ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΣΥΣΤΗΜΑ ΕΞΑΓΩΓΗΣ ΓΝΩΣΗΣ ΑΠΟ ΚΑΤΑΝΕΜΗΜΕΝΕΣ ΚΑΙ ΕΤΕΡΟΓΕΝΕΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ: ΕΦΑΡΜΟΓΗ ΣΕ ΙΑΤΡΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΚΩΝΣΤΑΝΤΙΝΟΣ Α. ΧΡΙΣΤΟΦΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ηράκλειο, Ιούνιος 2000

Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

ΣΥΣΤΗΜΑ ΕΞΑΓΩΓΗΣ ΓΝΩΣΗΣ ΑΠΟ ΚΑΤΑΝΕΜΗΜΕΝΕΣ ΚΑΙ ΕΤΕΡΟΓΕΝΕΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ: ΕΦΑΡΜΟΓΗ ΣΕ ΙΑΤΡΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Μεταπτυχιακή Εργασία που υποβλήθηκε από τον
Κωνσταντίνο Α. Χριστοφή
ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση του
ΔΙΠΛΩΜΑΤΟΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας: _____

Κωνσταντίνος Α. Χριστοφής
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Εισηγητική Επιτροπή: _____

Στέλιος Ορφανουδάκης, Καθηγητής, Επόπτης

_____

Απόστολος Τραγανίτης, Αναπληρωτής Καθηγητής, Μέλος

_____

Δημήτριος Πλεξουσάκης, Επίκουρος Καθηγητής, Μέλος

_____

Γεώργιος Ποταμιάς, Ερευνητής, Ινστιτούτο Πληροφορικής Ι.Τ.Ε, Έκτακτο Μέλος

Δεκτή: _____

Πάνος Κωνσταντόπουλος, Καθηγητής,
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο Κρήτης, Ιούνιος 2000

"…αφιερωμένο στην οικογένεια μου…"

Περιεχόμενα

ΕΥΡΕΤΗΡΙΟ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

ARM	Association Rule Mining
CCTR	Common Clinical Term Representation file
COAS	Clinical Observation Access Service
CORBA	Common Object Request Broker Architecture
DM	Data Mining
DTD	Document Type Definition
EHCR	Integrated Electronic Health Care Record
ICD	International Coding for Diseases
IDL	Interface Definition Language
KDD	Knowledge Discovery from Data
LQS	Lexicon Query Service
ML	Machine Learning
OMA	Object Management Architecture
OMG	Object Management Group
PCDD	Patient Clinical Data Directory
PIDS	Person Identification Service
UMLS	Unified Medical Language System
XML	Extensible Markup Language

ΣΥΣΤΗΜΑ ΕΞΑΓΩΓΗΣ ΓΝΩΣΗΣ ΑΠΟ ΚΑΤΑΝΕΜΗΜΕΝΕΣ ΚΑΙ ΕΤΕΡΟΓΕΝΕΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ: ΕΦΑΡΜΟΓΗ ΣΕ ΙΑΤΡΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Κωνσταντίνος Α. Χριστοφής

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Με την τρέχουσα εξάπλωση των δεδομένων, το πρόβλημα εύρεσης τρόπων και μεθοδολογιών σύνδεσης κατανεμημένων και ετερογενών πηγών πληροφόρησης, γίνεται ολοένα και περισσότερο κρίσιμο. Πέρα όμως από την συλλογή των τεράστιων αυτών μεγεθών δεδομένων, είναι πολύ σημαντικό να μελετηθεί και να εξεταστεί η γενική ανάγκη σημασιολογικής ομογενοποίησης τους, καθώς και η ανακάλυψη γνώσης που πηγάζει και απορρέει από τις αντίστοιχες πηγές πληροφόρησης. Η ανάγκη αυτή αποτελεί πρόκληση για το χώρο της Μηχανικής Μάθησης και της Εξόρυξης Δεδομένων, και κατά συνέπεια όλων των ML-DM/KDD ερευνητών.

Μολονότι η κατανεμημένη φύση των δεδομένων έχει λίγο έως πολύ ερμηνευτεί και κατανοηθεί, η ετερογένεια είναι μια περισσότερο πολύπλοκη έννοια. Το πραγματικό πρόβλημα που προκύπτει στο σημείο αυτό, είναι όχι μόνο πώς θα προσπελαστούν συγκεκριμένες πηγές πληροφόρησης που διατηρούν τμήματα της συνολικής πληροφορίας, αλλά πώς θα βρεθούν τρόποι καθορισμού και ευρετηριασμού της αναγκαίας μόνο πληροφορίας, σε αντιστοιχία με το σύνολο της εκεί αποθηκευμένης πληροφορίας. Μια πολλά υποσχόμενη προσέγγιση, σε αυτό το πρόβλημα *ολοκλήρωσης*, είναι η απόκτηση κεντρικού ελέγχου έναντι των επιμέρους οργανισμών και πληροφοριακών πηγών σε ένα επίπεδο '*μετά-δεδομένων*', διατηρώντας ταυτόχρονα την αυτονομία των ατομικών συστημάτων στο επίπεδο των 'ατομικών εγγραφών δεδομένων'.

Η παρούσα μεταπτυχιακή εργασία παρουσιάζει το πρόβλημα της ανακάλυψης και εξαγωγής γνώσης από κατανεμημένες και ετερογενείς ιατρικές πηγές πληροφοριών/ δεδομένων. Η βασική πρόκληση είναι 'πως οι λειτουργίες οι οποίες προέρχονται από τον χώρο του data mining και της Μηχανικής Μάθησης, υιοθετούνται και γίνονται λειτουργικές σε ένα τέτοιο κατανεμημένο και ετερογενές περιβάλλον'. Για το λόγο αυτό, προτείνεται και υλοποιείται μια πολυσύνθετη διαδικασία ολοκλήρωσης, η οποία αντιμετωπίζει θέματα όπως: (1) αποτελεσματική πρόσβαση σε κατανεμημένες και δομημένες πηγές πληροφόρησης, (2) αξιόπιστη ομογενοποίηση και ολοκλήρωση των ετερογενών δεδομένων(λαμβάνοντας υπόψη την οντολογία του πεδίου εργασίας και κάποιες ιδιαίτερα σημαντικές οντολογικές λειτουργίες), (3) επεξεργασία(statistical analysis, data mining, κ.τ.λ) των δεδομένων, και (4) παρουσίαση των αποτελεσμάτων. Αυτή η προσέγγιση

ολοκλήρωσης υπαγορεύεται από τον συνδυασμό και την παρουσία πολλών τεχνολογιών και λειτουργιών. Ενδεικτικά αναφέρουμε τη χρήση CORBA τεχνολογίας για ομοιόμορφη πρόσβαση σε κατανεμημένα δεδομένα, λειτουργίες σημασιολογικής ομογενοποίησης και προηγμένες DTD/ XML διαδικασίες. Οι λειτουργίες αυτές συσχετισμένες με σύγχρονες, προχωρημένες και αποτελεσματικές αναπαραστάσεις μοντέλων, διαμορφώνουν και προσδιορίζουν ένα σκελετό και ένα περιβάλλον, στο οποίο μπορούν να εκπονηθούν πλέον έξυπνα και αποτελεσματικά όλες οι απαιτούμενες και αναγκαίες KDD διεργασίες.

