

XANIA 2015



**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑ-
ΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ
ΠΑΡΑΓΩΓΗΣ**

**ΕΝΟΠΟΙΗΣΗ ΠΛΗΡΟ-
ΦΟΡΙΑΚΩΝ ΜΟΝΤΕ-
ΛΩΝ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ
ΑΝΑΠΤΥΞΗΣ ΤΟΥ
ΠΡΟΪΟΝΤΟΣ.**

Διατριβή που υπεβλήθη για τη μερική ικανοποίηση των απαιτήσεων για την
απόκτηση Διδακτορικής Διατριβής
υπό
Δερμιτζάκη Ελευθερίου

©Copyright 2015 υπό Ελευθέριο Ε. Δερμιτζάκη

Η διατριβή του Ελευθέριου Ε. Δερμιτζάκη εγκρίνεται

Όνομα

Υπογραφή

Καθ. Μπιλάλης Νικόλαος

Καθ. Μουστάκης Βασίλειος

Καθ. Σαπίδης Νικόλαος

Ευχαριστίες

Στην προσπάθειά μου για την απόκτηση του διδακτορικού διπλώματος στον τομέα συστημάτων παραγωγής πολλοί άνθρωποι συνέβαλαν, ο καθένας με το δικό του τρόπο.

Πρώτα από όλα, τις ευχαριστίες μου στον Καθηγητή του Πολυτεχνείου Κρήτης Δρ. Μπιλάλη Νικόλαο για την εμπιστοσύνη που έδειξε στο πρόσωπό μου, για την ανάθεση, υποστήριξη και καθοδήγηση σε όλο το διάστημα της εκπόνησης της παρούσας εργασίας. Τους καθηγητές Δρ. Μουστάκη Βασίλειο και Δρ. Σαπίδη Νικόλαο για την εποικοδομητική κριτική και το χρήσιμο σχολιασμό τους καθ' όλη τη διάρκεια της διατριβής μου.

Τον Καθηγητή του Πολυτεχνείου Κρήτης Δρ. Αντωνιάδη Αριστομένη για την αμέριστη συμπαράσταση και υποστήριξή του. Τον Επίκουρο Καθηγητή του Πολυτεχνείου Κρήτης Δρ. Μαρινάκη Ιωάννη για τη βοήθεια που μου προσέφερε.

Τέλος, ένα μεγάλο ευχαριστώ προς τη σύζυγο μου Αλεξάνδρα, τις κόρες μου Γωγώ και Ελένη που όλον αυτόν τον καιρό υπέμειναν μαζί μου τις καθημερινές μικρές και μεγάλες δυσκολίες στην προσπάθειά μου, με αγάπη και υπομονή και τέλος ένα μεγάλο ευχαριστώ στους γονείς μας Γιάννη, Ελένη και Γεωργία για τη στήριξη, την αγάπη και την εμπιστοσύνη που μου έδειξαν.

*Αφιερωμένο στον πατέρα μου ...
Εμμανουήλ*

Σύντομο Βιογραφικό Σημείωμα

Ο Δερμιτζάκης Ε. Ελευθέριος γεννήθηκε στα Χανιά το 1968. Το 1993 τελείωσε το τμήμα Επιστήμης Υπολογιστών από το Πανεπιστήμιο Κρήτης της Σχολής Θετικών Επιστημών. Το 2005 περάτωσε το μεταπτυχιακό πρόγραμμα σπουδών του Πολυτεχνείου Κρήτης στον τομέα της Επιχειρησιακής Έρευνας. Από το 1995 μέχρι το 2001 εργάστηκε στο ΤΕΙ Κρήτης ως υπεύθυνος διαχείρισης και ανάπτυξης υπολογιστικών δικτύων και υπηρεσιών ενώ παράλληλα δίδασκε τα μαθήματα Προγραμματισμός Ι και ΙΙ καθώς και Δίκτυα Υπολογιστών.

Από το 1996 μέχρι σήμερα έχει εργαστεί με συμβάσεις έργου σε πολλά ερευνητικά προγράμματα ΕΠΕΑΚ, ΠΕΝΕΔ σε συνεργασία με το Πολυτεχνείο Κρήτης, Ινστιτούτο Τεχνολογίας & Έρευνας (ΙΤΕ), ΤΕΙ Κρήτης καθώς και με άλλες υπηρεσίες (Νομαρχία Χανίων, ΕΡΤ Α.Ε., ΔΕΗ, 28^η ΕΒΑ Χανίων, κ.λπ.). Σήμερα εργάζεται ως μόνιμος Καθηγητής πληροφορικής στη δευτεροβάθμια εκπαίδευση.

Πίνακας περιεχομένων

Ευχαριστίες	4
Σύντομο Βιογραφικό Σημείωμα	6
Κατάλογος Εικόνων	8
Κατάλογος Πινάκων	12
Περίληψη	13
Abstract	15
1. Εισαγωγή.....	16
2. Στάθμη των γνώσεων	19
2.1 Γενικά για τα μοντέλα ανάπτυξης προϊόντος	19
2.2 Σειριακά και σχεδιαστικά μοντέλα	20
2.2.1 Πίνακας Δομής Σχεδίασης	20
2.2.2 Ταξινόμηση των ΠΔΣ.....	21
2.2.3 Μέθοδοι ανάλυσης ΠΔΣ	22
2.2.4 Αριθμητικοί ΠΔΣ	23
2.2.5 Συμπεράσματα για τον ΠΔΣ	24
2.3 Μοντέλα βασιζόμενα στον ΠΔΣ	25
2.3.1 Μοντέλα Ανατροφοδότησεων και διασταυρώσεων	25
2.3.2 Μοντέλο ακολουθιακών επαναλήψεων	25
2.3.3 Μοντέλο εκτίμησης επαναληπτικών χρόνων	26
2.3.4 Μοντέλα αποσύνθεσης μεγάλων συστημάτων	27
2.3.5 Μέθοδος πίνακα μετασχηματισμού εργασίας.....	27
2.3.6 Μοντέλο αποσύνθεσης δραστηριοτήτων σε δομικές μονάδες(modules)	28
2.3.7 Άλλα συστήματα αποσύνθεσης.....	28
2.4 Στοχαστικά μοντέλα χρονική ανοχής	29
2.4.1 Μοντέλο Q-GERT	29
2.4.2 Μοντέλο δικτύου αναμονής.....	29
2.5 Χρονικά μοντέλα ανασκόπησης.....	30
2.5.1 Συγχρονισμός των αναθεωρήσεων σχεδίου.....	30
2.5.2 Σχεδιασμός ενοτήτων	30
2.5.3 Μοντέλο έκδοσης προϊόντος	31
2.5.4 Σταδιακή στοχοθέτηση.....	31
2.6 Μοντέλα παραλληλισμού	32
2.6.1 Παράλληλος σχεδιασμός	32
2.6.2 Παράλληλο / Σειριακό Μοντέλο ΠΜΕ	32
2.6.3 Μοντέλο σταδιακής επικάλυψης.....	33
2.7 Μοντέλα σχεδίασης που δεν καλύπτουν οι παραπάνω κατηγορίες.	33
2.8. Μοντέλα Συναρμολόγησης.....	34
2.8.1. Εισαγωγή.....	34
2.8.2. Επισκόπηση μοντέλων συναρμολόγησης	35

2.9. Αλγόριθμοι Ομαδοποίησης.....	36
2.9.1. Εισαγωγή.....	36
2.9.2. Κατηγοριοποίηση Αλγορίθμων.....	37
2.10. Το μοντέλο Core Product Model (CPM).....	39
2.10.1 Εισαγωγή.....	39
2.10.2 Επισκόπηση του μοντέλου.....	40
2.10.3 Στοιχεία – συστατικά του μοντέλου.....	41
2.10.4 Μοντέλα CPM.....	45
2.10.5 Προεκτάσεις του CPM.....	47
2.11 Το μοντέλο Open Assembly Model /NIST-CPM (OAM/NIST-CPM).....	48
2.11.1 Επισκόπηση του μοντέλου.....	48
2.11.2. Στοιχεία – συστατικά του μοντέλου.....	48
3. OAM2DSM: Ενοποίηση των μοντέλων OAM και DSM.....	51
3.1 Εισαγωγή.....	51
3.2. Μοντελοποίηση του OAM2DSM.....	53
3.2.1. Εισαγωγή.....	53
3.3.2 Το μοντέλο ενοποίησης OAM2DSM.....	55
3.3.3. Σχεσιακή αναπαράσταση του OAM/NIST-CPM.....	55
3.3.4. Λεξικογραφική ανάλυση των αρχείων STEP– Εξαγωγή δεδομένων.....	61
3.3.5. Μοντελοποίηση του Πίνακα Δομής Σχεδίασης για το OAM2DSM.....	67
3.3.6. Υπολογισμός κόστους συναρμολόγησης.....	69
3.3.7. Ομαδοποίηση στοιχείων του DSM.....	70
3.3.8. Μελέτη για την Αποδοτικότητα και Αξιολόγηση του αλγόριθμου Ομαδοποίησης.....	74
- Μελέτη Περίπτωσης.....	82
4. Σύνοψη – Συμπεράσματα.....	102
Επεκτάσεις του μοντέλου.....	103
5. Βιβλιογραφία.....	105
Παράρτημα Ι – Αλγόριθμος Ομαδοποίησης.....	112
Παράρτημα ΙΙ – Αρχεία καταγραφής για την ομαδοποίηση.....	117
Παράρτημα ΙΙΙ – Τεκμηρίωση του αλγόριθμου λεξικογραφικής ανάλυσης για το αρχείο STEP/AP203.....	125
Παράρτημα ΙV – Τεκμηρίωση του λογισμικού OAM2DSM.....	132
Γραφικό περιβάλλον διασύνδεσης(GUI).....	132

Κατάλογος Εικόνων

Εικόνα 1: Οι βασικές αρχές υλοποίησης και ανάγνωσης του ΠΔΣ.....	20
Εικόνα 2: Οι τρεις βασικές δομές αλληλεπίδρασης μεταξύ στοιχείων.....	21
Εικόνα 3: Αλυσίδα Markov για το μοντέλο ακολουθιακών επαναλήψεων.....	26
Εικόνα 4: Πίνακας αποσύνθεσης δραστηριοτήτων.....	28

Εικόνα 5: Το μοντέλο σταδιακής επικάλυψης	33
Εικόνα 6: Το μοντέλο AIDA (51).....	34
Εικόνα 7: Διαδικασία Ομαδοποίησης. Η ανάλυση σε ομάδες αποτελείται από τέσσερα βήματα και ένα μονοπάτι ανάδρασης. Τα βήματα αυτά είναι άμεσα συνδεδεμένα μεταξύ τους και επηρεάζουν τα εξαγόμενα αποτελέσματα.....	39
Εικόνα 8: Κλάσεις για την υλοποίηση μιας ουράς ακεραίων και μια στοίβα πραγματικών αριθμών	41
Εικόνα 9: Το μοντέλο CPM ως περιγραφή οντοτήτων [88].	43
Εικόνα 10 Πλαίσιο μοντελοποίησης της πληροφορίας (NIST) [84].	45
Εικόνα 11: Αποτύπωση των τριών επιπέδων του Core Product Model	47
Εικόνα 12: Το μοντέλο OAM ως διεπαφή στην εξέλιξη της σχεδίασης και συναρμολόγησης ενός προϊόντος.	48
Εικόνα 13: Το μοντέλο OAM σε συνοπτική μορφή [97].	50
Εικόνα 14: Το μοντέλο OAM [97].	51
Εικόνα 15: Σχηματική αναπαράσταση της ενοποίησης των μοντέλων OAM και DSM.	55
Εικόνα 16: Μέρος 1- Βάση δεδομένων για τους παραπάνω πίνακες.....	57
Εικόνα 17: Κλάσεις συσχετισμών του OAM (97)	58
Εικόνα 18: Μέρος 2-Η βάση δεδομένων για την Artifact Association κλάση.	59
Εικόνα 19: Σύνδεση των τμημάτων της βάσης δεδομένων.....	60
Εικόνα 20: Κλάσεις για την αποτύπωση των χαρακτηριστικών συναρμολόγησης και των ανοχών [63]......	60
Εικόνα 21: Η κλάση Assembly Feature [63].	61
Εικόνα 22: Μεγέθυνση πάνω στο μοντέλο OAM2DSM που αποτυπώνει την εξαγωγή στοιχείων από το STEP αρχείο στη βάση δεδομένων.	61
Εικόνα 23: Αλγόριθμος ανάγνωσης αρχείου STEP	63
Εικόνα 24: Λογικό διάγραμμα της προτεινόμενης μεθόδου εύρεσης χαρακτηριστικών συναρμολόγησης.	64
Εικόνα 25: Η δομή του STEP– AP203.	65
Εικόνα 26: Εξαγωγή δεδομένων από το αρχείο STEP AP203 για τον κύβο με εσωτερική διαμόρφωση.	66
Εικόνα 27: Δύο κύβοι που εφάπτονται (α) και ο πίνακας χαρακτηριστικών συναρμολόγησης τους (b).....	67
Εικόνα 28: Μεγέθυνση πάνω στο μοντέλο OAM2DSM που αποτυπώνει τη μοντελοποίηση του ΠΔΣ η οποία περιγράφεται στην παρούσα ενότητα.	67
Εικόνα 29: Μετατροπή ενός ΠΔΣ (DSM) σε ένα ΠΔΣ Συναρμολογήσεων (ADSM). Ο γράφος αποτυπώνει τον αρχικό ΠΔΣ ενώ οι ακμές του αποτυπώνουν τη συναρμολόγηση (Π.χ As(1,2) - συναρμολόγηση του εξαρτήματος 1 με το εξάρτημα 2).....	68

Εικόνα 30: Πίνακας Δομής Σχεδίασης συστατικών που αποτυπώνει στις σχέσεις μεταξύ των εξαρτημάτων του προϊόντος.....	74
Εικόνα 31: Μη κατευθυνόμενος γράφος για το προϊόν. Οι κορυφές αναπαριστούν τα εξαρτήματα του προϊόντος ενώ οι ακμές, που για ευκολία είναι αριθμημένες, αναπαριστούν τις συνδέσεις μεταξύ τους.	75
Εικόνα 32: Ο ΠΔΣ συναρμολόγησης (ADSM) που εξάχθηκε από τον πίνακα γειτνίασης (αρχικό ΠΔΣ συστατικών).	75
Εικόνα 33: Στιγμιότυπο από τον αλγόριθμο ομαδοποίησης.	76
Εικόνα 34: Ο ΠΔΣ των συναρμολογήσεων χρωματισμένος ανά ομάδα στοιχείων.....	76
Εικόνα 35: Έξοδος από τον αλγόριθμο στον οποίο όλα τα στοιχεία είναι ομαδοποιημένα, ενώ δεξιά αποτυπώνεται η εξέλιξη ένταξης των στοιχείων στην κάθε ομάδα.....	77
Εικόνα 36: Αποτύπωση της μεγαλύτερης διαφοράς κόστους μεταξύ των ομάδων.	77
Εικόνα 37: Ομαδοποίηση για αριθμό ομάδων N=3	78
Εικόνα 38: Ομαδοποίηση για αριθμό ομάδων N=4	78
Εικόνα 39: Ομαδοποίηση για αριθμό ομάδων N=5	79
Εικόνα 40: Ομαδοποίηση για αριθμό ομάδων N=6	79
Εικόνα 41: Ομαδοποίηση για αριθμό ομάδων N=7	80
Εικόνα 42: Λογικό διάγραμμα ενοποίησης των μοντέλων OAM/CPM-NIST και ΠΔΣ.....	82
Εικόνα 43: Σχεδίαση του προσωπικού υπολογιστή με χρήση του InventorAutodesk®.....	84
Εικόνα 44: Τμήμα του αρχείου STEP που αντιστοιχεί στη σχεδίαση των εξαρτημάτων του προσωπικού υπολογιστή. Στην εικόνα φαίνονται οι οντότητες Next_Assembly που χρησιμοποιείται για την εύρεση της ιεραρχίας.	85
Εικόνα 45: Ο πίνακας Artifact αποθηκεύει όλα τα έργα που υπάρχουν μέσα στο OAM2DSM καθώς και τη διαδρομή των αρχείων που θα χρησιμοποιηθούν ή θα εξαχθούν.....	86
Εικόνα 46: Τα εξαρτήματα και η ποσότητά τους για τον υπό συναρμολόγηση υπολογιστή. .	86
Εικόνα 47: Στιγμιότυπο από τον πίνακα Assembly_ ο οποίος από το πεδίο AssemblyOrPart διακρίνει αν πρόκειται για εξάρτημα ή υποσυναρμολόγηση.	86
Εικόνα 48: Ο πίνακας Assembly_Hierarchy αποτυπώνει την ιεραρχία συναρμολόγησης κάνοντας χρήση των σχέσεων ArtifactHasPart και PartOfArtifact.....	87
Εικόνα 49: Αποτύπωση με γραφικό τρόπο της ιεραρχίας συναρμολόγησης όπως αυτή υπάρχει μέσα στο αρχείο STEP.	88
Εικόνα 50: Στιγμιότυπο του πίνακα Constraint που αποτυπώνει χαρακτηριστικά συναρμολόγησης που ανιχνεύτηκαν στην αρχική σχεδίαση.	89
Εικόνα 51: Στιγμιότυπο του πίνακα AC ο οποίος αποτυπώνει την συσχέτιση των εξαρτημάτων μεταξύ τους.	89
Εικόνα 52: Γραφική αναπαράσταση των συνδέσεων μεταξύ των εξαρτημάτων στην ήδη υπάρχουσα ιεραρχία συναρμολόγησης.....	90

Εικόνα 53: Στιγμιότυπο από τα είδη των συνδέσεων που επιτρέπονται στα συναρμολογούμενα εξαρτήματα.....	90
Εικόνα 54: Στιγμιότυπο από το διάγραμμα συναρμολόγησης σύμφωνα με το OAM/CPM-NIST.	91
Εικόνα 55: Στιγμιότυπο από το διάγραμμα συναρμολόγησης σύμφωνα με το OAM/CPM-NIST για τη Μητρική τον Επεξεργαστή και τις κάρτες Ήχου, Γραφικών και Δικτύου.	92
Εικόνα 56: Πίνακας Δομής Σχεδίασης για τον υπολογιστή.	96
Εικόνα 57: Ο ΠΔΣ όπως αποτυπώνεται στο OAM2DSM μετά την κοστολόγηση κάθε συναρμολόγησης.	96
Εικόνα 58: Ο πίνακας ADSM που περιγράφει τις σχέσεις μεταξύ των συναρμολογήσεων. ...	97
Εικόνα 59: Έξοδος από τον αλγόριθμο για N=2.	98
Εικόνα 60: Έξοδος από τον αλγόριθμο για N=3	99
Εικόνα 61: Έξοδος από τον αλγόριθμο για N=4	99
Εικόνα 62: Ομαδοποίηση συναρμολογήσεων για N=2	100
Εικόνα 63: Ομαδοποίηση συναρμολογήσεων για N=3	101
Εικόνα 64: Το μενού Αρχείο από το οποίο ο χρήστης μπορεί είτε να δημιουργήσει νέο έργο είτε να ανοίξει ένα από τα αποθηκευμένα.....	133
Εικόνα 65: Επιλογή του αρχείου STEP για τη δημιουργία νέου έργου.	134
Εικόνα 66: Επιλογή φακέλου για τα αρχεία του νέου έργου.....	134
Εικόνα 67: Πίνακας εξαρτημάτων του αρχείου STEP, περιγραφή και ποσότητα.	135
Εικόνα 68: Εισαγωγή των δεδομένων όπως αποτυπώνονται στο παράθυρο στη βάση δεδομένων του OAM.....	136
Εικόνα 69: Ξεκινά η διαδικασία εισαγωγής δεδομένων.	136
Εικόνα 70: Ολοκλήρωση της διαδικασίας εισαγωγής νέου έργου με τα πληροφοριακά παράθυρα.	137
Εικόνα 71: Η επιλογή OAM και το μενού εσωτερικά της.....	137
Εικόνα 72: Το μοντέλο OAM και οι σχέσεις του με τα υπόλοιπα μοντέλα στην ανάπτυξη του προϊόντος. Η επιλογή DSM οδηγεί στην ολοκλήρωση των δύο μοντέλων και είναι από τα τελικά βήματα στην επιλογή του χρήστη (Επιλογή OAM >OpenAssemblyModel)	138
Εικόνα 73: Αποτύπωση δύο στιγμιότυπων της συναρμολόγησης εξαρτημάτων καθώς και το κόστος τους.	139
Εικόνα 74: Τροποποίηση στις παραμέτρους της συναρμολόγησης.....	139
Εικόνα 75: Εμφάνιση των τεχνουργημάτων που διαθέτει η βάση δεδομένων.....	140
Εικόνα 76: Εμφάνιση των εικόνων των εξαρτημάτων κάθε τεχνουργήματος.	140

Κατάλογος Πινάκων

Πίνακας 1: Ταξινόμηση των ΠΔΣ	22
Πίνακας 2: Ανάλυση σχέσεων σε αριθμητικούς ΠΔΣ συστατικών (17)	24
Πίνακας 3: Οι αφηρημένες κλάσεις αντικειμένων και οντότητες του CPM-NIST	43
Πίνακας 4: Οι κλάσεις – στιγμιότυπα του CPM	44
Πίνακας 5: Κλάσεις σχέσεων για το CPM- NIST	44
Πίνακας 6: Πίνακας δομών δεδομένων για τις κλάσεις Artifact, Part και Assembly	57
Πίνακας 7: Ιδιότητες της Next_Assembly_Usage_Occurence οντότητας	62
Πίνακας 8: Συμβολισμός των ιδιοτήτων της παραπάνω οντότητας για τις ανάγκες της λεξικογραφικής ανάλυσης	62
Πίνακας 9: Πίνακας κριτηρίων κόστους συναρμολόγησης (118)	70
Πίνακας 10: Υπολογισμός του δείκτη Dunn για $N=2, \dots, 7$	81
Πίνακας 11: Πίνακας εξαρτημάτων του προσωπικού υπολογιστή	84
Πίνακας 12: Σχέσεις συναρμολόγησης μεταξύ των εξαρτημάτων CaseBox, Motherboard και PowerSupplyUnit.	91
Πίνακας 13: Σχέσεις συναρμολόγησης της Μητρικής πλακέτας	92
Πίνακας 14: Σχέσεις συναρμολόγησης για τον υπολογιστή όπως έχουν εξαχθεί από τη βάση δεδομένων που υλοποιεί το μοντέλο OAM/CPM-NIST	96
Πίνακας 15: Επιλογές για το χρήστη από το μενού File	133

Περίληψη

Η διαδικασία σχεδίασης και ανάπτυξης ενός προϊόντος περιλαμβάνει πολλές οργανωτικές μονάδες και δραστηριότητες οι οποίες απαιτούνται με σκοπό ένα τελικό προϊόν να εισαχθεί στην αγορά. Στη σχεδίαση και ανάπτυξη προϊόντων με χρήση υπολογιστή, υπάρχουν πολλοί διαφορετικοί τύποι και εκδόσεις των λογισμικών που χρησιμοποιούνται για την υποστήριξη των δραστηριοτήτων στην ανάπτυξη προϊόντων και την εκτέλεση των διαδικασιών σχεδιασμού και παραγωγής. Διαφορετικά συστήματα μπορούν να χρησιμοποιούν διαφορετικά σχήματα αναπαράστασης πληροφοριών. Αυτό δυσκολεύει την ολοκλήρωση και το μοίρασμα των πληροφοριών στο σχεδιασμό και την ανάπτυξη των προϊόντων ακόμη και μέσα σε μια ενιαία επιχείρηση. Η τεχνολογία μοντελοποίησης σε διαδικασίες σχεδιασμού προϊόντων διαδραμάτισε έναν θεμελιώδη ρόλο στην υποστήριξη της αποτελεσματικής διανομής πληροφοριών και γνώσης.

Η διαδικασία μοντελοποίησης μπορεί να διαιρεθεί σε δύο φάσεις: Η πρώτη φάση περιλαμβάνει κανονικά την ανάλυση των αντικειμένων που διαμορφώνονται και τον καθορισμό της γενικής δομής ενός προτύπου προϊόντων, ενώ στη δεύτερη φάση το πρότυπο αποκτά πλέον πιο φυσική και πραγματική μορφή. Αυτό σημαίνει ότι εκτός από τις μεθοδολογίες και την ανάλυση των αντικειμένων που ήδη έχουν γίνει στο προηγούμενο στάδιο θα πρέπει να αποτυπωθούν όλες εκείνες οι κλάσεις αντικειμένων, οι δομές δεδομένων, καθώς και οι μηχανισμοί στη διαχείριση της πληροφορίας για τις διάφορες δραστηριότητες του σχεδιασμού. Για παράδειγμα, μέσα στο πρότυπο προϊόντων είναι αποθηκευμένα τα πραγματικά στοιχεία όπως είναι μια βάση δεδομένων. Κατά συνέπεια, το πρότυπο προϊόντων μπορεί να θεωρηθεί ως ένας πυρήνας πληροφοριών ο οποίος υποστηρίζει τις διαδικασίες ανάπτυξης προϊόντος. Στην παρούσα έρευνα οι διάφορες μεθοδολογίες διαμόρφωσης προϊόντων θα επανεξετασθούν, με ιδιαίτερη έμφαση στο πρότυπο CPM (Core Product Model) του NIST (National Institute of Standards and Technology) όπως επίσης και η μεθοδολογία DSM (Design Structure Model) η οποία ανήκει στα ακολουθιακά και χρονικά μοντέλα.

Στην προτεινόμενη έρευνα θα ενοποιηθούν τα δύο είδη μοντέλων, τα μοντέλα του προϊόντος με τα μοντέλα ροής των εργασιών, ώστε να γίνει μια ολοκλήρωση των πρώτων πέραν των ορίων της παραγωγής και για τα δεύτερα να ανταπεξέλθουν στα προβλήματα που έχουν με την εγκυρότητα δεδομένων.

Η έρευνα στην εργασία αυτή σε ότι αφορά το μοντέλο CPM επικεντρώθηκε σε ζητήματα τα οποία παραμένουν ανοικτά για την ολοκλήρωση του μοντέλου. Συγκεκριμένα, κάνοντας χρήση μιας επέκτασης του αρχικού εννοιολογικού μοντέλου του Μοντέλου Ανοικτής Συναρμολόγησης (Open Assembly Model - OAM), επιχειρεί να υλοποιήσει το Ενδιάμεσο Μοντέλο παρέχοντας τις δομές δεδομένων που χρειάζονται για την αποθήκευση των αντικειμένων και στη συνέχεια προχωρά στην υλοποίηση μιας αντικειμενοστραφούς πλατφόρμας ολοκληρώνοντας το σε Μοντέλο Υλοποίησης.

Για την ενοποίηση του μοντέλου Πίνακα Δομής Σχεδίασης (Design structure Model - DSM) και του Μοντέλου Ανοικτής Συναρμολόγησης (Open Assembly Model - OAM/NIST) χρησιμοποιήθηκε ο Πίνακας Δομής Σχεδίασης Συστατικών ο οποίος με τη βοήθεια αλγορίθμων ομαδοποίησης εστιάζει στην εύρεση υποσυνόλων στοιχείων μιας συναρμολόγησης, ελαχιστοποιώντας προβλήματα στην αρχική σχεδίαση (επαναλήψεις και αναδράσεις).

Η ενοποίηση των δύο μοντέλων επιτυγχάνεται κάνοντας χρήση της τεχνολογίας αγωγών

(pipeline) που χρησιμοποιείται ευρέως στην τεχνολογία λογισμικού. Συγκεκριμένα, η πληροφορία ρέει από το μοντέλο OAM στο μοντέλο DSM δημιουργώντας έτσι εικονικούς αγωγούς ροής πληροφορίας από τα στάδια της αρχικής σχεδίασης μέχρι τη συναρμολόγηση του προϊόντος. Η τροφοδότηση πληροφορίας από το ένα μοντέλο στο άλλο απαιτεί βέβαια μετασχηματισμούς δεδομένων που πρέπει να παρέχουν και εννοιολογική και σημασιολογική αναπαράσταση.

Το μοντέλο ενοποίησης που προκύπτει (Open Assembly Model to Design Structure Matrix - OAM2DSM) συνδέει τις δύο αυτές μεθόδους χρησιμοποιώντας το OAM/NIST ως μοντέλο διεπαφής μεταξύ των λογισμικών σχεδίασης (Computer Aided Design systems - CAD systems) και του μοντέλου DSM. Η ενοποίηση αυτή:

- Υποστηρίζει τη διαδικασία σχεδίασης από τα πρώιμα στάδια της και καθ' όλη τη διάρκεια του κύκλου ζωής ενός προϊόντος.
- Ισχυροποιεί την ενοποίηση συστημάτων σχεδίασης CAD με συστήματα ανάλυσης.
- Δίνει μια δυνατότητα πρόβλεψης στη σχεδίαση, συμβάλλοντας έτσι στη μείωση του χρόνου σχεδίασης μιας και εντοπίζονται γρηγορότερα σχεδιαστικές αστοχίες ή λάθη.
- Παρέχει ένα ισχυρό πληροφοριακό μοντέλο, ικανό να αποτυπώσει την εξέλιξη της σχεδίασης και τη ροή της πληροφορίας σε όλα της τα στάδια.
- Είναι ανεξάρτητη πλατφόρμας σχεδίασης (Απαραίτητη προϋπόθεση είναι το λογισμικό σχεδίασης να υποστηρίζει το πρότυπο ISO10303 STEP).
- Προσθέτει στο OAM ένα επιπλέον χαρακτηριστικό, αυτό της τμηματικής σχεδίασης μιας και η ομαδοποίηση που προτείνεται στον ΠΔΣ αρχιτεκτονικής (συστατικών) δίνει ομάδες συναρμολογήσεων με ισχυρή αλληλεπίδραση και εξάρτηση. Αυτή η δυνατότητα ισχυροποιεί τη συνεργατική σχεδίαση (collaborative design).

Τέλος, το μοντέλο OAM2DSM θα μπορούσε να αποτελέσει τη βασική πλατφόρμα για τη δημιουργία ενός πληροφοριακού συστήματος που θα ενσωματώνει όλα εκείνα τα χαρακτηριστικά που είναι απαραίτητα για την πλήρη αποτύπωση της διαδικασίας σχεδίασης και συναρμολόγησης ενός προϊόντος.

Abstract

The product design and development process includes several organizational units and activities that are required so that a final product reaches the market. There are many different types and versions of software used to support the product development activities, such as product modeling, design and manufacturing process. Different systems may use different information representation schemes. This complicates the integration and sharing of information in the product design and development even within a single enterprise. The modeling technology in product design processes is playing a fundamental role in supporting the efficient distribution of information and knowledge. Consequently, the product model can be regarded as an information kernel which supports product development processes.

For our research the various products formulation methodologies will be reviewed. Particular emphasis will be given to CPM standard, developed by NIST, as well as to the Design Structure Model methodology which belongs to the sequential models. The proposed research will integrate the two types of models, product models with the information flow model in order to make an integration of the first beyond the boundaries of production and for the second one to cope with problems related with the validity of data.

The research in this thesis in terms of CPM model focused on issues that remain open in order to enhance it. Specifically, using an extension of the original conceptual model of the Open Assembly Model (OAM), we attempt to implement Intermediate model providing the data structures needed for the storage of objects and then proceeding with the implementation of an object-oriented platform in completing the Implementation Model.

For the integration of the Design Structure model (DSM) and the Open Assembly Model (OAM / NIST) a component-based Design Structure Matrix is used with the help of clustering algorithms focusing on finding subsets of a data assembly in order to minimize problems in the original design (repetitions and feedbacks). The integration of the two models is achieved using the technology of pipelines, which is widely used in software engineering.

The new model (OAM2DSM) connects these two methods using the CPM / NIST as a model interface between design software (CAD systems) and the DSM model. This integration helps to support the design process from the early stages and throughout the life cycle of a product. It strengthens the integration of design systems (CAD) with model analyzers. It gives predictability in product design, thus contributing to the reduction of time in planning and identifying faster design failures or mistakes. It provides a powerful information model, able to capture the evolution of the design and flow of information in all stages. It is independent from design platforms (prerequisite is the design software supports the standard ISO10303 STEP).

Finally, the model OAM2DSM could be the basic platform for the creation of an information system that integrates all the features needed for a complete mapping of the design process and assembly of a product.

1. Εισαγωγή

Η ανάπτυξη προϊόντων και συστημάτων μεγάλης κλίμακας είναι μια πολύπλοκη διεργασία διότι αφενός πρέπει να διευθετηθούν αρκετά τεχνικά θέματα, αφετέρου το ανθρώπινο δυναμικό που συμμετέχει σε αυτήν τη διεργασία πρέπει να οργανωθεί σε κατάλληλες δομές κατά τη διάρκεια της διεργασίας αυτής. Η διαδικασία σχεδίασης και ανάπτυξης ενός προϊόντος περιλαμβάνει πολλές οργανωτικές μονάδες και δραστηριότητες οι οποίες απαιτούνται με σκοπό ένα τελικό προϊόν να εισέλθει στην αγορά. Η έρευνα για τη ανάπτυξη προϊόντος περιλαμβάνει εργασίες από διαφορετικές περιοχές των ερευνητικών κοινοτήτων. Τα τελευταία χρόνια, διάφορες τεχνικές έχουν αναπτυχθεί συμβάλλοντας στην ενοποίηση - ολοκλήρωση των πληροφοριών για το προϊόν. Εντούτοις, στη σχεδίαση και ανάπτυξη προϊόντων με χρήση υπολογιστή, υπάρχουν πολλοί διαφορετικοί τύποι και εκδόσεις των λογισμικών που χρησιμοποιούνται για την υποστήριξη των δραστηριοτήτων στην ανάπτυξη προϊόντων. Διαφορετικά συστήματα χρησιμοποιούν διαφορετικά σχήματα αναπαράστασης πληροφοριών. Αυτό δυσκολεύει την ολοκλήρωση και το μοίρασμα των πληροφοριών στο σχεδιασμό και ανάπτυξη των προϊόντων, ακόμη και μέσα σε μια ενιαία επιχείρηση. Η τεχνολογία μοντελοποίησης σε διαδικασίες σχεδιασμού προϊόντων διαδραμάτισε ένα θεμελιώδη ρόλο στην υποστήριξη της αποτελεσματικής διανομής πληροφοριών και γνώσης.

Σε διαδικασίες σχεδιασμού προϊόντος ένα πρότυπο μοντέλο προϊόντων απαιτείται για να περιγράψει με πληρότητα και σαφήνεια όλες τις πληροφορίες που αφορούν τη σχεδίαση του προϊόντος σε όλες τις φάσεις του. Η διαδικασία μοντελοποίησης μπορεί να διαιρεθεί σε δύο φάσεις:

- Η πρώτη φάση περιλαμβάνει την ανάλυση των αντικειμένων που διαμορφώνονται και τον καθορισμό της γενικής δομής ενός προτύπου προϊόντων. Παράλληλα, οι μεθοδολογίες, η τεχνολογία μοντελοποίησης καθώς επίσης το λογισμικό και η τεχνολογία διαμόρφωσης, καθορίζονται και προετοιμάζονται.
- Στη δεύτερη φάση διαμορφώνεται ένα πρότυπο προϊόντων ή συγκεκριμένα το πρότυπο στοιχείων-δεδομένων προϊόντων. Αυτό το πρότυπο πληροφοριών είναι μια σημαντική βάση στις διαδικασίες μοντελοποίησης και ορίζεται ως ένας μηχανισμός για να οργανώσει και να δομήσει τα στοιχεία προϊόντων για τις διάφορες δραστηριότητες του σχεδιασμού τους. Το συνολικό μοντέλο και οι δομές που ορίζονται σε προηγούμενα στάδια, είναι λεπτομερή σε περιγραφή και αντιπροσωπεύονται από διαφορετικές τεχνολογίες διαμόρφωσης – μοντελοποίησης. Το πρότυπο πληροφοριακό μοντέλο προϊόντων μπορεί να χρησιμοποιηθεί περαιτέρω ως βασικό στοιχείο για την ανάπτυξη νέων προϊόντων.

Παράλληλα, στη δεύτερη και τελική φάση, το πρότυπο πληροφοριακό μοντέλο προϊόντων υλοποιείται για να παράγει το μοντέλο του προϊόντος. Μέσα στο πρότυπο προϊόντων είναι αποθηκευμένα τα πραγματικά στοιχεία, όπως είναι για παράδειγμα μια βάση δεδομένων. Κατά συνέπεια, το πρότυπο προϊόντων μπορεί να θεωρηθεί ως ένας πυρήνας πληροφοριών ο οποίος υποστηρίζει τις διαδικασίες ανάπτυξης προϊόντος.

Πολλές μεθοδολογίες μοντελοποίησης προϊόντων έχουν περιγραφεί στη βιβλιογραφία και μπορούν να ταξινομηθούν ως εξής [1]:

- Στερεά μοντελοποίηση προϊόντων (Solid product modelling),
- μοντελοποίηση προϊόντων με βάση τα μορφολογικά χαρακτηριστικά (Feature-based product modelling),
- βασισμένα στη γνώση μεθοδολογία διαμόρφωσης προϊόντων (Knowledge-based product modelling) και

- ενοποιημένη μεθοδολογία διαμόρφωσης προϊόντων (Integrated product modelling).

Στην παρούσα έρευνα οι διάφορες μεθοδολογίες διαμόρφωσης προϊόντων θα επανεξεταστούν με ιδιαίτερη έμφαση στο πρότυπο CPM του NIST. Στη διαδικασία σχεδίασης προϊόντων, όπως και σε άλλες διαδικασίες, είναι δυνατό και χρήσιμο να σχεδιαστούν ποσοτικά προσδιορίσιμα πρότυπα - μοντέλα. Οι στόχοι της μοντελοποίησης είναι αρκετοί, περιλαμβάνοντας την εκμάθηση της διαδικασίας σχεδιασμού καθώς και τις μεθοδολογίες με τις οποίες μια διαδικασία μπορεί να ελεγχθεί. Υπάρχουν διάφορα μοντέλα της ανάπτυξης προϊόντος και έχει γίνει σημαντική ερευνητική εργασία πρόσφατα στην ανάπτυξη ή τη βελτίωση των προτύπων. Υπάρχουν επίσης διάφορα κριτήρια από τα οποία πρέπει να κριθεί η αξία των προτεινόμενων προτύπων της διαδικασίας ανάπτυξης προϊόντος. Οι σχετικές ερωτήσεις που τίθενται είναι:

- Αντιμετωπίζει ένα σημαντικό διαχειριστικό ζήτημα;
- Η απόφαση που παίρνεται είναι βασισμένη σε πληροφορίες που είναι διαθέσιμες και ακριβείς;
- Οι υποθέσεις και οι απλοποιήσεις του προτύπου είναι λογικές;
- Το πρότυπο μοντέλο είναι υπολογιστικά υλοποιήσιμο;

Μόλις συγκεκριμενοποιηθούν τα κριτήρια από τα οποία τα πρότυπα μοντέλα μπορούν να κριθούν, πρέπει να ορισθεί μια πλατφόρμα αξιολόγησης στην οποία αυτά τα κριτήρια θα αξιολογηθούν. Τα πρότυπα διαδικασίας ανάπτυξης προϊόντος μπορούν να αντιμετωπιστούν ως ένα πληροφοριακό μοντέλο ροής της πληροφορίας, από στάδιο σε στάδιο, στη διαδικασία ανάπτυξης. Τέτοια πληροφοριακά μοντέλα έχουν χρήση όχι μόνο για τον έλεγχο και την αξιολόγηση των προγραμμάτων, αλλά και για την ανάπτυξη κοινών στόχων και την κατανόηση αυτών μεταξύ των μελών της ομάδας προγράμματος. Πρέπει επομένως να εξεταστούν, όχι μόνο τα άμεσα οφέλη των προτύπων (που είναι συχνά δύσκολο να καθοριστούν), αλλά και τα έμμεσα οφέλη από την υλοποίηση του μοντέλου.

Οι Smith και Morrow [2] έχουν κάνει μια πρώτη ανασκόπηση της διαδικασίας μοντελοποίησης του προϊόντος και έχουν ομαδοποιήσει τα μοντέλα σε πέντε πρότυπες κατηγορίες, ανάλογα με το τι προσπαθεί κάθε μοντέλο να επιτύχει. Οι πέντε κατηγορίες είναι:

- Τα ακολουθιακά και χρονικά μοντέλα (sequencing and scheduling models),
- τα μοντέλα αποσύνθεσης (decomposition models),
- τα στοχαστικά μοντέλα (stochastic lead time models),
- τα μοντέλα αναθεώρησης σχεδίου (design review timing models) και τα
- παράλληλα μοντέλα (parallelism models).

Η παρούσα έρευνα επικεντρώνεται κυρίως στη μεθοδολογία Πίνακα Δομής Σχεδίασης (ΠΔΣ) – Design Structure Model–DSM η οποία ανήκει στα ακολουθιακά και χρονικά μοντέλα. Η μέθοδος ΠΔΣ υποθέτει ότι κάθε κομμάτι της δραστηριότητας σχεδιασμού μπορεί να αποτυπωθεί ως τμήμα πληροφορίας, δημιουργώντας νέες πληροφορίες. Οι πληροφορίες παραγωγής από ένα τμήμα σχεδιασμού γίνονται οι πληροφορίες εισαγωγής σε ένα άλλο σημείο. Οι σχέσεις εισόδου-εξόδου μπορούν να περιλαμβάνουν και κύκλους, οι οποίοι δείχνουν την ανάγκη για επανάληψη. Ο ΠΔΣ υλοποιείται με τη βοήθεια πίνακα ο οποίος αποτυπώνει τις παραπάνω σχέσεις. Ο στόχος σε αυτήν τη μοντελοποίηση είναι να επανατοποθετηθούν διαδοχικά οι δραστηριότητες του σχεδιασμού, βοηθώντας έτσι να προσδιοριστούν οι κυκλικές και μη δραστηριότητες.

Στην προτεινόμενη έρευνα, θα ενοποιηθούν τα δύο είδη μοντέλων, τα μοντέλα του προϊόντος με τα μοντέλα ροής των εργασιών, ώστε να γίνει μια ολοκλήρωση των πρώτων πέραν των ορίων της παραγωγής και για τα δεύτερα να ανταπεξέλθουν στα προβλήματα που έχουν με την εγκυρότητα δεδομένων.

2. Στάθμη των γνώσεων

2.1 Γενικά για τα μοντέλα ανάπτυξης προϊόντος

Τα μοντέλα σχεδίασης και ανάπτυξης προϊόντων που έχουν προταθεί κατά καιρούς, ακολουθούν τη λογική ότι η διαδικασία σχεδίασης επιμερίζεται σε ένα διακριτό σύνολο εργασιών που πρέπει να υλοποιηθούν, από διαφορετικές ομάδες συνήθως, για να υλοποιηθεί το τελικό αποτέλεσμα [1]. Ένα μοντέλο προϊόντος θα πρέπει να περιγράφει με πληρότητα και σαφήνεια όλες τις πληροφορίες που αφορούν το σχεδιαζόμενο προϊόν. Η διαδικασία μοντελοποίησης διαιρείται σε διαφορετικές φάσεις που περιλαμβάνουν από την ανάλυση των αντικειμένων που πρόκειται να μοντελοποιηθούν, τον ορισμό της συνολικής δομής του μοντέλου καθώς και δεδομένα του μοντέλου. Το τελικό μοντέλο που προκύπτει μπορεί να χαρακτηριστεί ως πληροφοριακό μοντέλο μιας και αποτελεί ουσιαστικά το μηχανισμό που οργανώνει και δομεί τα δεδομένα που είναι απαραίτητα για όλες τις δραστηριότητες στην ανάπτυξη προϊόντων [2].

Ιστορικά μια από τις πρώτες και πιο δημοφιλείς τεχνικές που χρησιμοποιήθηκαν στην πράξη και χρησιμοποιείται και ως σήμερα για τον προγραμματισμό ενός έργου, δημιουργήθηκε από τον Henry Gantt το 1903. Η βασική ιδέα της μεθόδου στηρίζεται στο γεγονός ότι κάθε έργο μπορεί να χωριστεί σε μικρότερες εργασίες των οποίων το κόστος και τη διάρκεια είναι δυνατόν να εκτιμηθούν. Ένα διάγραμμα Gantt είναι ένα διάγραμμα στο επίπεδο που συσχετίζει και αποτυπώνει τις δραστηριότητες ενός έργου μέσα στο χρόνο [3].

Στη συνέχεια αναπτύχθηκαν και άλλες μέθοδοι με σημαντικότερη τη μέθοδο CPM (Critical Path Method - μέθοδος της κρίσιμης διαδρομής) που αναπτύχθηκε για πρώτη φορά τη δεκαετία του '50 από τους από Kelley J .E (Remington Rand Ltd) και Walker M. R (DuPont Ltd). Η μέθοδος προσπαθεί να ισορροπήσει δυο παράγοντες μέσα στην κατασκευή ενός έργου που είναι ανταγωνιστικοί μεταξύ τους. Το χρόνο για την περαίωση μιας δραστηριότητας και το κόστος που αυξάνει από την καθυστέρηση. Χρησιμοποιείται για τη χάραξη της κρίσιμης διαδρομής η οποία είναι μια σειρά επιμέρους εξαρτημένων εργασιών (όπου η κάθε εργασία εξαρτάται από την προηγούμενή της) που αποτελούν τη μεγαλύτερη διαδρομή στο δίκτυο και που θα πρέπει να υλοποιηθούν έγκαιρα προκειμένου το συνολικό έργο να ολοκληρωθεί με ικανοποιητικά αποτελέσματα [4].

Χρονικά κοντά με τη CPM αναπτύχθηκε και η μέθοδος PERT [5, 6] (Program Evaluation and Review Technique), που χρησιμοποιήθηκε για πρώτη φορά στα τέλη της δεκαετίας του '50 από το πολεμικό ναυτικό των ΗΠΑ για το πρόγραμμα Polaris. Η μέθοδος PERT διαφοροποιείται από την CPM επιτρέποντας την αβεβαιότητα στους χρόνους ολοκλήρωσης των επιμέρους εργασιών ενός έργου.

Οι μέθοδοι που ήδη έχουν αναφερθεί έχουν περιορισμούς ως προς τη λειτουργικότητα τους. Συχνά, κατά τη διαδικασία ανάπτυξης και σχεδίασης προϊόντων, υπάρχουν αναδράσεις και επαναλήψεις διεργασιών κάτι που δε χειρίζονται αποτελεσματικά [2]. Για το λόγο αυτό, διαμορφώθηκαν και άλλα μοντέλα που μπορούσαν να αποτυπώσουν και να μοντελοποιήσουν καλύτερα τα στάδια ενός προϊόντος από την αρχική ιδέα μέχρι και τη διάθεση του στην αγορά. Τα μοντέλα αυτά χωρίζονται σε πέντε κατηγορίες με κριτήριο την αρχιτεκτονική τους και τι επιδιώκεται με τη χρήση τους [2]. Οι πέντε αυτές κατηγορίες είναι:

- Σειριακά και ακολουθιακά μοντέλα,
- μοντέλα αποσύνθεσης,
- στοχαστικά μοντέλα χρονική ανοχής,

- χρονικά μοντέλα ανασκόπησης και
- μοντέλα παραλληλισμού.

2.2 Σειριακά και σχεδιαστικά μοντέλα

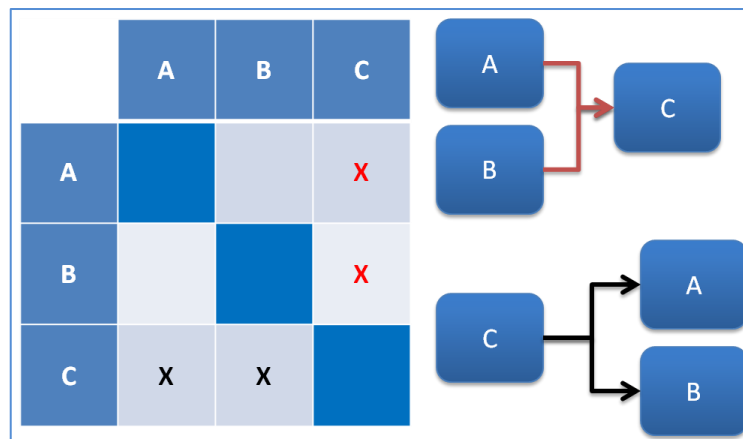
2.2.1 Πίνακας Δομής Σχεδίασης

Όπως ήδη αναφέρθηκε, μέθοδοι όπως οι PERT και CPM, αν και χρησιμοποιούνται στη βιομηχανία, δεν μπορούν να ανταποκριθούν επαρκώς στις σύγχρονες απαιτήσεις στην ανάπτυξη και σχεδίαση προϊόντων [7]. Μια δημοφιλής οικογένεια εργαλείων μοντελοποίησης που άρχισε να χρησιμοποιείται ευρέως, τόσο στην έρευνα, όσο και στη βιομηχανία, είναι ο Πίνακας Δομής Σχεδίασης - ΠΔΣ (DSM - Design Structure Matrix) [8]. Πρόκειται ουσιαστικά για έναν πίνακα ισοδύναμο με τον πίνακα γειτνίασης της θεωρίας γράφων. Η χρήση για μοντελοποίηση συστημάτων με τον ΠΔΣ, αν και ξεκίνησε από τη δεκαετία του '80 ουσιαστικά χρησιμοποιήθηκε στη βιομηχανία από τις αρχές της δεκαετίας του '90 [10].

Ο ΠΔΣ είναι ένας πίνακας που μπορεί να αποτυπώσει τη ροή πληροφορίας μεταξύ στοιχείων που αποτελούν ή περιγράφουν ένα σύστημα. Ο ΠΔΣ είναι επίσης γνωστός και ως:

- Πίνακας εξάρτησης.
- Πίνακας Επίλυσης Προβλημάτων.
- Πίνακας αλληλεπίδρασης.
- Πίνακας N2 κ.λπ..

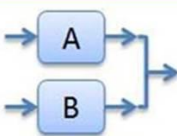
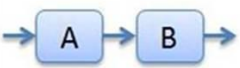
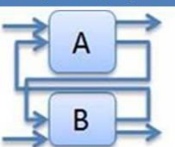
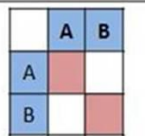
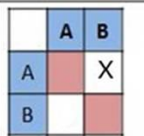
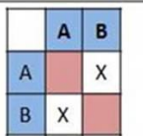
Η μοντελοποίηση του ΠΔΣ γίνεται με χρήση ενός τετραγωνικού πίνακα που κάθε στοιχείο ή δραστηριότητα αποτυπώνεται και αντιστοιχεί σε μια σειρά και μια στήλη. Αν δύο στοιχεία (έστω i και j) δε σχετίζονται μεταξύ τους, τότε το στοιχείο (i,j) του πίνακα παραμένει κενό. Αν δύο στοιχεία συσχετίζονται (για παράδειγμα τα i και j στοιχεία), τότε αυτό αποτυπώνεται με ένα σύμβολο (π.χ. το X) στη θέση (i,j) του πίνακα. Διαβάζοντας τον πίνακα ανά γραμμή i δηλώνονται όλα τα στοιχεία τα οποία η πληροφορία τους είναι αναγκαία για το στοιχείο της γραμμής i . Διαβάζοντας τον πίνακα ανά στήλη j δηλώνονται τα στοιχεία τα οποία απαιτούν την πληροφορία από το στοιχείο j . Με βάση τη μοντελοποίηση που ήδη περιγράφηκε, είναι φανερό ότι τα στοιχεία που είναι κάτω από την κύρια διαγώνιο του πίνακα δηλώνουν προώθηση της πληροφορίας σε στοιχεία που έπονται χρονικά, ενώ τα στοιχεία που είναι πάνω από την κεντρική διαγώνιο απεικονίζουν ανατροφοδότηση πληροφορίας σε στοιχεία που έχουν προηγηθεί. Στην εικόνα 1 αποτυπώνονται οι δύο βασικές αρχές υλοποίησης του ΠΔΣ.



Εικόνα 1: Οι βασικές αρχές υλοποίησης και ανάγνωσης του ΠΔΣ

Σε έναν ΠΔΣ υπάρχουν τρεις βασικές δομές με κριτήριο την αλληλεπίδραση μεταξύ των στοιχείων του πίνακα όπως φαίνονται στην Εικόνα 2:

- Η παράλληλη ή ανεξάρτητη δομή.
- Η σειριακή ή εξαρτημένη δομή.
- Η συζευγμένη δομή.

Αλληλεπίδραση	Παράλληλη ή ανεξάρτητη	Σειριακή ή εξαρτημένη	Συζευγμένη ή Αλληλοεξαρτώμενη
Γράφημα			
Αναπαράσταση σε ΠΔΣ			

Εικόνα 2: Οι τρεις βασικές δομές αλληλεπίδρασης μεταξύ στοιχείων

Στην **παράλληλη** δομή δεν υπάρχει συσχέτιση μεταξύ των στοιχείων και επομένως μπορούν να υλοποιηθούν ανεξάρτητα. Στη **σειριακή** δομή το ένα στοιχείο εξαρτάται πλήρως από την πληροφορία ή την περαίωση του άλλου και τέλος στη **συζευγμένη** δομή υπάρχει ταυτόχρονη κυκλική εξάρτηση μεταξύ των στοιχείων, γνωστή και στον παράλληλο προγραμματισμό ως κατάσταση αδιεξόδου (deadlock situation).

2.2.2 Ταξινόμηση των ΠΔΣ

Οι ΠΔΣ χωρίζονται σε τέσσερεις κατηγορίες ως προς τα στοιχεία που μπορούν να αποτυπώσουν [11]:

1. ΠΔΣ αρχιτεκτονικής (συστατικών). Μοντελοποιούν συστήματα που έχουν διακριτά συστατικά και υποσυστήματα. Περιγράφουν τις αλληλεπιδράσεις μεταξύ τους [11,12].
2. ΠΔΣ ομάδων εργασίας. Μοντελοποιούν οργανωτικές δομές μιας εταιρείας ή οργανισμού και βασίζονται στις αλληλεπιδράσεις μεταξύ των εργαζομένων ή ομάδων εργασίας [11], [13,14].
3. ΠΔΣ προγραμματισμού δραστηριοτήτων και διεργασιών. Μοντελοποιούν τις δραστηριότητες, την αλληλεπίδραση μεταξύ τους και αποτυπώνουν τη ροή της πληροφορίας μεταξύ τους [11], [15].
4. Παραμετρικοί ΠΔΣ. Χρησιμοποιούνται για τη μοντελοποίηση στοιχείων χαμηλού επιπέδου όπως: παραμέτρων, αποφάσεων στη σχεδίαση, συστήματα εξισώσεων, χρήση κοινών παραμέτρων από υπορουτίνες κ.ά. [16].

Ως προς το χρόνο, οι ΠΔΣ στις δύο πρώτες κατηγορίες χαρακτηρίζονται στατικοί, ενώ στις δύο τελευταίες χρονικά εξαρτημένοι [Πίνακας 1]. Τέλος, μία επέκταση των δυαδικών ΠΔΣ είναι ο αριθμητικοί ΠΔΣ [17] οι οποίοι θα εξεταστούν αναλυτικότερα στη συνέχεια.

	Είδος ΠΔΣ	Αναπαράσταση	Εφαρμογές	Μέθοδος Ανάλυσης
Στατικοί	Αρχιτεκτονικής (Συστατικών)	Σχέσεις μεταξύ των συστατικών	Αρχιτεκτονική Συστημάτων, Μηχανική και Σχεδίαση	Ομαδοποίηση
	Οργάνωσης (Ομάδων Εργασίας)	Σχέσεις μεταξύ των εργαζομένων ή ομάδων εργασίας	Οργανωτική Σχεδίαση, Διαχείριση Αλληλεπιδράσεων, Ενσωμάτωση Ομάδων	Ομαδοποίηση
Χρονικά Εξαρτημένοι	Προγραμματισμού Δραστηριοτήτων	Ροή πληροφοριών μεταξύ Δραστηριοτήτων	Προγραμματισμός, Προσδιορισμός αλληλουχίας δραστηριοτήτων, Μείωση χρόνου ανάπτυξης προϊόντων	Αναδιάρθρωση Καθορισμός αλληλουχίας
	Παραμετρικοί	Σχέσεις μεταξύ των παραμέτρων σχεδίασης	Προσδιορισμός αλληλουχίας δραστηριοτήτων χαμηλού επιπέδου	Αναδιάρθρωση Καθορισμός αλληλουχίας

Πίνακας 1: Ταξινόμηση των ΠΔΣ [11].

2.2.3 Μέθοδοι ανάλυσης ΠΔΣ

Όλες οι μέθοδοι ανάλυσης των ΠΔΣ προσπαθούν να ελαχιστοποιήσουν τις αναδράσεις και τις επαναλήψεις μεταξύ των στοιχείων του συστήματος. Οι αναδράσεις και οι επαναλήψεις συνήθως προέρχονται από:

- Αποτελέσματα ελέγχων που αναγκαστικά γίνονται αργότερα.
- Προγραμματισμένες ή μη αναθεωρήσεις στη σχεδίαση.
- Σφάλματα κατά τη φάση της σχεδίασης.
- Περιορισμοί στη σχεδίαση στα υλικά κ.λπ..

Υπάρχουν δύο γενικές μέθοδοι ανάλυσης: η αναδιάρθρωση (partitioning) και η ομαδοποίηση (clustering). Η μέθοδος της αναδιάρθρωσης σε ένα ΠΔΣ, είναι μία διαδικασία ανακατάταξης των σειρών και των στηλών, έτσι ώστε να μην εμφανίζονται αναδράσεις και επαναλήψεις μεταξύ των στοιχείων του συστήματος [18]. Μαθηματικά, η αναδιάρθρωση προσπαθεί να μετασχηματίσει τον πίνακα σε κάτω διαγώνιο. Σε πραγματικές συνθήκες, ο μετασχηματισμός δεν είναι εφικτός πάντα οπότε η μέθοδος προσπαθεί να μετακινήσει τα στοιχεία του πίνακα που περιέχουν αναδράσεις κοντά στην κεντρική διαγώνιο [19]. Έτσι, περιλαμβάνονται λιγότερα στοιχεία του συστήματος στον κύκλο επανάληψης με αποτέλεσμα τη μείωση του χρόνου ανάπτυξης του προϊόντος. Ως προς τον τρόπο εντοπισμού των βρόγχων και των αναδράσεων, οι μέθοδοι αναδιάρθρωσης διακρίνονται σε αναζήτησης διαδρομών [20], εκθετικής γειννίασης πίνακα και μέθοδο πίνακα προσέγγισης.

Για την ανάλυση των ΠΔΣ αρχιτεκτονικής (συστατικών) και ΠΔΣ οργάνωσης (ομάδων), χρησιμοποιείται η μέθοδος της ομαδοποίησης, ο στόχος της οποίας είναι η εύρεση υποσυνόλων (ομάδων) με στοιχεία τα οποία έχουν αποκλειστική αλληλεξάρτηση, ή υποσυνόλων που έχουν την ελάχιστη αλληλεπίδραση με άλλα υποσύνολα [21, 22]. Με αυτόν τον τρόπο οι ομά-

δες που προκύπτουν μετά την ανάλυση εμπεριέχουν όλες τις αναδράσεις ή επαναλήψεις και παράλληλα ελαχιστοποιούνται οι αλληλεπιδράσεις μεταξύ των ομάδων [23].

