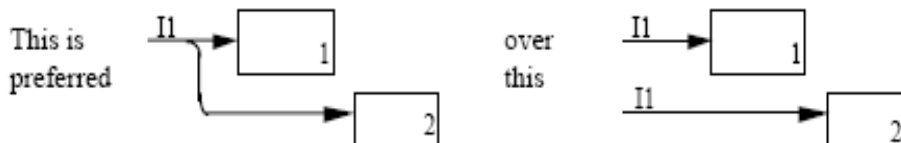


14. Βέλη ανατροφοδότησης mechanism πρέπει να σχεδιάζονται ως ακολούθως:



15. Το ελεύθερο άκρο ενός boundary βέλους, πρέπει να έχει κωδικοποίηση ICOM εξαρτώμενη από το parent box ή πρέπει να είναι tunneled τα βέλη.

16. Δύο βέλη που πάνε σε ξεχωριστά κουτιά, αλλά μεταφέρουν την ίδια πληροφορία, προτιμάται να σχεδιάζονται να διακλαδίζονται από μία πηγή και όχι σαν ξεχωριστά βέλη.



17. Τα ονόματα των κουτιών και των βελών δεν πρέπει να περιέχουν τις παρακάτω λέξεις: function, activity, process, input, control, mechanism και output.

### 3.2 WORKFLOW MODELER

Το Workflow Modeler είναι ένα πρόγραμμα της εταιρίας MetaSoftware που διατίθεται στο εμπόριο. Η χρησιμότητα του συγκεκριμένου πακέτου είναι η σχεδίαση επιχειρησιακών διαδικασιών χρησιμοποιώντας διάφορες γλώσσες μοντελοποίησης και η προσομοίωση των σχεδιασμένων διαδικασιών μέσω του Workflow Simulator. Όπως αναφέρθηκε και παραπάνω στόχος των πακέτων μοντελοποίησης επιχειρησιακών διαδικασιών, είναι η βελτίωση των υπό περιγραφή διαδικασιών. Αυτό κάνει και η προσομοίωση, με την γνώση που αποκτιέται από τα αποτελέσματα που δίνει.

Το πρόγραμμα δίνει την δυνατότητα σχεδίασης μοντέλων τριών διαφορετικών γλωσσών μοντελοποίησης, της IDEF0, της IDEF3 και της IDEF1X. Στην εργασία χρησιμοποιήθηκε η IDEF0. Στην εκκίνηση κατασκευής του μοντέλου και αφού επιλεγεί η γλώσσα στην οποία θα γίνει το μοντέλο, κατασκευάζεται το πρώτο διάγραμμα ,το A-0. Δίνεται στο κουτί το όνομα της διαδικασίας, και σχεδιάζονται τα βέλη που αυτή χρησιμοποιεί, ως εισόδους, εξόδους, βέλη ελέγχου και μηχανισμούς. Στην συνέχεια κατασκευάζεται το child diagram του πρώτου διαγράμματος που αποτελεί το διάγραμμα A0. Με το που επιλέγεται η ανάλυση (decomposition) του πρώτου κουτιού, αυτόματα το πρόγραμμα στο child diagram εμφανίζει την κωδικοποίηση ICOM. Δηλαδή ξεκινώντας την κατασκευή του διαγράμματος υπάρχουν οι είσοδοι, οι εξοδοι, τα βέλη ελέγχου και οι μηχανισμοί. Το πρώτο πράγμα που επιλέγει ο χρήστης κατά την κατασκευή του διαγράμματος είναι το πόσα κουτιά αυτό θα περιέχει. Στην συνέχεια μπορεί εύκολα να συνδέσει τα κουτιά με τα ICOMs και έπειτα αυτά μεταξύ τους. Έτσι η κατασκευή ενός διαγράμματος γίνεται εξαιρετικά απλή. Η συνέχεια της κατασκευής του μοντέλου γίνεται με τον ίδιο τρόπο αναλύοντας σε child diagrams της λειτουργίες που πρέπει. Μετά την κατασκευή του μοντέλου υπάρχει η δυνατότητα εισαγωγής μικρού ή μεγαλύτερου τύπου κώδικα, για καλύτερο προσδιορισμό των λειτουργιών του μοντέλου. Αυτά εκτός από κάποιες πληροφορίες που δίνουν με την εμφάνισή τους στον αναγνώστη του μοντέλου, κυρίως χρησιμοποιούνται κατά την προσομοίωση.

Αφού ολοκληρωθεί η κατασκευή του μοντέλου στο Workflow Modeler, μπορεί να περαστεί το μοντέλο στο Workflow Simulator, με την επιλογή “export to Workflow Simulator”. Εκεί μπορούν να γίνουν κάποιες ρυθμίσεις, όπως η εισαγωγή δεδομένων (κάποιες από τις οποίες μπορούν να γίνουν και στο Workflow Modeler). Στην συνέχεια με την επιλογή run, υλοποιείται η προσομοίωση. Τα αποτελέσματα της προσομοίωσης δίνουν μια πλειάδα στοιχείων που τα κυριότερα αφορούν κόστη, χρόνους και εμφανίσεις σε βέλη ή λειτουργίες (κουτιά).

Με την χρήση του πακέτου, αυτό που επιτυγχάνεται είναι η σχεδίαση του μοντέλου και η εξαγωγή συμπερασμάτων για αυτό μέσω της προσομοίωσης που μπορούν να οδηγήσουν σε βελτίωση του μοντέλου.

### 3.3 MICROSOFT VISIO

Για την κατασκευή διαγραμμάτων με λογικούς τελεστές έγινε η χρήση του προγράμματος της Microsoft το Visio. Το Visio, υπάρχει στο πακέτο του Office 2003 και 2007. Το Workflow Modeler δεν επέτρεπε την χρήση τελεστών, τέτοιων που απαιτούσε η κατασκευή διαγραμμάτων με εισαγωγή λαθών, χαμένων χρόνων και διάκριση των παράλληλων διαδικασιών. Τα Visio, δίνει δυνατότητα κατασκευής διαγραμμάτων για πολλές μοντελοποίησης επιχειρησιακών διαδικασιών και όχι μόνο. Επίσης διαθέτει και πλειάδα χρήσιμων σχημάτων.

### 3.4 ΟΡΟΛΟΓΙΑ

Για την καλύτερη κατανόηση της διπλωματικής από τον αναγνώστη, κρίθηκε σκόπιμο να αναπτυχθεί ένα κομμάτι με την επεξήγηση της ορολογίας που χρησιμοποιούμε.

Καταρχήν με τον όρο γλώσσα μοντελοποίησης επιχειρηματικών διαδικασιών, εννοούμε την τυποποίηση που δίνεται από κάποιον οργανισμό με κανόνες σύνταξης. Τέτοια παραδείγματα γλωσσών είναι η IDEF0, η SADT και τα EPC's. Αντίθετα ένα πακέτο που χρησιμοποιεί την εκάστοτε γλώσσα, αποτελεί ένα λογισμικό κατασκευασμένο ώστε να καλύπτει τις ανάγκες του χρήστη που θέλει να δημιουργήσει κάποιο μοντέλο επιχειρησιακής διαδικασίας.

Κάθε μοντέλο αποτελείται από μια σειρά διαγραμμάτων. Όσα απαιτούνται για την πλήρη κατανόηση της περιγραφόμενης διαδικασίας ή όσα θεωρεί ο χρήστης απαραίτητα. Δηλαδή με τον όρο μοντέλο εννοούμε ολόκληρη την απεικόνιση της διαδικασίας μέσω διαγραμμάτων. Διάγραμμα είναι κάθε ξεχωριστό διάγραμμα απεικόνισης ενός επιπέδου της διαδικασίας και αποτελείται από κουτιά και βέλη. Τα κουτιά απεικονίζουν λειτουργίες και κατά συνέπεια ως λειτουργίες αναφέρονται και σε όλη την διπλωματική.



## **ΚΕΦΑΛΑΙΟ 4**

### **ΑΠΟΤΕΛΕΣΜΑΤΑ ΒΙΒΛΙΟΓΡΑΦΙΚΗΣ ΕΡΕΥΝΑΣ**

#### **4.1 ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΤΗΣ ΓΛΩΣΣΑΣ IDEF0**

Επιλέξαμε να ασχοληθούμε με την IDEF0, γιατί είναι μια γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών πολύ σημαντική και ευρέως χρησιμοποιούμενη. Μέσω της βιβλιογραφίας επιχειρείται να δηχθεί η σημαντικότητά της.

Σε άρθρο των Anthony N. Godwin, Joseph W. Gleeson και Dean Gwillian (1988), παρουσιάζεται η IDEF0 ως μια σημαντική γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών. Η IDEF0 απευθύνεται σε συστήματα οποιουδήποτε μεγέθους ή πολυπλοκότητας και θεωρητικά δεν υπάρχει λόγος γιατί αυτό να μην συμβαίνει. Αυξάνει την αποτελεσματικότητα της σε αναλογία με το μέγεθος του υπό ανάλυση για τέσσερις βασικούς λόγους. Η ακρίβεια της top – down ανάλυσης της διαδικασίας δίνει τη δυνατότητα συνοχής σε όλα τα επίπεδα. Η ιεραρχική φύση της διαδικασίας προσαρμόζεται εύκολα σε μια φυσική ευκολοδιοίκητη υποδιαίρεση, μεγάλο προνόμιο της σε μεγάλα συστήματα ανάλυσης εργασιών. Η ICOM συμβάσεις οδηγούν σε ευκολία κατανόησης. Τέλος, η ικανότητα της IDEF0 να προσδιορίζει και να ενώνει διαφορετικές οπτικές του ίδιου συστήματος, επιτρέπει την κατασκευή πολυδιάστατων μοντέλων.

Σε άρθρο των Adrien Presley και Donald H. Liles, αναφέρεται η τυπικότητα της μεθόδου που απολαμβάνει ο χρήστης, ως το μεγαλύτερο προσόν της. Επίσης δυνατό σημείο της μεθόδου αποτελεί η ακρίβεια της περιγραφής της διαδικασίας που επιτυγχάνει, χάρη στην λεπτομερή ιεραρχική ανάλυση. Η δυνατότητα εισαγωγής σχολείων στο διάγραμμα, απαλλάσσει τον αναγνώστη από ανάγκη για πιθανές συνεντεύξεις σε εργαζόμενους ή γενικότερες απορίες Άλλο ένα σημαντικό στοιχείο της μεθόδου είναι ότι αποτελεί μια ευρέως αποδεκτή γλώσσα και είναι η πλέον χρησιμοποιούμενη για μοντελοποίηση επιχειρησιακών διαδικασιών. Επίσης σε συνεργασία με την IDEF0 είναι δυνατόν να χρησιμοποιηθούν αυτοματοποιημένα

εργαλεία (ηλεκτρονικοί υπολογιστές). Εξαιτίας του ότι στην IDEF0 δεν έχουμε συμβολισμό ακολουθίας μέσω των βελών είναι δυνατή η χρήση βελών ανατροφοδότησης.

Ο Yatin S. Karpe το 2006 υπέδειξε την IDEF0 ως μια από τις σημαντικότερες γλώσσες μοντελοποίησης επιχειρησιακών διαδικασιών για τους παρακάτω λόγους. Παρέχει μια εξαιρετική ερμηνεία της AS-IS κατάστασης (ακριβής περιγραφή μιας διαδικασίας χωρίς αλλαγές). Ευρέως αποδεκτή μέθοδος για επικοινωνία και παρουσίαση αποτελεσμάτων. Δυνατότητα για βελτίωση των περιγραφόμενων διαδικασιών. Οι σχέσεις μεταξύ δεδομένων και περιττών λειτουργιών αποκαλύπτεται. Το AS-IS μοντέλο μπορεί να αποτελέσει πρόδρομο του TO-BE μοντέλου (πως πρέπει να γίνει η διαδικασία για να είναι πιο αποτελεσματική).

Ο R. J. Mayer το 1992, έγραψε ότι η υψηλή δημοτικότητα της μεθόδου, οφείλεται στο ότι επικεντρώνεται στην βελτίωση της ανθρώπινης επικοινωνίας. Με το πέρασμα των χρόνων, δημιουργήθηκαν πολλά αυτοματοποιημένα εργαλεία που συνεργάζονται με την μέθοδο.

Τέλος οι K. J. Rogers και L. Whitman το 1998, υπέδειξαν ότι προσόν της IDEF0 είναι το ότι επικεντρώνεται στις λειτουργίες και όχι στα τμήματα και στο προσωπικό. Επίσης η top-down ανάλυση παρέχει μια ολιστική άποψη για την περιγραφόμενη διαδικασία σε οποιοδήποτε επιχειρησιακό σύστημα.

Αυτό που είναι εύκολο να παρατηρηθεί στην σύντομη βιβλιογραφική ανασκόπηση που προηγήθηκε, είναι ότι όλοι αναφέρουν το πόσο διαδεδομένη και ευρέως χρησιμοποιούμενη είναι η γλώσσα. Ένας παράγοντας που συνηγορεί ακόμα σε αυτό, είναι η διάρκεια στον χρόνο που παρατηρείται από την βιβλιογραφική ανασκόπηση. Τα άρθρα που παρατίθενται ξεκινάνε από το 1988 και φτάνουν μέχρι το 2006, ουσιαστικά δηλαδή μέχρι σήμερα. Αυτή η διάρκεια στον χρόνο της μεθόδου, μαζί με όλα τα παραπάνω συνηγορούν στην σημαντικότητά της και στο γιατί επιλέξαμε την IDEF0, για να ασχοληθούμε.

Τέλος πρέπει να αναφέρουμε ότι η IDEF0, είναι η πλέον κατάλληλη γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών για την περιγραφή κατασκευαστικών

διαδικασιών. Μια τέτοια είναι και το παράδειγμα που θα χρησιμοποιηθεί κατά την παρουσίαση του προβλήματος.

#### 4.2 ΠΩΣ ΠΡΟΚΥΠΤΕΙ Η ΑΝΑΓΚΗ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΗΣ IDEF0

Πέρα από τα πολλά και αδιαμφισβήτητα προσόντα τις μεθόδου, υπάρχουν και μερικά μειονεκτήματα. Κάποια από αυτά θα δούμε στην συνέχεια.

Στο άρθρο που αναφέραμε παραπάνω των Anthony N. Godwin, Joseph W. Gleeson και Dean Gwillian (1988), αναφέρεται ότι τα μοντέλα που προκύπτουν από την IDEF0, μπορούν εύκολα να μετατραπούν σε αντικείμενα μεγάλου μεγέθους και πολυπλοκότητας. Άρα ένας στόχος της παρούσας διπλωματικής θα είναι η απλοποίηση των διαγραμμάτων της IDEF0 ώστε να γίνουν πιο ευανάγνωστα.

Σε άρθρο που δημοσιεύτηκε τον Ιανουάριο του 2006, οι J.C. Hernandez-Matias, A. Viran, A. Hidalgo, J. Rios, βλέπουμε ότι ένα πρόβλημα της IDEF0 είναι ότι δεν εμφανίζονται τα πιθανά λάθη που μπορούν να συμβούν σε μια διαδικασία και έτσι τα μοντέλα δεν έχουν πρακτική εφαρμογή. Έτσι από το μοντέλο δεν μπορούν να εξαχθούν ασφαλή συμπεράσματα για τους ρυθμούς παραγωγής και τα κόστη. Πάνω σε αυτό το σημείο θα βασιστούμε για να αναπτύξουμε ένα μέρος της μεθοδολογίας μας.

Κάποια από τα μειονεκτήματα που θα αντιμετωπίσουμε στην συνέχεια της εργασίας είναι μερικά από αυτά που συναντάμε στην βιβλιογραφία, ενώ κάποια άλλα είναι μειονεκτήματα που διαπιστώθηκαν κατά την υλοποίηση της εργασίας.

#### 4.3 ΠΡΟΗΓΟΥΜΕΝΕΣ ΠΡΟΣΠΑΘΕΙΕΣ ΒΕΛΤΙΩΣΗΣ

Στο άρθρο του Jan Mendling, έχουμε μια πρώτη εκτίμηση για τις συνέπειες των λαθών, που μπορεί να συμβούν κατά την διάρκεια μιας επιχειρησιακής διαδικασίας. Όσο πιο αργά εντοπίζονται τα λάθη τόσο πιο πολύ δουλειά απαιτείται μετά και η περισσότερη σχεδιαστική δουλειά έχει πάει στραφεί. Στην φάση σχεδιασμού τα μοντέλα διαδικασιών δημιουργούνται με ημι-επίσημη γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών, αντίθετα επίσημη γλώσσα

χρησιμοποιείται κατά την εγκατάσταση. Αυτό το πρόβλημα αναφέρεται ως το κενό μεταξύ σχεδιασμού μιας επιχειρησιακής διαδικασίας και της φάσης εγκατάστασης.

Για την έγκαιρη διαπίστωση των λαθών που μπορεί να γίνουν κατά την μοντελοποίηση επιχειρησιακών διαδικασιών προτείνεται η εφαρμογή του πληροφοριακή μοντελοποίηση διαδικασιών. Παρουσιάζεται στην συνέχεια αποτελούμενη από οκτώ βήματα. Στο πρώτο βήμα γίνεται συλλογή πληροφοριών και αντικειμένων σχετικών με το θέμα. Στο δεύτερο βήμα αυτές οι είσοδοι καταγράφονται σε κείμενο που λειτουργεί ως ενοποιημένη τυποποίηση. Αυτό το κείμενο επαναδιοργανώνεται σύμφωνα με κάποια γενική γραμμή που λέει πως εκφράζονται κάποια γεγονότα που παρέχουν μια μη επίσημη προδιαγραφή. Στο πέμπτο βήμα αυτή η μορφή προδιαγραφής σχεδιάζεται σε μία γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών (π.χ. IDEF0) ώστε να κατασκευαστεί το μοντέλο της διαδικασίας. Στο βήμα έξι αυτό το μοντέλο επικυρώνεται για την γενική του ορθότητα. Στην συνέχεια, στο βήμα επτά παραφράζεται στην κανονική γλώσσα για να ώστε να αξιολογηθεί σε σύγκριση με τις προδιαγραφές (βήμα οκτώ).

Στη συνέχεια παρουσιάζονται κάποιες άλλες προσπάθειες παράκαμψης των μειονεκτημάτων της IDEF0.

#### 4.3.1 Χρήση IDEF0, IDEF3 και DFD

Σε αυτή την προσπάθεια των Hui Shen, Brian Wall, Michal Zaremba, Yuliu Chen και Jim Browne (Φεβρουάριος 2004), θέλουμε να εκμεταλλευτούμε τα πλεονεκτήματα των τριών μεθόδων και να παρακάμψουμε τα μειονεκτήματά τους, σε μια συνδυαστική χρήση των τριών.

Η μέθοδος IDEF0 μόνο περιγράφει τις λειτουργίες και τις πληροφορίες σύνδεσης (ICOM) ανάμεσα σε αυτές, ενώ οι λογικές σχέσεις και η συνέχεια ανάμεσα σε διαφορετικές λειτουργίες δεν είναι ξεκάθαρες. Η IDEF3 είναι λιγότερο αυστηρή μέθοδος από την IDEF0 και αυτό κάνει την μοντελοποίηση πιο εύκολη, αλλά δεν είναι εξίσου τυποποιημένη. Η DFD (Data Flow Diagrams) λειτουργεί σαν δίκτυο από λειτουργίες συνδεδεμένες η μία με την άλλη με κανάλια πληροφοριών ή αντικειμένων. Πλεονέκτημα της DFD είναι η ξεκάθαρη περιγραφή ροής

πληροφοριών, από την πηγή στον προορισμό. Η αυστηρότητα της σύνταξης και των σημασιολογικών κανόνων είναι ανάμεσα στην IDEF0 και την IDEF3.

Αυτή η συνδυαστική μέθοδος ακολουθεί πέντε βήματα που περιγράφονται αμέσως. Στο πρώτο στάδιο για την δημιουργία ενός IT σχεδίου για μια εταιρία, δημιουργούμε ένα απλό και γενικό top-down διάγραμμα IDEF0, που περιγράφει το λειτουργικό πλαίσιο του μοντέλου. Στο δεύτερο βήμα, έχουμε μία από κάτω προς τα πάνω προσέγγιση, με στόχο την αποσαφήνιση των διαδικασιών και της ροής των πληροφοριών στο κατώτατο διάγραμμα. Εδώ χρησιμοποιείται η IDEF3 και η DFD. Στο τρίτο στάδιο, βασισμένοι στα αποτελέσματα του προηγούμενου σταδίου, μπορούμε να εισάγουμε στο A-0 διάγραμμα της IDEF0 την κωδικοποίηση ICOM (τα βέλη). Επίσης αναπροσαρμόζεται η ιεραρχική σχέση μεταξύ, διαγραμμάτων και λειτουργιών. Στο τέταρτο βήμα, χρησιμοποιούμε την IDEF0 με σκοπό τη δημιουργία μιας γενικής λειτουργίας που αντανάκλα τις απαιτήσεις της εταιρίας (ένας TO-BE σχεδιασμός). Τέλος στο πέμπτο βήμα, το TO-BE σενάριο του προηγούμενου βήματος δίνεται στον τελικό χρήστη και παίρνουμε μια ανατροφοδότηση, την οποία χρησιμοποιούμε για να κάνουμε τις απαραίτητες διορθώσεις στο μοντέλο, αναλύοντας τις σημαντικές λειτουργίες με IDEF3 ή DFD διαγράμματα. Έτσι έχουμε μία ομοφωνία μεταξύ μοντελοποιού και τελικού χρήστη, φτάνοντας σε ένα ολοκληρωμένο σετ TO-BE διαγραμμάτων αποδεκτό από την εταιρία.

#### 4.3.2 Άλλες μέθοδοι

Είναι δύσκολο να συγκρίνουμε ένα στατικό μοντέλο (όπως αυτά που προκύπτουν από την IDEF) με μία φυσική ροή, από τη στιγμή που στην IDEF δεν αναπαριστάται ο χρόνος.

Οι ενέργειες σε ένα IDEF μοντέλο, είναι σε ένα σχετικά υψηλό επίπεδο αφηρημένες, κάνοντας δύσκολη την σύνδεση με συγκεκριμένη ποσοτική πληροφορία για την διαδικασία μεταβλητής ενδιαφέροντος.

Με την χρήση της IDEF0 και διαδικασιών προσομοίωσης μπορούμε να δημιουργήσουμε μεθόδους για την βελτίωση ενός συστήματος παραγωγής.

Η αληθινή αξία της IDEF0 προκύπτει όταν δημιουργούνται μηχανισμοί για επεξεργασία ποσοτικών πληροφοριών (Analysis Based Costing, ABC). Μία σύνδεση της IDEF0 με μια μέθοδο accounting cost, δηλαδή παραγόντων κόστους και χρόνο μας δίνει έναν βέλτιστο σχεδιασμό παραγωγής. Επίσης μπορεί να υπάρξει συνεργασία της IDEF0 με το Excel δίνοντας πολύ καλά αποτελέσματα.

Άλλοι τρόποι για να ξεπεραστούν οι περιορισμοί της IDEF είναι η χρήση τεχνικών όπως τα δίκτυα Petri και ασαφούς λογικής (Fuzzy Logic). Κάπως έτσι αναπτύχθηκε και μια επέκταση της IDEF1X (IDEF1 extended) που αναπαριστά και ασαφείς πληροφορίες, από τους Ma και Zahng (2002). Μια ακόμα μέθοδος είναι η χρήση της IDEF0 σε συνεργασία με την IDEF3 και τα δίκτυα Petri που χρησιμοποίησαν οι Bosil, Giaglis και Hlupic (2000).

Οι παραπάνω μέθοδοι επιτρέπουν την μετατροπή των φαινομενικά στατικών μοντέλων της IDEF, σε δυναμικά και η συγκομιδή χρήσιμων συμπερασμάτων για μια γραμμή παραγωγής ή για οποιαδήποτε επιχειρηματική διαδικασία.

#### 4.4 Η ΣΗΜΑΣΙΑ ΤΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

Κατά τους Mahmudi και Tavakkoli για να επαναπροσδιορίσουν τις διαδικασίες τους οι εταιρίες πρέπει, πρώτα να είναι γνωστός ο μηχανισμός του λειτουργικού συστήματος και έπειτα να μετρηθεί η αποδοτικότητά του για να αναγνωριστούν οι κρίσιμες αλλαγές που πρέπει να γίνουν. Οι εταιρίες θα πρέπει να αξιολογούν την αποτελεσματικότητα των αλλαγών και να προσαρμόζουν ανάλογα το πλάνο τους (αν είναι αναγκαίο). Τα προαναφερθέντα μπορούν να υλοποιηθούν με την μοντελοποίηση.

Η μοντελοποίηση παρέχει την δυνατότητα ανάλυσης μια επιχειρηματικής διαδικασίας στους παρακάτω τομείς:

- Προσδιορισμός καθυστερήσεων και διαρροών.
- Σχεδιασμός επανελέγχου της διαδικασίας με στόχο τη βελτίωσή της απόδοσής της.
- Επιλογή καλύτερης σχεδιασμένης διαδικασίας με σκοπό την παροχή καλύτερων αποτελεσμάτων.
- Υπολογισμός κόστους.
- Μέτρηση απόδοσης της νέα διαδικασίας.

Η μοντελοποίηση είναι διαθέσιμη σε δύο κατηγορίες, την αναλυτική και την προσομοίωση. Χαρακτηρίζουμε όλες τις κατηγορίες μοντελοποίησης (εκτός της προσομοίωσης) ως αναλυτικές. Τα αναλυτικά εργαλεία είναι ικανά, μόνο να μαντέψουν τα αποτελέσματα, αντίθετα με την προσομοίωση μπορεί να δοθεί ένας σωστός και αξιόπιστος υπολογισμός των αποτελεσμάτων.

Τα αναλυτικά εργαλεία δεν έχουν την δυνατότητα να βοηθήσουν τους αναλυτές στους παρακάτω τομείς:

- Χρησιμοποίηση μεταβλητών χρόνου σε διαδικασία.
- Διαδικασίες που η κατάστασή τους αλλάζει με την πάροδο του χρόνου.
- Μη γραμμικές σχέσεις μεταξύ στοιχείων της διαδικασίας.
- Τυχαιότητα της πραγματικής διαδικασίας.
- Μη επιθυμητά γεγονότα σε επιχειρησιακό περιβάλλον.

Συνήθως ένα σχέδιο επιχειρησιακής διαδικασίας έχει έναν, κάποιους ή όλους από τους παρακάτω στόχους:

- Αύξηση του επιπέδου υπηρεσιών.
- Μείωση του συνολικού χρόνου της διαδικασίας.
- Μείωση του χρόνου αναμονής.
- Μείωση του κόστους της διαδικασίας.

Ο Youngblood (1994) έκανε μια λίστα με τριάντα δύο τρόπους για να επιτευχθούν οι παραπάνω στόχοι. Οι περισσότεροι είναι βασικές αρχές της μηχανικής βιομηχανίας (industrial engineering), που εφαρμόστηκαν σε κατασκευαστικές βιομηχανίες για πολλές δεκαετίες. Μερικές από αυτές εφαρμόζονται στις επιχειρησιακές διαδικασίες και είναι:

- Χρήση επιτυχημένων ενεργειών (λύσεων).
- Απόκλιση πολλαπλών επιθεωρήσεων και αποδοχών.
- Μείωση εργασιών σε ομάδες.
- Παράλληλες διαδικασίες.
- Απόκλιση κίνησης κατά την εργασία.
- Οργάνωση πολύ-λειτουργικών ομάδων.



- Μείωση μη αποδοτικών ενεργειών.
- Εγκατάσταση pulling system για τις απαιτήσεις των πελατών.

Οι παραπάνω αρχές απαντάνε στο ερώτημα «τι πρέπει να γίνει;». Όμως η διοίκηση επιχειρησιακών διαδικασιών περιλαμβάνει αλλαγές στους ανθρώπους στις διαδικασίες και στην τεχνολογία με το πέρασμα του χρόνου. Οι αλληλεπιδράσεις μεταξύ των ανθρώπων, των διαδικασιών και της τεχνολογίας, μπορούν να δημιουργήσουν μια πλειάδα σεναρίων. Αυτά τα σενάρια είναι αδύνατον να κατανοηθούν και να αξιολογηθούν με τα αναλυτικά εργαλεία. Αντίθετα με την προσομοίωση μπορούμε να πάρουμε μια δυναμική εικόνα των μοντέλων.

Κάποιοι ερευνητές υποστηρίζουν ότι η προσομοίωση είναι η μόνη τεχνική που δίνει τόσο καλά αποτελέσματα, μιας και επιχειρησιακές διαδικασίες είναι πολυσύνθετες (Fathee, 1998). Αυτό ισχύει αφού:

- Πολλές επιχειρησιακές διαδικασίες είναι απροσδιόριστες και περιέχουν τυχαίες μεταβλητές.
- Οι ενέργειες και οι πηγές είναι τα κύρια στοιχεία, σε μια επιχειρησιακή διαδικασία, που έχουν αλληλεπιδράσεις.
- Οι επιχειρησιακές διαδικασίες οργανισμών επηρεάζουν η μία την άλλη και αλλάζουν από αίτια εκτός αυτών των οργανισμών.

### **ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΡΓΑΣΙΑΣ**

#### **5.1 ΤΡΟΠΟΣ ΥΛΟΠΟΙΗΣΗΣ**

Για την επίλυση και αντιμετώπιση των προβλημάτων της IDEF0, έπρεπε να δημιουργηθεί ένα μοντέλο. Ο λόγος είναι προφανής, για να μπορέσουν να εφαρμοστούν οι βελτιώσεις που θα γίνουν σε ένα πραγματικό παράδειγμα.

Το μοντέλο που επιλέχθηκε είναι αυτό που εμφανίζεται στο βιβλίο SADT των David A. Marca και Clement L. McGowan. Πρόκειται για μια διαδικασία που λαμβάνει χώρα σε ένα μηχανουργείο και σκοπό έχει την κατασκευή ενός κομματιού από μία εργαλειομηχανή (τόρνο). Το βιβλίο SADT αποτελεί ένα εγχειρίδιο της μεθόδου SADT, που συνθέτεται μέσα από μια σειρά παρατηρήσεων που έχουν γίνει κατά την χρήση της μεθόδου. Η μέθοδος αυτή αποτέλεσε τον προπομπό της IDEF0, καθώς πάνω στην SADT βασίστηκαν οι δημιουργοί της.

Στην διαδικασία που παρουσιάζεται, το μοντέλο παίρνει σαν εισόδους τις πρώτες ύλες, τις διαθέσιμες μηχανές και τα εργαλεία και τις πληροφορίες που απαιτούνται για την κατασκευή του κομματιού. Σαν εξόδους το μοντέλο δίνει το παραγόμενο κομμάτι και τον χρόνο που υπολογίζεται να διαρκέσει η κατασκευή του κομματιού. Το μοντέλο αποτελείται από έξι διαγράμματα με ιεραρχική ανάλυση από πάνω προς τα κάτω (top – down decomposition). Η λογική της διαδικασίας είναι ότι η πρώτη ύλη που μπαίνει στο σύστημα, ελέγχεται, διαπιστώνονται οι κατεργασίες που πρέπει να γίνουν και στέλνονται σαν πληροφορίες για την κατασκευή του επιθυμητού κομματιού. Το κομμάτι κατασκευάζεται σύμφωνα με τις υπάρχουσες προδιαγραφές και στη συνέχεια αφού υπόκειται σε έλεγχο ή γίνεται αποδεκτό και βγαίνει από το σύστημα ή δεν γίνεται αποδεκτό και στέλνεται πίσω για εκ νέου κατεργασία.

Για να το σχεδιαστεί το μοντέλο σύμφωνα με την γλώσσα IDEF0, χρησιμοποιήθηκε το πακέτο της MetaSoftware, Workflow Modeler. Εκεί σχεδιάστηκαν όλα τα διαγράμματα του παραδείγματος, από το A-0 διάγραμμα (το

πρώτο της γλώσσας IDEF0) ως το κατώτερο. Στην συνέχεια έγιναν οι απαιτούμενες αλλαγές στο μοντέλο ώστε αυτό να δίνει ρεαλιστικά αποτελέσματα κατά την προσομοίωση. Επίσης στο μοντέλο εισήχθηκαν κάποια στοιχεία κώδικα, με σκοπό την υλοποίηση διαφόρων επιλογών ή δραστηριοτήτων που έπρεπε να γίνουν στο μοντέλο.

Τέλος για την χρήση κάποιας συμβολογίας που δεν υποστηρίζεται από την IDEF0, χρησιμοποιήθηκε το πρόγραμμα των Windows, Visio. Εκεί με βάση τα αρχικά διαγράμματα που κατασκευάστηκαν στο Workflow Modeler, έγινε τροποποίησή τους για την εισαγωγή των νέων συμβόλων

## 5.2 ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΗΣ IDEF0

Οι καινοτομίες που εισάγει αυτή η εργασία, έχουν να κάνουν με «εκφράσεις», που είτε η γλώσσα μοντελοποίησης IDEF0 δεν υποστηρίζει είτε δεν είναι ευδιάκριτες σε αυτή.

Τα σφάλματα που μπορεί να εμφανιστούν κατά την διάρκεια μίας διαδικασίας, είναι ένα παράδειγμα που δεν μοντελοποιείται μέσω της IDEF0. Αυτό αντιμετωπίζεται σε αυτήν την εργασία μέσω της χρήσης λογικών πυλών. Οι λογικές πύλες είναι ένα εργαλείο που χρησιμοποιείται πολύ συχνά σε διαγράμματα ροής, αλλά και σε άλλα πακέτα μοντελοποίησης επιχειρησιακών διαδικασιών όπως είναι το ARIS.

Στην παρούσα εργασία προτείνεται μια μεθοδολογία, που η εφαρμογή της στα μοντέλα της IDEF0, δίνει την δυνατότητα συχνών ελέγχων για πιθανά σφάλματα. Αποσαφηνίζεται και γίνεται ξεκάθαρη η λειτουργία των παράλληλων διαδικασιών. Παράλληλες διαδικασίες ονομάζονται δύο διαδικασίες που λαμβάνουν χώρα ταυτόχρονα. Αυτό σημαίνει ότι μπορεί να ξεκινάνε μαζί και να μην ολοκληρώνονται την ίδια στιγμή, να μην ξεκινάνε μαζί αλλά να ολοκληρώνονται ταυτόχρονα ή η μία να ξεκινάει πριν αλλά να τελειώνει μετά την άλλη. Κύριο χαρακτηριστικό των παράλληλων διαδικασιών είναι ότι για να γίνει μια επόμενη λειτουργία, πρέπει να έχουν ολοκληρωθεί και οι δύο παράλληλες. Στα διαγράμματα IDEF0 δεν γίνονται εμφανή πιθανά σημεία που μπορεί μέσα στο μοντέλο να συμβεί μια καθυστέρηση.

Στην εργασία φανερώνονται τα σημεία στα οποία κάποιο αποτέλεσμα μπορεί να εμφανίσει καθυστέρηση. Επίσης ξεκαθαρίζεται το πόσα κομμάτια περιέχει κάθε βέλος ενός μοντέλου, πράγμα που στην IDEF0 δεν γίνεται φανερό.