Η βασική συνεισφορά της εργασίας μας είναι η συνεργασία και η τροποποίηση όλων των KDD/ARM λειτουργιών, ώστε να είναι απόλυτα εφαρμόσιμες στα παραγόμενα XML έγγραφα. Συγκεκριμένα, επικεντρωνόμαστε και αντιμετωπίζουμε το πρόβλημα παραγωγής ενδιαφερόντων συσχετίσεων μεταξύ των δεδομένων που είναι αποθηκευμένα σε κατανεμημένα και ετερογενή ιατρικά πληροφοριακά συστήματα. Το περιβάλλον εφαρμογής της προσέγγισης μας είναι το HYGEIAnet: The Integrated Health Care Network of Crete. Η παρούσα εργασία επεκτείνει την αρχιτεκτονική αναφοράς του HYGEIAnet προσθέτοντας: (α) λειτουργίες σημασιολογικής ομογενοποίησης, (β) τη διαδικασία δημιουργίας των DTD/XML εγγράφων, υποδεικνύοντας τον τρόπο αναπαράστασης της αποθηκευμένης πληροφορίας, (γ) αντικειμενοστραφή σχήματα δόμησης των δεδομένων και επιτελούμενες λειτουργίες σε αυτά και (δ) την υιοθέτηση των KDD λειτουργιών- υλοποιημένες από έναν εξειδικευμένο(Associations Rule Mining) αλγόριθμο- ονομαζόμενο AprioriXML. Βασισμένοι στην πρόβλεψη ότι οι μελλοντικές βάσεις θα χρησιμοποιούν XML-like αναπαραστάσεις και μορφές δεδομένων, προκειμένου να αποθηκεύεται και να εκμαιεύεται η προς επεξεργασία πληροφορία, η εργασία μας παρουσιάζει μια υποσχόμενη αρχιτεκτονική και ένα περιβάλλον εργασίας κινούμενο προς αυτήν ακριβώς την κατεύθυνση.

Επόπτες: Στέλιος Ορφανουδάκης
Καθηγητής
Τμήμα Επιστήμης Υπολογιστών,
και
Γεώργιος Ποταμιάς
Ερευνητής
Ινστιτούτο Πληροφορικής
Ίδρυμα Τεχνολογίας και Έρευνας
Ηράκλειο - Κρήτης

KNOWLEDGE DISCOVERY FROM DISTRIBUTED AND HETEROGENEOUS DATABASES: A CLINICAL INFORMATION SYSTEMS APPLICATION

Constantinos A. Christofis

Master of Science Thesis

Computer Science Department
University of Crete

Abstract

With the current explosion of data, the problem of how to combine *distributed* and *heterogeneous* information sources becomes more and more critical. Besides collecting enormous amount of data it is very important to consider the general need of *semantic integration* and *knowledge discovery* from these sources, an important and necessary challenge for machine learning- ML, and data mining/knowledge discovery- DM/KDD researchers.

If the distributed nature of data has a more-or-less clear definition (even hard, and most of the times tedious to achieve), *heterogeneity* is a more complex concept. The real issue here is not only how to access specific information systems that maintain the data, but also how to identify and *index* the essential information in them. A promising approach to this integration problem is to gain control of the organization's information resources at a *meta-data* level, while allowing autonomy of individual systems at the data instance level.

This thesis presents the problem of discovering and acquiring knowledge from *distributed* and *heterogeneous* data sources. The main challenge is 'how data mining and machine-learning operations are *adapted* and made operational in such a distributed and heterogeneous environment'. To this end, a *multi-phase* data integration procedure is proposed and implemented, which: (1) efficiently *accesses* structured and distributed data sources; (2) reliably *homogenize* and *integrate* the stored heterogeneous data (with a dedicated domain *ontology* and respective ontological operations playing a crucial role); (3) effectively data *processing* operations such as, traditional statistical analysis, *data mining*, etc; and (4) *presentation* of results (e.g., *visualization* operations). The integration approach is realized by the coupling of multi-disciplinary technologies ranging from, CORBA based seamless access to distributed data, to semantic data homogenization operations- based on the appropriate utilization of a domain specific data models and ontology, and to advanced DTD/XML operations. These operations- coupled with advanced and effective data representation models, forms a framework in which efficient and effective ML/KDD operations are performed.

The fundamental contribution of our work is the incorporation and customization of KDD/ARM (Association Rules Mining) operations on top of appropriately generated DTD/XML documents. In particular, we tackle the problem of inducing interesting *associations* between data items stored in remote clinical information systems. The test-bed environment of our approach and implementation is the *HYGEIAnet:* The *Integrated Health Care Network of Crete*. The presented work

expands the HYGEIAnet reference architecture by adding: (a) the information and data semantic indexing operations, (b) the generation of DTD/XML documents to represent and store data- coupled with ontological operations to semantically homogenize the data, (c) the object-oriented data structuring schemas and operations, and (d) the adaptation of KDD operations- realized by a specially devised Associations Rule Mining algorithm- named AproriXML. Based on the argument that, "*future databases will use XML-like structures in order to store and retrieve data"* then, the thesis presents a promising architecture and framework for hosting advanced and intelligent data processing operations in the emerging distributed and heterogeneous data and information environment.

Supervisors: Stelios Orphanoudakis
Professor
Computer Science Department
University of Crete,
and
George Potamias
Researcher
Institute of Computer Science
Foundation for Research and Technology
Heraclion - Crete

ΕΥΧΑΡΙΣΤΙΕΣ

Κατ' αρχήν θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. **Στέλιο Ορφανουδάκη** που μου έδωσε τη δυνατότητα να εκπονήσω αυτή την Μεταπτυχιακή Εργασία. Με τις γνώσεις και την εμπειρία του οδηγούσε πάντα την εργασία στις σωστές κατευθύνσεις.

Θα ήθελα από την καρδιά μου να απευθύνω κάτι πολύ περισσότερο από ένα μεγάλο ευχαριστώ στον κ. **Γεώργιο Ποταμιά**. Τον ευχαριστώ για τις αμέτρητες ώρες που διέθεσε μέσω των συναντήσεων μας, για τις συμβουλές του, τις γνώσεις του, αλλά πολύ περισσότερο για την ανθρώπινη συμπαράσταση και κατανόησή του σε όλη την διάρκεια της παρούσας εργασίας και συνεργασίας μας. Η συμβολή του ήταν αποφασιστική και καταλυτική για την εκπόνηση αυτής της μεταπτυχιακής εργασίας.

Επίσης, ένα μεγάλο ευχαριστώ στον συντονιστή του Κέντρου Ιατρικής Πληροφορικής και Τηλεματικών Εφαρμογών στην Υγεία, κ. Μανώλη **Τσικνάκη**, που μου έδωσε κύριες κατευθυντήριες γραμμές για την εργασία, κατά τη διάρκεια πολλών συζητήσεων που είχα μαζί του όλο αυτό το διάστημα. Ακόμη, ευχαριστώ όλα τα μέλη του CMI-HTA για τη βοήθειά τους, και ιδιαίτερα τους **Σ. Κωστομανωλάκη** και **Π. Λέλη**, για τις πολύτιμες παρατηρήσεις τους.

Κεφάλαιο 1:
Εισαγωγή

Με την τρέχουσα εξάπλωση και την έκρηξη διάδοσης των δεδομένων, το πρόβλημα εύρεσης τρόπων και μεθοδολογιών πρόσβασης και σύνδεσης *κατανεμημένων* και *ετερογενών* πηγών πληροφόρησης, γίνεται ολοένα και περισσότερο κρίσιμο. Πέρα όμως από την συλλογή των τεράστιων αυτών μεγεθών δεδομένων, είναι πολύ σημαντικό να μελετηθεί και να εξεταστεί η γενική ανάγκη *σημασιολογικής ομογενοποίησης* τους, καθώς και η ***ανακάλυψη γνώσης*** που πηγάζει και απορρέει από τις αντίστοιχες πηγές πληροφορίας/ δεδομένων. Η ανάγκη αυτή αποτελεί πρόκληση για το χώρο της *Μηχανικής Μάθησης* (*Machine Learning*- ML) και της *Εξόρυξης Δεδομένων* (Data Mining- DM) ή *Ανακάλυψης Γνώσεων από Δεδομένα* (*Knowledge Discovery from Data*- KDD), και κατά συνέπεια όλων των ML-DM/KDD ερευνητών.