2.2.4 Αριθμητικοί ΠΔΣ

Οι ΠΔΣ που αναφέρθηκαν μέχρι τώρα χρησιμοποιούν απλές δυαδικές αναπαραστάσεις καταγράφοντας το συσχετισμό ή μη μεταξύ των στοιχείων του πίνακα. Οι αριθμητικοί ΠΔΣ είναι μία επέκταση των δυαδικών ΠΔΣ που περιλαμβάνουν ένα πλήθος από χαρακτηριστικά με στόχο τη λεπτομερή περιγραφή των σχέσεων μεταξύ των στοιχείων. Αναλυτικότερα οι αριθμητικοί ΠΔΣ μπορεί να περιέχουν:

- Αριθμούς επιπέδων αναδράσεων που αποτυπώνουν τη σειρά με την οποία οι αναδράσεις θα πρέπει να αντιμετωπιστούν.
- Βαθμό εξάρτησης: Η συσχέτιση ή η εξάρτηση μεταξύ των στοιχείων μπορεί να χαρακτηριστεί ως μεγάλη, μεσαία, μικρή ή να πάρει μία μεταβλητή τιμή [Πίνακας 2].
- Την ποσότητα της πληροφορίας που ανταλλάσσεται μεταξύ των στοιχείων: Η αναδιάρθρωση ενός τέτοιου πίνακα μπορεί να μετασχηματιστεί ως πρόβλημα σε ελαχιστοποίηση του όγκου της πληροφορίας των αναδράσεων.
- Τη μεταβλητότητα της πληροφορίας: Ένα μέτρο μεταβλητότητας μπορεί να αποτυπώνει τη αβεβαιότητα με την οποία οι διεργασίες ανταλλάσσουν μεταξύ τους πληροφορίες [1, 17].
- Την πιθανότητα μίας επανάληψης ή της ανάγκης αναθεώρησης: Έχουν αναπτυχθεί μέθοδοι ανάλυσης με στόχο την ελαχιστοποίηση αυτής της πιθανότητας [10].
- Το βαθμό επίδρασης μιας πιθανής επανάληψης: Αυτό μπορεί να αποτυπωθεί ως το κλάσμα της εργασίας που θα χρειαστεί να επαναληφθεί [24]. Το μέτρο αυτό χρησιμοποιείται συνήθως σε συνδυασμό με το μέτρο της πιθανότητας της επανάληψης.

Ανάλυση Σχέσης	Τύπος Σχέσης	
Απαιτούμενη(+2)	Χωροταξική	Η γειτνίαση ...
	Ενεργειακή	Η μεταφορά ενέργειας ...
	Πληροφοριακή	Η ανταλλαγή πληροφορίας ...
	Υλική	Η ανταλλαγή υλικού ...
Επιθυμητή (+1)	Χωροταξική	Η γειτνίαση ...
	Ενεργειακή	Η μεταφορά ενέργειας ...
	Πληροφοριακή	Η ανταλλαγή πληροφορίας ...
	Υλική	Η ανταλλαγή υλικού ...
Αδιάφορη (0)	Χωροταξική	Η γειτνίαση ...
	Ενεργειακή	Η μεταφορά ενέργειας ...
	Πληροφοριακή	Η ανταλλαγή πληροφορίας ...
	Υλική	Η ανταλλαγή υλικού ...

είναι απαραίτητη για τη λειτουργικότητα

είναι ωφέλιμη αλλά δεν είναι εντελώς απαραίτητη για τη λειτουργικότητα

δεν επηρεάζει καθόλου τη λειτουργικότητα

Μη επιθυμητή (-1)	Χωροταξική	Η γειτνίαση ...	έχει αρνητική επίδραση, αλλά δεν εμποδίζει τη λειτουργικότητα
	Ενεργειακή	Η μεταφορά ενέργειας ...	
	Πληροφοριακή	Η ανταλλαγή πληροφορίας ...	
	Υλική	Η ανταλλαγή υλικού ...	
Ανεπιθύμητη (-2)	Χωροταξική	Η γειτνίαση ...	πρέπει να αποφευχθεί έτσι ώστε επιτευχθεί η λειτουργικότητα
	Ενεργειακή	Η μεταφορά ενέργειας ...	
	Πληροφοριακή	Η ανταλλαγή πληροφορίας ...	
	Υλική	Η ανταλλαγή υλικού ...	

Πίνακας 2: Ανάλυση σχέσεων σε αριθμητικούς ΠΔΣ συστατικών (17)

2.2.5 Συμπεράσματα για τον ΠΔΣ

Οι ΠΔΣ αποτελούν ένα ισχυρό εργαλείο στη μοντελοποίηση της διαδικασίας ανάπτυξης των προϊόντων. Η δυνατότητα αποτύπωσης της ροής της πληροφορίας, των συστατικών και των ομάδων ενός συστήματος, επιτρέπουν τον έλεγχο στη διαδικασία καθώς και στην οργάνωσή της. Διαφορετικοί τύποι ΠΔΣ μπορούν να χρησιμοποιηθούν για τις τρεις διαστάσεις της διαδικασίας ανάπτυξης προϊόντων [25]:

1. Άξονας του Προϊόντος: Αποσύνθεση και ενοποίηση των συστατικών του [12], [26].
2. Άξονας της Διαδικασίας: Αποτύπωση των σχέσεων και των απαιτήσεων πληροφορίας μεταξύ των δραστηριοτήτων, εντοπισμός των ανατροφοδοτήσεων και των επαναλήψεων [25], [27].
3. Άξονας της Οργάνωσης: Δημιουργία ομάδων ανάπτυξης [21] και αποτύπωση γνώσης [28].

Στη διαδικασία ανάπτυξης προϊόντων, ο ΠΔΣ έχει δύο εφαρμογές [2]:

1. Τον προγραμματισμό της σχεδίασης και τον εντοπισμό των βρόγχων και επαναλήψεων.
2. Την αποσύνθεση ή ομαδοποίηση των δραστηριοτήτων.

Ως εργαλείο ανάλυσης:

1. Αποτυπώνει με καθαρότητα και σαφήνεια τις συσχετίσεις και αλληλεπιδράσεις μεταξύ των στοιχείων (στοιχεία, υποσυστήματα, υποομάδες κ.λπ.) σε ένα σύνθετο πρόγραμμα ανάπτυξης προϊόντων.
2. Μπορεί να περιγράψει μια διαδικασία η οποία έχει επιμέρους στάδια.
3. Μπορεί να βοηθήσει στην εμφάνιση πληροφορικών για σύστημα που περιγράφεται που αποτελούν κομβικά σημεία.
4. Μπορεί να οδηγήσει σε αποτύπωση νέων προτύπων στην αρχιτεκτονική του προϊόντος και στην οργανωτική δομή.

Τέλος ως διαχειριστικό εργαλείο παρέχει μία εικόνα των συσχετίσεων, αναδράσεων και επαναλήψεων. Αναλυτικότερα:

1. Μπορεί να εξαχθούν διαγράμματα ροής πληροφοριών σε μεγάλα και σύνθετα προγράμματα.
2. Μπορεί να αποτελέσει ένα κοινό πλαίσιο συμφωνίας για την ομάδα εργασίας.
3. Βοηθά την ομάδα εργασίας να δει την πλήρη εικόνα του προγράμματος καθώς και τι υπολείπεται ακόμα.
4. Υποστηρίζει τον υπεύθυνο έργου παρέχοντάς του μια πλήρη εικόνα του έργου.

2.3 Μοντέλα βασιζόμενα στον ΠΔΣ

Μια από τις πιο σημαντικές οικογένειες στη μοντελοποίηση των προϊόντων είναι αυτή που βασίζεται στον Πίνακα Δομής Σχεδίασης [9]. Η μέθοδος ΠΔΣ βασίζεται στη λογική ότι κάθε δραστηριότητα στη διαδικασία σχεδιασμού προϊόντων μπορεί να αποτυπωθεί ως μια διεργασία η οποία χρησιμοποιεί και δημιουργεί πληροφορία. Η πληροφορία που παράγεται από μια δραστηριότητα μπορεί να γίνεται πληροφορία εισόδου για άλλες δραστηριότητες ή διεργασίες. Αυτή η σχέση στην πληροφορία, που λειτουργεί ως αγωγός μεταξύ των δραστηριοτήτων, μπορεί να περιέχει αναδράσεις βρόγχους ή επαναλήψεις. Παρ' όλα αυτά οι μέθοδοι που βασίζονται στον ΠΔΣ ανταποκρίνονται επαρκώς ακόμα και σε πολύπλοκες σχεδιαστικά διεργασίες [7].

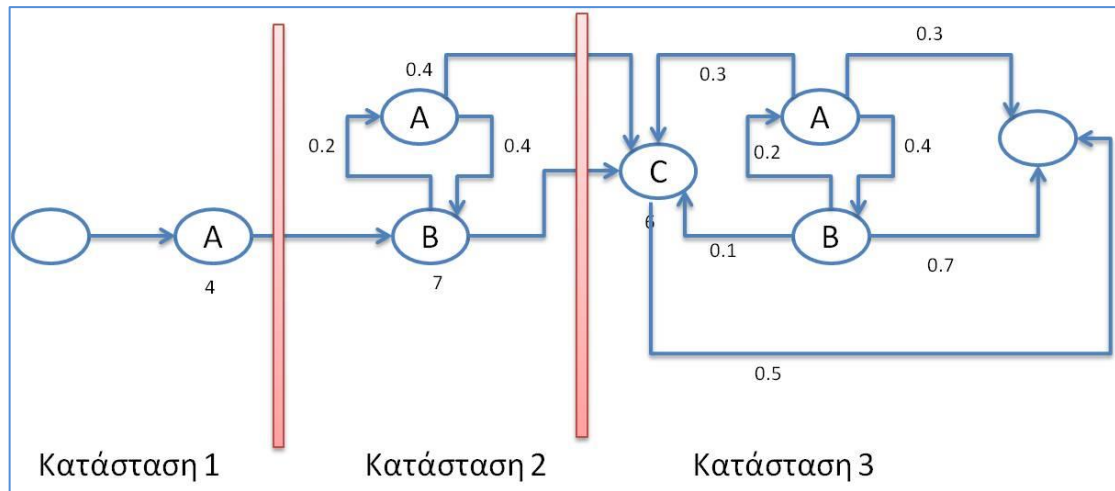
Στην ενότητα που ακολουθεί περιγράφονται μοντέλα που αποτελούν επεκτάσεις του ΠΔΣ με σειριακά χαρακτηριστικά δηλαδή μια δραστηριότητα ή διεργασία ανά μονάδα του χρόνου. Μοντέλα με παράλληλα χαρακτηριστικά παρουσιάζονται στην ενότητα 2.5.

2.3.1 Μοντέλα Ανατροφοδοτήσεων και διασταυρώσεων

Πρόκειται για δύο μοντέλα επεκτάσεις του ΠΔΣ [2]. Βασικός στόχος τους είναι η ελαχιστοποίηση των αναδράσεων και των διασταυρώσεων ή βρόγχων στην πληροφορία. Ο αριθμός των αναδράσεων και ο αριθμός των διασταυρώσεων αποτελούν κομβικό σημείο για τη διαδοχική σχεδίαση των δραστηριοτήτων. Οι ανατροφοδοτήσεις ή αναδράσεις εμπεριέχουν πιθανές επαναλήψεις σε διαδικασίες, ενώ οι διασταυρώσεις πληροφοριών μπορεί να προκαλέσουν αποσυντονισμό της διαδικασίας και κατά συνέπεια απώλεια πληροφοριών. Τέτοιου τύπου προβλήματα είναι διακριτά προβλήματα βελτιστοποίησης και έχει αποδειχθεί ότι οι γενετικοί αλγόριθμοι είναι αρκετά αποτελεσματικοί στην εύρεση πολύ καλών (near to optimal) λύσεων για προβλήματα μεγάλου μεγέθους [21].

2.3.2 Μοντέλο ακολουθιακών επαναλήψεων

Ένα άλλο σειριακό και μοντέλο βασιζόμενο στο ΠΔΣ είναι το μοντέλο ακολουθιακών επαναλήψεων [2], [29]. Σε αυτό το μοντέλο οι δραστηριότητες εκτελούνται μια ανά μονάδα χρόνου και κάθε μια δραστηριότητα έχει αιτιοκρατική διάρκεια με πιθανότητα επανάληψης (Εικόνα 3). Η αναμενόμενη διάρκεια οποιασδήποτε διαδοχής εκτέλεσης δραστηριοτήτων μπορεί να υπολογιστεί από μια αλυσίδα Markov. Η σειρά των δραστηριοτήτων μπορεί να αλλάξει με στόχο τη μείωση του αναμενόμενου χρόνου. Το συγκεκριμένο πρόβλημα ανήκει στα συνδυαστικά προβλήματα βελτιστοποίησης, που για αυτά υπάρχουν ευρετικοί αλγόριθμοι που προσεγγίζουν τη βέλτιστη λύση (near to optimal).



Εικόνα 3: Αλυσίδα Markov για το μοντέλο ακολουθιακών επαναλήψεων

Κάποια χρήσιμα αποτελέσματα που εξάγει το μοντέλο αυτό είναι ότι οι δραστηριότητες με μικρό χρόνο διάρκειας καλύτερα να ολοκληρώνονται πρώτες καθώς επίσης ότι είναι επιθυμητό να μειωθεί ο αριθμός των επαναλήψεων που εμφανίζονται ως ανατροφοδοτήσεις στη βέλτιστη διάταξη. Παρ' όλα αυτά, το μοντέλο για να εφαρμοστεί πρέπει να γίνουν μια σειρά από υποθέσεις, όπως για παράδειγμα ότι οι δραστηριότητες έχουν σταθερή διάρκεια ανεξάρτητα από το πόσες φορές επαναλαμβάνονται. Επίσης, γίνεται η υπόθεση ότι οι πιθανότητες επανάληψης είναι γνωστές, σταθερές και ανεξάρτητες από τη διάταξη των δραστηριοτήτων. Οι παραπάνω υποθέσεις δεν είναι πάντα εύκολο να θεωρηθούν σωστές ούτε να υπολογιστούν, πράγμα που κάνει το μοντέλο μη εφαρμόσιμο σε πολλές περιπτώσεις [29].

Κάποιοι από τους περιορισμούς που ήδη περιγράφηκαν αφαιρούνται σε μια επέκταση του μοντέλου η οποία επιτρέπει τυχαία διάρκεια στο χρόνο των δραστηριοτήτων, καθώς επίσης και παράλληλη επεξεργασία δραστηριοτήτων [30]. Οι περιορισμοί στο μοντέλο της επέκτασης μειώνουν την ευελιξία του για προβλήματα μεγάλου μεγέθους. Εφαρμόζοντας τη μέθοδο αυτή σε μεγάλα προβλήματα λαμβάνονται προσεγγιστικές λύσεις που όμως δεν είναι πάντα αξιόπιστες.

2.3.3 Μοντέλο εκτίμησης επαναληπτικών χρόνων

Μια επιπλέον επέκταση του ΠΔΣ είναι το μοντέλο που περιλαμβάνει πιθανολογικό OR ή XOR σχέσεις μεταξύ των δραστηριοτήτων. Για το μοντέλο αυτό, οι πιθανότητες της εκτέλεσης ενός ή περισσότερων από τα OR ή XOR είναι ανεξάρτητες από τον αριθμό των επαναλήψεων, και αυτές οι πιθανότητες καθορίζονται εξ αρχής. Στην τρέχουσα έκδοση χρόνιος κάθε δραστηριότητας είναι σταθερός και αιτιοκρατικά προσδιορισμένος και δεν αλλάζει καθώς η διαδικασία επανάληψης προχωράει [31].

Το μοντέλο ενσωματώνει ως μαθηματικό φορμαλισμό τον υπολογισμό της κατανομής πιθανότητας του συνολικού χρόνου που απαιτείται για να ολοκληρωθεί η διαδικασία σχεδίασης. Αυτό είναι χρήσιμο για την παρακολούθηση της προόδου, την πρόβλεψη των ημερομηνιών που συνήθως υπάρχουν περιορισμοί, καθώς και τη διανομή των πόρων στις δραστηριότητες. Οι δυσκολίες σε αυτό το μοντέλο είναι οι αρχικές υποθέσεις που πρέπει να γίνουν καθώς και η διαθεσιμότητα των δεδομένων. Επίσης, για την εφαρμογή του μοντέλου γίνεται η υπόθεση ότι ο χρόνος ολοκλήρωσης κάθε δραστηριότητας είναι γνωστός από την αρχή, καθώς επίσης και η δομή επανάληψης σε όσες δραστηριότητες υπάρχει. Τέλος, να

σημειωθεί ότι το μοντέλο δεν έχει χρησιμοποιηθεί ως παράδειγμα και επομένως δεν υπάρχουν εμπειρικά δεδομένα.

Όλα αυτά τα σειριακά και χρονικά μοντέλα προσπαθούν να διευθετήσουν προβλήματα διαχείρισης που υπάρχουν μέσα στο σχεδιασμό και την ανάπτυξη προϊόντων. Παρ' όλα αυτά, για την εφαρμογή τους απαιτούνται πολλές προϋποθέσεις με σκοπό να δώσουν αξιόπιστες λύσεις. Αν και η κατηγορία αυτών των μοντέλων είναι ενδιαφέρουσα, μένει να αποδειχθεί η χρησιμότητά τους σε επίπεδο βιομηχανίας.

2.3.4 Μοντέλα αποσύνθεσης μεγάλων συστημάτων

Αρκετά από τα μοντέλα σχεδιασμού και ανάπτυξης προϊόντων, στην προσπάθεια διαχείρισης του προβλήματος προτείνουν τον επιμερισμό μεγάλων συστημάτων σε μικρότερες δομές ή υποσυστήματα. Η απόφαση πως θα γίνει ο διαχωρισμός είναι και αυτή μέρος της μεθοδολογίας. Βασικοί στόχοι αυτών των μοντέλων είναι η κατανομή της εργασίας σε πολλές ομάδες σχεδίασης αλλά η ελαχιστοποίηση της επικάλυψης μεταξύ των υποσυστημάτων [32]. Η αποσύνθεση μπορεί επομένως:

- Να επιταχύνει τη διαδικασία σχεδίασης.
- Να δώσει καλύτερα αποτελέσματα στην τελική σχεδίαση [33].

Όλα τα μοντέλα αποσύνθεσης έχουν ως σημείο αναφορά το μοντέλο του Alexander [34]. Βασικό χαρακτηριστικό στο μοντέλο του Alexander είναι οι αλληλεπιδράσεις μεταξύ των στοιχείων σχεδίασης οι οποίες καθορίζουν τις υποομάδες στοιχείων. Αφού οι σχεδιαστές αποφασίσουν ποιες από τις αλληλεπιδράσεις θα θεωρήσουν σημαντικές και ποιες όχι, το μοντέλο, προτείνει τις αντίστοιχες ομάδες στοιχείων.

Η επιβεβαίωση της λειτουργίας του μοντέλου έρχεται από τους Lewis et. al.[35], που εφαρμόζουν το μοντέλο σε ένα πρόβλημα κατασκευής κτηρίου και στη συνέχεια συγκρίνουν τα αποτελέσματα που δίνει το μοντέλο, με τις λύσεις που προτείνουν οι σχεδιαστές του κτηρίου. Τα αποτελέσματα που πήραν ήταν παρόμοια κάτι που δείχνει ότι μοντέλο λειτουργεί σωστά και παρόμοια με τους σχεδιαστές.

2.3.5 Μέθοδος πίνακα μετασχηματισμού εργασίας

Μια άλλη μέθοδος που αποτελεί επέκταση του ΠΔΣ και ως στόχο έχει την αποσύνθεση των δραστηριοτήτων είναι η μέθοδος πίνακα μετασχηματισμού εργασίας - ΠΜΕ (Work Transformation Matrix - WTM) [36]. Σε αυτό το μοντέλο, τα στοιχεία του ΠΔΣ εκτός της διαγωνίου αντικαθίστανται με τιμές που δείχνουν εάν και κατά πόσο πρέπει να αναθεωρηθεί κάθε δραστηριότητα ενώ ο χρόνος για κάθε δραστηριότητα υπολογίζεται.

Τα στοιχεία του ΠΔΣ σε αυτό το μοντέλο, προσδιορίζουν ποσοτικά την αναθεώρηση που πρέπει να γίνει στην προηγούμενη επανάληψη. Για την εφαρμογή του μοντέλου χρειάζεται να γίνουν οι υποθέσεις ότι:

- Η διαδικασία σχεδίου είναι επαναληπτική.
- Η αναθεώρηση των δραστηριοτήτων έχει γραμμικά χαρακτηριστικά.

Η συνολική εργασία που ολοκληρώνεται κατά τη διάρκεια της διαδικασίας σχεδιασμού μπορεί να υπολογιστεί ελέγχοντας το άθροισμα της εργασίας που γίνεται κατά τη διάρκεια της κάθε επανάληψης. Το μοντέλο ως αποτέλεσμα δίνει τη συνολική εργασία που ολοκληρώθη-

κε. Η μέθοδος εφαρμόστηκε με επιτυχία στην εταιρία General Motors σε συστήματα πέδησης. Η αδυναμία σε αυτό το μοντέλο προκύπτει από τους περιορισμούς που πρέπει να τεθούν εξ' αρχής ως υποθέσεις για τη λειτουργία του.

2.3.6 Μοντέλο αποσύνθεσης δραστηριοτήτων σε δομικές μονάδες(modules)

Το μοντέλο αυτό χρησιμοποιεί την αποσύνθεση των δραστηριοτήτων βασιζόμενο στις ομοιότητες που παρουσιάζουν οι διαδικασίες σχεδίασης με τα κυψελοειδή κατασκευαστικά συστήματα [37-39] όπως φαίνεται στην (Εικόνα 4). Το μοντέλο υποθέτει ότι οι δραστηριότητες σχεδίασης είναι παρόμοιες με τις εργασίες κατασκευής που υποβάλλονται σε επεξεργασία, και τα μέλη της ομάδας ανάπτυξης (όπως οι μηχανικοί) είναι παρόμοιοι με τον εξοπλισμό που χρησιμοποιείται σε μια επεξεργασία. Το μοντέλο αποσυνθέτει το σύστημα σε δομικές μονάδες έτσι ώστε κάθε δομική μονάδα (κατασκευαστικό κύτταρο ή σχεδιαστικός πόρος) αποτελεί αποσύνθεση άλλων δομικών μονάδων.

Κάνοντας τον παραλληλισμό μεταξύ κυψελοειδών κατασκευών και δραστηριοτήτων σχεδίου εξαγονται χρήσιμα συμπεράσματα για την αποσύνθεση των δραστηριοτήτων. Για παράδειγμα, το κυψελοειδές πρόβλημα βελτιστοποίησης είναι γνωστό σαν NP-HARD και υπάρχουν ευρετικές μέθοδοι που παρέχουν αξιόπιστες λύσεις

		Δραστηριότητα										
Δομική Μονάδα		3	7	8	2	4	6	5	10	9	1	
	4	1		1						1		
	5	1	1	1						1		
	1				1	1	1					1
	6				1	1						1
	3				1					1		
	2							1	1	1		
	7									1	1	

Εικόνα 4: Πίνακας αποσύνθεσης δραστηριοτήτων

Το μοντέλο εφαρμόζεται σε ένα πρόβλημα σχεδιασμού ενός ελικοειδούς ελατηρίου πίεσης [40]. Φαίνεται ότι όταν τα δεδομένα σχεδίου είναι διαθέσιμα το μοντέλο μπορεί να καθορίσει αποσυνθέσεις.

2.3.7 Άλλα συστήματα αποσύνθεσης

Αυτή η κατηγορία μοντέλων βασίζεται στην προτεραιότητα εκτέλεσης δραστηριοτήτων στη διαδικασία σχεδιασμού και στους διαθέσιμους πόρους για κάθε δραστηριότητα. Ο συνδυασμός αυτών των πληροφοριών υποδεικνύει :

- Πώς μια διαδικασία σχεδίασης πρέπει να αποσυντεθεί.
- Ποιες αλληλεπιδράσεις μεταξύ των δραστηριοτήτων και των ανθρώπων είναι επιθυμητές.

Στα μοντέλα αυτά δεν υπάρχει τυποποίηση ούτε έχει γίνει αυστηρή ανάλυση . Υπάρχει η βασική ιδέα αλλά το μοντέλο δεν προσπαθεί να το κάνει αυτό με έναν επίσημο τρόπο [41].

Μια δεύτερη προσπάθεια που αποτελεί επέκταση στη μέθοδο ΠΔΣ προσπαθεί να επιμερίσει τις δραστηριότητες της διαδικασίας σχεδιασμού σε οργανωτικές μονάδες. Αυτή η μέθοδος επιχειρεί να εμπλουτίσει τα δεδομένα που έχει ο ΠΔΣ, περιλαμβάνοντας χωρικές και ενεργειακές πληροφορίες καθώς και τις υλικές αλληλεπιδράσεις μεταξύ των στοιχείων του σχεδίου. Αυτές οι αλληλεπιδράσεις χρησιμοποιούνται για να καθορίσουν τις ομάδες των στοιχείων σχεδίασης. Αν και το μοντέλο αυτό περιλαμβάνει ένα σημαντικό ποσό πληροφορίας της δομής των στοιχείων σχεδίου το τμήμα του αφορά την ομαδοποίηση δεν έχει ακόμα τυποποιηθεί.

2.4 Στοχαστικά μοντέλα χρονική ανοχής

Στα ακολουθιακά και χρονικά μοντέλα που παρουσιάστηκαν στην προηγούμενη ενότητα, δύο παράγοντες που είναι σημαντικοί στην ανάπτυξη προϊόντων δεν αποτυπώνονται ως δεδομένα. Η ανάγκη για επανάληψη δραστηριοτήτων αλλά και η αναμονή για κοινούς πόρους είναι βασικά συστατικά στη σχεδίαση και ανάπτυξη προϊόντων. Συνήθως, η περιορισμένη διαθεσιμότητα σε πόρους προκαλεί καθυστερήσεις στο συνολικό πρόγραμμα που αυτό έχει ως αποτέλεσμα τη συνολική χρονική καθυστέρηση για το τελικό προϊόν. Τα δύο μοντέλα που περιγράφονται παρακάτω προσπαθούν να διευθετήσουν τέτοια ζητήματα.

2.4.1 Μοντέλο Q-GERT

Το πρώτο μοντέλο είναι το μοντέλο Q-GERT [42] που αποτελεί επέκταση των μοντέλων PERT και CPM. Το Q-GERT διαφοροποιείται από το GERT επιτρέποντας καθυστερήσεις αναμονής. Το GERT (Graphical Evaluation and Review Technique) είναι μια τεχνική ανάλυσης δικτύων για διαχείριση έργων επιτρέποντας πιθανολογική εκτίμηση της διάρκειας μιας δραστηριότητας. Περιγράφηκε πρώτη φορά το 1966 από τον Pritsker [43]. Οι Taylor και Moore [42] έχουν εφαρμόσει το Q-GERT στην ανάπτυξη προϊόντων. Το μοντέλο περιλαμβάνει δρομολόγηση των δραστηριοτήτων με χρήση πιθανοτήτων, όπως επίσης και χρήση πιθανοτήτων για τις επαναλήψεις και διακλαδώσεις. Βασικό δομικό στοιχείο της μεθόδου Q-GERT είναι η ανάλυση με τη βοήθεια προσομοίωσης για να υπολογίσει τους αναμενόμενους χρόνους. Οι Taylor και Moore εφαρμόζουν τη μεθοδολογία Q-GERT σε ένα σύνολο αναπτυξιακών έργων μέσα σε μια κλωστοϋφαντουργία.

Αποτιμώντας το μοντέλο, είναι δύσκολο η πιθανολογική εκτίμηση να είναι ακριβής όπως επίσης και η αξιοπιστία της προσομοίωσης, μια και είναι δύσκολο πολλές φορές να καθοριστεί με σαφήνεια η σχέση μεταξύ των μεταβλητών – παραμέτρων. Κάθε αλλαγή στο δίκτυο που μοντελοποιεί τη μέθοδο Q-PERT απαιτεί νέα προσομοίωση η οποία πολλές φορές είναι σχεδόν μη υλοποιήσιμη για πολύπλοκα προβλήματα.

2.4.2 Μοντέλο δικτύου αναμονής

Σε αυτό το μοντέλο, η ανάπτυξη προϊόντων αντιμετωπίζεται ως ένα δίκτυο αναμονής και γίνεται χρήση των αναλυτικών αποτελεσμάτων σύμφωνα με την προσέγγιση Brown [44]. Τα αποτελέσματα του μοντέλου δείχνουν τη συνολική διαθέσιμη χωρητικότητα του δικτύου καθώς και δείκτες απόδοσης, διαθέσιμων πόρων κ.λ.π..

Για να μπορεί το μοντέλο αυτό να θεωρηθεί έγκυρο, θα πρέπει να ικανοποιούνται κάποιες υποθέσεις:

- Θα πρέπει να υπάρξει ένας επαρκής αριθμός παρόμοιων έργων έτσι ώστε να έχουν νόημα οι πιθανότητες που χρησιμοποιούνται και βέβαια, να υπάρχει διαθεσιμότητα στοιχείων κάτι που πολλές φορές δεν είναι εφικτό.
- Το δίκτυο αναμονής πρέπει να λειτουργεί για μια ικανοποιητική χρονική περίοδο.

- Πρέπει να γίνει η υπόθεση ότι δεν υπάρχει κανένας μετά-επίπεδος έλεγχος του δικτύου. Αυτό σημαίνει ότι αναδρομικοί έλεγχοι ή εργασίες που είναι εκτός χρόνου (υπερρωρίες για παράδειγμα) μπορούν να οδηγήσουν σε αστοχίες και όλα μαζί στη μη εφαρμογή του μοντέλου.

Οι συντάκτες αυτής της μελέτης δηλώνουν ότι

"διαπιστώσαμε πως η κατασκευή του ίδιου του μοντέλου, ήταν πιθανώς το πιο άμεσα χρήσιμο μέρος της έρευνάς μας. [44]"

2.5 Χρονικά μοντέλα ανασκόπησης

Αυτή η κατηγορία μοντέλων επικεντρώνεται σε προβλήματα που αφορούν το συγχρονισμό δραστηριοτήτων στην ανάπτυξη προϊόντων. Οι δραστηριότητες αυτές αφορούν αναθεωρήσεις σχεδίων και εισαγωγή προϊόντων στην αγορά.

2.5.1 Συγχρονισμός των αναθεωρήσεων σχεδίου

Το μοντέλο βασίζεται στην ιδέα του προσδιορισμού των χρόνων αναθεώρησης σχεδίων. Οι αναθεωρήσεις σχεδιασμού είναι σημαντικό και αναπόσπαστο τμήμα στην ανάπτυξη προϊόντων, διότι οι ομάδες που συμμετέχουν στο έργο ανταλλάσσουν ιδέες, συμβάλλοντας στη τελική διαμόρφωση του προϊόντος. Αν και οι αναθεωρήσεις σχεδίων αποτελούν αφορμή για ανταλλαγή ιδεών μεταξύ των ομάδων εργασίας είναι μη παραγωγικές λόγω του χρόνου προετοιμασίας ενώ αυξάνουν το συνολικό κόστος. Το μοντέλο υποθέτει ότι υπάρχουν δύο τύποι δραστηριοτήτων σχεδίου:

- Δραστηριότητες σχεδίου προϊόντων και
- Δραστηριότητες σχεδίου διαδικασίας.

Σύμφωνα με την περιγραφή του μοντέλου από τους Ha και Porteus [45] οι δραστηριότητες σχεδίου μπορεί να δημιουργήσουν πρόβλημα στις δραστηριότητες της διαδικασίας και αυτό, οι σχεδιαστές θα το αντιληφθούν στην επόμενη αναθεώρηση σχεδίου. Τέτοια προβλήματα αντιμετωπίζονται μέσω των επαναλήψεων. Με βάση αυτή τη θεώρηση, κάθε επανεξέταση σχεδίου, μεταφέρει την εργασία από τη σχεδίαση προϊόντων στη σχεδίαση δραστηριοτήτων. Ένας από τους στόχους του μοντέλου είναι να υπάρχουν πυκνές χρονικά αναθεωρήσεις σχεδίου ώστε τα προβλήματα να ανακαλύπτονται γρήγορα, μειώνοντας έτσι το χρόνο επανάληψης και κατά συνέπεια το συνολικό χρόνο του έργου. Το μοντέλο προσπαθεί να εξισορροπήσει μεταξύ των δύο αυτών παραγόντων (δραστηριότητες σχεδίου προϊόντων και διαδικασίας) ψάχνοντας πάντα τη βέλτιστη ισορροπία.

Το μοντέλο έχει εφαρμοστεί μόνο για απλές περιπτώσεις έργων με μόνο δύο τύπους σχεδιαστικών εργασιών. Γενικά το μοντέλο δεν εφαρμόζεται άμεσα. Υπάρχει δυσκολία στη συλλογή δεδομένων όπως επίσης και στον υπολογισμό τους για οποιαδήποτε συγκεκριμένη περίπτωση. Τέλος, οι αναθεωρήσεις είναι ο μόνος μηχανισμός στον οποίο οι πληροφορίες περνούν από μια ομάδα έργου στην άλλη και μπορούν να ανακαλυφθούν λάθη στη σχεδίαση.

2.5.2 Σχεδιασμός ενοτήτων

Πρόκειται για ένα μοντέλο που αναπτύχθηκε για να διευθετήσει έργα που αφορούν την ανάπτυξη λογισμικού. Συγκεκριμένα, το μοντέλο αντιμετωπίζει με επιτυχία προβλήματα συντονισμού σε επιμέρους ενότητες στην ανάπτυξη λογισμικού [46]. Οι υποθέσεις που κάνει το μοντέλο είναι:

- Ο αριθμός των προγραμματισμών για την ανάπτυξη λογισμικού επιλέγεται από τον υπεύθυνο του έργου.
- Ο αριθμός των ενοτήτων είναι σταθερός. Προβλήματα συντονισμού καταγράφονται κάθε φορά που ολοκληρώνεται κάποιες ενότητες.
- Ο αριθμός ενοτήτων είναι μια μεταβλητή απόφασης.
- Κατά τη διάρκεια και μετά τη διευθέτηση του συντονισμού, οι ομάδες εργασίας διασταυρώνουν τα αποτελέσματά τους ενώ οι ολοκληρωμένες ενότητες ενσωματώνονται στο τελικό σχέδιο.
- Αναθεωρήσεις ενοτήτων υπάρχουν μόνο όταν διαπιστωθούν ασυμβατότητες μεταξύ των ενοτήτων.
- Τέλος, ο χρόνος ανάπτυξης είναι σταθερός και εκ' των προτέρων καθορισμένος.

Όπως και σε πολλά από τα μοντέλα που έχουν ήδη αναφερθεί, η δυσκολία συλλογής ειδικευμένων δεδομένων για κάθε συγκεκριμένο έργο θέτει περιορισμό στην εφαρμογή του μοντέλου.

2.5.3 Μοντέλο έκδοσης προϊόντος

Πρόκειται για ένα μοντέλο που εξετάζει τη λειτουργική σχέση μεταξύ της απόδοσης και του κέρδους ως μια συνάρτηση του χρόνου έκδοσης του προϊόντος στην αγορά. Το μοντέλο κάνει την υπόθεση ότι η απόδοση των προϊόντων είναι συνάρτηση του χρόνου ανάπτυξης. Η απόδοση ενός προϊόντος στην αγορά είναι συνάρτηση πολλών μεταβλητών όπως για παράδειγμα, του χρόνου εισαγωγής, ποια ανταγωνιστικά προϊόντα είναι ήδη στην αγορά ή θα βγουν παράλληλα κ.λ.π.. Οι υπεύθυνοι σε κάθε εταιρία που επιφορτίζονται με αυτό το έργο, θα πρέπει να πάρουν αποφάσεις όπως [46]:

- Πότε ένα προϊόν θα εισαχθεί στην αγορά έτσι ώστε να έχει υψηλή απόδοση ικανοποιώντας όμως και τις προσδοκίες των πελατών.
- Πόσο θα πρέπει να είναι ο κύκλος ανάπτυξης του προϊόντος έτσι ώστε να μην καλυφθεί η αγορά από προϊόντα ανταγωνιστών.

Το μοντέλο είναι σε θέση να υποδείξει τους βέλτιστους χρόνους απόδοσης στην αγορά καθώς επίσης και άλλα χαρακτηριστικά ενώ υποθέτει ότι η απόδοση προϊόντων είναι μια συνάρτηση της μορφής Cobb-Douglas (Στατιστική εξίσωση που εκφράζει τη σχέση μεταξύ εργασίας και κεφαλαίων που χρησιμοποιούνται και της ποσότητας παραγωγής που επιτυγχάνεται από αυτά.)

Αν και το μοντέλο είναι μαθηματικά τεκμηριωμένο εμφανίζει προβλήματα στη διαμόρφωση διαδικασίας ανάπτυξης προϊόντων μια και τα δεδομένα δεν μπορούν να ελεγχθούν πάντα.

2.5.4 Σταδιακή στοχοθέτηση

Πρόκειται για ένα μοντέλο που μελετά τον τρόπο που οι εταιρίες διαθέτουν τους πόρους ανάπτυξης στα στάδια σχεδιασμού και ανάπτυξης ενός προϊόντος [47]. Η υπόθεση πάνω στην οποία βασίζεται το μοντέλο, είναι ο περιορισμός των πόρων κάθε εταιρίας σε ότι αφορά την ανάπτυξη προϊόντων (Για παράδειγμα, ανθρώπινο δυναμικό, υπηρεσίες κ.λ.π.).

Η διαχείριση των πόρων αποτελεί κομβικό σημείο στην ανάπτυξη προϊόντων, μια και η στρατηγική που ακολουθεί μια εταιρία στη διαχείρισή τους, δίνει το πλεονέκτημα ή όχι ένα προϊόν να είναι πρώτο στην αγορά. Κάθε εταιρία αποφασίζει πως θα χρησιμοποιεί τους πόρους που διαθέτει στα δύο στάδια της ανάπτυξης προϊόντων (έρευνα και ανάπτυξη). Το μοντέλο, με βάση τα αποτελέσματα που δίνει, δείχνει να είναι σε πλεονεκτική θέση εταιρίες που φτάνουν πρώτες στην αγορά, σε αντίθεση με εταιρίες που αυξάνουν τις τεχνικές αποδόσεις κατα-

σκευάζοντας καλύτερο προϊόν. Το μοντέλο, επιβεβαιώνεται εμπειρικά από τα αποτελέσματα εταιριών που το χρησιμοποίησαν στη βιομηχανία των κεντρικών υπολογιστών προς το τέλος της δεκαετίας του '80.

Τέλος, να σημειωθεί ότι το μοντέλο χρησιμοποιεί βασικές έννοιες της θεωρίας παιγνίων. Η μοντελοποίηση με βάση τη θεωρία παιγνίων είναι ερευνητικά ενδιαφέρουσα μια και το πρόβλημα ανάπτυξης προϊόντων εμπεριέχει αλληλεπίδραση μεταξύ των εταιριών, τόσο σε κοινούς πόρους όσο και σε αποφάσεις που επηρεάζονται άμεσα από ανταγωνιστικές ενέργειες.

2.6 Μοντέλα παραλληλισμού

Από τα βασικότερα προβλήματα στη σχεδίαση και ανάπτυξη προϊόντων είναι δυνατότητα εκτέλεσης παράλληλων δραστηριοτήτων. Μελέτες έχουν δείξει, ότι η παράλληλη επεξεργασία δραστηριοτήτων έχει συγκριτικά πλεονεκτήματα από μια σειριακή αρχιτεκτονική [48]. Βασικό πλεονέκτημα είναι προφανώς ο χρόνος ολοκλήρωσης του έργου που αποτελεί στη βιομηχανία καθοριστικό παράγοντα. Τα μοντέλα που περιγράφονται σε αυτή την ενότητα δείχνουν ότι αν και ο παραλληλισμός στην επεξεργασία δραστηριοτήτων είναι επιθυμητός, πολλές φορές αυξάνει την πολυπλοκότητα διαχείρισης και ολοκλήρωσης του έργου. Έτσι, ένα ποσοστό σειριακής εκτέλεσης στη ολοκλήρωση κάποιων δραστηριοτήτων πολλές φορές είναι απαραίτητο.

2.6.1 Παράλληλος σχεδιασμός

Το μοντέλο αυτό μελετά τα αποτελέσματα της παράλληλης επεξεργασίας δραστηριοτήτων. Υπάρχουν κάποιες υποθέσεις που λειτουργούν ως αρχικές συνθήκες για το μοντέλο. Αυτές είναι[49]:

- Υπάρχει ένα σύνολο από δραστηριότητες που πρέπει να ολοκληρωθούν που είναι από την αρχή γνωστές στο πλήθος τους.
- Υπάρχει μια καθορισμένη σειρά στη διαδικασία σχεδιασμού.
- Εάν μια δραστηριότητα δεν ολοκληρωθεί με επιτυχία πρέπει να επαναληφθεί μαζί με όλες τις δραστηριότητες που την ακολουθούσαν.

Βασικό πλεονέκτημα των παράλληλων δραστηριοτήτων είναι ότι το συνολικό έργο σχεδίασης και ανάπτυξης προϊόντος μπορεί να ολοκληρωθεί γρηγορότερα. Το μοντέλο ουσιαστικά μελετά και προσπαθεί να καθορίσει ποσοτικά τον παραλληλισμό έτσι ώστε η βελτίωση του συνολικού χρόνου να δικαιολογεί τις πιθανές αυξήσεις στο κόστος ανάπτυξης. Η αξιολόγηση μεγάλου αριθμού παραμέτρων που σχετίζονται με το χρόνο και το κόστος καθιστά το μοντέλο μη εφαρμόσιμο σε πραγματικές συνθήκες.

2.6.2 Παράλληλο / Σειριακό Μοντέλο ΠΜΕ

Ένα άλλο μοντέλο που προσπαθεί να προσδιορίσει ποσοτικά τον παραλληλισμό δραστηριοτήτων στην ανάπτυξη προϊόντων είναι η παράλληλη – σειριακή έκδοση του WTM (Work transformation Matrix ή Πίνακας Μετασχηματισμού Εργασίας – ΠΜΕ) [50]. Οι βασικές αλλαγές από το αρχικό ΠΜΕ είναι:

- Δεν υπάρχει καμία αλλαγή στις παραμέτρους μοντελοποίησης.
- Οι δραστηριότητες ομαδοποιούνται .
- Επιτρέπεται η σειριακή επεξεργασία των επαναλαμβανόμενων και αλληλεξαρτημένων δραστηριοτήτων.
- Η επεξεργασία των δραστηριοτήτων που ορίστηκαν εξ' αρχής συνεχίζεται όσο υπάρχουν αναθεωρήσεις.

- Το μοντέλο στη συνέχεια, παίρνει το επόμενο σύνολο δραστηριοτήτων κάνοντας επαναλήψεις και αναθεωρήσεις.
- Ο συνολικός χρόνος ανάπτυξης και ο φόρτος εργασίας μπορεί να υπολογιστεί βάση του ΠΜΕ μοντέλου.

Όλες οι παραπάνω υποθέσεις είναι αρκετά δεσμευτικές σε πρακτικό επίπεδο και ενδεχομένως μη υλοποιήσιμες. Το μοντέλο έχει χρησιμοποιηθεί σε ένα έργο σχεδιασμού τερματικών σταθμών υπολογιστών. Τα δεδομένα που είναι απαραίτητα για την υλοποίησή του φαίνεται ότι είναι εύκολα διαθέσιμα και τα αποτελέσματα του μοντέλου αξιόπιστα.

2.6.3 Μοντέλο σταδιακής επικάλυψης

Υπάρχουν δύο μοντέλα που μελετούν τη διαδικασία της επικάλυψης δραστηριοτήτων [50]. Αυξάνοντας την επικάλυψη δραστηριοτήτων μειώνεται ο χρόνος ολοκλήρωσης του έργου όπως συνάγεται από την εικόνα 5. Η επικάλυψη μπορεί να οδηγήσει σε επιλογή εναλλακτικών λύσεων στο έργο από πρώιμα στάδια που πολλές φορές όμως δημιουργεί απώλεια ποιοτικής πληροφορίας και πρόσθετες επαναλήψεις. Τέλος, επικαλύψεις μπορούν να προκύψουν όταν παράμετροι στο σχέδιο και την ανάπτυξη δεν αποφασίζονται από νωρίς αλλά αναβάλλονται σε επόμενα στάδια.

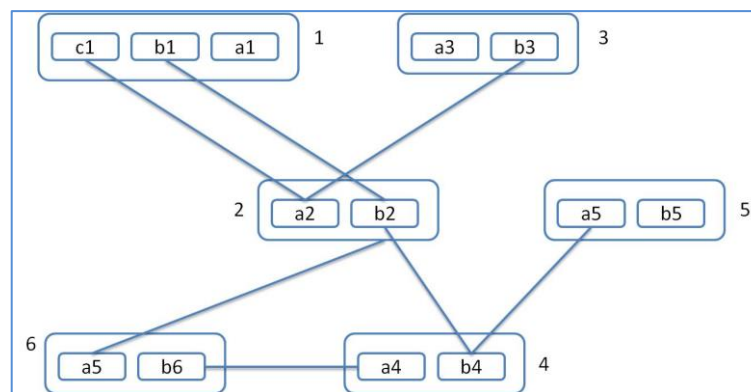
Τα μοντέλα αυτά προτείνουν πολιτικές επικάλυψης δραστηριοτήτων προσπαθώντας να ελαχιστοποιήσουν τους χρόνους στη διαδικασία σχεδιασμού για δεδομένα που μπορούν να προσδιοριστούν. Παράλληλα, προσδιορίζουν τις παραμέτρους οι οποίες σχετίζονται με το χρόνο ανάπτυξης, εξισορροπώντας με τη ποιότητα των προϊόντων. Τα μοντέλα είναι τεκμηριωμένα με παραδείγματα από τις αυτοκινητοβιομηχανίες και τις βιομηχανίες ηλεκτρονικών ειδών.



Εικόνα 5: Το μοντέλο σταδιακής επικάλυψης

2.7 Μοντέλα σχεδίασης που δεν καλύπτουν οι παραπάνω κατηγορίες.

Κάποια μοντέλα δεν εστιάζουν στο χρόνο ολοκλήρωσης ενός έργου ως μια παράμετρο σημαντική στην ανάπτυξη προϊόντων. Αντί αυτού, εστιάζουν σε άλλες πτυχές της σχεδίασης και ανάπτυξης. Ένα τέτοιο μοντέλο είναι γνωστό ως AIDA (Awareness – Interest – Desire – Action, Ανάλυση των Ενδοσυνδεδεμένων Περιοχών Απόφασης) μοντέλο [51]. Το AIDA υποθέτει ότι κάθε πρόβλημα σχεδίασης έχει επιμέρους υποπροβλήματα και κάθε ένα από αυτά πολλές πιθανές λύσεις (Εικόνα 6). Το AIDA στη συνέχεια προσπαθεί να βρει ποιες υπολύσεις είναι μη συμβατές με τις υπολύσεις άλλων υποπροβλημάτων. Αποκλείοντας αυτές τις λύσεις το μοντέλο οδηγείται σε εφικτά σύνολα λύσεων που δεν έχουν ασυμβατότητες μεταξύ τους [52].



Εικόνα 6: Το μοντέλο AIDA (51)

Ο φορμαλισμός του AIDA περιγράφει ουσιαστικά ένα χρήσιμο σύνολο από κανόνες που εφαρμόζονται στη διαδικασία σχεδίασης και ανάπτυξης. Για παράδειγμα, αποφάσεις στο σχεδιασμό μικρών εξαρτημάτων είναι πολλές φορές αλληλένδετες με το γενικότερο σχεδιασμό. Υπάρχουν και άλλοι φορμαλισμοί που μελετούν πως η σειρά των αποφάσεων αλλάζει τον τελικό σχεδιασμό της διαδικασίας. Για παράδειγμα, κάθε σχέδιο αποτελείται από ένα αριθμό αποφάσεων, και κάθε απόφαση αλλάζει το διάστημα απόφασης που είναι διαθέσιμο για τις μελλοντικές αποφάσεις. Αν και τα μοντέλα αυτά φαίνονται θεωρητικά να ανταποκρίνονται στις προσδοκίες τους, η βιομηχανία δεν τα έχει χρησιμοποιήσει.

2.8. Μοντέλα Συναρμολόγησης

2.8.1. Εισαγωγή

Τα περισσότερα προϊόντα σήμερα αποτελούν συναρμολογήσεις επιμέρους εξαρτημάτων τα οποία μπορεί ανάλογα με το προϊόν αριθμούν από ελάχιστα μέχρι μερικές δεκάδες χιλιάδες. Ως εκ τούτου, η συναρμολόγηση αποτελεί μια δύσκολη χρονοβόρα και με υψηλό κόστος διαδικασία, αν συνυπολογιστούν οι χρονικές πιέσεις και οι περιορισμοί της σύγχρονης αγοράς. Ειδικότερα, σε ότι αφορά τα ηλεκτρομηχανολογικά προϊόντα στα οποία εστιάζει η παρούσα εργασία, ο σχεδιασμός τους θα πρέπει να συμπεριλάβει δεδομένα που αφορούν τις σχέσεις των εξαρτημάτων μεταξύ τους, δεδομένα για τη σχετική κίνηση εξαρτημάτων ή ομάδας αυτών, στοιχεία ανοχών, σύνδεσης, κ.λπ. [53].

Αν και υπάρχουν πολλά προγράμματα σχεδίασης με χρήση υπολογιστή (CAD), κάθε ένα από αυτά προσεγγίζει με διαφορετικό τρόπο τη μοντελοποίηση προϊόντων καθώς και τις σχεδιαστικές πληροφορίες όπως τις ανοχές, τους περιορισμούς, τις πληροφορίες της κινηματικής κ.λπ.. Αυτή η ποικιλομορφία στην αντιμετώπιση της σχεδίασης και της συναρμολόγησης κάνει δυσδιάκριτη τη δυνατότητα αποτύπωσης με κοινό τρόπο της πληροφορίας που απαιτείται στο ίδιο επίπεδο σχεδίασης. Για παράδειγμα, στα πρώιμα στάδια σχεδίασης υπάρχουν άλλα ποσοτικά και ποιοτικά χαρακτηριστικά στην πληροφορία σε σχέση με τα μεταγενέστερα στάδια.

Κατά τη φάση του αρχικού σχεδιασμού, οι σχεδιαστές μπορούν να αξιολογήσουν τις κινηματικές ικανότητες ενός σχεδίου, ωστόσο το μεγαλύτερο μέρος των γεωμετρικών στοιχείων δεν είναι ακόμα προσδιορισμένο. Παράλληλα, πρέπει να καθοριστεί η σχετική θέση και ο προσανατολισμός των μεγάλων εξαρτημάτων (ή υποσυναρμολογήσεων) εντός του προϊόντος, λαμβάνοντας υπόψη τις αλληλεπιδράσεις μεταξύ τους. Στη συνέχεια κατά τη διάρκεια της φάσης λεπτομερούς σχεδιασμού, αυτά τα εξαρτήματα ή συγκροτήματα περαιτέρω αποσυντίθεται για να εξαχθούν τα γεωμετρικά στοιχεία των μικρότερων συναρμολογημένων εξαρτημάτων. Από

τα παραπάνω είναι σαφές ότι τα θέματα που σχετίζονται με τη συναρμολόγηση αποτελούν ένα σημαντικό μέρος του σχεδιασμού του. Σύμφωνα με τον Whitney [54], η συναρμολόγηση μπορεί να αποτελέσει τη βάση για την ενσωμάτωση όλων των δραστηριοτήτων σχεδιασμού και μοντελοποίησης του προϊόντος και του κύκλου ζωής του.

2.8.2. Επισκόπηση μοντέλων συναρμολόγησης

Μέχρι σήμερα δεν υπάρχει μια ολοκληρωμένη προσπάθεια προτυποποίησης της διαδικασίας αυτής η οποία να κάνει διακριτή την πληροφορία που απαιτείται σε κάθε στάδιο της σχεδίασης καθώς και τις σχέσεις μεταξύ των στοιχείων συναρμολόγησης. Μια πρώτη προσπάθεια προτυποποίησης έγινε από το Διεθνή Οργανισμό Προτυποποίησης (International Organization for Standardization - ISO) ο οποίος με το πρότυπο ISO-10303 [55] και πιο συγκεκριμένα από την ομάδα εργασίας ISO TC 184/SC4/WG12 που δίνει μια πρώτη περιορισμένη προσέγγιση της αναπαράστασης δεδομένων στη συναρμολόγηση. Στην ίδια λογική κινείται και η Εθνική Ιαπωνική Επιτροπή (Japanese National Committee - JNC) η οποία πάνω στο πρότυπο ISO-10303 κάνει τη δική της πρόταση για συναρμολόγηση προϊόντων [56]. Εκτενής αναφορά για το παραπάνω πρότυπα γίνεται στην επόμενη ενότητα.

Μια άλλη προσπάθεια έγινε από το χρηματοδοτούμενο έργο ESPRIT που είναι πλέον γνωστό ως MOKA (Methodology and tools Oriented to Knowledge-Based Engineering Applications) [57]. Σε αυτήν την προσπάθεια το μοντέλο προϊόντος υποστηρίζει πέντε διαφορετικούς διακριτούς τρόπους αναπαράστασης του προϊόντος. Συγκεκριμένα ως δομή, λειτουργία, συμπεριφορά, τεχνολογία και αναπαράσταση. Στη συγκεκριμένη μοντελοποίηση παρ' όλα αυτά δεν υποστηρίζονται πληροφορίες που αφορούν την κινηματική, τις ανοχές και τους περιορισμούς που προκύπτουν από τη συναρμολόγηση.

Οι Whitney και Mantripragada στην εργασία τους προτείνουν ένα μοντέλο που προσπαθεί να δώσει λύση στη συναρμολόγηση ξεκινώντας από τη θεώρηση ότι τα περισσότερα προβλήματα προκύπτουν από την κακή επιλογή της σειράς διαδικασιών συναρμολόγησης [58, 59]. Προτείνουν ένα μοντέλο γνωστό και ως Datum Flow Chain (DFC) το οποίο προσπαθεί να ορίσει πρώτα τις επιμέρους διαδικασίες – σειρές συναρμολόγησης και στη συνέχεια να ορίσουν τα επιμέρους συστατικά της συναρμολόγησης. Στην ουσία πρόκειται για ένα μοντέλο που αποτυπώνει τις πληροφορίες συναρμολόγησης στα τελευταία στάδιά της καθώς και τα βασικά χαρακτηριστικά της.

Από τις πρώτες εργασίες για μοντελοποίηση της συναρμολόγησης ήταν η εργασία των Lee, και Gossard στην οποία οι δομές δεδομένων που αναπαριστούν τη συναρμολόγηση χωρίζονται σε δύο μέρη [60]. Στο πρώτο μέρος χρησιμοποιείται μια δομή δεδομένων για την αποτύπωση τοπολογικών και γεωμετρικών στοιχείων κάθε συστατικού μέσα στη συναρμολόγηση. Στο δεύτερο μέρος χρησιμοποιείται μια δομή δεδομένων η οποία αποθηκεύει πληροφορίες για τον τρόπο με τον οποίο τα επιμέρους συστατικά μιας συναρμολόγησης συνδέονται μεταξύ τους.

Ο Van der Net στη διδακτορική του διατριβή εστιάζει στη μοντελοποίηση της συναρμολόγησης και στα προβλήματα που προκύπτουν αυξάνοντας το χρόνο και τις επαναλήψεις στη σχεδίαση [61]. Το πλάνο για τη συναρμολόγηση στην εργασία του λαμβάνει υπόψη τις σχέσεις μεταξύ των συναρμολογούμενων στοιχείων, τις ανοχές και τα γεωμετρικά χαρακτηριστικά.

Οι Noort, et. al. στην εργασία τους αποδεικνύουν ότι ενώ τα μοντέλα συναρμολόγησης υποστηρίζουν είτε τη μοντελοποίηση των βασικών δομικών στοιχείων ενός προϊόντος είτε τη μο-

ντελοποίηση των υποσυναρμολογήσεων, ένα ιδεατό μοντέλο θα πρέπει να κάνει και τα δύο [62]. Προτείνουν ένα μοντέλο το οποίο αποτυπώνει αυτήν την ολοκλήρωση για τις διαφορετικές φάσεις εξέλιξης στο σχεδιασμό. Η εργασία επικεντρώνεται στη λογική υποστήριξης των τμημάτων και των υποσυναρμολογήσεων ολόκληρου του προϊόντος. Τέλος, περιγράφουν τα χαρακτηριστικά και τα εργαλεία που πρέπει να χρησιμοποιηθούν για τη μοντελοποίηση.

Μια άλλη παράλληλη προσπάθεια έγινε από το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (NIST) αναπτύσσοντας το μοντέλο ανοικτής συναρμολόγησης (Open Assembly Model – OAM/NIST) το οποίο ουσιαστικά αποτελεί μια επέκταση του CPM/NIST [63]. Βασικός στόχος του είναι η αναπαράσταση πληροφοριών και τα πρωτόκολλα ανταλλαγής δεδομένων για τη συναρμολόγηση καθώς επίσης και για τις ανοχές των μερών ενός προϊόντος. Όπως και το CPM, έτσι και το OAM υλοποιείται κάνοντας χρήση κλάσεων και σχέσεων αποτυπώνοντας την πληροφορία που απαιτείται για τα επιμέρους στοιχεία μιας συναρμολόγησης – σύνθεσης, καθώς επίσης και τις σχέσεις που έχουν μεταξύ τους. Το μοντέλο χρησιμοποιεί τις δομές δεδομένων του ISO 10030/AP214 γνωστό και ως Πρότυπο για την ανταλλαγή δεδομένων στη μοντελοποίηση προϊόντος (Standard for The Exchange of Product model data – STEP) για να αποτυπώσει τη συναρμολόγηση και ως εννοιολογικό μοντέλο αλλά και ως δομή δεδομένων [64].

2.9. Αλγόριθμοι Ομαδοποίησης

2.9.1. Εισαγωγή

Ομαδοποίηση είναι η κατηγοριοποίηση δεδομένων, αντικειμένων σε υποομάδες με παρόμοια χαρακτηριστικά τα οποία έχουν νόημα για το συγκεκριμένο πρόβλημα. Μια ομάδα λοιπόν αποτελείται από έναν αριθμό παρόμοιων αντικειμένων τα οποία συγκεντρώνονται και ομαδοποιούνται. Η ομαδοποίηση αποτελεί εδώ και πολλά χρόνια πεδίο έρευνας για πολλά επιστημονικά πεδία [65]. Συγκεκριμένα, η ομαδοποίηση χρησιμοποιείται στη στατιστική [66] και τις θετικές επιστήμες [67]. Η πρώτη προσέγγιση και χρήση αλγόριθμων ομαδοποίησης για την αναγνώριση προτύπων έγινε το 1973 από τους Duda και Hard [68]. Από τις πιο κλασικές εφαρμογές που αφορούν αναγνώριση προτύπων είναι η αναγνώριση φωνής και χαρακτήρων.