### 5.3 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΜΟΝΤΕΛΟΥ

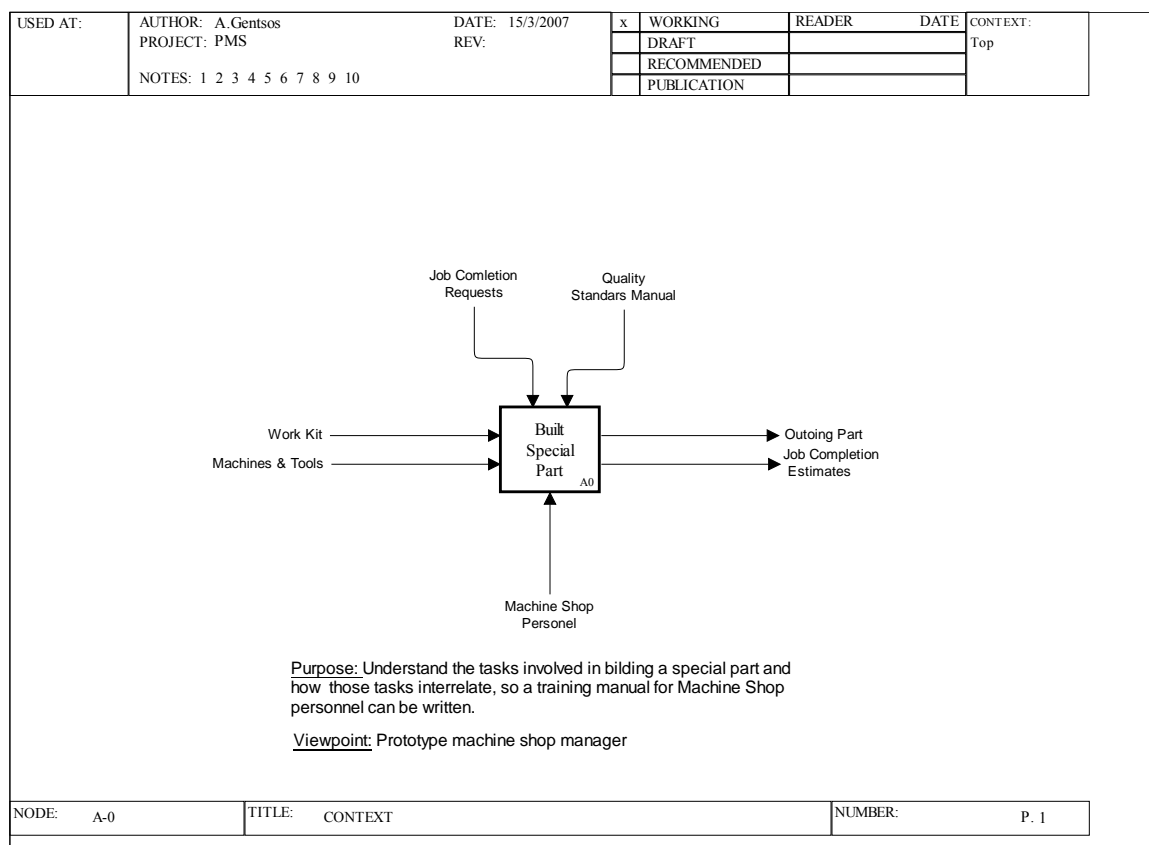
Η διαδικασία που επιλέχθηκε να μοντελοποιηθεί μέσω της μεθόδου IDEF0 και του λογισμικού της MetaSoftware, το Workflow Modeler, είναι η κατασκευή εξαρτημάτων σε ένα μηχανουργείο από μια εργαλειομηχανή (τόρνος).

Η διαδικασία δέχεται σαν εισόδους τις πρώτες ύλες και το τι κατεργασίες πρέπει να συμβούν σε αυτές, κάποιες μηχανές (από τις οποίες επιλέγεται αυτή που είναι καταλληλότερη για την κατεργασία) και τα εργαλεία που απαιτούνται για την κατασκευή του κομματιού. Σαν εξόδους δίνει το χρόνο που υπολογίζεται να διαρκέσει η κατεργασία και προφανώς το έτοιμο εξάρτημα. Η διαδικασία που ακολουθείται, σε γενικές γραμμές, είναι ότι εξετάζονται οι ενέργειες που πρέπει να γίνουν για την κατασκευή του εξαρτήματος, στέλνονται ως πληροφορίες (control βέλη) στο κατασκευαστικό τμήμα. Στη συνέχεια εκεί έχουμε την κατασκευή του κομματιού. Το κατασκεύασμα που προκύπτει τέλος υπόκειται σε έλεγχο και αν είναι καλό γίνεται αποδεκτό και εξέρχεται από το σύστημα, αλλιώς στέλνεται για εκ νέου κατεργασία.

Στη συνέχεια ακολουθεί η λεπτομερής περιγραφή του μοντέλου, για κάθε επίπεδο, όπως αυτό αναλύθηκε.

## Λεπτομερής περιγραφή του μοντέλου

### 5.3.1 Διάγραμμα A-0



*Σχήμα 5.1: Build Special Part, A-0 διάγραμμα [1].*

Στο πρώτο διάγραμμα του μοντέλου μπορούμε να δούμε όλες τις ροές πρώτου επιπέδου της διαδικασίας. Ως εισόδους του μοντέλου έχουμε το Work Kit, που όπως θα φανεί και παρακάτω αποτελείται από το Job Order and Blueprint (η σειρά των εργασιών και σχέδια), το Time Card (η κάρτα χρονικής διάρκειας) και τα Raw Materials (οι πρώτες ύλες). Οι παραπάνω εισοδοι, μετατρέπονται σε εξόδους οι οποίες είναι το Outgoing Part (το εξάρτημα που παράγεται) και το Job Completion Estimates (η εκτίμηση για την ολοκλήρωση της εργασίας). Αυτή η μετατροπή των εισόδων σε εξόδους γίνεται των Control βελών που είναι το Job Completion Requests (οι απαιτήσεις για την ορθή ολοκλήρωση της εργασίας) και το Quality Standards Manual (το εγχειρίδιο των ποιοτικών απαιτήσεων). Τέλος μηχανισμό της διαδικασίας αποτελεί του προσωπικό του εργαστηρίου (Machine Shop Personnel).

### 5.3.2 Διάγραμμα A0



Το Manage Job με τις πληροφορίες που παίρνει από το Work Kit, στέλνει μέσω control βελών πληροφορίες τόσο στο Complete Work Kit όσο και στο Inspect

Job για να μπορέσουν να διεκπεραιώσουν τις εργασίες τους. Επίσης το εξάρτημα που γίνεται αποδεκτό από Inspect Job ξανά περνάει από έλεγχο για να επιβεβαιωθεί ότι είναι σε θέση να εξέλθει από το σύστημα.

Το Complete Work Kit, όπως ειπώθηκε και πιο πριν, είναι το σημείο στο οποίο κατασκευάζεται το εξάρτημα. Παίρνει τις πρώτες ύλες (Raw Materials) ή τα απορριφθέντα εξαρτήματα (Rejected Job) και τις μετατρέπει στο επιθυμητό εξάρτημα, χρησιμοποιούνται επίσης η εργαλειομηχανή και τα εργαλεία που και αυτά εισέρχονται στη λειτουργία. Από το Complete Work Kit εξέρχεται και μια πληροφόρηση προς το Manage Job για το σε πιο σημείο βρίσκεται η κατεργασία.

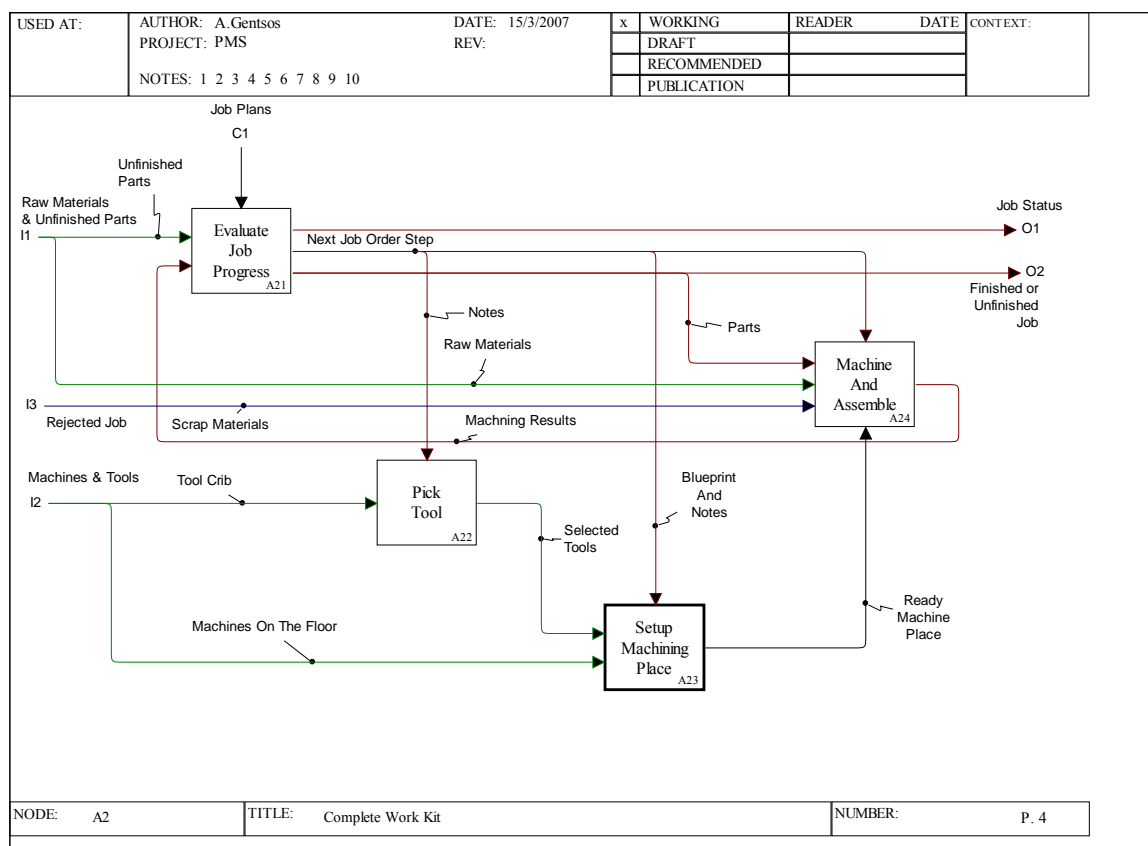




στάδια που πρέπει να ακολουθηθούν κατά την κατασκευή (Machining Steps), την κατάσταση στην οποία βρίσκεται η κατεργασία (Job Status, έρχεται από το A21 Evaluate Job Progress) και την ανάδραση από το Monitor Work που το πληροφορεί για την πρόοδο της εργασίας. Έτσι μέσω των παραπάνω είναι δυνατό να υπολογιστεί ο χρόνος ολοκλήρωσης της εργασίας και να δοθεί σαν πληροφορία προς την έξοδο του μοντέλου, αλλά και στο Monitor Work.

Χρησιμοποιώντας τα τρία control βέλη, με τα σχέδια (Job Plans), τον χρονικό υπολογισμό (Time Estimates) και τις απαιτήσεις για την ολοκλήρωση της εργασίας (Job Completion Estimates), το Monitor Work παρέχει την ανάδραση που αναφέραμε παραπάνω, με τα πιθανά προβλήματα (Problems) και με την μέχρι στιγμής πρόοδο (Progress). Επίσης μέσω πάλι control βέλους ενημερώνει το Complete Job ότι όλα τα βήμα τα έχουν ολοκληρωθεί (All Steps Done) (όταν αυτό είναι απαραίτητο), για να μπορέσει να προωθήσει το εξάρτημα εκτός του συστήματος. Για την επιβεβαίωση αυτής της εξόδου, εκτός φυσικά της παρουσίας του ίδιου του εξαρτήματος (Tagged Part), απαιτείται και το Approval Seal που στέλνει το A34 (Accept or Reject βλ. διάγραμμα A3) ταυτόχρονα με το αποδεχόμενο κομμάτι.

### 5.3.4 Διάγραμμα A2

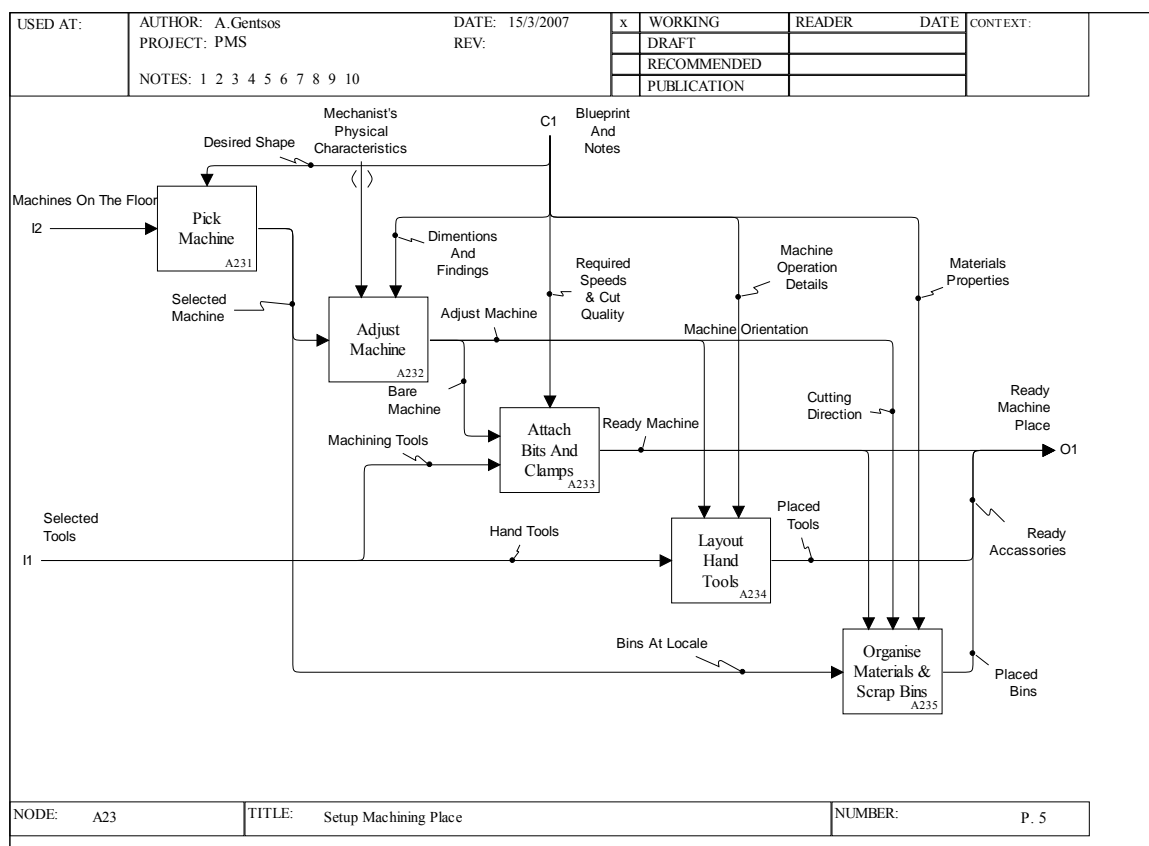


Σχήμα 5.4: Complete Work Kit, διάγραμμα A2 [4].

Παρατηρούμε ότι οι λειτουργίες A22 (Pick Tool) και A23 (Setup Machining Place) κατασκευάζουν ουσιαστικά τον μηχανισμό για το Machine and Assemble. Δηλαδή η λειτουργία τους είναι να ρυθμίσουν την εργαλειομηχανή με βάση τις πληροφορίες που τους στέλνουν τα control βέλη από το Evaluate Job Progress.

Η πρώτη λειτουργία του Machine and Assemble είναι να πάρει τις πρώτες ύλες ή τα απορριφθέντα εξαρτήματα και να τα στείλει για έλεγχο στο Evaluate Job Progress. Έπειτα το Evaluate Job Progress στέλνει τις απαραίτητες πληροφορίες ώστε να ρυθμιστεί η εργαλειομηχανή, ως προς το τι κατεργασία πρέπει να κάνει. Πληροφορίες στέλνει και στο Machine and Assemble όπως επίσης και τις πρώτες ύλες (ως Parts). Στη συνέχεια γίνεται η κατεργασία και το αποτέλεσμα στέλνεται πάλι στο Evaluate Job Progress, από όπου και βγαίνει στο σύστημα. Τέλος πρέπει να αναφερθεί ότι για κάθε είσοδο στο Evaluate Job Progress παρέχεται μια πληροφορία, για το σε τι κατάσταση βρίσκεται το επεξεργαζόμενο κομμάτι, προς το Manage Job.

### 5.3.5 Διάγραμμα A23



Σχήμα 5.5: Setup Machining Place, Διάγραμμα A23[5].

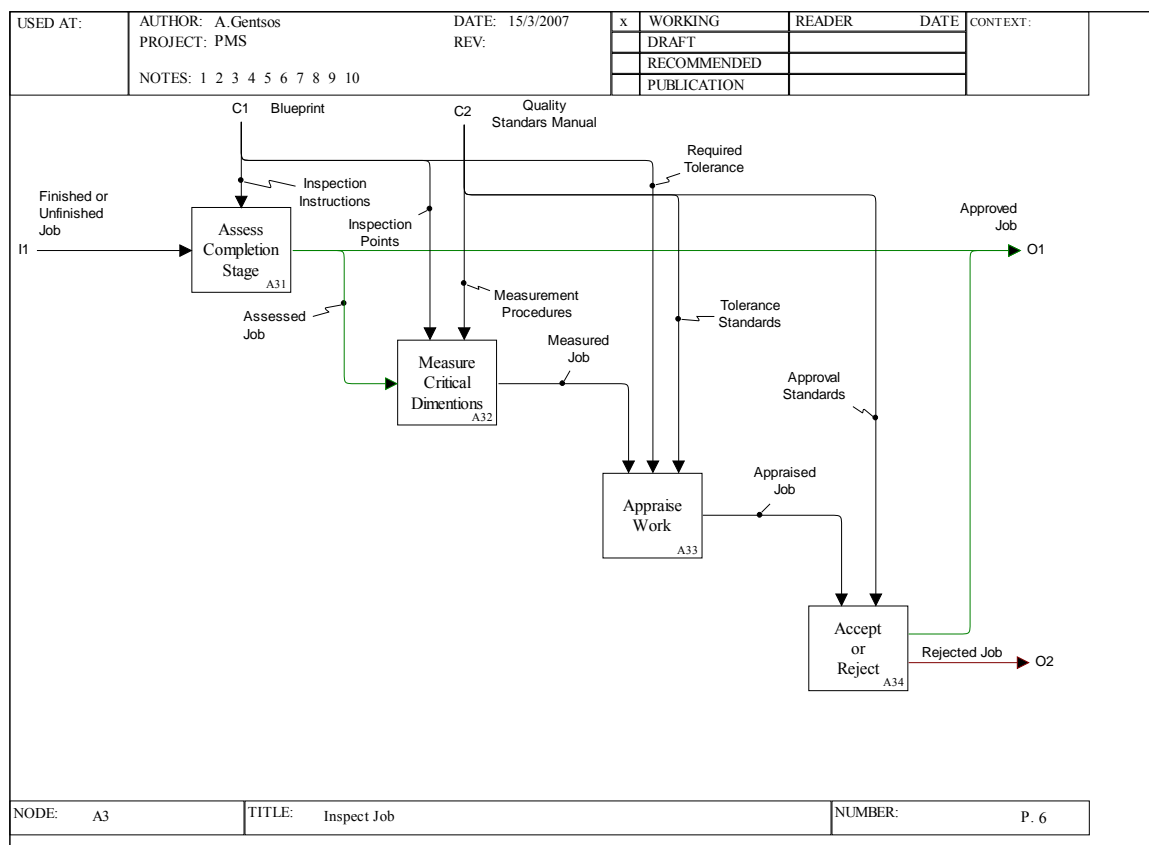
Αφού έχει επιλεγεί το κατάλληλο εργαλείο (προφανώς το κοπτικό που θα χρησιμοποιήσει ο τórνος) από το Pick Tools (A22, στο διάγραμμα A2), στη συνέχεια στο child diagram του Setup Machining Place, φαίνεται πως η εργαλειομηχανή ρυθμίζεται προκειμένου να είναι σε θέση να κατεργαστεί το κομμάτι. Σαν είσοδοι στο διάγραμμα εισάγονται τα επιλεγμένα εργαλεία και η μηχανές της γραμμής παραγωγής.

Με βάση τα σχέδια (Blueprint) και πιο συγκεκριμένα την επιθυμητή μορφή του αντικειμένου (Desired Shape, control βέλος), το Pick Machine επιλέγει την μηχανή που είναι η καταλληλότερη από αυτές που δέχεται σαν είσοδο. Η επιλεγμένη μηχανή (μπαίνει σαν είσοδος στο Adjust Machine) ρυθμίζεται σύμφωνα με τα σχέδια και πιο συγκεκριμένα τις διαστάσεις και τα υλικά που θα χρησιμοποιηθούν (Dimensions and Findings) και τα φυσικά χαρακτηριστικά του χειριστή (Machinist's Physical Characteristics). Η ρυθμισμένη μηχανή και τα εργαλεία που χρησιμοποιεί η μηχανή, είναι οι είσοδοι για το Attach Bits and Clamps. Μέσω των σημειώσεων

(Notes) η μηχανή είναι ρυθμισμένη και έτοιμη για χρήση (Ready Machine). Τα υπόλοιπα εργαλεία που έχουν επιλεγεί και αφορούν άλλες χρήσεις (Hand Tools) τοποθετούνται (Placed Tools), μέσω των control βελών που αναπαριστούν πληροφορίες για την ρυθμισμένη μηχανή (Adjust Machine) και των λεπτομερειών για την λειτουργία της μηχανής (λειτουργία A234, Layout Hand Tools). Το Organize Materials and Scrap Bins χρησιμοποιεί σαν control βέλη πληροφορίες που προέρχονται από την ρυθμισμένη μηχανή (Adjust Machine), από την έτοιμη μηχανή (Ready Machine) και πληροφορίες για τα υλικά (Material Properties). Σκοπός της λειτουργίας είναι η τοποθέτηση κουτιών για τις πρώτες ύλες (Raw Materials) και των απορριφθέντων κομματιών (Rejected Job), ώστε να είναι εύκολη η φόρτωσή τους στη μηχανή.

Τέλος, η έτοιμη μηχανή (Ready Machine), τα τοποθετημένα εργαλεία (Placed Tools) και τα τοποθετημένα κουτιά (Placed Bins) δημιουργούν την έξοδο του διαγράμματος, που είναι η μηχανή έτοιμη για χρήση (Ready Machine Place).

### 5.3.6 Διάγραμμα A3



Σχήμα 5.6: *Inspect Job*, διάγραμμα A3 [6].

Με τα την κατεργασία του στο Complete Work Kit, το κομμάτι έρχεται για να ελεγχθεί στο Inspect Job. Στο Asses Completion Stage γίνεται μια αποτίμηση της αποπεράτωσης. Στη συνέχεια ελέγχονται οι διαστάσεις του κομματιού (Measurer Critical Dimensions). Για τον σκοπό αυτό χρησιμοποιούνται σαν πληροφορίες, τα control βέλη που αφορούν τα σημεία που πρέπει να γίνει ο έλεγχος (Inspection Points) και τις διαδικασίες μέτρησης (Measurement Procedures). Μετά τις μετρήσεις γίνεται μία γενική εκτίμηση της εργασίας, με πληροφορίες που προέρχονται από την προηγούμενη εξέταση των διαστάσεων (Measured Job), από τις απαιτούμενες ανοχές (Required Standards) βάση των σχεδίων και τα στάνταρ των ανοχών (Tolerance Standards) που επιβάλλεται από τον εγχειρίδιο ποιότητας. Τέλος το αποτέλεσμα τις εκτιμημένης εργασίας (Appraised Job) στέλνεται σαν πληροφορία μαζί με τα στάνταρ αποδοχής (Approval Standards), για να μπορέσει να αποφασίσει το Accept or Reject αν θα αποδεχθεί ή όχι το κομμάτι. Αν το κομμάτι γίνει αποδεκτό, τότε πηγαίνει στο Manage Job (βλ. διάγραμμα A0) και από εκεί εξέρχεται από το σύστημα, αλλιώς επιστρέφει στο Complete Work Kit (βλ. διάγραμμα A0) σαν Rejected Job για να κατεργαστεί εκ νέου.

#### 5.4 ΑΛΛΑΓΕΣ ΣΤΟ ΜΟΝΤΕΛΟ

Οι αλλαγές που προέκυψαν στο μοντέλο οφείλονταν κατά κύριο λόγο στα λάθη που έβαζε το Workflow Modeler, κατά το πέρασμα του μοντέλου στο Simulator. Το πρόγραμμα είχε κάποιες διαφοροποιήσεις από την IDEF0, οι οποίες έπρεπε να ληφθούν υπόψη προκειμένου να χρησιμοποιηθεί το Simulator.

Μια κύρια διαφορά είναι, ότι αντίθετα με τους ορισμούς της IDEF0 που απαιτούν την ύπαρξη ενός βέλους control και ενός output, το πρόγραμμα απαιτεί την ύπαρξη ενός input βέλους. Αν σε μία λειτουργία δεν υπάρχει είσοδος τότε κατά το compilation το πρόγραμμα εμφανίζει λάθος.

Ακόμα μια διαφορά είναι ότι στο Workflow Modeler δεν επιτρέπεται η χρήση βέλους mechanism, όταν αυτό προέρχεται από την έξοδο κάποιας λειτουργίας. Όταν δηλαδή μια λειτουργία παρέχει μια έξοδο που οδηγεί στον μηχανισμό μιας άλλης λειτουργίας τότε και πάλι το πρόγραμμα εμφανίζει σφάλμα (χρησιμοποιεί μηχανισμούς που προέρχονται από εξωτερικούς παράγοντες του μοντέλου μόνο).

Ενώ τα βέλη στην IDEF0 αναπαριστούν πληροφορία ή κομμάτια και όχι συνέχεια λειτουργιών, εν τούτοις στο Workflow Modeler τα control βέλη ανατροφοδότησης εμποδίζουν την εξέλιξη κατά την προσομοίωση, καθώς περιμένει πληροφορία που δεν υπάρχει κατά την πρώτη φορά που περνάει από την συγκεκριμένη λειτουργία.

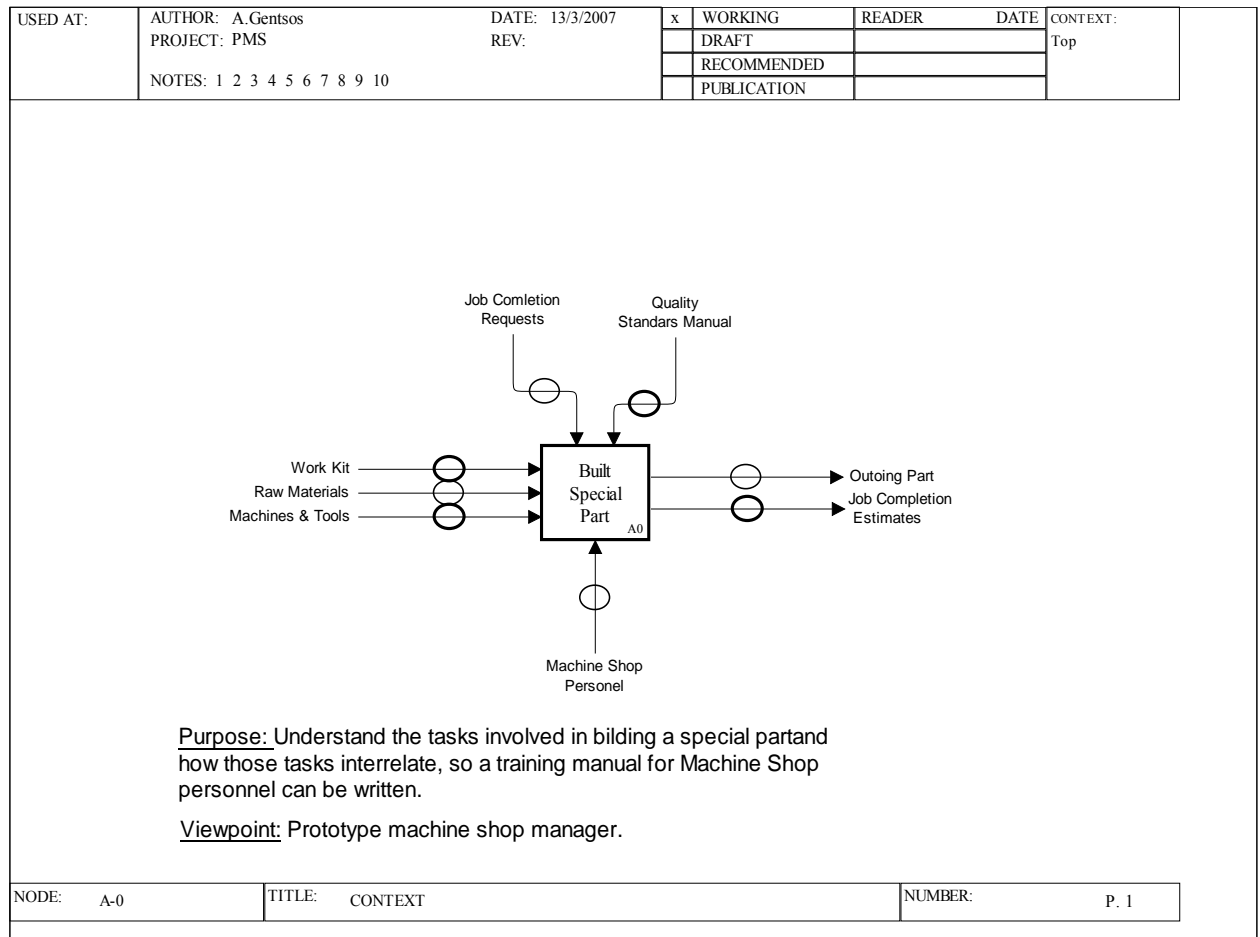
Λάθος κατά το compilation το πρόγραμμα εμφανίζει και στις περιπτώσεις tunneled βελών, τα οποία δεν δέχεται.

Τέλος κάποιες αλλαγές στο μοντέλο έγιναν προκειμένου η προσομοίωση να παρέχει λογικά αποτελέσματα.

Παρακάτω παρουσιάζονται αναλυτικά, διάγραμμα προς διάγραμμα, οι αλλαγές που έγιναν στο μοντέλο.

## Αλλαγές στα διαγράμματα

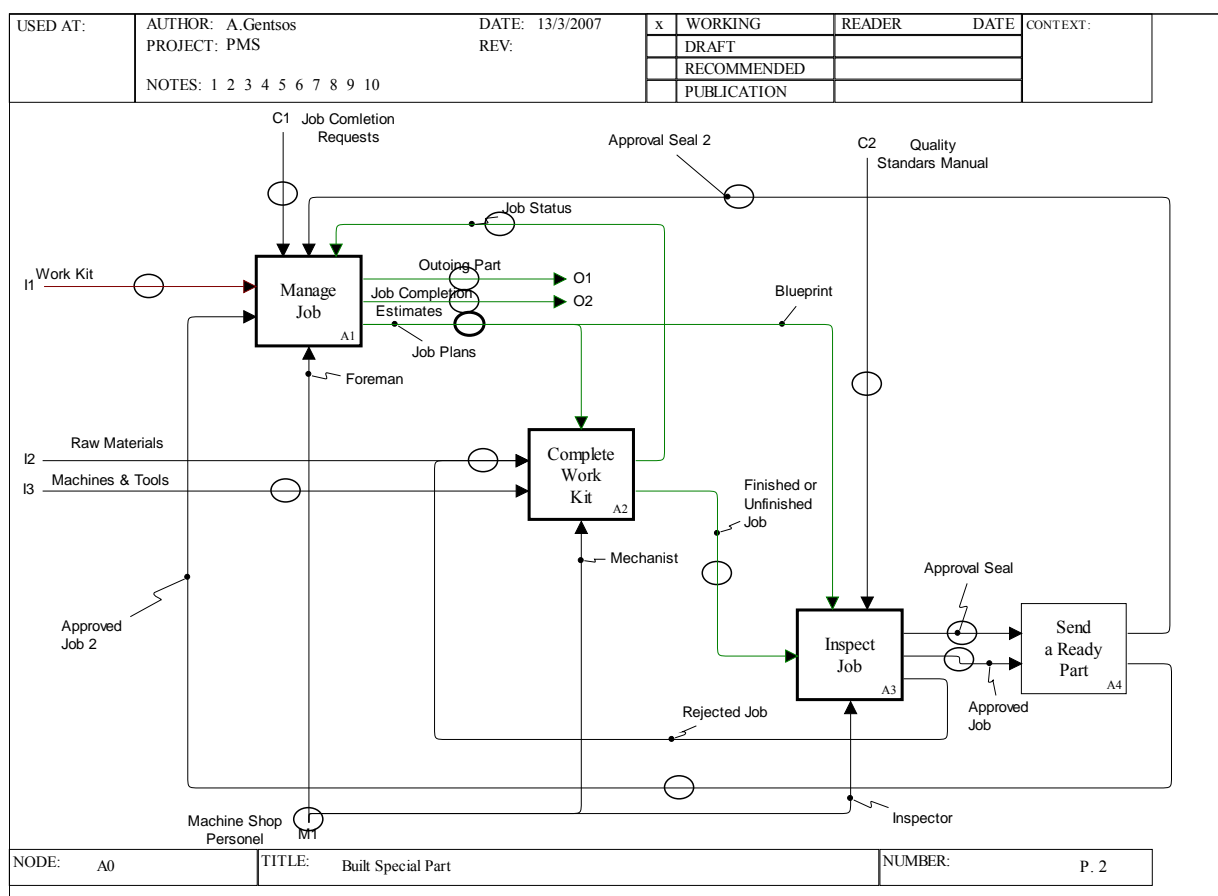
### 5.4.1 Διάγραμμα A-0



Σχήμα 5.7: *Build Special Part, A-0* διάγραμμα [7].

Εκ των πραγμάτων το A-0 διάγραμμα, δεν θα μπορούσε να περιέχει πολλές αλλαγές, μιας και πρόκειται για ένα εξαιρετικά απλό διάγραμμα. Παρόλα αυτά υπάρχει μια και μοναδική αλλαγή, που είναι ο διαχωρισμός των Raw Materials (πρώτων υλών) από το Work Kit, σαν ξεχωριστές εισόδους για το μοντέλο.

## 5.4.2 Διάγραμμα A0



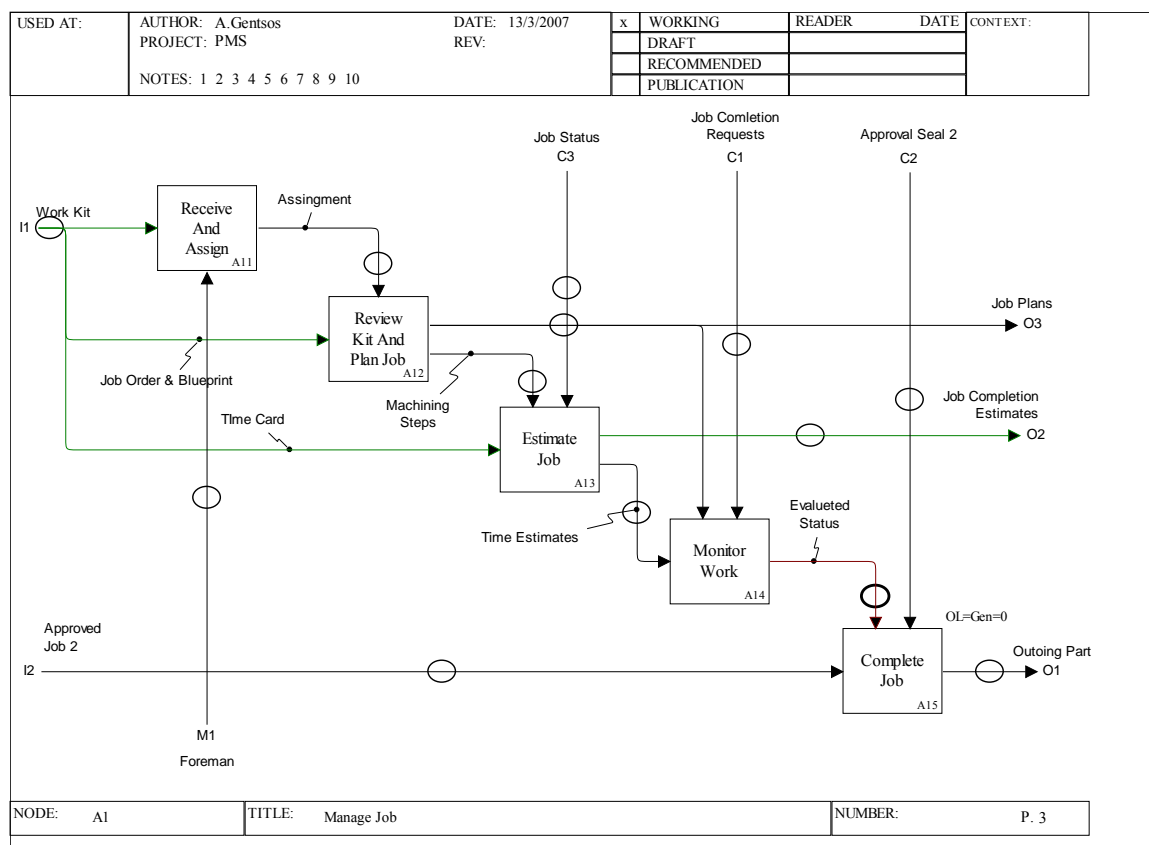
Σχήμα 5.8: *Build Special Part*, A0 διάγραμμα [8].