Η διαφορά στο σημείο αυτό και κατ' επέκταση η μεγάλη πρόκληση σε σχέση με την αντιμετώπιση απλών, στατικών και ομοιογενών πηγών πληροφορίας, είναι (μεταξύ άλλων): (α) η έκταση και διαβάθμιση του προβλήματος είναι μεγαλύτερη από κάθε προηγούμενη απόπειρα και προσπάθεια στο χώρο του ML και DM/KDD, και (β) η ανάγκη για ολοκλήρωση πολλαπλών μορφών αναπαράστασης της γνώσης (π.χ οντολογίες πεδίων) είναι από τα πλέον σημαντικά και ουσιώδη ζητήματα.

Η παρούσα μεταπτυχιακή εργασία παρουσιάζει το πρόβλημα της *ανακάλυψης και εξαγωγής γνώσης από κατανεμημένες και ετερογενείς ιατρικές πηγές πληροφοριών/ δεδομένων*. Επικεντρωνόμαστε στο πρόβλημα παραγωγής *ενδιαφερόντων συσχετίσεων*, μεταξύ των δεδομένων (data items) που είναι αποθηκευμένα σε κατανεμημένα και ετερογενή ιατρικά πληροφοριακά συστήματα. Παρά το γεγονός ότι επικεντρώνουμε την προσοχή μας και επεξεργαζόμαστε δεδομένα του ιατρικού πεδίου, η προτεινόμενη μεθοδολογία και οι παρουσιαζόμενες λύσεις, μπορούν εύκολα να επεκταθούν και να καλύψουν την γενικότερη περίπτωση διαχείρισης (με την έννοια του data mining) κατανεμημένων και ετερογενών πηγών πληροφορίας.

Μολονότι η κατανεμημένη φύση των δεδομένων έχει λίγο έως πολύ ερμηνευτεί και κατανοηθεί, η ετερογένεια είναι μια περισσότερο πολύπλοκη έννοια, η οποία χρίζει μεγαλύτερης αποσαφήνισης. Ας θεωρήσουμε για παράδειγμα την κατάσταση όπου η ίδια εφαρμογή βάσεων δεδομένων έχει εγκατασταθεί και τρέχει σε διαφορετικές και απομακρυσμένες περιοχές. (Η κατάσταση αυτή δεν περιλαμβάνει την περίπτωση client-server database εφαρμογών, όπου οι πελάτες (clients) είναι κατανεμημένοι, αλλά η βάση δεδομένων του server κρατείται σε ένα μέρος και έτσι η ομοιογένεια είναι προκαθορισμένη και διασφαλισμένη). Σε αυτήν την περίπτωση, και παρά το γεγονός ότι οι χρήστες χρησιμοποιούν την ίδια εφαρμογή δεδομένων, μπορεί να εισαγάγουν και να εγγράφουν δεδομένα, χωρίς να χρησιμοποιούν μια προκαθορισμένη, κοινή και ομοιογενή μορφή.

Τα προαναφερθέντα προβλήματα καθιστούν μια συνηθισμένη κατάσταση η οποία επικρατεί στον ιατρικό χώρο και κυρίως εμφανίζεται σε περιβάλλοντα ανάπτυξης Ολοκληρωμένων Ηλεκτρονικών Φακέλων Υγείας - *Integrated Electronic Health Care Record* problem (I-EHCR environment) [6, 7, 8]. Ένας γιατρός ο οποίος προσπελαύνει την εγγεγραμμένη πληροφορία (Healthcare Record) για έναν ασθενή, χρειάζεται μια συνοπτική παρουσίαση των τμημάτων της (EHCR

segments), καθώς τις περισσότερες φορές μόνο ένα μικρό μέρος από την συνολική πληροφορία θα ανακληθεί και θα παρουσιαστεί σε λεπτομέρεια. Το γεγονός αυτό σημαίνει ότι, όταν επιτυγχάνεται πρόσβαση σε ένα συγκεκριμένο πληροφοριακό σύστημα, υπάρχει ανάγκη και εν-τέλει απαιτείται να εξαχθεί ένα μικρό μόνο μέρος από την εκεί αποθηκευμένη πληροφορία. Το πραγματικό πρόβλημα που προκύπτει στο σημείο αυτό είναι όχι μόνο πώς θα προσπελαστούν συγκεκριμένες πηγές πληροφόρησης που διατηρούν τμήματα της συνολικής πληροφορίας, αλλά πώς θα βρεθούν τρόποι καθορισμού και ευρετηριασμού (indexing) της αναγκαίας μόνο πληροφορίας σε αντιστοιχία με το σύνολο της εκεί αποθηκευμένης πληροφορίας.

Μια πολλά υποσχόμενη προσέγγιση, σε αυτό το πρόβλημα *ολοκλήρωσης*, είναι η απόκτηση κεντρικού ελέγχου έναντι των επιμέρους οργανισμών και πληροφοριακών πηγών σε ένα επίπεδο '*μετά-δεδομένων*' (meta-data), διατηρώντας ταυτόχρονα την αυτονομία των ατομικών συστημάτων στο επίπεδο των 'ατομικών εγγραφών δεδομένων' (data instance level). Ο αντικειμενικός στόχος του 'meta-database' μοντέλου είναι η επίτευξη ολοκλήρωσης της πληροφορίας/ δεδομένων των κατανεμημένων ετερογενών συστημάτων, ενώ παράλληλα επιτρέπεται σε αυτά τα συστήματα να λειτουργούν ανεξάρτητα και ταυτόχρονα [9]. Εντούτοις, η επίτευξη ολοκλήρωσης σε *σημασιολογικό* (semantic) επίπεδο αποτελεί ένα μείζον πρόβλημα, καθότι η λογική, η γνώση, και οι μορφές δεδομένων που χρησιμοποιούνται στα διάφορα συστήματα είναι σύνθετες και συχνά ασύμβατες [10]. Επιπρόσθετα, όσο περισσότερο επιθυμεί και προσπαθεί να κρύψει κάποιος την ετερογένεια, τόσο περισσότερο ασχολείται και εμπλέκεται με θέματα ολοκλήρωσης. Έτσι μια ρεαλιστική λύση είναι η απόκρυψη της ετερογένειας στο κορυφαίο (top) επίπεδο, κάνοντας ταυτόχρονα τις *επιμέρους πηγές της πληροφορίας να εμφανίζονται στους τελικούς χρήστες σαν μια τεράστια συλλογή από αντικείμενα που συμπεριφέρονται ομοιόμορφα* [11].