Επόμενος σημαντικός σταθμός στην ανάπτυξη αλγορίθμων ομαδοποίησης είναι οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν για προβλήματα κατάμησης εικόνας και μηχανικής όρασης [69]. Η ομαδοποίηση μπορεί να ειπωθεί ως πρόβλημα εκτίμησης της πυκνότητας δεδομένων ή αντικειμένων γενικότερα. Αυτό είναι το αντικείμενο της «πολλών – μεταβλητών» στατιστικής ανάλυσης [70]. Η ομαδοποίηση επίσης χρησιμοποιείται ευρέως στη συμπίεση εικόνας στον τομέα της επεξεργασίας εικόνας με στόχο να δώσει εικόνες μικρότερου μεγέθους με τη μικρότερη απώλεια πληροφορίας [71]. Τέλος, μια σειρά από σύγχρονες εφαρμογές όπως: Γεωγραφικά συστήματα πληροφοριών [72], Εφαρμογές του Διαδικτύου [73], Ανάλυση DNA στον υπολογιστή [74] κ.λπ., έχουν ως βάση αλγόριθμους ομαδοποίησης.

Ένας αλγόριθμος ομαδοποίησης επιμερίζει τα αντικείμενα σε ένα αριθμό από ομάδες, υποσύνολα ή κατηγορίες. Μέχρι σήμερα δεν υπάρχει ένας καθολικός ορισμός για την έννοια της ομαδοποίησης [75]. Συνήθως η έννοια της ομαδοποίησης είναι συνυφασμένη με την εσωτερική ομοιότητα κάποιων χαρακτηριστικών των αντικειμένων που ανήκουν σε μία ομάδα και ταυτόχρονα με την ανομοιότητα των αντικειμένων που ανήκουν σε διαφορετική ομάδα [76].

2.9.2. Κατηγοριοποίηση Αλγορίθμων

Οι αλγόριθμοι ομαδοποίησης μπορούν να κατηγοριοποιηθούν στις παρακάτω κατηγορίες [77]:

- Την ιεραρχική ομαδοποίηση η οποία προσπαθεί να κατασκευάσει μια δομή των αντικειμένων που να μοιάζει με τη δομή ενός δένδρου ή ιεραρχίας. Η προσέγγιση μπορεί να είναι είτε “bottom up” που σημαίνει ότι κάθε στοιχείο θεωρείται μια ομάδα και στη συνέχεια ζεύγη ομάδων συγχωνεύονται φτιάχνοντας την ιεραρχία, είτε “top down” στην οποία μια ομάδα επιμερίζεται σε περισσότερες. Οι ιεραρχικοί αλγόριθμοι ομαδοποίησης επιπλέον κατηγοριοποιούνται σε:
 - i. Αλγόριθμους συσσωμάτων (agglomerative, όπως είναι οι αλγόριθμοι single linkage, media linkage, complete linkage κ.λπ.).
 - ii. Διαιρετικούς αλγόριθμους, όπως οι divisive analysis (DIANA), monothetic analysis (MONA) κ.λπ.).
- Την ομαδοποίηση επιμερισμού στην οποία γίνεται προσπάθεια να βρεθούν ομάδες αντικειμένων με κοινά χαρακτηριστικά, έτσι ώστε κάθε αντικείμενο να ανήκει σε μια μόνο ομάδα. Από τους πιο γνωστούς επιμεριστικούς αλγόριθμους είναι ο k-means αλγόριθμος [78].
- Αλγόριθμοι απόστασης και ομοιότητας οι οποίοι χρησιμοποιούν κάποια γνωστά μέτρα όπως την Ευκλείδειο απόσταση [79] μεταξύ δύο σημείων ή μέτρα συσχέτισης, προσπαθώντας να αποδώσουν διακριτά χαρακτηριστικά στα δεδομένα του προβλήματος για να αποφασισθεί σε ποια ομάδα ανήκουν.
- Αλγόριθμοι βασιζόμενοι στο τετραγωνικό σφάλμα, όπως οι K-means, iterative self-organizing data analysis technique (ISODATA), genetic – means algorithm (GKA) και partitioning around methods (PAM).
- Αλγόριθμοι εκτίμησης πυκνότητας δεδομένων, όπως Gaussian mixture density decomposition (GMDD) και Auto Class.
- Αλγόριθμοι βασιζόμενοι στη θεωρία γράφων, όπως Chameleon, Delaunay triangulation graph (DTG), highly connected subgraphs (HCS), clustering identification via connectivity kernels (CLICK) και cluster affinity search technique (CAST).
- Αλγόριθμοι βασιζόμενοι σε συνδυαστικές τεχνικές, όπως οι γενετικοί αλγόριθμοι (GGA), TS clustering και SA clustering.
- Αλγόριθμοι ασαφούς λογικής, όπως οι Fuzzy c-means (FCM), mountain method (MM), possibilistic c-means clustering algorithm (PCM) και fuzzy c-shells (FCS).
- Αλγόριθμοι Νευρωνικών δικτύων, όπως οι Learning vector quantization (LVQ), self-organizing feature map (SOFM), ART, simplified ART (SART), hyper ellipsoidal clustering network (HEC) και self-splitting competitive learning network (SPLL).
- Αλγόριθμοι βασιζόμενοι στον πυρήνα κάθε ομάδας (kernel based), όπως οι Kernel – means και support vector clustering (SVC).
- Αλγόριθμοι ακολουθιακών δεδομένων, όπως οι Sequence Similarity, indirect sequence clustering, Statistical sequence clustering.
- Αλγόριθμοι μεγάλης κλίμακας δεδομένων όπως CLARA, CURE, CLARANS, BIRCH, DBSCAN, DENCLUE, Wave Cluster, FC και ART.
- Αλγόριθμοι οπτικοποίησης δεδομένων και πολλών διαστάσεων, όπως PCA, ICA, Projection pursuit, Isomap, LLE, CLIQUE, OptiGrid και ORCLUS.
- Αλγόριθμοι στους οποίους ο χρήστης προσδιορίζει των αριθμό των ομάδων.

Τέλος οι αλγόριθμοι ομαδοποίησης μπορούν να κατηγοριοποιηθούν σε αλγόριθμους ισχυρής ομαδοποίησης όπου κάθε στοιχείο ανήκει αποκλειστικά σε μια μόνο ομάδα και σε αλγόριθ-

μους ασαφούς ομαδοποίησης, όπου ένα στοιχείο μπορεί να ανήκει σε περισσότερες από μία ομάδες, έχοντας ένα βαθμό μέλους για την κάθε ομάδα. Γενικά, η ανάλυση αντικειμένων σε ομάδες περιλαμβάνει τα παρακάτω βήματα, όπως φαίνονται και στη εικόνα 7.

1. Επιλογή χαρακτηριστικών: [80,81]

Σε αυτό το βήμα γίνεται επιλογή διακριτών χαρακτηριστικών στα αντικείμενα προς ομαδοποίηση, βάσει των οποίων θα γίνει ομαδοποίηση. Είναι προφανές ότι το βήμα αυτό είναι από τα πιο σημαντικά στην ομαδοποίηση, μια και θα επηρεάσει την αποτελεσματικότητα του αλγόριθμου καθώς επίσης και τα τελικά αποτελέσματα.

2. Επιλογή ή σχεδίαση αλγόριθμου ομαδοποίησης

Το βήμα αυτό συνήθως συνδυάζεται με την επιλογή ενός μέτρου εγγύτητας καθώς επίσης και με την κατασκευή ενός κριτηρίου ομαδοποίησης. Προφανώς το μέτρο εγγύτητας επηρεάζει άμεσα τη διαμόρφωση των ομάδων. Σχεδόν όλοι οι αλγόριθμοι ομαδοποίησης είναι άμεσα συνδεδεμένοι με κάποιο μέτρο εγγύτητας. Από τη στιγμή που θα επιλεγεί ένα μέτρο εγγύτητας, η κατασκευή ενός κριτηρίου ομαδοποίησης μετατρέπεται το πρόβλημα ομαδοποίησης σε πρόβλημα βελτιστοποίησης, το οποίο είναι μαθηματικά επιλύσιμο με μεγάλη βιβλιογραφία στη λύση του. Θα πρέπει να ειπωθεί παρ' όλα αυτά ότι δεν υπάρχει ένας καθολικός αλγόριθμος ομαδοποίησης που να λύνει όλα τα προβλήματα [82].

3. Επικύρωση ομαδοποίησης

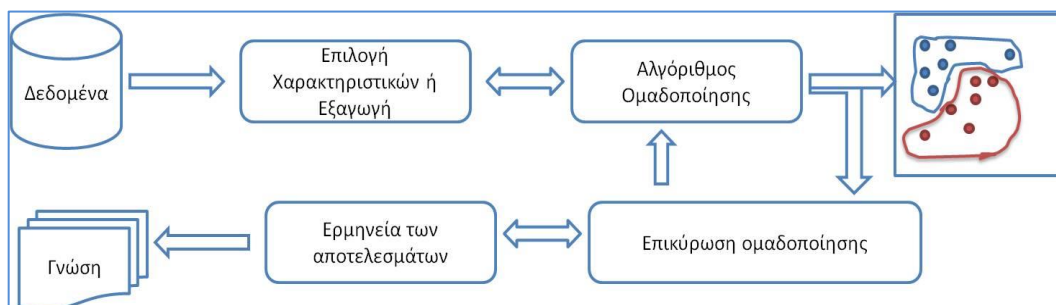
Έχοντας ένα σύνολο από δεδομένα, κάθε αλγόριθμος ομαδοποίησης μπορεί να παράγει αποτελέσματα ανεξάρτητα αν αυτά έχουν λογική βάση για το πρόβλημα ή όχι. Γενικά, διαφορετικές προσεγγίσεις συνήθως οδηγούν σε διαφορετικές ομάδες ακόμα και για τον ίδιο αλγόριθμο, όπως επίσης και η σειρά εισόδου των δεδομένων μπορούν να επηρεάσουν τα τελικά αποτελέσματα. Ως εκ τούτου, τα πρότυπα αξιολόγησης καθώς και τα κριτήρια είναι σημαντικά ώστε να παρέχουν στους χρήστες υψηλό βαθμό εμπιστοσύνης ως προς τα αποτελέσματα. Γενικά υπάρχουν τρεις κατηγορίες κριτηρίων ελέγχου [83]:

- Εξωτερικοί Δείκτες: Βασίζονται σε μια προκαθορισμένη δομή η οποία αποτυπώνει προγενέστερες μορφές των δεδομένων και χρησιμοποιείται ως πρότυπο για να επικυρώνει τις λύσεις ομαδοποίησης.
- Εσωτερικοί Δείκτες: Δεν εξαρτώνται από εξωτερική πληροφόρηση ούτε προηγούμενη γνώση αλλά εξετάζουν την ομαδοποίηση απευθείας από τα αρχικά δεδομένα.
- Σχετικοί Δείκτες: Δίνουν έμφαση στη σύγκριση των διαφορετικών δομών ομαδοποίησης, παρέχοντας την πληροφορία ποια από τις ομαδοποιήσεις μπορεί να καλύψει καλύτερα τα χαρακτηριστικά των αντικειμένων προς ομαδοποίηση.

4. Ερμηνεία των αποτελεσμάτων.

Βασικός στόχος της ομαδοποίησης είναι η παροχή γνώσης προς τον τελικό χρήστη έτσι ώστε να μπορεί να λύσει το αρχικό πρόβλημα. Κάθε ομαδοποίηση παράγει διαφορετικά αποτελέσματα οπότε θα πρέπει να ερμηνευτούν από ειδικούς για την εγκυρότητά τους και την αξιοπιστία τους. Πολλές φορές μάλιστα, μπορεί να χρειαστούν επιπλέον πειράματα και ανάλυση έτσι ώστε να εγγυηθούν για την εγκυρότητα των τελικών αποτελεσμάτων της ομαδοποίησης.

Η ανάλυση σε ομάδες δεν είναι μια εφάπαξ διαδικασία. Σε πολλές περιπτώσεις χρειάζεται μια σειρά από δοκιμές και επαναλήψεις. Επιπλέον, δεν υπάρχουν καθολικά και αποτελεσματικά κριτήρια που να οδηγούν στην επιλογή των σωστών χαρακτηριστικών για τη βέλτιστη ομαδοποίηση. Αν και τα κριτήρια επικύρωσης παρέχουν πληροφορία σχετικά με την ποιότητα των λύσεων ομαδοποίησης, η επιλογή κατάλληλου κριτηρίου εξακολουθεί να είναι ένα πρόβλημα που απαιτεί πολλές προσπάθειες.



Εικόνα 7: Διαδικασία Ομαδοποίησης: Η ανάλυση σε ομάδες αποτελείται από τέσσερα βήματα και ένα μονοπάτι ανάδρασης. Τα βήματα αυτά είναι άμεσα συνδεδεμένα μεταξύ τους και επηρεάζουν τα εξαγόμενα αποτελέσματα.

2.10. Το μοντέλο Core Product Model (CPM)

2.10.1 Εισαγωγή

Η υποστήριξη της διαδικασίας διαχείρισης του κύκλου ζωής του προϊόντος (Product Lifecycle Management - PLM) καθ' όλη τη διάρκεια ζωής του, από τη φάση της σχεδίασης μέχρι τη διάθεσή του στην αγορά απαιτεί αξιόπιστα, πλήρη και αποτελεσματικά μοντέλα σχεδίασης δεδομένων. Η νέα γενιά των μοντέλων σχεδίασης προϊόντων και εφαρμογών προσπαθούν να υποστηρίξουν τη δημιουργία δεδομένων για τη σχεδίασή τους. Παράλληλα υποστηρίζουν την επεξεργασία αυτών των δεδομένων, την αποθήκευση καθώς και την επικοινωνία μεταξύ τους κατά τη διάρκεια ζωής του προϊόντος, από την αρχική σύλληψη της ιδέας του προϊόντος μέχρι την τελική διάθεσή του στην αγορά. Για να επιτευχθεί αυτό είναι προφανές ότι υπάρχει ανάγκη για μοντέλα σχεδίασης που θα είναι αξιόπιστα, πλήρη και αποδοτικά. Στη διαδικασία διαχείρισης του κύκλου ζωής του προϊόντος, τα στοιχεία που είναι απαραίτητα για το σωστό σχεδιασμό είναι συνήθως ετερογενή, ενώ παράλληλα υπάρχει τεράστια ροή πληροφορίας, δύσκολη συνήθως να ελεγχθεί. Είναι σημαντικό λοιπόν για ένα μοντέλο να μπορεί:

- (α) να φιλτράρει τις πληροφορίες,
- (β) να τις δομεί,
- (γ) να μπορεί να ενσωματώσει μόνο την πληροφορία που είναι απαραίτητη και να έχει τον έλεγχο και
- (δ) να τις διοχετεύει έτσι ώστε κάθε επιμέρους διαδικασία να λαμβάνει μόνο την πληροφορία που πραγματικά την αφορά.

Οι αρχικοί στόχοι σχετικά με το CPM ήταν η δυνατότητα ενός κοινού μοντέλου δεδομένων για τις ερευνητικές δραστηριότητες του Εθνικού Ινστιτούτου Προτύπων και Τεχνολογίας (NIST) όπως επίσης και μια βασικού επιπέδου αναπαράσταση για ένα πολυεπίπεδο μοντέλο σχεδίασης για τη ροή της πληροφορίας. Η πρώτη έκδοση του CPM που ανταποκρινόταν σε αυτούς τους στόχους, παρουσιάστηκε στο άρθρο του Fenves τον Οκτώβριο του 2002 [84]. Από

την περιγραφή του μοντέλου γίνεται σαφές ότι πρόκειται για ένα μοντέλο που διαθέτει ορισμένα από τα βασικά χαρακτηριστικά που απαιτούνται για την υποστήριξη της διαδικασίας διαχείρισης του κύκλου ζωής ενός προϊόντος. Στόχος του μοντέλου CPM είναι να χρησιμεύσει ως το βασικό, κορυφαίο επιπέδου μοντέλο, για όλες τις πληροφορίες που απαιτούνται για το σχεδιασμό ενός προϊόντος [85].

2.10.2 Επισκόπηση του μοντέλου

Το μοντέλο CPM είναι ένα γενικό και αφηρημένο μοντέλο με γενική σημασιολογία. Βασίζεται σε δύο αρχές:

- Πρώτον το στοιχείο – κλειδί μέσα στο CPM είναι το αντικείμενο ή προϊόν ή τεχνούργημα (artifact). Το τεχνούργημα αναπαριστά μια διακριτή οντότητα μέσα στο αντικείμενο η οποία οντότητα είναι ένα στοιχείο, μέρος του αντικειμένου, υποσύστημα ή ολόκληρο σύστημα.
- Δεύτερον, το τεχνούργημα είναι μια συνάθροιση από τρία αντικείμενα αναπαριστώντας τις τρεις βασικές αρχές – θεωρήσεις ενός τεχνουργήματος οι οποίες είναι οι παρακάτω:
 - i. Η λειτουργία (function) του τεχνουργήματος περιγράφει τι υποτίθεται κάνει το αντικείμενο. Το τεχνούργημα πρέπει να ικανοποιεί τις ανάγκες των πελατών ή/και τις απαιτήσεις εφαρμοσμένης μηχανικής κατά ένα μεγάλο μέρος μέσω της λειτουργίας του. Ο όρος λειτουργία χρησιμοποιείται συχνά ταυτόσημα με τον όρο επιδιωκόμενη συμπεριφορά.
 - ii. Η μορφή (form) του τεχνουργήματος αντιπροσωπεύει την προτεινόμενη σχεδιαστική λύση βασιζόμενη στη λειτουργία του, όπως περιγράφηκε παραπάνω. Τα φυσικά χαρακτηριστικά του τεχνουργήματος διαμορφώνονται στην CPM με όρους γεωμετρίας και προσδιορισμού υλικών. Αυτό συμβαίνει διότι πολλές από τις προοριζόμενες εφαρμογές τείνουν να μεταχειριστούν διαφορετικά αυτές τις δύο σχεδιαστικές πτυχές.
 - iii. Η συμπεριφορά (behavior) του τεχνουργήματος συνθέτει τις δύο προηγούμενες αρχές, περιγράφοντας πώς η μορφή του τεχνουργήματος εφαρμόζει τη λειτουργία. Η μοντελοποίηση της συμπεριφοράς του τεχνουργήματος διέπεται από τις αρχές σχεδίασης βιομηχανικών προϊόντων και ενσωματώνεται σε ένα μοντέλο συμπεριφορών. Η εφαρμογή του μοντέλου συμπεριφοράς στο τεχνούργημα περιγράφει ή προσομοιώνει την παρατηρηθείσα συμπεριφορά του (observed behavior). Η παρατηρηθείσα συμπεριφορά μπορεί έπειτα να εξεταστεί και να αξιολογηθεί όσον αφορά τις πραγματικές απαιτήσεις, έτσι ώστε να παραχθεί ένα μοντέλο αξιολογημένης συμπεριφοράς (evaluated behavior).

Οι τρεις παραπάνω αρχές- άξονες που περιγράφουν όλη τη δομή καθώς και τη διαχείριση στον κύκλο ζωής ενός προϊόντος, κάνουν τη μέθοδο CPM κατάλληλη πλήρης και αξιόπιστη για τη μοντελοποίηση ενός προϊόντος. Το μοντέλο CPM περιγράφεται επακριβώς με τη βοήθεια της ενοποιημένης γλώσσας σχεδιασμού- Unified Modeling Language (UML). Η γλώσσα UML χρησιμοποιήθηκε αρχικά για τη μοντελοποίηση σύνθετων υπολογιστικών συστημάτων και λογισμικού. Βασίζεται στον αντικειμενοστραφή προγραμματισμό και περιγράφει με γραφικό τρόπο τα επιμέρους στοιχεία ενός συστήματος καθώς και τις συσχετίσεις μεταξύ τους. Τα διαγράμματα κλάσεων της UML ορίζουν γεωμετρικά σχήματα ως συμβολισμούς για τα αντικείμενα, τις κλάσεις και τις διασυνδέσεις, ενώ διαφόρων τύπων γραμμές χρησιμοποιούνται για να συνδέουν αυτά τα σχήματα και να υποδηλώνουν έτσι τον τρόπο που κληρονομούν, συνεργάζονται ή εξαρτώνται μεταξύ τους [86]. Χρησιμοποιώντας την UML περιγράφεται το

μοντέλο CPM με ένα διάγραμμα κλάσεων ή οντοτήτων – σχέσεων [87, 88]. Κάνοντας χρήση της UML επιτυγχάνονται τέσσερις στόχοι:

1. Παράγεται μια οπτική αποτύπωση του συστήματος που γίνεται προσπάθεια να υλοποιηθεί ή να σχεδιασθεί.
2. Προσδιορίζεται η δομή και η συμπεριφορά του συστήματος.
3. Δημιουργείται ένα πρότυπο για να βασιστεί η κατασκευή του συστήματος.
4. Τεκμηριώνονται οι αποφάσεις που ελήφθησαν.

2.10.3 Στοιχεία – συστατικά του μοντέλου

Όπως αναφέρθηκε και παραπάνω, το CPM αναπαριστάται με τη βοήθεια της UML η οποία κάνοντας χρήση του μοντέλου entity-relationship περιγράφει επακριβώς τις σχέσεις μεταξύ οντοτήτων. Κάθε κλάση προσπαθεί να αναπαραστήσει με αφαιρετικό τρόπο κάποια κατηγορία αντικειμένων. Κάθε αντικείμενο είναι στιγμιότυπο μιας κλάσης. Μια κλάση περιέχει, εκτός από το όνομα της και τον τύπο της, ένα σύνολο από ιδιότητες και μεθόδους ή λειτουργίες. Οι ιδιότητες αφορούν τα χαρακτηριστικά των αντικειμένων που περιγράφει η κλάση, ενώ οι μέθοδοι χρησιμοποιούνται για τη διαχείριση των χαρακτηριστικών του αντικειμένου ή για την εκτέλεση συγκεκριμένων ενεργειών. Τα ονόματα των ιδιοτήτων και μεθόδων της κλάσης μπορούν να έχουν ένα πρόθεμα ανάλογα με την ορατότητά τους.

- + Δημόσια πρόσβαση (public)
- # Προστατευμένη (protected)
- Ιδιωτική πρόσβαση (private)

Στην εικόνα 8 φαίνεται μια κλάση που υλοποιεί μια απλή ουρά ακεραίων και μια δεύτερη που υλοποιεί μια στοίβα πραγματικών.

Queue	Stack
-elements:int	-elements:float
-bp, fp:int	-tp, bp:int
+enqueue()	+pop()
+dequeue()	+push()
+lsempty :boolean	+lsempty :boolean
+resise()	+resise()

Εικόνα 8: Κλάσεις για την υλοποίηση μιας ουράς ακεραίων και μια στοίβα πραγματικών αριθμών

Ένα βασικό χαρακτηριστικό στη μοντελοποίηση με κλάσεις είναι η κληρονομικότητα. Κάθε υποκλάση κληρονομεί τις μεθόδους και τις ιδιότητες που έχουν οριστεί στη βασική κλάση. Τέλος θα πρέπει να σημειωθεί ότι στην UML και κατ' επέκταση στο CPM υπάρχουν τέσσερις βασικές σχέσεις [89]:

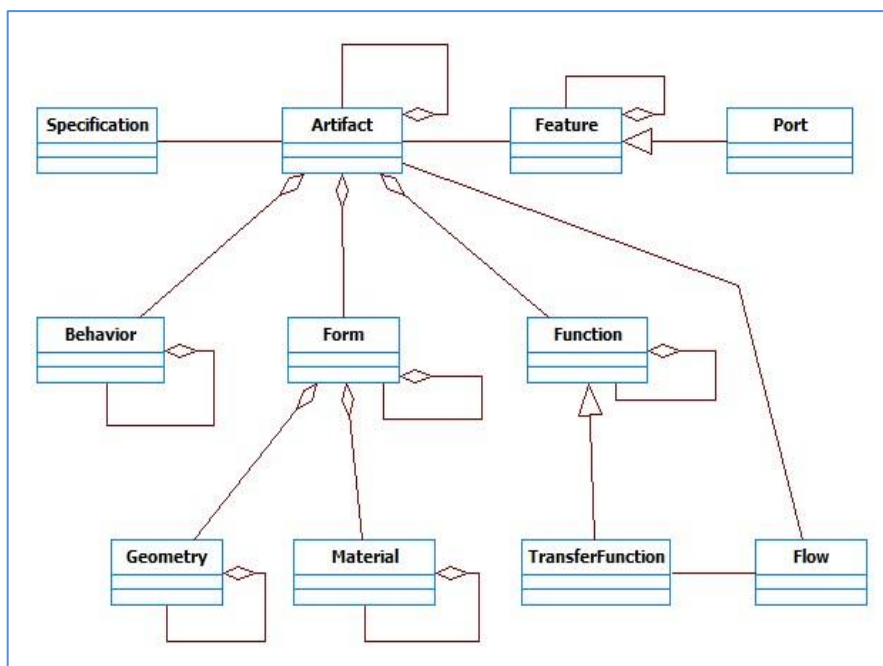
1. Η σχέση εξάρτησης (παριστάνεται με μια διακεκομμένη γραμμή με ανοιχτό βέλος που δείχνει προς την οντότητα που υπάρχει εξάρτηση). Η εξάρτηση δηλώνει πως μια αλλαγή σε μian οντότητα θα επηρεάσει μian άλλη αλλά όχι απαραίτητα και το αντίστροφο.

2. Η σχέση γενίκευσης (δηλώνεται με μια συνεχή γραμμή με κλειστό βέλος που δείχνει προς τη βασική κλάση). Η γενίκευση δηλώνει μια σχέση ανάμεσα σε κάτι γενικό (τη βασική κλάση ή αλλιώς γονέα) και κάτι ειδικό (μιαν υποκλάση ή αλλιώς παιδί της).
3. Η σχέση σύνδεσης. Η σύνδεση αναφέρεται σε αντικείμενα τα οποία συνδέονται με κάποιο τρόπο με άλλα. Όταν δύο κλάσεις είναι συνδεδεμένες μπορεί κανείς να μεταβεί από αντικείμενα της μιας σε αντικείμενα της άλλης. Η σύνδεση παριστάνεται με μια ευθεία γραμμή ανάμεσα στα δύο αντικείμενα όπου:
 - Αν η σύνδεση δεν είναι αμφίδρομη τότε η κατεύθυνσή της μπορεί να οριστεί με ένα ανοιχτό βέλος.
 - Το όνομα της σύνδεσης μπορεί να γραφεί πάνω από τη γραμμή ενώ η κατεύθυνση του ονόματος ορίζεται από ένα βέλος πλάι στο όνομα.
 - Ο ρόλος των οντοτήτων που συνδέονται προσδιορίζεται από ένα όνομα στην κάθε άκρη της γραμμής.
 - Ο αριθμός που δηλώνει πόσα αντικείμενα αντιστοιχούν σε κάθε αντικείμενο στην άλλη άκρη της σχέσης (πολλαπλότητα - multiplicity) δηλώνεται από έναν αριθμό (π.χ. 3) ή μια περιοχή αριθμών (π.χ. 1..* για ένα έως πολλά) πάνω από την αντίστοιχη άκρη της γραμμής.
 - Ένα X πάνω σε μια άκρη της γραμμής δηλώνει πως δεν προσφέρεται μετάβαση (navigation) προς τη συγκεκριμένη κατεύθυνση.
4. Η σχέση υλοποίησης. Η υλοποίηση δηλώνει πως ο εξυπηρετούμενος (αυτός που βρίσκεται στην ουρά του βέλους) υποστηρίζει τη διεπαφή (τουλάχιστον όλες τις πράξεις) που ορίζονται από τον πάροχο (αυτόν που βρίσκεται στην κεφαλή του βέλους).

Το CPM αποτελείται από δύο σύνολα κλάσεων, τις κλάσεις αντικειμένων (objects classes) και τις κλάσεις σχέσεων (relationship classes). Οι δύο αυτές κλάσεις ορίζονται και είναι ισοδύναμες με τις κλάσεις της UML [89,90]. Υπάρχουν πέντε αφαιρετικές κλάσεις οι οποίες δεν έχουν πραγματική αποτύπωση αλλά είναι χρήσιμες για τη θεωρητική αποτύπωση του μοντέλου και ως κλάσεις αναφοράς για τα πραγματικά αντικείμενα και τις σχέσεις τους.

- **CoreProductModel** : Αναπαριστά την πιο υψηλού επιπέδου κλάση γενίκευσης.
- **CommonCoreRelationship**: Είναι η βασική κλάση αναφοράς για όλες τις συσχετίσεις των αντικειμένων μεταξύ τους.
- **CommonCoreObject**: Είναι η βασική κλάση αναφοράς για όλες τις κλάσεις αντικειμένων.
- **CoreEntity**: Είναι η βασική κλάση από την οποία οι κλάσεις **Artifact** και **Feature** κληρονομούν τα στοιχεία τους.
- **CoreProperty**: Είναι η βασική κλάση από την οποία οι κλάσεις **Function**, **Flow**, **Form**, **Geometry** και **Material** κληρονομούν τα στοιχεία τους.

Στην εικόνα 9 φαίνονται οι βασικές οντότητες του μοντέλου καθώς και οι σχέσεις μεταξύ τους.



Εικόνα 9: Το μοντέλο CPM ως περιγραφή οντοτήτων [88].

Για το CPM υπάρχουν έντεκα κλάσεις αντικειμένων και τέσσερις κλάσεις σχέσεων οι οποίες αποτυπώνουν το μοντέλο. Στον [πίνακα 3] φαίνονται οι κλάσεις του μοντέλου καθώς και σε ποια αφαιρετική κλάση ανήκουν.

CPM Αφηρημένες κλάσεις	Πραγματικές κλάσεις - Οντότητες
CoreProductModel	
CommonCoreRelationship	Entity Association, Constraint, Usage, Trace
CommonCoreObject	Behavior, Requirements, Specification
CoreEntity	Artifact, Feature, Port
CoreProperty	Geometry, Material, Form, Function, Flow

Πίνακας 3: Οι αφηρημένες κλάσεις αντικειμένων και οντότητες του CPM-NIST.

Πιο αναλυτικά στον [πίνακα 4] αποτυπώνονται οι οντότητες του CPM και η περιγραφή τους:

Core Product Model (CPM-NIST) – Πίνακας Οντοτήτων - Κλάσεων		
A/A	Οντότητα	Περιγραφή
1	Artifact Τεχνούργημα	Αναπαριστά μια διακριτή οντότητα μέσα στο προϊόν είτε αυτή η οντότητα είναι ένα ολόκληρο προϊόν, ένα εξάρτημα του, μια υποσυναρμολόγηση ή μια συναρμολόγηση.
2	Feature Χαρακτηριστικό	Αποτυπώνει χαρακτηριστικά του σχεδίου, χαρακτηριστικά της ανάλυσης, κατασκευαστικά χαρακτηριστικά, κ.λπ..
3	Specification Προδιαγραφή	Αποτυπώνει όλη την πληροφορία που σχετίζεται με τις ανάγκες του πελάτη, τις κατασκευαστικές απαιτήσεις.
4	Form Μορφή	Περιέχει πληροφορίες για την προτεινόμενη σχεδιαστική λύση με όρους γεωμετρίας και υλικών.

5	Function Λειτουργία	Περιγράφει τι αναμένεται να γίνει το τεχνούργημα, δηλαδή ποια είναι η προβλεπόμενη συμπεριφορά του.
6	Behavior Συμπεριφορά	Περιγράφει πώς η μορφή του τεχνουργήματος εφαρμόζει τη λειτουργία.
7	Geometry Γεωμετρία	Αποτυπώνεται η χωρική περιγραφή του τεχνουργήματος.
8	Material Υλικό	Περιγράφει την εσωτερική σύνθεση του τεχνουργήματος.
9	Port Σύνδεση	Αποτελεί μια εξειδίκευση της οντότητας Feature στην οποία αποτυπώνονται πληροφορίες για πιθανή σύνδεση ή διασύνδεση του τεχνουργήματος με άλλα τεχνουργήματα.
10	Transfer Function Συνάρτηση Μεταφοράς	Αποτελεί στην ουσία ένα εξειδικευμένο χαρακτηριστικό της οντότητας Function που περιγράφει τη μεταφορά πληροφορίας από μια ροή εισόδου σε μία ροή εξόδου.
11	Flow Ροή	Είναι το μέσον που υλοποιεί τη μεταφορά πληροφορίας ως είσοδο και έξοδο των Transfer Functions. Προγραμματιστικά μοιάζει με τις παραμέτρους που μια συνάρτηση περνά ή επιστρέφει ενημερωμένες μετά την εκτέλεσή της.

Πίνακας 4: Οι κλάσεις – στιγμιότυπα του CPM.

Εκτός από τις παραπάνω έντεκα κλάσεις αντικειμένων υπάρχουν τέσσερις κλάσεις σχέσεων οι οποίες αποτυπώνουν τις σχέσεις μεταξύ των οντοτήτων. Ακολουθεί ο [πίνακας 5] με τις κλάσεις και την περιγραφή τους:

Core Product Model (CPM-NIST) – Πίνακας Κλάσεων Σχέσεων		
A/A	Οντότητα	Περιγραφή
1	Constraint Περιορισμός	Αποτυπώνει τη σχέση υπό μορφή περιορισμών μεταξύ οντοτήτων και είναι προσβάσιμη (κοινόχρηστη) για το σύνολο των οντοτήτων που εμπλέκονται στον περιορισμό που περιγράφεται.
2	Entity Association Συσχέτιση Στοιχείου	Περιλαμβάνει όλες τις σχέσεις μεταξύ των κλάσεων Artifact, Feature και Port.
3	Usage Χρήση	Είναι μια κλάση συσχέτισης ιδιαίτερα χρήσιμη όταν υπάρχουν περιορισμοί σε μια οντότητα αλλά όχι στη γενικευμένη από την οποία κληρονομεί ιδιότητες και μεθόδους. Επίσης χρήσιμη όταν γίνεται χρήση οντοτήτων από εξωτερικές πηγές ή από αποθετήρια σχεδίων.
4	Trace Ίχνος	Δομικά είναι ίδια με την κλάση Usage. Είναι χρήσιμη για να περιγράψει τη σχέση μιας οντότητας στην τρέχουσα περιγραφή του προϊόντος με μια οντότητα σε άλλη περιγραφή του ίδιου προϊόντος. Μια ιδιότητα της, η Type για παράδειγμα, παίρνει τις τιμές alternative_of, version_of, derived_of, is_based_on κ.λπ. και επομένως περιγράφει τη φύση της εξάρτησης.

Πίνακας 5 : Κλάσεις σχέσεων για το CPM- NIST

2.10.4 Μοντέλα CPM

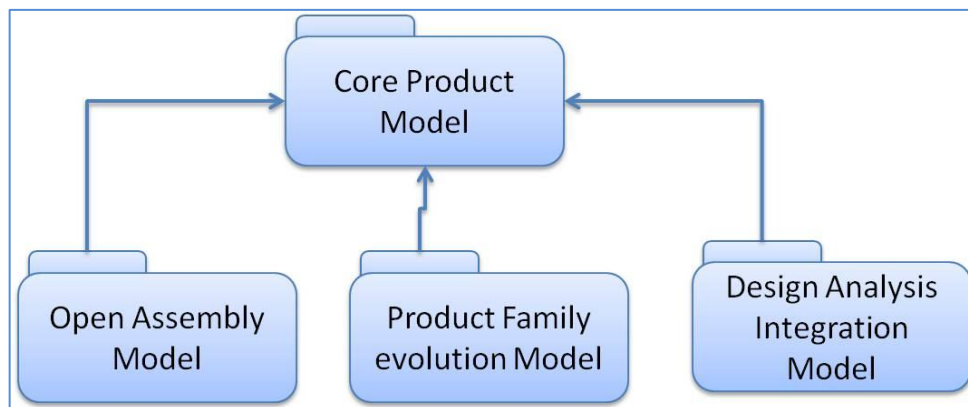
Το μοντέλο CPM λειτουργεί σε τρία επίπεδα μοντελοποίησης τα οποία δηλώνονται ως:

- Εννοιολογικό μοντέλο (Conceptual).
- Ενδιάμεσο μοντέλο (Intermediate).
- Μοντέλο Υλοποίησης (Implementation).

Το εννοιολογικό μοντέλο προσφέρει μια γενική εικόνα για το προϊόν που θα σχεδιαστεί, προσφέροντας πληροφορίες για τις κλάσεις αντικειμένων και τις σχέσεις μεταξύ τους, χωρίς να μπορεί να αποτυπώσει ιδιότητες των αντικειμένων που σχετίζονται με συγκεκριμένα χαρακτηριστικά. Παρ' όλα αυτά η χρήση του με αυτή τη μορφή, αν και έχει αφαιρετικό χαρακτήρα, προσφέρει δυνατότητες επέκτασης σε μοντέλα πιο ειδικευμένα όπως:

1. Μοντέλο συναρμολόγησης (Open Assembly Model - OAM) το οποίο παρέχει όλη την πληροφορία σε επίπεδο συστήματος, τόσο για τη συναρμολόγηση, όσο και για τις σχέσεις ιεραρχίας μεταξύ των υποσυναρμολογήσεων.
2. Μοντέλο οικογένειας προϊόντων (Product Family Evolution Model - PFEM) το οποίο ενσωματώνει την πληροφορία για σχεδιασμό οικογένειας προϊόντων με κοινά χαρακτηριστικά, μειώνοντας έτσι το κόστος ανάπτυξης [92].
3. Μοντέλο ενοποίησης σχεδιασμού και ανάλυσης (Design Analysis Integration model - DAIM), το οποίο ως εννοιολογική αρχιτεκτονική δεδομένων παρέχει όλη την πληροφορία από τεχνικής πλευράς ώστε να υπάρχει ενοποίηση των σύγχρονων εργαλείων ανάπτυξης προϊόντων με τα πληροφοριακά μοντέλα [93].

Στην εικόνα 10 παρουσιάζεται η σχέση των μοντέλων αυτών ως προς το αρχικό μοντέλο που όλα μαζί αποτελούν το πλαίσιο μοντελοποίησης της πληροφορίας ενός προϊόντος σύμφωνα με το NIST [100].



Εικόνα 10: Πλαίσιο μοντελοποίησης της πληροφορίας (NIST) [84].

Για να χρησιμοποιηθεί αυτό ως ενδιάμεσο μοντέλο, πρέπει στο αρχικό εννοιολογικό μοντέλο να ενσωματωθούν δύο ακόμα χαρακτηριστικά. Πρώτα από όλα θα πρέπει κάθε αντικείμενο και κάθε σχέση μεταξύ των αντικειμένων να διαθέτει μια κλάση μεταφορέα πληροφορίας (container class), η οποία στην πραγματικότητα είναι μια δομή δεδομένων, η οποία χρησιμοποιείται για να αποθηκεύει τα αντικείμενα και τις σχέσεις τους με ένα οργανωμένο τρόπο και συγκεκριμένους κανόνες. Για τη συγκεκριμένη υλοποίηση, ιδιαίτερο ενδιαφέρον έχει η αποτύπωση των ιδιοτήτων και οι τιμές τους σε αυτά τα χαρακτηριστικά που είναι εξειδικευμένα στον τομέα σχεδίασης που υλοποιείται το μοντέλο. Δεύτερον, κάθε κλάση αντικειμένου και

σχέση, εκτός από τις αφαιρετικές, θα πρέπει να έχει ένα επιπλέον χαρακτηριστικό το οποίο ονομάζεται τύπος (**Type**) του οποίου η τιμή λειτουργεί ως ιεραρχικός ταξινομητής. Ορίζοντας μια διακριτή ιεραρχική ταξινόμηση σε κάθε κλάση αντικειμένου και σχέση μπορεί εύκολα να κατηγοριοποιηθούν οι κλάσεις και οι σχέσεις με βάση τις πραγματικές ανάγκες της σχεδίασης.

Πρακτικά, όσο η σχεδίαση περνά από το εννοιολογικό στο ενδιάμεσο μοντέλο, οι κλάσεις αντικειμένων και σχέσεων μπορεί να απαιτούν μια πιο ατομική και διακριτή κατηγοριοποίηση η οποία μπορεί να οδηγήσει σε οντότητες με πλήρη περιγραφή όλων των εννοιών τους. Παρ' όλα αυτά, το ενδιάμεσο μοντέλο δεν μπορεί να αποτυπωθεί σε ένα μοντέλο υλοποίησης και αυτό διότι είναι πολλές διαφορετικές περιπτώσεις και επομένως μη υλοποιήσιμο.

Τέλος για να μπορεί το CPM να εφαρμοστεί σε πραγματικές περιπτώσεις, το εννοιολογικό αρχικό μοντέλο θα πρέπει να μετατραπεί σε ένα μοντέλο υλοποίησης. Αυτό απαιτεί τη μετάφραση όλων των κλάσεων αντικειμένων και σχέσεων από τον αρχικό εννοιολογικό σχεδιασμό σε μια πλατφόρμα αντικειμενοστραφούς γλώσσας. Πιο συγκεκριμένα ένα τέτοιο λογισμικό θα πρέπει:

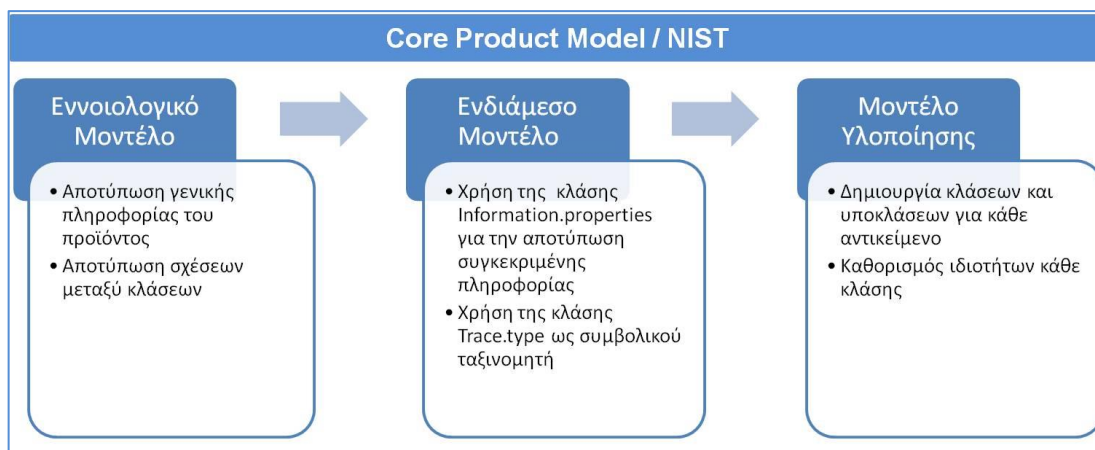
1. Να δημιουργεί όλες τις υποκλάσεις του αρχικού τεχνουργήματος βασιζόμενο στην ιεραρχική κατηγοριοποίηση, κάνοντας χρήση της ιδιότητας **τύπου** που αναφέρθηκε παραπάνω.
2. Να ορίζει τις ιδιότητες των υποκλάσεων βασιζόμενο στην κλάση μεταφορέα **πληροφορίας** και πιο συγκεκριμένα το τμήμα **ιδιότητες** το οποίο περιέχει ζεύγη ιδιοτήτων – τιμών για κάθε ξεχωριστή ιδιότητα.

Η μετατροπή αυτή γίνεται μεταφράζοντας το μοντέλο από την αρχική ανεξάρτητη πλατφόρμα (Platform Independent Models-PIMs) στην οποία έχει υλοποιηθεί, σε ένα μοντέλο ειδικευμένης πλατφόρμας (Platform Specific Models-SMs) το οποίο στη βιβλιογραφία συναντάται ως μοντέλο σύνθεσης και είναι ένα τμήμα της αρχιτεκτονικής καθοδηγούμενης από το μοντέλο (Model Driven Architecture -MDA), όπως αυτή ορίζεται από την Object Management Group (OMG) [95].

Σε επίπεδο υλοποίησης, το CPM χρησιμοποιώντας την κλάση **Artifact** για να αποτυπώσει την πληροφορία για κάθε προϊόν πετυχαίνει παράλληλα:

1. Την περιγραφή κάθε κλάσης και επομένως φυσικού αντικειμένου.
2. Την επαναχρησιμοποίηση τμημάτων για πιο σύνθετη σχεδίαση.
3. Την πλήρη παρακολούθηση κάθε αντικειμένου σε όλη τη διάρκεια της ζωής του.

Στην εικόνα 11 αποτυπώνεται η εξέλιξη του μοντέλου CPM από το εννοιολογικό επίπεδο στο επίπεδο υλοποίησης.



Εικόνα 11: Αποτύπωση των τριών επιπέδων του Core Product Model.

2.10.5 Προεκτάσεις του CPM

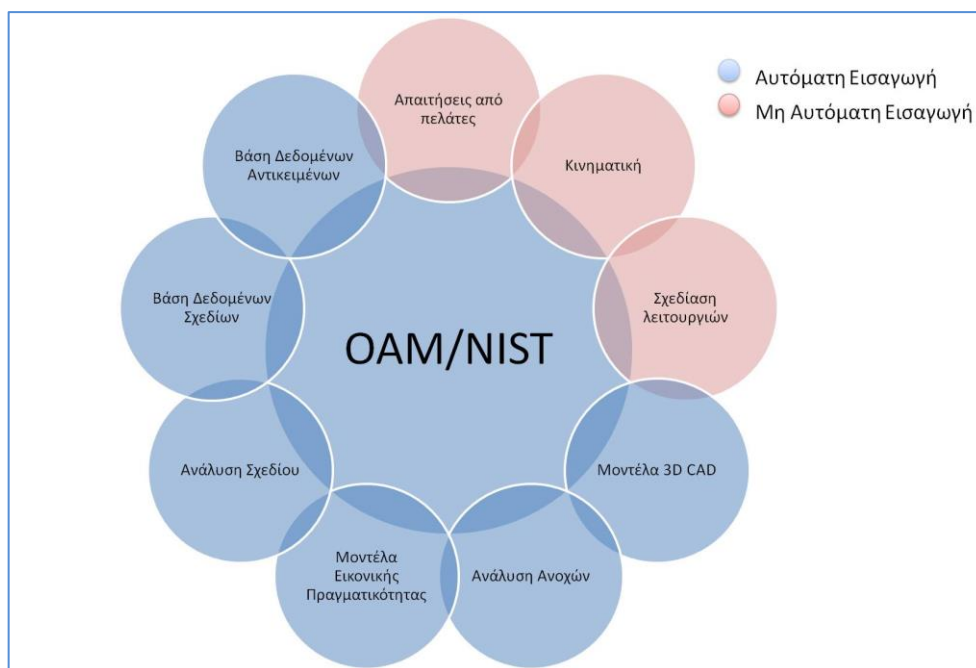
Μετά την ανάπτυξη του CPM ως ενός μοντέλου σχεδίασης προϊόντων ανεξάρτητου πλατφόρμας λογισμικού CAD ανοικτής αρχιτεκτονικής, επεκτάσιμου και ανεξάρτητου από οποιαδήποτε διεργασία εξέλιξης του προϊόντος, ήρθε να προστεθεί ένα σύνολο από προεκτάσεις – διαφοροποιήσεις του αρχικού μοντέλου. Συγκεκριμένα οι παρακάτω επεκτάσεις του αρχικού CPM έχουν ήδη τεκμηριωθεί:

1. Το μοντέλο ανοικτής συναρμολόγησης (Open Assembly Model - OAM). Το μοντέλο αυτό παρέχει όλα τα πρότυπα και τα πρωτόκολλα για τη συναρμολόγηση προϊόντων. Σχετική επισκόπηση του μοντέλου γίνεται στην επόμενη ενότητα [63], [96, 97].
2. Το μοντέλο-γλώσσα σημασιολογικής αναπαράστασης προϊόντων (Product Semantic Representation Language -PSRL). Η γλώσσα αυτή κάνει χρήση του CPM για μια αυστηρή αναπαράσταση της συνολικής πληροφορίας ενός προϊόντος [98].
3. Το μοντέλο ενοποιημένου σχεδιασμού και ανάλυσης (Design Analysis Integration Model) προτείνει μια πλατφόρμα εννοιολογικής αρχιτεκτονικής, συμβάλλοντας έτσι στην ενοποίηση χωροταξικών και λειτουργικών πληροφοριών στη σχεδίαση του προϊόντος [93].
4. Το μοντέλο εξέλιξης οικογένειας προϊόντων (Product Family Evolution Model) επεκτείνει το αρχικό CPM αναπαριστώντας την εξέλιξη οικογένειας προϊόντων. Αυτό επιτυγχάνεται με την αποτύπωση των συστατικών κάθε προϊόντος μέσω πληροφορίας για την οικογένεια που ανήκει, τη σειρά, την έκδοσή του, κ.λπ. [99].
5. Το μοντέλο ετερογενών υλικών (Heterogeneous Material Model) επεκτείνει το CPM για συστατικά του προϊόντος με ετερογενή χαρακτηριστικά και ιδιότητες [100].
6. Το μοντέλο μηχανικής συσκευής (Mechatronic Device Model) είναι ένα πλαίσιο που υποστηρίζει τον εννοιολογικό σχεδιασμό πολλαπλών θεωρητικών μηχανών που αλληλεπιδρούν μεταξύ τους στη λογική καταστάσεων μετάβασης και κάθε συσκευή έχει διαφορετικά ποιοτικά χαρακτηριστικά [101].
7. Το μοντέλο συστημάτων ενσωμάτωσης (Embedded System Model) είναι μια προσέγγιση του CPM βασισμένη στα χαρακτηριστικά ενός συστήματος και λειτουργεί ως παράλληλη σχεδίαση λογισμικού και υλικού σε ενσωματωμένα συστήματα [102].

2.11 Το μοντέλο Open Assembly Model /NIST-CPM (OAM/NIST-CPM)

2.11.1 Επισκόπηση του μοντέλου

Το βασικό χαρακτηριστικό του μοντέλου OAM/NIST που το διαφοροποιεί από άλλα μοντέλα πρότυπα είναι η δυνατότητά του να παρακολουθεί την εξέλιξη της σχεδίασης και συναρμολόγησης ενός προϊόντος σε κάθε του φάση. Αυτό είναι εφικτό μιας και το OAM αποτελεί επέκταση του CPM το οποίο αποτυπώνει με τρόπο, ανεξάρτητο της πλατφόρμας, τη σχεδίαση από τα πρώιμα της στάδια. Το μοντέλο αυτό αποτελεί ένα ισχυρό εργαλείο το οποίο μπορεί να αναπαραστήσει όλη την πληροφορία που απαιτείται και ως έννοια αλλά και ως δομή δεδομένων για την πλήρη αποτύπωση της σχεδίασης και συναρμολόγησης του προϊόντος. Παράλληλα, ενσωματώνει όλες τις κλάσεις αντικειμένων καθώς και τις σχέσεις μεταξύ τους για την αποτύπωση πληροφορίας που σχετίζονται με ανοχές (Tolerance), κινηματική (Kinematic) και λογικές σχεδίασης (Design Rationale). Επιγραμματικά, το OAM αποτελεί ένα μοντέλο διασύνδεσης με όλα εκείνα τα λογισμικά που συνεισφέρουν το καθένα από τη δική του σκοπιά στην τελική σχεδίαση ενός προϊόντος. Παράλληλα, λόγω της ανοικτής αρχιτεκτονικής του η οποία είναι ανεξάρτητη της πλατφόρμας υλοποίησης, επιλύει προβλήματα διασύνδεσης που συνήθως οι σχεδιαστές αντιμετωπίζουν από τη μια πλατφόρμα στην άλλη. Στην εικόνα 12 φαίνεται η χρήση του OAM/NIST σε κάθε φάση της σχεδίασης και προφανώς γίνεται ξεκάθαρος ο διαλειτουργικός του χαρακτήρας.



Εικόνα 12: Το μοντέλο OAM ως διεπαφή στην εξέλιξη της σχεδίασης και συναρμολόγησης ενός προϊόντος.

2.11.2 Στοιχεία – συστατικά του μοντέλου

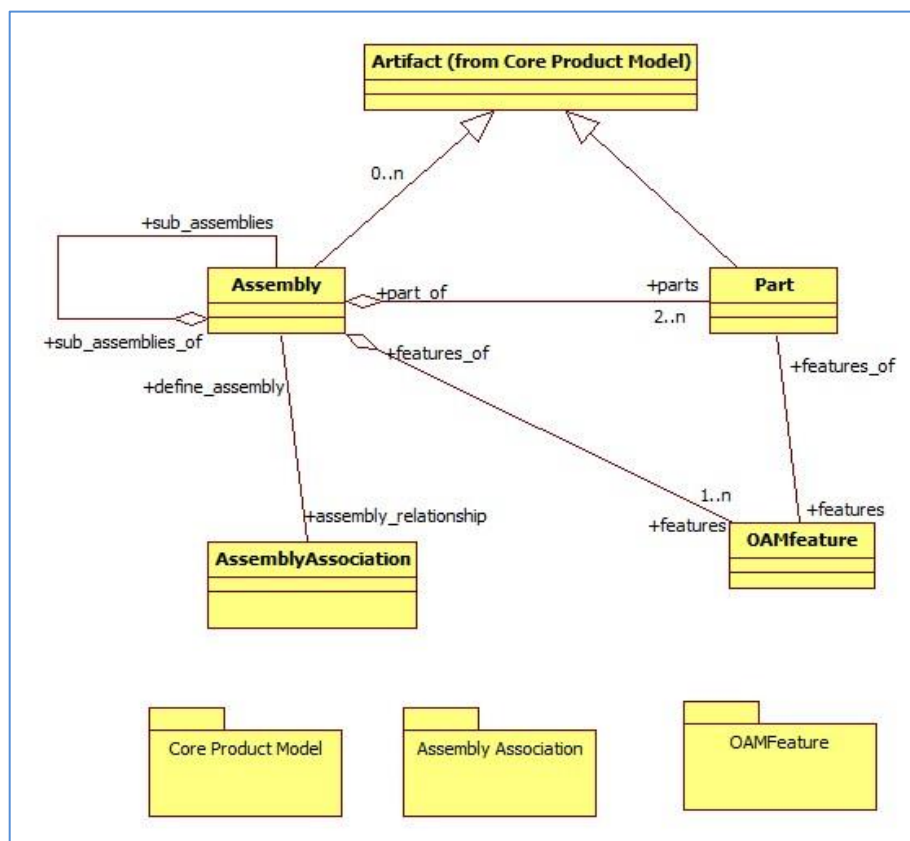
Όπως και στο μοντέλο CPM/NIST, το μοντέλο OAM/NIST περιγράφεται κάνοντας χρήση της γλώσσας UML ως ένα σύνολο οντοτήτων και σχέσεων, επεκτείνοντας και εξειδικεύοντας τις κλάσεις που ήδη έχουν περιγραφεί στο CPM/NIST. Τα βασικά χαρακτηριστικά του OAM/NIST είναι:

- Κάθε προϊόν αντιμετωπίζεται ως μια συναρμολόγηση (**Assembly**) η οποία απαρτίζεται από υποσυναρμολογήσεις (**subassemblies**) και εξαρτήματα του προϊόντος

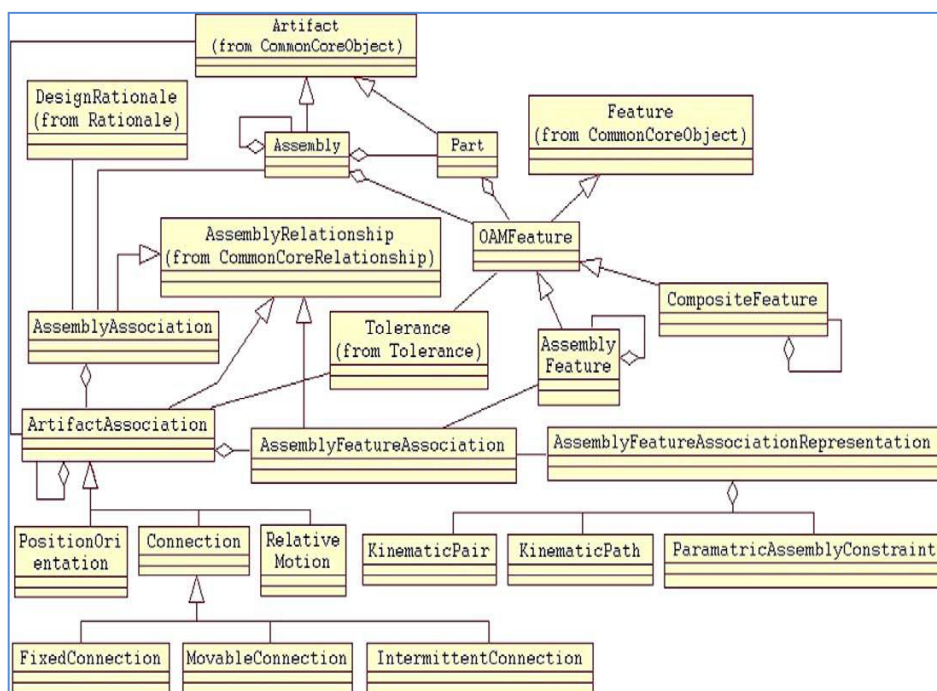
- (parts). Οι κλάσεις **συναρμολόγηση** και **Εξάρτημα** είναι υποκλάσεις του **τεχνουργήματος** όπως αυτή ορίζεται στο CPM/NIST.
- Ως **εξαρτήματα** ορίζονται τα μέρη του προϊόντος που δεν μπορούν να αποσυντεθούν περαιτέρω, είναι δηλαδή τα δομικά στοιχεία μιας συναρμολόγησης.
 - Κάθε συναρμολόγηση ή υποσυναρμολόγηση γίνεται από δύο τουλάχιστον εξαρτήματα.
 - Κάθε συναρμολόγηση (υποσυναρμολόγηση ή εξάρτημα) αποτελείται από ένα ή περισσότερα χαρακτηριστικά που αποτυπώνονται στο μοντέλο με τη βοήθεια της κλάσης **OAMFeature** η οποία είναι υποκλάση της **Feature** κατά CPM/NIST. Η κλάση **OAMFeature** επιμερίζεται στις:
 - Κλάση **Tolerance** στην οποία αποτυπώνονται πληροφορίες ανοχής στη συναρμολόγηση. Η κλάση **Tolerance** ορίζεται σύμφωνα με τα πρότυπα ISO 286.
 - Υποκλάση **AssemblyFeature** στην οποία αποτυπώνονται γεωμετρικά χαρακτηριστικά των στοιχείων ενός προϊόντος.
 - Υποκλάση **CompositeFeature** στην οποία αποτυπώνονται σύνθετα χαρακτηριστικά τα οποία μπορούν να αποσυντεθούν σε απλά.
 - Οι σχέσεις συναρμολόγησης (**Assembly Association**) αποτυπώνονται στο μοντέλο κάνοντας χρήση της σχέσης μέρος-του (part-of) και είναι μια συνάθροιση των σχέσεων που περιγράφονται στην κλάση **Artifact Association**, όπως αυτή ορίζεται στο CPM/NIST. Η **Artifact Association** εξειδικεύεται στις παρακάτω κλάσεις:
 - **PositionOrientation** η οποία περιγράφει και αποτυπώνει τους περιορισμούς όσο αφορά τη σχετική θέση και κατεύθυνση δύο ή περισσότερων αντικειμένων που δε συνδέονται με φυσικό τρόπο μεταξύ τους.
 - **RelativeMotion** η οποία περιγράφει τη σχετική κίνηση και τους περιορισμούς που υπάρχουν μεταξύ δυο αντικειμένων που συνδέονται μεταξύ τους.
 - **ConnectionType** η οποία περιγράφει τον τύπο της σύνδεσης μεταξύ των φυσικά συνδεδεμένων αντικειμένων. Η κλάση αυτή εξειδικεύεται περαιτέρω σε:
 - **FixedConnection** η οποία περιγράφει σταθερές συνδέσεις όπως συνδέσεις με βίδες, πριτσίνια, συγκολλήσεις κ.λπ..
 - **MovableConnection** η οποία περιγράφει την κινηματική της σύνδεσης αντικειμένων με κίνηση του ενός προς το άλλο, όπως για παράδειγμα σύνδεση δυο γραναζιών.
 - **IntermittentConnection** η οποία περιγράφει συνδέσεις οι οποίες έχουν παροδικό χαρακτήρα. Τέτοιες συνδέσεις είναι για παράδειγμα διακόπτες, επαφές με χρήση ελατηρίων κ.λπ..
 - Η κλάση **AssemblyFeatureAssociationRepresentation** αναπαριστά τις σχέσεις συναρμολόγησης μεταξύ δύο χαρακτηριστικών συναρμολόγησης. Στην κλάση αυτή αποτυπώνονται όλες οι πληροφορίες που αφορούν τους περιορισμούς των συναρμολογήσεων, πληροφορίες κινηματικής και ότι σχετίζεται με την κίνηση μεταξύ των συναρμολογούμενων στοιχείων. Πιο συγκεκριμένα η κλάση είναι μια συνάθροιση των κλάσεων:
 - **ParametricAssemblyConstraint** η οποία αποτυπώνει όλους τους γεωμετρικούς περιορισμούς μεταξύ των συναρμολογήσεων (θέση, κατεύθυνση κ.λπ.). Η υποκλάση αυτή ορίζεται επακριβώς στο ISO 10303 Part108.