Η πρώτη αλλαγή στο μοντέλο, που έγινε, είναι ο διαχωρισμός των Raw Materials από το Work Kit. Αυτό έγινε γιατί τα Raw Materials, που είναι οι πρώτες ύλες, θεωρήθηκε ότι πρέπει να είναι μια ξεχωριστή εισαγόμενη οντότητα, ώστε να παίρνει δεδομένα για την προσομοίωση από την αρχή. Οι πρώτες ύλες είναι η σημαντικότερη είσοδος για μια γραμμή παραγωγής. Ακόμα μια αλλαγή έχει να κάνει με την αφαίρεση του Approved but Unfinished Job. Θεωρήθηκε πλεονασμός αφού προκύπτει από το Approved Job χωρίς ενδιάμεσο έλεγχο. Επίσης ήταν και δυσδιάκριτο στο σχήμα. Το Rejected Job ενώθηκε με το Raw Materials γιατί σε αντίθετη περίπτωση το Machine and Assemble (βλ. A2) θα περίμενε και τα δύο για να λειτουργήσει, ενώ στην πραγματικότητα κάθε φορά επεξεργάζεται ένα από τα δύο. Πρόκειται για έναν αυστηρό διαχωρισμό του Workflow Modeler, που η IDEF0 δεν το ξεκαθαρίζει. Το Approved Job έχει διαχωριστεί από το Approval Seal, κάτι που θα φανεί καλύτερα στο A3. Τέλος προστέθηκε το κουτί A4 (Send a Ready Part), που αναλαμβάνει να βοηθάει κατά την προσομοίωση, να αντιστοιχεί σε κάθε κομμάτι που



γίνεται αποδεκτό (Approved Job) και αποστέλλεται στο Manage Job, ένα Approval Seal. Σε αντίθετη περίπτωση άλλη στιγμή θα έφτανε το Approval Seal και άλλη το Approved Job, προκειμένου να γίνει η έγκριση για να γίνει αποδεκτό το κομμάτι, κάτι που θα ήταν παράλογο.

### 5.4.3 Διάγραμμα A1

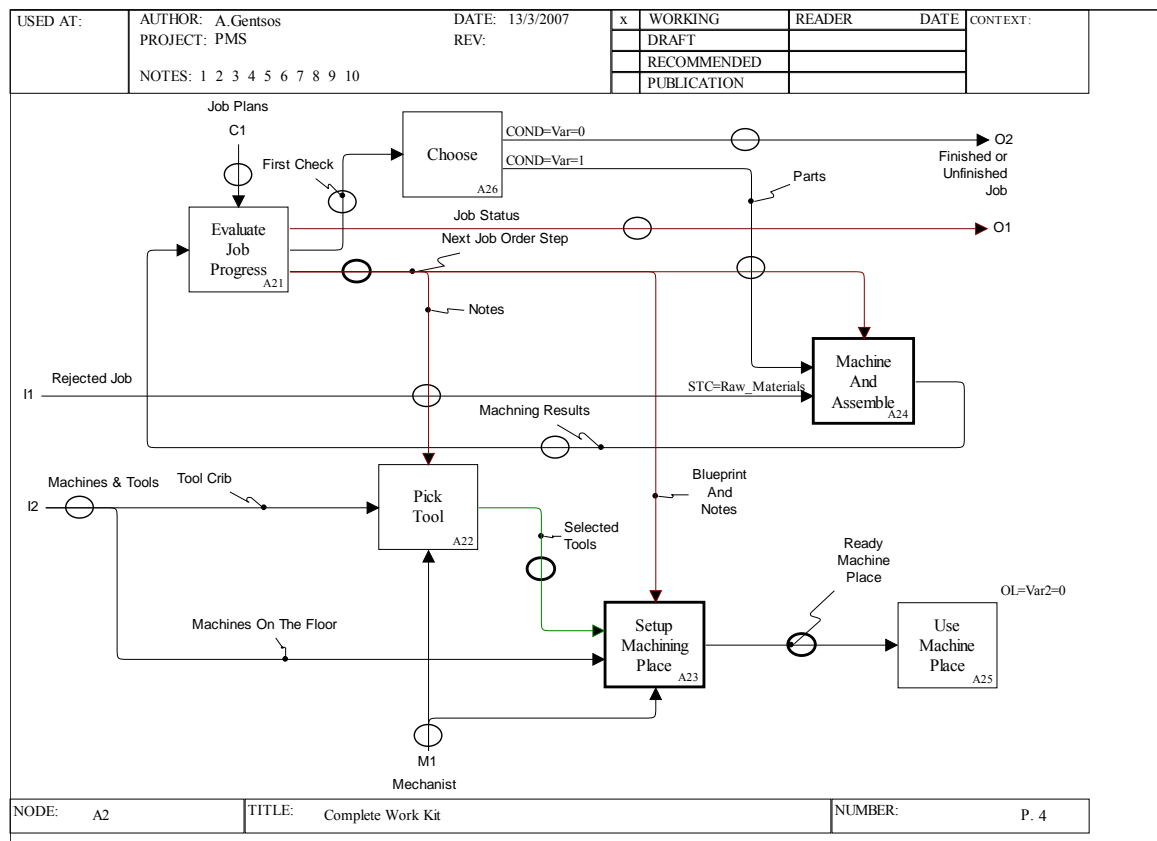


Σχήμα 5.9: Manage Job, A1 διάγραμμα [9].

Οι πρώτες αλλαγές στο μοντέλο που αφορούν το συγκεκριμένο διάγραμμα, έχουν να κάνουν με τα tunneled tail βέλη, που όπως αναφέρθηκε το Workflow Modeler δεν αναγνωρίζει. Πιο συγκεκριμένα το tunneled tail βέλος Ready Mechanic παραλείπεται, ενώ το tunneled tail βέλος Incoming Bench αντικαθίσταται με το Forman, προκειμένου να μην δημιουργηθεί και tunneled head βέλος στο parent diagram. Η δεύτερη διαφοροποίηση, όπως αυτή παρατηρείται, είναι η έλλειψη των control βελών Problems και Progress, στο Review Kit and Plan Job και στο Estimate Job αντίστοιχα. Αυτό γίνεται γιατί τα βέλη ανατροφοδότησης δεν μπορούν να χρησιμοποιηθούν την πρώτη φορά που εισέρχεται το κομμάτι στις αντίστοιχες

λειτουργίες, αφού δεν περιέχουν κάποια πληροφορία. Τα control βέλη το πρόγραμμα τα βλέπει σαν βέλη εισόδου, τα βέλη εισόδου για να πραγματοποιηθεί μια διαδικασία πρέπει να έχουν φτάσει όλα. Άρα αφού την πρώτη φορά τα control βέλη ανατροφοδότησης δεν περιέχουν πληροφορία, η προσομοίωση κολλάει. Μια ακόμα αλλαγή είναι η μετατροπή του Time Estimates από βέλος control σε βέλος εισόδου. Το Workflow Modeler απαιτεί, σε αντίθεση με την IDEF0, την ύπαρξη μιας τουλάχιστον εισόδου σε κάθε λειτουργία. Έτσι κι αλλιώς όπως αναφέρθηκε και προηγουμένως, το πρόγραμμα τα βλέπει και τα control βέλη σαν input, άρα η διαφορά είναι μόνο οπτική. Αντίθετα με το Workflow Modeler, η IDEF0 απαιτεί για να γίνει μια λειτουργία, την ύπαρξη ενός control και ενός output βέλους. Τέλος τα βέλη Machining Steps και Time Estimates έχουν χωριστεί από τα output 2 και 3 (O2 και O3).

#### 5.4.4 Διάγραμμα A2



Σχήμα 5.10: Complete Work Kit, διάγραμμα A2 [10].

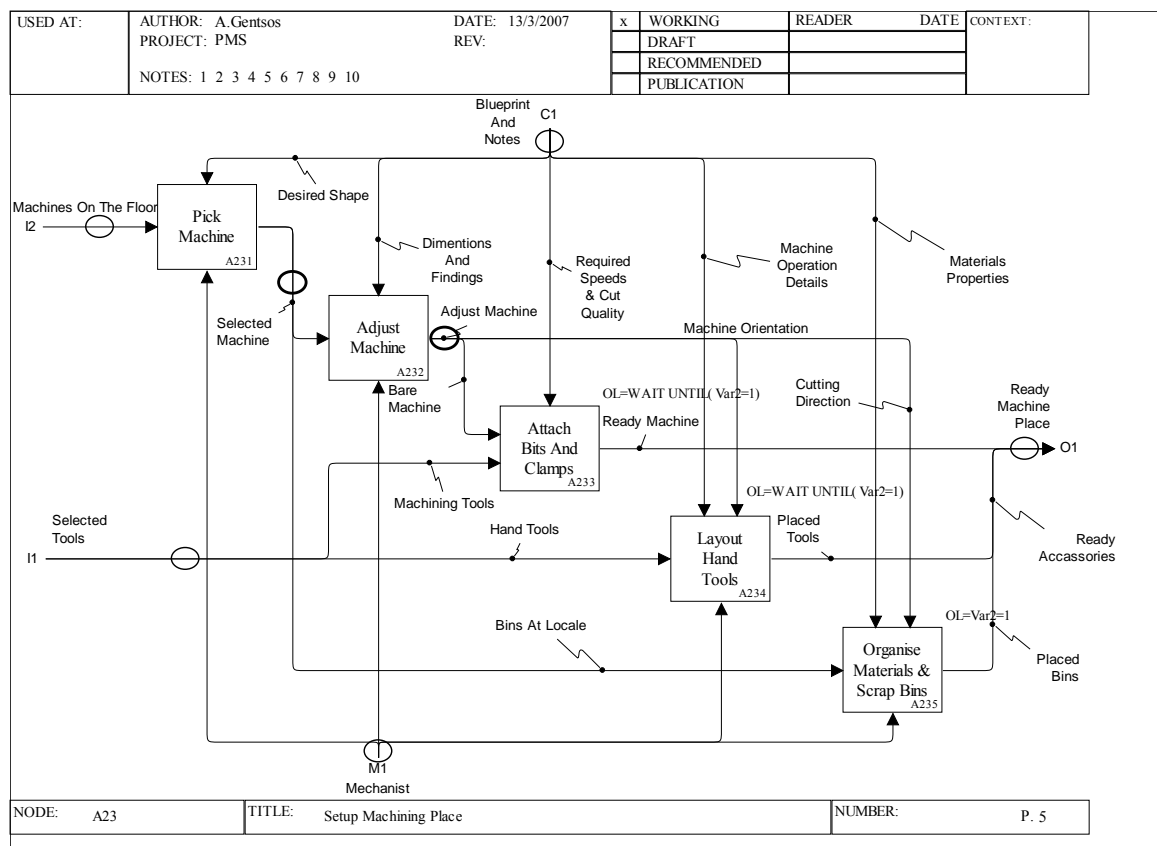
Στο Complete Work Kit, οι αλλαγές που έχουν γίνει είναι αρκετές. Κατ' αρχήν όπως έχει αναφερθεί πρωτότερα, αυτό διάγραμμα χωρίζεται σε δύο βασικές λειτουργίες. Η μία είναι η κατεργασία του κομματιού και η άλλη είναι η προετοιμασία της μηχανής για την κατασκευή του κομματιού. Στο πρωτότυπο μοντέλο οι λειτουργίες Pick Tool και Setup Machine Place έχουν βέλος mechanism, σε αυτό το διάγραμμα προστέθηκαν προκειμένου να αποφευχθεί ένα tunneled head τόξο στο parent diagram. Η λογική με την οποία προστέθηκαν τα βέλη είναι ότι ο υπεύθυνος για την μηχανή, επιλέγει τα εργαλεία που θα χρειαστούν για την κατεργασία και αναλαμβάνει τη ρύθμιση της μηχανής. Στο πρωτότυπο διάγραμμα η έξοδος από το κουτί A23 (Setup Machine Place), κατευθύνεται στο A34 (Machine and Assemble) και αποτελεί το mechanism βέλος. Όμως το Workflow Modeler δεν επιτρέπει σε βέλος που προέρχεται από την έξοδο άλλου κουτιού, να αποτελεί μηχανισμό σε άλλο. Για τον λόγο αυτό το βέλος εξόδου από το A23, σχεδιάστηκε να κατευθύνεται σε ένα νέο κουτί το Use Machine Place (A25). Όπως αναφέρθηκε και παραπάνω, το πρόγραμμα χρειάζεται μόνο ένα βέλος εισόδου για να μπορέσει να δεχτεί το κουτί σαν λειτουργία.

Στο υπόλοιπο διάγραμμα η πρώτη διαφορά είναι η απουσία του Unfinished Parts, που έχει βγει και από το parent diagram. Επίσης απουσιάζει και το Scrap Materials, που έχει γίνει ένα με το Raw Materials και αυτό από το parent diagram. Στη συνέχεια υπάρχει η μεγαλύτερη αλλαγή στο διάγραμμα, η οποία έγινε για να δίνει η προσομοίωση σωστά αποτελέσματα σύμφωνα με την λογική του διαγράμματος. Οι αλλαγές είναι η δημιουργία του κουτιού Choose και η δημιουργία child διαγράμματος για το Machine and Assemble.

Όπως έχει ξαναειπωθεί, η λογική αυτού του διαγράμματος είναι ότι μπαίνει η πρώτη ύλη (ή τα Rejected Job), ελέγχεται από το Evaluate Job Status που στέλνει πληροφορίες για την κατεργασία που πρέπει να γίνει και στη συνέχεια μπαίνει σαν Parts για κατεργασία στο Machine and Assemble. Τέλος το κομμάτι ξαναστέλνεται στο Evaluate Job Status και από εκεί βγαίνει από το διάγραμμα. Η δυσκολία ήταν να το Machine and Assemble παίρνει μια φορά Raw Materials ή Rejected Job και μια Parts και αυτό να φαίνεται και σε νούμερα κατά την προσομοίωση. Δηλαδή στην αρχική του μορφή το διάγραμμα έπαιρνε τυχαία ένα από τα δύο (όποιο έφτανε πρώτο), με αποτέλεσμα να εμφανίζεται ότι ο αριθμός των Raw Materials ή Rejected

Job και των Parts, διαφέρει πολύ. Αντίθετα με την λογική του διαγράμματος που θα πρέπει να είναι ίδιος ο αριθμός των Raw Materials ή Rejected Job και των Parts. Η λογική με την οποία λειτουργεί το Machine and Assemble θα φανεί αργότερα. Σε γενικές γραμμές όμως, μια μεταβλητή παίρνει τιμές μηδέν ή ένα, ανάλογα με το αν πρόκειται για Parts ή για Raw Materials ή Rejected Job (με πρώτα να παίρνει τα Raw Materials ή Rejected Job). Το Choose στη συνέχεια, ανάλογα με το τι τιμή έχει η μεταβλητή, αποφασίζει αν το κομμάτι θα το στείλει στο Machine and Assemble ως Parts ή έξω από το A2 ως Finished or Unfinished Job.

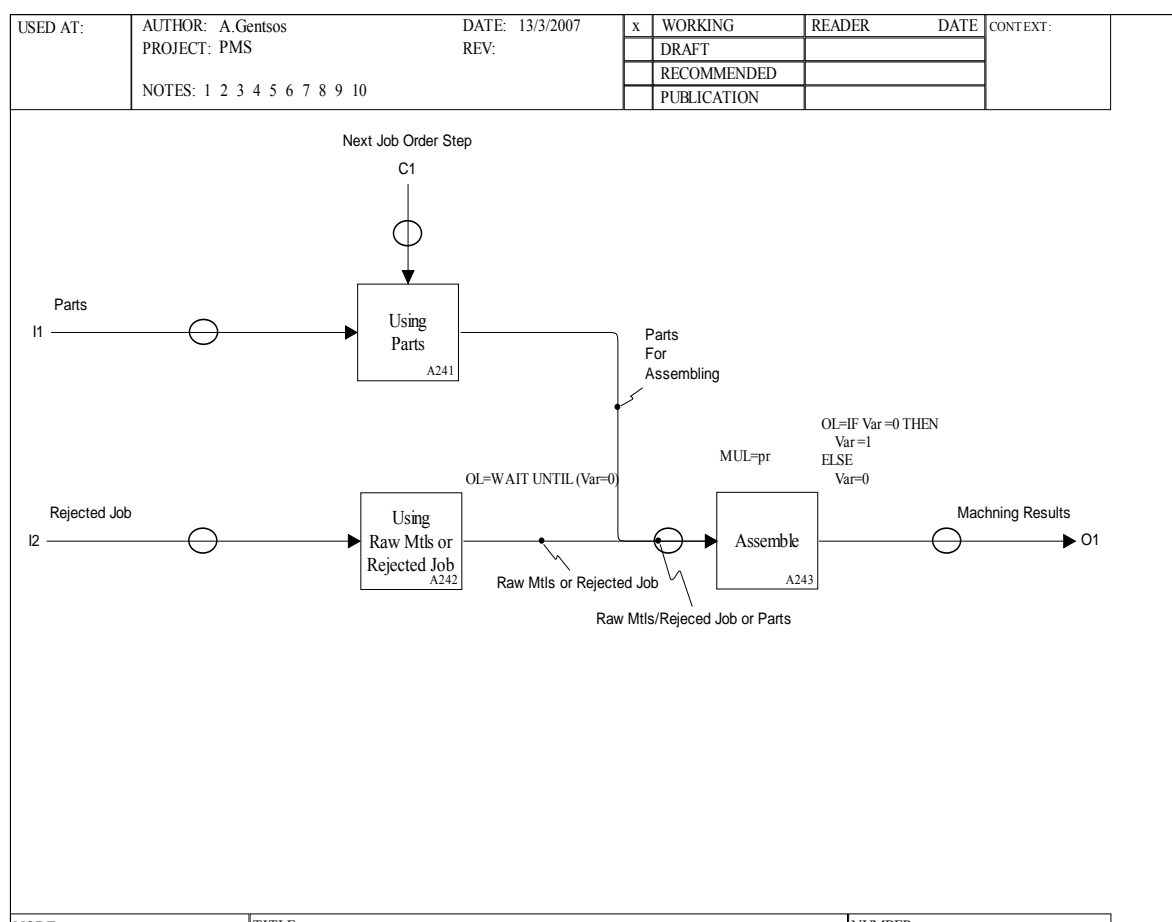
#### 5.4.5 Διάγραμμα A23



Σχήμα 5.11: Setup Machining Place, διάγραμμα A23 [11].

Οι διαφορές αυτού του διαγράμματος με το πρωτότυπο είναι μικρές και έχουν να κάνουν με τα tunneled head βέλη. Για τον λόγο αυτόν το tunneled head control βέλος, Machinist' s Physical Characteristics παραλήφθηκε. Για να αποφευχθεί η εμφάνιση tunneled head βέλους στο parent diagram, το mechanism βέλος Machinist στάλθηκε στα κουτιά A231, 232, A234 και A235.

#### 5.4.6 Διάγραμμα A24

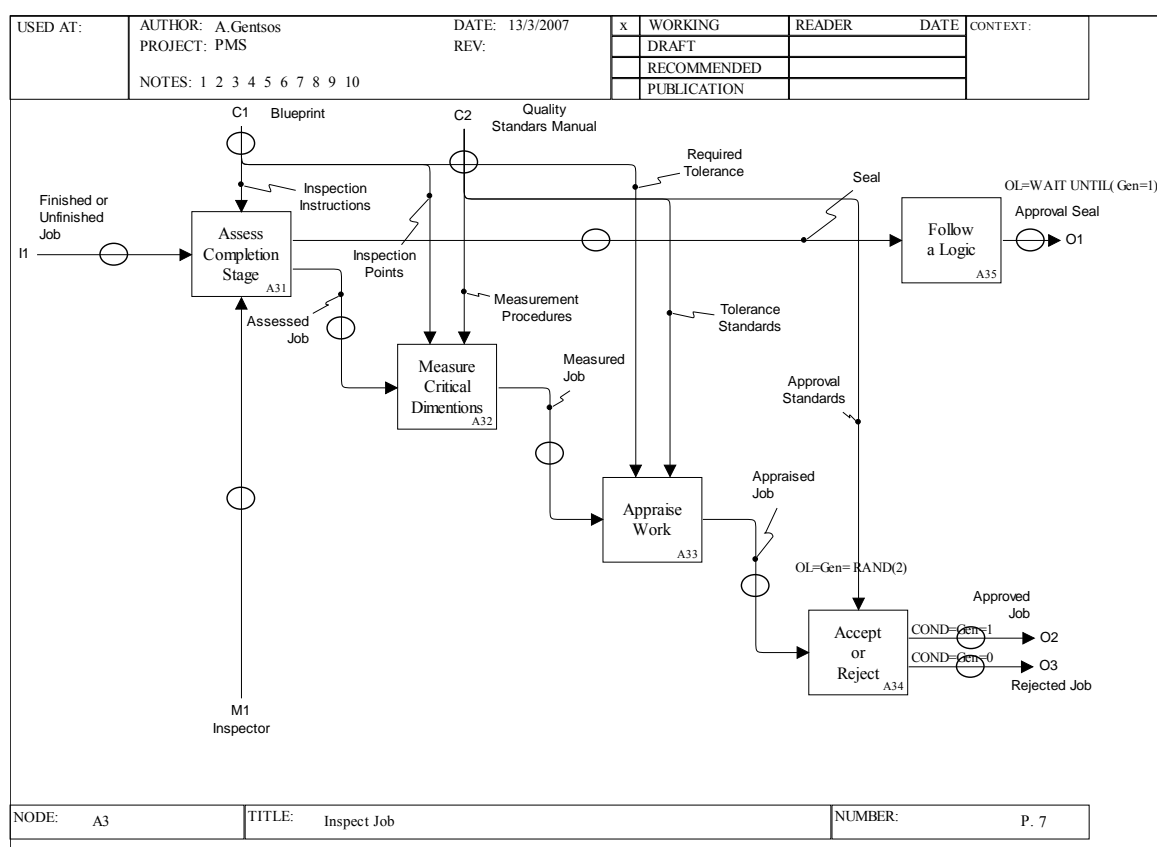


Σχήμα 5.12: Machine and Assemble, διάγραμμα A24 [12].

Η δημιουργία ενός child diagram για το Machine and Assemble, ήταν επιβεβλημένη προκειμένου το μοντέλο να λειτουργεί σωστά και να δίνει ορθολογικά αποτελέσματα. Η λογική του διαγράμματος είναι, τα Raw Materials ή το Rejected Job να ελέγχονται την πρώτη φορά που εισέρχονται στην λειτουργία και την δεύτερη να χρησιμοποιούνται ως Parts. Αναλυτικότερα η πρώτη είσοδος που πρέπει να χρησιμοποιήσει η λειτουργία είναι τα Raw Materials ή το Rejected Job (στην αρχή

Raw Materials). Μετά την είσοδό τους στέλνονται στο Evaluate Job Progress, από όπου αξιολογούνται και ξαναστέλνονται στο Machine and Assemble, μαζί με πληροφορίες για την κατεργασία που πρέπει να γίνει. Αυτό σημαίνει ότι στον κύκλο που δημιουργείται (Raw Materials ή το Rejected Job – Assemble – Evaluate Job Progress – Parts – Assemble) την πρώτη φορά πρέπει το Assemble να πάρει κομμάτι από το Raw Materials ή το Rejected Job, ενώ την δεύτερη από το Parts. Στα αποτελέσματα τις προσομοίωσης πρέπει να γίνεται εμφανές ότι η λειτουργία διεκπεραιώνεται σύμφωνα με την παραπάνω λογική (θα πρέπει να έχει χρησιμοποιηθεί ίσος αριθμός Raw Materials ή το Rejected Job και Parts). Με την κατασκευή του παρόντος διαγράμματος αυτό έγινε δυνατό. Η ακριβής λειτουργία του σύντομου κώδικα και της λογικής λειτουργίας του Using Raw Materials or Rejected Job, θα αναλυθούν αργότερα.

#### 5.4.7 Διάγραμμα A3

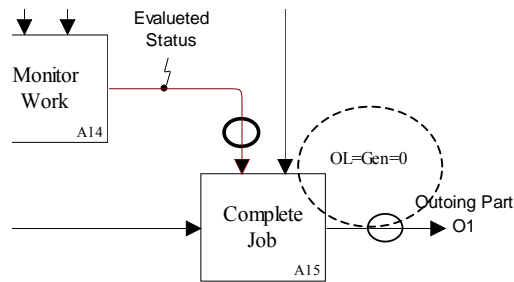


Σχήμα 5.13: *Inspect Job*, διάγραμμα A3 [13].

Η πρώτη εμφανής αλλαγή η προσθήκη του mechanism βέλους Inspector, με σκοπό να μην εμφανιστεί στο parent diagram tunneled head βέλος που όπως έχουμε ξαναπεί το Workflow Modeler δεν δέχεται. Στη συνέχεια βλέπουμε ότι τα control βέλη Measured Job και Appraised Job, που αποτελούν εξόδους των λειτουργιών Measure Critical Dimensions και Appraised Work αντίστοιχα, έχουν μετατραπεί σε input βέλη. Όπως έγινε γνωστό και παραπάνω το πρόγραμμα απαιτεί την ύπαρξη οπωσδήποτε ενός βέλους εισόδου, στην κατεύθυνση αυτή έγιναν και αυτές οι αλλαγές. Η επόμενες αλλαγές έχουν πάλι να κάνουν με τα σωστά αποτελέσματα που πρέπει να δίνει η προσομοίωση. Η λογική του διαγράμματος λέει ότι κάθε Approved Job θα πρέπει να συνοδεύεται από ένα Approval Seal. Αλλά χωρίς τις αλλαγές που έγιναν αυτό δεν συνέβαινε. Για τον σκοπό αυτό μπήκε στο διάγραμμα το κουτί Follow a Logic και διαχωρίστηκε το Approved Job από το Approval Seal. Έτσι το Approved Job και το Approval Seal συναντιούνται στο Send a Ready Part (βλ. διάγραμμα A1) και έτσι το ένα περιμένει το άλλο, αφού για να πραγματοποιηθεί η λειτουργία πρέπει να έχουν φτάσει και οι δύο εισοδοί. Άρα μετά που ξαναστέλνονται υπάρχουν και τα δύο. Ουσιαστικά το Follow a Logic και το Send a Ready Part, λειτουργούν σαν ρυθμιστικές «βάνες», προκειμένου να στέλνεται για κάθε Approved Job ένα Approval Seal και ταυτόχρονα. Σε αντίθετη περίπτωση τα αποτελέσματα της προσομοίωσης έδειχναν ότι κάποια κομμάτια χάνονταν.

## 5.5 ΕΞΗΓΗΣΗ ΜΙΚΡΩΝ ΚΩΔΙΚΩΝ ΚΑΙ ΛΕΙΤΟΥΡΓΙΩΝ

Την πρώτη εμφάνιση τύπου κώδικα την βλέπουμε στο διάγραμμα A1, όπου το Operation Logic (OL) της λειτουργίας μηδενίζει την μεταβλητή Gen. Είναι μια εκχώρηση τιμής η οποία γίνεται για αρχικοποίηση της μεταβλητής και δεν είναι απαραίτητη, αφού όπως θα δούμε και στο A3 παίρνει τυχαίες τιμές σε κάθε είσοδο στο Accept or Reject χωρίς να λαμβάνει υπόψη την προηγούμενη τιμή που είχε. Ο μηδενισμός της μεταβλητής σε αυτό το σημείο θεωρήθηκε σωστό να μπει για λόγους προγραμματιστικής ορθότητας. Εδώ πρέπει να αναφερθεί ότι το Operation Logic δεν είναι απαραίτητα η ακριβής λειτουργία του κουτιού, αλλά και ότι άλλο πρέπει να κάνει το μοντέλο κατά την προσομοίωση στο συγκεκριμένο σημείο.

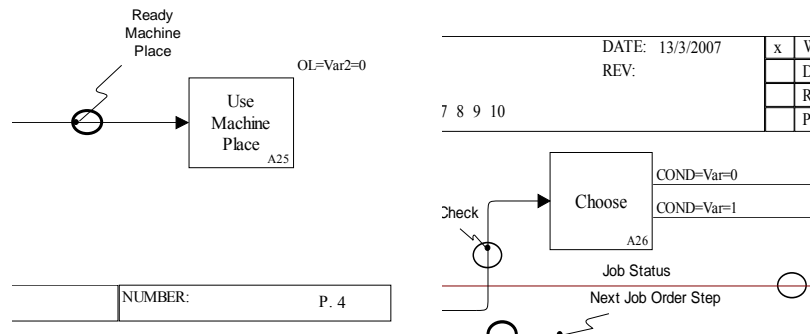


NUMBER: P. 3

Σχήμα 5.14: Operation Logic του Complete Job, διάγραμμα A1.

Στο διάγραμμα A2 Complete Work Kit, βλέπουμε η λειτουργία Choose να βγάζει είτε Finished or Unfinished Job είτε Parts, ανάλογα με το τι τιμή έχει πάρει η μεταβλητή Var στο Assemble (βλ. διάγραμμα A24 Machine and Assemble). Το COND συμβολίζει την συνθήκη (condition) κάτω από την οποία δίνεται η συγκεκριμένη έξοδος. Περισσότερες λεπτομέρειες της συγκεκριμένης λειτουργίας θα αναφερθούν παρακάτω στο κομμάτι που αφορά το διάγραμμα A24.

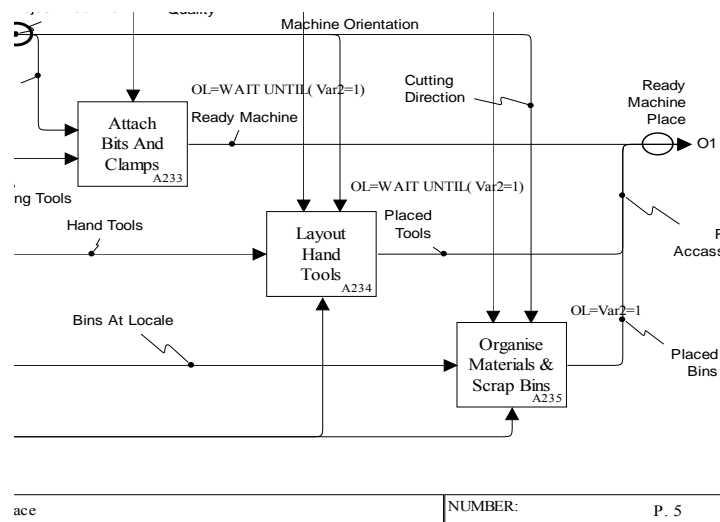
Στο Use Machine Place, το Operation Logic μηδενίζει την τιμή της μεταβλητής Var 2. Η μεταβλητή Var 2 χρησιμοποιείται από το διάγραμμα A23 ώστε οι λειτουργίες Attach Bits and Clamps, Layout Hand Tools και Organize Materials and Scrap Bins. Όπως θα δούμε και στο αντίστοιχο κομμάτι στο διάγραμμα A23 οι δύο πρώτες λειτουργίες περιμένουν την τρίτη να τελειώσει και να κάνει την μεταβλητή Var 2 = 1. Για τον σκοπό αυτόν η μεταβλητή πρέπει να ξαναπάρει την τιμή μηδέν στο Use Machine Place.



Σχήμα 5.15: Λειτουργίες από το διάγραμμα A2.



Στο διάγραμμα A23 βλέπουμε ότι οι λειτουργίες Attach Bits and Clamps και Layout Hand Tools περιμένουν την ολοκλήρωση της λειτουργίας Organize Materials and Scrap Bins. Αυτό επιτυγχάνεται με την χρήση της συνάρτησης Wait Until. Αυτή η συνάρτηση δίνει την εντολή στην λειτουργία να μην δώσει το αποτέλεσμά της, έως ότου μια συνθήκη γίνει αληθείς (στην προκειμένη περίπτωση Var 2 = 1). Η μεταβλητή έχει την τιμή μηδέν μέχρι να ολοκληρωθεί η λειτουργία Organize Materials and Scrap Bins, που η τιμή της γίνεται ίση με ένα. Τότε και οι τρεις λειτουργίες δίνουν ταυτόχρονα το αποτέλεσμά τους στην έξοδο του διαγράμματος. Έπειτα η μεταβλητή Var 2 παίρνει ξανά την τιμή μηδέν στο Use Machine Place, όπως αναφέρθηκε και παραπάνω. Η εφαρμογή αυτής της συνάρτησης στο μοντέλο έχει λογική, αφού για να λειτουργήσει η μηχανή πρέπει να βγάλουν και τα τρία αποτελέσματα. Όμως πιο πολύ η συγκεκριμένη εφαρμογή έγινε για δοκιμαστεί η λειτουργία της συνάρτησης.

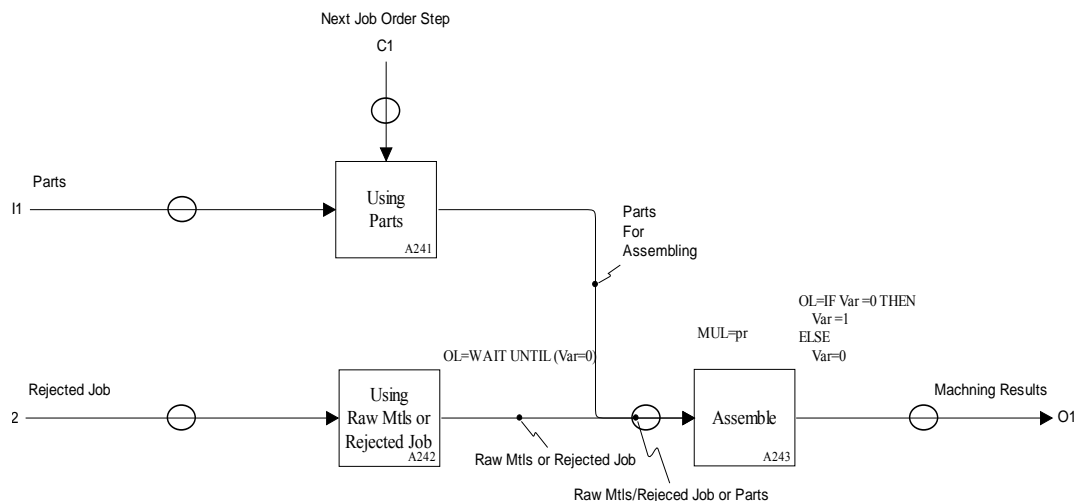


Σχήμα 5.16: Εφαρμογή της συνάρτησης Wait Until στο διάγραμμα A23.