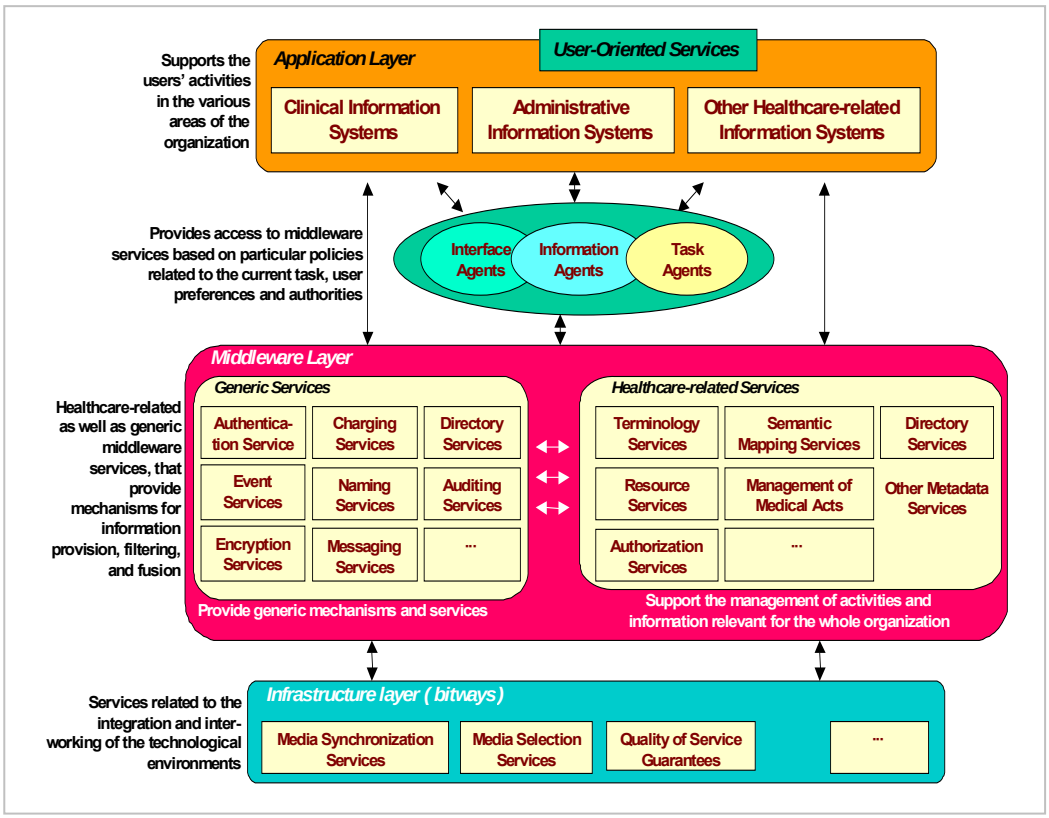
Προκειμένου να αντιμετωπιστούν αυτά τα προβλήματα, μια πολυσύνθετη διαδικασία ολοκλήρωσης (multi-phase data integration) θα πρέπει να ακολουθηθεί, προτού εφαρμοστούν οι διαδικασίες data mining και machine-learning επάνω στα δεδομένα. Μια λογική προσέγγιση της διαδικασίας αυτής θα πρέπει να αντιμετωπίζει έξυπνα και αποδοτικά θέματα όπως:

1. Αποτελεσματική ***πρόσβαση*** σε κατανεμημένες και δομημένες πηγές πληροφόρησης.
2. Αξιόπιστη ***ομογενοποίηση*** και ***ολοκλήρωση*** των ετερογενών δεδομένων – λαμβάνοντας υπόψη την *οντολογία* του πεδίου εργασίας και κάποιες οντολογικές λειτουργίες (με ιδιαίτερη σημασία στο σημείο αυτό).
3. ***Επεξεργασία*** (analyze, mine, κ.τ.λ) των δεδομένων και
4. ***Παρουσίαση*** των αποτελεσμάτων (visualization).

Στην εργασία αυτή παρουσιάζουμε την μεθοδολογία, και όλες τις διεργασίες που πρέπει να ακολουθηθούν, προκειμένου να πραγματοποιηθούν αυτές οι λειτουργίες. Το περιβάλλον εφαρμογής της προσέγγισης μας σε θέματα mining για κατανεμημένες πηγές πληροφόρησης, είναι το HYGEIAnet: The Integrated Health Care Network of Crete [12, 32]. Μια από τις βασικές υπηρεσίες υγείας που προσφέρεται μέσω του δικτύου του HYGEIAnet (όπως φαίνεται και από τη σχετική αρχιτεκτονική αναφοράς στο σχήμα 1), είναι η πρόσβαση σε κλινική πληροφορία του ασθενή, η οποία είναι αποθηκευμένη σε αυτόνομα κλινικά πληροφοριακά συστήματα.

Στο κεφάλαιο 2, παρουσιάζουμε την βασική τεχνολογία για πρόσβαση σε κατανεμημένη πληροφορία και δομημένες πηγές πληροφόρησης, βασισμένη σε

τεχνολογίες Common Object Request Broker Architecture (CORBA). Το κεφάλαιο 3, παρουσιάζει τις θεμελιώδεις διαδικασίες για την επίτευξη της σημασιολογικής ομογενοποίησης και ολοκλήρωσης κατανεμημένων πηγών πληροφοριών/ δεδομένων. Αναλύονται και περιγράφονται το μοντέλο πρόσβασης Clinical Observation Access Service (COAS), ο ρόλος και η χρήση του- αναφερόμαστε στο ιατρικό πεδίο οντολογίας και παρουσιάζουμε την μορφή αναπαράστασης της πληροφορίας που επιλέγουμε, χρησιμοποιώντας το πρότυπο αναπαράστασης πληροφοριών Extensible Markup Language (XML). Στο κεφάλαιο 4, παρουσιάζουμε τεχνικές για parsing XML εγγράφων, τα οποία είναι συμβατά με τη σχετική Document Type Definition(DTD) που έχουμε αναπτύξει. Στο κεφάλαιο αυτό εξάγουμε και απλά στατιστικά ιατρικά συμπεράσματα και αποτελέσματα (υγειονομικούς δείκτες), χρησιμοποιώντας τις παραγόμενες δομές. Στο κεφάλαιο 5, αναφερόμαστε γενικά σε KDD και DM, και παρουσιάζουμε την γενικότερη αρχιτεκτονική που υποστηρίζουμε. Το κεφάλαιο 6, κατατοπίζει πλήρως τον αναγνώστη σχετικά με την παραγωγή *Κανόνων Αλληλοσυσχέτισης* (Association Rule Mining- ARM), αποσαφηνίζοντας έννοιες και παρουσιάζοντας προηγούμενες εργασίες και αλγορίθμους. Στο κεφάλαιο 7, παρουσιάζουμε τον βασικό αλγόριθμο που υλοποιούμε για ανεύρεση κανόνων αλληλοσυσχέτισης στο ιατρικό πεδίο, αποσαφηνίζοντας και παραλληλίζοντας ιατρικές έννοιες με έννοιες του χώρου του ARM. Στο κεφάλαιο 8, παρουσιάζονται παραδείγματα εκτέλεσης του αλγορίθμου χρησιμοποιώντας πραγματικά ιατρικά δεδομένα. Συνέχεια του κεφαλαίου αυτού αποτελεί το κεφάλαιο 9, όπου γίνεται επίδειξη του συστήματος μας, παρουσιάζοντας τον τρόπο εμφάνισης/ παρουσίασης των αποτελεσμάτων επεξεργασίας (visualization) και λειτουργίας του στον τελικό χρήστη. Τέλος, στο κεφάλαιο 10 παρουσιάζονται συμπεράσματα από τη δουλειά μας, με ταυτόχρονη παράθεση των μελλοντικών κατευθύνσεων έρευνας και ανάπτυξης.



Σχήμα 1. *Αρχιτεκτονική Αναφοράς του HYGEIAnet*

Κεφάλαιο 2:
Πρόσβαση σε Κατανεμημένες (Ιατρικές) Πηγές Πληροφόρησης

Στο κεφάλαιο αυτό περιγράφουμε τον τρόπο με τον οποίο η σύγχρονη τεχνολογία αντιμετωπίζει θέματα *πρόσβασης* σε κατανεμημένα πληροφοριακά συστήματα και αναφερόμαστε τόσο στα υιοθετημένα πρότυπα, όσο και στο αρχιτεκτονικό υπόβαθρο που έχει επικρατήσει για κατανεμημένες εφαρμογές. Αναφερόμαστε συνοπτικά στο *Patient Clinical Data Directory*- PCDD [31] και σχετίζουμε την όλη προσπάθεια ανάπτυξης με το δίκτυο τηλεματικών υπηρεσιών το οποίο αναπτύσσεται στην Περιφέρεια Κρήτης.

2.1. Κατανεμημένες Αρχιτεκτονικές

Ο κυρίαρχος, αντικειμενικός και απώτερος στόχος μας είναι η *ομοιογενής ολοκλήρωση* και *συγχώνευση* κατανεμημένων και ετερογενών πηγών πληροφορίας στο διαδίκτυο, καθώς και ο ομοιόμορφος τρόπος μεταφοράς και παρουσίασης της παραγόμενης πληροφορίας στους τελικούς χρήστες.