- **KinematicPair** η οποία ορίζει τα χαρακτηριστικά και τους περιορισμούς κινηματικής μεταξύ των συναρμολογήσεων. Ορίζεται επακριβώς στο ISO 10303 Part105.
- **KinematicPath** η οποία αποτυπώνει τη σχετική κίνηση μεταξύ των συναρμολογούμενων μερών. Ορίζεται επακριβώς στο ISO 10303 Part105.

Στις εικόνες 13 και 14 αποτυπώνεται το διάγραμμα σε UML για το μοντέλο OAM σε συνοπτική και αναλυτική μορφή. Αναλυτική περιγραφή του μοντέλου γίνεται στο άρθρο των Sudarsnan et.al [63].



Εικόνα 13: Το μοντέλο OAM σε συνοπτική μορφή [97].



Εικόνα 14: Το μοντέλο OAM [97].

3. OAM2DSM: Ενοποίηση των μοντέλων OAM και DSM

3.1 Εισαγωγή

Βασικός στόχος και επιδίωξη της νέας γενιάς μοντέλων σχεδίασης προϊόντων είναι η υποστήριξη και διαχείριση ολόκληρου του κύκλου ζωής τους (PLM) με τη δημιουργία δεδομένων και πληροφοριών για το προϊόν, από τη σχεδίαση, τη συναρμολόγηση, τη χρήση, τη συντήρηση μέχρι και την ανακύκλωσή του. Το NIST κινούμενο σε αυτήν τη λογική παρουσίασε το μοντέλο OAM που αποτελεί μια επέκταση του CPM και επομένως κληρονομεί τα χαρακτηριστικά του, δηλαδή είναι ανοικτό σε διαφορετικές αρχιτεκτονικές και πλατφόρμες σχεδίασης [84]. Το βασικό πλεονέκτημα του OAM σε σχέση με τα υπάρχοντα μοντέλα συναρμολόγησης είναι η δυνατότητα χρήσης του από τα πρώιμα στάδια της σχεδίασης συμβάλλοντας έτσι στην ενοποίηση των λειτουργικών αναγκών και των προτεινόμενων τεχνικών λύσεων. Παράλληλα, αποτελεί ένα ισχυρό εργαλείο το οποίο μπορεί να αναπαραστήσει όλη την πληροφορία που απαιτείται και ως έννοια αλλά και ως δομή δεδομένων, για την πλήρη αποτύπωση της σχεδίασης και συναρμολόγησης του προϊόντος. Αποτελεί λοιπόν μια ικανή αλλά και αναγκαία πλατφόρμα πάνω στην οποία μπορούν να προστεθούν χαρακτηριστικά άλλων μοντέλων ανάπτυξης προϊόντων, ενοποιώντας μοντέλα ανάπτυξης ή ενδυναμώνοντας το OAM.

Τέλος, το OAM είναι ένα μοντέλο αναπαράστασης της συναρμολόγησης ενός προϊόντος που βασίζεται στο STEP. Το STEP, όπως περιγράφεται στο ISO 10303, εστιάζει μόνο στα γεωμετρικά χαρακτηριστικά του προϊόντος ή των μερών και υποσυναρμολογήσεων, ενώ το OAM αποτυπώνει και σημασιολογικές πληροφορίες για τη συναρμολόγηση. Βέβαια, το μοντέλο OAM, αν και έχει τεκμηριωθεί πλήρως αποτυπώνοντας τη συναρμολόγηση και σχεδιαστικά αλλά και ως σύνολο από δομές δεδομένων, δεν έχει φτάσει σε επίπεδο υλοποίησης [63]. Η πολυπλοκότητα των δομών δεδομένων του έτσι ώστε να καλύψει τη σχεδίαση και ανάπτυξη ενός προϊόντος από τα πρώιμα στάδια, το κάνει πρακτικά μη υλοποιήσιμο.

Στην παρούσα διατριβή επιχειρείται η δημιουργία ενός νέου μοντέλου το οποίο περιλαμβάνει τη μετατροπή του ΟΑΜ από το εννοιολογικό επίπεδο στο επίπεδο υλοποίησης, τεκμηριώνοντας επακριβώς, τόσο τις οντότητες που περιλαμβάνει εξ ορισμού, όσο και τις κλάσεις αντικειμένων και σχέσεων. Στη συνέχεια, πάνω στο υλοποιούμενο μοντέλο ΟΑΜ, ενσωματώνονται χαρακτηριστικά του Πίνακα Δομής Σχεδίασης (ΠΔΣ – DSM) και συγκεκριμένα η δυνατότητα ομαδοποίησης συναρμολογήσεων και επομένως εξαρτημάτων. Όπως έχει ήδη αναλυθεί στην ενότητα 2.2.1, ένα μοντέλο ΠΔΣ είναι σε θέση να αποτυπώσει τις συσχετίσεις και τις αλληλεπιδράσεις μεταξύ των στοιχείων (βασικά δομικά στοιχεία, υποσυναρμολογήσεις) ενός προϊόντος. Πιο συγκεκριμένα, η χρήση ΠΔΣ αρχιτεκτονικής (συστατικών) με τη βοήθεια αλγορίθμων ομαδοποίησης εστιάζει στην εύρεση υποσυνόλων στοιχείων μιας συναρμολόγησης που έχουν ισχυρή εξάρτηση μεταξύ τους, ελαχιστοποιώντας έτσι προβλήματα στην αρχική σχεδίαση (αναδράσεις ή επαναλήψεις).

Αυτό που προτείνεται στην παρούσα διατριβή, είναι ένας τροποποιημένος ΠΔΣ που τα στοιχεία του πίνακα δεν είναι τα μέρη ή εξαρτήματα ενός προϊόντος αλλά οι συναρμολογήσεις μεταξύ τους. Στη συνέχεια, αντιμετωπίζοντας τον ΠΔΣ ως ένα πίνακα αρχιτεκτονικής ή συστατικών, αναπτύσσεται μια μέθοδος ομαδοποίησης που στην ουσία ψάχνει συναρμολογήσεις με πολλά κοινά στοιχεία μεταξύ τους.

Η εργασία αυτή κάνοντας χρήση ενός ευρετικού και επιμεριστικού αλγόριθμου ομαδοποίησης, προτείνει πιθανές ομάδες συναρμολογήσεων των εξαρτημάτων ενός προϊόντος, κρατώντας σε ισορροπία το χρόνο συναρμολόγησης κάθε ομάδας σε σχέσεις με τις άλλες. Το μοντέλο που προκύπτει αποτελεί μια νέα πλατφόρμα στη μεθοδολογία σχεδίασης και ανάπτυξης προϊόντων και ταυτόχρονα ένα πλήρες πληροφοριακό σύστημα με κύριο προσανατολισμό στη συναρμολόγηση. Αυτό το επιπλέον χαρακτηριστικό ενσωματώνεται στο τελικό μοντέλο, παρέχοντας αυξημένες δυνατότητες, τόσο στη ροή της πληροφορίας μεταξύ των σταδίων ανάπτυξης του προϊόντος, αλλά και μια εν γένει δυνατότητα καλύτερης διαχείρισης του χρόνου συναρμολόγησης.

Η μεταφορά του ΟΑΜ από το εννοιολογικό στο μοντέλο υλοποίησης διευθετεί τα παρακάτω ζητήματα για να υλοποιηθεί:

- Κάθε αντικείμενο και κάθε σχέση μεταξύ των αντικειμένων πρέπει να διαθέτει μια κλάση μεταφορέα πληροφορίας (container class) η οποία στην πραγματικότητα είναι μια δομή δεδομένων που χρησιμοποιείται για να αποθηκεύει τα αντικείμενα και τις σχέσεις τους με ένα οργανωμένο τρόπο και συγκεκριμένους κανόνες.
- Κάθε κλάση αντικειμένου και σχέση θα πρέπει να έχει ένα επιπλέον χαρακτηριστικό το οποίο ονομάζεται τύπος (**Type**) του οποίου η τιμή λειτουργεί ως ιεραρχικός ταξινομητής.

Για τις ανάγκες της διατριβής, οι κλάσεις μεταφορέα υλοποιήθηκαν με τη βοήθεια σχεσιακών οντοτήτων και αντικειμενοστραφούς λογικής στη σχεδίαση των κλάσεων, στο μέτρο της πολυπλοκότητας που είναι εφικτό (Ενότητα 3.3.3). Το τελικό μοντέλο, που περιγράφεται στην παρακάτω ενότητα συνδέει τις δύο αυτές μεθόδους χρησιμοποιώντας το ΟΑΜ ως μοντέλο διεπαφής μεταξύ των λογισμικών σχεδίασης και του μοντέλου ΠΔΣ. Η ενοποίηση αυτή συμβάλει:

- Υποστηρίζοντας τη διαδικασία σχεδίασης από τα πρώιμα στάδιά της και καθ' όλη τη διάρκεια του κύκλου ζωής ενός προϊόντος.
- Ισχυροποιώντας την ενοποίηση συστημάτων σχεδίασης (συστήματα CAD) με συστήματα ανάλυσης.

- Δίνοντας μια δυνατότητα πρόβλεψης (look ahead) στη σχεδίαση, συμβάλλοντας έτσι στη μείωση του χρόνου σχεδίασης, μιας και εντοπίζονται γρηγορότερα σχεδιαστικές αστοχίες ή λάθη.
- Παρέχοντας ένα ισχυρό πληροφοριακό μοντέλο, ικανό να αποτυπώσει την εξέλιξη της σχεδίασης και τη ροή της πληροφορίας σε όλα της τα στάδια.
- Παρέχοντας ανεξαρτησία από την υποδομή στα συστήματα σχεδίασης (Απαραίτητη προϋπόθεση είναι το λογισμικό σχεδίασης να υποστηρίζει το πρότυπο ISO10303 STEP).
- Προσθέτοντας στο OAM ένα επιπλέον χαρακτηριστικό, αυτό της τμηματικής σχεδίασης μιας και η ομαδοποίηση που προτείνεται στον ΠΔΣ αρχιτεκτονικής δίνει ομάδες συναρμολογήσεων με ισχυρή αλληλεπίδραση και εξάρτηση. Αυτή η δυνατότητα ισχυροποιεί τη συνεργατική σχεδίαση (collaborative design).

3.2 Μοντελοποίηση του OAM2DSM

3.2.1 Εισαγωγή

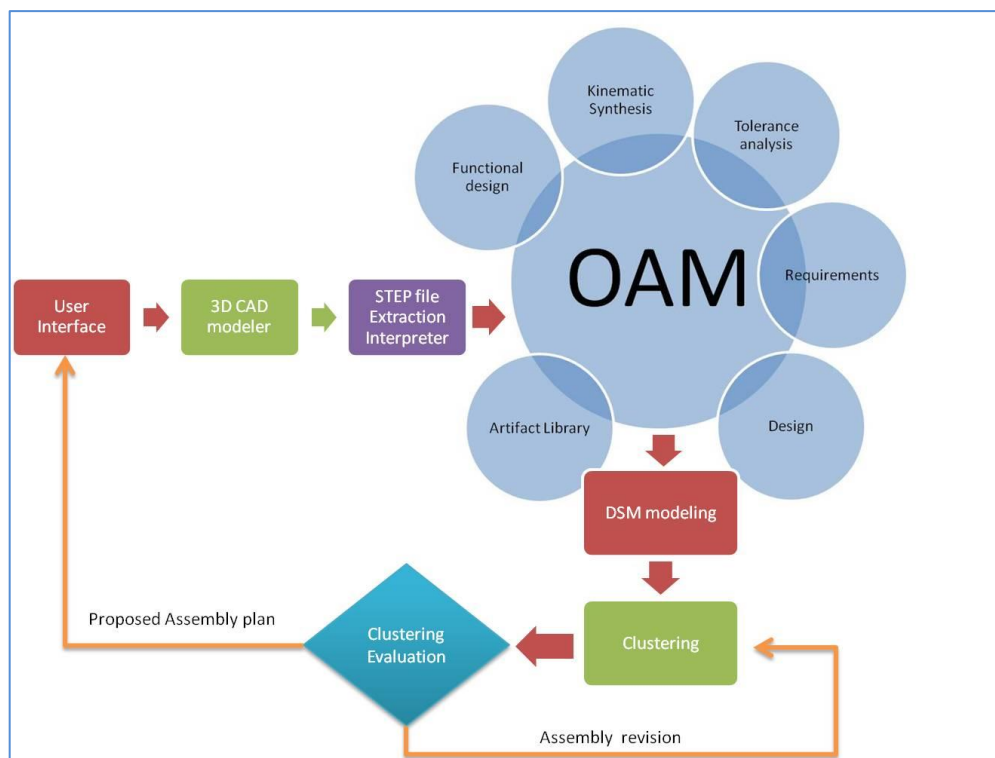
Η δημιουργία ομάδων και ακολουθιών συναρμολόγησης είναι ένα δύσκολο πρόβλημα συνδυαστικής βελτιστοποίησης [103, 104]. Οι περισσότερες υπάρχουσες μεθοδολογίες βασίζονται στη θεωρία γραφημάτων και περιλαμβάνουν σύνθετες αναλύσεις. Για παράδειγμα, μία συνηθισμένη προσέγγιση χρησιμοποιεί έναν αλγόριθμο ο οποίος εκτελεί μια εξαντλητική αναζήτηση για το γράφημα διασύνδεσης ενός προϊόντος [105]. Ωστόσο τέτοιες προσεγγίσεις απαιτούν σημαντική υπολογιστική ισχύ και πόρους ακόμη και για ένα προϊόν με ελάχιστα εξαρτήματα. Γενικά όλες αυτές οι τεχνικές αναζήτησης που επιλέγουν το καλύτερο σχέδιο συναρμολόγησης συγκρίνοντας έγκυρες λύσεις είναι μη ρεαλιστικές σε πραγματικές συνθήκες και χρονοβόρες.

Στην εικόνα 15 αποτυπώνεται σχηματικά η προτεινόμενη ενοποίηση οποία απαντά στα ζητήματα του OAM που παραμένουν ανοικτά, ενώ παράλληλα χρησιμοποιεί τον ΠΔΣ ως ένα εργαλείο που συμβάλλει με το δικό του τρόπο στο τελικό αποτέλεσμα. Ο τελικός στόχος ήταν και παραμένει η πλήρης υποστήριξη από τα αρχικά στάδια της σχεδίασης και ανάπτυξης ενός προϊόντος και για όλη τη διάρκεια της ζωής του. Πιο συγκεκριμένα:

1. Ξεκινώντας από τη φάση της σχεδίασης σε ένα λογισμικό 3D-CAD γίνεται σχεδίαση των συστατικών και των πιθανών υποσυναρμολογήσεων ενός προϊόντος. Απαραίτητη προϋπόθεση είναι το λογισμικό σχεδίασης να υποστηρίξει έξοδο σε αρχεία STEP (ISO 10303/AP214), κάτι που είναι διαθέσιμο σε όλες τις σύγχρονες πλατφόρμες σχεδίασης.
2. Στη συνέχεια το αρχείο συναρμολόγησης αναλύεται από ένα διερμηνέα (interpreter) ο οποίος βρίσκει:
 - Τα συστατικά της συναρμολόγησης (BOM).
 - Την ιεραρχία συναρμολόγησης.
 - Τα χαρακτηριστικά (features) συναρμολόγησης.
3. Τα δεδομένα από την ανάλυση αυτή (βήμα 2) ενημερώνουν τη βάση δεδομένων (Artifact Library DataBase) η οποία ουσιαστικά μοντελοποιεί το OAM. Εφαρμόζοντας τη μεθοδολογία των Baysal et.al [64] που περιγράφουν επακριβώς ποιες πληροφορίες πρέπει να εξαχθούν από τα αρχεία STEP και πως αυτές θα χρησιμοποιηθούν από το

ΟΑΜ. Συγκεκριμένα, για την υλοποίηση του ΟΑΜ θα πρέπει να ολοκληρωθούν τα παρακάτω εννέα βήματα:

1. Εξαγωγή /καταγραφή όλων των μερών και υποσυναρμολογήσεων ενός προϊόντος.
 2. Καταγραφή όλων των περιορισμών και των ζευγών συναρμολόγησης (mates / constraints) με τα χαρακτηριστικά της κάθε συναρμολόγησης (assembly features).
 3. Ομαδοποίηση όλων των χαρακτηριστικών συναρμολόγησης (assembly features) και καθορισμός των σχέσεων των αντικειμένων (artifact associations) με τις απαραίτητες ιδιότητες (attributes) για κάθε ένα.
 4. Με βάση την ομαδοποίηση του προηγούμενου βήματος, καθορισμός των σχέσεων των χαρακτηριστικών (assembly feature associations).
 5. Για κάθε σχέση των χαρακτηριστικών συναρμολόγησης καθορισμός των:
 - Γεωμετρικών περιορισμών (parametric assembly constraints).
 - Των ζευγών κινηματικής (kinematic pairs).
 - Των μονοπατιών κινηματικής (kinematic paths).
 6. Δημιουργία του πίνακα σχέσεων αντικειμένων (artifact associations).
 - Καθορισμός του τύπου σύνδεσης (connection type).
 - Καθορισμός του προσανατολισμού και της θέσης κάθε αντικειμένου (Position / Orientation).
 - Καθορισμός της σχετικής κίνησης.
 7. Δημιουργία των σχέσεων συναρμολόγησης (assembly associations).
 8. Καταγραφή / Εξαγωγή των γεωμετρικών ανοχών (geometric tolerances).
 9. Δημιουργία της κλάσης των χαρακτηριστικών της συναρμολόγησης (ΟΑΜ Feature class).
-
4. Από τη βάση δεδομένων του παραπάνω βήματος (βήμα 3) γίνεται η μοντελοποίηση του ΠΔΣ αρχιτεκτονικής (συστατικών) ο οποίος αποτυπώνει τις συναρμολογήσεις μεταξύ των συστατικών ενός προϊόντος.
 5. Κάνοντας χρήση ενός ευρετικού αλγόριθμου ομαδοποίησης, γίνεται ομαδοποίηση των συναρμολογήσεων σε αριθμό ομάδων που καθορίζει ο χρήστης ανάλογα με τις πραγματικές δυνατότητες που υπάρχουν (πόσες ομάδες συναρμολόγησης μπορεί να υπάρξουν κ.λπ.).
 6. Αξιολογώντας τις ομαδοποιήσεις που δίνει το μοντέλο προτείνεται, είτε ένα πιθανό πλάνο συναρμολόγησης (αριθμός ομάδων συναρμολόγησης, πιθανή συναρμολόγηση), είτε αναθεωρείται η αρχική σχεδίαση.



Εικόνα 15: Σχηματική αναπαράσταση της ενοποίησης των μοντέλων OAM και DSM.

Στις ενότητες που ακολουθούν αναλύεται ο τρόπος υλοποίησης όλων των επιμέρους προβλημάτων που πρέπει να επιλυθούν.

3.3.2 Το μοντέλο ενοποίησης OAM2DSM

Το προτεινόμενο μοντέλο ενοποίησης διαιρείται σε μια σειρά από επιμέρους υποπροβλήματα που αναλύονται στη συνέχεια. Συγκεκριμένα:

1. Από τα βασικά δομικά χαρακτηριστικά για την ενοποίηση των μοντέλων είναι η σωστή αποτύπωση των κλάσεων και των σχέσεων του OAM. Όπως αναφέρθηκε και στην περιγραφή του OAM, πρόκειται για ένα σύνολο οντοτήτων και σχέσεων, επεκτείνοντας και εξειδικεύοντας τις κλάσεις που ήδη έχουν περιγραφεί στο CPM/NIST. Έτσι λοιπόν η ανάλυση και σχεδίαση μιας σχεσιακής βάσης δεδομένων που να αναπαριστά με λειτουργικό τρόπο το μοντέλο OAM αποτελεί την πρώτη προτεραιότητα.
2. Στο επόμενο στάδιο είναι η σχεδίαση ενός διερμηνέα (interpreter) ο οποίος θα εξάγει από τα αρχεία STEP όλες τις πληροφορίες που πρέπει να αποθηκευτούν στη σχεσιακή βάση δεδομένων που αναφέρθηκε προηγουμένως.
3. Τέλος θα πρέπει να οριστεί ένας ΠΔΣ αρχιτεκτονικής (συστατικών) βασιζόμενος στα στοιχεία του OAM, ο οποίος θα περιγράφει τις συναρμολογήσεις, τη διάρκεια και την πολυπλοκότητα των επιμέρους υποσυναρμολογήσεων.
4. Ως τελευταίο βήμα είναι η ανάπτυξη ενός ευρετικού αλγορίθμου ομαδοποίησης του ΠΔΣ αρχιτεκτονικής (συστατικών) ο οποίος θα προτείνει βέλτιστο αριθμό ομάδων συναρμολόγησης και πιθανές συναρμολογήσεις.

3.3.3. Σχεσιακή αναπαράσταση του OAM/NIST-CPM

Για τη σχεσιακή αναπαράσταση του OAM χρησιμοποιείται το μοντέλο Οντοτήτων-Σχέσεων (Entity – Relationship Model) που προτάθηκε αρχικά το 1976 από τον Chen με σκοπό να αποτυπώσει ουσιαστικά με ένα εννοιολογικό τρόπο τον πραγματικό κόσμο ως ένα σύνολο

οντοτήτων και σχέσεων [88]. Στην αρχική του έκδοση εστίαζε στην εννοιολογική και λογική αποτύπωση της βάσης δεδομένων αλλά δεν μπορούσε να αναπαραστήσει δυναμικές συμπεριφορές βάσεων δεδομένων. Σε μετέπειτα επεκτάσεις [106, 107] υιοθετήθηκαν τέτοια χαρακτηριστικά (συνάθροιση, κληρονομικότητα κ.λπ.) δίνοντάς τους τα δομικά στοιχεία που ήταν αναγκαία για την πραγματική αποτύπωση προβλημάτων βάσεων δεδομένων. Χρησιμοποιώντας τη μοντελοποίηση από την UML σε σχεσιακή βάση δεδομένων, όπως αυτή παρουσιάζεται στο άρθρο των Song et. al. γίνεται μια αποτύπωση του OAM σε μια σχεσιακή βάση δεδομένων [108].

Συγκεκριμένα, κάθε κλάση του OAM αποτυπώνεται ως ένας σχεσιακός πίνακας στη βάση δεδομένων μιας και κάθε κλάση στην UML αναπαριστά τη βασική λογική οντότητα η οποία περιγράφει και τα δεδομένα και τη συμπεριφορά κάθε δομικής μονάδας. Η κλάση δηλαδή είναι ένα πρότυπο ή αλλιώς ένα μοντέλο από το οποίο δημιουργούνται τα πραγματικά αντικείμενα κατά την εκτέλεση του προγράμματος. Η συμπεριφορά κάθε αντικειμένου αποτυπώνεται μέσα στην κλάση από τις λειτουργίες που ορίζονται για την κάθε κλάση. Οι λειτουργίες αυτές μπορεί να είναι ορατές από όλους (public), ορατές μόνο στους απογόνους τις κλάσεις (protected) ή κρυμμένες (private). Οι σχέσεις μεταξύ κλάσεων είναι ο μηχανισμός που αποτυπώνει τη χρήση ή το χειρισμό στοιχείων της μιας κλάσης από την άλλη. Οι συσχετισμοί αυτοί μπορεί να είναι λειτουργικοί (functional – do something for me) ή δομικοί (structural – be something for me).

Ιδιαίτερα για τις σχέσεις των συναθροίσεων (aggregations) αυτές εμπεριέχουν τη λογική της συλλογής μιας κλάσης αντικειμένων σε μία άλλη. Οι σχέσεις σύνθεσης (composition) από την άλλη πλευρά είναι πιο δυνατές από τις σχέσεις συνάθροισης μιας και εμπεριέχουν τη λογική της σύνθεσης ενός αντικειμένου από άλλα. Στο σχεδιασμό μιας βάσης δεδομένων για την αποτύπωση του OAM, καθοριστικό ρόλο παίζει η στρατηγική που θα ακολουθηθεί στην κληρονομικότητα των κλάσεων. Θα πρέπει να τονιστεί ότι κληρονομικότητα (inheritance) εμπεριέχει την έννοια της επαναχρησιμοποίησης και της πολυπλοκότητας. Στη συγκεκριμένη περίπτωση, για το μοντέλο OAM, αυτό εστιάζεται στις σχέσεις γενίκευσης μεταξύ των κλάσεων **artifact-Assembly** και **Artifact-Part**. Υπάρχουν τρεις τρόποι για να αποτυπωθεί η κληρονομικότητα σε ένα σχεσιακό μοντέλο [109]:

1. Κάθε κλάση ιεραρχίας έχει ένα μοναδικό πίνακα που περιέχει όλες τις κληρονομούμενες ιδιότητες για όλα τα στοιχεία. Ουσιαστικά πρόκειται για μια ένωση όλων των κλάσεων μέσα στην ιεραρχία.
2. Κάθε κλάση μέσα στην ιεραρχία έχει το δικό της αντίστοιχο πίνακα που περιέχει μόνο τις ιδιότητες που είναι προσβάσιμες από την ίδια την κλάση (περιλαμβάνοντας και τις κληρονομούμενες ιδιότητες).
3. Κάθε σχέση γενίκευσης στην ιεραρχία έχει έναν πίνακα για κάθε κλάση που περιλαμβάνει μόνο τις ιδιότητες της συγκεκριμένης κλάσης.

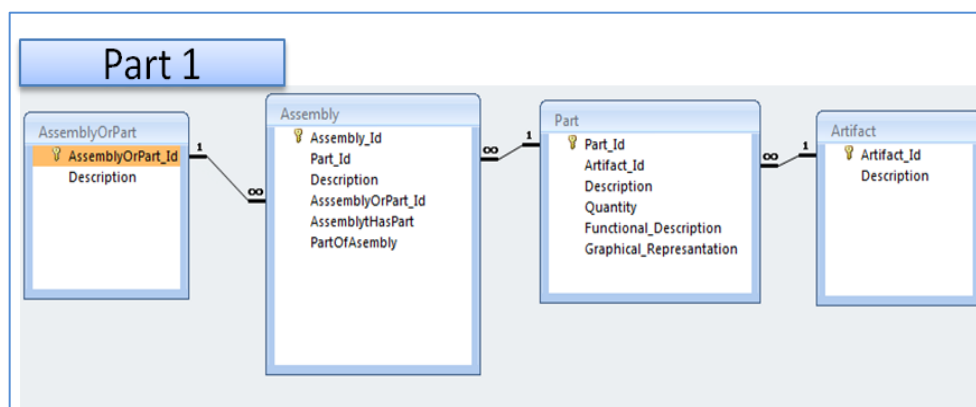
Στην πρώτη περίπτωση ουσιαστικά καταργείται η ιεραρχία. Αν και αποτελεί μια εύκολη λύση για τη διαχείριση της βάσης δεδομένων, παρουσιάζει μεγάλη πολυπλοκότητα στον ορισμό των εξαρτήσεων μεταξύ των κλάσεων. Παράλληλα, η οποιαδήποτε αλλαγή μιας κλάσης στην ιεραρχία απαιτεί πολύ χρόνο στον επανασχεδιασμό του σχεσιακού πίνακα. Στη δεύτερη περίπτωση, αν και είναι μια καλύτερη προσέγγιση από την πρώτη, η αλλαγή στοιχείων σε μια γενικευμένη κλάση μέσα στην ιεραρχία απαιτεί ενημέρωση όλων των κληρονομούμενων ιδιοτήτων στις ιεραρχικά από κάτω κλάσεις (child classes).

Η τρίτη προσέγγιση αποτελεί την καλύτερη στρατηγική στο χειρισμό της κληρονομικότητας μιας και οι αλλαγές σε μία κλάση μέσα στην ιεραρχία απαιτούν τοπικές αλλαγές μόνο μέσα στην ίδια την κλάση, χωρίς να υπάρχει αντίκτυπος στις υπόλοιπες. Τέλος, για την αποτύπωση πολύπλοκων ιδιοτήτων κλάσεων που αναπαριστούν άλλες κλάσεις, ακολουθείται η στρατηγική της μοντελοποίησης με χρήση συσχετισμών μεταξύ πινάκων. Μια συσχέτιση είναι στην πραγματικότητα μια δομική σχέση μεταξύ των πινάκων – αντικειμένων. Για παράδειγμα, ο διαχωρισμός δύο στοιχείων μεταξύ assembly ή part μπορεί να γίνει με ένα επιπλέον πίνακα που διαχωρίζει αν ένα στοιχείο είναι υποσυναρμολόγηση (subassembly) ή μέρος του προϊόντος (part). Σχεδιαστικά αυτό υλοποιείται με τη χρήση ενός εξωτερικού κλειδιού (FK – foreign key) από τον ένα πίνακα στον άλλο.

Στη συνέχεια και υιοθετώντας όλα τα παραπάνω, περιγράφεται η μοντελοποίηση του OAM σε ένα σχεσιακό μοντέλο. Οι συμβολισμοί PK και FK μπροστά από κάποια πεδία υποδηλώνουν τα κλειδιά που χρησιμοποιούνται μεταξύ των πινάκων για τους απαραίτητους συσχετισμούς. Να σημειωθεί ότι το PK (Primary Key) υποδηλώνει τη μοναδικότητα κάθε εγγραφής (γραμμής) μέσα σε έναν πίνακα ενώ το FK δηλώνει το συσχετισμό μεταξύ δύο πινάκων – αντικειμένων:

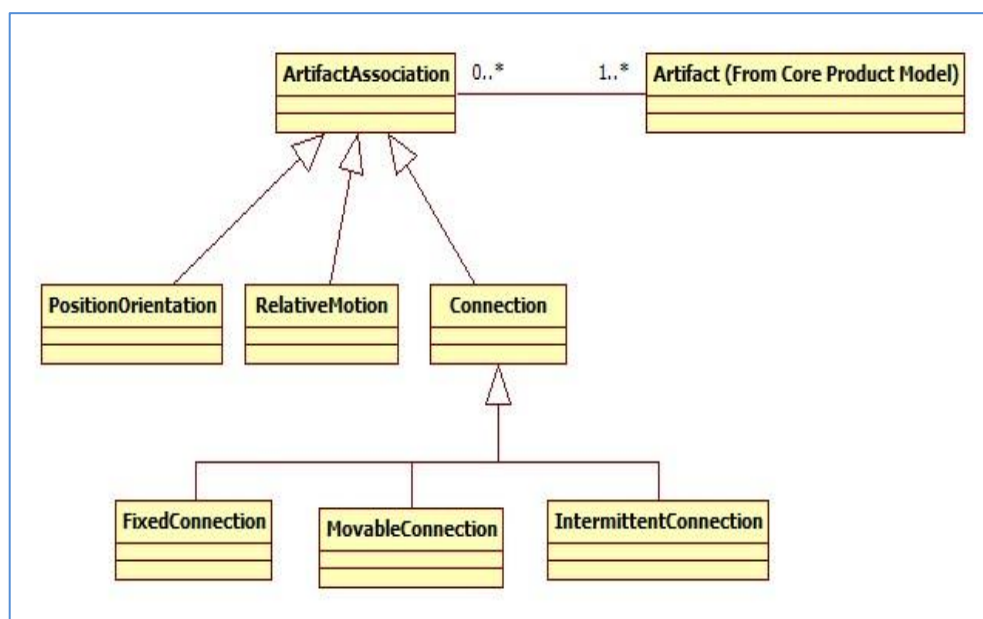
Table – Object	Attributes	Data type	Related Table
Artifact	PK:Id	Number	
	Description	Text	
Part	PK:Id	Number	
	FK:Artifact_Id	Number	Artifact
	Description	Text	
	Quantity	Number	
	Functional_Description	Text	
	Graphical_Representation	Image Object	
Assembly	PK:Id	Number	
	FK:Part_Id	Number	Part
	Description	Text	
	FK:AssemblyOrPart	Number	AssemblyOrPart
	AssemblyHasPart	Number	
	PartOfAssembly	Number	
AssemblyOrPart	PK:Id	Number	
	Description	Text	

Πίνακας 6: Πίνακας δομών δεδομένων για τις κλάσεις Artifact, Part και Assembly



Εικόνα 16: Μέρος 1- Βάση δεδομένων για τους παραπάνω πίνακες.

Οι κλάσεις που αναπαριστούν τις σχέσεις συναρμολόγησης, όπως φαίνονται στην εικόνα 16, αποτυπώνονται στον πίνακα 6 και στην εικόνα 17.

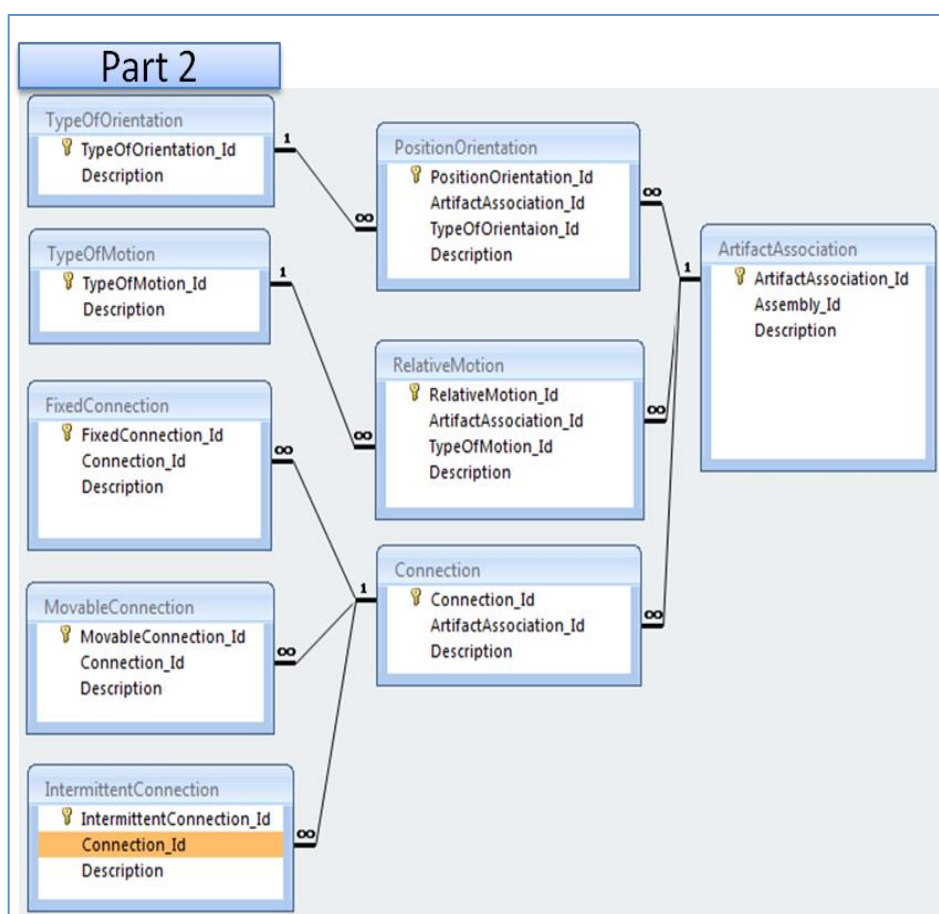


Εικόνα 17: Κλάσεις συσχετισμών του OAM (97)

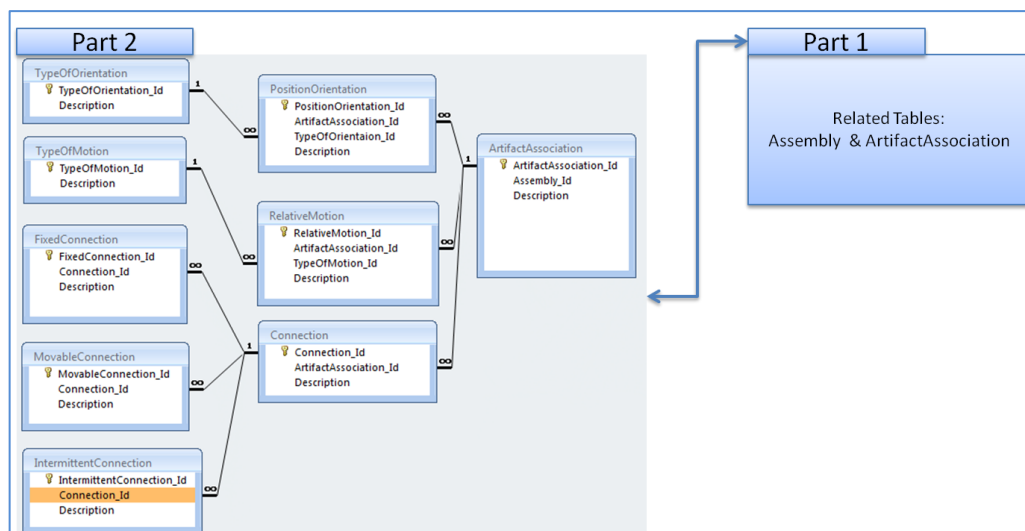
Table – Object	Attributes	Data type	Related Table
ArtifactAssociation	PK:Id	Number	
	FK:Assembly_Id	Number	Assembly
	Description	Text	
PositionOrientation	PK:Id	Number	
	FK:ArtifactAssociation_Id	Number	ArtifactAssociation
	FK:TypeofOrientation	Number	TypeOfOrientation
	Description	Text	
TypeofOrientation	PK:Id	Number	
	Description	text	
RelativeMotion	PK:Id	Number	
	FK:Artifact_Association_Id	Number	ArtifactAssociation
	FK:TypeofMotion	Number	TypeOfMotion
	Description	Text	
TypeOfMotion	PK:Id	Number	
	Description	Text	
Connection	PK:Id	Number	
	FK:Artifact_Association_Id	Number	ArtifactAssociation
	Description	Text	
FixedConnection	PK:Id	Number	
	FK:Connection_Id	Number	Connection
	Description	Text	
MovableConnection	PK:Id	Number	
	FK:Connection_Id	Number	Connection
	Description	Text	

IntermittentConnection	PK:Id	Number	
	FK:Connection_Id	Number	Connection
	Description	Text	
ArtifactAssociation	PK:Id	Number	
	FK:Assembly_Id	Number	Assembly
	Description	Text	

Πίνακας7: Πίνακας δομών δεδομένων για τις κλάσεις Artifact Association, Position Orientation, Relative Motion, Connection, Fixed Connection, Movable Connection, Intermittent Connection.

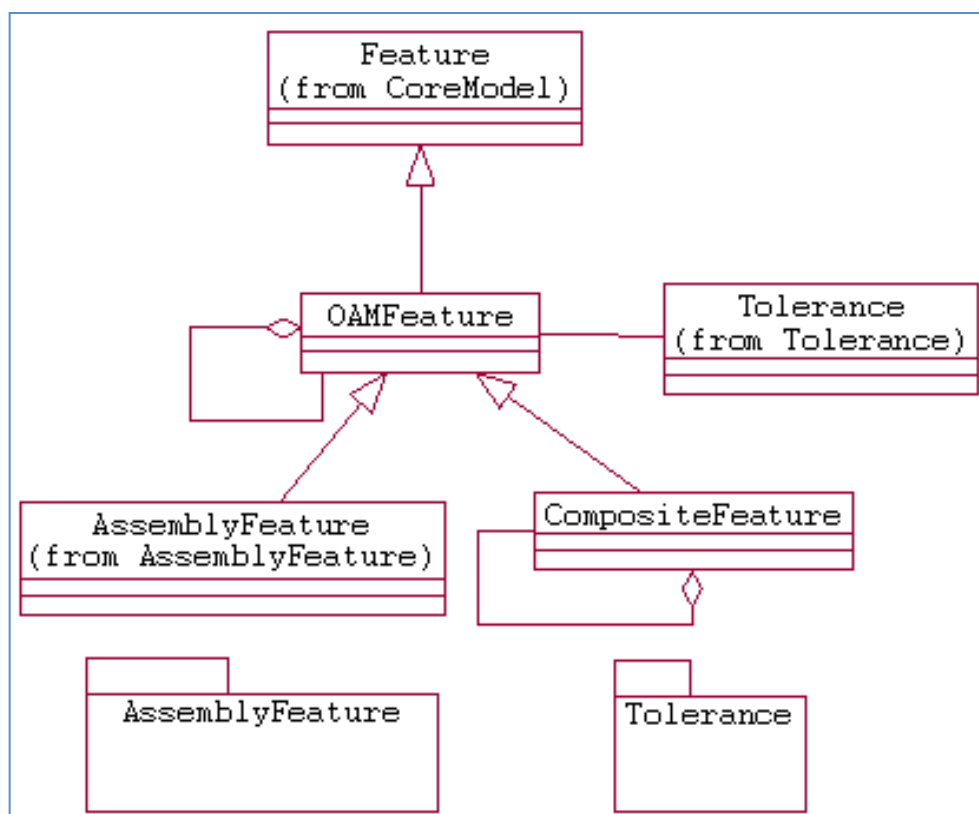


Εικόνα 18: Μέρος 2-Η βάση δεδομένων για την Artifact Association κλάση.

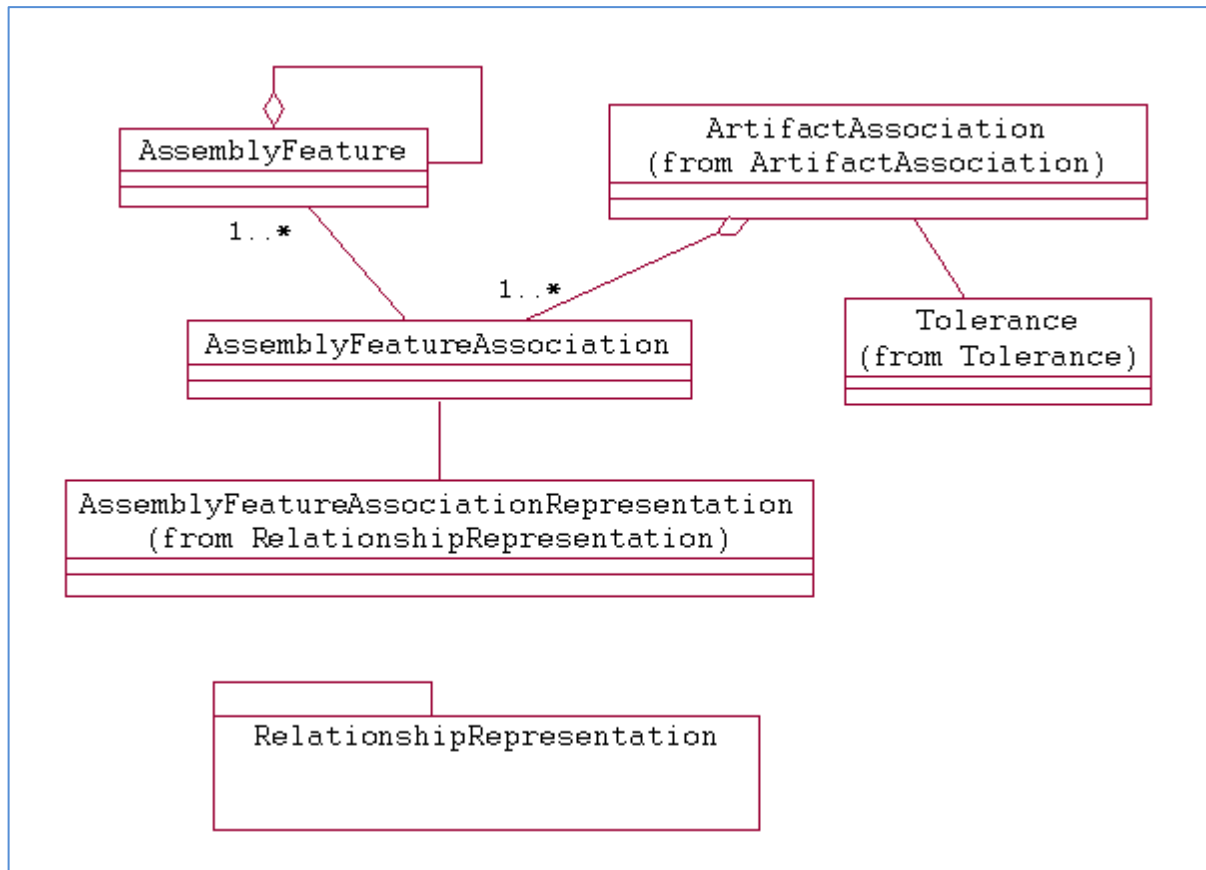


Εικόνα 19: Σύνδεση των τμημάτων της βάσης δεδομένων.

Στη συνέχεια περιγράφονται οι κλάσεις που αφορούν τα χαρακτηριστικά (OAMFeatures) της συναρμολόγησης. Η κλάση OAMFeature κληρονομεί τα χαρακτηριστικά της κλάσης Feature, όπως αυτή ορίζεται στο NIST-CPM. Η κλάση περιέχει πληροφορίες για τις ανοχές (κλάση Tolerance) καθώς επίσης και για τα χαρακτηριστικά της συναρμολόγησης. Στον πίνακα 7 περιγράφονται οι οντότητες που αποτυπώνουν αυτό το τμήμα του OAM.



Εικόνα 20: Κλάσεις για την αποτύπωση των χαρακτηριστικών συναρμολόγησης και των ανοχών [63].

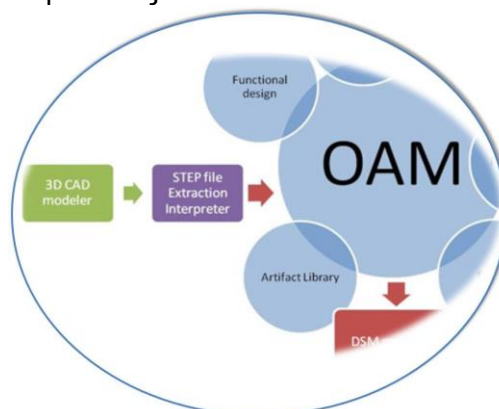


Εικόνα 21: Η κλάση Assembly Feature [63].

3.3.4. Λεξικογραφική ανάλυση των αρχείων STEP– Εξαγωγή δεδομένων

Εξαγωγή BOM, Ιεραρχίας και ποσοτήτων

Η λεξικογραφική ανάλυση για την εξαγωγή πληροφορίας από τα αρχεία STEP βασίζεται στην οντότητα *Next_Assembly_Usage_Occurence*, όπως αυτή περιγράφεται στη γλώσσα EXPRESS η οποία αποτελεί το ISO 10303-AP203. Η οντότητα αυτή ουσιαστικά συσχετίζει κάθε συναρμολόγηση με τα στοιχεία που αποτελείται, ενώ παράλληλα δηλώνει μια νέα συναρμολόγηση ή νέο συστατικό του προϊόντος.



Εικόνα 22: Μεγέθυνση πάνω στο μοντέλο OAM2DSM που αποτυπώνει την εξαγωγή στοιχείων από το STEP αρχείο στη βάση δεδομένων.

Στο AP203 δηλώνεται ως:

```
ENTITY next_assembly_usage_occurrence
SUBTYPE OF (assembly_component_usage);
END_ENTITY;
```

Στο [πίνακα 7] φαίνονται οι ιδιότητες (τοπικές και κληρονομούμενες) της οντότητας που περιγράφουμε.

Ιδιότητα	Τύπος	Ορίζεται από...
id	identifier(String)	product_definition_relationship
name	label (String)	product_definition_relationship
description	text (String)	product_definition_relationship
relating_product_definition	product_definition (ENTITY)	product_definition_relationship
related_product_definition	product_definition (ENTITY)	product_definition_relationship
reference_designator	identifier(String)	assembly_component_usage

Πίνακας 7: Ιδιότητες της Next_Assembly_Usage_Occurrence οντότητας

Για τις ανάγκες της λεξικογραφικής ανάλυσης, στον [πίνακα 8] φαίνεται ο συμβολισμός που ακολουθείται και η περιγραφή κάθε ιδιότητας.

Συμβολισμός	Ορισμός
NAUO_Entity	Next_Assembly_Usage_Occurrence ENTITY
CNAUO_Entity	Current_Next_Assembly_Usage_Occurrence_Entity
RPD1	Relating_Product_Definition1
RPD2	Relating_Product_Definition2

Πίνακας 8: Συμβολισμός των ιδιοτήτων της παραπάνω οντότητας για τις ανάγκες της λεξικογραφικής ανάλυσης

Ο αλγόριθμος που ακολουθεί στην εικόνα 23 διαβάζει ένα αρχείο STEP το οποίο έχει εξαχθεί από ένα λογισμικό σχεδίασης και στη συνέχεια παράγει και εξάγει όλα τα στοιχεία που είναι χρήσιμα και αφορούν τη συναρμολόγηση.

Algorithm STEP_Parser**Input** :A STEP file exported from 3D CAD modeling system**Output**: BOM, Parts & Assembly Hierarchy, part quantities, part features.

Begin

Open STEP file for READ

While Not EOF(StepFile) Do

Search_for_next_NAUO_Entity()

If Find_Next_NAUO_Entity = TRUE

/*Extract from line text the Id,RPD1,RPD2 and put them in CNAUO_Entity

CNAUO_Entity ← (Id, RPD1, RPD2)

Find_Father_Flag ← False

While Not EOF(StepFile)) and (not Find_Father_Flag) do

/*Search for each NAUO_Entity_iGet_Next_NAUO_Entity_i()If CNAUO_Entity.RPD1 = NAUO_Entity_i.RPD2 Then

Find_Father_Flag = True

Exit While

End if

End While

If Find_Father_Flag = True Then

MakeATreeNode(NAUO_Entity_i.Id As Father, CNAUO_Entity.Id As Child)

Else

MakeATreeNode("Root" As Father, CNAUO_Entity.Id As Child)

End if

End If

End While

End

Εικόνα 23: Αλγόριθμος ανάγνωσης αρχείου STEP

Στο **Παράρτημα III** υπάρχει η τεκμηρίωση του παραπάνω αλγόριθμου καθώς και ο ίδιος ο αλγόριθμος σε εκτελέσιμο κώδικα.

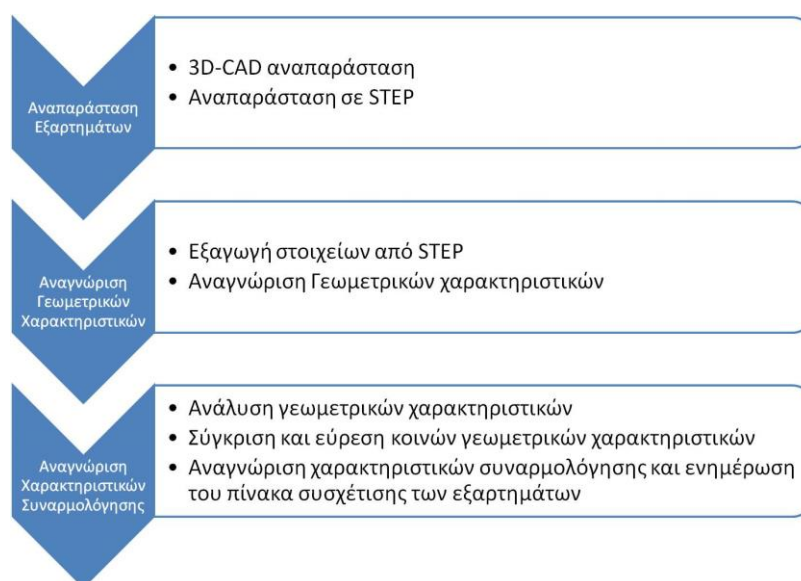
Εξαγωγή χαρακτηριστικών συναρμολόγησης

Η Εξαγωγή των χαρακτηριστικών συναρμολόγησης γίνεται κάνοντας χρήση των συστατικών του STEP αρχείου όπως αυτό ορίζεται στο πρωτόκολλο εφαρμογής STEP - AP203. Η διαδικασία της αναγνώρισης των χαρακτηριστικών οποιουδήποτε μοντέλου, η ανάλυση των χαρακτηριστικών αυτών και η σωστή ερμηνεία του μοντέλου είναι γνωστή ως εξαγωγή χαρακτηριστικών [110]. Τα χαρακτηριστικά συναρμολόγησης έχουν χρησιμοποιηθεί από αρκετούς ερευνητές με στόχο να αυξήσουν την αποδοτικότητα και την ευελιξία της διαδικασίας συναρμολόγησης ενός προϊόντος αν και υπάρχουν σημαντικές διαφορές ως προς τον ορισμό του χαρακτηριστικού συναρμολόγησης [111]. Στη βιβλιογραφία το χαρακτηριστικό συναρμολόγησης ορίζεται με διαφορετικούς τρόπους όπως:

- Ως την εξειδικευμένη πληροφορία συναρμολόγησης μέσα στη μοντελοποίηση και στη σχεδίαση [112].
- Ως μια ιδιότητα της κάθε συναρμολογούμενης μονάδας σε σχέση με άλλα στοιχεία συναρμολόγησης η οποία παρέχει πληροφορίες σχετικά με τη σχεδίαση την κατασκευή και τη συναρμολόγηση [113].

- Ως ιδιαίτερα γεωμετρικά χαρακτηριστικά της μορφής του εξαρτήματος που επηρεάζουν τις εργασίες συναρμολόγησης τα οποία ορίζονται από τις μεταξύ τους συνδέσεις [114].
- Ως τη βασική πληροφορία σύνδεσης που περιέχεται μεταξύ των συσχετιζόμενων εξαρτημάτων [115].
- Ως τη σύνδεση δυο γεωμετρικών χαρακτηριστικών των υπό συναρμολόγηση εξαρτημάτων [113].

Η διαδικασία για την αναγνώριση χαρακτηριστικών συναρμολόγησης αποτελείται από τρία στάδια: αναπαράσταση εξαρτημάτων, αναγνώριση γεωμετρικών χαρακτηριστικών και τέλος αναγνώριση χαρακτηριστικών συναρμολόγησης. Για την παρούσα εργασία εκτός από την εύρεση των χαρακτηριστικών συναρμολόγησης, ενδιαφέρον παρουσιάζει και το κόστος συναρμολόγησης για το οποίο γίνεται εκτενής αναφορά σε επόμενη ενότητα. Στην εικόνα 24 φαίνεται σχηματικά το διάγραμμα ροής της προτεινόμενης μεθόδου



Εικόνα 24: Λογικό διάγραμμα της προτεινόμενης μεθόδου εύρεσης χαρακτηριστικών συναρμολόγησης.

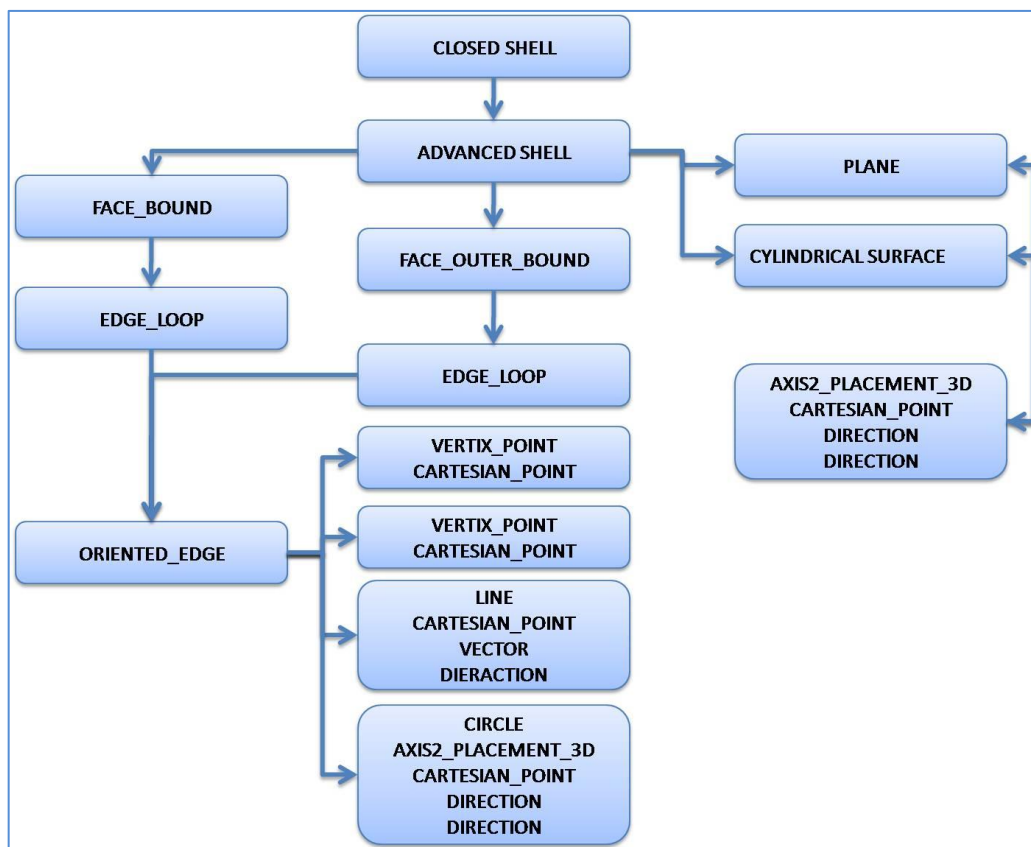
Ένα χαρακτηριστικό συναρμολόγησης ορίζεται ως ένα ή περισσότερα κοινά ζεύγη γεωμετρικών χαρακτηριστικών που σχετίζονται με συναρμολόγηση [111]. Ένα χαρακτηριστικό συναρμολόγησης μπορεί να αποτυπωθεί μαθηματικά ως

$$AF = (F_i U F_j) \cup R_{ij} \quad [1]$$

Όπου AF δείχνει ένα χαρακτηριστικό συναρμολόγησης, F_i και F_j αναπαριστούν τα ζεύγη των γεωμετρικών χαρακτηριστικών που ταιριάζουν στη συναρμολόγηση και R_{ij} δείχνει τη σχέση συναρμολόγησης μεταξύ των χαρακτηριστικών F_i και F_j .