Έναν σύντομο κώδικα έχουμε στο διάγραμμα A24, στην λειτουργία Assemble. Στον κώδικα αυτόν ρυθμίζεται η σειρά με την οποία θα πάρει κομμάτια η μηχανή. Πρώτα πρέπει να παίρνει τις πρώτες ύλες να τις επιθεωρεί και μετά να τις κατεργάζεται ως Parts. Για τον σκοπό αυτό χρησιμοποιείται η μεταβλητή Var, που την αρχικοποιούμε να έχει τιμή μηδέν. Στο Operation Logic του Using Raw Materials or Unfinished Job υπάρχει μια συνάρτηση Wait Until, που περιμένει την μεταβλητή να γίνει μηδέν για να στείλει κομμάτι. Έτσι από την πρώτη φορά η μηχανή θα πάρει κομμάτι από τις πρώτες ύλες. Στη συνέχεια που το κομμάτι μπαίνει

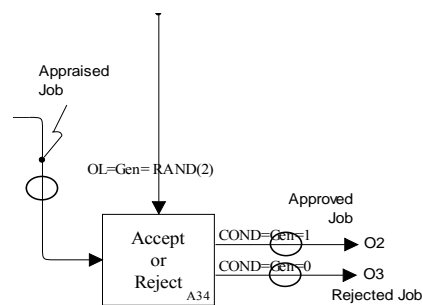
στο Assemble υπάρχει ο σύντομος κώδικας. Κώδικας: IF Var=0 THEN Var=1 ELSE Var=0. Αυτό που κάνει αυτό το if είναι να αλλάζει την τιμή στην μεταβλητή, με σκοπό κάθε φορά που είναι να πάρει κομμάτι να το παίρνει από αλλού (μία από Parts μια από Raw Materials or Unfinished Job). Έτσι μετά την πρώτη φορά που θα πάρει κομμάτι από το Using Raw Materials or Unfinished Job, θα κάνει την μεταβλητή ένα και το Using Raw Materials or Unfinished Job δεν θα μπορεί να στείλει άλλο κομμάτι. Στη συνέχεια το κομμάτι θα σταλεί στο Evaluate Job Progress και από εκεί αφού Var = 1, θα επιλέξει μέσω του condition την έξοδο Parts (βλ. διάγραμμα A2). Οπότε το Parts φτάνοντας στο Assemble θα μπει και θα κάνει την μεταβλητή μηδέν. Έτσι το Assemble θα μπορεί να πάρει πάλι κομμάτι από το Using Raw Materials or Unfinished Job και να ξεκινήσει ο κύκλος από την αρχή. Ταυτόχρονα το κατεργασμένο κομμάτι θα πάει πάλι στο Evaluate Job Progress και από εκεί θα επιλέξει την έξοδο Finished or Unfinished Job αφού θα πληροί το condition Var = 0.

Αξίζει να αναφερθεί ότι τα Parts for Assembling και Raw Materials or Unfinished Job κατασκευάστηκαν να ενώνονται σε ένα βέλος, γιατί αλλιώς η προσομοίωση δεν θα λειτουργούσε αφού το Assemble θα περίμενε κομμάτια και από τα δύο για να λειτουργήσει. Ενώνοντας τα δύο βέλη, ρυθμίστηκε μέσω του Workflow Modeler, το Assemble, να παίρνει κομμάτι από τη μια είσοδο κάθε φορά και πάντα προτεραιότητα να έχει το παλιότερο.



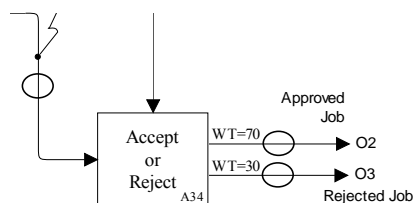
Σχήμα 5.17: Διάγραμμα A24.

Στο τελευταίο διάγραμμα, το Inspect Job και πιο συγκεκριμένα στη λειτουργία Accept or Reject, στο operation logic έχουμε την παρουσία μιας γεννήτριας τυχαίων αριθμών. Σκοπός της, είναι να εκχωρεί τις τιμές που παίρνει στην μεταβλητή Gen και βάση της τιμής της να επιλέγεται η επιθυμητή έξοδος. Η γεννήτρια λειτουργεί ως ακολούθως, παράγει πραγματικούς αριθμούς από μηδέν έως δύο και κρατάει μόνο το ακέραιό τους μέρος (αναπαριστάται ως RAND(2)). Έτσι η τιμές που παίρνει η μεταβλητή Gen είναι μηδέν ή ένα, επίσης υπάρχει η πιθανότητα η μεταβλητή να πάρει την τιμή δύο. Η τελευταία εκδοχή είναι απίθανη, αφού αν το δεκαδικό μέρος του αριθμού που παράγεται αποτελείται από τρία ψηφία (που είναι σίγουρα περισσότερα), τότε η πιθανότητα να γεννηθεί ο αριθμός δύο είναι μία στις δύο χιλιάδες (δύο χιλιάδες μία). Στις δύο δυνατές εξόδους υπάρχει condition και ανάλογα με τι αριθμός έχει γεννηθεί αποφασίζεται και η αντίστοιχη έξοδος. Έτσι υπάρχει μια ίση κατανομή των εξόδων (περίπου 50-50).



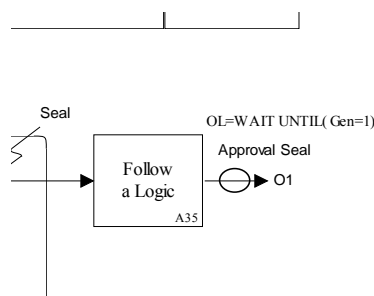
Σχήμα 5.18: Λειτουργία Accept or Reject με condition, διάγραμμα A3.

Μια άλλη δυνατότητα για επιλογή εξόδου είναι η χρήση βάρους, που συμβολίζεται από το πρόγραμμα με WT. Σύμφωνα με αυτή, ορίζουμε ένα βάρος για κάθε έξοδο και το συνολικό τους άθροισμα πρέπει να είναι εκατό. Σε μια δοκιμαστική προσέγγιση αυτής της επιλογής, θέσαμε βάρος ίσο με εβδομήντα για το Approved Job και βάρος ίσο με τριάντα για το Rejected Job. Η λειτουργία που ακολουθείται είναι ότι η έξοδος που θα παραχθεί θα είναι με πιθανότητα εβδομήντα της εκατό Approved Job και κατά τριάντα τις εκατό Rejected Job ή αλλιώς με την ολοκλήρωση της προσομοίωσης περίπου το εβδομήντα της εκατό των εξόδων που θα έχουν παραχθεί από την λειτουργία θα είναι Approved Job ενώ το τριάντα θα είναι Rejected Job.



Σχήμα 5.19: Λειτουργία Accept or Reject με βάρος, διάγραμμα A3.

Τέλος, όπως έχει ειπωθεί και παραπάνω, θέλουμε το Follow a Logic να στέλνει αποτελέσματα όταν το Accept or Reject παράγει Approved Job. Για τον σκοπό αυτό στο Follow a Logic δόθηκε operation logic wait until (Gen=1), που είναι το condition με το οποίο το Accept or Reject δίνει Approved Job.



Σχήμα 5.20: Operation Logic του Follow a Logic, διάγραμμα A3.

## 5.6 ΠΕΡΙΓΡΑΦΗ ΣΦΑΛΜΑΤΩΝ, ΚΑΘΥΣΤΕΡΗΣΕΩΝ ΚΑΙ ΠΑΡΑΛΛΗΛΩΝ ΔΙΑΔΙΚΑΣΙΩΝ ΣΤΑ ΔΙΑΓΡΑΜΜΑΤΑ IDEF0

Σφάλμα είναι οτιδήποτε μπορεί να επηρεάσει την εξέλιξη μιας διαδικασίας, κατά τρόπο τέτοιο που να παραχθεί κάποιο ανεπιθύμητο ή απρόβλεπτο αποτέλεσμα. Σφάλμα μπορεί να προκύψει σχεδόν σε οποιαδήποτε λειτουργία. Ένα σφάλμα μπορεί να είναι προβλέψιμο μπορεί και όχι. Το ίδιο ισχύει και για την χρονική στιγμή που θα γίνει το σφάλμα, που είτε μπορεί να υπολογιστεί με μαθηματική ακρίβεια (αιτιοκρατικό) ή μπορεί να είναι εντελώς απρόβλεπτο (στοχαστικό). Ένα σφάλμα μπορεί να προκληθεί από έναν άνθρωπο και καλείται ανθρώπινο σφάλμα. Τα ανθρώπινα σφάλματα δεν είναι δυνατόν να προβλεφθούν, όσον αφορά την χρονική στιγμή που θα συμβούν. Σφάλματα μπορεί να προκαλέσουν και οι μηχανές. Τα σφάλματα των μηχανών είναι πιο εύκολο να προβλεφθούν, χωρίς αυτό να σημαίνει

ότι προβλέπονται με μεγάλη ακρίβεια. Απλά από την κατασκευή τους υπάρχει μια πρόβλεψη για τον χρόνο ζωής τους και τον χρόνο μέχρι να συμβεί βλάβη. Τέλος σφάλματα μπορούν να συμβούν από αστάθμητους παράγοντες όπως είναι ο καιρός, κάτι που όμως δεν αφορά τις διαδικασίες που περιγράφουμε.

Στις περισσότερες από τις λειτουργίες μιας διαδικασίας υπάρχει η πιθανότητα δημιουργίας λάθους. Στην μοντελοποίηση της όμως, σε γλώσσες όπως η IDEF0, αυτό δεν γίνεται εμφανές γιατί η μεθοδολογία δεν περιλαμβάνει σφάλματα. Ο λόγος είναι ότι οι μεθοδολογίες αυτές, είναι κατασκευασμένες να περιγράφουν την διαδικασία που εκτελείται απλά. Έτσι διαβάζοντας ένα διάγραμμα IDEF0, ο απλός αναγνώστης δεν μπορεί να κατανοήσει τι λάθη μπορεί να γίνουν. Επίσης δεν είναι διακριτές οι συνέπειες που θα έχει το σφάλμα αυτό στην διαδικασία. Στα διαγράμματα που ακολουθούν έγινε προσπάθεια να εμπλουτιστεί η IDEF0 έτσι ώστε να λαμβάνει υπόψη και τα λάθη που μπορεί να συμβούν.

Για την μοντελοποίηση των σφαλμάτων, η ιδέα ήταν, να γίνει χρησιμοποίηση τελεστών. Τελεστές χρησιμοποιούν και άλλες μέθοδοι μοντελοποίησης επιχειρησιακών διαδικασιών όπως το πακέτο ARIS. Από τους τελεστές που υπάρχουν, για την κατασκευή των διαγραμμάτων, χρησιμοποιήθηκε ο τελεστής “και” και ο τελεστής “αποκλειστικό ή”. Ο τελεστής “και” σημαίνει ότι τα δύο βέλη που ακολουθούν ή προηγούνται, ενεργοποιούνται. Αντίστοιχα ο τελεστής “αποκλειστικό ή” σημαίνει ότι από τα βέλη που ακολουθούν ή προηγούνται ενεργοποιείται πάντα μόνο ένα από τα δύο.






Επίσης κάποιο λάθος μπορεί να μην έχει εμφανές αντίκτυπο στην διαδικασία και απλά να δημιουργεί μια καθυστέρηση. Για αυτή την περίπτωση θεωρήθηκε ένας τελεστής που η εμφάνισή του σημαίνει ότι αν ενεργοποιηθεί το συγκεκριμένο βέλος, θα υπάρξει μια καθυστέρηση στην διαδικασία. Ο τελεστής αυτός έχει σαν σύμβολο το ρολόι, για ευνόητους λόγους.

Δύο βέλη που προκύπτουν από δύο ή περισσότερες λειτουργίες, μπορεί να ενώνονται σε ένα βέλος. Αυτό μπορεί να σημαίνει ότι αυτές οι λειτουργίες αποτελούν παράλληλες διαδικασίες, αλλά μπορεί και όχι. Όμως κάτι τέτοιο σε ένα μοντέλο IDEF0 δεν είναι εμφανές. Για τον λόγο αυτόν χρησιμοποιήθηκαν δύο τελεστές ώστε

μια τέτοια περίπτωση να γίνεται ξεκάθαρη. Πρώτα θα επαναλάβουμε τι είναι παράλληλες διαδικασίες. Παράλληλες διαδικασίες ονομάζονται δύο διαδικασίες που λαμβάνουν χώρα ταυτόχρονα. Αυτό σημαίνει ότι μπορεί να ξεκινάνε μαζί και να μην ολοκληρώνονται την ίδια στιγμή, να μην ξεκινάνε μαζί αλλά να ολοκληρώνονται ταυτόχρονα ή η μία να ξεκινάει πριν αλλά να τελειώνει μετά την άλλη. Κύριο χαρακτηριστικό των παράλληλων διαδικασιών είναι ότι για να γίνει μια επόμενη λειτουργία, πρέπει να έχουν ολοκληρωθεί και οι δύο παράλληλες. Ο τελεστής που συμβολίζει ότι η οι λειτουργίες αποτελούν παράλληλες διαδικασίες, συμβολίζεται από δύο παράλληλες γραμμές. Στην αντίθετη περίπτωση έχουμε για τελεστή μία κουκίδα. Ο δεύτερος τελεστής συμβολίζει ότι από τα δύο βέλη περνάει πρώτη η πληροφορία ή κομμάτι που φτάνει πρώτο.

Πρέπει να αναφερθεί ότι στα διαγράμματα που θα ακολουθήσουν οι λογικοί τελεστές θα χρησιμοποιηθούν γενικότερα ως λογική μοντελοποίησης και όχι μόνο όπου πρέπει να μοντελοποιηθούν λάθη, παράλληλες διαδικασίες ή χαμένοι χρόνοι.

Ακολουθεί ένας πίνακας με τους τελεστές που χρησιμοποιούνται στα διαγράμματα που ακολουθούν.

Τελεστής	Όνομα τελεστή	Τι συμβολίζει
	Και	Και τα δύο βέλη που ακολουθούν ή προηγούνται ενεργοποιούνται
	Αποκλειστικό ή	Από τα βέλη που ακολουθούν ή προηγούνται ενεργοποιείται πάντα μόνο ένα από τα δύο.
	Καθυστέρηση.	Αν ενεργοποιηθεί το συγκεκριμένο βέλος, θα υπάρξει μια καθυστέρηση στην διαδικασία.
	Παράλληλες διαδικασίες.	Τα βέλη που ενώνονται από τις λειτουργίες που προηγούνται προέρχονται από παράλληλες διαδικασίες.
	Όχι παράλληλες διαδικασίες.	Από τα δύο βέλη περνάει πρώτη η πληροφορία ή κομμάτι που φτάνει πρώτο.

Πίνακας 5.1: Λογικοί τελεστές που χρησιμοποιούνται στην νέα μεθοδολογία.

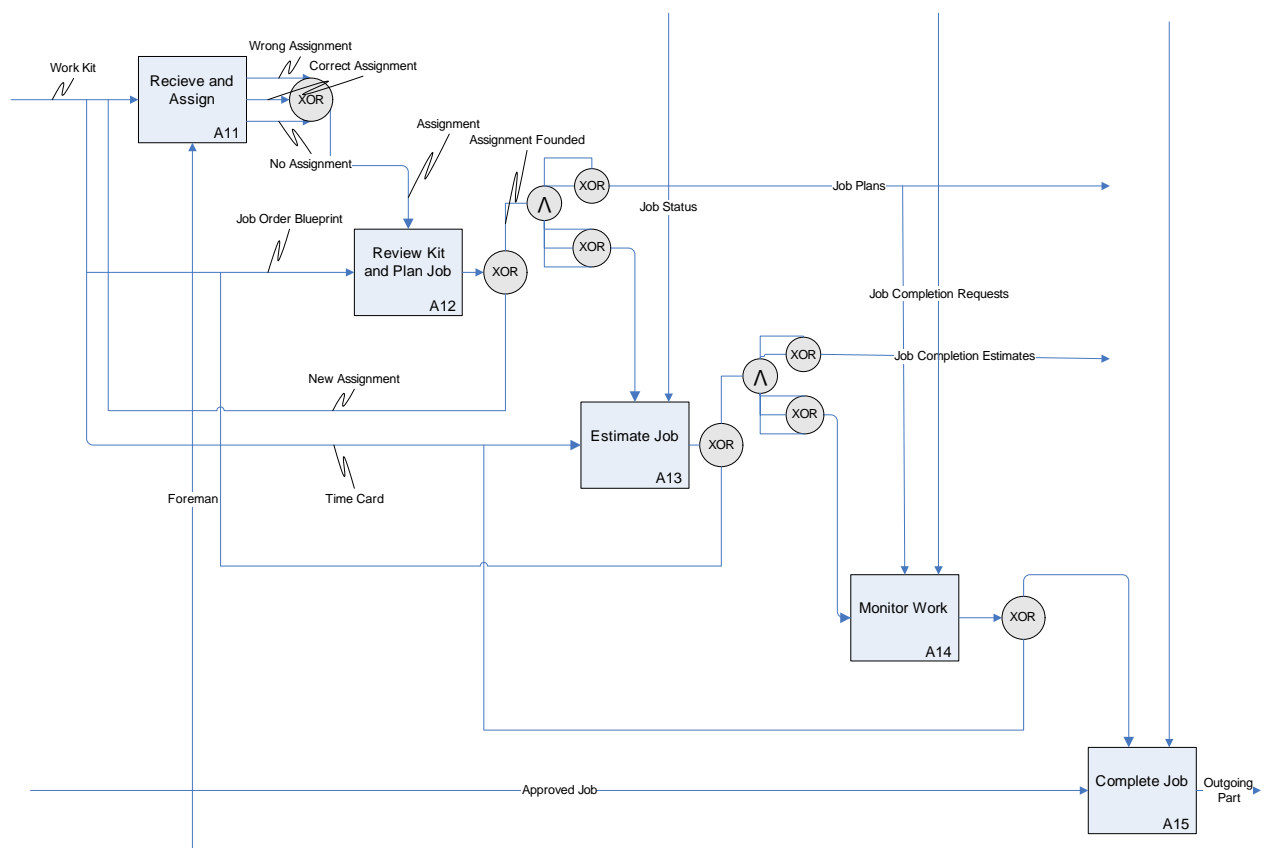
Με την χρήση των λογικών τελεστών, στην έξοδο κάθε διαδικασίας, που θα χρησιμοποιήσουμε, θα δείξουμε τις πιθανές διαδρομές που θα μπορούν να ακολουθηθούν αν συμβεί σφάλμα και αν όχι. Δηλαδή το μοντέλο που ήδη υπάρχει, προβλέπει κάποιες εξόδους χωρίς την υπόθεση κάποιου σφάλματος. Στο παράδειγμα που θα ακολουθήσει, θα δίνονται και πιθανές εξοδοι, πρόσθετα για την περίπτωση κάποιου σφάλματος.

Σε μία λειτουργία μπορεί να συμβεί κάποιο σφάλμα, αυτό θα έχει σαν συνέπεια ή να βγει λάθος αποτέλεσμα ή να μην βγει καθόλου αποτέλεσμα. Έτσι στην έξοδο των περισσότερων λειτουργιών θα παρατηρούμε έναν λογικό τελεστή “αποκλειστικό ή” που θα δίνει τρεις δυνατές επιλογές, να βγαίνει σωστό αποτέλεσμά, λάθος αποτέλεσμα ή να μην βγαίνει αποτέλεσμα. Ο τελεστής “και” θα

χρησιμοποιείται συχνότερα, σε περιπτώσεις που η λειτουργία δίνει παραπάνω από ένα αποτελέσματα και πρέπει να δείξουμε ότι βγαίνουν όλα.

Στη συνέχεια παρουσιάζονται τα διαγράμματα και πως αυτά προέκυψαν με την προσθήκη των σφαλμάτων. Στο διάγραμμα A-0, είναι προφανώς αδύνατο να μοντελοποιηθεί κάποιο σφάλμα. Τα ίδιο ισχύει και για το A0 διάγραμμα, αφού όλες του οι λειτουργίες αναλύονται σε child diagrams. Για την εξυπηρέτηση του σκοπού μας, χρησιμοποιήθηκε πάλι το παράδειγμα του μηχανουργείου, Build Special Part.

### 5.6.1 Διάγραμμα A1



Σχήμα 5.21: Διάγραμμα A1 μετά την προσθήκη σφαλμάτων [14].

Στο διάγραμμα A1 και πιο συγκεκριμένα στην λειτουργία Receive and Assign, παρέχεται μια ανάθεση για τις εργασίες που πρέπει να γίνουν κατά την κατασκευή του κομματιού από κάθε τμήμα τις διαδικασίας. Σε αυτό το σημείο τα λάθη που μπορεί να προκύψουν είναι, να σταλεί λάθος ανάθεση ή να μην σταλεί καθόλου ανάθεση. Φυσικά υπάρχει η πιθανότητα να μην συμβεί κανένα λάθος και να σταλεί το σωστό Assignment. Όπως είναι εύκολο να παρατηρηθεί η λέξη που



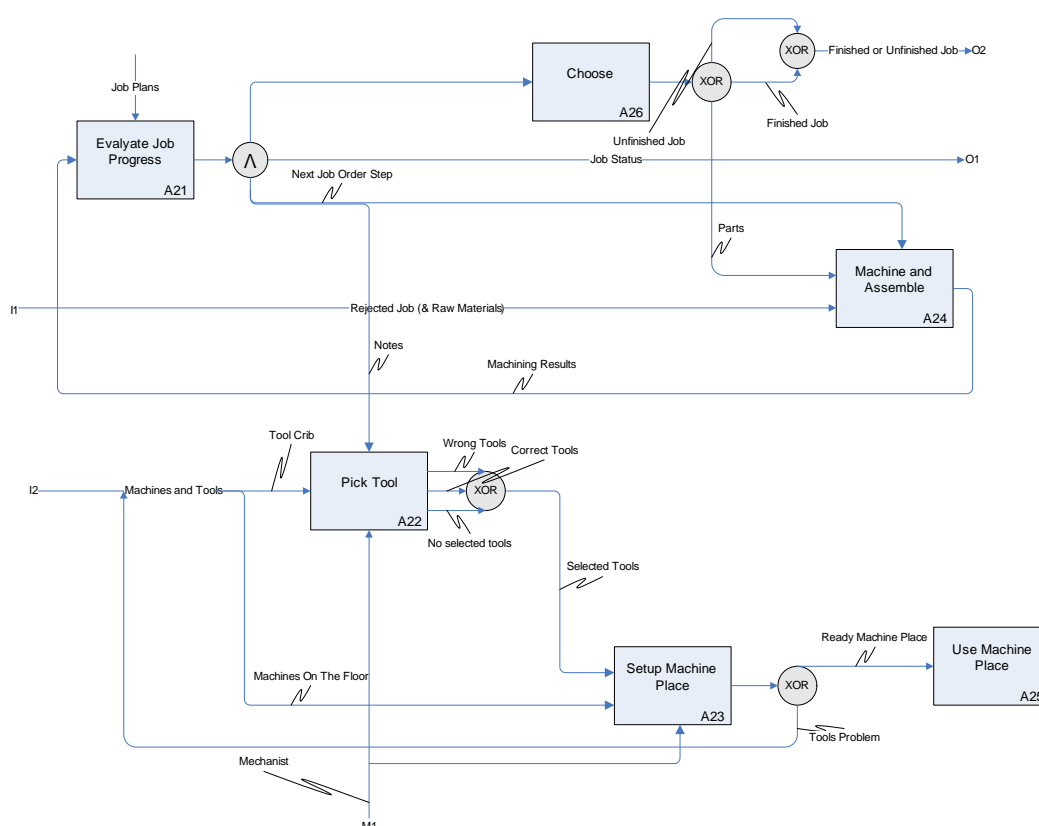
χρησιμοποιούμε είναι η λέξη “ή” (ή λάθος ανάθεση ή να μην σταλεί καθόλου ανάθεση ή να σταλεί το σωστό Assignment). Ως εκ τούτου γίνεται προφανές ότι ο τελεστής που πρέπει να χρησιμοποιήσουμε είναι το αποκλειστικό ή. Οι τρεις προαναφερθείσες εκδοχές είναι πλέον οι εξόδοι του Receive and Assign και συνδέονται πάνω στον τελεστή. Από αυτά τα βέλη ένα και μόνο είναι η τελική έξοδος του κουτιού, το Assignment.

Η επόμενη λειτουργία είναι η Review Kit and Plan Job. Εδώ από το Assignment που περιμένει η λειτουργία από πριν μπορεί να εμφανιστεί λάθος. Δηλαδή, αν το Assignment δεν έχει αποσταλεί διαπιστώνεται εδώ και η διαδικασία επιστρέφει στο Receive and Assign. Πρέπει να διευκρινιστεί ότι αν το Assignment είναι λάθος δεν θα φανεί πουθενά, είναι η περίπτωση που το κομμάτι θα κατασκευαστεί λάθος. Το ίδιο ισχύει για τις αντίστοιχες περιπτώσεις από εδώ και πέρα. Το Review Kit and Plan Job έχει δύο εξόδους, τα σχέδια της κατεργασίας (Job Plans) και τα βήματα που πρέπει να γίνουν (Machining Steps). Αν έχει αποσταλεί Assignment, υπάρχουν τρεις εκδοχές που μπορεί να συμβούν για τα βήματα που πρέπει να ακολουθηθούν και δύο για τα σχέδια. Αναλυτικά Μπορεί είτε να σταλούν σωστά είτε λάθος σχέδια και σωστά λάθος ή καθόλου βήματα που πρέπει να ακολουθηθούν. Άρα στο σχέδιο έχουμε έναν τελεστή αποκλειστικό ή, για τον αν έχει ή όχι αποσταλεί Assignment. Αν δεν έχει αποσταλεί η διαδικασία επιστρέφει στο Receive and Assign. Αλλιώς έχουμε τις δύο εξόδους που δίνει η λειτουργία χρησιμοποιώντας τον τελεστή και, για να δηξουμε ότι βγαίνουν από το κουτί και τα δύο βέλη. Στην συνέχεια χρησιμοποιούνται δύο τελεστές αποκλειστικό ή, για να φανεί ότι ένα μόνο από τα δύο και τα τρία αντίστοιχα βέλη, θα αποτελούν τα βέλη εξόδου.

Στην λειτουργία Estimate Job, έχουμε ακριβώς την ίδια διάταξη βελών και τελεστών με πριν. Δηλαδή αν δεν έχει αποσταλεί από πριν δεδομένο για τα βήματα που πρέπει να ακολουθηθούν η διαδικασία επιστρέφει στο Review Kit and Plan Job. Αλλιώς η λειτουργία βγάζει κανονικά τις εξόδους Job Completion Estimates και Time Estimates. Οι οποίες μπορεί να είναι είτε σωστές είτε λάθος ή και να μην αποσταλεί καθόλου Time Estimates.

Τέλος στο Monitor Work τα πράγματα είναι πολύ συγκεκριμένα, αν δεν έχει αποσταλεί Time Estimates, η διαδικασία επιστρέφει στο Estimate Job, αλλιώς παρέχει μια αξιολόγηση για την κατάσταση του κομματιού (Evaluated Status) στο Complete Job. Οπότε πάλι υπάρχει χρήση του τελεστή αποκλειστικό ή. Το Complete Job παρέχει την έξοδο του κομματιού από την διαδικασία, οπότε δεν κρίθηκε σκόπιμη η εισαγωγή κάποιας άλλης εκδοχής.

### 5.6.2 Διάγραμμα A2



Σχήμα 5.22: Διάγραμμα A2 μετά την προσθήκη σφαλμάτων [15].

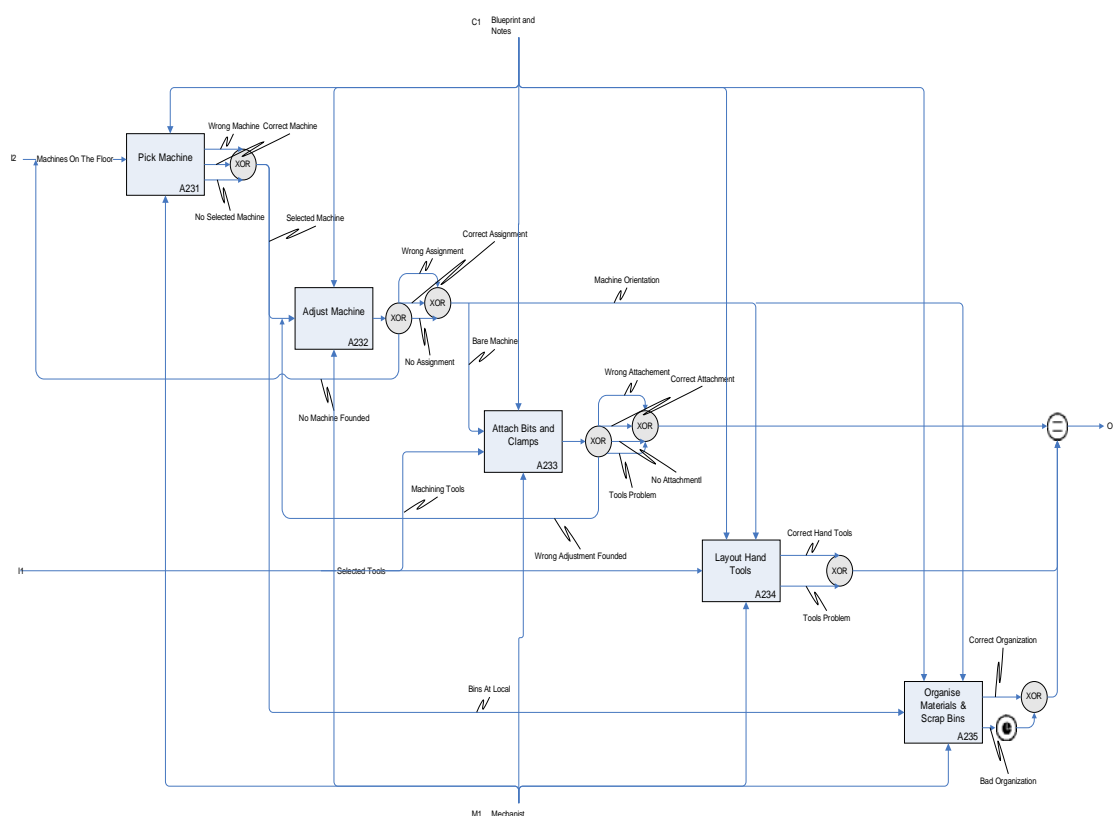
Για την περιγραφή του διαγράμματος A2, θα ξεκινήσουμε πάλι από το κάτω μέρος του διαγράμματος που δημιουργεί την μηχανή για την λειτουργία κατασκευής του κομματιού (Machine and Assemble). Στο Pick Tool γίνεται η επιλογή των εργαλείων, αυτό σημαίνει ότι μπορεί να επιλεγούν τα σωστά εργαλεία, μπορεί όμως να επιλεγούν λάθος εργαλεία ή καθόλου εργαλεία. Άρα οι τρεις έξοδοι συνδέονται σε έναν τελεστή αποκλειστικό ή και από εκεί μία μόνο από τις τρεις πιθανές θα είναι η πραγματική έξοδος. Στο Setup Machining Place, αν δεν έχει γίνει κάποιο λάθος με τα εργαλεία, η μηχανή είναι έτοιμη να λειτουργήσει κανονικά. Έχουμε λοιπόν ένα

αποκλειστικό ή το οποίο θα δώσει έξοδο ή στο Ready Machine Place ή στο Tools Problem και η διαδικασία θα επιστρέψει στο Pick Tool.

Το Machine and Assemble αναλύεται σε child diagram και τα σφάλματα που μπορεί να συμβούν κατά την λειτουργία του θα φανούν εκεί. Το Evaluate Job Status τώρα είναι μια λειτουργία που πάντα, ανεξάρτητα αν δέχεται πρώτες ύλες για έλεγχο ή κομμάτια προς έγκριση, δίνει τρεις συγκεκριμένες εξόδους. Για τον λόγο αυτόν προφανώς θα έχει τελεστή και. Οι τρεις εξοδοί είναι το Job Status, το First Check και το Next Job Order.

Η έξοδος της λειτουργίας Choose εξαρτάται από το τι έχει στείλει το Machine and Assemble. Αν έχει στείλει πρώτες ύλες για έλεγχο τότε η έξοδός του είναι η αποστολή τους πίσω στο Machine and Assemble, ως Parts, για να γίνει η κατεργασία. Σε αντίθετη περίπτωση θα δώσει το κατεργασμένο κομμάτι στο A3 (Inspect Job) για έλεγχο. Προφανώς το κομμάτι μπορεί να είναι είτε καλό είτε σκάρτο, άρα το Choose σε αυτή την περίπτωση, θα δώσει ή Finished Job ή Unfinished Job. Άρα θα έχουμε έναν τελεστή αποκλειστικό ή που θα δίνει ή Parts στο Machine and Assemble ή κομμάτι στο A3, και έναν αποκλειστικό ή που θα ξεχωρίζει ότι το κομμάτι αυτό μπορεί να είναι σκάρτο ή όχι.

### 5.6.3 Διάγραμμα A23



Σχήμα 5.23: Διάγραμμα A32 μετά την προσθήκη σφαλμάτων [16].

Ξεκινώντας το διάγραμμα A23, πρέπει να γίνει η επιλογή της μηχανής που θα κάνει την κατεργασία του κομματιού σύμφωνα με τα σχέδια που υπάρχουν. Αυτό σημαίνει ότι το Pick Machine, μπορεί να κάνει την σωστή επιλογή μηχανής, την λάθος επιλογή μηχανής ή να μην επιλέξει καν μηχανή. Αυτές οι τρεις πιθανές εξοδοι συνδέονται με έναν τελεστή αποκλειστικό ή και έχουμε την μία έξοδο που θα προκύψει.

Το αποτέλεσμα της λειτουργίας A231, μπαίνει σαν είσοδος στο Adjust Machine και αν δεν έχει γίνει επιλογή μηχανής πριν, επιστρέφει για να κάνει αυτή την φορά την επιλογή. Αλλιώς (αποκλειστικό ή) υπάρχουν οι τρεις γνωστές πιθανές εξοδοι, να σταλεί σωστό Adjustment, λάθος ή καθόλου. Οπότε γίνεται χρήση ενός ακόμη τελεστή αποκλειστικό ή και προκύπτει η διάταξη που φαίνεται στο διάγραμμα.

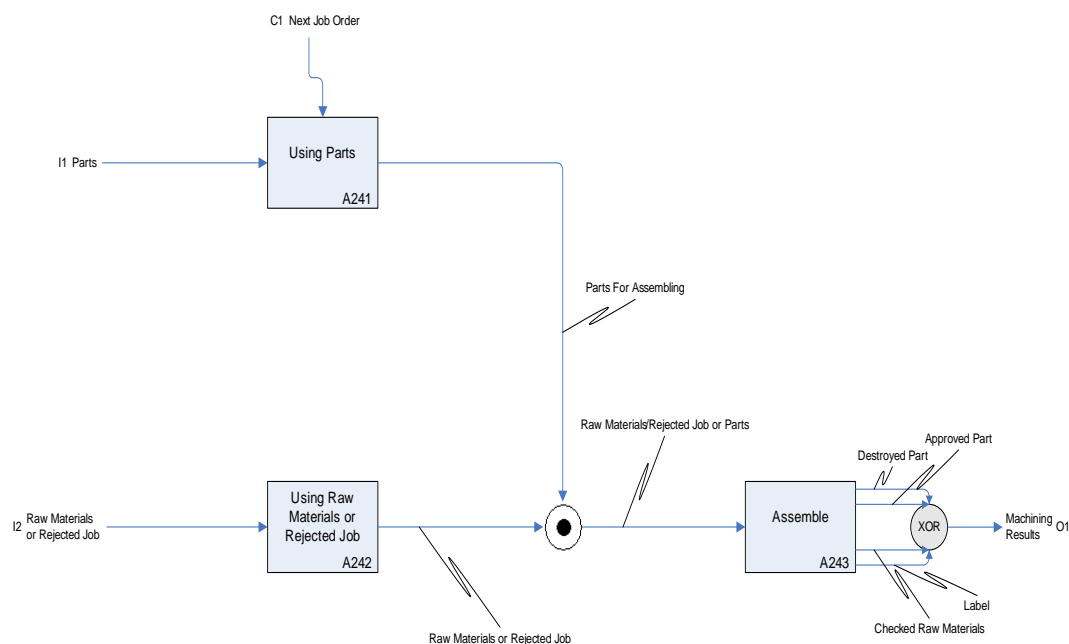
Αν στο Attach Bits and Clamps δεν έχει αποσταλεί Adjustment, τότε έχουμε επιστροφή στο Adjust Machine για να σταλεί. Σε αντίθετη περίπτωση η λειτουργία

εκτελείται κανονικά με τέσσερις πιθανές εξόδους. Όπως συνήθως, σωστό, λάθος ή κανένα Attachment αλλά και πιθανό πρόβλημα που μπορεί να έχει προκύψει με τα εργαλεία. Η διάταξη πάλι γίνεται με την χρήση δύο τελεστών αποκλειστικό ή. Η έξοδος που προκύπτει σε περίπτωση που το Adjustment έχει αποσταλεί κανονικά ενώνεται με αυτές από τις δυο επόμενες λειτουργίες. Αυτές οι τρεις αποτελούν παράλληλες διαδικασίες και για τον λόγο αυτό εκεί που συνδέονται τα βέλη των εξόδων, τους υπάρχει ο αντίστοιχος τελεστής.