Προφανώς, η συλλογή της πληροφορίας δεν είναι αυτοσκοπός. Αυτό που είναι επιθυμητό είναι η αξιοποίηση της, άρα η δυνατότητα για εξαγωγή χρήσιμων και κατανοητών συμπερασμάτων. Ένας σημαντικός στόχος της πληροφορικής είναι να παρέχει στους τελικούς χρήστες ολοκληρωμένες υπηρεσίες υψηλού επιπέδου. Για να επιτευχθεί όμως κάτι τέτοιο πρέπει να είναι δυνατή η συγκέντρωση όλης της σχετικής πληροφορίας σε *ενιαία μορφή*, ώστε να είναι δυνατή η *αυτοματοποιημένη επεξεργασία* της και ανάλυση της. Προφανώς όμως, η ανάπτυξη τέτοιων υπηρεσιών δεν είναι καθόλου εύκολη υπόθεση!

Τα χαρακτηριστικά, τα οποία προαναφέρθηκαν επιβάλλουν πρώτα απ' όλα να αναπτυχθεί η κατάλληλη δικτυακή υποδομή, ώστε να είναι δυνατή η εύκολη επικοινωνία ανάμεσα στα διάφορα πληροφοριακά συστήματα. Έπειτα πρέπει να βρεθούν οι τρόποι ώστε να είναι δυνατή η *εννοιολογική* και *σημασιολογική* τους ομογενοποίηση. Αυτό αναλύεται περαιτέρω: στο να παρέχεται πρόσβαση στην πληροφορία, στο να είναι κατανοητή και αποδεκτή η δομή της, και τέλος να είναι δυνατή η αντιστοίχηση των όρων που χρησιμοποιούνται (κεφάλαια 3-4). Επιπλέον δεν πρέπει να παραβλέπονται οι περιορισμοί οι οποίοι τίθενται από την πλατφόρμα ανάπτυξης η οποία χρησιμοποιείται.

Αυτά τα κενά έρχονται να καλύψουν οι διάφοροι οργανισμοί τυποποίησης. Ο ρόλος τους είναι να ορίσουν κοινά αποδεκτά πρότυπα τα οποία θα ακολουθούνται από τους κατασκευαστές λογισμικού προκειμένου να επικοινωνούν τα συστήματα χωρίς να παρουσιάζονται ιδιαίτερα προβλήματα.

Κάποια από τα παραπάνω βήματα έχουν ήδη γίνει. Οι κατανεμημένες αρχιτεκτονικές *CORBA* και *DCOM* [26], του *Object Management Group* [13], και της *Microsoft* [25] αντίστοιχα, οι οποίες έχουν οριστεί ως *πρότυπα*, προσφέρουν την απαιτούμενη λειτουργικότητα, ώστε τα συστήματα να επικοινωνούν και να συνεργάζονται με σχετική ευκολία, ανεξάρτητα από την πλατφόρμα ανάπτυξης, το λειτουργικό, και το υλικό υπόβαθρο (hardware) που χρησιμοποιείται τοπικά. Επιπλέον, οι υπηρεσίες καταλόγων (LDAP/X.500) [14, 16] επιτρέπουν την ταχύτατη εύρεση αναφορών στα δεδομένα και γενικότερα τον προσδιορισμό της πληροφορίας που ενδιαφέρει την κάθε εφαρμογή. Η συμβολή αυτών των γενικών προτύπων στον σχεδιασμό και την ανάπτυξη οποιασδήποτε κατανεμημένης

εφαρμογής είναι πολύ σημαντική.

Όσον αφορά το βήμα της εννοιολογικής και σημασιολογικής ομογενοποίησης υπάρχει συνεχής δραστηριότητα, ώστε να προσδιοριστούν οι διάφοροι χώροι εφαρμογών, όπως για παράδειγμα η υγεία, η οικονομία, κλπ.

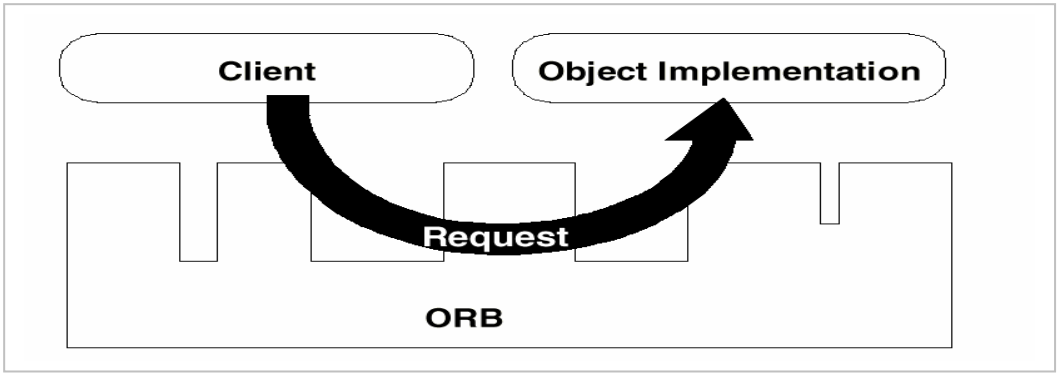
2.1.1. Object Management Group (OMG)

Το *Object Management Group*- OMG [OMG] είναι η μεγαλύτερη κοινοπραξία εταιριών λογισμικού στον κόσμο και αριθμεί περισσότερα από 800 μέλη, υποστηρικτές (developers) και τελικούς χρήστες.

Κύριος στόχος του OMG είναι να ορίσει ένα κοινά αποδεκτό αντικειμενοστραφές αρχιτεκτονικό υπόβαθρο για κατανεμημένες εφαρμογές, βασισμένες σε προδιαγραφές που υποστηρίζουν κατανεμημένα αντικείμενα. Επίσης, στους στόχους του συμπεριλαμβάνονται η ικανότητα για επαναχρησιμοποίηση, μεταφορά, και συνεργασία των οντοκεντρικών συνιστωσών λογισμικού (software components) σε ετερογενή περιβάλλοντα. Προς αυτήν την κατεύθυνση, το OMG υιοθετεί προδιαγραφές διεπαφών (interface) και πρωτοκόλλων, βασισμένες κυρίως σε εμπορικά διαθέσιμη τεχνολογία, και τα οποία όλα μαζί ορίζουν το *Object Management Architecture* (OMA).

2.1.2. Common Object Request Broker Architecture (CORBA)

Η *Common Object Request Broker Architecture*- CORBA [17] ορίζει την διεπιφάνεια προγραμματισμού για την OMA και τον *ORB*. Ο ORB (σχήμα 2) είναι ο βασικός μηχανισμός με τον οποίο τα αντικείμενα μπορούν να στέλνουν αιτήσεις (requests) και να παίρνουν απαντήσεις (responces) το ένα από το άλλο, είτε βρίσκονται στο ίδιο μηχάνημα, είτε επικοινωνούν πάνω από κάποιο δίκτυο. Οι πελάτες δεν χρειάζεται πλέον να απασχολούνται με τους μηχανισμούς επικοινωνίας και ενεργοποίησης (activation) των αντικειμένων, τον τρόπο με τον οποίο υλοποιήθηκαν αυτά, ή το που βρίσκονται. Άρα ο ORB χτίζει τα θεμέλια για την ανάπτυξη εφαρμογών που στηρίζονται σε κατανεμημένα αντικείμενα, και για την *διαλειτουργικότητα* (interoperability) εφαρμογών σε ομογενή αλλά και σε ετερογενή περιβάλλοντα.