Στην παρακάτω εικόνα αποτυπώνεται η δομή του αρχείου STEP σε ότι αφορά τη γεωμετρική περιγραφή των εξαρτημάτων ενός προϊόντος. Η εξαγωγή της πληροφορίας των χαρακτηριστικών από το αρχείο STEP είναι η εξαγωγή γεωμετρικών και τοπολογικών πληροφοριών των σχεδιαζόμενων εξαρτημάτων καθώς και η σχέση μεταξύ τους (χαρακτηριστικά συναρμολόγησης). Η δομή δεδομένων του αρχείου STEP είναι μια ιεραρχική δομή με τη μορφή δένδρου που αποτελείται από το επίπεδα οντότητα, ακμή και σημείο. Ένα STEP αρχείο αποτε-

λείται από δύο τμήματα. Το πρώτο τμήμα που περιλαμβάνει διαχειριστικές πληροφορίες όπως για παράδειγμα το όνομα του δημιουργού την ημερομηνία και άλλα εισαγωγικά και το τμήμα δεδομένων που αρχίζει με τη λέξη κλειδί “DATA” και τελειώνει με τη λέξη κλειδί “ENDSEC” που περιέχει τη γεωμετρική αποτύπωση των εξαρτημάτων ενός προϊόντος [116]. Κάθε γραμμή μέσα σε αυτό το τμήμα είναι μια οντότητα που αρχίζει με ένα σειριακό αριθμό ακολουθεί το όνομα της και στη συνέχεια οι ιδιότητες της που συνήθως κάνουν αναφορά σε άλλες οντότητες.



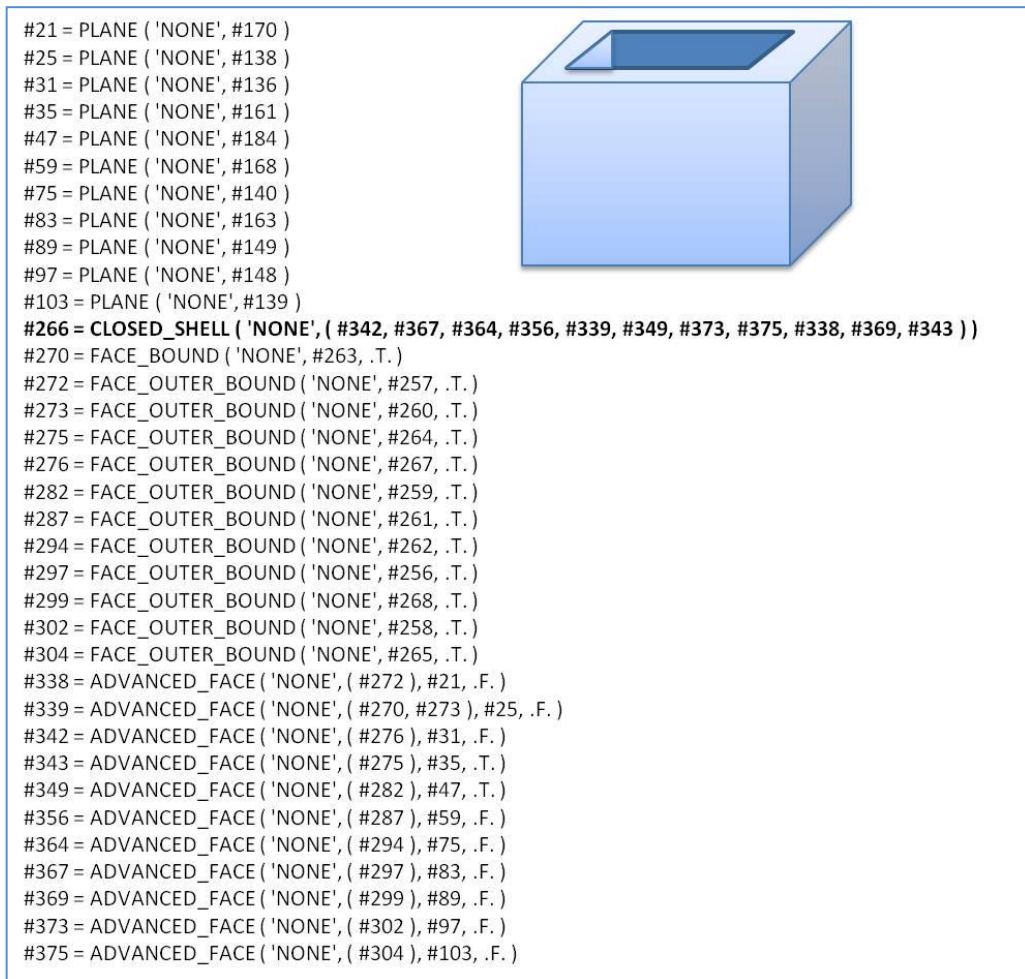
Εικόνα25: Η δομή του STEP– AP203.

Από τις πιο σημαντικές οντότητες μέσα στο αρχείο STEP αποτελεί η οντότητα CLOSED_SHELL η οποία καθορίζει όλες τις επιφάνειες οι οποίες σχηματίζουν το εξάρτημα. Στην εικόνα 26 η οντότητα #266 καθορίζει όλες τις επιφάνειες του κύβου που φαίνεται στην εικόνα ο οποίος έχει έντεκα επίπεδες επιφάνειες οι οποίες αναφέρονται στις οντότητες μέσα από την έμμεση αναφορά που γίνεται από τις ιδιότητες της οντότητας. Για την εξαγωγή χαρακτηριστικών συναρμολόγησης, που αποτελεί μέρος της παρούσας εργασίας, η διαδικασία περιγράφεται στα παρακάτω βήματα.

1. Αναγνώριση γεωμετρικών χαρακτηριστικών (Εύρεση όλων των οντοτήτων CLOSED_SHELL)
2. Ανάλυση των γεωμετρικών στοιχείων κάθε εξαρτήματος και εύρεση κοινών επιφανειών, γραμμών ή σημείων μεταξύ των εξαρτημάτων.
3. Για κάθε κοινό γεωμετρικό στοιχείο που βρίσκεται μέσα στο αρχείο STEP ορίζεται το αντίστοιχο χαρακτηριστικό συναρμολόγησης.

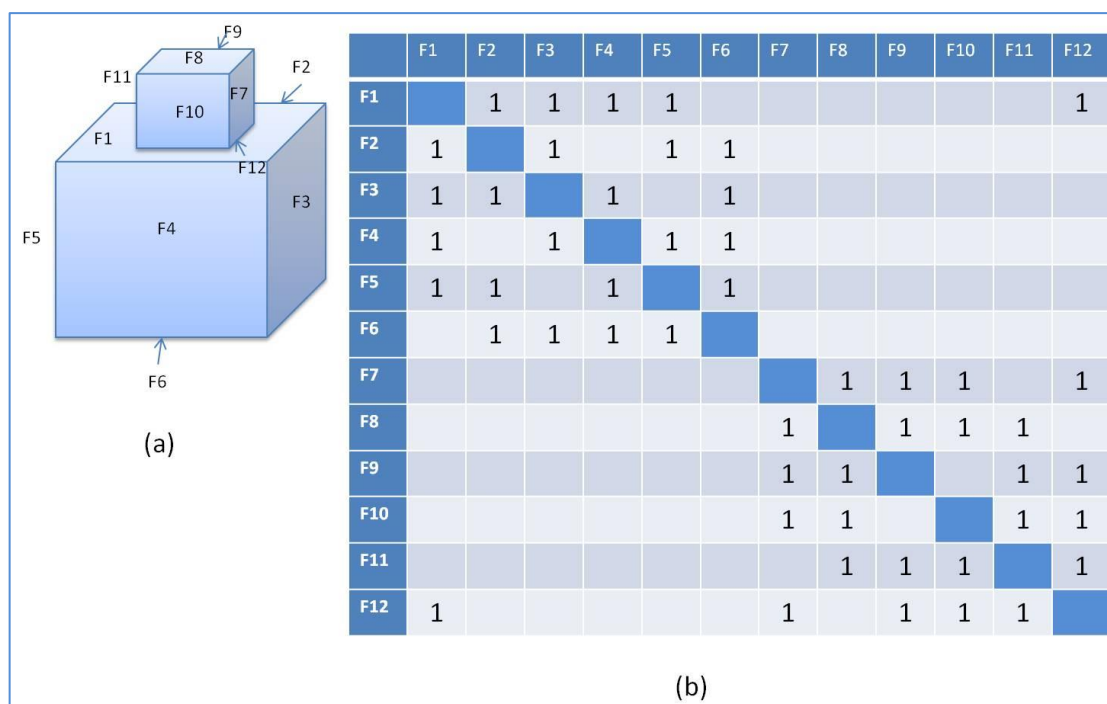
Αυτό που ουσιαστικά επιδιώκεται με τον παραπάνω αλγόριθμο είναι να βρεθούν κοινά σημεία επαφής (επιφάνειες, γραμμές, σημεία) μεταξύ δύο ή περισσότερων εξαρτημάτων. Η συ-

σχέτιση αυτή μπορεί να αποτυπωθεί ως ένας πίνακας με γραμμές και στήλες που κάθε στοιχείο του πίνακα αποτελεί ένα χαρακτηριστικό συναρμολόγησης μεταξύ των εξαρτημάτων.



Εικόνα 26: Εξαγωγή δεδομένων από το αρχείο STEPAP203 για τον κύβο με εσωτερική διαμόρφωση.

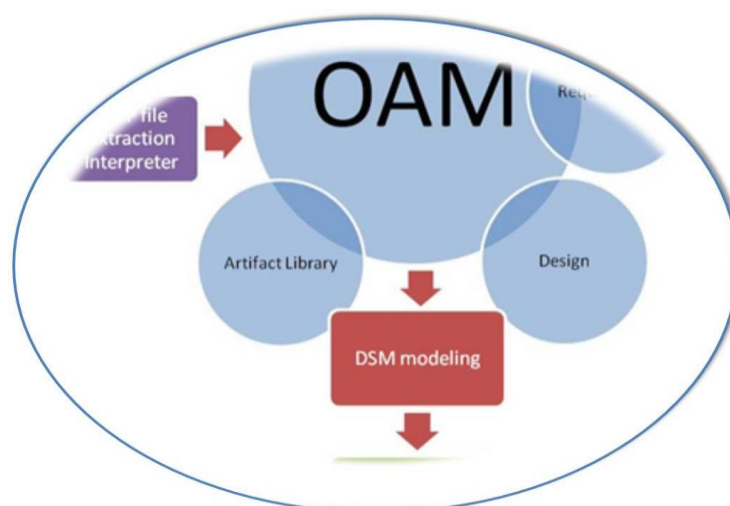
Στην εικόνα 27-α φαίνονται δύο κύβοι που έχουν μια κοινή επιφάνεια καθώς και οι αριθμημένες επιφάνειες τους. Στην εικόνα 27-β φαίνεται ο πίνακας χαρακτηριστικών συναρμολόγησης που προκύπτει από την εφαρμογή του αλγορίθμου που ήδη περιγράφηκε.



Εικόνα 27: Δύο κύβοι που εφάπτονται (a) και ο πίνακας χαρακτηριστικών συναρμολόγησης τους (b).

3.3.5. Μοντελοποίηση του Πίνακα Δομής Σχεδίασης για το OAM2DSM

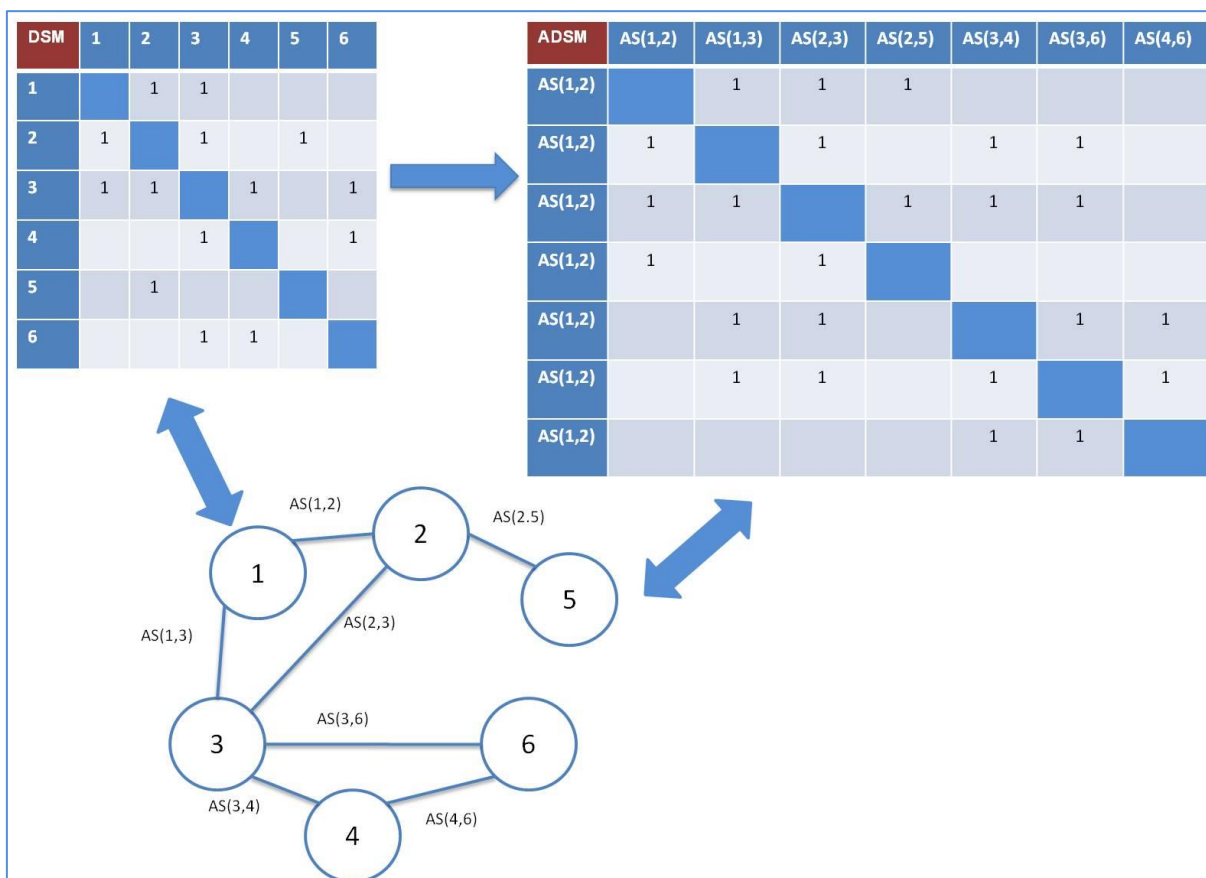
Η μοντελοποίηση του ΠΔΣ περιλαμβάνει το μετασχηματισμό του αρχικού ΠΔΣ (DSM) συστατικών ο οποίος αποτυπώνει σχέσεις και εξαρτήσεις μεταξύ των στοιχείων του πίνακα σε ένα νέο ΠΔΣ (ADSM) ο οποίος ως γραμμές και στήλες έχει τις εξαρτήσεις του αρχικού ΠΔΣ.



Εικόνα 28: Μεγέθυνση πάνω στο μοντέλο OAM2DSM που αποτυπώνει τη μοντελοποίηση του ΠΔΣ η οποία περιγράφεται στην παρούσα ενότητα.

Ο αρχικός ΠΔΣ, προκύπτει από τη βάση δεδομένων των τεχνουργημάτων, η οποία υλοποιεί το μοντέλο OAM που ενημερώνεται από το αρχείο STEP του λογισμικού σχεδίασης, μεταξύ των οποίων υπάρχουν και τα συστατικά της συναρμολόγησης (BOM). Στη συνέχεια, ορίζονται δύο μετρικές (δείκτες) για κάθε στοιχείο του νέου πίνακα, βάσει των οποίων ένας επιμεριστικός αλγόριθμος ομαδοποίησης, πραγματοποιεί ομαδοποίηση των στοιχείων του νέου πίνακα (ADSM).

Η διαδικασία μετασχηματισμού του αρχικού πίνακα μπορεί να περιγραφεί με ένα παράδειγμα με τη βοήθεια της εικόνας 29.



Εικόνα 29: Μετατροπή ενός ΠΔΣ(DSM) σε ένα ΠΔΣ Συναρμολογήσεων (ADSM). Ο γράφος αποτυπώνει τον αρχικό ΠΔΣ ενώ οι ακμές του αποτυπώνουν τη συναρμολόγηση (Π.χ As(1,2) - συναρμολόγηση του εξαρτήματος 1 με το εξάρτημα 2)

Στην εικόνα 29 φαίνεται ένας αρχικός ΠΔΣ συστατικών που έχει εξαχθεί από το αρχείο STEP. Στη συνέχεια ο πίνακας αυτός μετασχηματίζεται σε ένας νέο ΠΔΣ συναρμολογήσεων (ADSM) ο οποίος περιλαμβάνει τις σχέσεις και εξαρτήσεις μεταξύ των συναρμολογήσεων του αρχικού πίνακα. Ο ενδιαμέσος γράφος αποτυπώνει με γραφικό τρόπο αυτόν το μετασχηματισμό. Ως κόμβοι είναι τα αρχικά στοιχεία του ΠΔΣ και ως ακμές οι συναρμολογήσεις των στοιχείων. Στο νέο ΠΔΣ(ADSM) οι συναρμολογήσεις έχουν γίνει γραμμές και στήλες ενώ τα στοιχεία του νέου ΠΔΣ αποτυπώνουν τη συσχέτιση κάθε συναρμολόγησης με οποιαδήποτε άλλη. Θα πρέπει να τονιστεί ότι στο μοντέλο που προτείνεται, τόσο ο αρχικός ΠΔΣ όσο και ο μετασχηματισμένος ΠΔΣ, είναι συμμετρικοί μιας και δεν εξετάζεται η ιεραρχία των συστατικών των πινάκων. Αυτό γίνεται διότι η ιεραρχία της συναρμολόγησης είναι γνωστή από το αρχείο STEP και έχει εξαχθεί στη βάση δεδομένων των τεχνουργημάτων που υλοποιεί το μοντέλο OAM. Τέλος, σε ότι αφορά το μέγεθος του νέου πίνακα θα πρέπει να σημειωθεί ότι αν ο αρχικός πίνακας είναι διαστάσεων $n \times n$ ο νέος πίνακας στη χειρότερη περίπτωση θα έχει διαστάσεις $[n(n-2)/2] \times [n(n-2)/2]$. Η πρόταση αυτή αποδεικνύεται στο παρακάτω λήμμα.

Λήμμα : Για κάθε τετραγωνικό πίνακα $A = [a_{ij}]_{n \times n}$. Τα στοιχεία του πάνω από τη διαγώνιο (χωρίς τη διαγώνιο) μπορεί να είναι το πολύ $n*(n-2)/2$.

Απόδειξη : Έστω ο τετραγωνικός πίνακας $A = [a_{ij}]_{n \times n}$. Το πλήθος των στοιχείων του είναι n^2 . Ο αριθμός των στοιχείων πάνω από τη διαγώνιο είναι το σύνολο των στοιχείων του αφαιρώντας τα στοιχεία κάτω από τη διαγώνιο (δηλαδή $n^2/2$) καθώς και τα στοιχεία της διαγωνίου. Επομένως:

$$n^2 - \frac{n^2}{2} - n = \frac{n(n-2)}{2} \quad [2]$$

Ο ΠΔΣ συστατικών μετασχηματίζεται σε ένα μη κατευθυνόμενο γράφο με $X = \{x_1, \dots, x_n\}$ σημεία δεδομένων όπου A είναι ο πίνακας ομοιότητας μέσα στον οποίο κάθε $a_{ij} > 0$ δείχνει την ομοιότητα ανάμεσα στα σημεία δεδομένων x_i και x_j . Αυτή η θεώρηση μπορεί να παρασταθεί με ένα γράφο στον οποίο κάθε κορυφή δηλώνει ένα εξάρτημα ή μια υποσυναρμολόγηση ενός προϊόντος και κάθε ακμή του γράφου δηλώνει τη σχέση μεταξύ των κορυφών. Σε μια τέτοια θεώρηση το πρόβλημα ομαδοποίησης μετασχηματίζεται στην εύρεση υπογράφων μέσα στον αρχικό γράφο έτσι ώστε οι ακμές μέσα σε κάθε υπογράφο να έχουν υψηλή σχέση συνδεσιμότητας ανάμεσα τους και οι κορυφές που ανήκουν σε διαφορετικούς υπογράφους να έχουν όσο γίνεται μικρότερη σχέση συνδεσιμότητας [117].

Κάθε γράφος μπορεί να θεωρηθεί ως μια τριάδα $G=(V,E,W)$, όπου $V = \{v_1, \dots, v_n\}$ είναι το σύνολο των κορυφών του όπου για την παρούσα εργασία δηλώνουν τα εξαρτήματα και τις υποσυναρμολογήσεις ενός προϊόντος. Το $E \subseteq V \times V$ είναι το σύνολο όλων των ακμών όπου $W = [w_{i,j}]_{n \times n}$ είναι ένας $n \times n$ τετραγωνικός πίνακας που κάθε στοιχείο του αποτυπώνει μια θετική ή μηδενική τιμή ως βάρος (συσχέτιση) μεταξύ των στοιχείων i και j μιας συναρμολόγησης.

3.3.6. Υπολογισμός κόστους συναρμολόγησης

Υπάρχουν αρκετοί τρόποι να υπολογιστεί το κόστος κάθε υποσυναρμολόγησης και στη συνέχεια αθροιστικά το συνολικό κόστος. Στην παρούσα εργασία θα υιοθετηθεί η μέθοδος υπολογισμού του κόστους συναρμολόγησης τροποποιώντας τον τελικό τύπο υπολογισμού του βάρους κάθε συναρμολόγησης [118]. Συνοπτικά η μέθοδος κάνει χρήση τεσσάρων κατηγοριών - δεικτών πληροφορίας που σχετίζεται με τη συναρμολόγηση:

- **Το τύπο επαφής(CT).** Ο τύπος επαφής περιγράφει τον τρόπο με τον οποίο δυο εξαρτήματα ή υποσυναρμολογήσεις συνδέονται μεταξύ τους. Πολλά σημεία επαφής δείχνουν ισχυρό δέσιμο μεταξύ των εξαρτημάτων.
- Τον **τύπο συνδυασμών(CB)** που μπορούν να υπάρχουν. Η πληροφορία αυτή περιγράφει ουσιαστικά το βαθμό σύνδεσης μεταξύ δυο εξαρτημάτων (π.χ βάλε, εισήγαγε, στρέψε κ.λ.π.).
- Τον **τύπο του εργαλείου(TT)** που χρειάζεται για τη συναρμολόγηση. Το χαρακτηριστικό αυτό εξετάζει κυρίως τη δυσκολία στο χειρισμό του εργαλείου συναρμολόγησης. Διαφορετικοί τύποι περιλαμβάνουν χειροκίνητη λειτουργία, μικρά εργαλεία, μεγάλα εργαλεία κ.λ.π. Παίρνοντας υψηλή τιμή αυτός ο δείκτης σημαίνει ότι είναι δύσκολο να συναρμολογηθούν τα δύο εξαρτήματα.
- Την **κατεύθυνση συναρμολόγησης(AD).** Υψηλή τιμή σε αυτό το δείκτη σημαίνει ότι συναρμολογούνται εύκολα διότι δεν εξαρτώνται από άλλα εξαρτήματα.

Στον πίνακα 9 αποτυπώνονται τα κριτήρια για το κόστος καθώς και οι τιμές που μπορεί να πάρουν.

Δείκτης	Ιδιότητα	Τιμή
Τύπος Επαφής (CT)	Σημείο	6
	Γραμμή	12
	Επιφάνεια	18
	Πολλαπλά Σημεία	24
	Πολλαπλές Επιφάνειες	30
Τύπος Συνδυασμού(CB)	Τοποθέτηση	4
	Εισαγωγή	8
	Περιστροφή	12
	Ισχυρή Συναρμολόγηση	16
	Μόνιμη Συναρμολόγηση	20
Τύπος Εργαλείου(TT)	Με το χέρι	7
	Με κατσαβίδι	14
	Μικρό εργαλείο	21
	Ειδικό εργαλείο	28
	Μεγάλο εργαλείο	35
Κατεύθυνση Πρόσβασης(AD)	5 Γωνίες	3
	4 Γωνίες	6
	3 Γωνίες	9
	2 Γωνίες	12
	1 Γωνίες	15

Πίνακας 9: Πίνακας κριτηρίων κόστους συναρμολόγησης (118).

Κάνοντας χρήση των παραπάνω ιδιοτήτων στη συναρμολόγηση ορίζεται το κόστος κάθε συναρμολόγησης i ως :

$$W_i = \lambda_1 CT_i + \lambda_2 CB_i + \lambda_3 TT_i + \lambda_4 AD_i \quad [3]$$

Όπου $\sum_{i=1}^4 \lambda_i = 1$, λ_i Πραγματικοί αριθμοί.

Προσδιορίζοντας τη σειρά σημαντικότητας κάθε παραμέτρου από τις τέσσερεις, μπορεί να προσδιοριστεί πως θα διαμορφωθούν τα τελικά κόστη σε κάθε συναρμολόγηση. Για παράδειγμα αν θεωρηθεί σε σειρά σημαντικότητας ο τύπος επαφής, τον τύπο του εργαλείου, την κατεύθυνση πρόσβασης και τέλος τον τύπο συνδυασμού, οι παράμετροι λ_i θα μπορούσαν να διαμορφωθούν ως εξής: $\lambda_1 = 40\%$, $\lambda_2 = 10\%$, $\lambda_3 = 30\%$, $\lambda_4 = 20\%$. Άρα το W_i αποτελεί επιλογή του χρήστη πως θα διαμορφωθεί δίνοντας του τη δυνατότητα να επιλέξει τη σειρά σημαντικότητας των κριτηρίων.

3.3.7. Ομαδοποίηση στοιχείων του DSM

Ο αλγόριθμος ομαδοποίησης που παρουσιάζεται στην παρούσα εργασία είναι επιμεριστικός διαχωρίζοντας τα στοιχεία ενός Πίνακα δομής Σχεδίασης Συστατικών σε ένα προκαθορισμένο αριθμό από ομάδες. Βασική διαφοροποίηση από τους υπάρχοντες αλγορίθμους είναι ότι δεν ομαδοποιεί τα στοιχεία του πίνακα, τα οποία αποτελούν για τους ΠΔΣ συστατικών τα μέλη μιας συναρμολόγησης, αλλά τις ίδιες τις συναρμολογήσεις και ταυτόχρονα εστιάζει πρόβλημα της ισχυρής ομαδοποίησης πράγμα το οποίο σημαίνει ότι:

- Δεν υπάρχει άδεια ομάδα
- Κάθε ακμή ανήκει σε μια και μόνο μια ομάδα.

- Η ένωση των ομάδων δίνει ξανά τον αρχικό γράφο.

Για να επιτευχθεί αυτό, ο αρχικός ΠΔΣ συστατικών μετασχηματίζεται σε έναν νέο ΠΔΣ που ως στήλες και γραμμές έχει τις σχέσεις – συναρμολογήσεις μεταξύ των στοιχείων του αρχικού πίνακα. Ο νέος αυτός ΠΔΣ για το υπόλοιπο της εργασίας ονομασθεί ΠΔΣ συναρμολογήσεων (ADSM – Assembly based Design Structure Matrix). Η ομαδοποίηση παράγει ομάδες, τα στοιχεία των οποίων είναι ομαδοποιημένα κάνοντας χρήση δύο μετρικών.

- Της αλληλεπίδρασης κάθε στοιχείου του ADSM με άλλα στοιχεία.
- Των κοινών στοιχείων που κάθε στοιχείο του ADSM έχει με τα υπόλοιπα στοιχεία.

Ως έξοδο ο αλγόριθμος θα δώσει ένα σύνολο ομάδων που κάθε συναρμολόγηση ανήκει σε μία ομάδα (Ισχυρή Ομαδοποίηση- hard clustering). Αυτή η προσέγγιση δίνει τη δυνατότητα να ελεγχθεί από τα πρώτα στάδια της σχεδίασης ποια στοιχεία του αρχικού πίνακα ADSM μπορούν να αποτελέσουν υποσυναρμολογήσεις καθώς επίσης να φανούν πιθανά λάθη στη συνολική συναρμολόγηση μειώνοντας έτσι το χρόνο σχεδίασης. Παράλληλα, επειδή η λύση αποτυπώνεται και με γραφικό τρόπο ως ένα σύνολο υπογράφων είναι εύκολο για τους σχεδιαστές να εντοπίσουν ποια στοιχεία του αρχικού ΠΔΣ συστατικών αποτελούν τους συνδετικούς κρίκους μεταξύ των υποσυναρμολογήσεων. Για να μοντελοποιηθεί ο ADSM θα πρέπει να οριστεί επακριβώς τι είναι ένας ΠΔΣ και πώς αναπαριστά την πληροφορία.

Ένας ΠΔΣ συστατικών είναι ένας τετραγωνικός $n \times n$ όπου n δηλώνει τον αριθμό των υποσυναρμολογήσεων και εξαρτημάτων σε ένα προϊόν. Επίσημα, ένας ΠΔΣ είναι ένας δυαδικός πίνακας $P = [p_{ij}]_{n \times n}$. Κάθε στοιχείο του μπορεί να είναι 0 ή 1 αποτυπώνοντας έτσι την εξάρτηση των στοιχείων του. Για κάθε στοιχείο του ΠΔΣ ορίζεται ότι:

$$p_{i,j} = \begin{cases} 0 & \text{αν } p_i \text{ δεν συνδέεται με το } p_j \\ 1 & \text{αν } p_j \text{ συνδέεται με το } p_i \end{cases} \quad [4]$$

Πάνω στη διαγώνιο ισχύει ότι: $p_{ii} = 0$. Από την άλλη αν, $p_{ij} = 1$ αυτό υποδηλώνει ότι τα στοιχεία i και j συσχετίζονται και άρα συναρμολογούνται. Για το υπόλοιπο της εργασίας κάθε συναρμολόγηση μεταξύ των στοιχείων i και j θα συμβολίζεται ως $As(i,j)$.

Από τον αρχικό πίνακα ΠΔΣ συστατικών παράγεται ο πίνακας γειννίας $D = [d_{ij}]_{n \times n}$ ο οποίος αποτυπώνει ποιες κορυφές του γράφου είναι παρακείμενες σε άλλες. Αυτός ο πίνακας είναι διάστασης $n \times n$ (όπου n ο αριθμός των εξαρτημάτων ή υποσυναρμολογήσεων όπως ορίζονται στον αρχικό ΠΔΣ συστατικών). Ο αριθμός μη κατευθυνόμενων συνδέσεων (συναρμολογήσεις) ανάμεσα στα στοιχεία του πίνακα D είναι ο συνδυασμός των n ανά 2 που είναι $m = C_2^n = [(n-1) \cdot n] / 2$.

Από τον πίνακα γειννίας εξάγεται ο ΠΔΣ συναρμολογήσεων (ADSM), $A = [a_{ij}]_{m \times m}$ που είναι ένας τετραγωνικός πίνακας διαστάσεων $m \times m$, όπου m ο αριθμός των υποσυναρμολογήσεων ($m = C_2^n$). Έχοντας ως βάση τη θεώρηση αυτή, ο πίνακας A ορίζεται ως:

$$a_{d(i,j),d(k,l)} = \begin{cases} 0 & \text{αν } (i \neq k, l) \text{ ή } (i = k \text{ and } j = l) \\ 1 & \text{αν } (i = k \text{ or } l) \text{ ή } (j = k \text{ or } l) \end{cases} \quad [5]$$

Όπου $i, j, k, l = 1, \dots, n$

Για κάθε στοιχείο κάθε γραμμής i του πίνακα ADSM ορίζεται ο δείκτης:

- **Degree of Participation(Βαθμός συμμετοχής) - (PI_i)** που δείχνει πόσες επιπλέον υποσυναρμολογήσεις ή εξαρτήματα εμπλέκονται στην i υποσυναρμολόγηση στον πίνακα A . Στην ουσία, το PI_i είναι το άθροισμα του βαθμού της εκκεντρότητας για κάθε συναρμολόγηση ανάμεσα στα στοιχεία k και l του αρχικού ΠΔΣ συστατικών. Το PI_i μπορεί να υπολογιστεί ως το άθροισμα των μη μηδενικών στοιχείων της i γραμμής στον πίνακα συναρμολογήσεων (ADSM) που είναι :

$$PI_i = \sum_{k=1}^m a_{i,k}, \quad i=1, \dots, m \quad [6]$$

Μεταξύ των στοιχείων i και j του πίνακα A ορίζουμε το δείκτη:

- **Degree of affinity(Βαθμός Συγγένειας) - ($AI_{i,j}$)** που είναι το άθροισμα όλων των κοινών συναρμολογήσεων μεταξύ των στοιχείων i και j . Το $AI_{i,j}$ υπολογίζεται ως:

$$AI_{i,j} = \sum_{k=1}^m a_{i,k} \otimes a_{j,k} \quad [7],$$

Όπου $i, j = 1, \dots, m$, και \otimes είναι ο τελεστής λογικής σύζευξης

Ο αλγόριθμος ξεκινά δίνοντας ως είσοδο τον αριθμό των ομάδων που ο χρήστης επιθυμεί, και στη συνέχεια υπολογίζεται το μέσο κόστος ανά ομάδα (που είναι το συνολικό κόστος των ομάδων διαιρεμένο με τον αριθμό των ομάδων). Στη συνέχεια υπολογίζει το δείκτη PI_i ($i=1, \dots, m$). Ο αριθμός των ομάδων είναι μια επιλογή του χρήστη, για παράδειγμα ο αριθμός των γραμμών παραγωγής στη συναρμολόγηση. Για κάθε ομάδα το στοιχείο i με τη μέγιστη τιμή στο PI_i θα είναι υποψήφιο για να δημιουργήσει τη δική του ομάδα (υποψήφιος αρχηγός ομάδας). Στη συνέχεια ο αλγόριθμος υπολογίζει το $AI_{i,j}$, $j=1, \dots, m$ για το επιλεγμένο i .

Το στοιχείο j με το μεγαλύτερο $AI_{i,j}$ επιλέγεται και όλα τα κοινά στοιχεία μεταξύ των στοιχείων i και j ομαδοποιούνται υπό τον περιορισμό ότι η τρέχουσα ομάδα δεν ξεπέρασε το μέσο κόστος ομάδας που έχει ήδη υπολογιστεί. Αν μετά την παραπάνω διαδικασία το συνολικό κόστος της ομάδας είναι μικρότερο από το μέσο κόστος, ο αλγόριθμος συνεχίζει να ομαδοποιεί παρακείμενα (διπλανά) στοιχεία μέχρις ότου το κόστος της ομάδας να φτάσει στο μέσο όρο.

Κατόπιν ο δείκτης PI υπολογίζεται ξανά για την επόμενη ομάδα. Στο τέλος ο αλγόριθμος εντοπίζει όλα εκείνα τα στοιχεία που δεν έχουν ενταχθεί ακόμα σε ομάδα και για κάθε ανένταχτο στοιχείο βρίσκει υποψήφιες ομάδες και επιλέγει αυτή με το μικρότερο τρέχον κόστος. Από τον ίδιο τον ορισμό του, είναι φανερό ότι ο αλγόριθμος δημιουργεί ισορροπημένες ομάδες ως προς το κόστος.

Ακολουθεί η περιγραφή του αλγόριθμου σε ψευδογλώσσα και στην επόμενη ενότητα γίνεται η τεκμηρίωση.

Algorithm Graph_Clustering

Input: An adjacency matrix ADSM, Number Of Clusters(N)

Output : N Clusters with balanced Time cost

begin

Compute_Cluster_Average_Cost()

Compute_Participated Index(PI_i), $i=1..M$

For each Cluster (i), $i=1,..,N$

 Find_Max_PI()

 Cluster_It ()

 Compute_Affinity_Index_ $AI_{i,j}$ between Max_ PI_i and $j=1,..,M$

 /*PASS 1*/

 Find_Max_AI

 Cluster_All_Assemblies_from_Max_AI()

 Update_Cluster_Cost()

 /*PASS 2*/

 While Cluster_Cost \leq Cluster_Average_Cost

 Cluster All assemblies that can belonged to cluster()

 Update_Cluster_Cost()

 End While

End For

/*PASS 3*/

For Each Assembly Not Cluster yet

 Find Candidate Clusters which assemblies can belonged ()

 Select Cluster(c), $c=1..N$, with min Cluster_Cost

 Cluster Assembly()

 Update_Cluster_Cost()

End for

End

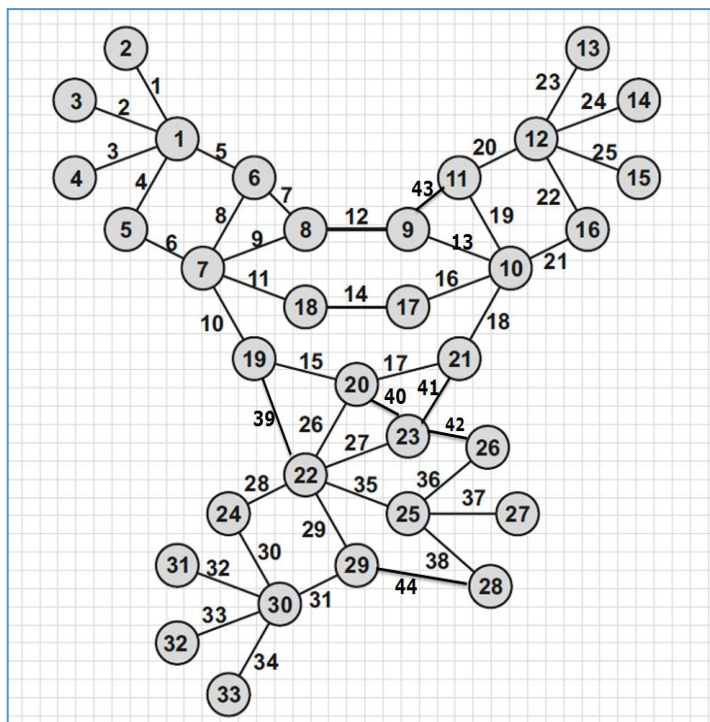
3.3.8. Μελέτη για την Αποδοτικότητα και Αξιολόγηση του αλγόριθμου Ομαδοποίησης

Στην παρούσα ενότητα γίνεται η μελέτη του αλγόριθμου ομαδοποίησης που περιγράφηκε και μοντελοποιήθηκε στην προηγούμενη ενότητα ενώ ελέγχεται η αποδοτικότητά του καθώς και η αξιοπιστία του με παραδείγματα. Για την παρουσίαση του αλγόριθμου θα χρησιμοποιηθεί ένα σχέδιο προϊόντος όπως αυτό αποτυπώνεται στον επόμενο ΠΔΣ συστατικών που ακολουθεί.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	1	1	1	1	1																												
2	1																																
3	1																																
4	1																																
5	1																																
6	1																																
7																																	
8																																	
9																																	
10																																	
11																																	
12																																	
13																																	
14																																	
15																																	
16																																	
17																																	
18																																	
19																																	
20																																	
21																																	
22																																	
23																																	
24																																	
25																																	
26																																	
27																																	
28																																	
29																																	
30																																	
31																																	
32																																	
33																																	

Εικόνα 30: Πίνακας Δομής Σχεδίασης συστατικών που αποτυπώνει στις σχέσεις μεταξύ των εξαρτημάτων του προϊόντος.

Με βάση τη μεθοδολογία που αναπτύχθηκε στην προηγούμενη ενότητα, ο ΠΔΣ αποτυπώνεται σε ένα μη κατευθυνόμενο γράφο όπως φαίνεται στην εικόνα 31. Κάθε κόμβος του γράφου αντιστοιχεί σε ένα στοιχείο του ΠΔΣ και κάθε ακμή αποτυπώνει τη σχέση μεταξύ των στοιχείων. Να σημειωθεί ότι οι αριθμοί πάνω στις ακμές του μη κατευθυνόμενου γράφου δεν είναι βάρη αλλά ένα αριθμός – αναγνωριστικό για κάθε ακμή. Στο παράδειγμα, όλες οι συναρμολογήσεις έχουν κοινό χρόνο συναρμολόγησης.



Εικόνα 31: Μη κατευθυνόμενος γράφος για το προϊόν. Οι κορυφές αναπαριστούν τα εξαρτήματα του προϊόντος ενώ οι ακμές, που για ευκολία είναι αριθμημένες, αναπαριστούν τις συνδέσεις μεταξύ τους.

Ο αντίστοιχος ΠΔΣ συναρμολόγησης (ADSM) όπως έχει οριστεί, φαίνεται στην παρακάτω εικόνα.

Weight Assembly DSM			Components DSM			Assembly DSM			Participated Index(PI)			Clusters / Costs															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
▶	1	1		1	1	1																					
	2	1	1		1	1																					
	3	1	1	1		1	1																				
	4	1	1	1	1		1	1																			
	5	1	1	1	1	1		1	1																		
	6				1		1		1	1	1	1															
	7				1		1	1		1		1															
	8				1	1	1	1		1	1	1															
	9				1	1	1	1	1	1	1	1															
	10					1		1	1	1	1	1				1											
	11						1	1	1	1	1	1															
	12							1		1		1	1			1											
	13											1	1				1		1	1		1					
	14													1			1										
	15										1					1		1									1
	16													1	1		1										
	17															1		1		1	1		1				1
	18													1			1	1	1	1		1					
	19												1				1		1	1	1		1				
	20																		1	1	1		1	1	1	1	

Εικόνα 32: Ο ΠΔΣ συναρμολόγησης (ADSM) που εξάχθηκε από τον πίνακα γειτνίασης (αρχικό ΠΔΣ συστατικών).

Ο πίνακας αυτός (ADSM) μαζί με τον επιλεγόμενο αριθμό ομάδων (N στον αριθμό) αποτελούν την είσοδο για τον αλγόριθμο ομαδοποίησης. Στην εικόνα 33 φαίνεται ένα στιγμιότυπο του δείκτη PI. Σε κάθε γραμμή υπάρχει η συναρμολόγηση, η τιμή του PI, η πληροφορία αν έχει μπει σε ομάδα η συναρμολόγηση καθώς και το κόστος που έχει. Στη συνέχεια με γραφικό τρόπο ο αλγόριθμος χρωματίζει κάθε συναρμολόγηση εντάσσοντάς της σε μια ομάδα. Όταν ο αλγόριθμος τερματίζει παράγει ως έξοδο τις ομάδες των εξαρτημάτων, το κόστος κάθε ομάδας και τη μεγαλύτερη διαφορά κόστους που βρέθηκε μεταξύ όλων των ομάδων.

Weight Assembly DSM Components DSM Assembly DSM Participated Index(PI) Clustes / Costs							
	#	Assembly	PI	Clustered	HI Current	Cluster Number	Weight
▶	1	1-- 2	4	0	0	0	1
	2	1-- 3	4	0	0	0	1
	3	1-- 4	4	0	0	0	1
	4	1-- 5	5	0	0	0	1
	5	1-- 6	6	0	0	0	1
	6	5-- 7	5	0	0	0	1
	7	8-- 6	4	0	0	0	1
	8	6-- 7	6	0	0	0	1
	9	7-- 8	6	0	0	0	1
	10	7-- 19	6	0	0	0	1
	11	7-- 18	5	0	0	0	1
	12	8-- 9	4	0	0	0	1
	13	9-- 10	6	0	0	0	1
	14	18-- 17	2	0	0	0	1
	15	19-- 20	5	0	0	0	1
	16	17-- 10	5	0	0	0	1
	17	20-- 21	5	0	0	0	1
	18	10-- 21	6	0	0	0	1
	19	10-- 11	6	0	0	0	1
	20	11-- 12	6	0	0	0	1
	21	10-- 16	5	0	0	0	1

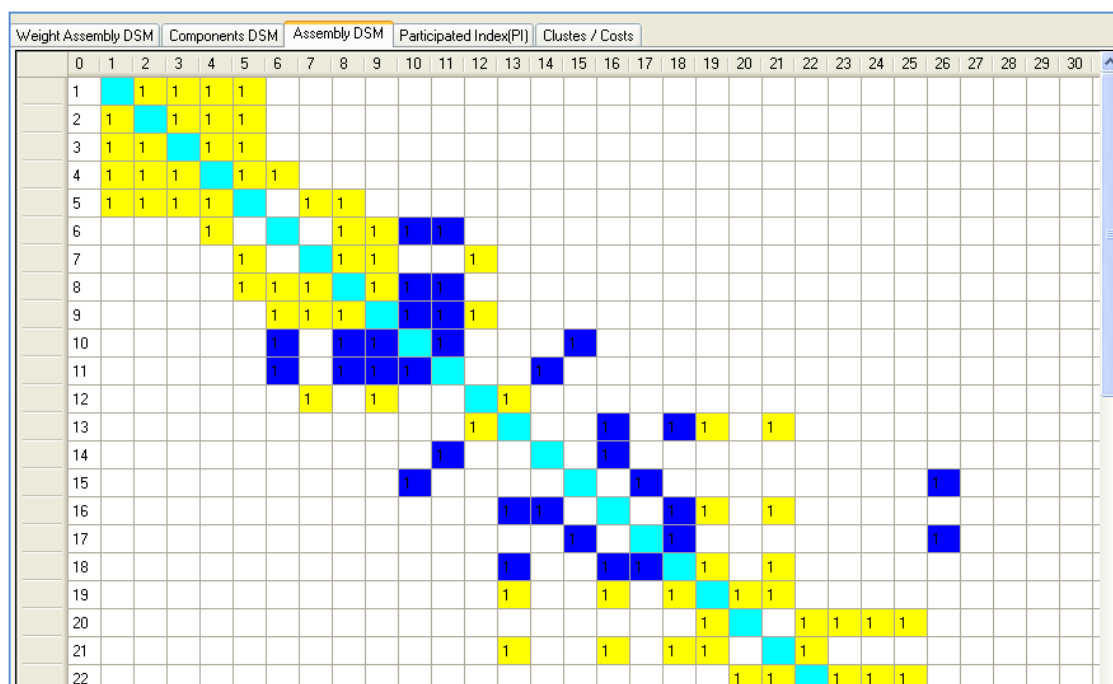
Open File

Clustering

StepByStep Clustering

☐ Cluster Number
☐ Cluster Homogeneity Limit

Εικόνα 33: Στιγμιότυπο από τον αλγόριθμο ομαδοποίησης.



Εικόνα 34: Ο ΠΔΣ των συναρμολογήσεων χρωματισμένος ανά ομάδα στοιχείων.

Weight Assembly DSM Components DSM Assembly DSM Participated Index(PI) Clustes / Costs						
#	Assembly	PI	Clustered	HI Current	Cluster Number	Weight
1	1-2	0	1	0	2	1
2	1-3	0	1	0	2	1
3	1-4	0	1	0	2	1
4	1-5	0	1	0	2	1
5	1-6	0	1	0	2	1
6	5-7	0	1	0	2	1
7	8-6	0	1	0	2	1
8	6-7	0	1	0	2	1
9	7-8	0	1	0	2	1
10	7-19	0	1	0	1	1
11	7-18	0	1	0	1	1
12	8-9	0	1	0	2	1
13	9-10	0	1	0	2	1
14	18-17	0	1	0	1	1
15	19-20	0	1	0	1	1
16	17-10	0	1	0	1	1
17	20-21	0	1	0	1	1
18	10-21	0	1	0	1	1
19	10-11	0	1	0	2	1
20	11-12	0	1	0	2	1
21	10-16	0	1	0	2	1
22	12-16	0	1	0	2	1
23	12-13	0	1	0	2	1
24	12-14	0	1	0	2	1

File Name: C:\Documents and Settings\demiriz\Επιφάνεια εργασίας\DSM
Number of Clusters :3

Cluster NO: 1 Leader: 26 Initial Cost: 1
assembly In: 28 Current Total Cost: 2
assembly In: 29 Current Total Cost: 3
assembly In: 35 Current Total Cost: 4
assembly In: 39 Current Total Cost: 5
assembly In: 40 Current Total Cost: 6
assembly In: 17 Current Total Cost: 7
assembly In: 27 Current Total Cost: 8
assembly In: 15 Current Total Cost: 9

Pass 2 - Unclustered Elements
assembly In: 10 Current Total Cost: 10
assembly In: 11 Current Total Cost: 11
assembly In: 14 Current Total Cost: 12
assembly In: 16 Current Total Cost: 13
assembly In: 18 Current Total Cost: 14
Final Cost: 14

Cluster NO: 2 Leader: 5 Initial Cost: 1
assembly In: 2 Current Total Cost: 2
assembly In: 3 Current Total Cost: 3
assembly In: 4 Current Total Cost: 4
assembly In: 7 Current Total Cost: 5
assembly In: 8 Current Total Cost: 6

Pass 2 - Unclustered Elements
assembly In: 1 Current Total Cost: 7
assembly In: 6 Current Total Cost: 8
assembly In: 9 Current Total Cost: 9
assembly In: 12 Current Total Cost: 10
assembly In: 13 Current Total Cost: 11
assembly In: 19 Current Total Cost: 12
assembly In: 20 Current Total Cost: 13
assembly In: 21 Current Total Cost: 14
Final Cost: 14

Εικόνα 35: Έξοδος από τον αλγόριθμο στον οποίο όλα τα στοιχεία είναι ομαδοποιημένα, ενώ δεξιά αποτυπώνεται η εξέλιξη ένταξης των στοιχείων στην κάθε ομάδα.

Weight Assembly DSM Components DSM Assembly DSM Participated Index(PI) Clustes / Costs			
Cluster No	Cost	Max Distance	
1	15	4	
2	19	0	
3	10	9	
*			

Clustering

Open File

Visual Graph

StepByStep Clustering

>>Run Clustering

☒ Cluster Number 3

☐ Cluster Homogeneity Limit

Label1

Label2

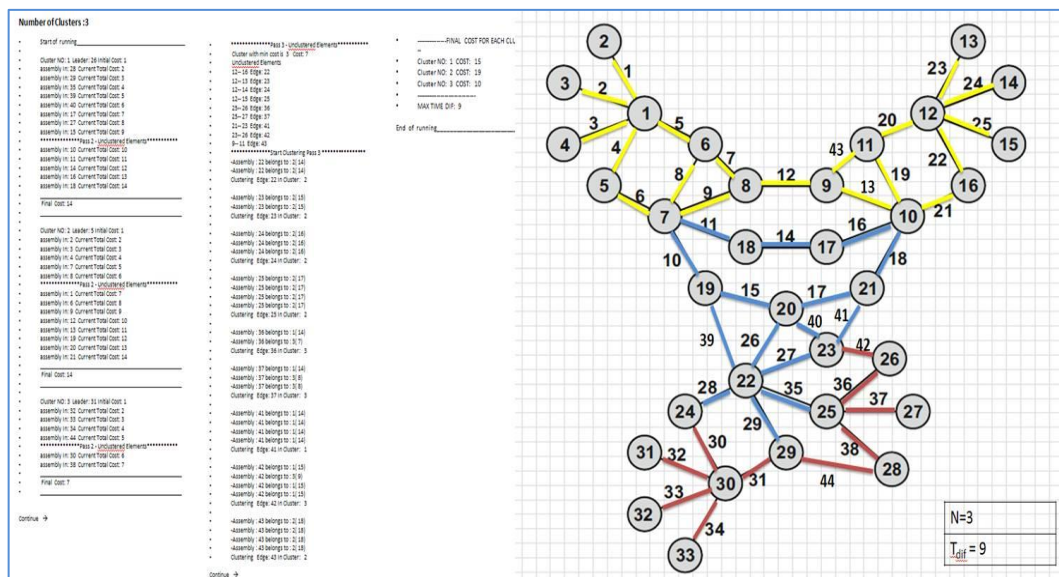
Εικόνα 36: Αποτύπωση της μεγαλύτερης διαφοράς κόστους μεταξύ των ομάδων.

Για διάφορες τιμές του $N=2,3,\dots,7$ ο αλγόριθμος δίνει τα παρακάτω αποτελέσματα. Σε κάθε διαφορετική εκδοχή αποτυπώνονται:

1. Η έξοδος από τον αλγόριθμο σε πραγματικό χρόνο.
2. Ο γράφος χρωματισμένος ανά ομάδα για καλύτερη κατανόηση των ομάδων.
3. Ο επιθυμητός αριθμός των ομάδων.

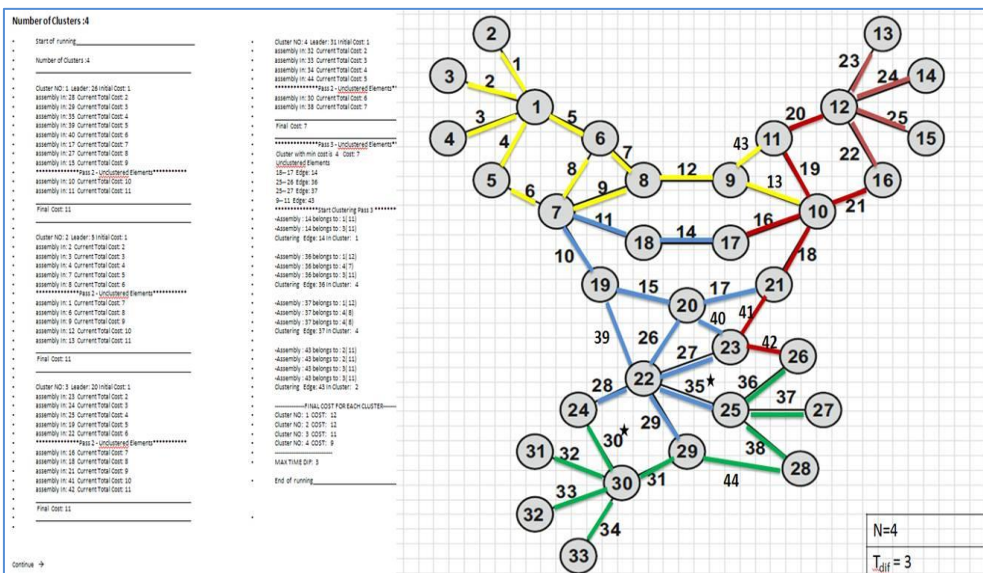
Μια παράμετρος T_{dif} η οποία κάθε φορά δείχνει τη μεγαλύτερη διαφορά που έχει το κόστος των ομάδων που δημιουργήθηκαν από τη μέση τιμή. Αυτό από μόνο του μπορεί να αποτε-

λέξει κριτήριο για την επιλογή ή μη της προτεινόμενης λύσης. Να σημειωθεί ότι η συναρμο-
λόγηση κάθε ομάδας εξαρτημάτων μπορεί να γίνεται από διαφορετική γραμμή παραγωγής
και το ισορροπημένο χρονικό κόστος μεταξύ των ομάδων αποτελεί σημαντική παράμετρο. Η
εικόνα 37 δείχνει την ομαδοποίηση για αριθμό ομάδων $N=3$.



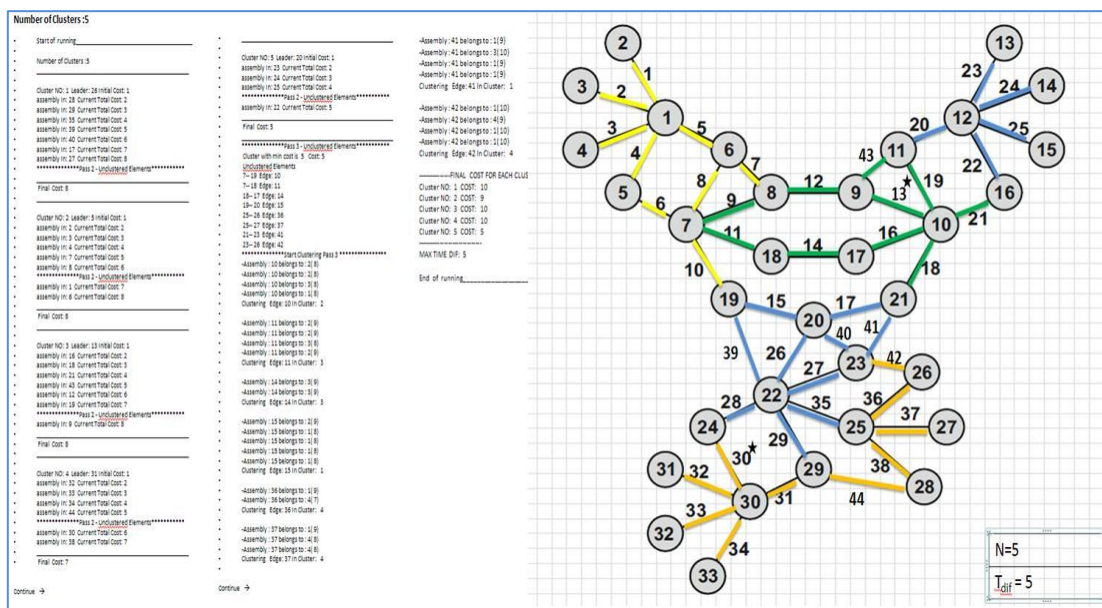
Εικόνα 37: Ομαδοποίηση για αριθμό ομάδων $N=3$

Για $N=4$



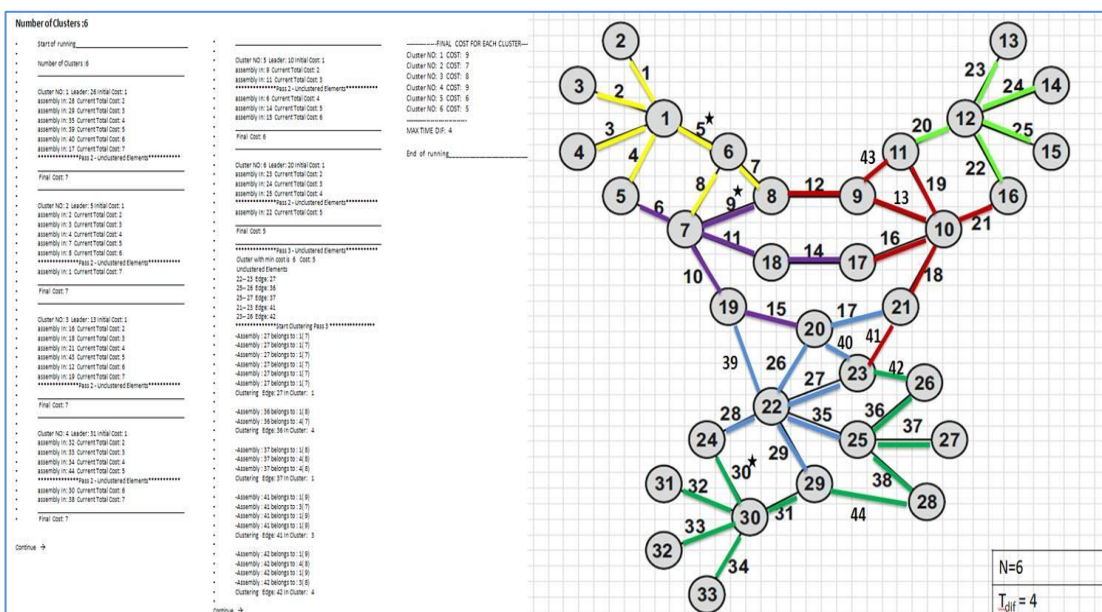
Εικόνα 38: Ομαδοποίηση για αριθμό ομάδων $N=4$

Για $N=5$



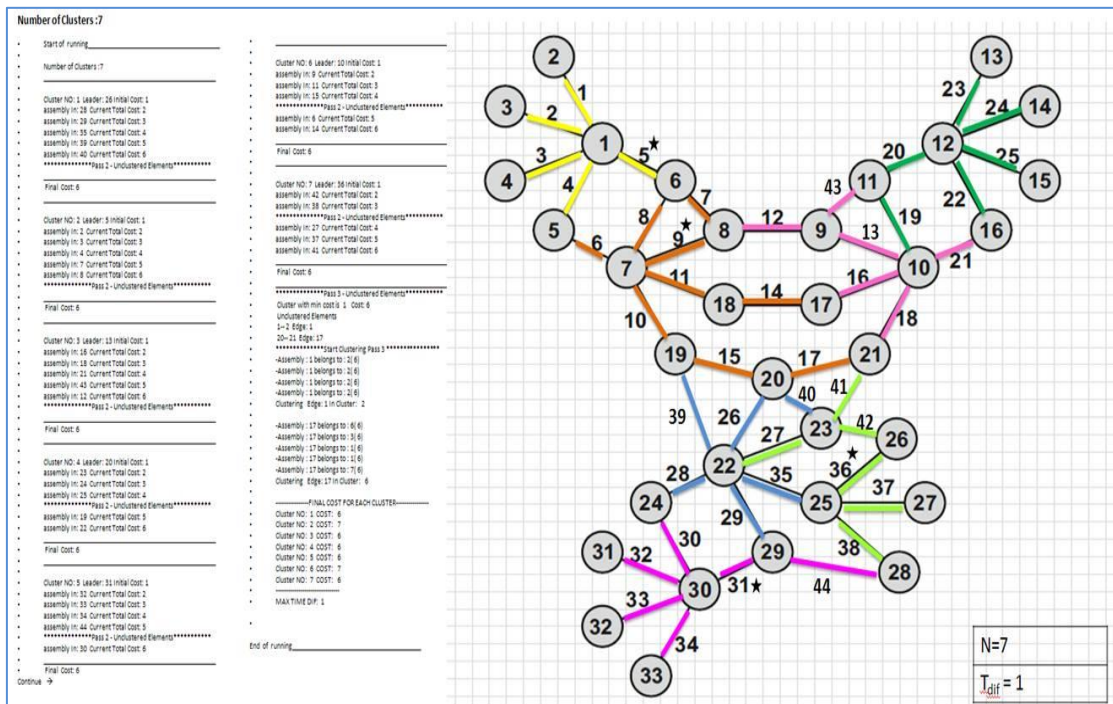
Εικόνα 39: Ομαδοποίηση για αριθμό ομάδων $N=5$

Για $N=6$



Εικόνα 40: Ομαδοποίηση για αριθμό ομάδων $N=6$

Για $N=7$



Εικόνα 41: Ομαδοποίηση για αριθμό ομάδων $N=7$

Γενικά, η ομαδοποίηση με τη μέθοδο των γραφών είναι ένα NP-Hard πρόβλημα, έτσι η προτεινόμενη μέθοδος ομαδοποίησης οδηγούν σε μία κατά προσέγγιση βέλτιστη λύση [126]. Για την αποδοτικότητα και την αξιοπιστία της μεθόδου αναφέρεται στη βιβλιογραφία μεγάλος αριθμός από δείκτες εγκυρότητας και αποδοτικότητας [78]. Για το συγκεκριμένο αλγόριθμο που παρουσιάστηκε στην προηγούμενη ενότητα θα χρησιμοποιηθεί μια τροποποιημένη εκδοχή του δείκτη του Dunn [120] που βασίζεται στην ιδέα της διαμέτρου της κάθε ομάδας που δημιουργείται και στη διαφορετικότητα της κάθε ομάδας [121]. Ο δείκτης αυτός αποτυπώνει αν οι ομάδες είναι καλά χωρισμένες και συμπαγείς.