Το Layout Hand Tool, έχει δύο πιθανές εξόδους, το να επιλεγούν σωστά τα χειροκίνητα εργαλεία και το να υπάρχει πρόβλημα με αυτά. Το δεύτερο περιλαμβάνει δύο καταστάσεις. Μία να μην έχουν δοθεί τα σωστά εργαλεία από το Pick Tool και μία να γίνει εκεί λάθος επιλογή εργαλείων. Οι δύο παραπάνω εξοδοί συνδέονται με έναν τελεστή αποκλειστικό ή. Αν είτε στο Attach Bits and Clams είτε στο Layout Hand Tool, υπάρχει πρόβλημα με τα εργαλεία, η έξοδος της διαδικασίας Setup Machine Place είναι η Tools Problem και επιστρέφει η διαδικασία στο Pick Tool, προκειμένου αυτή τη φορά να γίνουν όλα σωστά (βλ. διάγραμμα A2).

Τέλος στο Organize Materials and Scrap Bins (A235), δεν γίνεται να μην γίνει καθόλου οργάνωση. Άρα μπορεί να γίνει καλή η κακή οργάνωση. Μια κακή οργάνωση κατά την λειτουργία είναι πιθανό να επισύρει μια καθυστέρηση, γι' αυτό χρησιμοποιούμε τον τελεστή για την καθυστέρηση. Φυσικά μία από τις δύο πιθανές εξόδους θα προκύψει και αυτός είναι ο λόγος που χρησιμοποιούμε τον τελεστή αποκλειστικό ή.

#### 5.6.4 Διάγραμμα A24

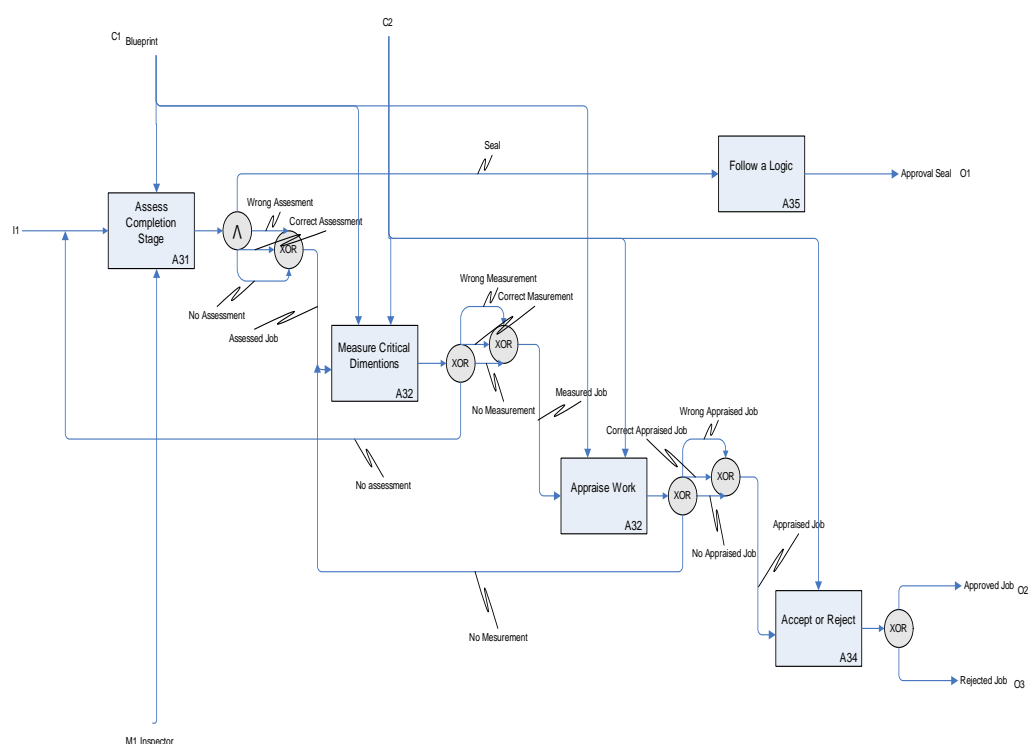


Σχήμα 5.24: Διάγραμμα A24 μετά την προσθήκη σφαλμάτων [17].

Το διάγραμμα A24 Machine and Assemble, δεν προσφέρεται για πολλά σχόλια. Το σημαντικότερο εδώ είναι η ένωση των βελών των εξόδων των λειτουργιών Using Parts και Using Raw Materials or Rejected Job. Οι δυο αυτές διαδικασίες παρότι μοιάζουν δεν αποτελούν περίπτωση παράλληλων διαδικασιών. Όπως έχει γίνει γνωστό και στην περιγραφή του μοντέλου το Assemble παίρνει κομμάτι μία από το Using Raw Materials or Rejected Job και μία από το Using Parts. Σκοπός είναι το Assemble να στέλνει τις πρώτες ύλες για έλεγχο και έπειτα με τα δεδομένα που παίρνει να τα κατεργάζεται ως Parts. Άρα στην ένωση των δύο βελών θα μπει ο τελεστής που συμβολίζει ότι δεν πρόκειται για παράλληλες διαδικασίες.

Το Assemble τώρα, έχει την δυνατότητα να βγάλει τέσσερις διαφορετικές εξόδους, ανάλογα με το τι λειτουργία εκτέλεσε. Αν έχει απλά ελέγξει τις πρώτες ύλες στέλνει Checked Raw Materials, αλλιώς μπορεί να στείλει ή κομμάτι κατασκευασμένο ορθά ή κομμάτι κατασκευασμένο λάθος ή κομμάτι που κατά την κατεργασία καταστράφηκε τελείως. Οι τέσσερις αυτές πιθανές εξοδοι συνδέονται με έναν τελεστή αποκλειστικό ή.

### 5.6.5 Διάγραμμα A3



Σχήμα 5.25: Διάγραμμα A3 μετά την προσθήκη σφαλμάτων [18].

Το Assess Completion Stage στο διάγραμμα A3, έχει δύο εξόδους, το Seal και το Assessed Job. Άρα για να τα ξεχωρίσουμε θα χρησιμοποιήσουμε τον λογικό τελεστή και. Το Assessed Job όμως μπορεί εκτός από το να είναι σωστό να είναι λάθος ή ακόμα και να μην αποσταλεί καθόλου. Οπότε για τον λόγο αυτόν θα χρησιμοποιήσουμε και έναν τελεστή αποκλειστικό ή.

Αν δεν αποσταλεί Assessment, τότε η διαδικασία επιστρέφει στο Assess Completion Stage, για να αποσταλεί. Αλλιώς (αποκλειστικό ή ) γίνονται κανονικά οι μετρήσεις των διαστάσεων του κομματιού. Αυτές μπορεί όπως και πριν, να είναι

σωστές, λάθος ή να μην γίνουν καθόλου. Όπως και πριν, έτσι και για αυτή την περίπτωση θα χρησιμοποιήσουμε τελεστή αποκλειστικό ή και προκύπτει η διάταξη που φαίνεται στο σχήμα. Ίδια ακριβώς διάταξη υπάρχει και στην έξοδο του Appraise Work, για τους ίδιους λόγους. Αν δεν έχουν γίνει μετρήσεις η διαδικασία επιστρέφει στο A32, για να γίνουν. Αλλιώς η λειτουργία βγάζει Appraised Job με τις γνωστές τρεις εκδοχές.

Τέλος στο Accept or Reject υπάρχουν οι δύο προφανείς επιλογές, ή το κομμάτι να γίνει αποδεκτό είτε όχι. Με την χρήση του αποκλειστικού ή, διασφαλίζουμε ότι μία μόνο από τις δύο εξόδους θα είναι η απόφαση που θα παρθεί.

### 5.7 ΕΛΕΓΧΟΣ ΜΕΣΑ ΣΤΑ ΔΙΑΓΡΑΜΜΑΤΑ ΓΙΑ ΠΙΘΑΝΑ ΣΦΑΛΜΑΤΑ

Κάθε διάγραμμα ενός IDEF0 μοντέλου, εκτός από το A-0, αποτελεί το child diagram μιας λειτουργίας. Άρα στην ουσία έχουμε μια σειρά διαγραμμάτων που αναπαριστούν μια σειρά από λειτουργίες. Κατά την διάρκεια αυτών των λειτουργιών μπορεί να προκληθούν σφάλματα εξαιτίας, είτε γνωστών παραγόντων που είναι πιθανόν να συμβούν είτε εξαιτίας αστάθμητων παραγόντων. Για να δημιουργήσουμε έναν εσωτερικό έλεγχο σε κάθε διάγραμμα (ανατροφοδότηση), προκειμένου τέτοιες περιπτώσεις σφαλμάτων να ελέγχονται, ζητάμε κάθε ένα ξεχωριστό διάγραμμα, ενός συνολικού μοντέλου, να ακολουθεί μια λογική τριών σταδίων. Έλεγχο, εκτέλεση, έλεγχο – απόφαση. Σκοπός είναι να φαίνεται στα διαγράμματα ότι εξετάζεται η πιθανότητα κάποιου σφάλματος και κατά την προσομοίωση αυτά να υπολογίζονται, προκειμένου να εξάγονται χρήσιμα συμπεράσματα.

Το πρώτο στάδιο ελέγχει την κατάσταση στην οποία βρίσκεται η διαδικασία (όποια και να είναι η διαδικασία, π.χ. γραμμή παραγωγής, λειτουργία ενός μηχανήματος, λήψη μιας απόφασης) και ποιες είναι οι ενέργειες που πρέπει να γίνουν στο πλαίσιο της λειτουργίας. Στη συνέχεια στέλνει πληροφορίες στα άλλα κουτιά – λειτουργίες, μέσω control βελών, για το πώς αυτά πρέπει να ενεργήσουν. Συνήθως το πρώτο στάδιο του ελέγχου αποτελείται από ένα μόνο κουτί. Στο δεύτερο στάδιο εκτελείται η λειτουργία του child διαγράμματος σύμφωνα με τις προδιαγραφές που υπάρχουν. Σε αυτό το στάδιο συνήθως συμπεριλαμβάνονται παραπάνω από ένα κουτιά. Η έξοδος (output) του τελευταίου κουτιού (ή του μοναδικού αν είναι ένα)

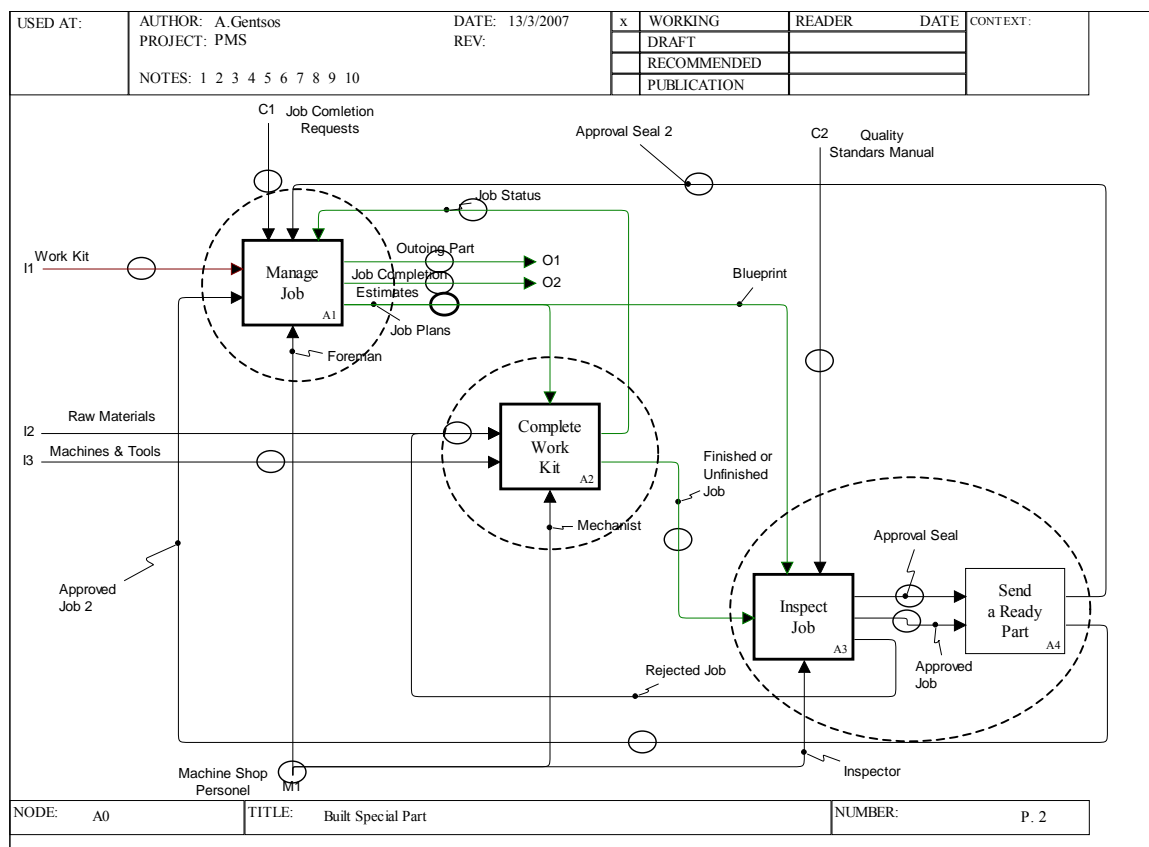


αποτελεί την είσοδο (input) για το επόμενο στάδιο. Στο τρίτο και τελευταίο στάδιο υπάρχει ένας νέος έλεγχος που εξετάζει αν όλα πήγαν καλά κατά την διάρκεια της εκτέλεσης. Μετά τον έλεγχο μπορεί να υπάρχει είτε ανατροφοδότηση (αν διαπιστωθεί κάποιο σφάλμα) η οποία θα πηγαίνει σαν είσοδος στον πρώτο έλεγχο είτε θα αποτελεί έναν τυπικό έλεγχο και θα συνεχίζει σαν έξοδος ολόκληρου του διαγράμματος. Η διαφορά του πρώτου ελέγχου από τον τελευταίο είναι ότι στον πρώτο ελέγχεται τι και πως πρέπει να υλοποιηθεί στην εκτέλεση, ενώ στον τελευταίο ελέγχεται αν όλα πήγαν καλά και *αποφασίζεται* τι πρέπει να γίνει.

Κάθε στάδιο μπορεί να περιέχει μια ή παραπάνω λειτουργίες – κουτιά. Θα πρέπει πάντα να τηρείται, για λόγους σαφήνειας των διαγραμμάτων, ο περιορισμός της μεθόδου ότι κάθε διάγραμμα πρέπει να περιέχει από τρία έως έξι κουτιά. Για αυτό το λόγο, αν για τον σχηματισμό των τριών σταδίων απαιτούνται παραπάνω από έξι κουτιά θα πρέπει να δημιουργούνται εκ' νέου υποδιαγράμματα (child diagram). Στην περίπτωση δηλαδή που απαιτούνται εφτά κουτιά για να σχηματιστούν αυτά τα τρία στάδια, θα υπάρχει ένα στάδιο που θα περιλαμβάνει τουλάχιστον τρία και άρα θα μπορεί να γίνει ξεχωριστό διάγραμμα. Σε αντίθετη περίπτωση (αν δηλαδή τα κουτιά είναι έξι ή λιγότερα) δεν υπάρχει λόγος σχηματισμού κάποιου child diagram. Αυτό θα μπορούσε να συμβεί μόνο σε περίπτωση που το διάγραμμα είναι πολύ πολύπλοκο και κάποιο στάδιο περιλαμβάνει τρία ή τέσσερα κουτιά. Ουσιαστικά δηλαδή δημιουργία child diagram δεν θα πρέπει να επιδιώκεται, χωρίς να αντιπροσωπεύει μια εξ' ολοκλήρου ξεχωριστή λειτουργία, παρά μόνο στην περίπτωση που αυτό επιβάλλεται προκειμένου να υπάρχει μεγαλύτερη σαφήνεια.

Παρακάτω ακολουθούν τα διαγράμματα που δείχνουν πως αυτή η λογική εφαρμόζεται.

## Παραδείγματα εφαρμογής

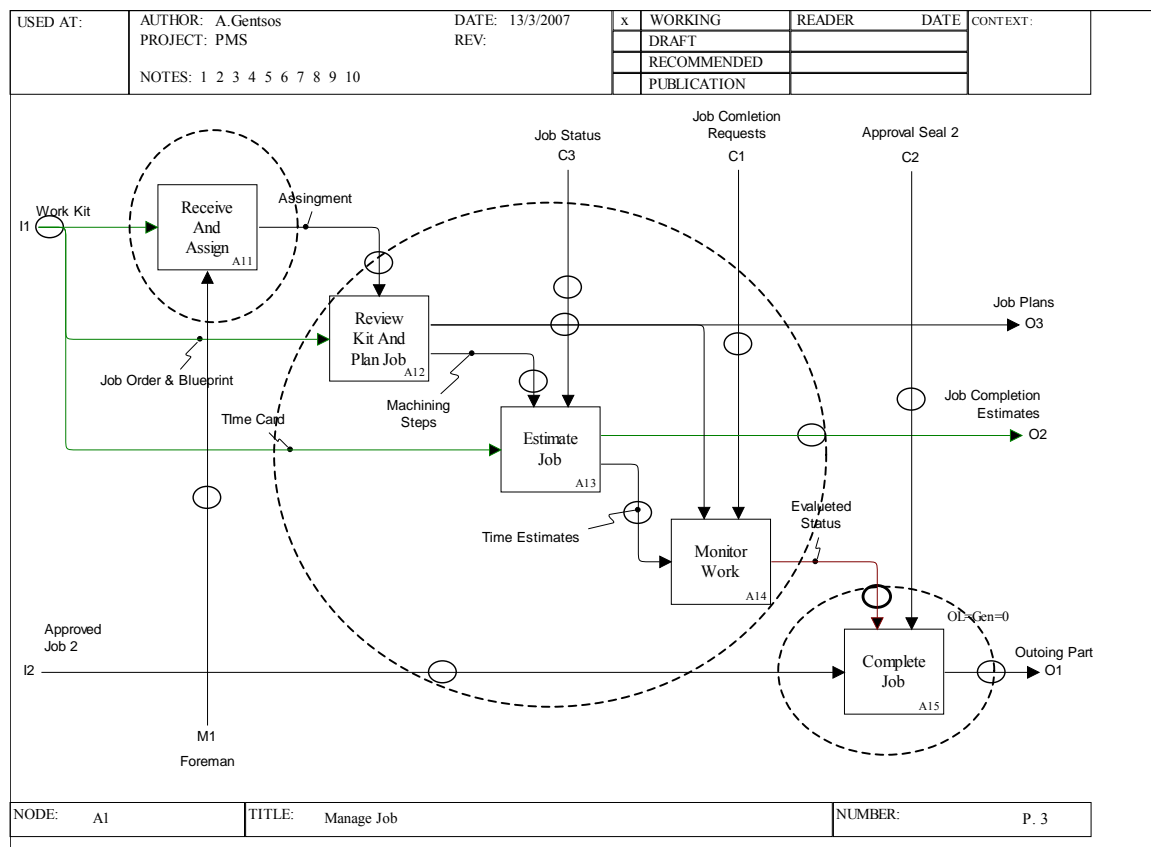


Σχήμα 5.26: *Build Special Part*, A0 διάγραμμα.

Το A0 διάγραμμα του παραδείγματος που αναλύουμε είναι μια χαρακτηριστική εφαρμογή της λογικής που αναφέρθηκε παραπάνω. Το A1 κουτί (Manage Job) αποτελεί τον πρώτο έλεγχο, που στέλνει πληροφορίες μέσω των control βελών τόσο στο A2 κουτί (Complete Work Kit), όσο και στο A3 κουτί (Inspect Job). Το Complete Work Kit αποτελεί το κομμάτι της εκτέλεσης της λειτουργίας (στην συγκεκριμένη περίπτωση και όλου του μοντέλου κιάλας). Δέχεται, εκτός από τις πρώτες ύλες και τα εργαλεία, πληροφορίες από το Manage Job για το πώς πρέπει να ενεργήσει σύμφωνα με τις απαιτήσεις που υπάρχουν. Τέλος το τρίτο μέρος αποτελείται από τα κουτιά A3 και A4 (Send a Ready Part). Στο τρίτο μέρος ελέγχεται το προϊόν που δημιουργήθηκε από το Complete Work Kit και αποφασίζει είτε να απορριφθεί και επιστρέφει για επανεπεξεργασία στο Complete Work Kit, είτε οδηγείται στον αρχικό έλεγχο για να εξέλθει από το σύστημα. Για αυτόν το σκοπό

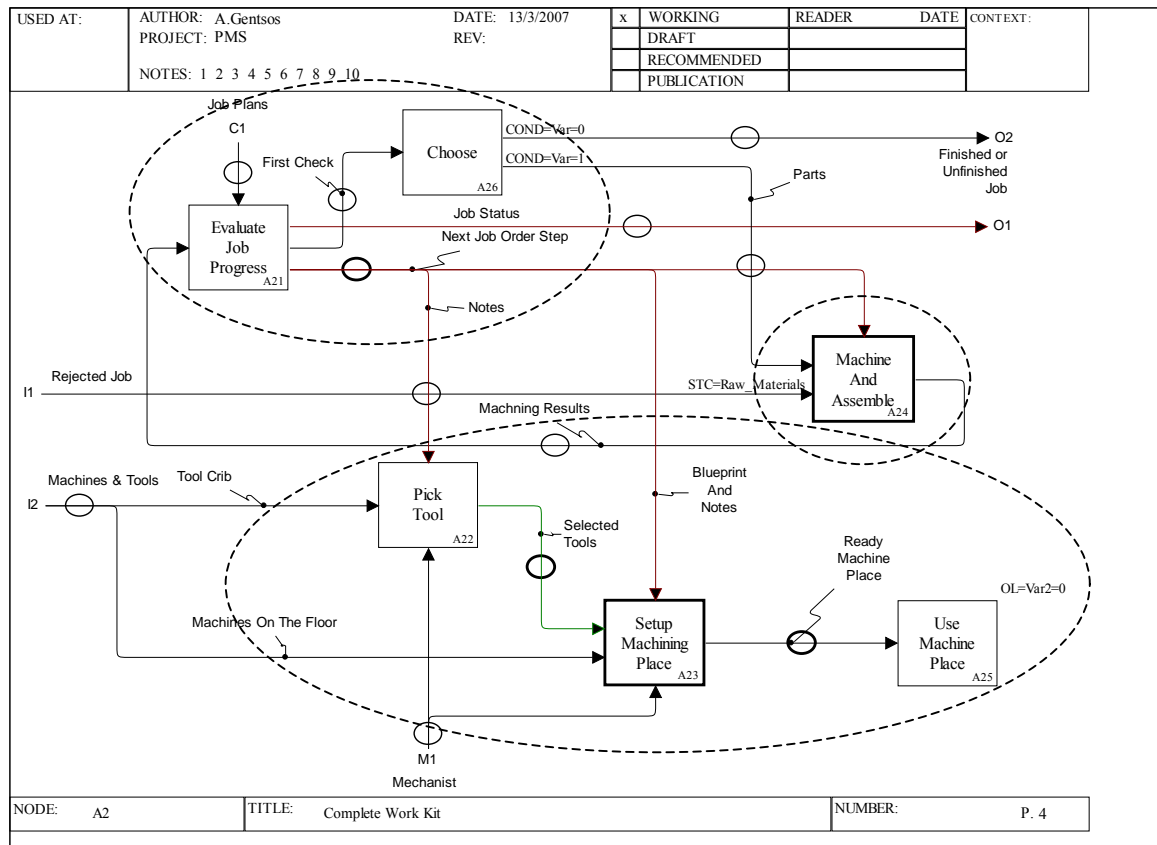
(του ελέγχου) το Manage Job στέλνει τα σχέδια (Blueprint) για να μπορέσει να γίνει ο νέος έλεγχος και να παρθεί η απόφαση.

Το A4 κουτί (Send a Ready Part) έχει προστεθεί για να δίνονται πιο λογικά αποτελέσματα από την προσομοίωση και αποτελεί λογική συνέχεια του A3 για αυτό συμπεριλαμβάνεται στο στάδιο έλεγχος – απόφαση.



Σχήμα 5.27: *Build Special Part*, A1 διάγραμμα

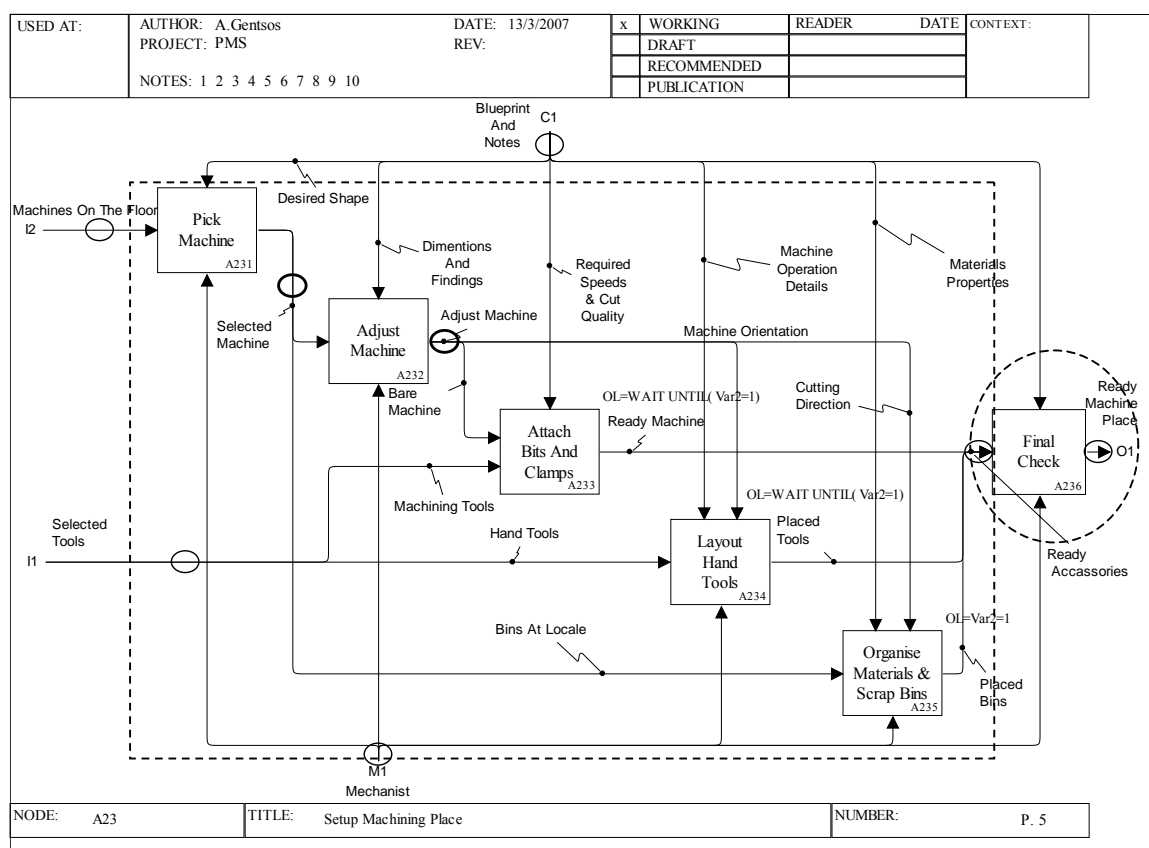
Στο A2 διάγραμμα έχουμε σαν πρώτο στάδιο έλεγχου το A11 κουτί (Receive and Assign) το οποίο αναθέτει τις λειτουργίες που πρέπει να γίνουν κατά την εκτέλεση. Το στάδιο της εκτέλεσης περιλαμβάνει τα κουτιά A12, A13 και A14. Η έξοδος αυτού του σταδίου παρέχει την απαιτούμενη γνώση ώστε να αποδεχθεί το κομμάτι στο τρίτο στάδιο το κουτί A15 (Complete Job) και να το διώξει από το σύστημα. Το στάδιο της εκτέλεσης παρέχει και άλλες πληροφορίες στο σύστημα.



Σχήμα 5.28: Build Special Part, A2 διάγραμμα

Το διάγραμμα A2 περιέχει τα κουτιά A22, A23 και A25 που θα μπορούσαν να αποτελούν ένα ξεχωριστό child diagram. Αυτό όμως δεν γίνεται καθώς θα αποτελούταν μόνο από δύο κουτιά (αρχικά υπήρχαν μόνο τα A22 και A23, αλλά στη συνέχεια προστέθηκε και το A25 για να τρέχει η προσομοίωση), τα οποία ουσιαστικά αποτελούν τον μηχανισμό του A24 (Machine and Assemble). Σε αυτό το διάγραμμα έχουμε το εξής παράδοξο, το A21 (Evaluate Job Progress) κουτί μαζί με το A26 (Choose) είναι ταυτόχρονα έλεγχος και πρώτου σταδίου αλλά και τρίτου. Πιο αναλυτικά το Evaluate Job Progress δέχεται αρχικά τις πρώτες ύλες από το Machine and Assemble, τις αξιολογεί και στη συνέχεια τις στέλνει ξανά στο Machine and Assemble για να τις επεξεργαστεί σύμφωνα με τα δεδομένα που του στέλνει ταυτόχρονα (Next Job Order Step) (πρώτο στάδιο – έλεγχος). Το δεύτερο στάδιο της εκτέλεσης αποτελείται από το A24 (Machine and Assemble) στο οποίο λαμβάνει χώρα η κατεργασία των πρώτων υλών ή των απορριφθέντων από το κουτί A3 (Inspect Job). Τέλος το κατεργασμένο κομμάτι ξανά περνάει από έλεγχο στο Evaluate

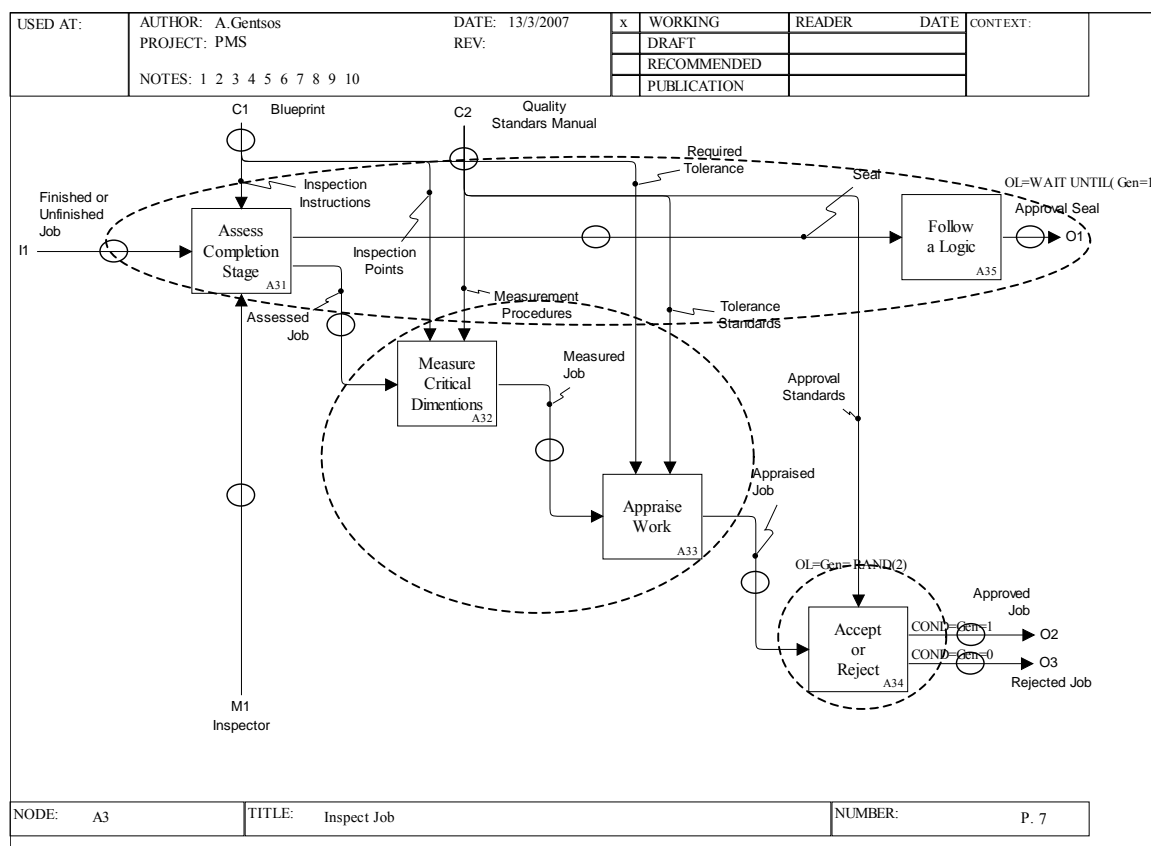
Job Progress και από εκεί κατευθύνεται στο κουτί A3 (Inspect Job) (τρίτο στάδιο, έλεγχος – απόφαση).



Σχήμα 5.29: Build Special Part, A23 διάγραμμα

Προηγουμένως συναντήσαμε ένα διάγραμμα που στο ίδιο σημείο εκτελούνταν δύο στάδια (οι δύο έλεγχοι). Στο διάγραμμα A23 βλέπουμε κάτι άλλο, αρχικά δεν υπάρχει κάποιο σημείο στο οποίο να γίνεται ο έλεγχος του τρίτου σταδίου. Για τον σκοπό αυτό προστέθηκε το κουτί A236 (Final Check) όπου εκτελείται το τρίτο στάδιο. Επίσης παρατηρούμε ότι δεν υπάρχει ούτε κουτί που να εκτελεί τον αρχικό έλεγχο. Με πιο προσεκτική ματιά όμως φαίνεται ότι το ρόλο αυτόν τον πληροί το κουτί A21 (Evaluate Job Progress) του parent διαγράμματος, που στέλνει μέσω control βελών πληροφορίες στα κουτιά του διαγράμματος (A231 (Pick Machine), A232 (Adjust Machine) A233 (Attach Bits and Clamps, A234 (Layout Hand Tools) και A235 (Organize Materials and Scrap Bins)) για το πώς αυτά πρέπει να ενεργήσουν. Άρα είναι προφανές ότι η παραπάνω ομάδα κουτιών αποτελεί το στάδιο της εκτέλεσης. Το κουτί A236 (Final Check) αποτελεί το τρίτο στάδιο που ελέγχεται αν όλα πήγαν καλά. Εδώ απλά το output εξέρχεται του διαγράμματος, όμως θα

μπορούσε να υπάρχει και ένα βέλος ανατροφοδότησης, που θα εξέφραζε την περίπτωση που ο έλεγχος έδειξε κάποιο πρόβλημα και θα επέστρεφε σε κάποιο προηγούμενο σημείο. Αυτό θα μπορούσε να γίνει είτε μέσω της χρήσης βάρους είτε με την χρήση κάποιας κατανομής.