Σχήμα 2. *Ο Object Request Broker (ORB)*

Η OMG *Interface Definition Language* (IDL) παρέχει ένα πρότυπο τρόπο για να ορίζουμε διεπαφές CORBA μεταξύ αντικειμένων. Ο IDL ορισμός είναι το "*συμβόλαιο*" ανάμεσα σ' αυτόν που υλοποιεί το αντικείμενο και στον πελάτη. Η IDL είναι μια αυστηρά δηλωτική (declarative) γλώσσα, η οποία είναι ανεξάρτητη από τις γλώσσες προγραμματισμού που μπορεί να χρησιμοποιηθούν. Οι

αντιστοιχίσεις της IDL στις διάφορες γλώσσες προγραμματισμού δίνουν τη δυνατότητα στα αντικείμενα να υλοποιούνται και να στέλνουν αιτήσεις στην γλώσσα που επιλέγει αυτός που αναπτύσσει την εφαρμογή, με ένα στυλ φυσικό για τη χρησιμοποιούμενη γλώσσα [18].

2.1.3. Συστατικά της αρχιτεκτονικής του OMG

Το *CORBAmed* [23, 24] ως το βασικό πεδίο εφαρμογών της OMG στον χώρο της υγείας, ορίζει κοινά αποδεκτά οντοκεντρικές διεπαφές, οι οποίες συμβάλουν στη διαλειτουργικότητα και συνεργασία ποικιλίας πλατφόρμων, λειτουργικών συστημάτων, γλωσσών, και εφαρμογών. Ο κύριος στόχος του CORBAmed είναι η βελτίωση της ποιότητας των υπηρεσιών, καθώς και η μείωση των εξόδων, με τη χρήση της τεχνολογίας CORBA. Αυτή τη στιγμή το CORBAmed έχει ξεκινήσει τις διαδικασίες για την παραγωγή κοινά αποδεκτών διεπαφών σε διάφορους τομείς της υγείας. Μέχρι στιγμής έχουν ολοκληρωθεί τρεις από αυτές, το *Clinical Observation Access Service*- COAS, το *Person Identification Service*- PIDS, και το *Lexicon Query Service*- LQS. Στις επόμενες ενότητες θα σχολιαστούν περιληπτικά οι δυο τελευταίες από τις παραπάνω τρεις προαναφερθείσες και κοινά αποδεκτές διεπαφές. Πριν από αυτό όμως, θα σχολιαστεί μια υπηρεσία της CORBA, το *Naming Authority*, στην οποία βασίζονται οι δύο από αυτές τις προδιαγραφές. Αυτές είναι το *LQS* [27] και το *PIDS* [20]. Στο *COAS* [19] θα σταθούμε αναλυτικά στο επόμενο κεφάλαιο, καθώς αποτελεί ουσιαστικό και βασικό συστατικό της δουλειάς μας.

Naming Authority. Το Naming Authority module έχει την δυνατότητα να δίνει απόλυτα μοναδικά ονόματα σε χώρους ονομάτων (name spaces) και συνεπώς στα ονόματα που περιέχονται σε αυτούς τους χώρους ονομάτων. Το απαραίτητο στοιχείο είναι η δυνατότητα της σύγκρισης δύο ονομάτων για ισοδυναμία. Αν είναι ισοδύναμα τότε αναπαριστούν την ίδια οντότητα, έννοια, ή πράγμα. Αυτό χρειάζεται, όταν ανεξάρτητες οντότητες παράγουν ονόματα, τα οποία υπάρχει πιθανότητα να συγκριθούν για ισοδυναμία. Παρόλα αυτά το αντίστροφο δεν ισχύει απαραίτητα. Έτσι μια οντότητα μπορεί να έχει διάφορα ονόματα.

Η *εξουσιοδότηση* για τον χώρο ονομάτων μπορεί να προέρχεται από διάφορους τύπους αποδεκτών ριζών ονομάτων. Η επιλογή κάποιας εξαρτάται από τις ανάγκες του χρήστη εφόσον κάθε ρίζα έχει διαφορετική ποιότητα διαχείρισης και μοναδικότητας. Αυτές που αναφέρονται στο [27] και το [20] είναι οι εξής:

- Η ISO ιεραρχία καταχώρησης.
- Τα Domain Name Services- DNS.
- Το OMG Interface Repository.
- Τα Universally Unique Ids- UUIDs του Distributed Computing Enviroment- DCE.
- Το OTHER το οποίο χρησιμοποιείται για απλοποίηση των ονομάτων σε κάποιες περιπτώσεις.

Person Identification Service- PIDS. Το Person Identification Service [20] είναι πλέον ένα κοινά αποδεκτό πρότυπο της OMG. Ψηφίστηκε από το CORBAmed τον Φεβρουάριο του 1998 και υιοθετήθηκε τελικά από την OMG board τον Αύγουστο του ίδιου έτους. Ο στόχος του PIDS είναι να οριστούν οι κατάλληλες διεπαφές, έτσι ώστε να είναι δυνατός ο *μονοσήμαντος προσδιορισμός* της ταυτότητας προσώπων (ασθενών στη περίπτωση μας).

Με βασικό γνώμονα την παραπάνω αρχή, το PIDS σχεδιάστηκε, με τέτοιο τρόπο, ώστε να :

- είναι ικανό να ψάχνει και να εντοπίζει ασθενείς, ανεξάρτητα από τον αλγόριθμο ταιριάσματος, είτε αυτόματα, είτε με τη βοήθεια κάποιου ειδικού.

- υποστηρίζει ομοσπονδίες από υπηρεσίες απόδοσης ταυτότητας σε ασθενείς.
- υποστηρίζει υλοποιήσεις του PIDS οι οποίες θα προστατεύουν το απόρρητο των ασθενών και θα βασίζονται σε μια μεγάλη ποικιλία από πολιτικές και μηχανισμούς ασφάλειας.
- επιτρέπει την εύκολη διασυνεργασία διαφορετικών υπηρεσιών PIDS.
- ορίσει τα διάφορα επίπεδα συμβατότητας, από τα απλά για επερωτήσεις μόνο ID Domains, μέχρι τις σύνθετες ομοσπονδίες από συσχετιζόμενα ID Domains.

Επιπλέον, πρέπει να τονιστεί ότι η λειτουργικότητα του PIDS περιορίζεται στην διαχείριση των IDs και των στοιχείων που προσδιορίζουν την ταυτότητα των προσώπων, και συνεπώς δεν προσφέρει αναφορές στις κατεξοχήν πληροφορίες που σχετίζονται με αυτά τα πρόσωπα ως οντότητες του χώρου εφαρμογής (χώρος/ πεδίο της υγείας για τη περίπτωση μας). Για να αποκτηθεί πρόσβαση σε τέτοιου είδους πληροφορίες όπως, κλινικές παρατηρήσεις, ασφάλιση ασθενών κλπ, πρέπει να χρησιμοποιηθούν άλλες υπηρεσίες όπως το COAS.

Lexicon Query Service- LQS. Το Lexicon Query Service [27] είναι πλέον ένα κοινά αποδεκτό πρότυπο της OMG. Ψηφίστηκε από το CORBAmed τον Απρίλιο του 1998 και υιοθετήθηκε τελικά από το OMG board τον Αύγουστο του ίδιου έτους. Ο στόχος αυτής της υπηρεσίας είναι ο καθορισμός ενός συνόλου κοινών, για ανάγνωση μόνο (read-only), μεθόδων πρόσβασης στα περιεχόμενα συστημάτων *ιατρικής ορολογίας*.