Η ευαισθησία του δείκτη Dunn σε προβλήματα μεγάλων διαστάσεων και στο θόρυβο των δεδομένων, αντιμετωπίζεται με μείωση των διαστάσεων. Εάν μια ομαδοποίηση σε n κλάσεις περιέχει καλά διαχωρισμένες και συμπαγείς κλάσεις, η απόσταση μεταξύ κλάσεων αναμένεται να είναι μεγάλη και η ενδο-απόσταση ελάχιστη, καθώς τα κέντρα των ομάδων τοποθετούνται απομακρυσμένα στο χώρο και οι διάμετροί τους είναι περιορισμένες. Συνεπώς, μεγάλες τιμές του δείκτη Dunn αντιστοιχούν σε ομαδοποιήσεις με καλούς σχηματισμούς κλάσεων. Συγκεκριμένα, έστω E_i το σύνολο όλων των ακμών ενός γράφου που ανήκουν στην ομάδα i .

$$E_i = \{e_{ij}, 1 \leq j \leq l_i, |E_i|: 1 \leq i \leq c\} \quad [8]$$

Η διάμετρο κάθε ομάδας ορίζεται ως :

$$d_i = \max_j \{ e_{ij}, j = 1, \dots, l_i \} \quad [9]$$

Η οποία εκφράζει τη μέγιστη απόσταση μεταξύ δύο σημείων μιας συγκεκριμένης κλάσης. Και ως μέγιστη διάμετρο D την:

$$D = \max \{d_i, 1 \leq i \leq c\} \quad [10]$$

Η διαφορετικότητα μεταξύ των ομάδων X_i , και X_j ορίζεται ως :

$$d_{i,j} = \text{dist}(X_i, X_j), \text{ όπου } \text{dist}(X_i, X_j) = \|v_i - v_j\| \quad [11]$$

που είναι η απόσταση ανάμεσα στα κέντρα των ομάδων i και j , Τέλος, ο δείκτης εγκυρότητας ορίζεται ως:

$$V_d(c) = \min_{1 \leq i \leq c} \left\{ \min_{\substack{1 \leq j \leq c \\ j \neq i}} \left\{ \frac{d_{ij}}{\max_{1 \leq k \leq c} \{d_{ik}\}} \right\} \right\} \quad [12]$$

Στον πίνακα 10 γίνεται υπολογισμός του δείκτη Dunn.

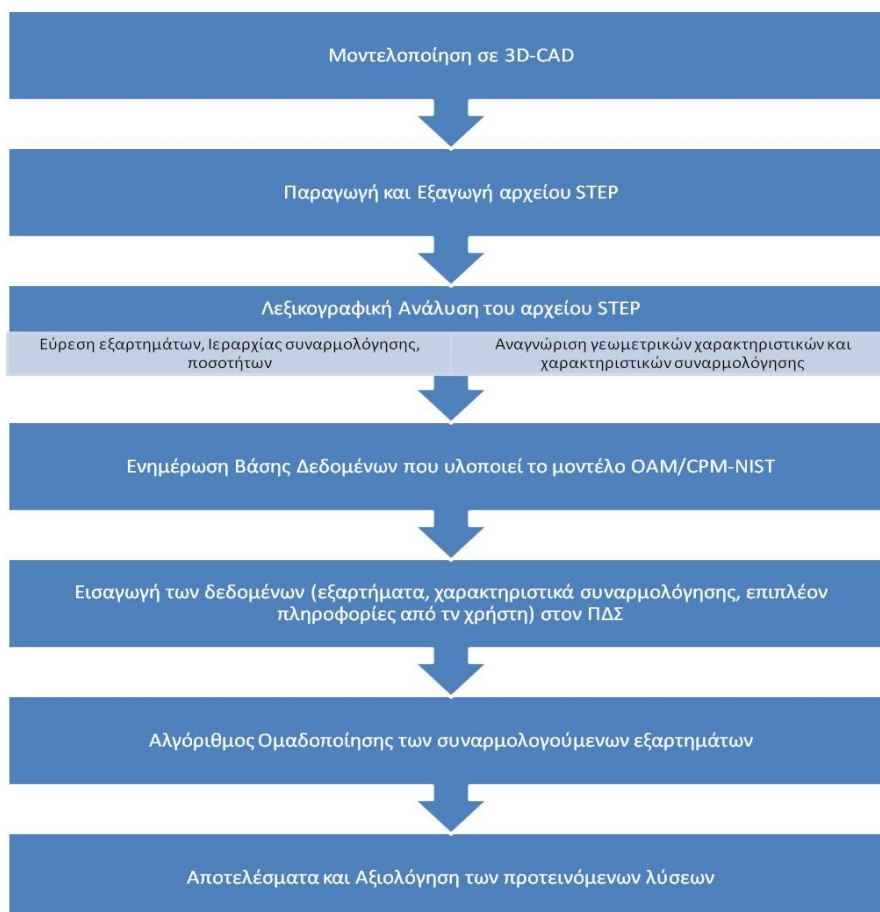
	$V_{d(1)}$	$V_{d(2)}$	$V_{d(3)}$	$V_{d(4)}$	$V_{d(5)}$	$V_{d(6)}$	$V_{d(7)}$	Min $V_d(c)$
N=2	0,666667	0,888889						0,666667
N=3	0,888889	0,500000	0,600000					0,500000
N=4	0,250000	0,200000	0,250000	0,333333				0,200000
N=5	0,256665	0,365456	0,452000	0,300000	0,423000			0,256665
N=6	0,423000	0,232000	0,500000	0,333333	0,400000	0,313333		0,232000
N=7	0,560000	0,450000	0,600000	0,423100	0,540000	0,320000	0,120000	0,120000

Πίνακας 10: Υπολογισμός του δείκτη Dunn για $N=2, \dots, 7$

Καλύτερη τιμή του δείκτη είναι για $N=2$ οπότε και η λύση αυτή προτείνεται για την παραπάνω ομαδοποίηση.

Μελέτη Περίπτωσης

Στην παρούσα ενότητα παρουσιάζεται η προτεινόμενη μεθοδολογία εφαρμοσμένη πάνω σε ένα πραγματικό προϊόν. Στόχος της μεθοδολογίας είναι η συναρμολόγηση υπολογιστών σε όσες γραμμές συναρμολόγησης αποφασίσει ο τελικός χρήστης, προτείνοντας ισορροπημένες ομάδες συναρμολόγησης ως προς το χρόνο. Το προϊόν σχεδιάστηκε σε ένα λογισμικό 3D-CAD (InventorAutodesk®) και στη συνέχεια κάνοντας χρήση του αρχείου STEP που εξάγεται από το λογισμικό 3D-CAD αποτυπώνεται η ενοποίηση των μοντέλων OAM/CPM - NIST και Πίνακα Δομής Σχεδίασης(DSM). Η εικόνα 42 δείχνει τη λογική σειρά των βημάτων που ακολουθεί η μεθοδολογία για την ενοποίηση των μοντέλων.







Εικόνα 42: Λογικό διάγραμμα ενοποίησης των μοντέλων OAM/CPM-NIST και ΠΔΣ.

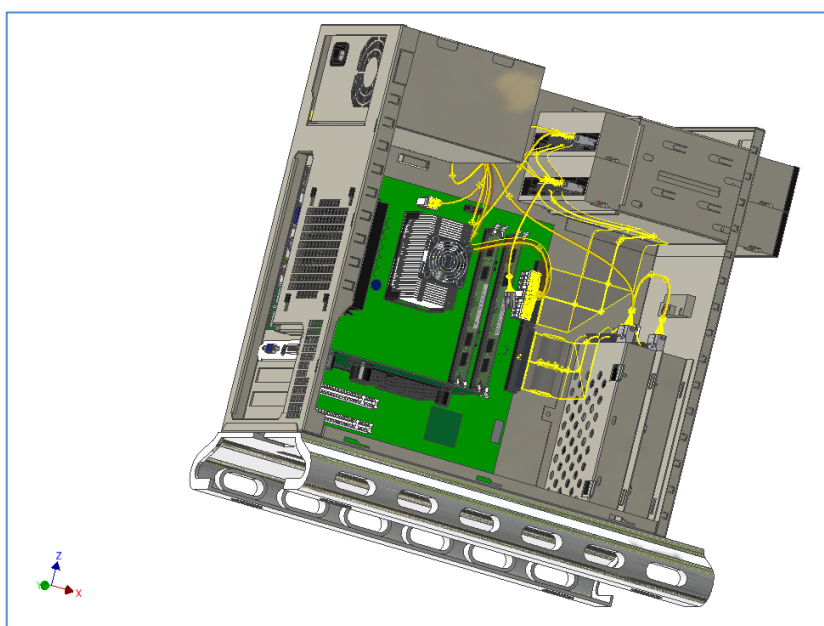
Μοντελοποίηση σε 3D-CAD

Ο υπολογιστής που σχεδιάστηκε έχει τα παρακάτω εξαρτήματα όπως φαίνονται στον [πίνακα 11] ο οποίος περιγράφει όχι μόνο τα είδη αλλά και τις ποσότητες ανά είδος.

#	Όνομα	Ποσό- τητα	Id - Όνομα	Γραφική αναπα- ράσταση
1	Κουτί συναρμολόγησης	1	1 – Case Box	
2	Μητρική Πλακέτα	1	2 – M/B	
3	Τροφοδοτικό	1	3 – PSU	
4	Επεξεργαστής	1	4 – CPU	
5	Ανεμιστήρας επεξεργαστή	1	5 – CPU Fan	
6	Κάρτα οθόνης	1	6 – GPU	
7	Κάρτα Δικτύου	1	7- NIC	
8	Κάρτα Ήχου	1	8-SCU	
9	Σκληρός Δίσκος	2	9 – HDD1	
			10 – HDD2	
10	Καλώδιο Σκληρού Δίσκου	1	11-HD Cable	
12	Οδηγός οπτικών δίσκων	2	12 – DVD1	
			13 – DVD2	

13	Καλώδιο σύνδεσης Οπτικών Δίσκων	1	14 – DVD Cable	
14	Εφεδρικός Οδηγός αποθήκευσης	1	15 – BDD	
15	Καλώδιο Σύνδεσης εφεδρικού οδηγού	1	16- BDD Cable	
16	Κύρια Μνήμη RAM	2	17 – RAM1	
			18 – RAM2	

Πίνακας 11: Πίνακας εξαρτημάτων του προσωπικού υπολογιστή



Εικόνα 43: Σχεδίαση του προσωπικού υπολογιστή με χρήση του InventorAutodesk®.

Παραγωγή και εξαγωγή αρχείου STEP

Στη συνέχεια γίνεται εξαγωγή του αρχείου STEP που αντιστοιχεί στην παραπάνω σχεδίαση. Ένα τμήμα του αρχείου STEP φαίνεται στην εικόνα 44.

```
ISO-10303-21;
HEADER;
/*****
 * Generated by software with PDElib inside
 * PDElib Version v51a, created Tue 11/30/2010
 * ITI Transcendata (http://www.transcendata.com/)
 *****/
FILE_DESCRIPTION(('','2;1');
FILE_NAME('C:\Users\luser\Desktop\IPC2\Main.stp','2012-01-26T12:29:15','(user)','(','Autodesk Inventor 2012','Autodesk Inventor 2012',')');
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
ENDSEC;
DATA;
#5=APPLICATION_CONTEXT('configuration controlled 3D designs of mechanical parts and assemblies');
#6=APPLICATION_PROTOCOL_DEFINITION('International
Standard','ap203_configuration_controlled_3d_design_of_mechanical_parts_and_assemblies_mim_if_2004.#5);
#7=PRODUCT_CONTEXT('','5','mechanical');
#8=PRODUCT('Main','Main','$(#7));
#106447=NEXT_ASSEMBLY_USAGE_OCCURRENCE('Case:1','Case:1','Case:1','57,#102','Case:1');
#227345=NEXT_ASSEMBLY_USAGE_OCCURRENCE('PSU:1','PSU:1','PSU:1','57,#106462','PSU:1');
#227357=NEXT_ASSEMBLY_USAGE_OCCURRENCE('Case_ASM:1','Case_ASM:1','Case_ASM:1','12,#57','Case_ASM:1');
#243401=NEXT_ASSEMBLY_USAGE_OCCURRENCE('CDROM:1','CDROM:1','CDROM:1','12,#227372','CDROM:1');
#243413=NEXT_ASSEMBLY_USAGE_OCCURRENCE('CDROM:2','CDROM:2','CDROM:2','12,#227372','CDROM:2');
#254607=NEXT_ASSEMBLY_USAGE_OCCURRENCE('Hard Drive:1','Hard Drive:1','Hard Drive:1','12,#243428','Hard Drive:1');
#352402=NEXT_ASSEMBLY_USAGE_OCCURRENCE('Power_cable:1','Power_cable:1','Power_cable:1','12,#254622','Power_cable:1');
#743355=NEXT_ASSEMBLY_USAGE_OCCURRENCE('Mobo2:1','Mobo2:1','Mobo2:1','352417,#352462','Mobo2:1');
#818613=NEXT_ASSEMBLY_USAGE_OCCURRENCE('CPU:1','CPU:1','CPU:1','352417,#743370','CPU:1');
#820756=NEXT_ASSEMBLY_USAGE_OCCURRENCE('RAM - DDR:1','RAM - DDR:1','RAM - DDR:1','352417,#818628','RAM - DDR:1');
#820768=NEXT_ASSEMBLY_USAGE_OCCURRENCE('RAM - DDR:2','RAM - DDR:2','RAM - DDR:2','352417,#818628','RAM - DDR:2');
#982401=NEXT_ASSEMBLY_USAGE_OCCURRENCE('GPU:1','GPU:1','GPU:1','352417,#820783','GPU:1');
#982413=NEXT_ASSEMBLY_USAGE_OCCURRENCE('MoboASM:1','MoboASM:1','MoboASM:1','12,#352417','MoboASM:1');
#1026890=NEXT_ASSEMBLY_USAGE_OCCURRENCE('HD_Cable:1','HD_Cable:1','HD_Cable:1','12,#982428','HD_Cable:1');
#1099143=NEXT_ASSEMBLY_USAGE_OCCURRENCE('CD_Cable:1','CD_Cable:1','CD_Cable:1','12,#1026905','CD_Cable:1');
```

Εικόνα 44: Τμήμα του αρχείου STEP που αντιστοιχεί στη σχεδίαση των εξαρτημάτων του προσωπικού υπολογιστή. Στην εικόνα φαίνονται οι οντότητες Next_Assembly που χρησιμοποιείται για την εύρεση της ιεραρχίας.

Τέλος, το αρχείο STEP εισάγεται στο λογισμικό OAM2DSM που ενοποιεί τα δύο μοντέλα.

Λεξικογραφική Ανάλυση και Ενημέρωση της Σχεσιακής Βάσης Δεδομένων

Στη συνέχεια ενημερώνεται η βάση δεδομένων των προϊόντων του OAM2DSM και αρχίζει η λεξικογραφική ανάλυση στο αρχείο STEP για την εύρεση των εξαρτημάτων του τελικού προϊόντος, την εύρεση της ιεραρχίας των εξαρτημάτων καθώς και την αναγνώριση των χαρακτηριστικών συναρμολόγησης. Όλα τα παραπάνω στοιχεία ενημερώνουν τη σχεσιακή βάση δεδομένων που υλοποιεί το μοντέλο OAM/CPM-NIST.

Ποιο συγκεκριμένα, η εισαγωγή του αρχείου STEP στο λογισμικό OAM2DSM ενεργοποιεί τον αλγόριθμο λεξικογραφικής ανάλυσης που περιγράφηκε στην παραπάνω ενότητα. Να σημειωθεί ότι κάθε νέα εισαγωγή αρχείου STEP στο OAM2DSM αντιμετωπίζεται ως ένα νέο έργο το οποίο παίρνει έναν μοναδικό κωδικό – κλειδί που το ακολουθεί όσο το έργο είναι αποθηκευμένο στη βάση δεδομένων του λογισμικού. Ακολουθούν οι εικόνες 45,46,47 και 48 από το λογισμικό OAM2DSM αποτυπώνουν τα στιγμιότυπα της σχεσιακής βάσης δεδομένων που υλοποιεί το μοντέλο OAM/CPM-NIST. Στην εικόνα 45 φαίνεται ο πίνακας **Artifact** στον οποίο αποθηκεύονται τα βασικά στοιχεία κάθε αρχείου STEP που αντιστοιχεί σε ένα διαφορετικό τεχνούργημα.

Artifact				
	Id	Description	Directory_Path	File_Name
+	1	Planetary Gear System		
+	2	7733 Fork Lift	C:\Users\user\Documents\ABBY Finereader 7	7733 Fork Lift
+	4	Main	C:\Users\user\Documents\PC	Main
+	5	Main	C:\Users\user\Documents\OAM2DSM	Main
+	6	Stapler Assembly	C:\Users\user\Documents\Stapler	Stapler Assem

Εικόνα 45: Ο πίνακας Artifact αποθηκεύει όλα τα έργα που υπάρχουν μέσα στο OAM2DSM καθώς και τη διαδρομή των αρχείων που θα χρησιμοποιηθούν ή θα εξαχθούν.

Η λεξικογραφική ανάλυση στο αρχείου STEP θα δημιουργήσει νέες εγγραφές στους αντίστοιχους πίνακες μιας και θα βρει όλα τα εξαρτήματα που αποτελούν στην τελική συναρμολόγηση όπως φαίνεται στην εικόνα 46.

Artifact				
Part				
	Id	Artifact_id	Description	Quantity
	217	4	'CDROM:2	1
	218	4	'Hard Drive:1	1
	219	4	'Power_cable:1	1
	220	4	'Mobo2:1	1
	221	4	'CPU:1	1
	222	4	'RAM - DDR:1	1
	223	4	'RAM - DDR:2	1
	224	4	'GPU:1	1
	225	4	'MoboASM:1	1
	226	4	'HD_Cable:1	1
	227	4	'CD_Cable:1	1

Εικόνα 46: Τα εξαρτήματα και η ποσότητά τους για τον υπό συναρμολόγηση υπολογιστή.

Στη συνέχεια ανιχνεύονται αν στο αρχείο STEP υπάρχουν υποσυναρμολογήσεις από εξαρτήματα αναζητώντας επιπλέον και την ιεραρχία της συναρμολόγησης όπως έχει αποτυπωθεί από το σχεδιαστή. Στην εικόνα 47 φαίνονται με διακριτό τρόπο οι υποσυναρμολογήσεις και τα εξαρτήματα

Artifact					
Assembly					
	Id	Artifact_id	Description	AssemblyOrPart	Part_id
+	160	2	'4X6X1 Tan:1	0	85
+	261	4	Main	1	1
+	262	4	'Case_ASM:1	1	3
+	263	4	'PSU:1	0	2
+	264	4	'Case:1	0	1
+	265	4	'CDROM:1	0	4
+	266	4	'CDROM:2	0	5
+	267	4	'Hard Drive:1	0	6
+	268	4	'Power_cable:1	0	7
+	269	4	'MoboASM:1	1	13
+	270	4	'GPU:1	0	12
+	271	4	'RAM - DDR:2	0	11
+	272	4	'RAM - DDR:1	0	10
+	273	4	'CPU:1	0	9
+	274	4	'Mobo2:1	0	8
+	275	4	'HD_Cable:1	0	14
+	276	4	'CD_Cable:1	0	15

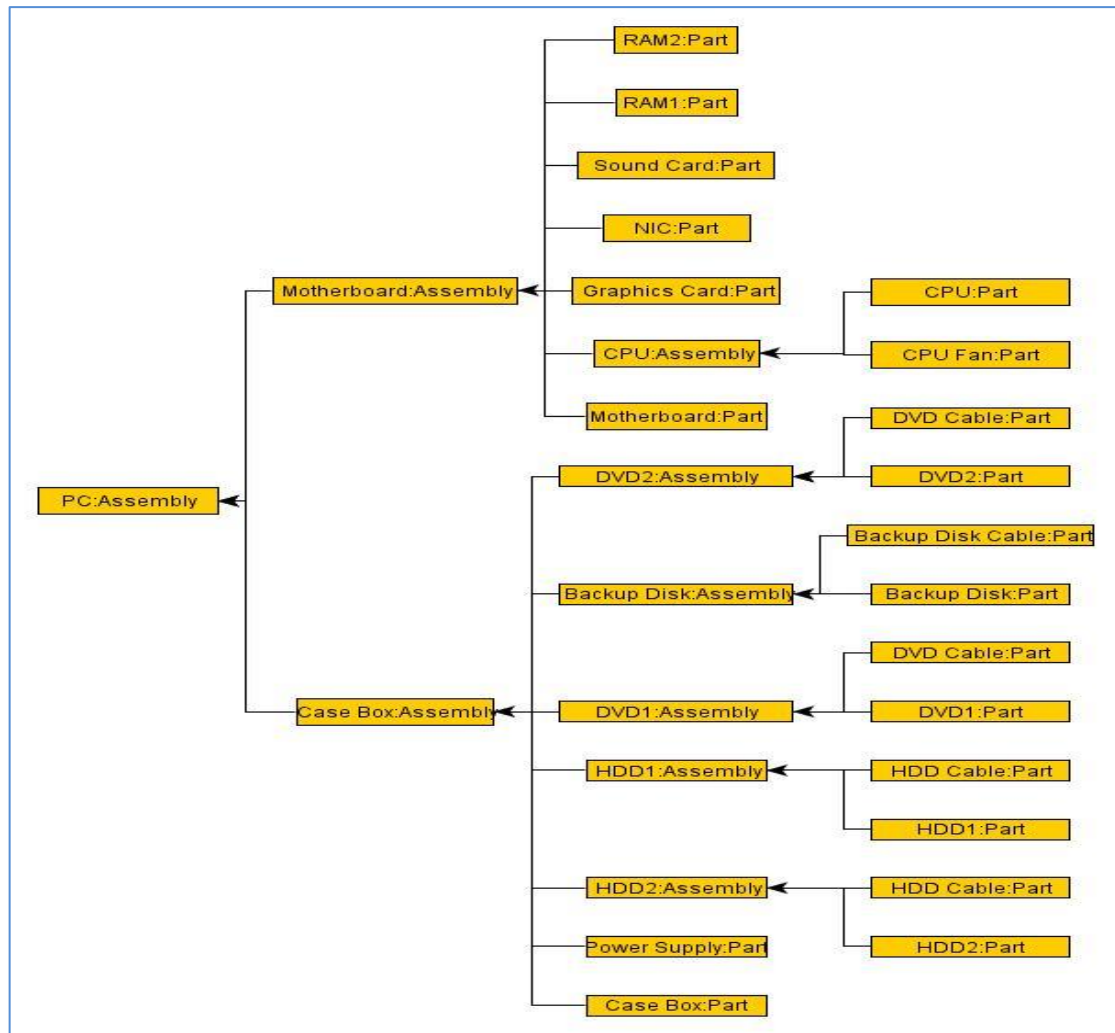
Εικόνα 47: Στιγμιότυπο από τον πίνακα Assembly_ ο οποίος από το πεδίο AssemblyOrPart διακρίνει αν πρόκειται για εξάρτημα ή υποσυναρμολόγηση.

Συνεχίζοντας την ανάλυση του αρχείου STEP και με τη βοήθεια του πίνακα Assembly_ ενημερώνεται ο πίνακας που αποτυπώνει την ιεραρχία της συναρμολόγησης όπως φαίνεται και στην εικόνα 48.

Artifact		Assembly_		Assembly_Hierarchy	
Id	Assembly_Id	ArtifactHasF	PartOfArtifa	Artifact_Id	
260	261		-1	4	
261	262		261	4	
262	263		262	4	
263	264		262	4	
264	265		261	4	
265	266		261	4	
266	267		261	4	
267	268		261	4	
268	269		261	4	
269	270		269	4	
270	271		269	4	
271	272		269	4	
272	273		269	4	
273	274		269	4	
274	275		261	4	
275	276		261	4	

Εικόνα 48: Ο πίνακας Assembly_Hierarchy αποτυπώνει την ιεραρχία συναρμολόγησης κάνοντας χρήση των σχέσεων ArtifactHasPart και PartOfArtifact.

Να σημειωθεί ότι ο πίνακας Assembly_Hierarchy αποτελεί στη ουσία μια ιεραρχική αποτύπωση της τελικής συναρμολόγησης κάθε τεχνουργήματος σύμφωνα πάντα με τον τρόπο που ορίζεται μέσα στο αρχείο STEP. Η τιμή -1 που φαίνεται στην παραπάνω εικόνα σηματοδοτεί προγραμματιστικά ότι η εγγραφή αυτή αποτελεί τη συνολική συναρμολόγηση του προϊόντος. Η εικόνα 49 δείχνει τη αποτύπωση των μερών του υπολογιστή, αφού πρώτα έγινε μετατροπή των αναγνωριστικών του παραπάνω πίνακα στα αντίστοιχα λεκτικά.



Εικόνα 49: Αποτύπωση με γραφικό τρόπο της ιεραρχίας συναρμολόγησης όπως αυτή υπάρχει μέσα στο αρχείο STEP.

Επόμενο βήμα για τη λεξικογραφική ανάλυση είναι η ανίχνευση γεωμετρικών χαρακτηριστικών και στη συνέχεια η εύρεση χαρακτηριστικών συναρμολόγησης. Να σημειωθεί ότι για την ενοποίηση των δύο μοντέλων (OAM/CPM-NIST και ΠΔΣ) αρκεί η εύρεση χαρακτηριστικών συναρμολόγησης χωρίς να χρειάζεται η σημασιολογική τους ερμηνεία. Αυτή θα μπορούσε να αποτελέσει μια μελλοντική επέκταση της παρούσας εργασίας. Στην εικόνα 50 αποτυπώνεται ένα στιγμιότυπο από τα χαρακτηριστικά συναρμολόγησης που βρέθηκαν ακολουθώντας τη μεθοδολογία που αναπτύχθηκε στην προηγούμενη ενότητα.

Artifact	Assembly_	Assembly_Hierarchy	Connection	Constraint
Id	Type_of_Constraints_Id	Specific_Description		
319	1	Constraint between 'kapaki and 'CR 40X62X7 HMS5 RG_PART1:1		
320	1	Constraint between 'Case_ASM:1 and 'Case:1		
321	1	Constraint between 'Case_ASM:1 and 'PSU:1		
322	1	Constraint between 'MoboASM:1 and 'Mobo2:1		
323	1	Constraint between 'MoboASM:1 and 'CPU:1		
324	1	Constraint between 'MoboASM:1 and 'RAM - DDR:1		
325	1	Constraint between 'MoboASM:1 and 'RAM - DDR:2		
326	1	Constraint between 'MoboASM:1 and 'GPU:1		
327	1	Constraint between 'Case_ASM:1 and 'Case:1		
328	1	Constraint between 'Case_ASM:1 and 'PSU:1		
329	1	Constraint between 'MoboASM:1 and 'Mobo2:1		
330	1	Constraint between 'MoboASM:1 and 'CPU:1		
331	1	Constraint between 'MoboASM:1 and 'RAM - DDR:1		
332	1	Constraint between 'MoboASM:1 and 'RAM - DDR:2		
333	1	Constraint between 'MoboASM:1 and 'GPU:1		

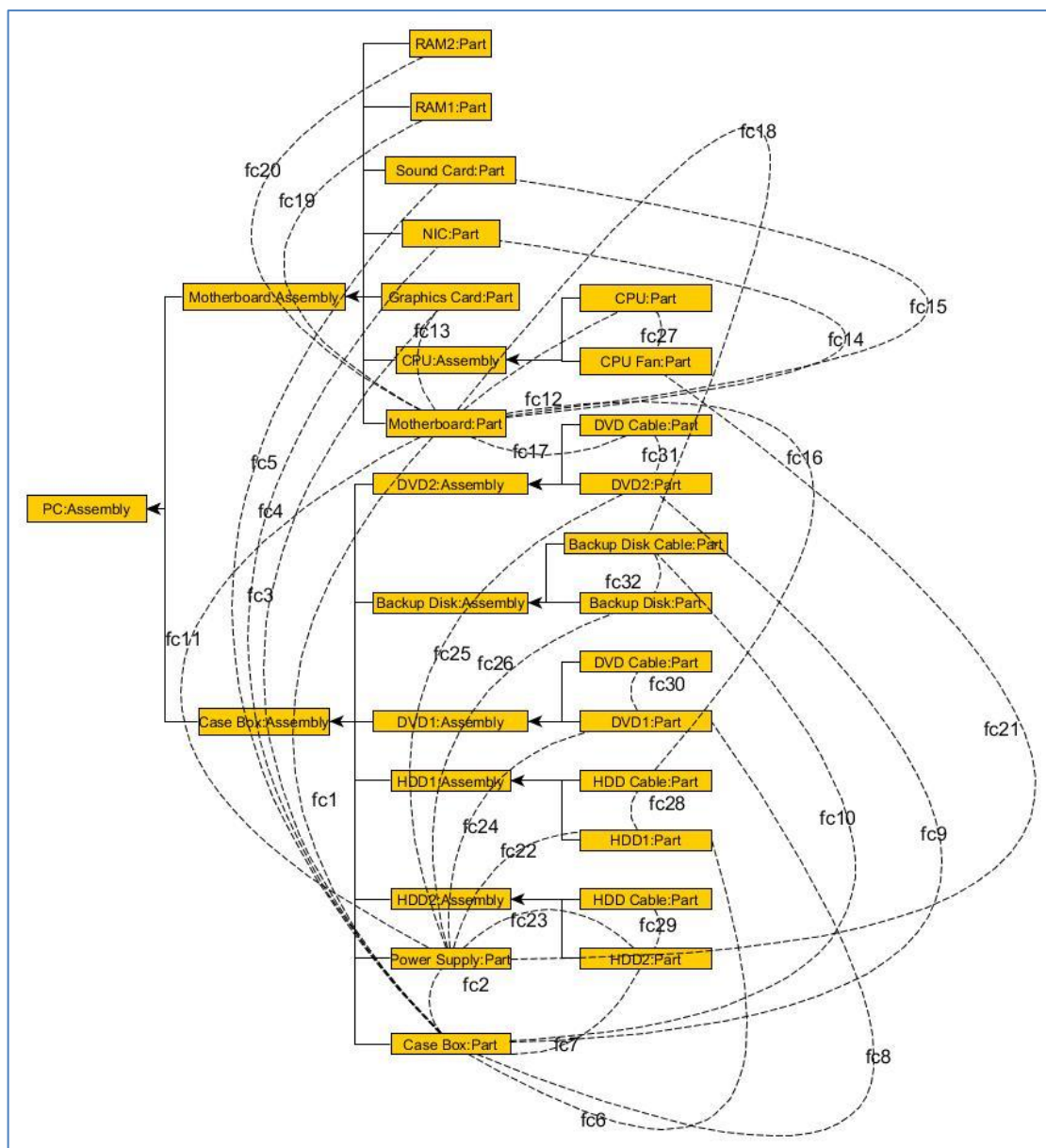
Εικόνα 50: Στιγμιότυπο του πίνακα Constraint που αποτυπώνει χαρακτηριστικά συναρμολόγησης που ανιχνεύτηκαν στην αρχική σχεδίαση.

Η πληροφορία αυτή αναλύεται και μετασχηματίζεται σε μορφή που να μπορεί να αποτυπωθεί στη συνέχεια σε έναν πίνακα δομής σχεδίασης με γραμμές και στήλες τα εξαρτήματα και τιμές 1 αν υπάρχει συναρμολόγηση μεταξύ των εξαρτημάτων και 0 αν δε συσχετίζονται.

Artifact	Assembly_	Assembly_Hierarchy	Connection	Constraint	AFARToKinematicPair	AC		
Id	Artifact_ID	I_Element	I_Element_Desc	A_Element	A_Element_Desc	ToC	Lr	ALC_Phase
4	4	1	'PSU:1	5	'Hard Drive:1	Overall G	1	Phase 3----Co
5	4	1	'PSU:1	11	'Mobo2:1	Overall G	1	Phase 3----Co
6	4	2	'Case:1	3	'CDROM:1	Overall G	1	Phase 2----Co
7	4	2	'Case:1	4	'CDROM:2	Overall G	1	Phase 2----Co
8	4	2	'Case:1	5	'Hard Drive:1	Overall G	1	Phase 2----Co
9	4	2	'Case:1	11	'Mobo2:1	Overall G	1	Phase 2----Co
10	4	3	'CDROM:1	13	'CD_Cable:1	Overall G	1	Phase 3----Co
11	4	4	'CDROM:2	13	'CD_Cable:1	Overall G	1	Phase 3----Co
12	4	5	'Hard Drive:1	12	'HD_Cable:1	Overall G	1	Phase 3----Co
13	4	7	'GPU:1	11	'Mobo2:1	Overall G	1	Phase 1----Co
14	4	8	'RAM - DDR:2	11	'Mobo2:1	Overall G	1	Phase 1----Co
15	4	9	'RAM - DDR:1	11	'Mobo2:1	Overall G	1	Phase 1----Co
16	4	10	'CPU:1	11	'Mobo2:1	Overall G	1	Phase 1----Co
17	4	11	'Mobo2:1	12	'HD_Cable:1	Overall G	1	Phase 1----Co
18	4	11	'Mobo2:1	13	'CD_Cable:1	Overall G	1	Phase 1----Co

Εικόνα 51: Στιγμιότυπο του πίνακα AC ο οποίος αποτυπώνει τη συσχέτιση των εξαρτημάτων μεταξύ τους.

Ο παραπάνω μετασχηματισμός εμπλουτίζει την ιεραρχία συναρμολόγησης με τα χαρακτηριστικά συναρμολόγησης που βρέθηκαν εκχωρώντας τον τύπο σύνδεσης των εξαρτημάτων όπως φαίνεται στην εικόνα 52.



Εικόνα 52: Γραφική αναπαράσταση των συνδέσεων μεταξύ των εξαρτημάτων στην ήδη υπάρχουσα ιεραρχία συναρμολόγησης.

Στην εικόνα 53 φαίνονται τα είδη των συνδέσεων που επιτρέπονται στη συγκεκριμένη εφαρμογή. Συγκεκριμένα υπάρχουν τέσσερις τύπους συνδέσεων σύμφωνα με το μοντέλο OAM/CPM-NIST που είναι οι : Μη καθορισμένη σύνδεση (ni), Σταθερή σύνδεση (fc), Κινούμενη σύνδεση (mc), Διακοπτόμενη σύνδεση (ic).

Artifact		Type_of_Connection		
	Id	Description	Abbreviation	Προσθήκη νέου πεδίου
+	0	Non-identifiable	ni	
+	1	FixedConnection	fc	
+	2	MovableConnection	mc	
+	3	IntermittentConnection	ic	

Εικόνα 53: Στιγμιότυπο από τα είδη των συνδέσεων που επιτρέπονται στα συναρμολογούμενα εξαρτήματα.

Artifact As- sociations	Artifacts	Assembly features	Assembly Constraints
fc1	Case Box Motherboard	(CaseBoxMB:AF) (MBCaseBox:AF)	<ul style="list-style-type: none"> Alignment guide
fc2	Case Box Power Supply Unit	(CaseBoxPSU:AF) (PSUCaseBox:AF)	<ul style="list-style-type: none"> Alignment guide Slide the PSU inward
fc3	Case Box Graphics Card	(CaseBoxGC:AF) (GCCaseBox:AF)	<ul style="list-style-type: none"> Filler bracket Alignment guide Alignment bar Card retention door Release tab
Fc4	Case Box Network Card	(CaseBoxNC:AF) (NCCaseBox:AF)	<ul style="list-style-type: none"> Filler bracket Alignment guide Alignment bar Card retention door Release tab
Fc5	Case Box Sound Card	(CAsBoxSC:AF) (SCCaseBox:AF)	<ul style="list-style-type: none"> Filler bracket Alignment guide Alignment bar Card retention door Release tab
Fc6	Case Box HDD1	Case box Slot1(CaseboxHDD1:AF) HDD1- 2 επιφάνειες (HDD1CaseBox:AF)	<ul style="list-style-type: none"> Align the two surfaces of HDD with the corresponding socket Slide the hard disk inward
Fc7	Case Box HDD2	Case box Slot2(CaseboxHDD2:AF) HDD2 - 2 επιφάνειες (HDD2CaseBox:AF)	<ul style="list-style-type: none"> Align the two surfaces of HDD with the corresponding socket Slide the hard disk inward
Fc8	Case Box DVD1	Case box Slot3(CaseboxDVD1:AF) DVD1- 2 επιφάνειες (DVD1CaseBox:AF)	<ul style="list-style-type: none"> Align the two surfaces of DVD with the corresponding socket Slide the hard disk inward
Fc9	Case Box DVD2	Case box Slot4(CaseboxDVD2:AF) DVD2- 2 επιφάνειες (DVD2CaseBox:AF)	<ul style="list-style-type: none"> Align the two surfaces of DVD with the corresponding socket Slide the hard disk inward
Fc10	Case Box Backup Drive	Case box Slot5(CaseboxBDD:AF) BackUpDrive- 2 επιφάνειες (BDDCaseBox:AF)	<ul style="list-style-type: none"> Align the two surfaces of DVD with the corresponding socket Slide the hard disk inward

Fc11	Motherboard	MB Power Supply Connector(MBPSU:AF)	<ul style="list-style-type: none"> Slide the power supply into place
	Power Supply Unit	Power Supply Unit MB Connector(PSUMB:AF)	
Fc12	Motherboard	MB CPU Socket (MBCPU:AF)	<ul style="list-style-type: none"> Position the processor correctly in the socket Orient the front and rear alignment notches on the processor with the front and rear alignment notches on the socket Align the pin-1 corners of the processor and socket.
	CPU	CPU Pins (CPUMB:AF)	
Fc13	Motherboard	MB AGP Slot (AGPGC:AF)	<ul style="list-style-type: none"> Orient the card with the slot Align Push
	Graphics Card	Graphics Card Connectors (GCAGP:AF)	
Fc14	Motherboard	MB PCI Express Slot1 (PCI1NC:AF)	<ul style="list-style-type: none"> Orient the card with the slot Align Push
	Network Card	Network Card Connectors (NCPCI1:AF)	
Fc15	Motherboard	MB PCI Express Slot2 (PCI2NC:AF)	<ul style="list-style-type: none"> Orient the card with the slot Align Push
	Sound Card	Sound Card Connectors (SCPCI2:AF)	
Fc16	Motherboard	MB SATA Slot (MBHDDCable:AF)	<ul style="list-style-type: none"> Connect cable (one way connectors) with M/B
	HDD Cable	HDD cable connector (HDDCableMB:AF)	
Fc17	Motherboard	MB IDE Connector 1 (MBDVDCable:AF)	<ul style="list-style-type: none"> Connect cable (one way connectors) with M/B
	DVD Cable	DVD cable Connector (DVDCableMB:AF)	
Fc18	Motherboard	MB IDE Connector 2 (MBBDDCable:AF)	<ul style="list-style-type: none"> Connect cable (one way connectors) with M/B
	Backup Drive Cable	BDD cable Connector (BDDCableMB:AF)	
Fc19	Motherboard	MB DIMM memory 1 (MBRAM1:AF)	<ul style="list-style-type: none"> Align the notch on the bottom of the module with the crossbar in the connector.
	RAM1	RAM1 Connectors (RAMMB1:AF)	
Fc20	Motherboard	MB DIMM memory 2 (MBRAM2:AF)	<ul style="list-style-type: none"> Align the notch on the bottom of the

	RAM2	RAM2 Connectors (RAMMB2:AF)	module with the crossbar in the con- nector.
Fc21	Power Supply Unit	PSU CPU Cable (PSUCPUFan:AF)	<ul style="list-style-type: none"> Plug in the into a three-pin labeled "CPU Fan"
	CPU Fan	CPU Fan Power Supply connector (CPUFanPSU:AF)	
Fc22	Power Supply Unit	Power Connector SATA (PSUHDD1:AF)	<ul style="list-style-type: none"> Connect power sup- ply unit cable into HDD1 – one way connector
	HDD1	Power Connector HDD (HDD1PSU:AF)	
Fc23	Power Supply Unit	Power Connector SATA (PSUHDD2:AF)	<ul style="list-style-type: none"> Connect power sup- ply unit cable into HDD2 – one way connector
	HDD2	Power Connector HDD (HDD2PSU:AF)	
Fc24	Power Supply Unit	Power Connector SATA (PSUDVD1:AF)	<ul style="list-style-type: none"> Connect power sup- ply unit cable into DVD1 – one way connector
	DVD1	Power Connector DVD (DVD1PSU:AF)	
Fc25	Power Supply Unit	Power Connector SATA (PSUDVD2:AF)	<ul style="list-style-type: none"> Connect power sup- ply unit cable into DVD2 – one way connector
	DVD2	Power Connector DVD (DVD2PSU:AF)	
Fc26	Power Supply Unit	Power Connector SATA (PSUBDD:AF)	<ul style="list-style-type: none"> Connect power sup- ply unit cable into BDD – one way con- nector
	Backup Drive	Power Connector BDD (BDDPSU:AF)	
Fc27	CPU	CPU Fan connectors (CPUCPUFan:AF)	<ul style="list-style-type: none"> Mount CPU Fan on to CPU
	CPU Fan	CPU fan connectors (CPUFanCPU:AF)	
Fc28	HDD1	HDD Sata Connector (HDD1HDCable:AF)	<ul style="list-style-type: none"> Connect cable (one way connectors) with HDD1
	HD Cable	HD Cable Connector 1 (HDCableHDD1:AF)	
Fc29	HDD2	HDD Sata Connector (HDD2HDCable:AF)	<ul style="list-style-type: none"> Connect cable (one way connectors) with HDD2
	HD Cable	HD Cable Connector 2 (HDCableHDD2:AF)	
Fc30	DVD1	DVD IDE Connector (DVD1DVDCable:AF)	<ul style="list-style-type: none"> Connect cable (one way connectors) with DVD
	DVD cable	DVD Cable Connector 1 (DVDCableDVD1:AF)	
Fc31	DVD2	DVD IDE Connector (DVD2DVDCable:AF)	<ul style="list-style-type: none"> Connect cable (one way connectors) with DVD
	DVD Cable	DVD Cable Connector 2 (DVDCableDVD2:AF)	

Fc32	Backup Drive	BDD Cable Connector (BDDBDDCable:AF)	• Connect cable (one way connectors) with BDD
	Backup Drive Cable	BDD Cable Connector (BDDCableBDD:AF)	

Πίνακας 14: Σχέσεις συναρμολόγησης για τον υπολογιστή όπως έχουν εξαχθεί από τη βάση δεδομένων που υλοποιεί το μοντέλο OAM/CPM-NIST

Εισαγωγή των δεδομένων στο Μοντέλο του Πίνακα Δομής Σχεδίασης

Στη συνέχεια από το αρχείο STEP παράγεται ο ΠΔΣ που ακολουθεί (εικόνα 56).

Personal Computer DSM		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Case Box	1	1	1				1	1	1	1	1		1	1		1			
Motherboard	2		1	1			1	1	1			1			1		1	1	1
Power Supply Unit	3			1		1				1	1		1	1		1			
CPU	4				1														
CPU Fan	5					1													
Graphics Card	6						1												
Network Card	7							1											
Sound Card	8								1										
HDD1	9									1									
HDD2	10										1								
HD Cable	11											1							
DVD1	12												1						
DVD2	13													1					
DVD Cable	14														1				
Backup Drive	15															1			
BackUp Cable	16																1		
RAM1	17																	1	
RAM2	18																		1

Εικόνα 56: Πίνακας Δομής Σχεδίασης για τον υπολογιστή.

Αφού προσδιοριστεί το κόστος κάθε συναρμολόγησης σύμφωνα με τον [πίνακα 11] που ορίζει τα κριτήρια ο πίνακας έχει την παρακάτω μορφή στο λογισμικό OAM2DSM εικόνα 57.

Weight Assembly DSM	Components DSM	Assembly DSM	Participated Index(PI)	Clusters / Costs															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	15	13				13,5	13,5	13,5	12,75	12,75		12,75	12,75		12,75				
2	15	11,25	11,25			8,25	8,25	8,25			8,25			8,25		8,25	8,25	8,25	
3	13	11,25			11,25					11,25	11,25		11,25	11,25		11,25			
4		11,25			13,5														
5			11,25	13,5															
6	13,5	8,25																	
7	13,5	8,25																	
8	13,5	8,25																	
9	12,75		11,25								11,25								
10	12,75		11,25								11,25								
11		8,25							11,25	11,25									
12	12,75		11,25											11,25					
13	12,75		11,25											11,25					
14		8,25											11,25	11,25					
15	12,75		11,25													11,25			
16		8,25															11,25		
17		8,25																11,25	
18		8,25																	11,25

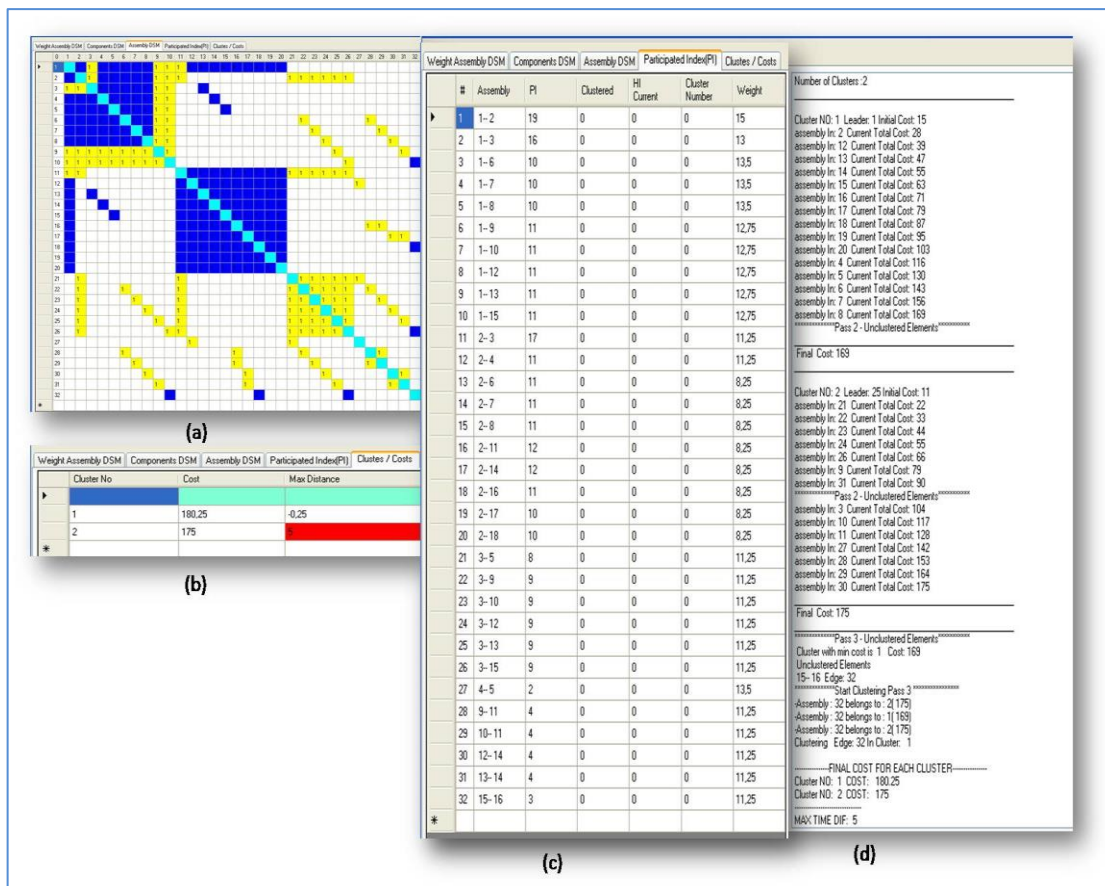
Εικόνα 57: Ο ΠΔΣ όπως αποτυπώνεται στο OAM2DSM μετά την κοστολόγηση κάθε συναρμολόγησης.

Ακολουθώντας τη μεθοδολογία που ήδη έχει περιγραφεί, ο πίνακας μετασχηματίζεται σε ένα ΠΔΣ συναρμολογήσεων ADSM. Ως στήλες και γραμμές έχει πλέον τις συναρμολογήσεις όπως φαίνεται στην εικόνα 58.

Weight Assembly DSM	Components DSM	Assembly DSM	Participated Index(PI)	Clustes / Costs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

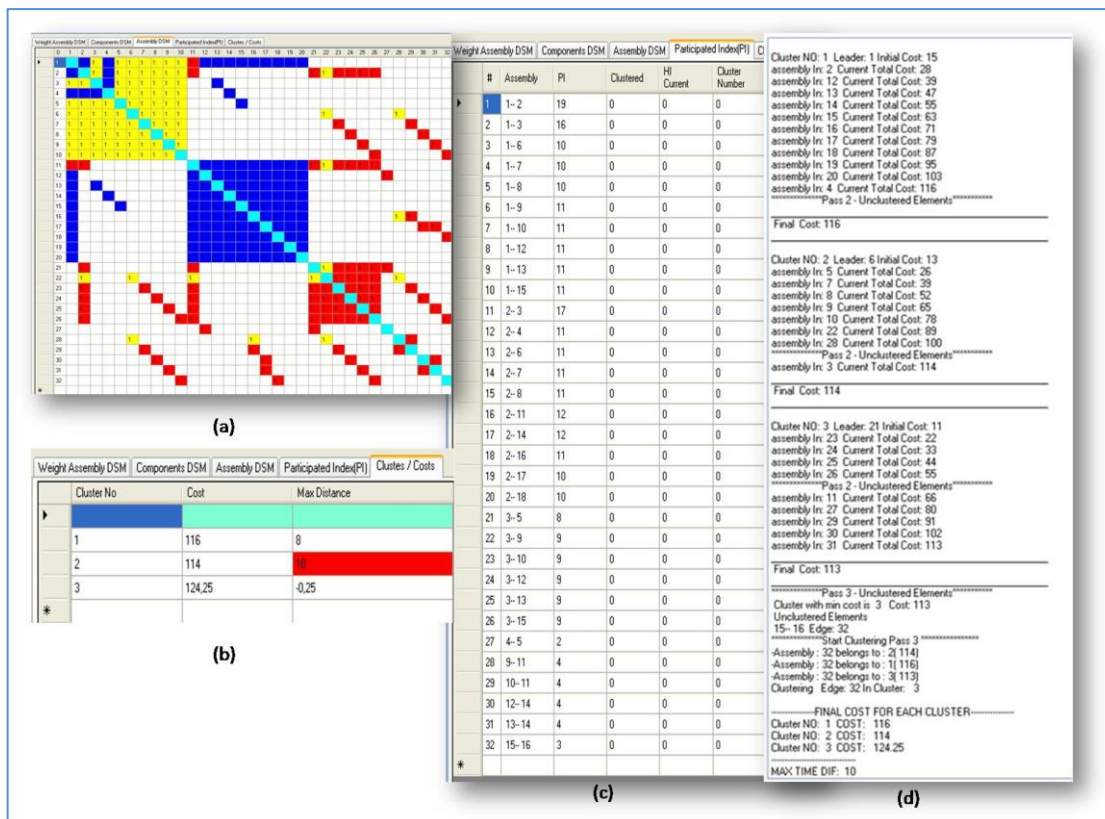
Εικόνα 58: Ο πίνακας ADSM που περιγράφει τις σχέσεις μεταξύ των συναρμολογήσεων.

Στη συνέχεια γίνεται ορισμός του αριθμού των ομάδων που θα επιμεριστούν οι συναρμολογήσεις. Για αριθμό ομάδων $N=2$, η εικόνα 59 περιγράφει συνοπτικά την έξοδο του αλγόριθμου.

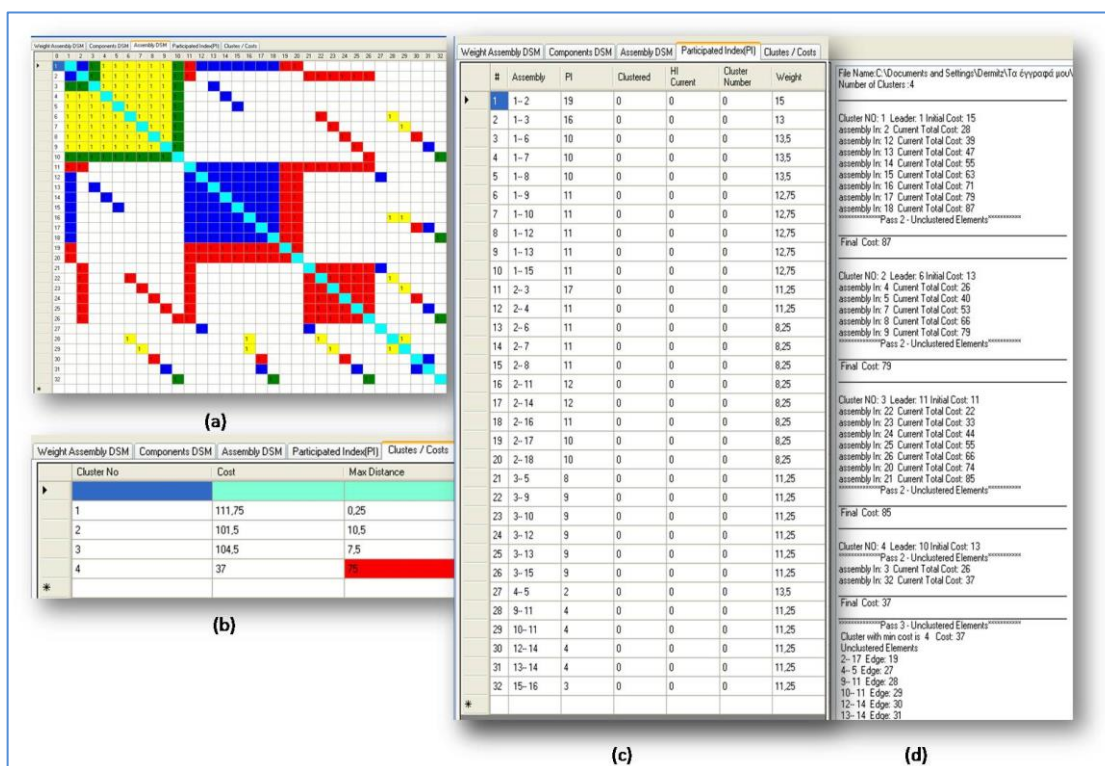


Εικόνα 59: Έξοδος από τον αλγόριθμο για N=2.

Στην εικόνα 59.a αποτυπώνεται με γραφικό τρόπο η ομαδοποίηση. Στην εικόνα 59.b αποτυπώνεται το συνολικό κόστος κάθε ομάδας καθώς επίσης και η διαφορά της κάθε ομάδας από τον υπολογισμένο μέσο όρο. Στην εικόνα 59.c εμφανίζονται όλες οι συναρμολογήσεις, ποια εξαρτήματα τις δημιουργούν, ο δείκτης PI που έχει αρχικά υπολογιστεί και το κόστος κάθε συναρμολόγησης. Τέλος στην εικόνα 59.d, ο αλγόριθμος δίνει όλη την πορεία της εξέλιξης της ομαδοποίησης. Για αριθμό ομάδων N=3 η εικόνα 60 περιγράφει συνοπτικά την έξοδο του αλγόριθμου.