Σχήμα 5.30: Build Special Part, A3 διάγραμμα

Στο διάγραμμα A3 (Inspect Job), τα κουτιά A31 (Asses Completion Stage) και A35 (Follow a Logic) αποτελούν τον έλεγχο, κάνοντας αρχικά μια αποτίμηση το κουτί A31 και παρέχοντας αυτή την πληροφορία στο στάδιο της εκτέλεσης αλλά και σαν πληροφορία για την αποδοχή του κομματιού στο Manage Job (A1). Ουσιαστικά το κουτί A31 δεν παίζει κάποιο ρόλο, αλλά έχει προστεθεί για να δίνει λογικά αποτελέσματα η προσομοίωση (η αιτιολόγηση δεν είναι του παρόντος και έχει δοθεί νωρίτερα). Τα κουτιά A32 (Measure Critical Dimensions) και A33 (Appraise Work) αποτελούν το στάδιο της εκτέλεσης και η έξοδός του παρέχεται, μέσω control βέλους, σαν πληροφορία για το επόμενο στάδιο του ελέγχου. Στο τρίτο και τελευταίο στάδιο

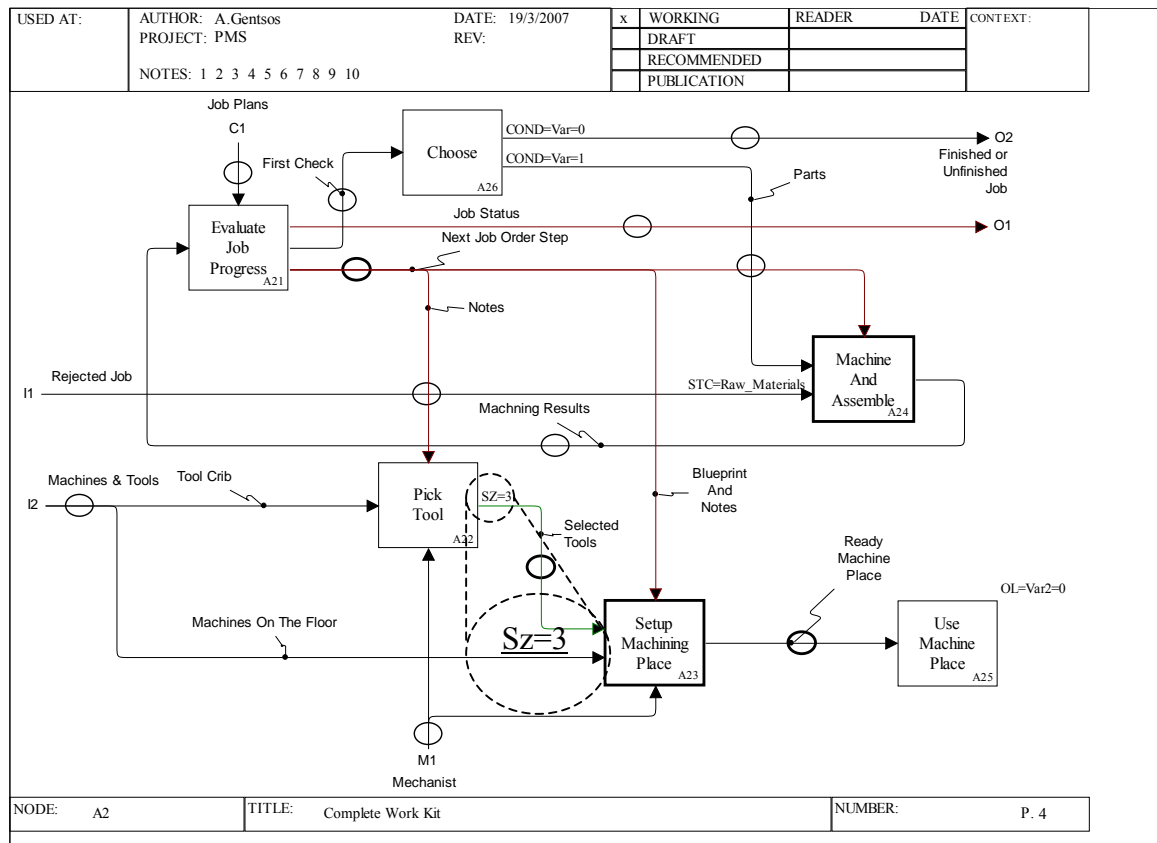
παίρνεται η απόφαση αν θα γίνει ή όχι αποδεκτό το κομμάτι. Σε περίπτωση που το κομμάτι απορριφθεί επιστρέφει στο Complete Work Kit (A2) για να επεξεργαστεί εκ νέου.

Το child διάγραμμα του Machine and Assemble, δημιουργήθηκε για να βοηθήσει στο να δώσει σωστά αποτελέσματα η προσομοίωση και ως εκ τούτου δεν υπήρξε λόγος να γίνει χρήση της ανωτέρω λογικής.

## 5.8 ΑΡΙΘΜΟΣ ΠΛΗΡΟΦΟΡΙΩΝ Η ΚΟΜΜΑΤΙΩΝ ΑΝΑ ΒΕΛΟΣ

Ο αριθμός των στοιχείων που περιέχει ένα βέλος μπορεί να ποικίλει από σημείο σε σημείο στο διάγραμμα. Αν υποθέσουμε ότι ένα βέλος μεταφέρει κάποια εξαρτήματα, θα ήταν χρήσιμο να φαίνεται πάω στο διάγραμμα πόσα είναι αυτά. Επίσης είναι πιθανόν ο αριθμός των εξαρτημάτων που φτάνουν σε μια λειτουργία να διαφέρει από τον αριθμό των εξαρτημάτων που βγαίνουν από αυτή. Κάτι τέτοιο θα μπορούσε να συμβαίνει αν για παράδειγμα η λειτουργία επιλέγει κάποια από αυτά τα εξαρτήματα, για να χρησιμοποιηθούν σε μια άλλη παρακάτω λειτουργία. Άρα για καλύτερη κατανόηση ενός διαγράμματος είναι λογικό να υπάρχει μια πληροφορία στο βέλος, πριν και μετά από κάθε λειτουργία, για την ποσότητα των κομματιών ή των πληροφοριών που περιέχονται σε καθένα από αυτά.

Μέσω του Workflow Modeler, υπάρχει επιλογή που δηλώνεται ο αριθμός των στοιχείων που περιέχονται στο βέλος. Η επιλογή αυτή εμφανίζεται ως size και η τιμή της φαίνεται και πάνω στο διάγραμμα, στο σημείο εκκίνησης κάθε βέλους, με την ένδειξη sz = αριθμός στοιχείων. Στο παρακάτω διάγραμμα έχουμε ένα παράδειγμα για το πώς αναπαριστάται η ένδειξη size πάνω στο διάγραμμα. Αναλυτικά, με δεδομένο ότι αρχικά τα εργαλεία είναι έξι (δηλαδή το βέλος Tool Crib μεταφέρει έξι εργαλεία) βλέπουμε ότι στην λειτουργία Pick Tool (A22) επιλέγονται τρία από αυτά που είναι κατάλληλα για την κατεργασία, σύμφωνα με τις πληροφορίες που έρχονται από το Evaluate Job Progress (A31). Έτσι στο βέλος εξόδου της λειτουργίας Pick Tool (A22), βλέπουμε στην αρχή του την ένδειξη sz = 3, Που σημαίνει ότι το βέλος Selected Tools περιέχει τρία εξαρτήματα.



Σχήμα 5.31: Εμφάνιση ένδειξης size πάνω στο διάγραμμα [19].

Ακόμα πιο καλό όμως θα ήταν, αυτή η πληροφορία για τον αριθμό των στοιχείων κάθε βέλους να γινόταν φανερή στην αρχή αλλά και στο τέλος κάθε βέλους. Επειδή ένα βέλος μπορεί μέσα στο διάγραμμα να διανεύει μεγάλη απόσταση και επιπρόσθετα να υπάρχουν πολλά βέλη, πράγμα που καθιστά το διάγραμμα δυσανάγνωστο, θα ήταν πιο σωστό να μην χρειάζεται να ανατρέχει ο αναγνώστης το διάγραμμα στην αρχή του βέλους (ή ανάποδα) για να βρει τον αριθμό των στοιχείων που περιέχει. Δεν είναι απαραίτητο τέτοια πληροφορία να υπάρχει σε κάθε βέλος, για παράδειγμα βέλη που μεταφέρουν σχέδια είναι αυτονόητο ότι μεταφέρουν μόνο ένα σχέδιο. Η πληροφορία για τον αριθμό των στοιχείων ενός βέλους, είναι χρήσιμη για βέλη που αφορούν, κομμάτια, εξαρτήματα ή εργαλεία.

Τέλος η χρήση της επιλογής size μπορεί να μας δώσει χρήσιμες πληροφορίες για την παραγωγή, τους χρόνους και τα κόστη κατά την προσομοίωση της διαδικασίας.



## 5.9 ΣΦΑΛΜΑΤΑ ΠΟΥ ΜΠΟΡΕΙ ΝΑ ΠΡΟΚΥΨΟΥΝ ΣΤΟ ΜΟΝΤΕΛΟ

Εδώ παρουσιάζονται αναλυτικά τα σφάλματα που είναι πιθανό να εμφανιστούν σε κάθε μία από τις λειτουργίες του μοντέλου. Αναφέρονται με την σειρά που εμφανίζονται στα διαγράμματα οι λειτουργίες.

### Διάγραμμα A1 Manage Job

#### A11: Receive and Assign

- Λάθος ανάθεση εργασιών.
- Μη ανάθεση εργασιών.
- Λάθος ανάθεση ορισμένων εργασιών.
- Μη ανάθεση ορισμένων εργασιών.
- Ανάθεση περιττών εργασιών.
- Ανάθεση εργασιών που οδηγούν σε καταστροφή κομματιού.

#### A12: Review Kit and Plan Job

- Λάθος σειρά βημάτων κατασκευής κομματιού.
- Μη αποστολή σειράς βημάτων κατασκευής κομματιού.
- Μη υπόδειξη κάποιων βημάτων κατασκευής κομματιού.
- Λάθος υπόδειξη διαστάσεων και ανοχών.
- Λάθος σχέδια.

#### A13: Estimate Job

- Κακός υπολογισμός χρόνου κατεργασίας κομματιού.
- Κακός υπολογισμός χρόνου ολοκλήρωσης της διαδικασίας.
- Μη υπολογισμός χρόνου κατεργασίας.
- Μη υπολογισμός χρόνου ολοκλήρωσης της διαδικασίας.
- Λάθος χρονικής αξιολόγησης κάποιων βημάτων.
- Λάθος χρονικού υπολογισμού λόγω λανθασμένων πληροφοριών που στέλνει το Job Status.

#### A14: Monitor Work

- Λάθος αξιολόγηση.
- Μη αποστολή αξιολόγησης.
- Σωστή αξιολόγηση αλλά βάση λανθασμένων δεδομένων που στέλνει το Job Status.

#### A15: Complete Job

- Μη αποστολή ολοκληρωμένου κομματιού.
- Αποδοχή ελαττωματικού κομματιού.

## Διάγραμμα A2 Complete Work Kit

### A21 Evaluate Job Progress

- Πιθανότητα αξιολόγησης ως καλού ένα σκάρτο κομμάτι.
- Αποστολή λανθασμένων σχεδίων.
- Πιθανότητα αποδοχής ενός κομματιού / πρώτης ύλης πριν αυτό υποστεί κατεργασία.
- Λάθος αποστολή πληροφοριών για την κατάσταση στην οποία βρίσκεται η κατεργασία (Job Status).
- Λάθος αποστολή Notes ή Blueprints ή Next Job Order.

### A22 Pick Tool

- Μη επιλογή εργαλείου.
- Λάθος εργαλεία.
- Λάθος αριθμός εργαλείων.
- Λάθος επιλογή χειροκίνητων εργαλείων (Hand Tools).
- Μη επιλογή χειροκίνητων εργαλείων (Hand Tools).
- Λάθος επιλογή εργαλείων για την εργαλειομηχανή (Machining Tools).

- Μη επιλογή εργαλείων για την εργαλειομηχανή (Machining Tools).
- Περισσότερα εργαλεία (προκαλούν καθυστέρηση).

#### Διάγραμμα A23 Setup Machining Place

##### A231: Pick Machine

- Επιλογή λάθος εργαλειομηχανής.
- Μη επιλογή εργαλειομηχανής.
- Επιλογή χαλασμένης εργαλειομηχανής.
- Μη διαθεσιμότητα εργαλειομηχανής λόγω εργασιών συντήρησης.

##### A232: Adjust Machine

- Λάθος ρύθμιση κοπτικού.
- Λάθος ρύθμιση ατράκτου.
- Λάθος ρύθμιση διαστάσεων κοπής (βάθος κοπής).
- Λάθος ρύθμιση ταχυτήτων μηχανής.
- Λάθος ρύθμιση παράλληλης κοπής (πρόωση).
- Ρυθμίσεις για κατασκευή άλλου σχεδίου.

- Καμία ρύθμιση.

#### A233: Attach Bits and Clamps

- Λάθος επιλογή κοπτικού εξαιτίας λάθους πληροφοριών από Blueprints and Notes.
- Λάθος επιλογή κοπτικού εξαιτίας λάθους από το Machining Tools.
- Λάθος επιλογή υλικού κοπτικού.
- Λάθος επιλογή κοπτικού από γεωμετρικής άποψης.
- Μη επιλογή κοπτικού.
- Κακή επιλογή σφιχτήρα.
- Μη επιλογή σφιχτήρα.

#### A234: Layout Hand Tools

- Λάθος επιλογή χειροκίνητων εργαλείων.
- Μη διαθεσιμότητα απαιτούμενων εργαλείων.
- Λάθος επιλογή λίμας.
- Μη επιλογή χειροκίνητων εργαλείων.
- Περιττά εργαλεία (πιθανότητα λάθους επιλογείς αλλά και χαμένου χρόνου).

#### A235: Organize Materials and Scrap Bins

- Μη τοποθέτηση εργαλείων.
- Κακή τοποθέτηση εργαλείων (σε λάθος θέση).
- Τοποθέτηση περιττών εργαλείων (επιφέρει καθυστέρηση).
- Έλλειψη κάποιων εργαλείων.
- Κακή διάταξη (σειρά κουτιών).
- Τοποθέτηση κουτιών σε δυσπρόσιτη θέση.

#### Διάγραμμα A24 Machine and Assemble

#### A243: Assemble

- Λάθος κατασκευή κομματιού.
- Λάθος διαστάσεις.
- Καταστροφή κομματιού.
- Ελαττωματικό αλλά επανακατεργάσιμο κομμάτι.
- Κακή ποιότητα επιφάνειας.
- Μπλοκάρισμα μηχανής και μη κατεργασία κομματιού.

### Διάγραμμα A3 Inspect Job

#### A31: Asses Completion Stage

- Λάθος καθορισμός εργασιών ελέγχου.
- Μη καθορισμός εργασιών ελέγχου.
- Μη καθορισμός ορισμένων εργασιών ελέγχου.
- Καθορισμός εργασιών ελέγχου για άλλο είδος παραγόμενου κομματιού.

#### A32: Measure Critical Dimensions

- Λάθος σημεία ελέγχου.
- Λάθος μέτρηση.
- Σωστή μέτρηση, αλλά σύγκριση με λάθος πρότυπα.
- Λάθος διαδικασία μέτρησης.
- Μετρήσεις όχι σε όλα τα σημεία ελέγχου.
- Μη μετρήσεις.

#### A33 Appraise Work

- Λάθος αποτίμηση εργασίας.
- Μη αποτίμηση της εργασίας.

- Λάθος απαραίτητων ανοχών.
- Λάθος μέτρηση ανοχών.
- Λάθος έλεγχος ανοχών σε κάποια σημεία.
- Μη έλεγχος ανοχών σε κάποια σημεία.
- Μη έλεγχος ανοχών.

#### A34: Accept or Reject

- Αποδοχή ενός σκάρτου κομματιού εξαιτίας κάποιου σφάλματος που δεν εντοπίστηκε.
- Αποδοχή ενός σκάρτου κομματιού εξαιτίας λάθους από τα στάνταρτ.
- Αποδοχή ενός σκάρτου κομματιού εξαιτίας λάθος εφαρμογής των στάνταρτ αποδοχής.
- Απόρριψη ενός καλού κομματιού εξαιτίας λάθους από τα στάνταρτ.
- Απόρριψη ενός καλού κομματιού εξαιτίας λάθος εφαρμογής των στάνταρτ αποδοχής.



## **ΚΕΦΑΛΑΙΟ 6**

### **ΣΥΜΠΕΡΑΣΜΑΤΑ ΠΕΡΙΟΡΙΣΜΟΙ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ**

#### **6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ**

Επιλέξαμε να ασχοληθούμε με την γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών IDEF0. Πρόκειται για μια γλώσσα εξαιρετικά δημοφιλή, με ευρεία αποδοχή και ιδιαίτερα αποτελεσματική. Μέσα από την βιβλιογραφία διαπιστώσαμε, ότι οι περισσότεροι συγγραφείς εντοπίζουν το πόσο διαδεδομένη είναι η γλώσσα και ότι είναι η πλέον κατάλληλη για την επικοινωνία στην περιγραφή διαδικασιών στον επιχειρηματικό χώρο. Επίσης παρατηρήσαμε ότι αυτή η βιβλιογραφία έχει τις αρχές της κοντά στα πρώτα χρόνια που δημιουργήθηκε η μέθοδος και φτάνει μέχρι σήμερα σε σύγχρονες μελέτες. Αυτό δείχνει ότι πρόκειται για μία γλώσσα που έχει διάρκεια στον χρόνο και παρά το πέρασμά του συνεχίζει να είναι εξαιρετικά χρηστική. Η από πάνω προς τα κάτω ιεραρχική ανάλυση μαζί με τα semantics είναι τα πιο σημαντικά σημεία της μεθόδου, όπως επίσης και οι ICOM συμβάσεις που οδηγούν σε ευκολία επικοινωνίας.

Όμως με την πάροδο των χρόνων, όπως είναι λογικό, έχουν διαπιστωθεί κάποιες ελλείψεις της μεθόδου. Ένα σημαντικό μειονέκτημα της μεθόδου, είναι η τάση που υπάρχει να δημιουργούνται διαγράμματα ιδιαίτερα σύνθετα και δυσνόητα. Επίσης τα διαγράμματα της IDEF0, δεν περιλαμβάνουν μοντελοποίηση σφαλμάτων και καθυστερήσεων. Εμείς, εκκινήσαμε με σκοπό την βελτίωση της γλώσσας με την μοντελοποίηση των σφαλμάτων και των καθυστερήσεων, στη πορεία όμως διαπιστώσαμε πως θα ήταν χρήσιμο να ασχοληθούμε και με άλλες δύο περιπτώσεις. Αυτές είναι οι παράλληλες διαδικασίες που στην IDEF0 δεν είναι ευδιάκριτες και τα κομμάτια που περιέχονται ανά βέλος. Έτσι παρουσιάσαμε μια ολοκληρωμένη έρευνα γύρω από την IDEF0 και τις βελτιώσεις που κατά την γνώμη μας μπορούν να γίνουν.

Για να αναπτύξουμε την ερευνητική μας προσπάθεια, χρησιμοποιήσαμε ένα ήδη υπάρχον μοντέλο, που περιέχεται στο βιβλίο SADT των David A. Marca και

Clement L. McGowan. Πρόκειται για την κατασκευή ενός εξαρτήματος σε ένα μηχανουργείο. Το μοντέλο παίρνει σαν δεδομένο τις πρώτες ύλες, τα μηχανήματα και το σε τι κατάσταση πρέπει να βρεθεί το κομμάτι και δίνει σαν έξοδο το τελικό εξάρτημα. Οφείλουμε να προσθέσουμε ότι το παράδειγμα που χρησιμοποιούμε, είναι ένα από τα πλέον κατάλληλα, μιας και IDEF0 ενδείκνυται ιδιαίτερα για μοντελοποίηση παραγωγικών διαδικασιών.

Ένα σφάλμα σε μια επιχειρησιακή διαδικασία, όσο πιο αργά εντοπιστεί τόσο μεγαλύτερο θα είναι το κόστος που θα προκύψει. Αυτό το κόστος έχει να κάνει εν πρώτης με ζημιά που μπορεί να έχει προκαλέσει το σφάλμα, αλλά και με το κόστος επιδιόρθωσης. Πρόσθετο κόστος μπορεί να δημιουργήσει η καθυστέρηση που θα προκύψει λόγω της επιδιόρθωσης του σφάλματος. Επίσης θα έχει πάει χαμένος ο μέχρι εκείνη την στιγμή σχεδιασμός. Για αυτούς τους λόγους θεωρούμε ότι είναι σημαντικό, τα πιθανά σφάλματα κατά την μοντελοποίηση επιχειρησιακών διαδικασιών να προβλέπονται. Ένα σφάλμα, σχεδόν σε όλες τις περιπτώσεις, είναι δυνατόν να εμφανιστεί σε όλες τις λειτουργίες μιας διαδικασίας. Ένα σφάλμα, είναι δυνατόν να είναι προβλέψιμο (αιτιοκρατικό) ή μη προβλέψιμο (στοχαστικό). Τα σφάλματα μπορούν να προκαλούνται από ανθρώπους (τότε σπάνια μπορούν να προβλεφθούν) ή από μηχανές (που σε μερικές περιπτώσεις είναι δυνατόν να προβλεφθούν). Για την μοντελοποίηση των σφαλμάτων στην γλώσσα IDEF0, χρησιμοποιήσαμε τους λογικούς τελεστές που χρησιμοποιούν τα EPC συστήματα. Χρησιμοποιήσαμε τον λογικό τελεστή “και” και τον λογικό τελεστή “αποκλειστικό ή”. Θεωρήσαμε ότι κάθε λειτουργία μπορεί να δώσει τριών ειδών αποτελέσματα, το επιθυμητό, ένα εσφαλμένο ή να μην δώσει καθόλου αποτέλεσμα λόγω σφάλματος της λειτουργίας. Για να διαχωρίσουμε ότι μπορεί να συμβεί μόνο μία από αυτές τις περιπτώσεις, χρησιμοποιήσαμε τον λογικό τελεστή “αποκλειστικό ή”. Επίσης θεωρήσαμε, ότι το αποτέλεσμα της εσφαλμένης λειτουργίας θα γίνεται φανερό στην επόμενη. Ο τελεστής “και” χρησιμοποιήθηκε για να δειχθεί ότι και οι δύο εξοδοί της λειτουργίας εξέρχονται ταυτόχρονα. Πρέπει να σημειώσουμε ότι οι λογικοί τελεστές στα μοντέλα με τα σφάλματα δεν χρησιμοποιήθηκαν μόνο για την περιγραφή των σφαλμάτων αλλά για όλο το μοντέλο σαν λογική.

Για την πρόβλεψη των λαθών εισάγαμε και έναν νέο τρόπο μοντελοποίησης. Αυτός βασίστηκε στην στον διαχωρισμό του μοντέλου σε τρία στάδια. Στο πρώτο

στάδιο, που αποτελείται συνήθως από μία λειτουργία, γίνεται ένας έλεγχος των εργασιών που πρέπει να γίνουν και παρέχονται πληροφορίες για το τι κατεργασίες απαιτούνται. Στο δεύτερο στάδιο, που συνήθως αποτελείται από παραπάνω από μία λειτουργίες, γίνονται οι κατεργασίες βάση των πληροφοριών που έστειλε ο πρώτος έλεγχος. Τέλος, στο τρίτο στάδιο ελέγχεται το αποτέλεσμα που προέρχεται από την κατεργασία και ή γίνεται αποδεκτό και βγαίνει από το διάγραμμα ή απορρίπτεται και το κομμάτι γυρνάει σε κάποιο άλλο σημείο του μοντέλου.

Στον σύγχρονο επιχειρηματικό κόσμο υπάρχει ανάγκη να γίνονται όλα γρήγορα. Επειδή όντως ο χρόνος είναι χρήμα και μάλιστα χαμένο, δεν υπάρχει περιθώριο για χαμένους χρόνους. Έτσι εστιάσαμε την προσοχή μας και σε αυτόν τον τομέα. Θεωρήσαμε ότι σε ένα διάγραμμα που περιγράφει μια επιχειρησιακή διαδικασία, θα πρέπει να φανερώνει τα σημεία στα οποία μπορεί να συμβεί μια καθυστέρηση. Ο λόγος είναι ότι πρέπει ο κάθε άγνωστος του μοντέλου, είτε αυτός είναι ο υπεύθυνος manager είτε ο τελευταίος εργάτης, να γνωρίζει τα σημεία που μπορεί να εμφανιστεί μια καθυστέρηση προκειμένου να το έχει υπόψη του ή να κάνει κάτι για αυτό. Για τον σκοπό αυτό, επινοήσαμε ένα δικό μας σύμβολο, με σήμα το ρολόι, που προσαρμόζεται πάνω στο βέλος και συμβολίζει ότι η συγκεκριμένη πληροφορία ή το συγκεκριμένο κομμάτι μπορεί να καθυστερήσει να φτάσει στον προορισμό του.

Σε όλα τα παραδείγματα διαδικασιών είναι δυνατόν να συναντήσουμε το παράδειγμα των παράλληλων διαδικασιών. Δύο λειτουργίες “βαφτίζονται” παράλληλες, αν για κάποιο χρονικό διάστημα εκτελούνται ταυτόχρονα και αν απαιτούνται και οι δύο για να πραγματοποιηθεί μια επόμενη λειτουργία. Στην περίπτωση της γλώσσα IDEF0, ο διαχωρισμός των παράλληλων διαδικασιών από τις μη παράλληλες δεν είναι ευδιάκριτος. Για τον σκοπό αυτόν επινοήσαμε δύο σύμβολα, που το ένα χρησιμοποιείται κατά την ένωση βελών που προέρχονται από παράλληλες διαδικασίες και το άλλο στην αντίθετη περίπτωση. Με αυτόν τον τρόπο πλέον είναι διακριτές οι δύο αυτές περιπτώσεις σε ένα διάγραμμα IDEF0.

Σε όλες τις παραγωγικές διαδικασίες παρατηρούμε ότι εισέρχονται βέλη σε λειτουργίες και εξέρχονται από αυτές, χωρίς να γνωρίζουμε πόσα κομμάτια περιέχονται στο καθένα. Έτσι χάνεται μια πολύτιμη πληροφορία, που θα επέτρεπε να

γνωρίζουμε την ακριβή ενέργεια που κάνει κάθε λειτουργία. Προτάθηκε στα βέλη που μεταφέρουν κομμάτια και όχι πληροφορίες, να αναγράφετε στην αρχή και στο τέλος κάθε βέλους το πόσα κομμάτια περιέχει, σε label. Κάτι τέτοιο πέρα από την χρησιμότητα που έχει σε απλά διαγράμματα, αφού μέχρι τώρα οι πληροφορίες αυτή δεν υπήρχε, δίνει την δυνατότητα ακόμα και πολύπλοκα διαγράμματα να παίρνουμε τέτοιου είδους πληροφορίες. Έτσι με μερικές πληροφορίες ακόμα, το κάθε διάγραμμα παραμένει λυτό και εν τούτοις περιέχει πολλές παραπάνω πληροφορίες. Σε έναν επιχειρηματικό κόσμο που όλα πρέπει να γίνονται όλο και σε λιγότερο χρόνο, ένα σχέδιο που θα δίνει έτοιμες πληροφορίες και δεν θα υποχρεώνει τον αναγνώστη να ξοδέψει χρόνο για να το κατανοήσει και να αντιληφθεί τι συνέπειες μπορεί να έχει κάθε λειτουργία, είναι εξαιρετικά χρήσιμο.

Η IDEF0 είναι μια γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών με διαγράμματα, που όπως είδαμε, μπορούν εύκολα να μετατραπούν σε αντικείμενα μεγάλης πολυπλοκότητας. Εμείς εκτός από την μοντελοποίηση των σφαλμάτων, με τις υπόλοιπες ενέργειες, μας επιδιώξαμε την αποσυμφόρησή τους. Έτσι τα διαγράμματα γίνονται πιο αποτελεσματικά με μικρό έως καθόλου κόστος σύγχυσης του διαγράμματος.

## 6.2 ΠΕΡΙΟΡΙΣΜΟΙ

Αυτό που δεν γίνεται στην παρούσα εργασία είναι η προσομοίωση αυτών των νέων διαγραμμάτων. Αν αυτό μπορούσε να γίνει (χρειάζεται την απαραίτητη τεχνική υποστήριξη) θα ήταν ένα ολοκληρωμένο πακέτο που θα μπορούσε να προβλέπει τα λάθη σε μία διαδικασία και τις συνέπειες που αυτά έχουν, χρονικά και οικονομικά. Επίσης μέσα από την προσομοίωση μπορούν να εξαχθούν αποτελέσματα που εκ των προτέρων δεν είναι δυνατόν να προβλεφθούν.

Φυσικά από την στιγμή που η προσπάθειά μας είναι πρωτοποριακή και τα περισσότερα πακέτα του εμπορίου δεν είναι ανοικτού κώδικα μια τέτοια επιλογή είναι προς το παρόν αδύνατη. Ανοικτού κώδικα λέγονται τα πακέτα software, στα οποία μπορεί κάποιος να επέμβει στον κώδικα που υπάρχει από πίσω τους και να κάνει τις διορθώσεις που θέλει προς όφελος του χρήστη.

Επίσης ένας τρόπος εισαγωγής λαθών στο Workflow Modeler, θα ήταν έπειτα από καθυστερήσεις προηγούμενων λειτουργιών. Δηλαδή η καθυστέρηση μιας προηγούμενης λειτουργίας να έχει σαν συνέπεια, κάποια που την απαιτεί σαν δεδομένο να εμφανίζει σφάλμα. Το Workflow Modeler δίνει την δυνατότητα παραγωγής τυχαίων χρόνων και άρα την επιλογή να πούμε ότι πάνω από έναν χρόνο διαδικασία θεωρείται καθυστερημένη. Όμως κάτι τέτοιο δεν είναι δυνατόν, μιας και ο χρόνος αυτός δεν αποθηκεύεται σε μία μεταβλητή ώστε να την συγκρίνουμε σε κάποιο άλλο σημείο του μοντέλου και να δούμε αν η λειτουργία έχει καθυστερήσει.

Τα control βέλη είναι ξεκάθαρο ότι πρέπει αυστηρά να έχουν φτάσει όλα για να γίνει η διαδικασία. Αυτό όμως δημιουργεί ένα πρόβλημα κατά την προσομοίωση μέσω του Workflow Simulator, αχρηστεύοντας ένα από τα πιο δυνατά όπλα της IDEF0 που είναι η ανατροφοδότηση. Πιο συγκεκριμένα την πρώτη φορά που ενεργεί ένα κουτί που χρησιμοποιεί control βέλος ανατροφοδότησης, όπως είναι λογικό αυτό δεν έχει φτάσει αλλά το πρόγραμμα το περιμένει, γιατί απαιτεί όλα τα βέλη για να εκτελεστεί η διαδικασία. Έτσι η προσομοίωση κολλάει εκεί. Αυτό ο κατασκευαστής θα έπρεπε να το έχει σκεφτεί, από την στιγμή κιόλας που τα βέλη της IDEF αναπαριστούν πληροφορίες ή αντικείμενα και όχι αλληλουχία ενεργειών.

Επίσης άλλα δύο προβλήματα που υπάρχουν στην προσομοίωση της IDEF0 είναι ότι mechanism βέλη δεν μπορούν να προέρχονται από output άλλου κουτιού και ότι τα tunneled βέλη δεν μπορεί να τα χρησιμοποιήσει. Ειδικά για το τελευταίο θα έπρεπε να τα δέχεται και απλά κατά την προσομοίωση να τα χρησιμοποιεί μόνο στο επίπεδο που είναι απαραίτητα, έχοντας πάρει δεδομένα για αυτά από την αρχή.

### 6.3 ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ

Μια εργασία που θα προέβλεπε και την αντιμετώπιση των σφαλμάτων και των χαμένων χρόνων θα ήταν σαφώς πιο ολοκληρωμένη. Όμως ο σκοπός της δουλειάς μας δεν ήταν αυτός. Εμείς θέλαμε απλά να τα μοντελοποιήσουμε πάνω στα πρότυπα της IDEF0, ώστε η γλώσσα να εμπλουτιστεί σε πληροφορίες. Ένα τέτοιο ενδεχόμενο θα μπορούσε αν αποτελέσει αντικείμενο μιας παραπέρα έρευνας.

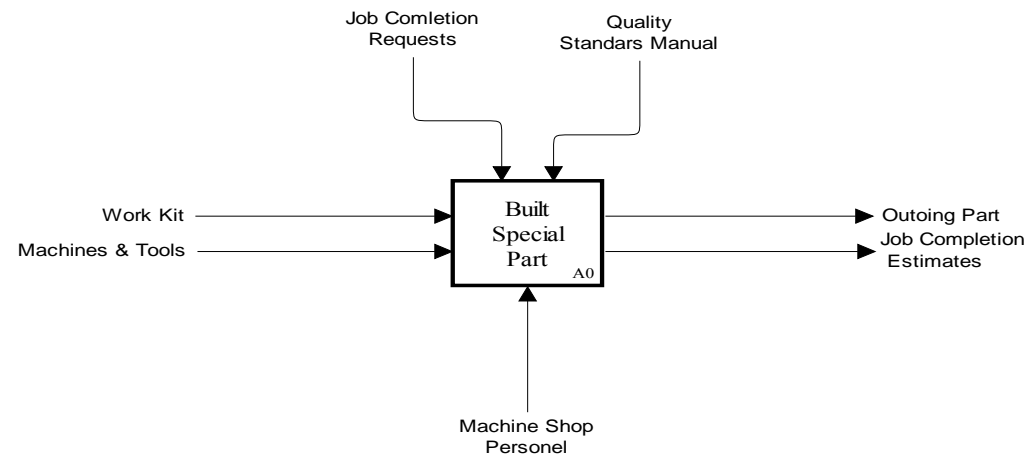
Σε επόμενες εργασίες θα ήταν χρήσιμο να επιχειρηθεί η εισαγωγή στα IDEF0 μοντέλα σφαλμάτων και καθυστερήσεων, μέσα από την χρήση πακέτων

προσομοίωσης και χωρίς την επέμβαση στο κάθε αυτό διάγραμμα. Δηλαδή μέσω κώδικα σε επιλεγμένες λειτουργίες – κουτιά του μοντέλου. Στην λειτουργία που θα θέλουμε να δημιουργηθεί το σφάλμα ή η καθυστέρηση, υπό κάποιες συνθήκες, θα γράφεται ένας κώδικας που θα τα δημιουργεί. Ως εκ τούτου τα αποτελέσματα θα γίνονται φανερά έπειτα από προσομοίωση.

Τέλος πολύ χρήσιμη θα ήταν και χρησιμοποίηση ποιοτικών λαθών και γενικότερα ποιοτικών πληροφοριών σε μοντέλα IDEF0 ή κάποιας άλλης γλώσσας μοντελοποίησης επιχειρησιακών διαδικασιών. Κάτι τέτοιο θα μπορούσε να αποτελέσει αντικείμενο παραπέρα μελέτης πάνω σε διαγράμματα ροής εργασιών.