Ο όρος «συστήματα ιατρικής ορολογίας» καλύπτει όλο το φάσμα των συστημάτων, από τα απλά που αποτελούνται από λίστες ενός συνόλου από κώδικες και φράσεις, έως και συστήματα δυναμικά, με πολλαπλά σχήματα ιεραρχίας και κατηγοριοποίησης. Αυτού του είδους τα συστήματα χρησιμοποιούνται συνήθως για κάποιον από τους παρακάτω λόγους :

- *Πρόσληψη Πληροφορίας*. Χρησιμοποίηση υπηρεσιών ορολογίας για την διευκόλυνση της διαδικασίας εισαγωγής κωδικοποιημένων δεδομένων.
- *Παρουσίαση Πληροφορίας*. Χρησιμοποίηση υπηρεσιών ορολογίας για την μετάφραση κωδικοποιημένων δεδομένων σε μορφή που μπορεί να διαβαστεί από άνθρωπο ή υπολογιστή.
- *Διαμεσολάβηση*. Χρησιμοποίηση υπηρεσιών ορολογίας για την μετατροπή μηνυμάτων ή αρχείων δεδομένων από μία αναπαράσταση σε κάποια άλλη.
- *Ευρετηρίαση και Εξαγωγή Συμπερασμάτων*. Χρησιμοποίηση υπηρεσιών ορολογίας για επερωτήσεις που αφορούν σχέσεις που υπάρχουν ή όχι ανάμεσα στα δεδομένα, και για τον εντοπισμό δεδομένων που είναι σχετικά με κάποιο συγκεκριμένο θέμα ή κάποια οντότητα.
- Πλοήγηση. Χρησιμοποίηση υπηρεσιών ορολογίας για τον έλεγχο της δομής και της σημασιολογίας ενός συστήματος ορολογίας.
- *Διαχείριση Σύνθετων Εννοιών*. Χρησιμοποίηση υπηρεσιών ορολογίας για την διευκόλυνση της εισαγωγής, της μετάφρασης, της απλοποίησης, και του ελέγχου ορθότητας σύνθετων εννοιών.

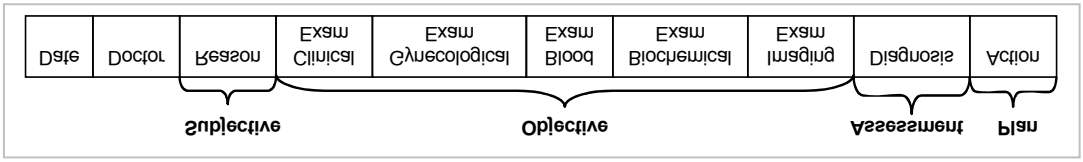
Προφανώς η παραπάνω λίστα δεν είναι πλήρης. Παρόλα αυτά είναι ένα αντιπροσωπευτικό δείγμα χρήσεων τέτοιων συστημάτων. Έχοντας υπόψη τα παραπάνω, το CORBAmed καθόρισε το σύνολο των λειτουργιών που πρέπει να παρέχει μια *υπηρεσία επερώτησης βασισμένη σε λεξικά* (Lexicon Query Service).

2.2. Το Ευρετήριο Ιατρικών Δεδομένων Ασθενών

(PCDD: Patient Clinical Data Directory)

Το *Patient Clinical Data Directory* [31] αποτελεί μια υλοποίηση του Ολοκληρωμένου Ηλεκτρονικού Φακέλου Υγείας (*Integrated Electronic HealthCare Record*- I-EHC) και εντάσσεται στα πλαίσια ανάπτυξης του *HYGEIAnet* [HYGEIAnet] (βλέπε σχήμα 4). Το HYGEIAnet είναι ένα συνεπές, επαναχρησιμοποιούμενο και επεκτάσιμο ολοκληρωμένο περιφερειακό δίκτυο τηλεματικών υπηρεσιών στην περιφέρεια της Κρήτης. Χρησιμοποιείται με σκοπό να επιτυγχάνεται η πρόσβαση και η ανάκληση κλινικών πληροφοριών/ δεδομένων ενός ασθενή. Οι πληροφορίες και τα δεδομένα αυτά βρίσκονται αποθηκευμένα σε απομακρυσμένες και κατανεμημένες περιοχές της περιφέρειας Κρήτης. Το PCDD *ευρετηριάζει* ασθενείς, όπως επίσης και πληροφορίες/ δεδομένα για τα κλινικά αντικείμενα των τμημάτων του ηλεκτρονικού τους φακέλου. Η πρόσβαση στην πληροφορία που περιέχει το PCDD προσφέρεται σε *εξουσιοδοτημένους* (authorized) χρήστες μέσω κατάλληλων IDL διεπαφών. Το βασικό κομμάτι του PCDD είναι ένας X.500/LDAP κατάλογος, ο οποίος διατηρεί καταχωρήσεις για τα δημογραφικά στοιχεία των ασθενών, τα κομμάτια του ηλεκτρονικού φακέλου τους, κλινική μετά-πληροφορία, και αναφορές στα κλινικά αντικείμενα.

Τα κλινικά αντικείμενα του καταλόγου, περιέχουν τα κλινικά δεδομένα που παράγονται κατά την επαφή του ασθενή με έναν ή περισσότερους ειδικούς από τον χώρο της υγείας. Αυτή η επαφή ονομάζεται *encounter*. Οι πιο κοινός τύπος του encounter είναι μια *επίσκεψη* σε μια κλινική ή ένα κέντρο υγείας. Οι καταχωρήσεις του encounter στον κατάλογο ακολουθούν το διεθνές αναγνωρισμένο και αποδεκτό *Subjective Objective Assessment Plan*- SOAP μοντέλο που παρουσιάζεται στο σχήμα 3.



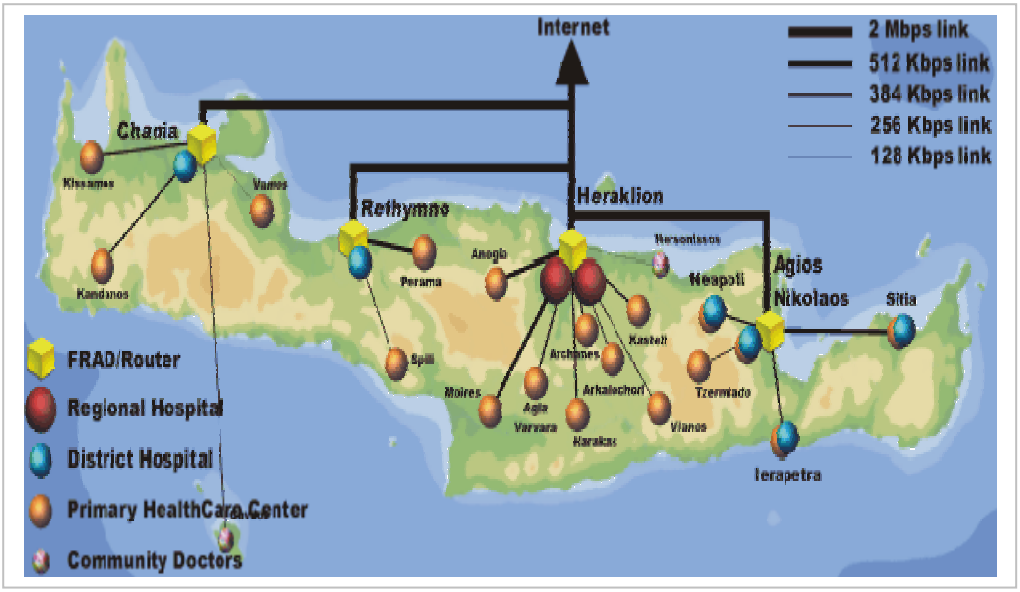
Σχήμα 3. *Οι καταχωρήσεις του encounter στον κατάλογο ακολουθούν το SOAP μοντέλο*