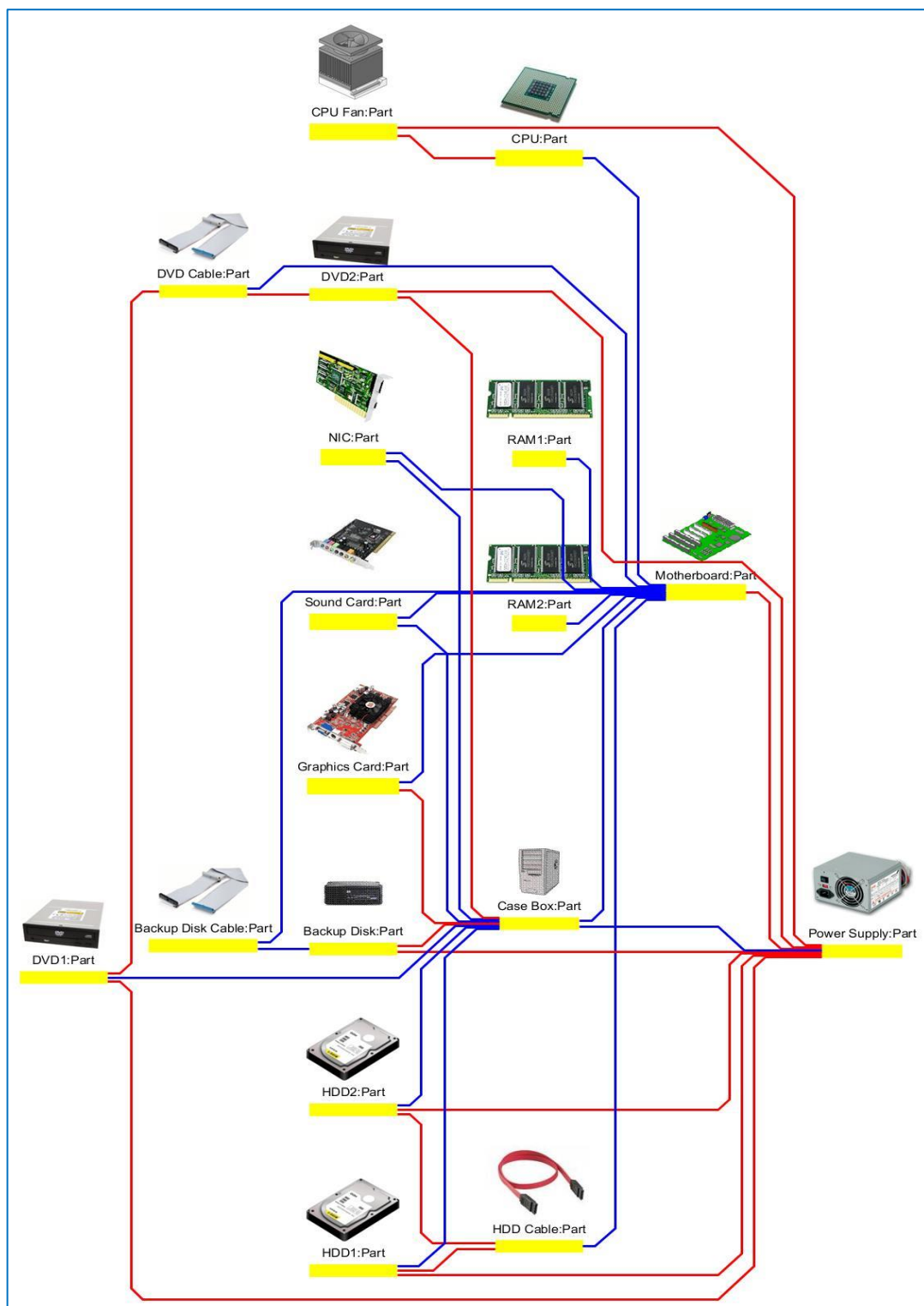


Εικόνα 60: Έξοδος από τον αλγόριθμο για N=3

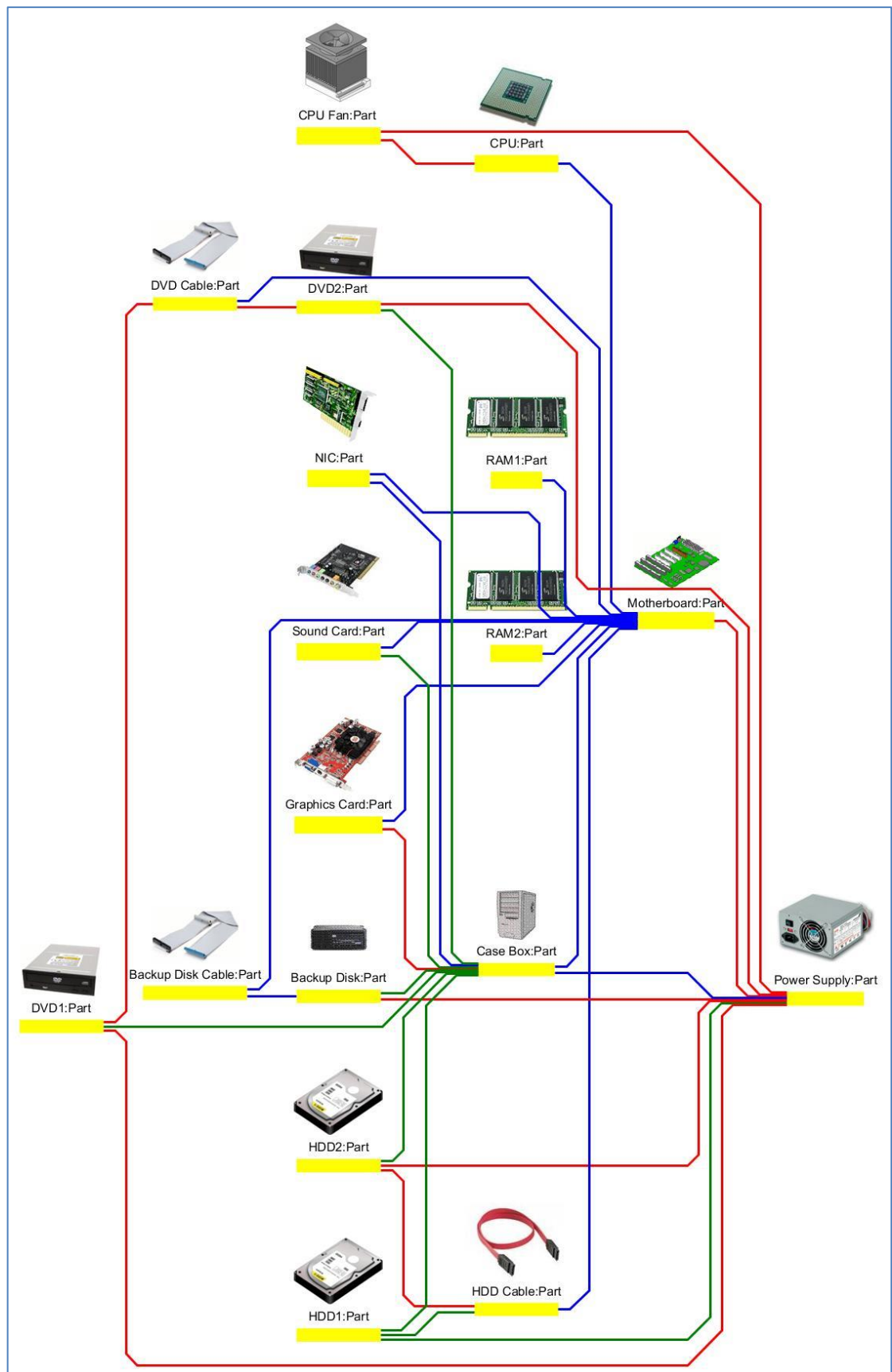


Εικόνα 61: Έξοδος από τον αλγόριθμο για N=4

Αποτυπώνοντας τα αποτελέσματα για τις διαφορετικές τιμές του $N(2,3)$ στις εικόνες 62 και 63 εμφανίζονται οι ομαδοποιημένες συναρμολογήσεις.



Εικόνα 62: Ομαδοποίηση συναρμολογήσεων για $N=2$



Εικόνα 63: Ομαδοποίηση συναρμολογήσεων για N=3

4. Σύνοψη – Συμπεράσματα

Η εργασία αυτή ως στόχο έχει να λύσει τα προβλήματα υλοποίησης και εφαρμογής σε πραγματικές συνθήκες, του μοντέλου ανοικτής συναρμολόγησης (OAM) που ορίζει το NIST. Προτείνει μια μεθοδολογία για την υλοποίηση του από το εννοιολογικό επίπεδο στο επίπεδο υλοποίησης και ενσωματώνει χαρακτηριστικά του πίνακα δομής σχεδίασης. Το νέο μοντέλο που προκύπτει (OAM2DSM), αποτελεί μια νέα πλατφόρμα για τη διαδικασία ανάπτυξης του προϊόντος με προσανατολισμό στη συναρμολόγηση. Για την επίτευξη των παραπάνω στόχων σχεδιάστηκε και υλοποιήθηκε ένα πληροφοριακό μοντέλο ροής της πληροφορίας από στάδιο σε στάδιο στη διαδικασία ανάπτυξης του προϊόντος. Η έρευνα επικεντρώθηκε στις παρακάτω δύο μεθοδολογίες:

- Τη μεθοδολογία του Πίνακα Δομής Σχεδίασης (ΠΔΣ) η οποία υποθέτει ότι κάθε τμήμα της δραστηριότητας της ανάπτυξης μπορεί να αποτυπωθεί ως πληροφορία δημιουργώντας νέες πληροφορίες.
- Τη μεθοδολογία Core Product Model CPM/NIST της οποίας βασικός στόχος είναι η αναπαράσταση της ροής της πληροφορίας στη διαδικασία ανάπτυξης ενός προϊόντος σε όλα τα στάδια της εξέλιξης του.

Η έρευνα στην εργασία αυτή σε ότι αφορά το μοντέλο CPM επικεντρώθηκε σε ζητήματα τα οποία παραμένουν ανοικτά για την ολοκλήρωση του μοντέλου. Συγκεκριμένα, κάνοντας χρήση μιας επέκτασης του αρχικού εννοιολογικού μοντέλου του Μοντέλου Ανοικτής Συναρμολόγησης (OAM), επιχειρεί να υλοποιήσει το Ενδιάμεσο Μοντέλο παρέχοντας τις δομές δεδομένων που χρειάζονται για την αποθήκευση των αντικειμένων και στη συνέχεια προχωρά στην υλοποίηση μιας αντικειμενοστραφούς πλατφόρμας ολοκληρώνοντάς το σε Μοντέλο Υλοποίησης.

Το βασικό χαρακτηριστικό του OAM είναι η δυνατότητά του, σε θεωρητικό επίπεδο, να παρακολουθεί την εξέλιξη της σχεδίασης και συναρμολόγησης ενός προϊόντος σε κάθε του φάση. Αυτό είναι εφικτό μιας και το OAM αποτελεί επέκταση του CPM το οποίο αποτυπώνει με ανεξάρτητο από πλατφόρμα τρόπο, τη σχεδίαση από τα πρώιμα στάδια της. Παράλληλα αποτελεί ένα ισχυρό εργαλείο το οποίο μπορεί να αναπαραστήσει όλη την πληροφορία που απαιτείται και ως έννοια αλλά και ως δομή δεδομένων για την πλήρη αποτύπωση της σχεδίασης και συναρμολόγησης του προϊόντος. Τέλος, ενσωματώνει όλες τις κλάσεις αντικειμένων καθώς και τις σχέσεις μεταξύ τους για την αποτύπωση πληροφορίας που σχετίζονται με ανοχές, κινηματική και λογικές σχεδίασης.

Επιγραμματικά, το OAM αποτελεί ένα μοντέλο διασύνδεσης με όλα εκείνα τα λογισμικά που συνεισφέρουν το καθένα από τη δική του σκοπιά στην τελική σχεδίαση ενός προϊόντος. Παράλληλα, λόγω της ανοικτής αρχιτεκτονικής του, η οποία είναι ανεξάρτητη πλατφόρμας υλοποίησης, επιλύει προβλήματα διασύνδεσης που συνήθως οι σχεδιαστές αντιμετωπίζουν από τη μια πλατφόρμα στην άλλη.

Για την ενοποίηση του μοντέλου Πίνακα Δομής Σχεδίασης (DSM) και του Μοντέλου Ανοικτής Συναρμολόγησης (OAM) χρησιμοποιήθηκε ο Πίνακας Δομής Σχεδίασης Συστατικών ο οποίος με τη βοήθεια αλγορίθμων ομαδοποίησης εστιάζει στην εύρεση υποσυνόλων στοιχείων μιας συναρμολόγησης ελαχιστοποιώντας προβλήματα στην αρχική σχεδίαση (επαναλήψεις και αναδράσεις). Η ενοποίηση των δύο μοντέλων επιτυγχάνεται κάνοντας χρήση της τεχνολογίας αγωγών (pipeline) που χρησιμοποιείται ευρέως στην τεχνολογία λογισμικού. Συγκεκριμένα, η πληροφορία ρέει από το μοντέλο OAM στο μοντέλο DSM δημιουργώντας έτσι εικονικούς

αγωγούς ροής πληροφορίας από τα στάδια της αρχικής σχεδίασης μέχρι τη συναρμολόγηση του προϊόντος. Η τροφοδότηση πληροφορίας από το ένα μοντέλο στο άλλο απαιτεί βέβαια μετασχηματισμούς δεδομένων τα οποία να παρέχουν και εννοιολογική και σημασιολογική αναπαράσταση.

Το μοντέλο ενοποίησης που προέκυψε (OAM2DSM) συνδέει τις δύο αυτές μεθόδους χρησιμοποιώντας το OAM ως μοντέλο διεπαφής μεταξύ των λογισμικών σχεδίασης (CAD systems) και του μοντέλου DSM. Η ενοποίηση αυτή συμβάλει :

- Υποστηρίζοντας τη διαδικασία σχεδίασης από τα πρώιμα στάδια της και καθ' όλη τη διάρκεια του κύκλου ζωής ενός προϊόντος
- Ισχυροποιεί την ενοποίηση συστημάτων σχεδίασης (CAD) με συστήματα ανάλυσης
- Δίνει μια δυνατότητα πρόβλεψης στη σχεδίαση, συμβάλλοντας έτσι στη μείωση του χρόνου σχεδίασης μια και εντοπίζονται γρηγορότερα σχεδιαστικές αστοχίες ή λάθη.
- Παρέχει ένα ισχυρό πληροφοριακό μοντέλο, ικανό να αποτυπώσει την εξέλιξη της σχεδίασης και τη ροή της πληροφορίας σε όλα της τα στάδια.
- Είναι ανεξάρτητη πλατφόρμα σχεδίασης (Απαραίτητη προϋπόθεση είναι το λογισμικό σχεδίασης να υποστηρίζει το πρότυπο ISO10303 STEP).
- Προσθέτει στο OAM ένα επιπλέον χαρακτηριστικό, αυτό της τμηματικής σχεδίασης μια και η ομαδοποίηση που προτείνεται στον ΠΔΣ αρχιτεκτονικής (συστατικών) δίνει ομάδες συναρμολογήσεων με ισχυρή αλληλεπίδραση και εξάρτηση. Αυτή η δυνατότητα ισχυροποιεί τη συνεργαζόμενη σχεδίαση (collaborative design).

Επιγραμματικά, το προτεινόμενο μοντέλο αποτελεί μια πληροφοριακή πλατφόρμα διαχείρισης της πληροφορίας σε όλες τις φάσεις σχεδιασμού ενός προϊόντος με τα παρακάτω χαρακτηριστικά:

- Παρέχει υποστήριξη για το σχεδιασμό και τη δημιουργία προϊόντων σε όλες τις φάσεις της ζωής τους (PLM).
- Είναι ανεξάρτητο πλατφόρμα σχεδίασης.
- Παρέχει βασικού επιπέδου αναπαράσταση πληροφορίας κατά το σχεδιασμό.
- Προτείνει βέλτιστο αριθμό ομάδων συναρμολόγησης, βασιζόμενο σε στοιχεία κατά τη φάση του σχεδιασμού.

Επεκτάσεις του μοντέλου

Το μοντέλο OAM2DSM θα μπορούσε να αποτελέσει τη βασική πλατφόρμα για τη δημιουργία ενός πληροφοριακού συστήματος που θα ενσωματώνει όλα εκείνα τα χαρακτηριστικά που είναι απαραίτητα για τη πλήρη αποτύπωση της διαδικασίας σχεδίασης και συναρμολόγησης ενός προϊόντος. Συγκεκριμένα, το μοντέλο θα μπορούσε να εμπλουτιστεί με τα παρακάτω χαρακτηριστικά:

1. Ενσωμάτωση αλγορίθμου ο οποίος να προσδίδει σημασιολογικά χαρακτηριστικά στα χαρακτηριστικά συναρμολόγησης. Συγκεκριμένα, όπως έχει αναφερθεί στην ενότητα 3.3.4 στην οποία περιγράφεται η εξαγωγή χαρακτηριστικών συναρμολόγησης από το αρχείο STEP, θα μπορούσε η γεωμετρική αποτύπωση των εξαρτημάτων ενός προϊόντος να αναγνωρίζεται και η συσχέτιση χαρακτηριστικών συναρμολόγησης να γίνεται μεταξύ των ίδιων των εξαρτημάτων. Αυτή η δυνατότητα αποτυπώνεται προγραμματιστικά με την έννοια της κλάσης των αντικειμένων και την ιεραρχίας στη συναρμολόγηση.

2. Αποτύπωση των κινηματικών δεδομένων καθώς και των δεδομένων ανοχών στο μοντέλο OAM2DSM, ολοκληρώνοντας έτσι την πλήρη αποτύπωση του μοντέλου ανοικτής συναρμολόγησής του προτείνει ο NIST.
3. Ενσωμάτωση αλγορίθμων ομαδοποίησης των εξαρτημάτων όχι μόνο ως προς των αριθμό των ομάδων που επιλέγει ο χρήστης αλλά και με άλλα χαρακτηριστικά όπως για παράδειγμα των αριθμό των χαρακτηριστικών συναρμολόγησης κάθε εξαρτήματος ή το κόστος πιθανής αποσυναρμολόγησης σε μελλοντική φάση.

Όλες αυτές οι επεκτάσεις που περιγράφονται θα έδιναν στο μοντέλο μια ολοκληρωμένη μορφή, ενσωματώνοντας την ανοικτή αρχιτεκτονική που προτείνεται στο μοντέλο NIST/CPM-OAM και τη δυναμική του ΠΔΣ που αποτυπώσει με σαφήνεια τη ροή πληροφοριών μεταξύ των διάφορων στοιχείων που αποτελούν ένα σύστημα – προϊόν.

5. Βιβλιογραφία

1. **Yang W. Z., Xie S. Q., Ai, Q. S., Zhou Z. D.** 2007,"Recent Development on Integrated Product Modelling: A Review.". *International Journal of Production Research*. Vol 46,No 21.Pp. 6055-6085
2. **Smith R.P., Morrow J. A.** 1999,"Product development process modeling". *Design Studies*. Vol 20, No 3, pp. 237-261.
3. **Wilson, James M.** 2003,"Gantt charts: A centenary appreciation". *Science Direct,European Journal of Operational Research*. No 149,pp.430–437.
4. **Shi Q., Blomquist T.,** 2012, "A new approach for project scheduling using fuzzy dependency structure matrix". Elsevier. Vol 30, No 4, pp. 503-510.
5. **Hall N.G,** 2012, "Project management: Recent developments and research opportunities". Systems Engineering Society of China & Springer-Verlag Berlin Heidelberg. Vol 21, No 2, pp. 129-143
6. **Deng X., Huet G., Tan T., Fortin C.,**2012," Product decomposition using design structure matrix for intellectual property protection in supply chain outsourcing", Elsevier-Computers in Industry. Vol 63, No 6, pp. 632–641.
7. **Eppinger S., Whitney D., Smith R., Gebala D.** 1994,"A model-based method for organizing tasks in product development". *Research in Engineering Design*. Vol 6 No 1, pp.1-13.
8. **Karniel A, Reich Y.** 2013, "Multi-level modelling and simulation of new product development processes", *Journal of Engineering Design*, Taylor & Francis. Vol 24, No 3, pp:185–210
9. **Huang Y., Chen C-H., Khoo L. P.,**2012," Kansei clustering for emotional design using a combined design structure matrix", Elsevier - *International Journal of Industrial Ergonomics*. Vol 42, No 5, pp. 416-427.
10. **Browning T.R.** 2014, "Managing complex project process models with a process architecture framework" ,Elsevier - *International Journal of Project Management*. Vol 32, No 2, pp. 229-241
11. **Browning T. R.** 2001,"Applying the Design Structure Matrix to System Decomposition and Integration Problems:A Review and New Directions". *IEEE transaction on engineering managment*. Vol 48, No 3.
12. **Eppinger S.D., Pimmler T.U.** 1994,"Integration Analysis of Product Decompositions". In *T. K. Hight and F. Mistree, editor, Design theory and methodology, DTM '94*. ISBN: 0-7918-1282-0.
13. **Warren N., Thomas R. J.** 2000,"Applying the Design Structure Matrix to High-Level Organisational Architectures". *2nd Design Structure Matrix Workshop, 18 and 19 September 2000*. Massachusetts Institute of Technology, Cambridge, MA.
14. **Morelli M. D., Eppinger S. D., Gulati R. K.** 1995,"Predicting Technical Communication in Product Development Organizations.". *IEEE Transactions on Engineering Management*. Vol 42, No 3, pp.215-222.
15. **Grose D.,** 1994,"Reengineering the Aircraft Design Process". *Technical Report,Document number: A94-36228 12-66,5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and ,Optimization, Panama City Beach, FL, 07.-09.09.1994,Washington, DC*. ISBN: 1-563-47097-7.
16. **Krishnan V., Eppinger S. D., Whitney D. E.** 1997,"Simplifying Iterations in Cross-Functional Design Decision Making". *Journal of Mechanical Design*. Vol 119, No 4, pp.485-493.

17. **Yassine A. A.**, 2004, "An Introduction to Modeling and Analyzing Complex Product Development". *Italian Management Review, Invited paper*.
18. **Wang A., Kusiak J.** 1993, "Efficient Organizing of Design Activities". *International Journal of Production Research*. Vol 31, No4, pp.753-769.
19. **Tang D., Zheng L., Li L., Li D., Zhang D.** 2000, "Re-engineering of the Design Process for Concurrent Engineering". *Computers & Industrial Engineering*. Vol 38, No 4, pp.479-491.
20. **Eppinger S.D., Gebala D.A.** 1991, "Methods for Analyzing Design Procedures.". In L. A. Stauffer, editor, *Design theory and methodology, DTM '91*, Publisher: ASME, New York, NY. Vol 31, pp.227-233.
21. **Eppinger S.D., McCord K. R.** 1993, "Managing the Integration Problem in Concurrent Engineering". *Massachusetts Institute of Technology, Sloan School of Management, Publisher: Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA*.
22. **Zhanga X., Xua Z.**, 2013, "Hesitant fuzzy agglomerative hierarchical clustering algorithms", *International Journal of Systems Science*, Vol 46, No 3, pp.562–576.
23. **Fernandez C. I.** 1998, "Integration Analysis of Product Architecture to Support Effective Team Co-Location.". *Cambridge: Massachusetts Institute of Technology, Master Thesis*.
24. **Carrascosa M., Eppinger S. D., Whitney D. E.** 1998, "Using the Design Structure Matrix to Estimate Product Development Time.". *Proceedings of DETC'98 1998 ASME International Design Engineering Technical Conferences September 13-16, 1998, Atlanta, Georgia, USA Volume Design Automation Conference*.
25. **Browning A. A., Yassine and T. R.** 2011, "Analyzing Multiple Product Development Projects Based On Information and Resource Constraints". *Massachusetts Institute of Technology*.
26. **Dong Q., Whitney D.** 2001, "Designing a Requirement Driven Product Development Process.". In: *Proceedings of the DETC 2001: ASME 2001 International Design Engineering Technical Conferences. 13th International Conference on Design Theory and Methodology, Pittsburgh., Pittsburgh: ASME*.
27. **Kusiak A., Larson N., Wang J.** 1994, "Reengineering of Design and Manufacturing Processes.". *Computers and Industrial Engineering*. Vol 26, No 3, pp. 521-536.
28. **Yassine A., Eppinger S., Whitney D.** 1999, "1st Design Structure Matrix Workshop". *Massachusetts Institute of Technology, Cambridge, MA*.
29. **Smith R. P., Eppinger S.D.,** 1997, "A predictive model of sequential iteration in engineering design". *Management Science*. Vol 43, No 8, pp. 1104–1120.
30. **Eppinger S. D., Nukala M. V., Whitney D. E.** 1997, "Generalized models of design iteration using signal flow graphs". *Research in Engineering Design*. Vol 9, No 2, pp. 112–123.
31. **Belhe U., Kusiak.** 1996, "A Modeling relationships among design activities",. *Journal of Mechanical Design*. Vol 118, No 4, pp 454–460.
32. **Bird S. D., Kasper G. M.** 1995, "Problem formalization techniques for collaborative systems". *IEEE Transactions on Systems Man and Cybernetics*. Vol 25, No 2, pp 231–242.
33. **Rechtin E., Maier M. W.** 1997, "The art of systems architecting". *CRC Press, Boca Raton*.
34. **Alexander C.**, 1964, "Notes on the synthesis of form". *Harvard University Press, Cambridge*.

35. **Lewis W. P., Samuel A. E., Field B. W.** 1973, "An example of the application of a systematic method to design". *Operational Research Quarterly*. Vol 24, No 2, pp 217–233.
36. **Smith R. P., Eppinger S.D.** 1997, "Identifying controlling features of engineering design iteration". *Management Science* Vol 43 No 3 (1997) pp 276–293. Vol 43, No 3, pp. 276–293.
37. **Kusiak A., Park K.** 1990, "Concurrent engineering: decomposition and scheduling of design activities". *International Journal of Production Research*. Vol 28, No 10, pp. 1883–1900.
38. **Kusiak A., Wang J.** 1993, "Decomposition of the design process". *Journal of Mechanical Design*. Vol 115, pp. 687–695.
39. **Michelen N. F., Papalambros P. Y.** 1997, "A hypergraph framework for optimal model based decomposition of design problems". *Computational Optimization and Applications*. Vol 8, No 2, pp. 173–195.
40. **Kusiak A., Wang J.** 1993, "Decomposition of the design process". *Journal of Mechanical Design*. Vol 115, pp. 687–695.
41. **Olton B.**, 1988, "Polyhedral dynamics applied to design and project management". *IEE Proceedings*. Part A, Vol 135, No 4, pp. 241–244.
42. **Taylor B. W., Moore L. J.** 1980, "R and D project planning with Q-GERT network modeling and simulation". *Management Science*. Vol 26, No 1, pp 44–59.
43. **Pritsker A.A.B.**, 1996, "GERT: Graphical Evaluation and Review Technique". *National Aeronautics and Space Administration under Contract No. NASr-21*.
44. **Adler P. S., Mandelbaum A., Nguyen V., Schwerer E.** 1996, "From project to process management: an empirically-based framework for analyzing product development time". *Management Science*. Vol 41, No 3, pp 458–484.
45. **Ha A. Y., Porteus E. L.** 1995, "Optimal timing of reviews in concurrent design for manufacturability". *Management Science*. Vol 41, No 9, pp. 1431–1447.
46. **Koushik M. V., Mookerjee V. S.** 1995, "Modeling coordination in software construction: an analytical approach". *Information Systems Research*. Vol 6, No 3, pp. 220–254.
47. **Khanna T., Iansiti M.** 1997, "Firm asymmetries and sequential R and D: theory and evidence from the mainframe computer industry". *Management Science*. Vol 43, No 4, pp. 405–42.
48. **Nevins D.E., Whitney J. L.** 1989, "Concurrent design of products and processes: a strategy for the next generation in manufacturing". *McGraw Hill, New York*. ISBN-10: 0070463417.
49. **Ait-Sahlia F., Johnson E. Will P.** 1995, "Is concurrent engineering always a sensible proposition?". *IEEE Transactions on Engineering Management*. Vol 42, No 2, pp. 166–170.
50. **Smith R. P., Eppinger S.D.** 1998, "Deciding between sequential and concurrent tasks in engineering design". *Concurrent Engineering: Research and Applications*. Vol 6, No 1, pp. 15–25.
51. **Harary F., Jessop W. N., Stringer J., Luckman J.** 1968, "An algorithm for project development". *Nature*. Vol 206, p 118.
52. **Blandford S., Hope R. P.** 1985, "Systematic methods for the problem solving process with particular reference to design". *IEE Proceedings*. Part A, Vol 132, No 4, pp. 199–212.
53. **Ulrich K. T., Eppinger S. D.** 1995, "Product Design and Development". *McGraw-Hill*.
54. **Whitney, E. D.** 1996, "The Potential for Assembly Modeling in Product Development and Manufacturing". <http://web.mit.edu/ctpid/www/whitney/papers.html>.

55. **Kennicott P. R.**, 1994,"ISO TC 184/SC4: Product Data Representation and Exchange, Part: 44, Title: Industrial Automation Systems and Integration Product Data Representation and Exchange – Integrated Generic Resources: Product Structure Configuration". *ISO*.
56. **Sugimura N., Ohtaka A.** 2000,"ISO TC 184/SC4/WG12 N597, JNC Proposal of STEP Assembly Model for Products (June 2000)". *ISO*.
57. **MOKA.** "MOKA: A Framework for Structuring and Representing Engineering".<http://www.kbe.coventry.ac.uk/moka/miginfo.htm>. [Electronic]
58. **Whitney D. E., Mantripragada R.** 1998,"The Datum Flow Chain: A Systematic Approach to Assembly Design and Modeling". *ASME Design Engineering Technical Conferences and Computers in Engineering Conference*.
59. **Whitney D. E.**, 2004,"Mechanical Assemblies: Their Design, Manufacture and Role in Product Development.". *Oxford University Press, Oxford*. ISBN-13:9780195157826.
60. **Lee K., Gossard D. C.** 1985,"A Hierarchical Data Structure for Representing Assemblies: Part 1,". *CAD*. Vol 17, No 1, pp. 15–19.
61. **Net, Van der.** 1998,"Designing and Manufacturing Assemblies,". *Ph.D thesis, Eindhoven University of Technology*.
62. **Noort A., Hoek G. F. M., Bronsvort W. F.** 2002,"Integrating Part and Assembly Modeling". *CAD*. Vol 34, No 12, pp. 899–912.
63. **Sudarsan R., Young-Hyun H., Feng S. C., Roy U., Fujun W., Sriram R. D., Lyons K. W.** 2003,"Object-oriented Representation of Electro-Mechanical Assemblies Using UML,". *National Institute of Standards and Technology, NISTIR 7057, Gaithersburg, MD 20899, USA*.
64. **Baysal M. M., Roy U., Sudarsan R., Sriram R. D., Lyons K. W.** 2004,"The open assembly model for the exchange of assembly and tolerance information: overview and example.". *Proceedings of The 2004 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*.
65. **Pavel, Berkhin.** 2006,"Survey of Clustering Data Mining Techniques ". *citeseerx*.
66. **Arabie L. J., Hubert P.** 1996,"Clustering and Classification". *World Scientific Publishing Co., NJ*. pp. 5-63.
67. **Massart D., Kaufman L.** 1983,"The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis.". *John Wiley & Sons, New York, NY*.
68. **Duda R., Hart P.** 1973," Pattern Classification and Scene Analysis." . *John Wiley & Sons, New York, N.Y.*
69. **Jain A. K., Mao J.** 1996,"Artificial neural networks: A tutorial.". *IEEE Computer*. Vol 29, No 3, pp. 31-44.
70. **Scott D. W.**, 1992,"*Multivariate Density Estimation*". *Wiley, New York, NY*. : s.n. 0-471-54770-0.
71. **Gersho A., Gray R. M.** 1992," Vector Quantization and Signal Compression Communications and Information Theory.". *Kluwer Academic Publishers, Norwell, MA*.
72. **Ester M., Kriegel H-P., Sander, J., XU X.** 1996,"A density-based algorithm for discovering clusters in large spatial databases with noise.". *In Proceedings of the 2nd ACM SIGKDD, Portland, Oregon*. pp. 226-231.
73. **Cooley R., Mobasher B., Srivasta J.** 1999", Data preparation for mining world wide web browsing.". *Journal of Knowledge Information Systems*. Vol 1, No 1, pp.5-32.
74. **Ben-Dor A., Yakhini Z.** 1999," Clustering gene expression patterns. ". *In Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB 99)*, 11-14, Lyon, France.
75. **Everitt B., Landau S., Leese M.** 2011,"Cluster Analysis". *London, Arnold*.

76. **Hansen P., Jaumard B.** 1997, "Cluster analysis and mathematical programming". *Math Program.* Vol 79, pp. 191–215.
77. **Xu, Rui.** 2005, "Survey of Clustering Algorithms". *IEEE Transtaction on neural networks.* Vol 16, No 3.
78. **Halkidi M., Batistakis Y., Vazirgiannis M.** 2002, "Cluster validity methods: Part I & II.". *SIGMOD Record.* Vol ,Vol 31, No 2-3.
79. **J., MacQueen.** 1967, "Some methods for classification and analysis of multivariate observations". in *Proc. 5th Berkeley Symp.* Vol 1.
80. **A. Jain, R. Duin, and J. Mao,.** 2000, "Statistical pattern recognition: A review",. *IEEE Trans. Pattern Anal. Mach. Intel.* Vol 22, No 1, pp. 4–37.
81. **Jain A., Murty M., Flynn P.** 1999, "Data clustering: A review". *ACM Computer Survey.* Vol 31, No. 3, pp. 264–323.
82. **Kleinberg J.,** 2002, "An impossibility theorem for clustering,". in *Proc. 2002 Conf. Advances in Neural Information Processing Systems.* Vol 15, pp. 463–470.
83. **Jain A.K., Dubes R.C.** 1988, "Algorithms for Clustering Data". *Englewood Cliffs, NJ: Prentice-Hall.* ISBN:0-13-022278-X.
84. **Fenves, S. J.** 2001, "A Core Product Model For Representing Design Information". *National Institute of Standards and Technology, NISTIR 6736, Gaithersburg, MD 20899, USA.*
85. **Szykman S., Fenves S. J., Keirouz W. T., Shooter S.** 2001, "A foundation for interoperability in next-generation product development systems.". *Computer-Aided Design.* Vol 33, No 7, pp. 545-559.
86. **[Electronic], Wikipedia.** <http://el.wikipedia.org/wiki/UML>.
87. **Qing Li, Yu-Liu Chen.** 2009, "Modeling and Analysis of Enterprise and Information Systems From Requirements to Realization". *Springer Berlin Heidelberg.*
88. **Chen P.,** 1976, "The Entity-Relationship Model: Toward a Unified View of Data,". *ACM Transactions on Database Systems.* Vol 1, No 1, pp. 9-36.
89. **Spinellis.D.,** 2008, "UML courses". <http://www.dmst.aueb.gr/dds/ism/oo/indexw.htm>.
[Electronic]
90. **Booch G., Rumbaugh J., Jacobson I.** 1997, "The Unified Modeling Language User Guide". *Addison-Wesley.* ISBN: 0-201-57168-4.
91. **Fenves S.J., Fofou S., Bock C., Sudarsan R., Bouillon N., Sriram R.D.** 2004, "CPM 2: A revised core product model for representing". *NISTIR 7185 U.S. Department of Commerce.*
92. **Ho T., Tang C.** 1998, "Product Variety Management". *Kluwer Academic Publishers, Boston, MA.* ISBN:0-7923-8226-9.
93. **Fenves S.J., Choi Y., Gurumoorthy B., Mocko G., Sriram R. D.** 2003, "Master Product Model for the Support of Tighter Design-Analysis Integration,". *National Institute of Standards and Technology, Gaithersburg, MD 20899, NISTIR 7004.*
94. **Sudarsan R., Fenves S.J., Sriram R.D., Wang F.** 2005, "A product information modeling framework for product lifecycle management.". *Computer-Aided Design* 37. Vol 37, pp.1399–1411.
95. **OMG.** 2004, Model-Driven Architecture. <http://www.omg.org/mda/>. [Electronic]
96. **Sudarsan R., Baysal M. M., Roy U., Fofou S., Bock C., Fenves S. J., Eswaran E., Lyons K. W., Sriram R. D.** 2004, "Information Models for Product Representation: Core and Assembly Models.". *NISTIR 7173, NIST, Gaithersburg, MD 20899,.*
97. **Sudarsan R., Young-Hyun H., Fofou S., Feng S. C., Roy U., Fujun W., Sriram R. D., Lyons K. W.,** 2005, "A Model for Capturing Product Assembly Information.". *Journal of Computing and Information Science in Engineering.*

98. **Patil L., Dutta D., Sriram R. D.,** 2005, "Ontology-based Exchange of Product Data Semantics,". *IEEE Transactions on Automation Science and Engineering*. Vol 2, No 3, pp. 213-255.
99. **Wang F., Fenves S. J., Sudarsan R., Sriram R. D.,** 2003, "Towards Modeling the Evolution of Product Families,". *Proceedings of the 2003 ASME Design Engineering Technical Conferences, Chicago, IL*.
100. **Biswas A., Fenves S. J., Shapiro V., Sriram R. D.,** 2005, "Representation of Heterogeneous Material Properties in the Core Product Model." *Engineering with Computers*.
101. **Xu C., Gupta S. K., Yao Z., Gruninger M., Sriram R. D.,** 2005, "Toward Computer-Aided Conceptual Design of Mechatronic Devices with Multiple Interaction-States,". *Manufacturing System Integration Division, National Institute of Standards and Technology, Gaithersburg, MD*.
102. **Zha X. F., Fenves S. J., Sriram R. D.** 2005, "A Feature-based Approach to Embedded System Hardware and Software Co-Design." *Manufacturing System Integration Division, National Institute of Standards and Technology, Gaithersburg, MD 20899*.
103. **De Fazio T. L., Whitney D. E.** 1987, "Simplified generation of all mechanical assembly sequences." *IEEE Journal of Robotics and Automation*. Vol 3, No 6, pp. 640–658.
104. **Gottipolu R. B., Ghosh K.** 1997, "Representation and selection of assembly sequences in computer-aided assembly process planning." *International Journal of Product Research*. Vol 35, No 12, pp. 3447–3465.
105. **Liu, Shiang-Fong Chen, Yong-Jin.** 2001, "An adaptive genetic assembly-sequence planner". *INT. J. Computer Integrated Manufacturing*. Vol 14, No 5, pp.489–500.
106. **Tomas B.A.,** 1992, "Designing Quality Databases with IDEF1X Information Models." *Dorset House Publishing Company Incorporated, New York, USA,*.
107. **Teorey J. T., Dongqing Y., James P. F.** 1986, "A logical design methodology for relational database using the extended entity-relationship model." *Computing Survey*. Vol 18, No 2, pp. 197-222.
108. **Eunjee Song, Shuxin Yin, Indrakshi Ray.** 2007, "Using UML to Model Relational Database Operations". *Computer Standards & Interfaces*. Vol 29, pp.343–354.
109. **Sparks, Geoffrey.** 2001, "Database Modelling in UML". *Methods & Tools*.
110. **Arun Kumar P., Anand S., Deshpande A. C., Kumar S.** 2008, "A System for Extracting Product Features from CAD Models – A STEP Approach." *Contemporary Engineering Sciences*. Vol 1, No 3, pp. 139 - 146.
111. **Ketan H. S., Haleel A. J.** 2013, "STEP- Based Assembly Feature Recognition Using Attribute Adjacency Graph for Prismatic Parts. ". *Eng. &Tech. Journal*. Vol 31, Part (A), No.10.
112. **Holland W.V., Bronsvoort W. F.** 2000, "Assembly Features In Modeling and Planning". *Robotic and Computer Integrated Manufacturing, Elsevier Science*. Vol 16, pp.277-294.
113. **Hamidullah B. E., Irfan M. A.** 2006, "Assembly Features: Definition, Classification, and Instantiation". *2nd International Conference on Emerging Technologies, Peshawar, Pakistan, IEEE-ICET,*.
114. **Zha X.F., Du H.** 2002, "A PDES\STEP Based Model and System for Concurrent Integrated Design and Assembly Planning". *Computer Aided Design*, 34, 1087-1110, Elsevier Science Ltd.,.
115. **Chan C.K., Tan S.T.** 2003, "Generating Assembly Features onto Split Solid Models". *Computer-Aided Design, Elsevier Ltd*. Vol 35, pp.1315–1336.
116. **Pan C.,** 2005, "Integrating CAD Files and Automatic Assembly Sequence Planning". Ph.D. Thesis. Iowa State University.

117. **Zheng Chen, Heng Ji.** 2010,"Graph-based Clustering for Computational Linguistics: A Survey". *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing,Uppsala, Sweden.* pp.1-9.
118. **Hwai-En Tseng, Chien-Chen Chang , Chih-Jen Cheng.**2010,"Disassembly-oriented assessment methodology for product modularity". *International Journal of Production Research.* Vol 48, No 14, pp.4297–4320.
119. **Dunn J.,** 1974,"Well separated clusters and optimal fuzzy partitions". *J. Cybern.* Vol 4, pp. 95-104.
120. **Biswas N. R, Pal J.** 1997,"Cluster validation using graph theoretic concepts". *Elsevier Science.* Vol 30, No 6, pp.847-857.
121. **Aho A.V., Ullman J. D.** 1992,"Chapter 10. Patterns, Automata, and Regular Expressions". s.l. : Foundations of Computer Science.
122. **Aho A.V., Sethi R.,Ullman J. D.** 1986",Compilers: Principles, Techniques, and Tools.". *Addison-Wesley, Reading MA,ISBN 0-201-10088-6.*

Παράρτημα Ι – Αλγόριθμος Ομαδοποίησης

Ο αλγόριθμος που υλοποιεί την ομαδοποίηση σε Microsoft® Visual Studio 2010 Premium και .NET Framework Version 4.

```
PublicSub Run_clustering()
Dim i, j, k, l, m As Integer
Dim Cluster_met As Integer

Dim temp_cost As Integer
If OpenFileDialog = False Then
    MsgBox("You must open an assembly file before clustering")
Return
EndIf

Cluster_AV_Cost = Cluster_AV_Cost / Convert.ToInt32(TextBox1.Text)
'MsgBox(Cluster_AV_Cost)
ReDim Cluster_Cost(Convert.ToInt32(TextBox1.Text))
DataGridView5_Setup()

Dim FN As String

FN = OpenFileDialog1.FileName

FN = FN.Replace(".xlsx", "_NoC=" + TextBox1.Text + "_out.txt")
Dim swWriter AsNewStreamWriter(FN)
Dim s_txt AsString
Output("File Name:" + FN, swWriter)
Output("Number of Clusters :" + TextBox1.Text, swWriter)
Output("_____ ", swWriter)

For Cluster_met = 1 ToConvert.ToInt32(TextBox1.Text)
    SR = Find_Max_PI()
    DataGridView3.Item(2, SR - 1).Style.BackColor = Color.Yellow
    'Put in cluster
    PI(SR, 4) = Cluster_met 'Cluster no
    PI(SR, 1) = 0
    DataGridView4.Rows(SR - 1).Cells(2).Value() = PI(SR, 1)
    PI(SR, 2) = 1 'Clustered =1
    PI(SR, 3) = 0 'HI =0
    DataGridView4.Rows(SR - 1).Cells(3).Value() = PI(SR, 2)
    DataGridView4.Rows(SR - 1).Cells(4).Value() = PI(SR, 3)
    DataGridView4.Rows(SR - 1).Cells(5).Value() = PI(SR, 4)
    Final_Array(SR, 4) = Cluster_met
    temp_cost = Cost(SR)
    Output(" ", swWriter)
    Output("Cluster NO:" + Str(Cluster_met) + " Leader:" + Str(SR) + " Initial Cost:" +
    Str(temp_cost), swWriter)
```



```

For i = 1 To NofLines

'Compute HI
For m = 1 To NofLines
    HI = 0
For j = 1 To NofLines ' And PI(m, 2) = 0
If AA(SR, j) * AA(m, j) = 1 And m <> SR And PI(m, 2) = 0 Then
    HI = HI + 1
EndIf
Next

    PI(m, 3) = HI
'Output("Cluster :" + Str(Cluster_met) + "Ass:" + Str(m) + " --> " + Str(PI(m, 3)), swWriter)
    DataGridView4.Rows(m - 1).Cells(4).Value() = PI(m, 3)
Next

For k = 1 To NofLines
    SR_HI = Find_Max_HI()
'Output("MAX HI is:" + Str(SR_HI), swWriter)
If AA(SR_HI, k) * AA(SR, k) = 1 And PI(k, 2) = 0 And temp_cost + Cost(k) <= Cluster_AV_Cost Then
    PI(k, 4) = Cluster_met
    PI(k, 2) = 1
    PI(k, 3) = 0
    PI(k, 1) = 0
    DataGridView4.Rows(k - 1).Cells(5).Value() = PI(k, 4)
    DataGridView4.Rows(k - 1).Cells(3).Value() = PI(k, 2)
    DataGridView4.Rows(k - 1).Cells(2).Value() = PI(k, 1)
    temp_cost = temp_cost + Cost(k)
    Final_Array(i, 4) = Cluster_met
    Output("assembly In:" + Str(k) + " Current Total Cost:" + Str(temp_cost),
swWriter)
'Delete Current Assembly In
For j = 1 To NofLines
    AA(j, k) = 0
    AA(k, j) = 0
Next
EndIf

Next
Next

    Output("*****Pass 2 - Unclustered Elements*****", swWriter)

    k = 1
DoWhile k <= NofLines And temp_cost < Cluster_AV_Cost
If PI(k, 2) = 0 And temp_cost + Cost(k) <= Cluster_AV_Cost Then
For i = 1 To NofLines
For j = 1 To NofLines

```

```

If AA_BackUp(k, j) * AA_BackUp(i, j) = 1 And PI(j, 4) = Cluster_met And PI(k, 2) = 0
Then
    PI(k, 4) = Cluster_met
    PI(k, 2) = 1
    PI(k, 1) = 0
    PI(k, 3) = 0
    DataGridView4.Rows(k - 1).Cells(5).Value() = PI(k, 4)
    DataGridView4.Rows(k - 1).Cells(3).Value() = PI(k, 2)
    DataGridView4.Rows(k - 1).Cells(2).Value() = PI(k, 1)
    temp_cost = temp_cost + Cost(k)

    Final_Array(k, 4) = Cluster_met
    Output("assembly In:" + Str(k) + " Current Total Cost:" +
Str(temp_cost), swWriter)
For m = 1 To NofLines
    AA(m, k) = 0
    AA(k, m) = 0
Next
EndIf
Next
Next
EndIf

    k = k + 1
Loop
For j = 1 To NofLines
    AA(SR, j) = 0
    AA(j, SR) = 0
Next

'INITIALIZE PI(i,1)
For k = 1 To NofLines
For l = 1 To NofLines
    PI(k, 1) = 0
Next
Next

For k = 1 To NofLines
For l = 1 To NofLines
If AA(k, l) = 1 Then
    PI(k, 1) = PI(k, 1) + 1
EndIf
Next

    oSheet.cells(k, 4 + Cluster_met).value = PI(k, 1)
Next

Cluster_Cost(Cluster_met) = temp_cost
Output("_____ ",
swWriter)
Output(" Final Cost:" + Str(temp_cost), swWriter)

```

```

        Output("_____ ",
swWriter)
Next

        Output("*****Pass 3 - Unclustered Elements*****", swWriter)
Dim min_cluster AsInteger = Cluster_Cost(1)
Dim Cluster_No AsInteger = 1
For i = 2 ToConvert.ToInt32(TextBox1.Text)
If min_cluster > Cluster_Cost(i) Then
    min_cluster = Cluster_Cost(i)
    Cluster_No = i
EndIf
Next

        Output(" Cluster with min cost is " + Str(Cluster_No) + " Cost:" +
Str(Cluster_Cost(Cluster_No)), swWriter)
        Output(" Unclustered Elements ", swWriter)

For i = 1 To NofLines
If PI(i, 4) = 0 Then
    Output(DataGridView4.Rows(i - 1).Cells(1).Value() + " Edge:" + Str(i), swWrit-
er)
EndIf
Next
'START PASS 3
        Output("*****Start Clustering Pass 3 *****", swWriter)
Dim Selected_Cluster AsInteger'The cluster which put the element, min cost
Dim min_cost AsInteger
For k = 1 To NofLines
    Selected_Cluster = 0 'Initial Cluster
    min_cost = 9999
If PI(k, 2) = 0 Then
For j = 1 To NofLines
If AA_BackUp(k, j) = 1 And PI(j, 2) = 1 Then
    Output("-Assembly :" + Str(k) + " belongs to :" + Str(PI(j, 4)) + "(" +
Str(Cluster_Cost(PI(j, 4))) + ")", swWriter)
If min_cost > Cluster_Cost(PI(j, 4)) Then
        Selected_Cluster = PI(j, 4) '!!
        min_cost = Cluster_Cost(PI(j, 4))
EndIf
EndIf
Next
'Put the element on cluster selected_cluster
If Selected_Cluster <> 0 Then

    PI(k, 4) = Selected_Cluster
    PI(k, 2) = 1
    PI(k, 1) = 0
    PI(k, 3) = 0
    DataGridView4.Rows(k - 1).Cells(5).Value() = PI(k, 4)

```

```

        DataGridView4.Rows(k - 1).Cells(3).Value() = PI(k, 2)
        DataGridView4.Rows(k - 1).Cells(2).Value() = PI(k, 1)
        Cluster_Cost(Selected_Cluster) = Cluster_Cost(Selected_Cluster) + Cost(k)
        Output("Clustering " + " Edge:" + Str(k) + " In Cluster: " +
Str(Selected_Cluster), swWriter)
        Output(" ", swWriter)
    EndIf
EndIf
Next
'Print FINAL COST for each cluster
    Output("-----FINAL COST FOR EACH CLUSTER-----", swWriter)
'Find max Cluster Cost in order to fill Datagridview5
Dim Max_distance AsInteger = Cluster_Cost(1)

For i = 2 ToConvert.ToInt32(TextBox1.Text)
If Max_distance < Cluster_Cost(i) Then
    Max_distance = Cluster_Cost(i)
EndIf
Next
For i = 1 ToConvert.ToInt32(TextBox1.Text)
    Output("Cluster NO: " + Str(i) + " COST: " + Str(Cluster_Cost(i)), swWriter)
    DataGridView5.Rows(i).Cells(1).Value = Cluster_Cost(i)
    DataGridView5.Rows(i).Cells(2).Value = Max_distance - Cluster_Cost(i)
Next
Dim Max_Value AsInteger = DataGridView5.Rows(1).Cells(2).Value
Dim MAx_Distance_Row AsInteger = 1
For i = 1 ToConvert.ToInt32(TextBox1.Text)
If Max_Value < DataGridView5.Rows(i).Cells(2).Value Then
    Max_Value = DataGridView5.Rows(i).Cells(2).Value
    MAx_Distance_Row = i
EndIf
Next
    DataGridView5.Rows(MAx_Distance_Row).Cells(2).Style.BackColor = Color.Red

    Output("-----", swWriter)
    Output("MAX TIME DIF: " + Str(Max_Value), swWriter)

    swWriter.Flush()
    swWriter.Close()
    Button4.Enabled = True
    Color_graph()
oBook.save()
    oSheet = Nothing
    oBook = Nothing
    oExcel.Quit()
    oExcel = Nothing
GC.Collect()
EndSub

```

Παράρτημα II – Αρχεία καταγραφής για την ομαδοποίηση

N=2

File-

Name: C:\DocumentsandSettings\Dermitz\Ταέγγραφάμου\Dropbox\Phd\PhdDoc\PCCaseStudy\N=2\PCDSMW_NoC=2_out.txt

Number of Clusters :2

Cluster NO: 1 Leader: 1 Initial Cost: 15
 assembly In: 2 Current Total Cost: 28
 assembly In: 12 Current Total Cost: 39
 assembly In: 13 Current Total Cost: 47
 assembly In: 14 Current Total Cost: 55
 assembly In: 15 Current Total Cost: 63
 assembly In: 16 Current Total Cost: 71
 assembly In: 17 Current Total Cost: 79
 assembly In: 18 Current Total Cost: 87
 assembly In: 19 Current Total Cost: 95
 assembly In: 20 Current Total Cost: 103
 assembly In: 4 Current Total Cost: 116
 assembly In: 5 Current Total Cost: 130
 assembly In: 6 Current Total Cost: 143
 assembly In: 7 Current Total Cost: 156
 assembly In: 8 Current Total Cost: 169
 *****Pass 2 - Unclustered Elements*****

Final Cost: 169

Cluster NO: 2 Leader: 25 Initial Cost: 11
 assembly In: 21 Current Total Cost: 22
 assembly In: 22 Current Total Cost: 33
 assembly In: 23 Current Total Cost: 44
 assembly In: 24 Current Total Cost: 55
 assembly In: 26 Current Total Cost: 66
 assembly In: 9 Current Total Cost: 79
 assembly In: 31 Current Total Cost: 90
 *****Pass 2 - Unclustered Elements*****
 assembly In: 3 Current Total Cost: 104
 assembly In: 10 Current Total Cost: 117
 assembly In: 11 Current Total Cost: 128
 assembly In: 27 Current Total Cost: 142
 assembly In: 28 Current Total Cost: 153
 assembly In: 29 Current Total Cost: 164
 assembly In: 30 Current Total Cost: 175

Final Cost: 175

*****Pass 3 - Unclustered Elements*****

Cluster with min cost is 1 Cost: 169

Unclustered Elements

15-- 16 Edge: 32

*****Start Clustering Pass 3 *****

-Assembly : 32 belongs to : 2(175)

-Assembly : 32 belongs to : 1(169)

-Assembly : 32 belongs to : 2(175)

Clustering Edge: 32 In Cluster: 1

-----FINAL COST FOR EACH CLUSTER-----

Cluster NO: 1 COST: 180.25

Cluster NO: 2 COST: 175

MAX TIME DIF: 5

N=3

File-

Name:C:\DocumentsandSettings\Dermitz\Ταέγγραφάμου\Dropbox\Phd\PhdDoc\PCCaseStudy\N=3\PCDSMW_NoC=3_out.txt

Number of Clusters :3

Cluster NO: 1 Leader: 1 Initial Cost: 15

assembly In: 2 Current Total Cost: 28

assembly In: 12 Current Total Cost: 39

assembly In: 13 Current Total Cost: 47

assembly In: 14 Current Total Cost: 55

assembly In: 15 Current Total Cost: 63

assembly In: 16 Current Total Cost: 71

assembly In: 17 Current Total Cost: 79

assembly In: 18 Current Total Cost: 87

assembly In: 19 Current Total Cost: 95

assembly In: 20 Current Total Cost: 103

assembly In: 4 Current Total Cost: 116

*****Pass 2 - Unclustered Elements*****

Final Cost: 116

Cluster NO: 2 Leader: 6 Initial Cost: 13

assembly In: 5 Current Total Cost: 26

assembly In: 7 Current Total Cost: 39

assembly In: 8 Current Total Cost: 52

assembly In: 9 Current Total Cost: 65

assembly In: 10 Current Total Cost: 78

assembly In: 22 Current Total Cost: 89

assembly In: 28 Current Total Cost: 100

*****Pass 2 - Unclustered Elements*****

assembly In: 3 Current Total Cost: 114

Final Cost: 114

Cluster NO: 3 Leader: 21 Initial Cost: 11

assembly In: 23 Current Total Cost: 22

assembly In: 24 Current Total Cost: 33

assembly In: 25 Current Total Cost: 44

assembly In: 26 Current Total Cost: 55

*****Pass 2 - Unclustered Elements*****

assembly In: 11 Current Total Cost: 66

assembly In: 27 Current Total Cost: 80

assembly In: 29 Current Total Cost: 91

assembly In: 30 Current Total Cost: 102

assembly In: 31 Current Total Cost: 113

Final Cost: 113

*****Pass 3 - Unclustered Elements*****

Cluster with min cost is 3 Cost: 113

Unclustered Elements

15-- 16 Edge: 32

*****Start Clustering Pass 3 *****

-Assembly : 32 belongs to : 2(114)

-Assembly : 32 belongs to : 1(116)

-Assembly : 32 belongs to : 3(113)

Clustering Edge: 32 In Cluster: 3

-----FINAL COST FOR EACH CLUSTER-----

Cluster NO: 1 COST: 116

Cluster NO: 2 COST: 114

Cluster NO: 3 COST: 124.25

MAX TIME DIF: 10

N=4

File-

Name:C:\DocumentsandSettings\Dermitz\Ταέγγραφάμου\Dropbox\Phd\PhdDoc\PCCaseStudy\N=4\PCDSMW_NoC=4_out.txt

Number of Clusters :4

Cluster NO: 1 Leader: 1 Initial Cost: 15

assembly In: 2 Current Total Cost: 28

assembly In: 12 Current Total Cost: 39

assembly In: 13 Current Total Cost: 47

assembly In: 14 Current Total Cost: 55

assembly In: 15 Current Total Cost: 63

assembly In: 16 Current Total Cost: 71
 assembly In: 17 Current Total Cost: 79
 assembly In: 18 Current Total Cost: 87
 *****Pass 2 - Unclustered Elements*****

Final Cost: 87

Cluster NO: 2 Leader: 6 Initial Cost: 13
 assembly In: 4 Current Total Cost: 26
 assembly In: 5 Current Total Cost: 40
 assembly In: 7 Current Total Cost: 53
 assembly In: 8 Current Total Cost: 66
 assembly In: 9 Current Total Cost: 79
 *****Pass 2 - Unclustered Elements*****

Final Cost: 79

Cluster NO: 3 Leader: 11 Initial Cost: 11
 assembly In: 22 Current Total Cost: 22
 assembly In: 23 Current Total Cost: 33
 assembly In: 24 Current Total Cost: 44
 assembly In: 25 Current Total Cost: 55
 assembly In: 26 Current Total Cost: 66
 assembly In: 20 Current Total Cost: 74
 assembly In: 21 Current Total Cost: 85
 *****Pass 2 - Unclustered Elements*****

Final Cost: 85

Cluster NO: 4 Leader: 10 Initial Cost: 13
 *****Pass 2 - Unclustered Elements*****
 assembly In: 3 Current Total Cost: 26
 assembly In: 32 Current Total Cost: 37

Final Cost: 37

*****Pass 3 - Unclustered Elements*****
 Cluster with min cost is 4 Cost: 37
 Unclustered Elements
 2-- 17 Edge: 19
 4-- 5 Edge: 27
 9-- 11 Edge: 28
 10-- 11 Edge: 29
 12-- 14 Edge: 30
 13-- 14 Edge: 31
 *****Start Clustering Pass 3 *****

-Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 3(85)
 -Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 1(87)
 -Assembly : 19 belongs to : 3(85)
 Clustering Edge: 19 In Cluster: 3

-Assembly : 27 belongs to : 1(87)
 -Assembly : 27 belongs to : 3(93.25)
 Clustering Edge: 27 In Cluster: 1

-Assembly : 28 belongs to : 2(79)
 -Assembly : 28 belongs to : 1(100.5)
 -Assembly : 28 belongs to : 3(93.25)
 Clustering Edge: 28 In Cluster: 2

-Assembly : 29 belongs to : 2(90.25)
 -Assembly : 29 belongs to : 1(100.5)
 -Assembly : 29 belongs to : 3(93.25)
 -Assembly : 29 belongs to : 2(90.25)
 Clustering Edge: 29 In Cluster: 2

-Assembly : 30 belongs to : 2(101.5)
 -Assembly : 30 belongs to : 1(100.5)
 -Assembly : 30 belongs to : 3(93.25)
 Clustering Edge: 30 In Cluster: 3

-Assembly : 31 belongs to : 2(101.5)
 -Assembly : 31 belongs to : 1(100.5)
 -Assembly : 31 belongs to : 3(104.5)
 -Assembly : 31 belongs to : 3(104.5)
 Clustering Edge: 31 In Cluster: 1

-----FINAL COST FOR EACH CLUSTER-----

Cluster NO: 1 COST: 111.75
 Cluster NO: 2 COST: 101.5
 Cluster NO: 3 COST: 104.5
 Cluster NO: 4 COST: 37