## **ΠΑΡΑΡΤΗΜΑ**

USED AT:	AUTHOR: A.Gentsos PROJECT: PMS  NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 15/3/2007 REV:	x	WORKING	READER	DATE	CONTEXT: Top
				DRAFT			
				RECOMMENDED			
				PUBLICATION			



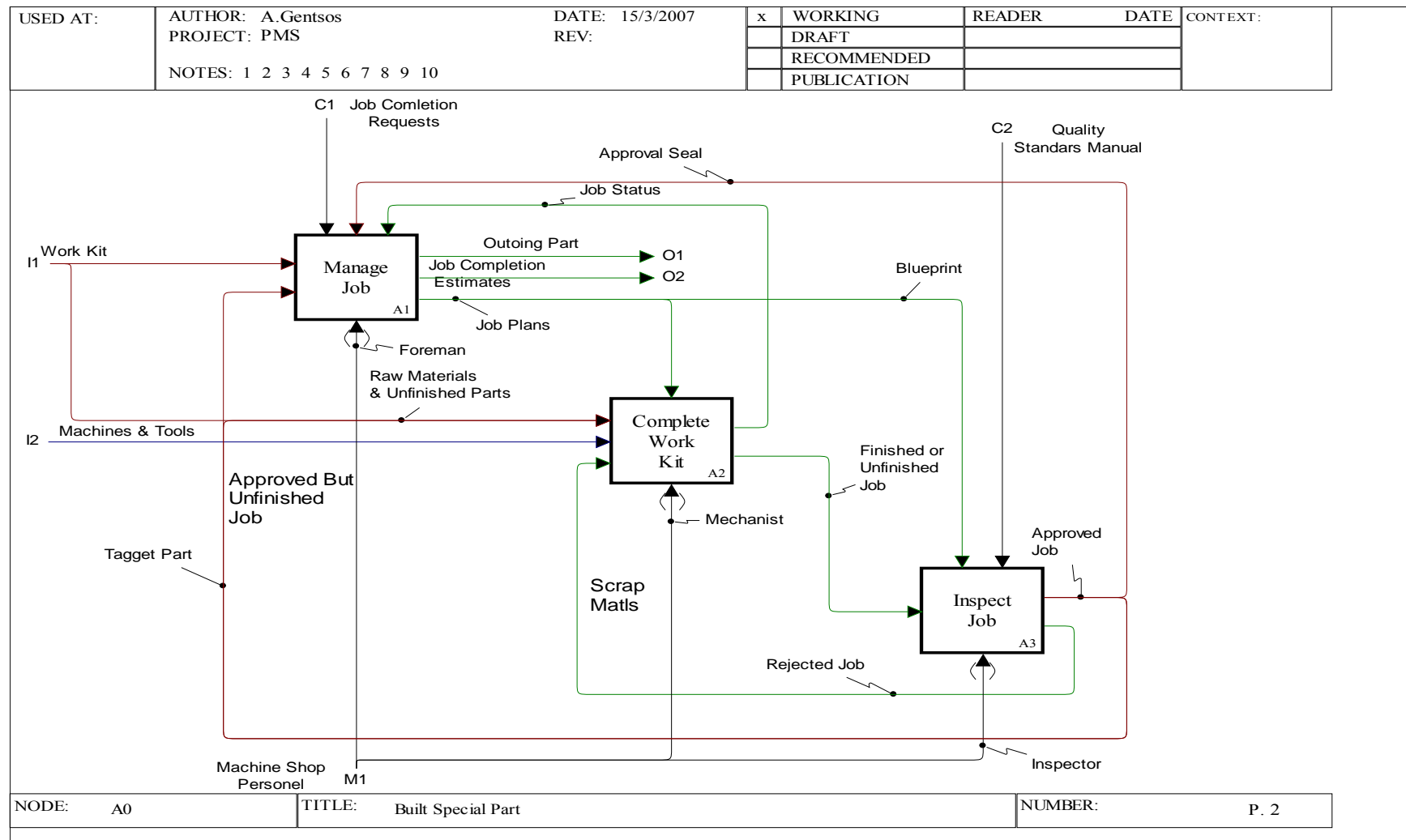
Purpose: Understand the tasks involved in bilding a special part and how those tasks interrelate, so a training manual for Machine Shop personnel can be written.

Viewpoint: Prototype machine shop manager

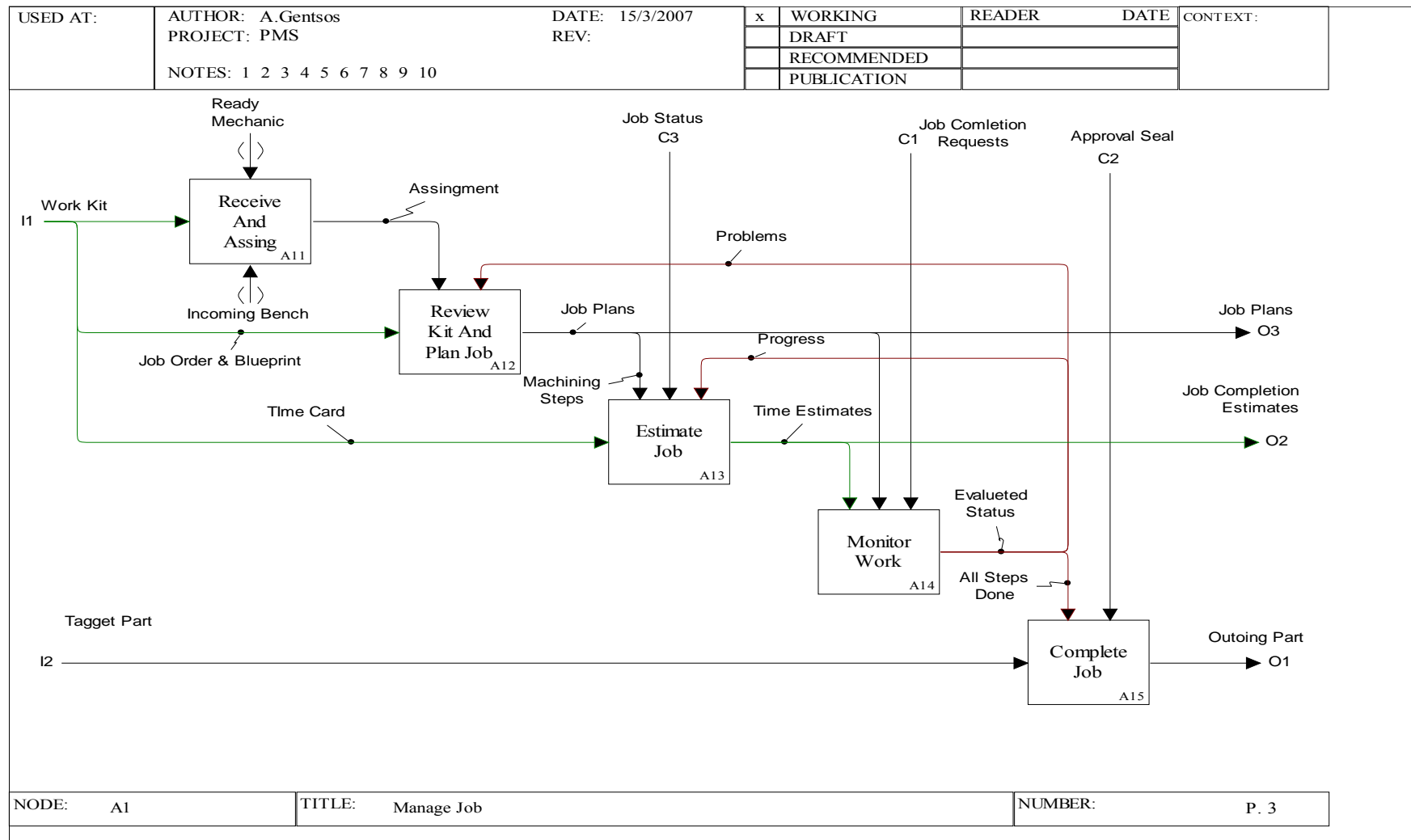
NODE: A-0	TITLE: CONTEXT	NUMBER: P. 1
-----------	----------------	--------------

Σχήμα 5.1: Build Special Part, A-0 διάγραμμα [1].

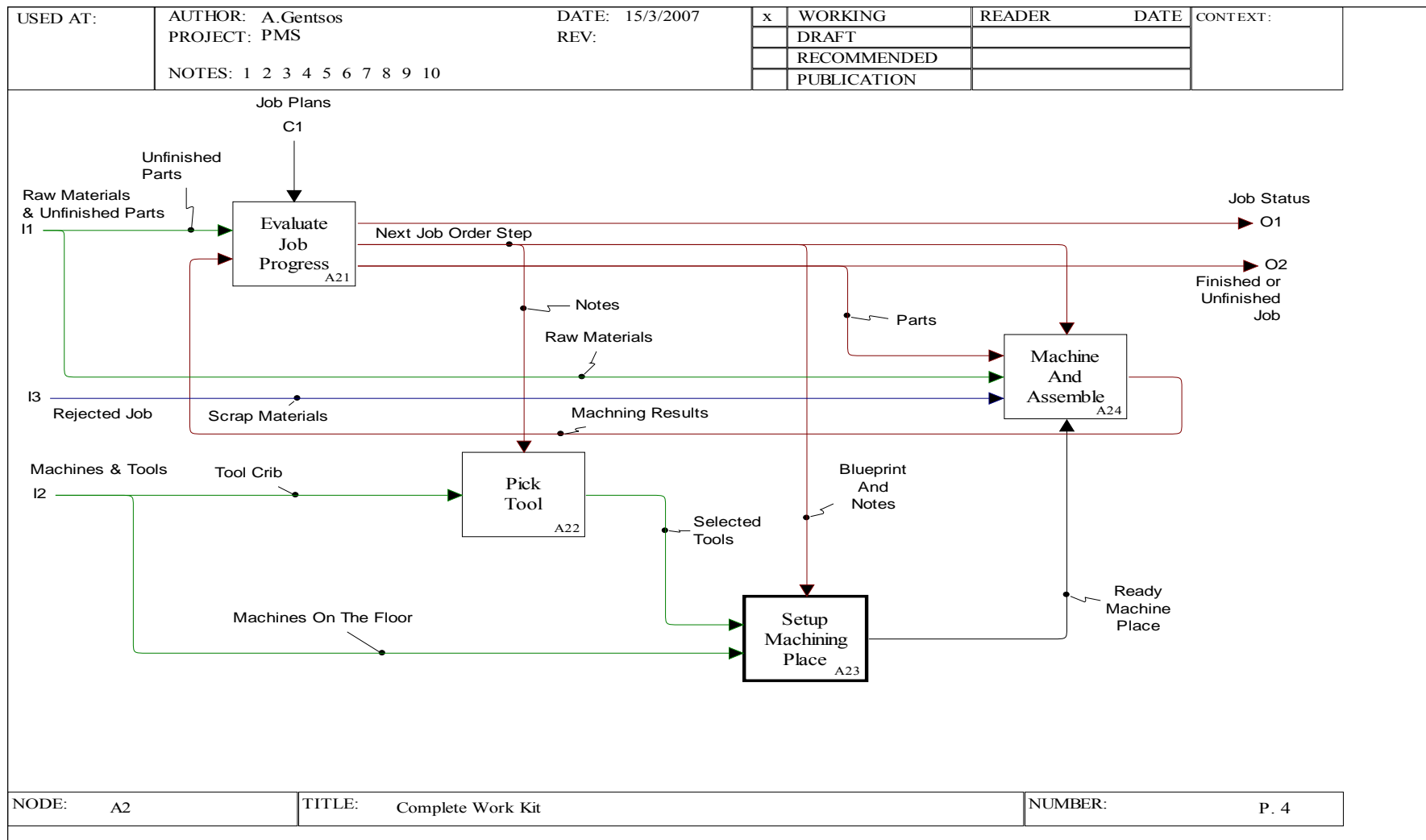




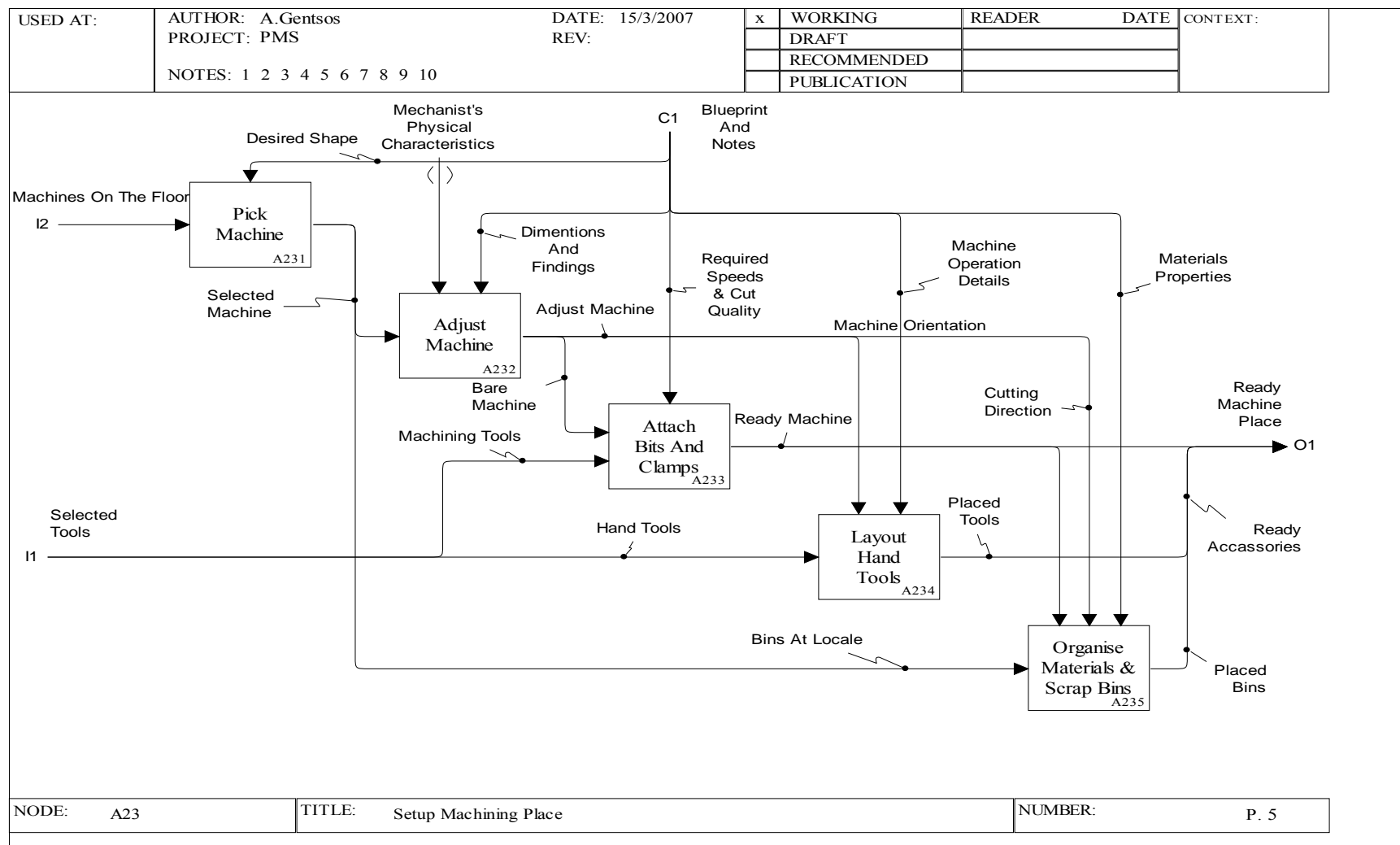
Σχήμα 5.2: Build Special Part, A0 διάγραμμα [2].



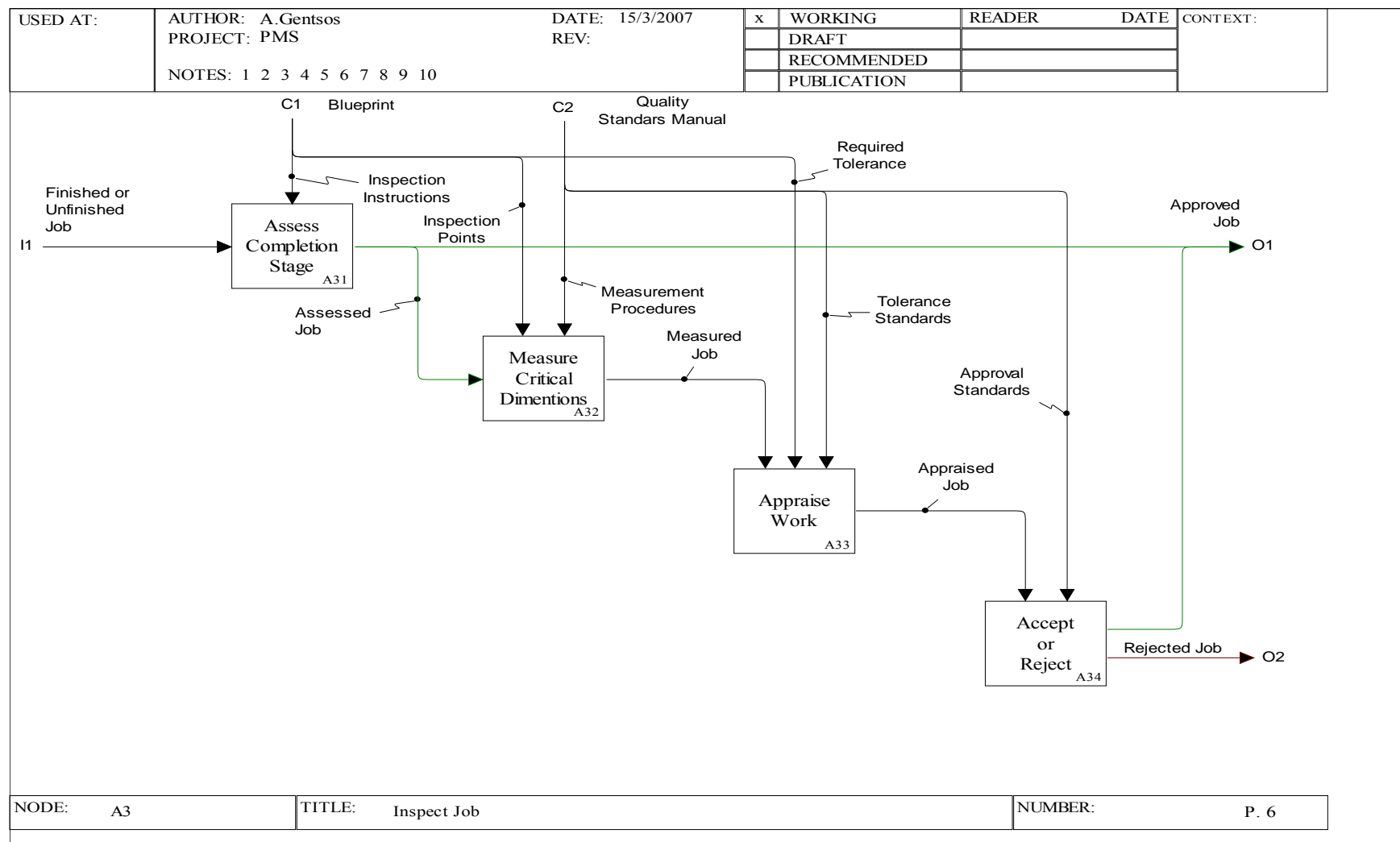
Σχήμα 5.3: Manage Job, A1 διάγραμμα [3].



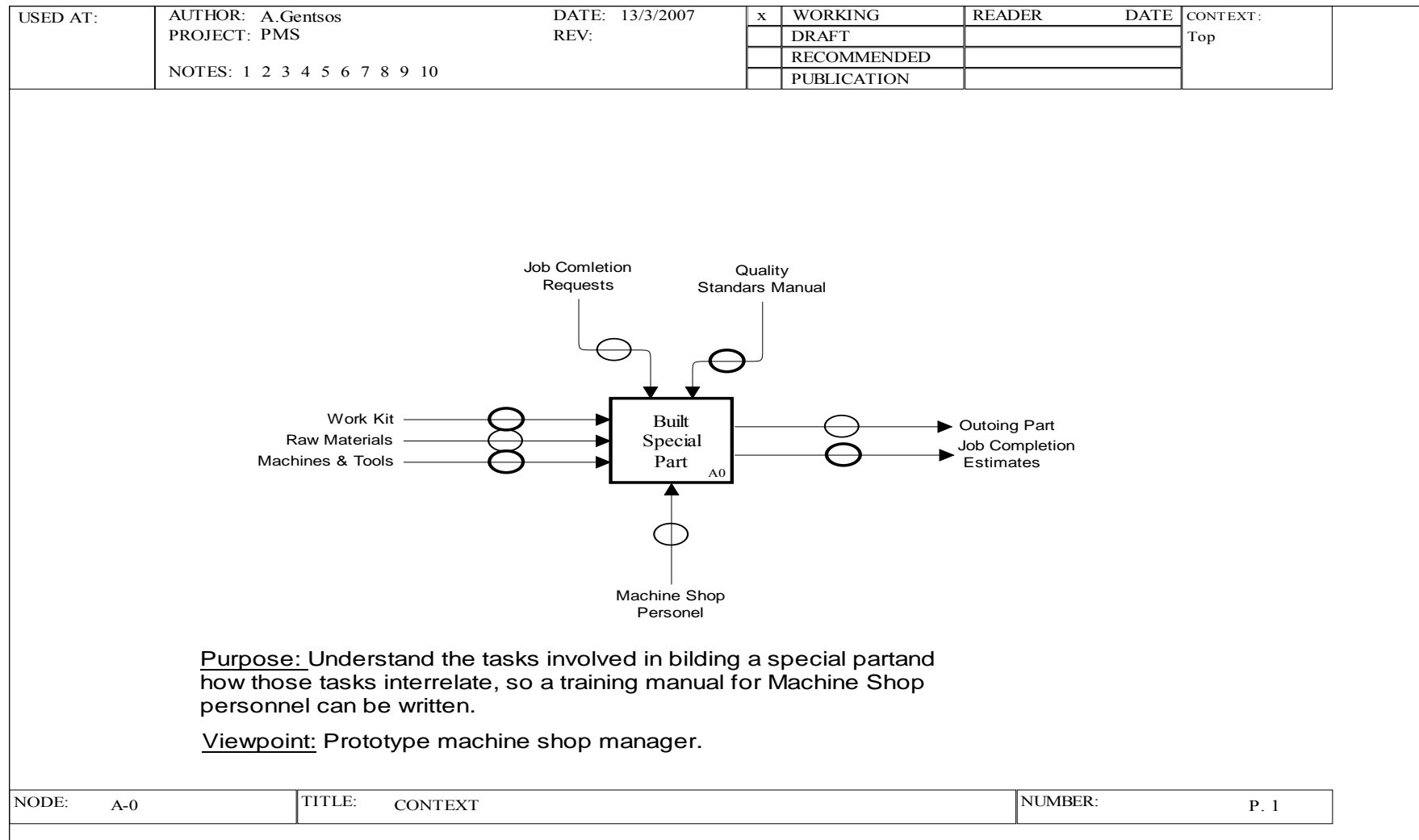
Σχήμα 5.4: Complete Work Kit, διάγραμμα A2 [4].



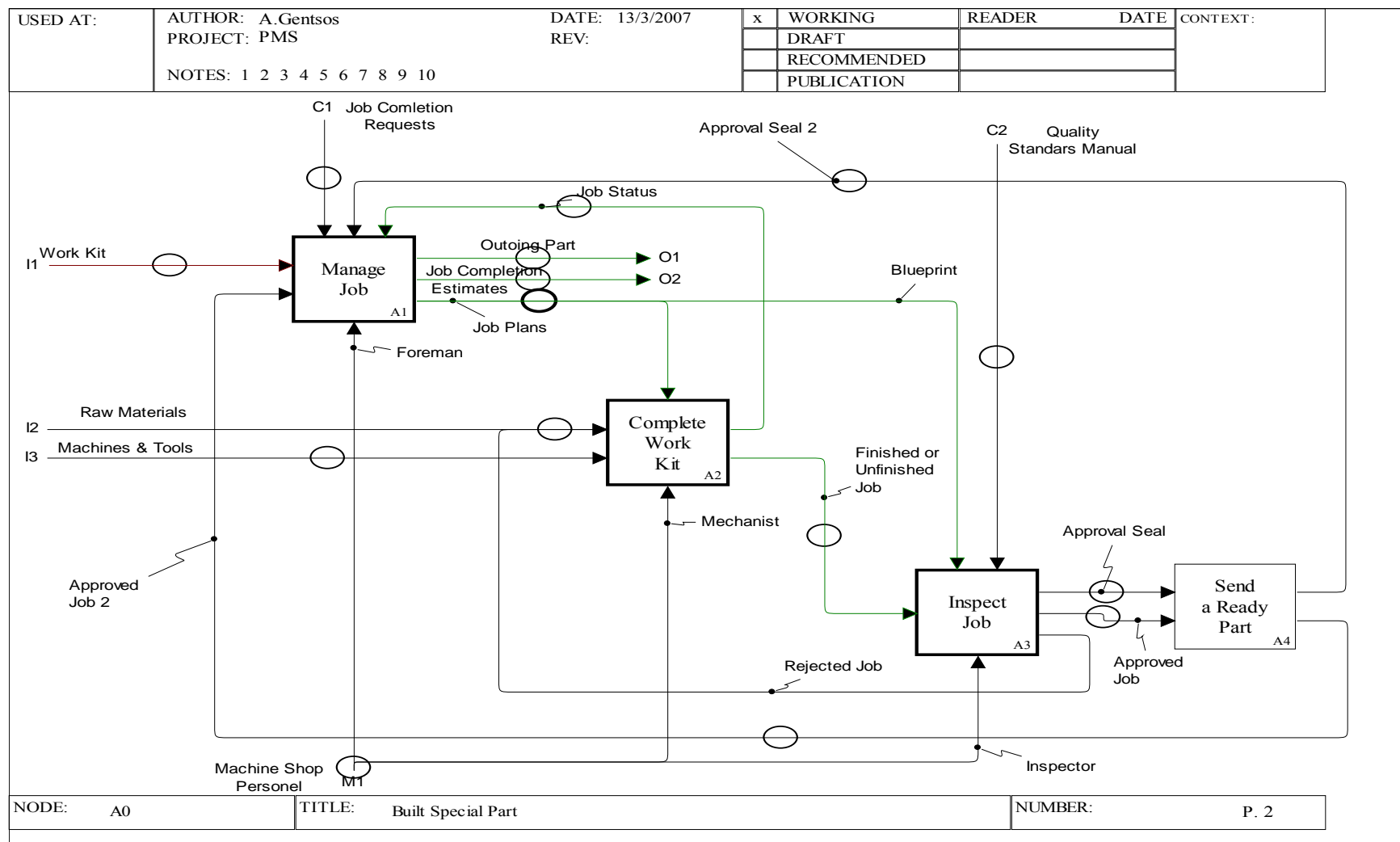
Σχήμα 5.5: Setup Machining Place, Διάγραμμα A23[5].



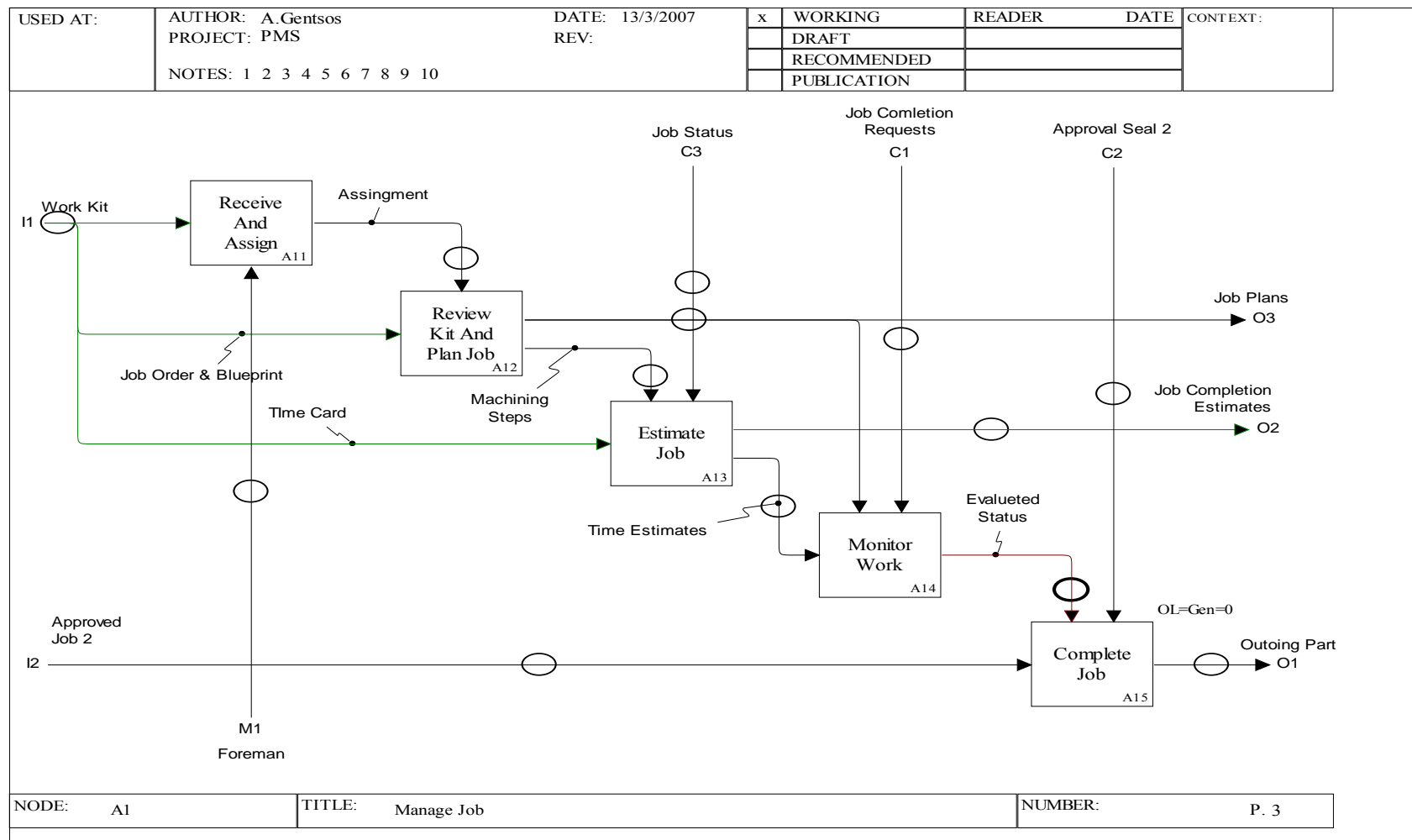
Σχήμα 5.6: *Inspect Job*, διάγραμμα A3 [6].



Σχήμα 5.7: Build Special Part, A-0 διάγραμμα [7].

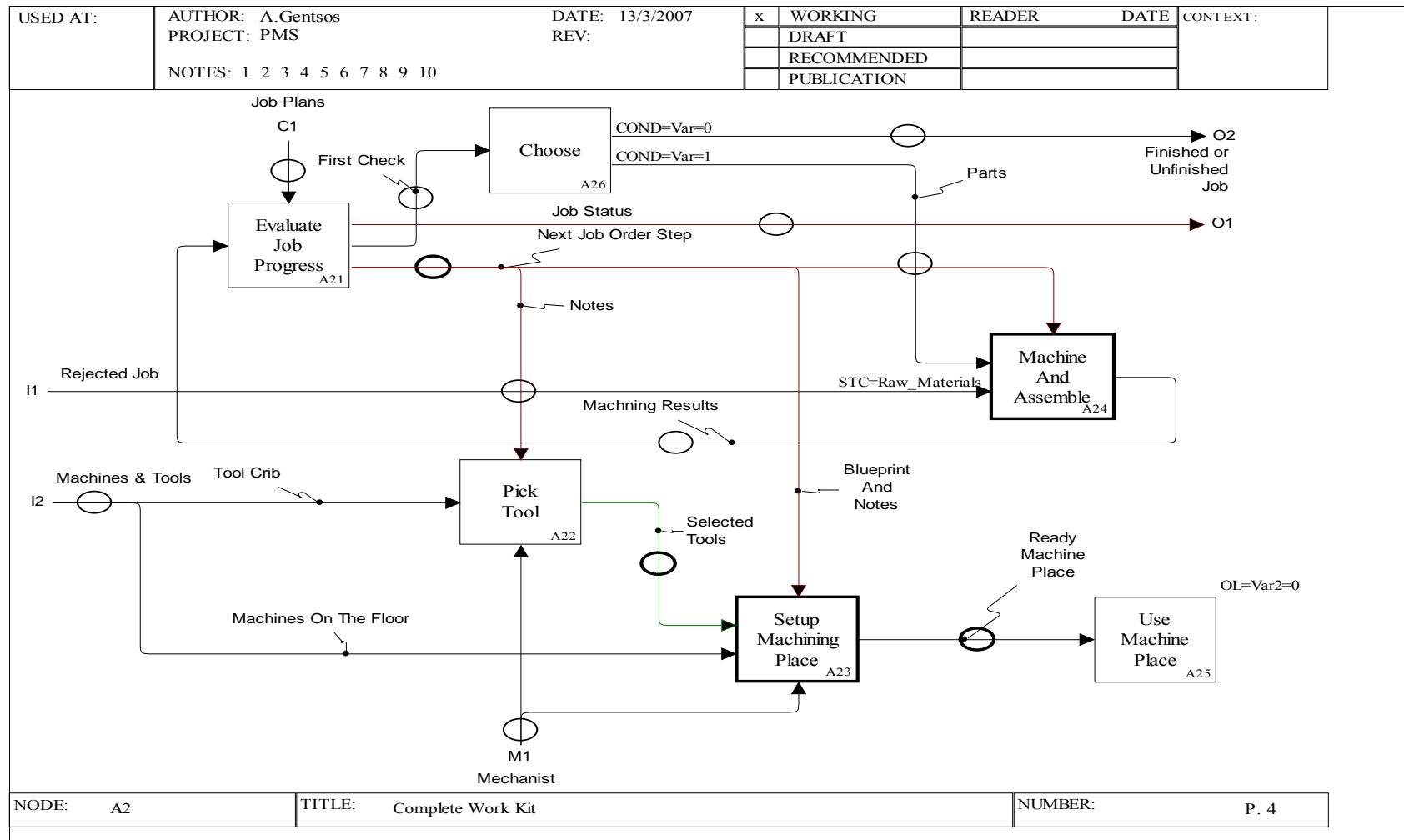


Σχήμα 5.8: Build Special Part, A0 διάγραμμα [8].

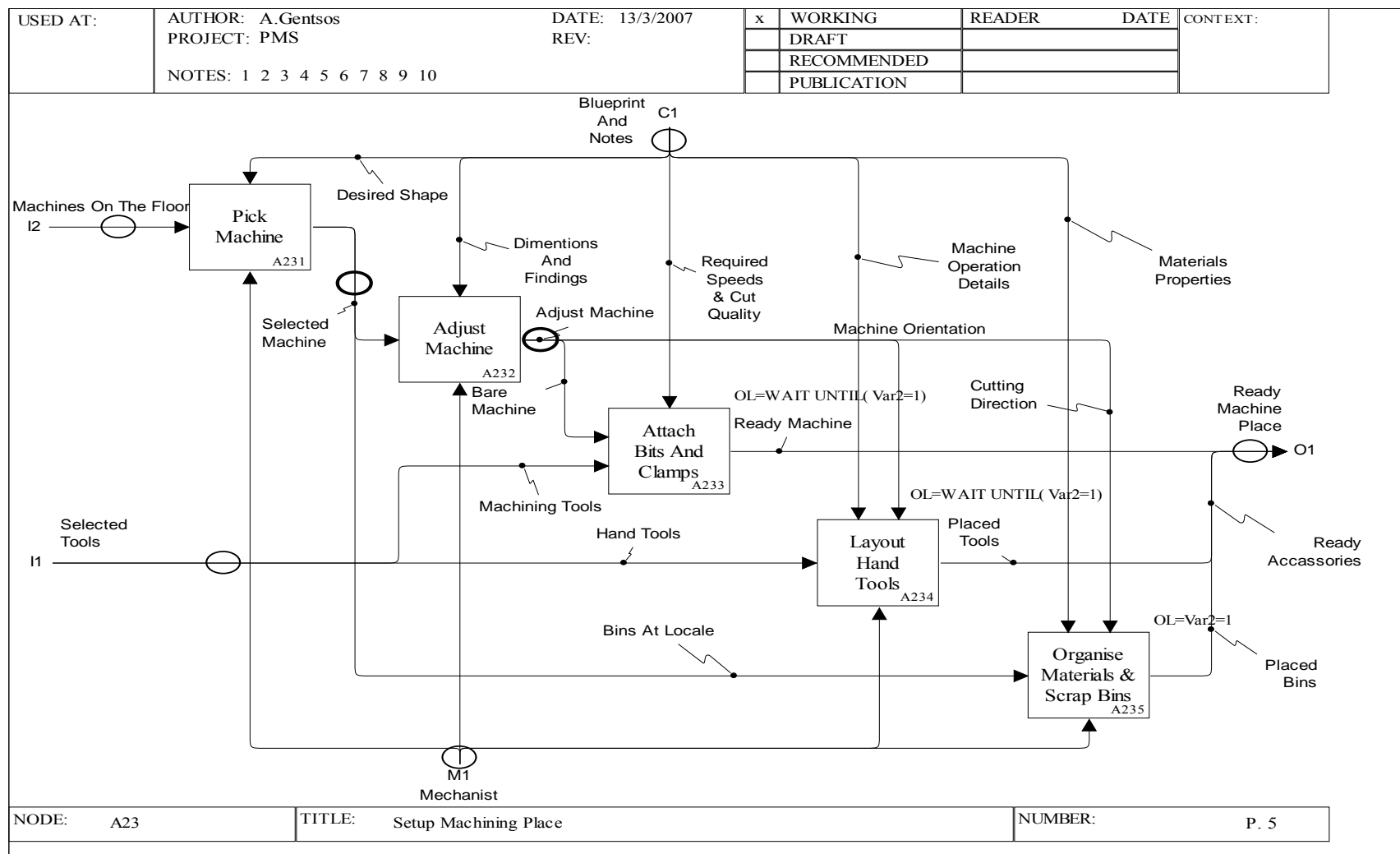


Σχήμα 5.9: *Manage Job*, A1 διάγραμμα [9].

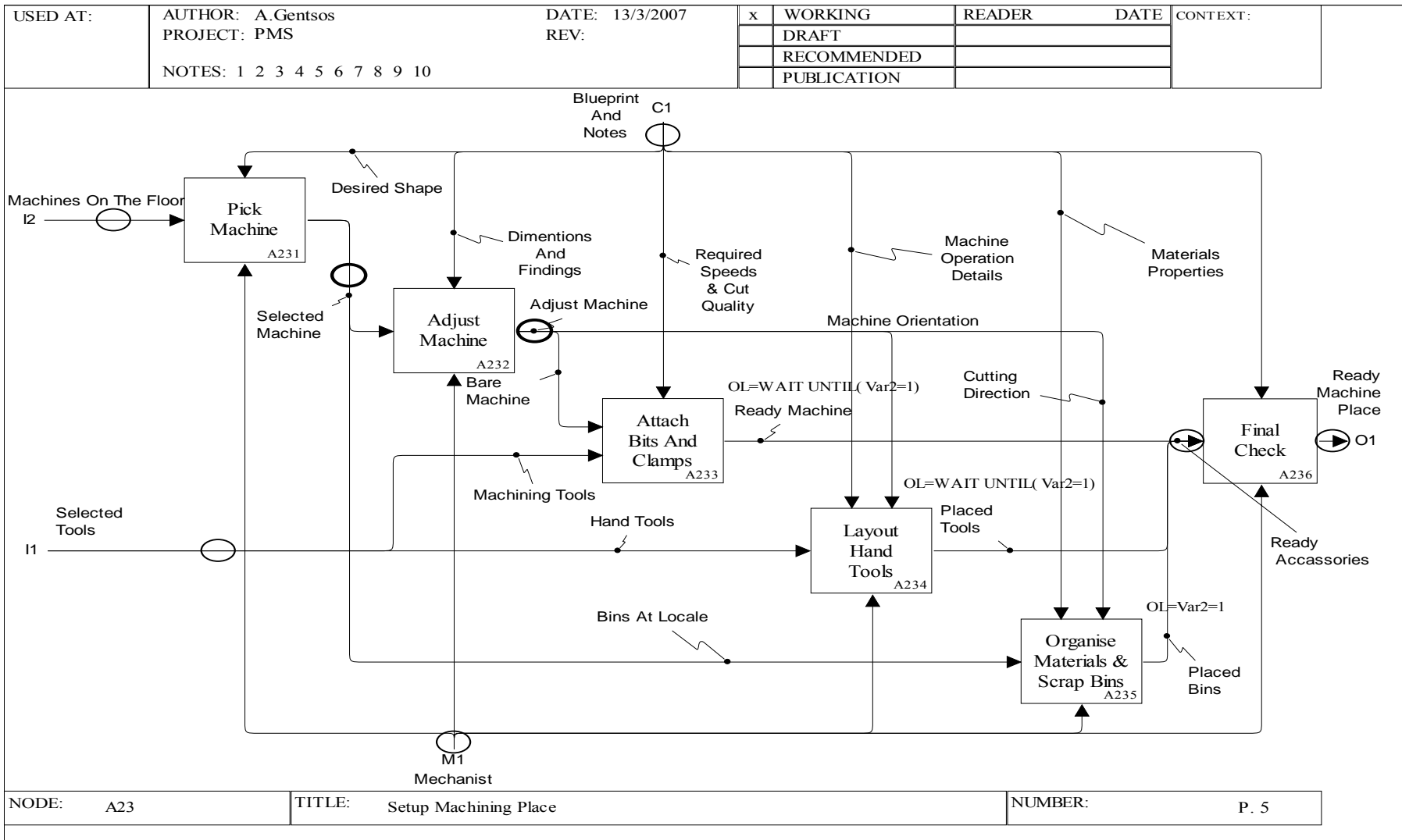


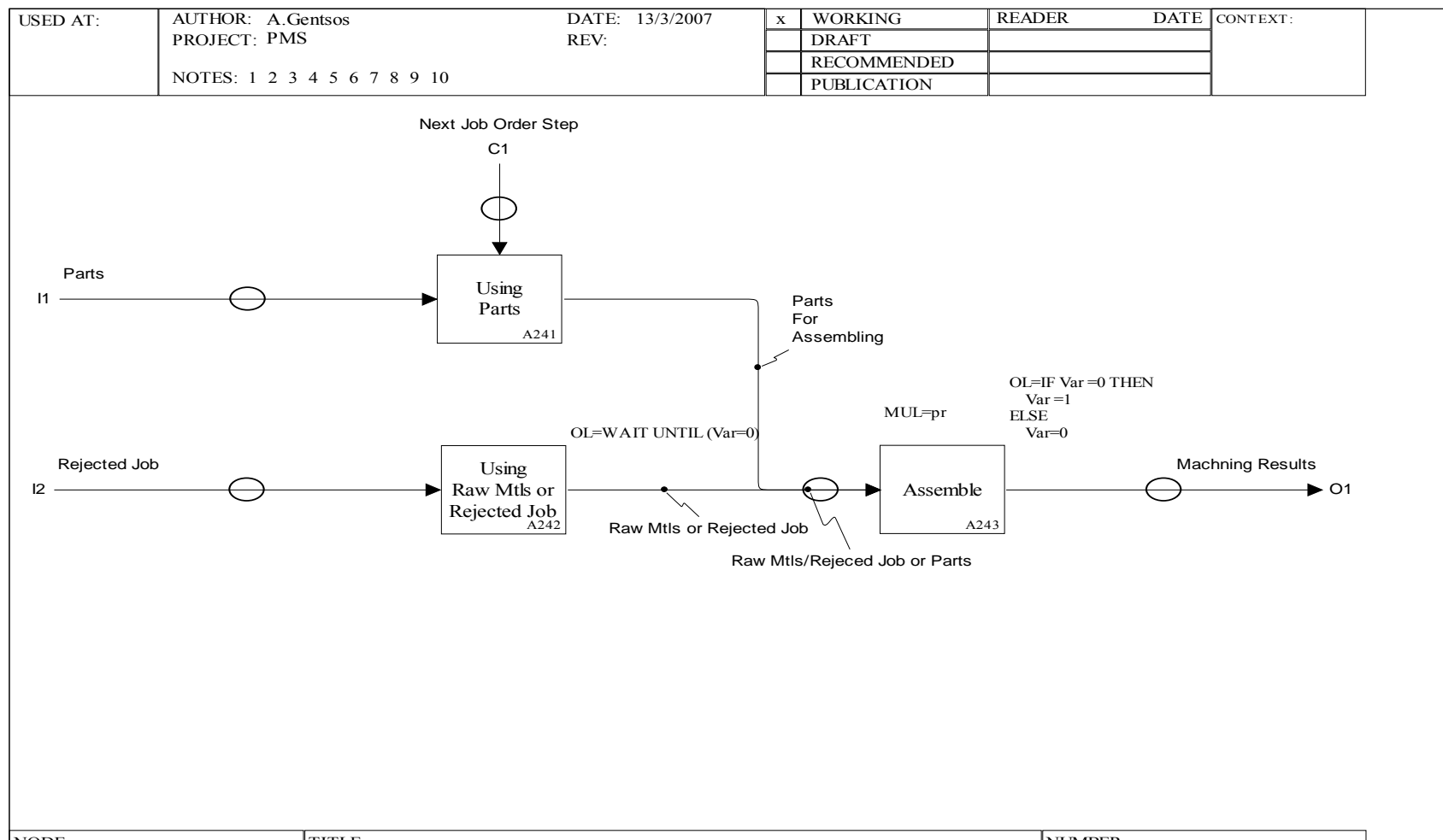


Σχήμα 5.10: Complete Work Kit, διάγραμμα A2 [10].

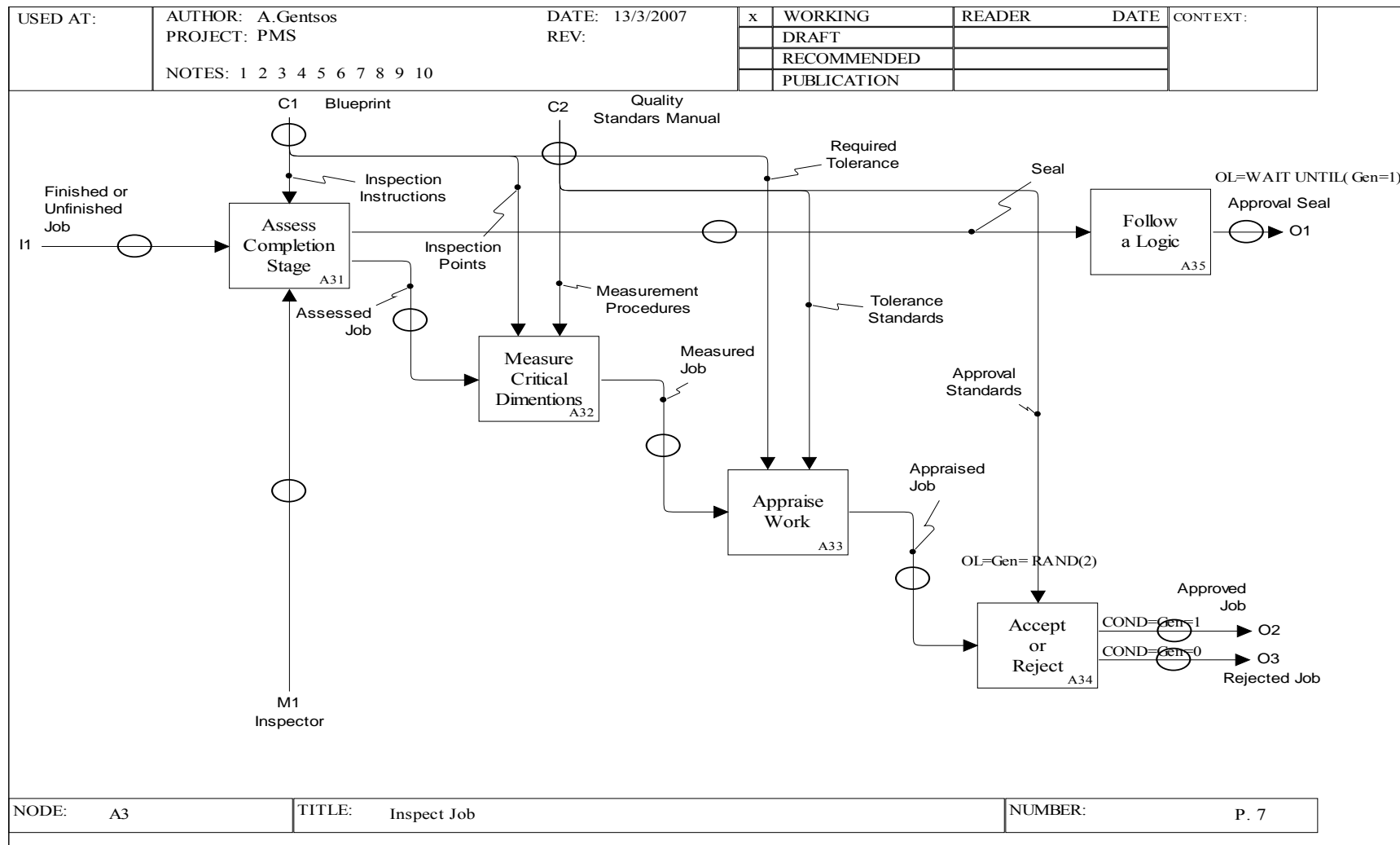


Σχήμα 5.11: Setup Machining Place, διάγραμμα A23 [11].

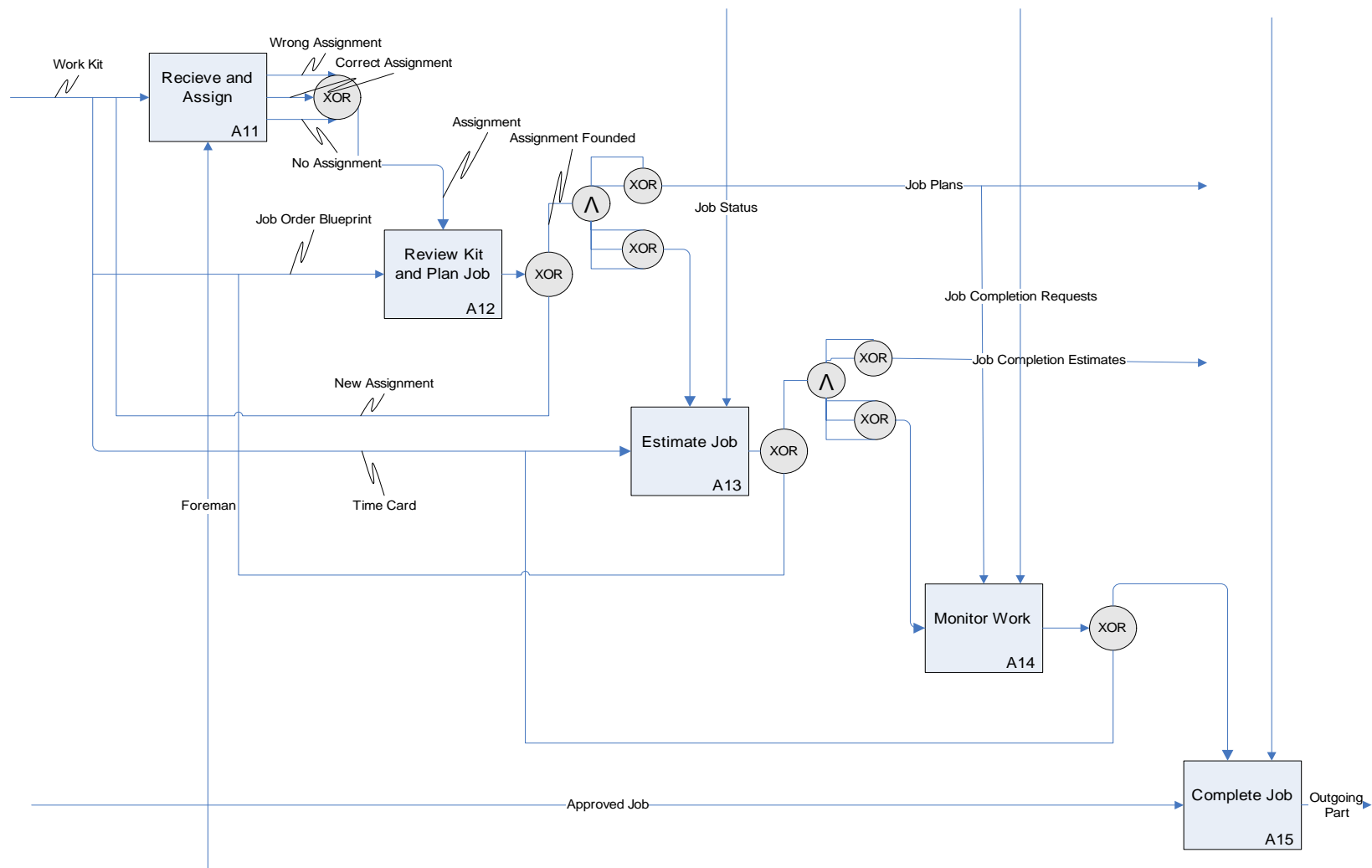




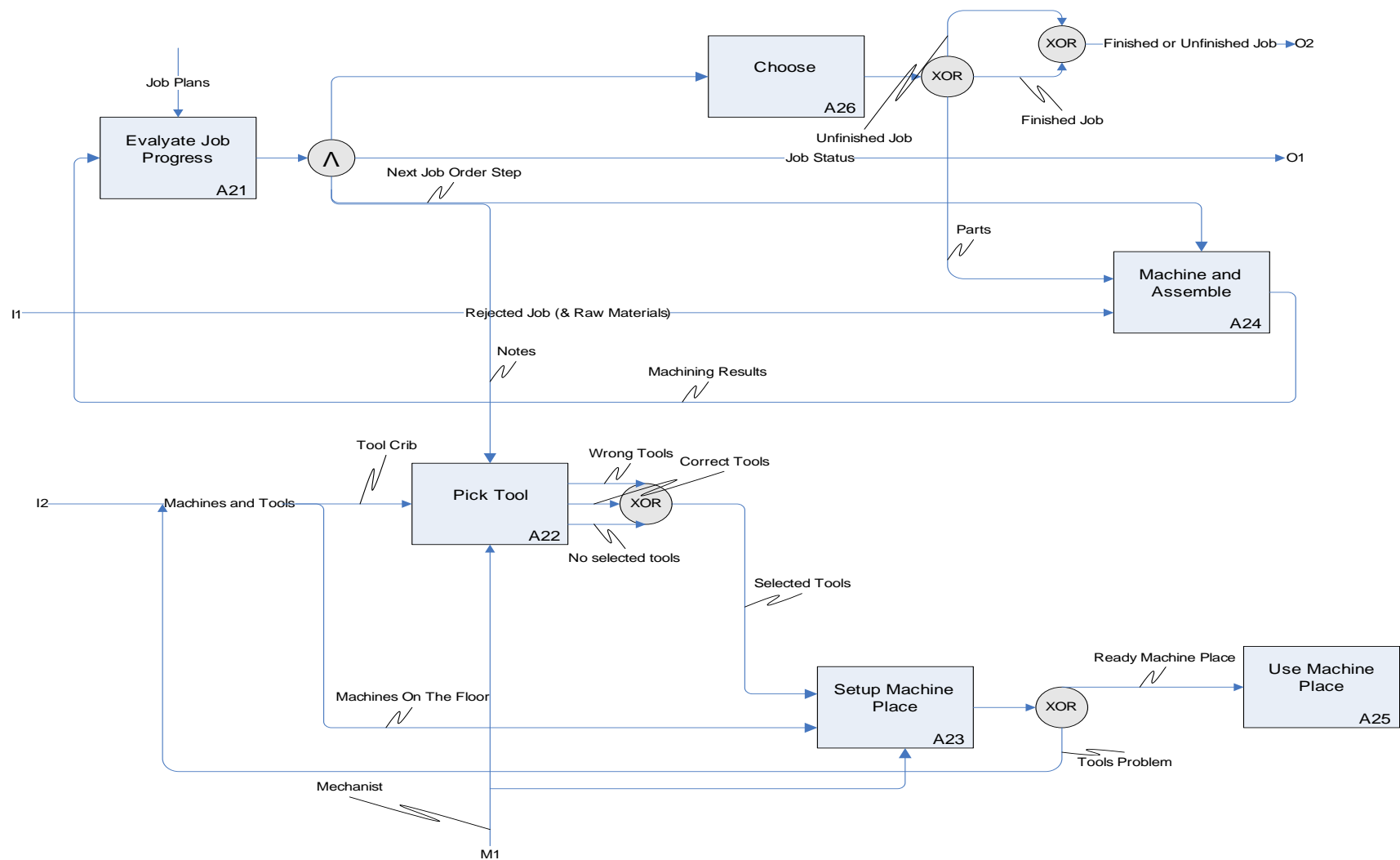
Σχήμα 5.12: Machine and Assemble, διάγραμμα A24 [12].



Σχήμα 5.13: *Inspect Job*, διάγραμμα A3 [13].



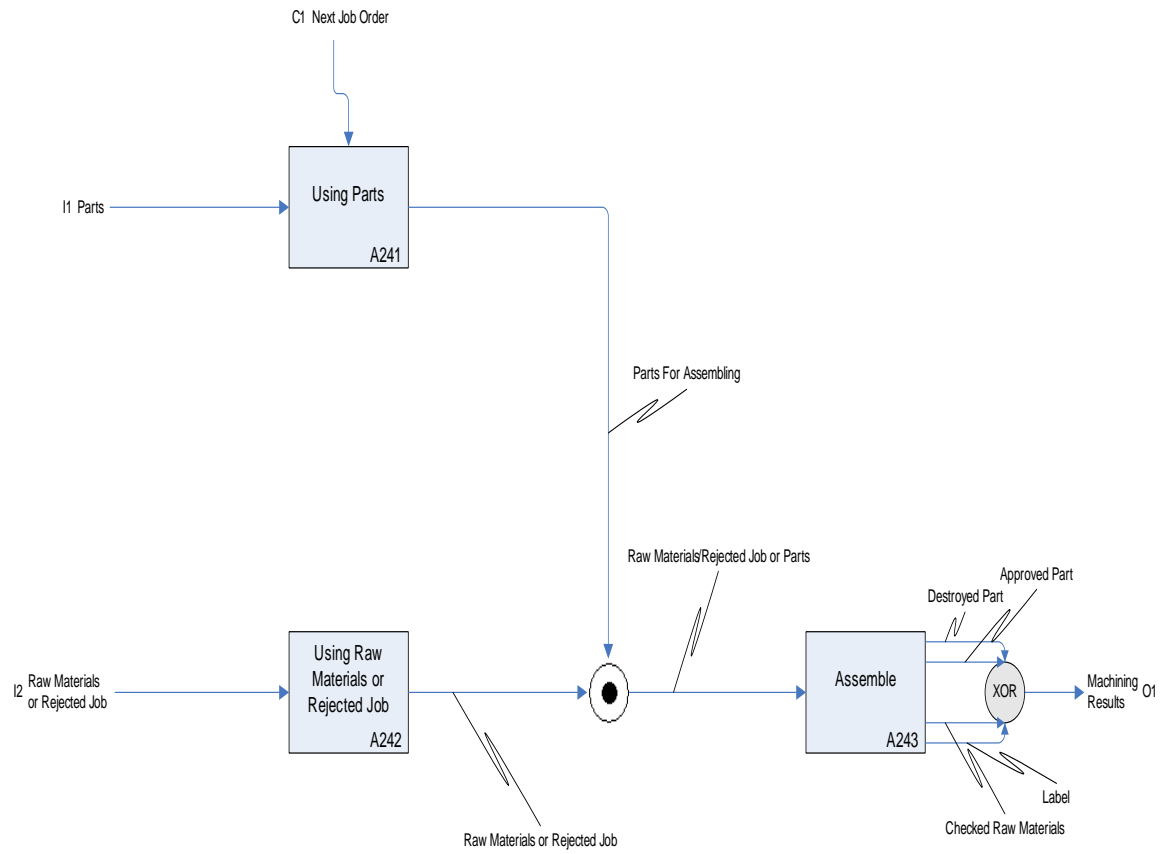
Σχήμα 5.21: Διάγραμμα A1 μετά την προσθήκη σφαλμάτων [14].



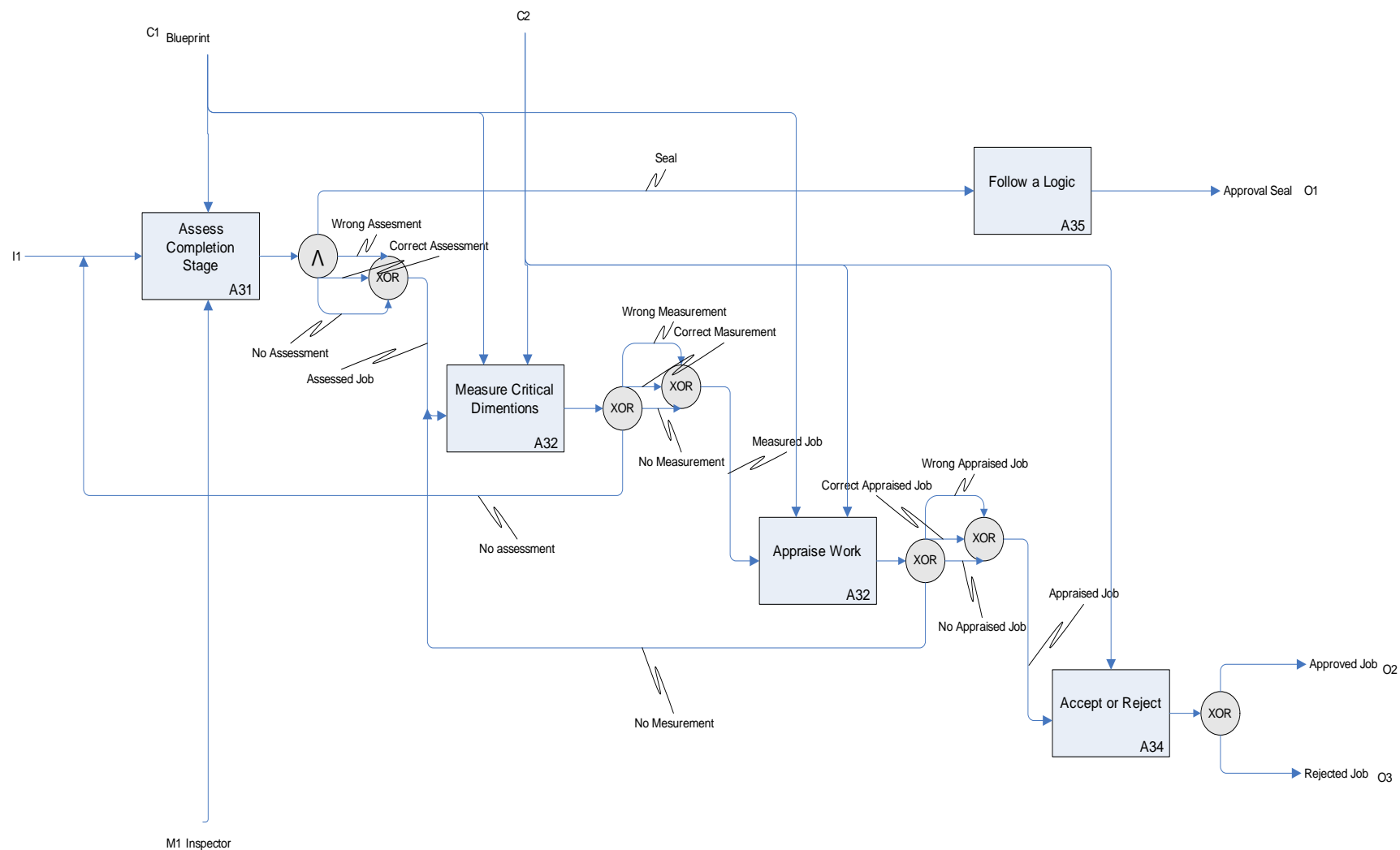
Σχήμα 5.22: Διάγραμμα A2 μετά την προσθήκη σφαλμάτων [15].



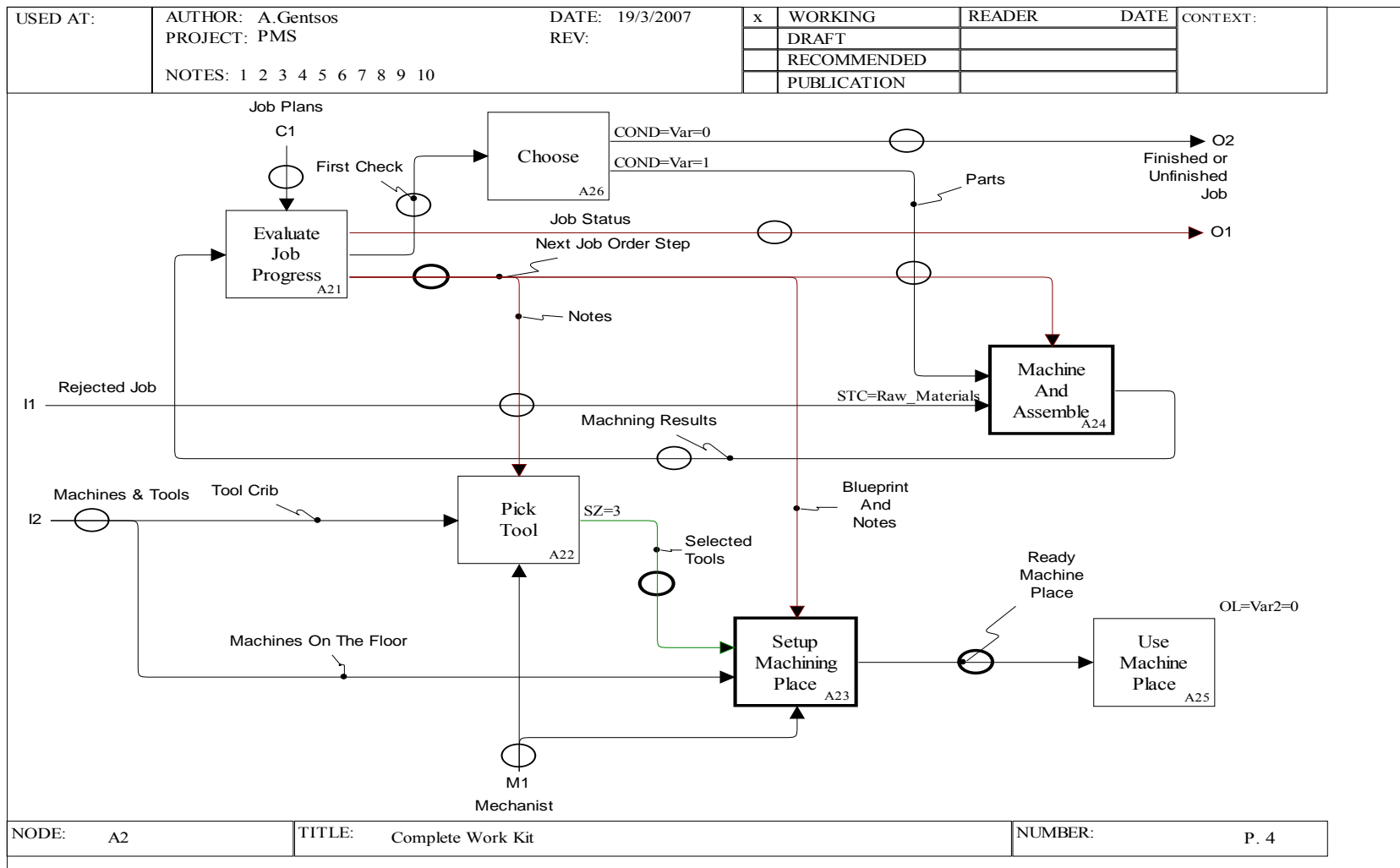




Σχήμα 5.24: Διάγραμμα A24 μετά την προσθήκη σφαλμάτων [17].



Σχήμα 5.25: Διάγραμμα A3 μετά την προσθήκη σφαλμάτων [18].



Σχήμα 5.31: Εμφάνιση ένδειξης size πάνω στο διάγραμμα [19].

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] Ruth Sara Aguilar-Saven (2004). Business Process Modeling: Review and framework. *Production Economics* (pp. 129-149).
- [2] Ralf R. Bravoco and Surya B. Yadav (1985). Required definition architecture – An overview. *Computers in industry*, Vol 6 (pp. 237-251).
- [3] Carla Carnaghan (2006). Business Process Modeling approaches in the context of process level audit risk assessment: An analysis and comparison. *Accounting Information System* (170- 204).
- [4] Jan Fulscher and Stephen Powell (1999). Anatomy of a process mapping workshop. *Business Process Management Journal*, Vol 5 No 3, (pp. 208-237).
- [5] Bosil, J., Giaglis, G., & Hlupic, V. (2000). IDEF diagrams and petri nets for business process modelling. In: *Proceedings of the 2000 winter simulation conference* (pp. 1676–1683).
- [6] Godwin, A., Gleeson, J. and Gwillian, D (1989), “An assessment of the IDEF notations as descriptive tools. *Information Systems*, Vol 14 No 1, pp. 13-28.
- [7] Hernandez-Matias, J.1; Vizan, A.; Hidalgo, A.; Rios, J. (2006) Evaluation of techniques for manufacturing process analysis, *Journal of Intelligent Manufacturing*, Volume 17, Number 5, pp. 571-583(13)
- [8] IDEF0 (1993) *Integration Definition for Function Modelling*, FIPS Publication 183, National Institute of Standards and Technology.

- [9] J. Mendling: Detection and Prediction of Errors in EPC Business Process Models. Doctoral Thesis, Vienna University of Economics and Business Administration. Vienna, Austria, May 2007.
- [10] Karpe, Y. S. (2006) Weave-Room Performance Decision-Making Process In Textiles: Mapping An Information Engineering Methodology, Doctor of Philosophy dissertation, North Carolina State University
- [11] Laguna, M., Marklund, J., "*Business Process Modeling, Simulation, and Design*", Prentice Hall, United States of America (2005)
- [12] Ma, Z., Zhang, W. and Ma, W. (2002). Extending IDEF1X to model fuzzy data. *Journal of Intelligent Manufacturing*, 13, 295–307.
- [13] Mahmudi, J. and V. Tavakkoli. Simulation: The best solution for BPR.
- [14] Marca, D. A. and McGowan, C. L. 1987 *SADT: Structured Analysis and Design Technique*. McGraw-Hill, Inc.
- [15] Mayer, R.J. (1992), "IDEF0 function modelling", A reconstruction of the original Air Force Wright Aeronautical Laboratory Technical Report.
- [16] Metasoftware (2007), Workflow Modeler Users Guide, ([www.metasoftware.com](http://www.metasoftware.com)).
- [17] Metasoftware (2007), Activity expretion, ([www.metasoftware.com](http://www.metasoftware.com)).
- [18] Metasoftware (2007), Workflow Simulator Users Guide, ([www.metasoftware.com](http://www.metasoftware.com)).
- [19] Presley, A. and D. Liles (1995). "The Use of IDEF0 for the Design and Specification of Methodologies". 4th Industrial Engineering Research Conference, Nashville.

- [20] Hui Shen, Brian Wall, Michal Zaremba, Yuliu Chen and Jim Browne (2004).  
Integration of business modeling methods for enterprise information system  
analysis and user requirements gathering. Computers in Industry (pp. 307-323).
- [21] Roger, K.J., Whitman, L., Underdown, R. (1998). The Enterprise Integration  
issues Encountered with Agile Process Introduction. Retrieved Sept 22 2004,  
from the World Wide Web: <http://arri.uta.edu/eif/rogersfaim98.pdf>
- [22] Vernadat, F.B, "Enterprise modeling and integration: principles and  
applications", Chapman & Hall, London (1995)
- [23] Whitman, L. and Presley, A. (1997). Structured models and dynamic systems  
analysis: The integration of the IDEF0/IDEF3 modeling methods and discrete  
event simulation