 MAX TIME DIF: 75

N=5

File-

Name:C:\DocumentsandSettings\Dermitz\Ταέγγραφάμου\Dropbox\Phd\PhdDoc\PCCaseStu

dy\N=5\PCDSMW_NoC=5_out.txt

Number of Clusters :5

Cluster NO: 1 Leader: 1 Initial Cost: 15

assembly In: 2 Current Total Cost: 28

assembly In: 12 Current Total Cost: 39

assembly In: 13 Current Total Cost: 47

assembly In: 14 Current Total Cost: 55

assembly In: 15 Current Total Cost: 63

assembly In: 16 Current Total Cost: 71

*****Pass 2 - Unclustered Elements*****

Final Cost: 71

Cluster NO: 2 Leader: 11 Initial Cost: 11

assembly In: 22 Current Total Cost: 22

assembly In: 23 Current Total Cost: 33

assembly In: 24 Current Total Cost: 44

assembly In: 25 Current Total Cost: 55

assembly In: 26 Current Total Cost: 66

*****Pass 2 - Unclustered Elements*****

Final Cost: 66

Cluster NO: 3 Leader: 6 Initial Cost: 13

assembly In: 4 Current Total Cost: 26

assembly In: 5 Current Total Cost: 40

assembly In: 7 Current Total Cost: 53

assembly In: 8 Current Total Cost: 66

*****Pass 2 - Unclustered Elements*****

Final Cost: 66

Cluster NO: 4 Leader: 17 Initial Cost: 8

assembly In: 19 Current Total Cost: 16

assembly In: 20 Current Total Cost: 24

assembly In: 31 Current Total Cost: 35

assembly In: 18 Current Total Cost: 43

*****Pass 2 - Unclustered Elements*****

assembly In: 9 Current Total Cost: 56

assembly In: 10 Current Total Cost: 69

Final Cost: 69

Cluster NO: 5 Leader: 21 Initial Cost: 11

*****Pass 2 - Unclustered Elements*****

assembly In: 27 Current Total Cost: 24

Final Cost: 24

*****Pass 3 - Unclustered Elements*****

Cluster with min cost is 5 Cost: 24

Unclustered Elements

1-- 6 Edge: 3

9-- 11 Edge: 28

10-- 11 Edge: 29

12-- 14 Edge: 30

15-- 16 Edge: 32

*****Start Clustering Pass 3 *****

-Assembly : 3 belongs to : 1(71)

-Assembly : 3 belongs to : 1(71)

-Assembly : 3 belongs to : 3(66)

-Assembly : 3 belongs to : 3(66)

-Assembly : 3 belongs to : 3(66)

-Assembly : 3 belongs to : 3(66)

-Assembly : 3 belongs to : 3(66)

-Assembly : 3 belongs to : 4(69)

-Assembly : 3 belongs to : 4(69)

-Assembly : 3 belongs to : 1(71)

Clustering Edge: 3 In Cluster: 3

-Assembly : 28 belongs to : 3(79.5)

-Assembly : 28 belongs to : 1(71)

-Assembly : 28 belongs to : 2(66)

Clustering Edge: 28 In Cluster: 2

-Assembly : 29 belongs to : 3(79.5)

-Assembly : 29 belongs to : 1(71)

-Assembly : 29 belongs to : 2(77.25)

-Assembly : 29 belongs to : 2(77.25)

Clustering Edge: 29 In Cluster: 1

-Assembly : 30 belongs to : 3(79.5)

-Assembly : 30 belongs to : 4(69)

-Assembly : 30 belongs to : 2(77.25)

-Assembly : 30 belongs to : 4(69)

Clustering Edge: 30 In Cluster: 4

-Assembly : 32 belongs to : 4(80.25)

-Assembly : 32 belongs to : 4(80.25)

-Assembly : 32 belongs to : 2(77.25)

Clustering Edge: 32 In Cluster: 2

-----FINAL COST FOR EACH CLUSTER-----

Cluster NO: 1 COST: 82.25

Cluster NO: 2 COST: 88.5

Cluster NO: 3 COST: 79.5

Cluster NO: 4 COST: 80.25

Cluster NO: 5 COST: 24

MAX TIME DIF: 64

Παράρτημα III – Τεκμηρίωση του αλγόριθμου λεξικογραφικής ανάλυσης για το αρχείο STEP/AP203

1. *While Not EOF(StepFile) Do*
Ανοίγει το αρχείο STEP σε ASCII mode και όσο υπάρχουν γραμμές σαρώνεται γραμμή προς γραμμή.
2. *Search for next NAUO_Entity*
Αναζήτηση γραμμών μέσα στο αρχείο οι οποίες περιλαμβάνουν την έκφραση NAUO_Entity όπως αυτή ορίζεται στο STEP/AP203. Η αναζήτηση γίνεται κάνοντας χρήση της τεχνικής των κανονικών εκφράσεων (regular expressions) οι οποίες χρησιμοποιούνται στην υλοποίηση συμβολομεταφραστών και διερμηνέων για την περιγραφή γλωσσών προγραμματισμού. Μια κανονική έκφραση αναζητά μοτίβα χαρακτήρων που ταιριάζουν στην περιγραφή της κανονικής έκφρασης. Περισσότερα για τις κανονικές εκφράσεις στα βιβλία των Aho et.al. (121),(122).
3. *If Find_Next_NAUO_Entity = TRUE*
4. */*Extract from line text the Id,RPD1,RPD2 and put them in CNAUO_Entity*
5. *CNAUO_Entity ← (Id, RPD1, RPD2)*
Κάθε φορά που εντοπίζεται μια νέα οντότητα NAUO_Entity γίνεται εξαγωγή των στοιχείων της, όπως ορίστηκε από τους παραπάνω πίνακες που περιγράφουν τον ορισμό της, και στην συνέχεια δημιουργείται στη βάση δεδομένων των τεχνουργημάτων μια νέα οντότητα.
6. *While Not EOF(StepFile)) and (not Find_Father_Flag) do*
7. */*Search for each NAUO_Entity_i*
 - i. *Get Next NAUO_Entity_i*
 - ii. *If CNAUO_Entity.RPD1 = NAUO_Entity_i.RPD2 Then*
 1. *Find_Father_Flag = True*
 2. *Exit While*
 - iii. *End if*
8. *End While*
Για κάθε νέα οντότητα που εντοπίζεται NAUO_Entity_i ελέγχεται αν υπάρχει συσχέτιση με την αρχική που εντοπίστηκε στο βήμα 5. Αν υπάρχει αυτό αντιμετωπίζεται ως συναρμολόγηση και μάλιστα με γνωστή την ιεραρχία μιας και η αρχική οντότητα σύμφωνα με το ορισμό του STEP αποτελεί τον πρόγονο (άρα πρόκειται για υποσυναρμολόγηση ή η τελική συναρμολόγηση) της οντότητας που πληρεί το κριτήριο της γραμμής 7.ii
9. *If Find_Father_Flag = True Then*
 - a. *MakeATreeNode(NAUO_Entity_i.Id As Father, CNAUO_Entity.Id As Child)*
10. *Else*
 - a. *MakeATreeNode("Root" As Father, CNAUO_Entity.Id As Child)*
11. *End if*
Σε αυτό το σημείο γίνεται η επιλογή αν η οντότητα που βρέθηκε είναι υποσυναρμολόγηση και τότε δημιουργείται ένας ακόμα κόμβος στο δένδρο ιεραρχίας της συναρμολόγησης (βήμα 9.a) ή πρόκειται για τη συνολική συναρμολόγηση, και επομένως για το τελικό προϊόν, οπότε παίρνουμε τη ρίζα του δένδρου συναρμολόγησης (βήμα 10.a). Τέλος, ο αλγόριθμος για κάθε εξάρτημα ενημερώνει τη βάση δεδομένων για την ποσότητα του. Ακολουθεί ο αλγόριθμος σε εκτελέσιμη μορφή (module)

```

Imports System.IO
Imports System.Xml
Module Module1
Public Component_Image_Path AsString
Public Artifact_Image_Path AsString

Const j_Col AsInteger = 6    'Column Dimesnion
Public NAUO_Array(,) AsString
Public Sec_Array(,) AsString 'Keep copies of parts or subassemblies
Public F_Array(,) AsString 'Final Array of NAUO
Public Final_i_dim AsInteger 'Final row dimension

Public NumberOfLines AsInteger = 0
PublicFunction ReadFromStepFile(ByVal FN AsString) AsBoolean
Dim FILE_NAME AsString = FN
Dim objWriter AsNew StreamWriter("c:\test\shape.txt")
Dim objReader AsNew StreamReader(FILE_NAME)
Dim TextLine AsString

'Search for this pattern
Dim pattern AsString = "NEXT_ASSEMBLY_USAGE_OCCURRENCE"
' Dim pattern As String = "=SHAPE_REPRESENTATION"
Dim check AsNew System.Text.RegularExpressions.Regex(pattern)
Dim valid AsBoolean = False

If File.Exists(FILE_NAME) = TrueThen
DoWhile objReader.Peek() <> -1
    TextLine = objReader.ReadLine() '& vbNewLine
    valid = check.IsMatch(TextLine)
If valid Then
    objWriter.WriteLine(TextLine)
'Used to REdim NAUO_Array
    NumberOfLines = NumberOfLines + 1
EndIf
Loop
Else
    MsgBox("File Does Not Exist")
EndIf
    objWriter.Close()
    objReader.Close()
If NumberOfLines <> 0 Then 'If file exists
ReDim NAUO_Array(NumberOfLines, j_Col)
ReDim Sec_Array(NumberOfLines, j_Col)
EndIf
EndFunction

PublicSub MakeTree()
Dim FILE_NAME AsString = "C:\test\shape.txt" 'STEP File

```

```

Dim objWriter AsNew StreamWriter("c:\test\BTree.txt") 'Before Tree Construction
Dim objWriter1 AsNew StreamWriter("c:\test\ATree.txt") 'After Tree Construction
Dim objReader AsNew StreamReader(FILE_NAME)
Dim TextLine AsString
Dim RPD0, RPD1, RPD2, RPDLine AsString
Dim i AsInteger = 0
Dim Everything_OK AsBoolean = False

'Find Name of part
Dim pattern1 AsString = "[a-zA-Z0-9._%+-.:~\s]*"
Dim pattern2 AsString = "[0-9]*,"
Dim pattern5 AsString = "[0-9]*"
'find RPD2
Dim pattern3 AsString = "[0-9]*,[0-9]*"
Dim pattern4 AsString = ",[0-9]*"

Dim check AsNew System.Text.RegularExpressions.Regex(pattern1)
Dim valid AsBoolean = False

If File.Exists(FILE_NAME) = TrueThen
DoWhile objReader.Peek() <> -1
    TextLine = objReader.ReadLine() '& vbNewLine
    i = i + 1
    RPD0 = System.Text.RegularExpressions.Regex.Match(TextLine, pattern1).Value
    RPD1 = System.Text.RegularExpressions.Regex.Match(TextLine, pattern2).Value
    RPD1 = System.Text.RegularExpressions.Regex.Match(RPD1, pattern5).Value
    RPD2 = System.Text.RegularExpressions.Regex.Match(TextLine, pattern3).Value
    RPD2 = System.Text.RegularExpressions.Regex.Match(RPD2, pattern4).Value
    RPD2 = System.Text.RegularExpressions.Regex.Match(RPD2, pattern5).Value
    NAUO_Array(i, 1) = RPD0 'RPD0 - ID
    NAUO_Array(i, 2) = RPD1 'RPD1 - NUMBER
    NAUO_Array(i, 3) = RPD2 'RPD2 - NUMBER
'NAUO_Array(i,0) - FATHER NODE
    NAUO_Array(i, 4) = i
    NAUO_Array(i, 5) = 0 '0 means not in tree yet, 1 means that the node is
in the tree
    NAUO_Array(i, 6) = 0 'Keep the quantity for each part
    RPDLine = Str(i) + "," + RPD0 + "," + RPD1 + "," + RPD2 + "."
    objWriter.WriteLine(RPDLine)
Loop
Else
    MsgBox("File Does Not Exist")
EndIf

```

```

If File.Exists(FILE_NAME) Then
    Find_Father()
    TextLine = "A/A NodeFather ID FLAG1 FLAG2 InTheTRee Quantity"
    objWriter1.WriteLine(TextLine)
For i = 1 To NumberOfLines
    TextLine = String.Format("{0,5}", Str(i)) + " " + String.Format("{0,5}",
NAUO_Array(i, 0)) + " " + String.Format("{0,-5}", NAUO_Array(i, 1)) + " " +
String.Format("{0,-5}", NAUO_Array(i, 2)) + " " + String.Format("{0,-5}",
NAUO_Array(i, 3)) + " " + String.Format("{0,-5}", NAUO_Array(i, 4)) + " " +
String.Format("{0,-5}", NAUO_Array(i, 5)) + " " + String.Format("{0,-5}",
NAUO_Array(i, 6))
    objWriter1.WriteLine(TextLine)
Next
    TextLine = "COPIES-----"
    objWriter1.WriteLine(TextLine)
    TextLine = "A/A NodeFather ID FLAG1 FLAG2 InTheTRee Quantity"
    objWriter1.WriteLine(TextLine)
For i = 1 To Final_i_dim - NumberOfLines
    TextLine = Str(i) + " " + Sec_Array(i, 0) + " " + Sec_Array(i, 1) + "
" + Sec_Array(i, 2) + " " + Sec_Array(i, 3) + " " + Sec_Array(i, 4) + " "
+ Sec_Array(i, 5) + " " + Sec_Array(i, 6)
    objWriter1.WriteLine(TextLine)
Next
EndIf
    objWriter.Close()
    objWriter1.Close()
    objReader.Close()
    Create_Tree(Step_Form.TreeView2, NAUO_Array, NumberOfLines)
EndSub
PublicSub Find_Father()
Dim T(NumberOfLines, 6) AsString
Dim i, j, k, temp AsInteger
Dim SearchFor AsString
Dim ST_counter AsInteger = 0
    T = NAUO_Array
'Computes quantities for Level 0,1
For i = 1 To NumberOfLines
    T(i, 0) = 0 'By default, all parts belongs to Root
If T(i, 0) = 0 Or T(T(i, 0), 0) = 0 Then
    T(i, 6) = 1
EndIf
Next
    temp = 0
For i = 1 To NumberOfLines
For j = 1 To NumberOfLines
    SearchFor = T(i, 2)
If SearchFor = T(j, 3) Then
If T(i, 6) = 1 Then
    T(i, 0) = j 'Part i or i-node has father the j-node or h subassembly

```



```

        T(i, 6) = T(i, 6) + 1 ' Quantity for level 2
Else
    temp = temp + 1
    Sec_Array(temp, 0) = j
    Sec_Array(temp, 1) = T(i, 1)
    Sec_Array(temp, 2) = T(i, 2)
    Sec_Array(temp, 3) = T(i, 3)
    Sec_Array(temp, 4) = T(i, 4)
    Sec_Array(temp, 5) = T(i, 5)
    T(i, 6) = T(i, 6) + 1
EndIf
EndIf
Next
Next
    Final_i_dim = NumberOfLines + temp

For i = 1 To NumberOfLines
If T(i, 6) > 1 Then
    T(i, 6) = T(i, 6) - 1
EndIf
Next
ReDim F_Array(NumberOfLines + temp, j_Col)
For i = 1 To NumberOfLines
For j = 0 To 6
    F_Array(i, j) = T(i, j)
Next
Next
For i = NumberOfLines + 1 To NumberOfLines + temp
For j = 0 To 6
    F_Array(i, j) = Sec_Array(i - NumberOfLines, j)
Next
Next
    NAUO_Array = T
EndSub

PublicSub Create_Tree(ByVal TV As TreeView, ByVal NAUO_Array1(,) AsString,
ByVal i_dim AsInteger)
Dim T(NumberOfLines, j_Col) AsString
ReDim F_Array(i_dim, j_Col)
Dim i, j, k, l AsInteger
Dim L1, L2, L3, L4, L5 AsInteger

For i = 1 To i_dim
If NAUO_Array1(i, 0) = 0 And NAUO_Array1(i, 5) = 0 Then
Dim TNode AsNew TreeNode
    NAUO_Array1(i, 5) = 1
    TNode.Name = NAUO_Array1(i, 4)
    TNode.Text = NAUO_Array1(i, 1) + "--F" + NAUO_Array1(i, 0) + "--Cp" +
NAUO_Array1(i, 6)

```

```

        TNode.Tag = NAUO_Array1(i, 0)
        TV.Nodes(0).Nodes.Add(TNode)
    EndIf
Next
For i = 1 To i_dim
    If (NAUO_Array1(i, 0) <> 0) And NAUO_Array1(i, 5) = 0 Then
    If NAUO_Array1(NAUO_Array1(i, 0), 0) = 0 Then
    Dim TNode AsNew TreeNode
        NAUO_Array1(i, 5) = 1
    'NAUO_Array1(i, 6) = 1
        TNode.Name = NAUO_Array1(i, 4)
        TNode.Text = NAUO_Array1(i, 1) + "--F" + NAUO_Array1(i, 0) + "--Cp" +
NAUO_Array1(i, 6)
        TNode.Tag = NAUO_Array1(i, 0)
        TV.Nodes(0).Nodes(NAUO_Array1(i, 0)).Nodes.Add(TNode)
    'MsgBox(TNode.FullPath)

    EndIf
    EndIf
Next

For i = 1 To i_dim
    If (NAUO_Array1(i, 0) <> 0) And NAUO_Array1(i, 5) = 0 Then
    If NAUO_Array1(NAUO_Array1(NAUO_Array(i, 0), 0), 0) = 0 Then
    Dim TNode AsNew TreeNode
        TNode.Name = NAUO_Array1(i, 4)
        TNode.Text = NAUO_Array1(i, 1) + "--F" + NAUO_Array1(i, 0) + "--Cp" +
NAUO_Array1(i, 6)
        TNode.Tag = NAUO_Array1(i, 0)
        TV.Nodes(0).Nodes(NAUO_Array1(NAUO_Array1(i, 0),
0)).Nodes((NAUO_Array1(i, 0))).Nodes.Add(TNode)
        NAUO_Array1(i, 5) = 1

    EndIf
    EndIf
Next

For i = 1 To i_dim
    If (NAUO_Array1(i, 0) <> 0) And NAUO_Array1(i, 5) = 0 Then
        L1 = NAUO_Array1(i, 0)
        L2 = NAUO_Array1(L1, 0)
        L3 = NAUO_Array1(L2, 0)
        L4 = NAUO_Array1(L3, 0)
    ' MsgBox(L4)
    If NAUO_Array1(NAUO_Array1(NAUO_Array1(NAUO_Array(i, 0), 0), 0), 0) = 0 Then
    Dim TNode AsNew TreeNode
        TNode.Name = NAUO_Array1(i, 4)
        TNode.Text = NAUO_Array1(i, 1) + "--F" + NAUO_Array1(i, 0) + "--Cp" +
NAUO_Array1(i, 6)
        TNode.Tag = NAUO_Array1(i, 0)

```

```

        TV.Nodes(0).Nodes(NAUO_Array1(NAUO_Array1(NAUO_Array1(i, 0),
0), 0)).Nodes((NAUO_Array1(NAUO_Array1(i, 0), 0))).Nodes(NAUO_Array1(i,
0)).Nodes.Add(TNode)
        NAUO_Array1(i, 5) = 1
EndIf
EndIf
Next

    k = Final_i_dim - i_dim
For i = 1 To k
If NAUO_Array1(Sec_Array(i, 0), 0) <> 0 Then
Dim TNode AsNew TreeNode
    TNode.Name = Sec_Array(i, 4)
    TNode.Text = Sec_Array(i, 1) + "AAAAAA" + Sec_Array(i, 0) + "--Cp" +
Sec_Array(i, 6)
    TNode.Tag = Sec_Array(i, 0)
    TV.Nodes(0).Nodes(NAUO_Array1(Sec_Array(i, 0),
0)).Nodes(Sec_Array(i, 0)).Nodes.Add(TNode)
    TV.SelectedNode = TV.Nodes(0).Nodes(NAUO_Array1(Sec_Array(i, 0),
0)).Nodes(Sec_Array(i, 0))

Else
EndIf
Next

For i = 1 To k
If NAUO_Array1(NAUO_Array1(Sec_Array(i, 0), 0), 0) <> 0 Then
Dim TNode AsNew TreeNode
    TNode.Name = Sec_Array(i, 4)
    TNode.Text = Sec_Array(i, 1) + "BBBBB" + Sec_Array(i, 0) + "--Cp" +
Sec_Array(i, 6)
    TNode.Tag = Sec_Array(i, 0)
    TV.Nodes(0).Nodes(NAUO_Array1(NAUO_Array1(Sec_Array(i, 0), 0),
0)).Nodes(NAUO_Array1(Sec_Array(i, 0), 0)).Nodes(Sec_Array(i,
0)).Nodes.Add(TNode)
Else
EndIf
Next
EndSub
EndModule

```

Παράρτημα IV – Τεκμηρίωση του λογισμικού OAM2DSM

Στην ενότητα αυτή περιγράφετε η εφαρμογή που σχεδιάστηκε και υλοποιήθηκε στα πλαίσια της παρούσας εργασίας με σκοπό να τεκμηριώσει με αποτελέσματα τη μεθοδολογία που ακολουθήθηκε για την ενοποίηση των μοντέλων OAM/NIST και DSM. Ουσιαστικά πρόκειται για τέσσερις αυτόνομες λειτουργικά προγραμματιστικές μονάδες – ενότητες που ενσωματώθηκαν στη συνέχεια σε κοινό γραφικό περιβάλλον διασύνδεσης. Συγκεκριμένα οι δομικές μονάδες είναι:

- Υλοποίηση του λεξικογραφικού αναλυτή για τα αρχεία STEP.
- Σχεσιακή αναπαράσταση του OAM/NIST-CPM.
- Μοντελοποίηση του OAM2DSM μοντέλου.
- Αλγόριθμος ομαδοποίησης των εξαρτημάτων και υποσυναρμολογήσεων.

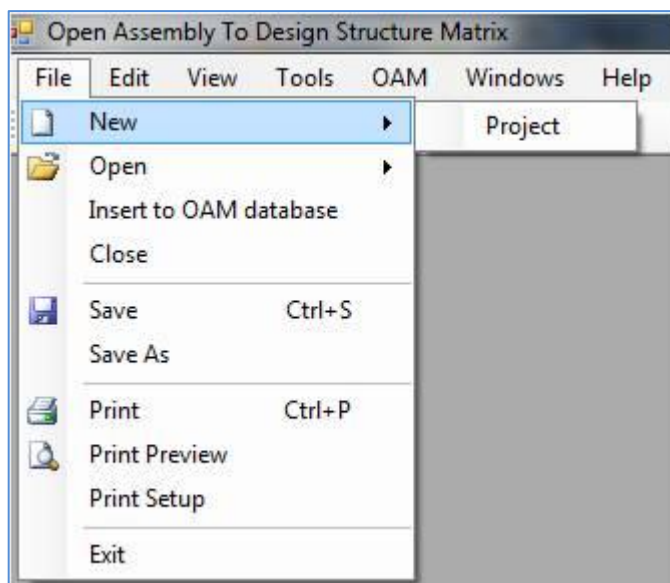
Ακολουθεί ο πίνακας με το λογισμικό που χρησιμοποιήθηκε.

Πίνακας Λογισμικού που χρησιμοποιήθηκε για τις ανάγκες της διατριβής		
A/A	Λογισμικό	Εταιρία
1	Microsoft® Visual Studio® .NET™ 2010	Microsoft®
2	Inventor® 3D CAD	AUTODESK
3	yEd Graph Editor 3.12.2	yWorks
4	STP viewer 2.3	STPviewer.com

Ο κώδικας, τα παραδείγματα και οι προσομοιώσεις σχεδιάστηκαν και πραγματοποιήθηκαν σε έναν υπολογιστή Dell Dual Core 3,8 MHz, 4Gb RAM, 160 Gb HDD, Microsoft Windows 7 professional edition 64Bit.

Γραφικό περιβάλλον διασύνδεσης(GUI)

Η εφαρμογή ενοποιεί αυτόνομες λειτουργικές μονάδες κώδικα με σκοπό να ενοποιήσει τα μοντέλα OAM-NIST και DSM. Κάνοντας εκκίνηση έχουμε το βασικό παράθυρο της εφαρμογής από το οποίο θα εισάγουμε το αρχείο STEP που έχει παραχθεί – εξαχθεί από ένα λογισμικό μοντελοποίησης 3DCAD (π.χ Inventor®, Pro- Enginer κλπ). Κάθε αρχείο STEP θεωρείται ένα νέο έργο και παίρνει έναν μοναδικό αριθμό ως αναγνωριστικό που ακολουθεί το έργο κατά την αποθήκευση του όπως επίσης και κατά την επεξεργασία των δεδομένων του.



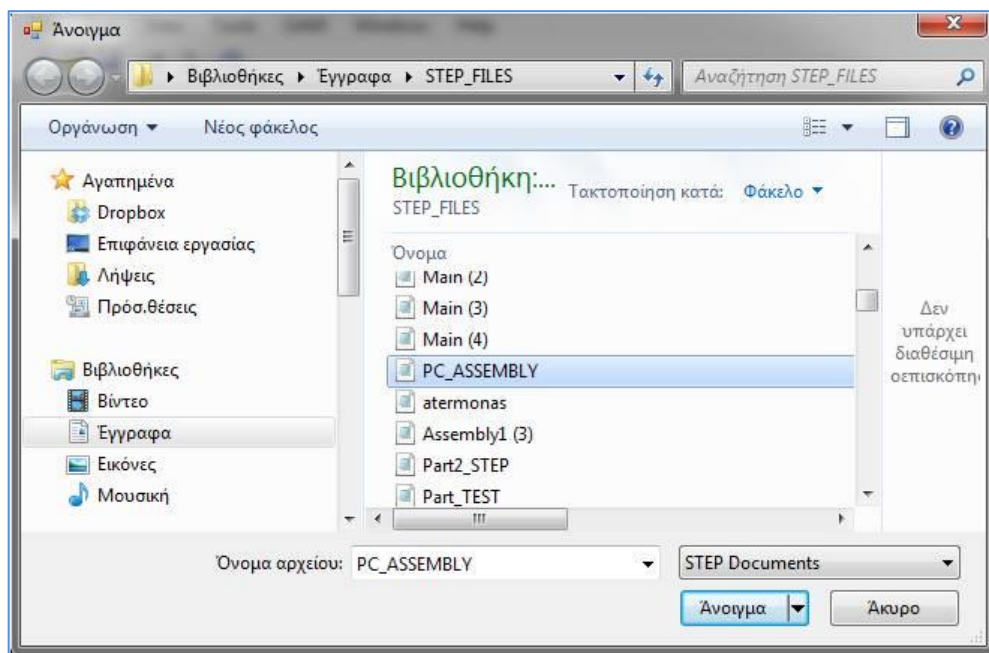
Εικόνα 64: Το μενού Αρχείο από το οποίο ο χρήστης μπορεί είτε να δημιουργήσει νέο έργο είτε να ανοίξει ένα από τα αποθηκευμένα.

Στην παραπάνω εικόνα φαίνονται οι επιλογές του βασικού μενού και στο επόμενο πίνακα γίνεται περιγραφή των επιλογών.

Μενού File	
New	Δρομολογεί τη δημιουργία νέου έργου
Open	Ανοίγει ένα από τα αποθηκευμένα έργα για επεξεργασία, τροποποίηση κλπ
Insert to OAM database	Η επιλογή αυτή ενεργοποιείται κάθε φορά που ένα νέο έργο πρόκειται να εισαχθεί. Επιλέγοντας τη, το αρχείο ενημερώνεται η βάση δεδομένων που μοντελοποιεί το OAM με το νέο τεχνούργημα που περιγράφεται στο αρχείο STEP.
Close	Το ενεργό έργο κλείνει και ο χρήστης μπορεί να ανοίξει άλλο ή να δημιουργήσει νέο.
Save	Αποθήκευση του συνολικού έργου καθώς και της διαδρομής στο δίσκο που θα παραχθούν τα αρχεία καταγραφής, τα βοηθητικά αρχεία κλπ.
SaveAS	Λειτουργεί όπως και η επιλογή Save με τη διαφορά ότι ο χρήστης μπορεί να επιλέξει άλλο όνομα για το έργο ή άλλο φάκελο αποθήκευσης των βοηθητικών αρχείων.
Print	Εκτύπωση για το τρέχων έργο των εγγραφών στη βάση δεδομένων του OAM.
PrintPreview	Προεπισκόπηση εκτύπωσης
PrintSetup	Επιλογή Εκτυπωτή για εκτύπωση
Exit	Έξοδος από το πρόγραμμα.

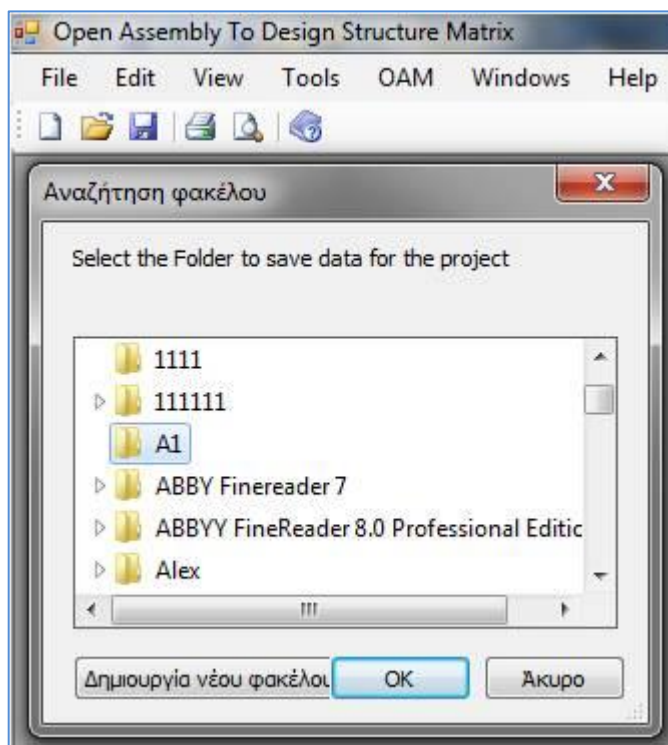
Πίνακας 15: Επιλογές για το χρήστη από το μενού File.

Ενεργοποιώντας την επιλογή File>New ζητείται η επιλογή του αρχείου STEP (επομένη εικόνα) για την εισαγωγή του στο πρόγραμμα.



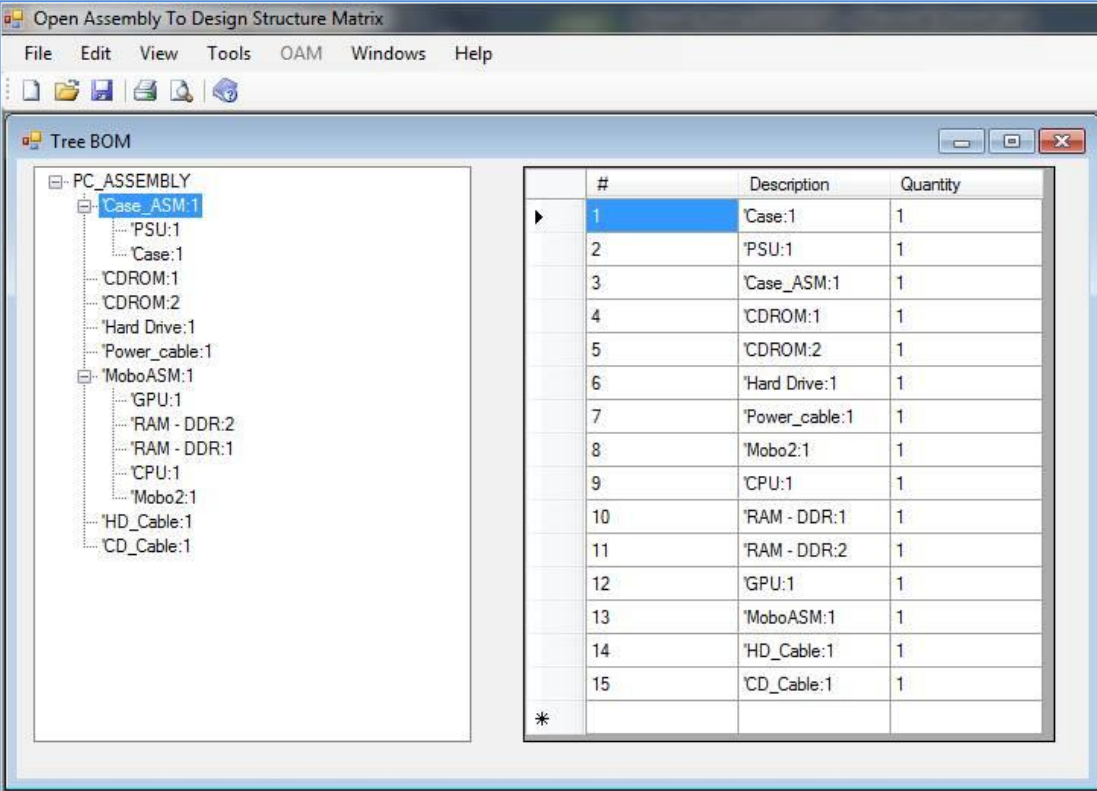
Εικόνα 65:Επιλογή του αρχείου STEP για τη δημιουργία νέου έργου.

Στη συνέχεια ο χρήστης πρέπει να ορίσει το φάκελο που το AOM2DSM θα τοποθετήσει τα προσωρινά αρχεία, τα αρχεία καταγραφής κ.λ.π. (εικόνα 66).



Εικόνα 66: Επιλογή φακέλου για τα αρχεία του νέου έργου.

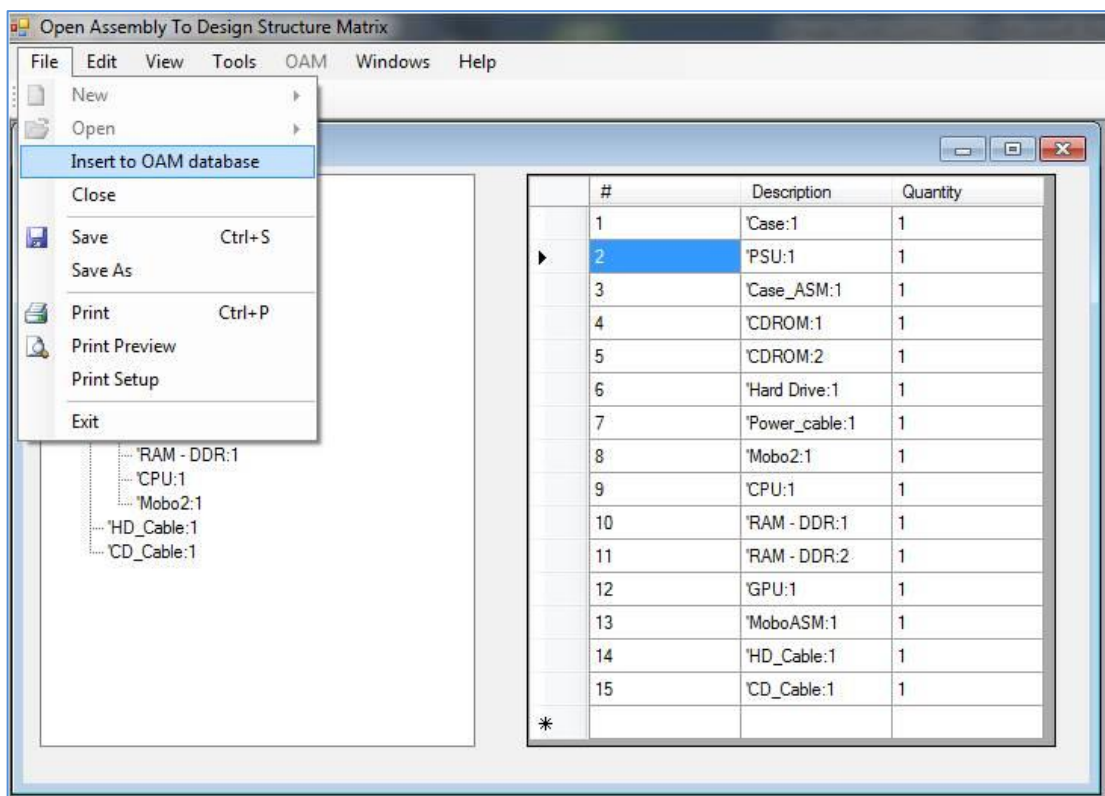
Έχοντας επιλέξει το αρχείο STEP και το φάκελο, το πρόγραμμα προχωρά στον αλγόριθμο λεξικογραφικής ανάλυσης του αρχείου STEP και εμφανίζει τον πίνακα εξαρτημάτων της συναρμολόγησης, την ποσότητα κάθε εξαρτήματος και την ιεραρχία των υποσυναρμολογήσεων όπως έχουν αποτυπωθεί στο αρχείο. Αν ο χρήστης συμφωνεί με τον τρόπο που τα αποτυπώνει το πρόγραμμα θα πρέπει να προχωρήσει στην επόμενη επιλογή η οποία είναι η εισαγωγή των δεδομένων στη βάση δεδομένων που υλοποιεί το μοντέλο OAM. Επιλέγοντας *File>Insert to OAM database* ξεκινά η διαδικασία εισαγωγής από την οποία θα πάρουμε ως ανάδραση τρία πληροφοριακά παράθυρα ότι η διαδικασία τμηματικά ολοκληρώνεται.



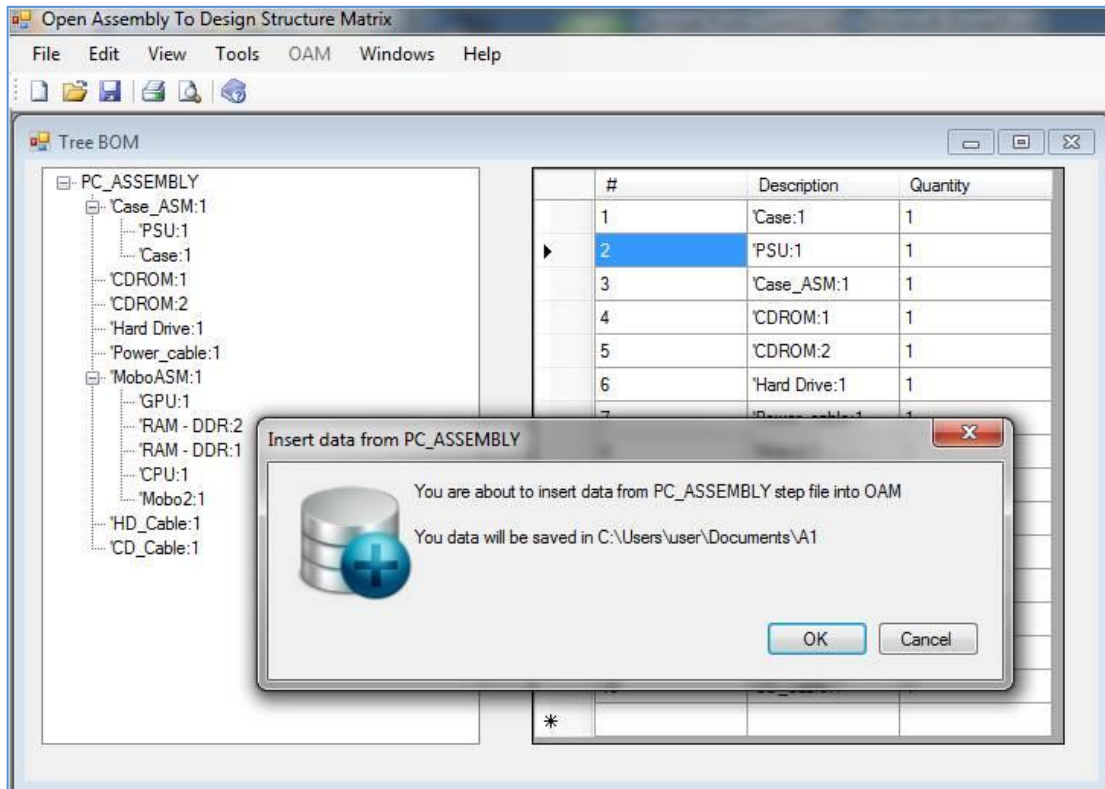
The screenshot shows a software window titled "Open Assembly To Design Structure Matrix". It has a menu bar with "File", "Edit", "View", "Tools", "OAM", "Windows", and "Help". Below the menu is a toolbar with icons for file operations. The main area is divided into two panes. The left pane, titled "Tree BOM", shows a hierarchical tree structure of a PC assembly. The right pane contains a table with 15 rows of component data.

#	Description	Quantity
1	'Case:1	1
2	'PSU:1	1
3	'Case_ASM:1	1
4	'CDROM:1	1
5	'CDROM:2	1
6	'Hard Drive:1	1
7	'Power_cable:1	1
8	'Mobo2:1	1
9	'CPU:1	1
10	'RAM - DDR:1	1
11	'RAM - DDR:2	1
12	'GPU:1	1
13	'MoboASM:1	1
14	'HD_Cable:1	1
15	'CD_Cable:1	1

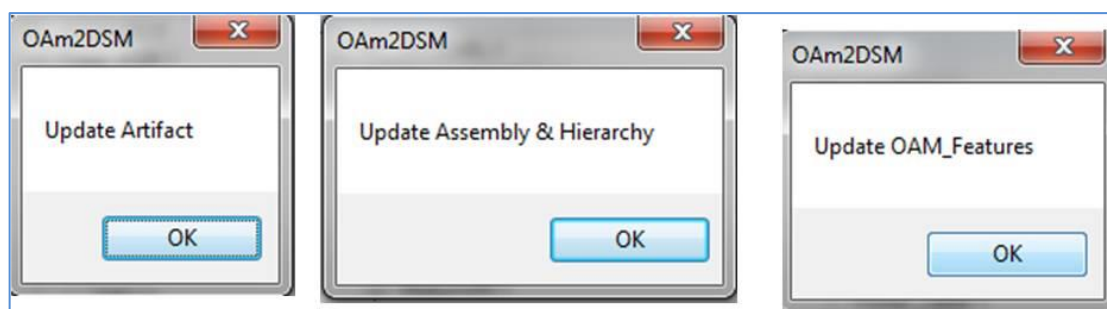
Εικόνα 67: Πίνακας εξαρτημάτων του αρχείου STEP, περιγραφή και ποσότητα.



Εικόνα 68: Εισαγωγή των δεδομένων όπως αποτυπώνονται στο παράθυρο στη βάση δεδομένων του OAM



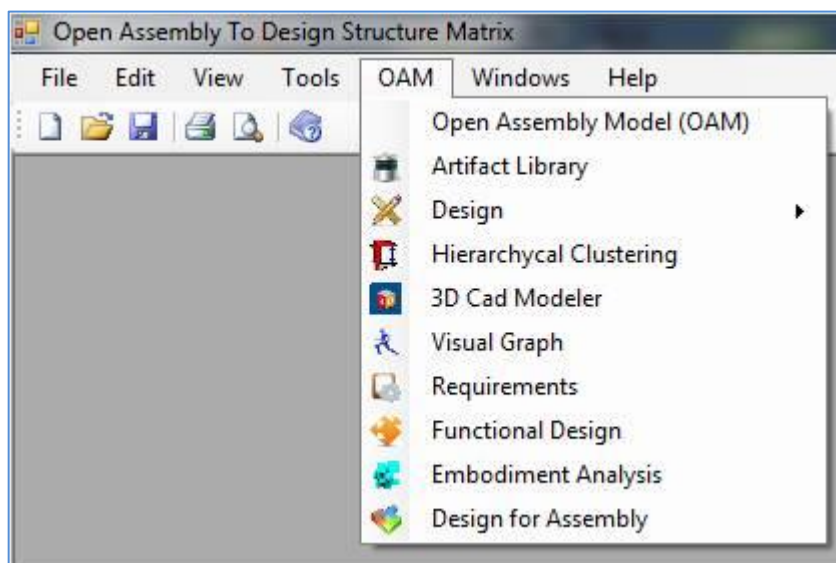
Εικόνα 69: Ξεκινά η διαδικασία εισαγωγής δεδομένων.



Εικόνα 70: Ολοκλήρωση της διαδικασίας εισαγωγής νέου έργου με τα πληροφοριακά παράθυρα.

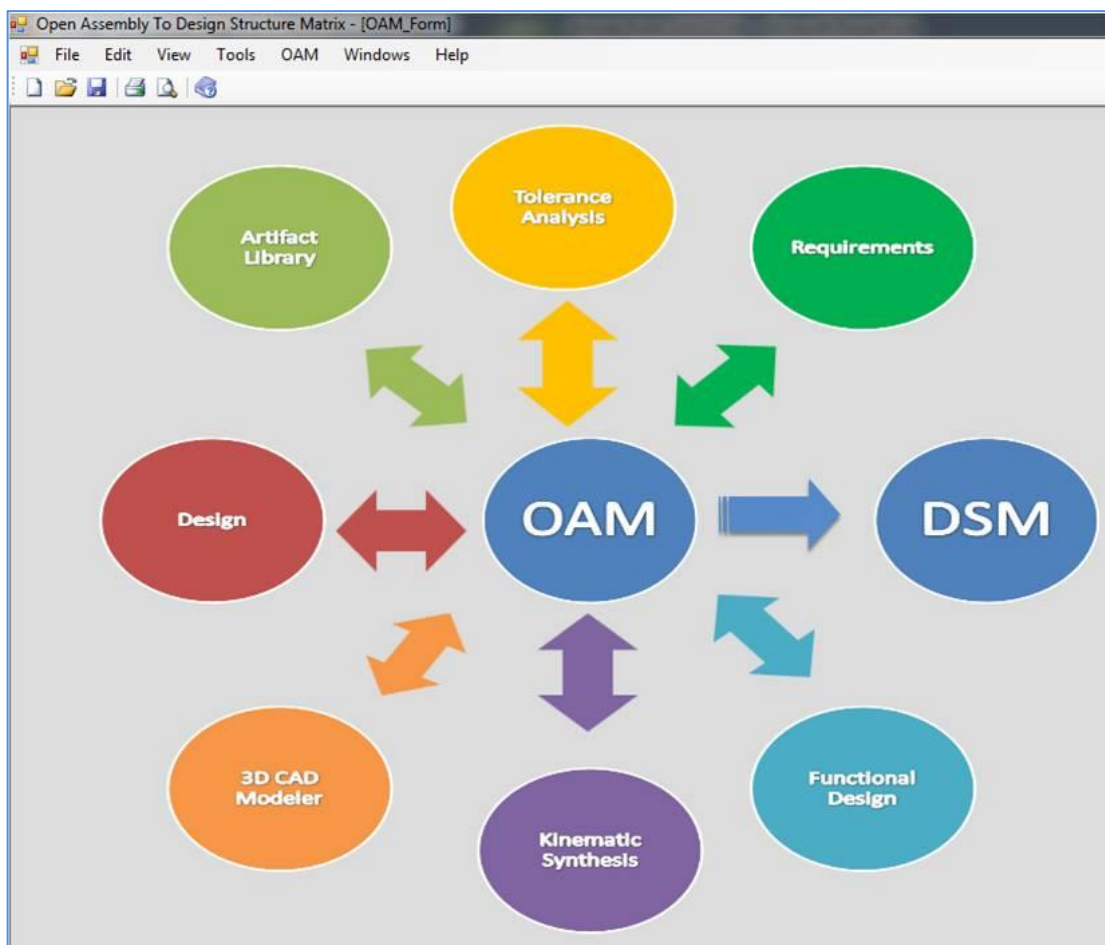
Αφού έχουμε εισάγει τα δεδομένα επιλέγουμε από το βασικό μενού την επιλογή OAM. Στον παρακάτω πίνακα περιγράφονται όλες οι επιλογές του μενού OAM.

Μενού OAM	
Open Assembly Model	Ανοίγει το παράθυρο για την εμφάνιση των στοιχείων που μοντελοποιεί το OAM
Artifact Library	Εμφανίζει όλα τα έργα – τεχνουργήματα που είναι αποθηκευμένα στη βάση δεδομένων
3D CAD Modeler	Συνδέει το αρχείο STEP με ένα πρόγραμμα 3DCAD που για την ανάγκες της παρούσας εργασίας επιλέχθηκε το Inventor®.
VisualGraph	Αποτυπώνει με γραφικό τρόπο την ιεραρχία των εξαρτημάτων του τεχνουργήματος. Για την ανάγκες της εργασίας το OAM2DSM συνδέεται με την εφαρμογή yWorks® για την αναπαράσταση των εξαρτημάτων υπό μορφή γράφου, αποτυπώνοντας τις εξαρτήσεις.
Requirements	Για το τρέχων τεχνούργημα ο χρήστης – σχεδιαστής μπορεί να περιγράψει επιπλέον απαιτήσεις για το τελικό προϊόν που πρέπει να πληροί.



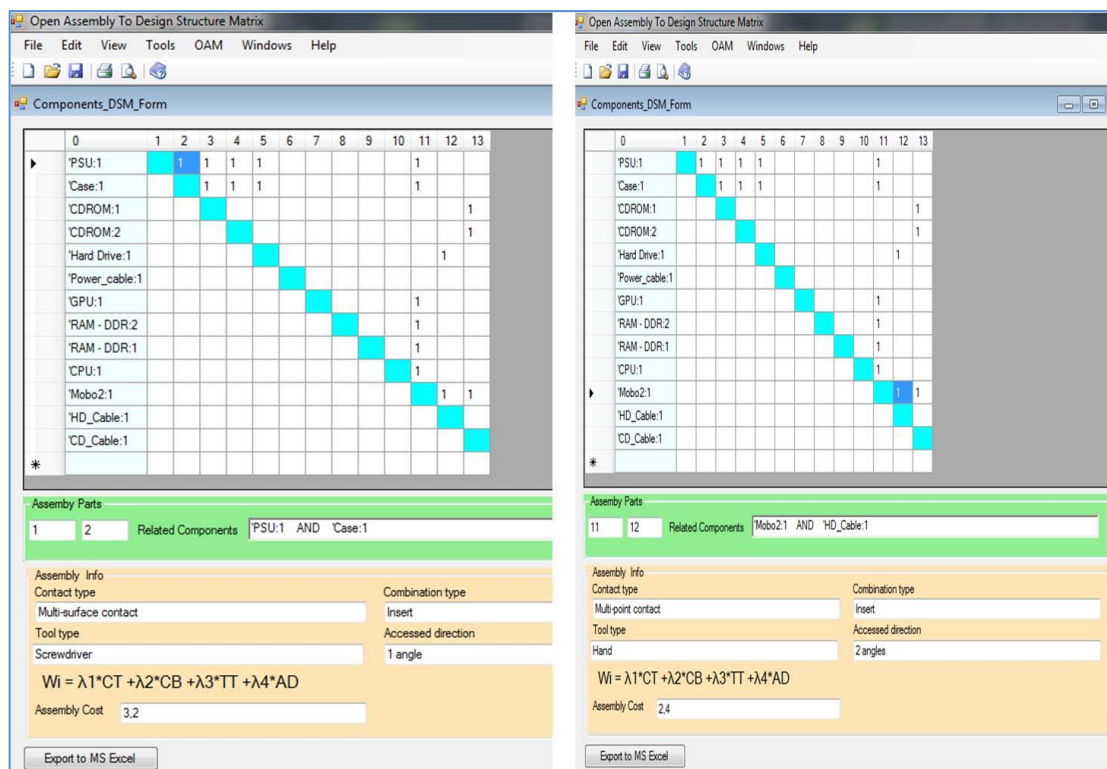
Εικόνα 71: Η επιλογή OAM και το μενού εσωτερικά της.

Στη συνέχεια, επιλέγοντας το OAM >OpenAssemblyModel (OAM) εμφανίζεται το παρακάτω παράθυρο στο οποίο έχουμε όλες τις δυνατές επιλογές για το μοντέλο OAM.



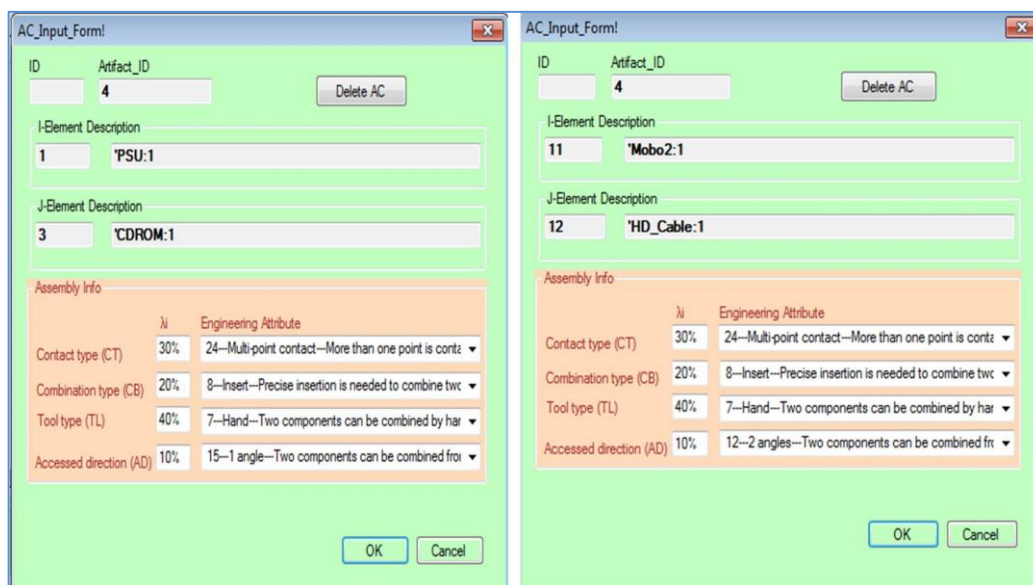
Εικόνα 72: Το μοντέλο OAM και οι σχέσεις του με τα υπόλοιπα μοντέλα στην ανάπτυξη του προϊόντος. Η επιλογή DSM οδηγεί στην ολοκλήρωση των δύο μοντέλων και είναι από τα τελικά βήματα στην επιλογή του χρήστη (Επιλογή OAM >OpenAssemblyModel)

Επιλέγοντας το OAM στο κέντρο οδηγούμαστε στην επόμενη εικόνα η οποία για το τρέχων έργο αποτυπώνει τα εξαρτήματα του τεχνουργήματος, όπως εξάχθηκαν από το STEP, όλες τις διαθέσιμες πληροφορίες για τη συναρμολόγηση καθώς και το κόστος της συναρμολόγησης βασιζόμενοι στο μοντέλο που περιγράφηκε στην ενότητα 3.3.5. Στο παράθυρο φαίνονται δύο στιγμιότυπα από τις εξαρτήσεις όπου περιγράφεται το είδος της εξάρτησης με βάση τους τέσσερεις δείκτες και το συνολικό κόστος συναρμολόγησης.



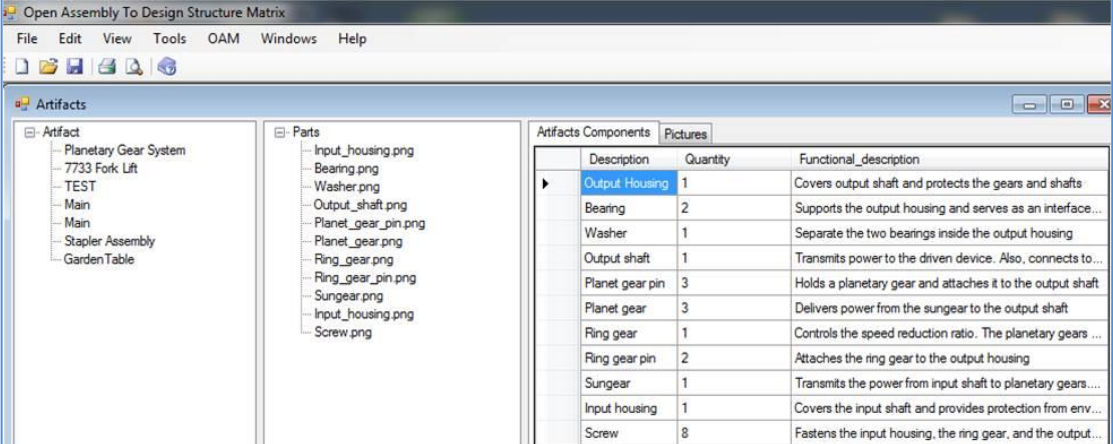
Εικόνα 73: Αποτύπωση δύο στιγμιότυπων της συναρμολόγησης εξαρτημάτων καθώς και το κόστος τους.

Οποιαδήποτε αλλαγή στον τύπο συναρμολόγησης είναι εφικτή όπως φαίνεται και στο παρακάτω παράθυρο στο οποίο φαίνονται δύο στιγμιότυπα αλλαγής των παραμέτρων στη συναρμολόγηση.



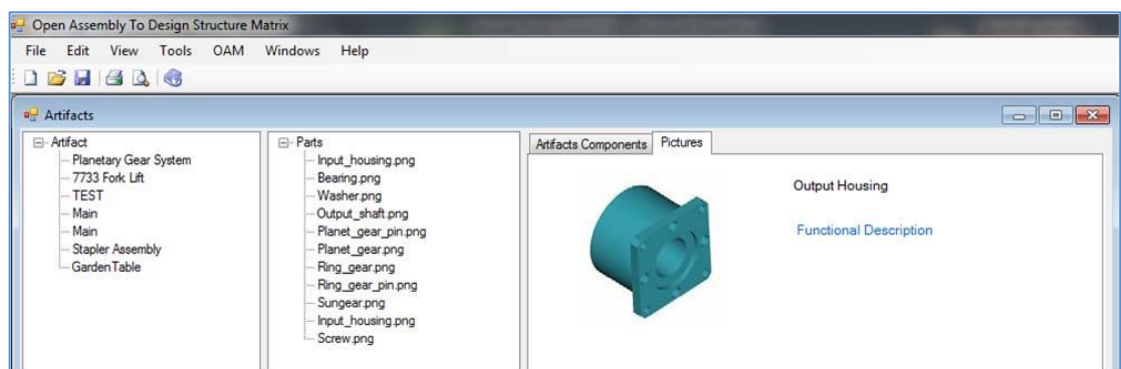
Εικόνα 74: Τροποποίηση στις παραμέτρους της συναρμολόγησης.

Η επιλογή *ArtifactLibrary* οδηγεί στα επόμενα δύο παράθυρα στα οποία ο χρήστης έχει τη δυνατότητα να εμφανίσει, να διαγράψει και να τροποποιήσει τα τεχνουργήματα που έχει αποθηκεύσει στη βάση δεδομένων του OAM2DSM.



Description	Quantity	Functional_description
Output Housing	1	Covers output shaft and protects the gears and shafts
Bearing	2	Supports the output housing and serves as an interface...
Washer	1	Separate the two bearings inside the output housing
Output shaft	1	Transmits power to the driven device. Also, connects to...
Planet gear pin	3	Holds a planetary gear and attaches it to the output shaft
Planet gear	3	Delivers power from the sun gear to the output shaft
Ring gear	1	Controls the speed reduction ratio. The planetary gears ...
Ring gear pin	2	Attaches the ring gear to the output housing
Sungear	1	Transmits the power from input shaft to planetary gears....
Input housing	1	Covers the input shaft and provides protection from env...
Screw	8	Fastens the input housing, the ring gear, and the output...

Εικόνα 75: Εμφάνιση των τεχνουργημάτων που διαθέτει η βάση δεδομένων



Εικόνα 76: Εμφάνιση των εικόνων των εξαρτημάτων κάθε τεχνουργήματος.