Πρέπει να σημειώσουμε ότι το PCDD δεν περιέχει την πληροφορία που έχει καταγραφεί για ένα encounter, αλλά *δείκτες* (links) για το αν υπάρχει πληροφορία και πού. Η πληροφορία παραμένει αποθηκευμένη στα αυτόνομα και κατανεμημένα κλινικά πληροφοριακά συστήματα; Με άλλα λόγια το PCDD δεν επαναντιγράφει (replicate) τα δεδομένα. Αυτό που το PCDD, και σε τελική ανάλυση το SOAP μοντέλο προσφέρει, είναι η *αφαίρεση* (abstraction) ή *γενίκευση* σε ένα πρώτο *μετά-επίπεδο* των δεδομένων και των πληροφοριών που βρίσκονται διάσπαρτες στα κατανεμημένα πληροφοριακά συστήματα. Αν θέλουμε λοιπόν να έχουμε πρόσβαση στις συγκεκριμένες πληροφορίες, που αφορούν στο encounter (δηλαδή στο αντίστοιχο *σύστημα τροφοδοσίας- feeder system*), το PCDD μας προσφέρει την απαραίτητη πληροφορία για να πάρουμε τα δεδομένα από την πηγή τους. Οι μέθοδοι πρόσβασης που προσφέρονται είναι συνήθως αναφορές σε CORBA αντικείμενα και HTTP URLs. Η *σημασιολογική αντιστοίχηση* αυτών των δεικτών με την κλινική πληροφορία του ασθενή, διεκπεραιώνεται μέσω της ανάπτυξης εξειδικευμένων για κάθε πληροφοριακό σύστημα '*προσαρμοστών*' (*adapters* ή *wrappers)*. Έτσι, εάν για παράδειγμα κάποιος ασθενής έχει υποβληθεί σε εξέταση αίματος, σε κάποιες από τις κλινικές μονάδες του όλου κατανεμημένου συστήματος, ο προσαρμοστής του αντίστοιχου πληροφοριακού συστήματος

εγγράφει και αποθηκεύει μια ανάλογη εξέταση αίματος στο PCDD, με την μορφή δείκτη (link/ pointer). Ο χρήστης μπορεί να ακολουθήσει (κάνοντας απλά click) αυτόν το δείκτη και να έχει άμεση πρόσβαση στις λεπτομέρειες της εξέτασης.

Προκείμενου λοιπόν αυτή η πληροφορία να λαμβάνεται με ένα γενικά αποδεκτό και *ομοιογενή* τρόπο, και να επιτρέπει την περαιτέρω επεξεργασία της με απώτερο στόχο την *εξαγωγή σημαντικής γνώσης* από αυτήν, παραπέμπουμε τον χρήστη στα επόμενα κεφάλαια, όπου δίνουμε λύση σε αυτό ακριβώς το πρόβλημα.

Σημείωση. Η σημασιολογική αντιστοίχηση ανάμεσα στο PCDD και στα συστήματα τροφοδοσίας είναι υπευθυνότητα των κατανεμημένων συστημάτων τροφοδοσίας; όπου *το σχήμα του αντίστοιχου πληροφοριακού μοντέλου αντιστοιχίζεται στο σχήμα του πληροφοριακού μοντέλου του PCDD*.



Σχήμα 4. *Περιφερειακό Δίκτυο Τηλεματικών Υπηρεσιών της Κρήτης (HYGEIAnet)*

Κεφάλαιο 3:
Σημασιολογική Ομογενοποίηση Ετερογενών Κλινικών Πληροφοριών

Έχοντας ορίσει και υιοθετήσει ένα '*meta-data*' επίπεδο, το οποίο δείχνει σε συγκεκριμένες πηγές πληροφορίας και αρμόδια-ζητούμενα τμήματα πληροφορίας του ασθενή, το επόμενο πρόβλημα είναι πως θα εξάγουμε αυτήν την κλινική πληροφορία η οποία βρίσκεται αποθηκευμένη στα αυτόνομα κλινικά πληροφοριακά συστήματα. Αυτό είδαμε ότι αντιμετωπίζεται μέσω των CORBA IDL interfaces.

Ωστόσο η CORBA δεν προσφέρει ιδιαίτερη (έως καθόλου) βοήθεια σε επίπεδο παροχής γνώσης, καθώς δεν εγγυάται και δεν εξασφαλίζει ότι συγκεκριμένα τμήματα (components) της πληροφορίας, μπορούν να λειτουργήσουν μαζί. Επιπρόσθετα μολονότι τα IDL καθορίζουν την αναγκαία σύνταξη για συνεργασία και πρόσβαση σε κατανεμημένες πληροφορίες δεν περιγράφουν την σημασιολογία και το λόγο ύπαρξης αυτής της πληροφορίας. Η προσθήκη της *σημασιολογίας* θα παρείχε ουσιαστικά αυτό που λείπει από τα IDL: *πληροφορία για το νόημα και την ύπαρξη ενός 'component', πληροφορία για το τι επιδιώκει να πραγματοποιήσει, πληροφορία για την σχέση ανάμεσα στα δεδομένα εισόδου και εξόδου μιας διαδικασίας* [35].

Αυτή η λειτουργία έρχεται να πραγματοποιηθεί όπως θα δούμε με την εισαγωγή και παρουσίαση ενός μοντέλου δεδομένων του πεδίου εφαρμογής (domain *data-model)* και την συνεργασία του με το αντίστοιχο πεδίο οντολογίας (*domain-ontology)*. Συνδυάζοντας τα παραπάνω με την σύγχρονη και πολλά υποσχόμενη τεχνολογία της XML, νομίζουμε ότι προσεγγίζουμε ικανοποιητικά τη λύση στο ζητούμενο πρόβλημα, που δεν είναι άλλο από την σημασιολογική ομογενοποίηση της ετερογενούς κλινικής πληροφορίας, την οποία και επιθυμούμε να επεξεργαστούμε.

3.1. Η Υπηρεσία Clinical Observation Access Service (COAS) και το σχετικό μοντέλο

Το *Clinical Observation Access Service* [19] είναι ένα σύνολο από διεπαφές και δομές δεδομένων με τα οποία οι εξυπηρετητές μπορούν να παρέχουν κλινικές παρατηρήσεις (clinical observations), και είναι πλέον στην τελική του έκδοση από τον Απρίλιο του 1999. Όπως θα δούμε στο *COAS* μοντέλο θα βασίσουμε την *DTD* γραμματική (Document Type Description grammar) που θα παρουσιάσουμε και κατ' επέκταση, η αναπαράσταση των αρχείων στα οποία θα αναζητήσουμε την γνώση, είναι εξολοκλήρου βασισμένη στην μορφή και δομή του. Για το λόγο αυτό θα το παρουσιάσουμε με ιδιαίτερη λεπτομέρεια.

Ο όρος *κλινικές παρατηρήσεις* ορίστηκε από την CORBAmed ως ένα σημαντικό κομμάτι τις πληροφορίας που καταγράφεται για κάθε ασθενή. Παραδείγματα κλινικών παρατηρήσεων είναι τα εξής: *εργαστηριακές εξετάσεις*, *βιοσήματα*, *υποκειμενικές και αντικειμενικές παρατηρήσεις* και *εκτιμήσεις*, παρατηρήσεις και *μετρήσεις* που παρέχει κάποιος ειδικός όπως ένας ακτινολόγος ή ένας παθολόγος ο οποίος αναλύει εικόνες και άλλα *δεδομένα πολύ-μέσων* (multi-media data).

Μερικά κοινά γνωρίσματα αυτών των παρατηρήσεων είναι: (α) οι κλινικές παρατηρήσεις αναφέρονται σε αντικείμενα στα οποία παρέχεται ιατρική φροντίδα,

όπως ένας ασθενής, ή ένα πληθυσμός, (β) αναπαριστούν την κατάσταση του αντικειμένου στο χρόνο, είτε σε μια συγκεκριμένη χρονική στιγμή, είτε σε κάποιο συγκεκριμένο χρονικό διάστημα, και (γ) γίνονται ή καταγράφονται από ένα μηχάνημα ή από κάποιον ειδικό και διακρίνονται από κάποιο βαθμό αξιοπιστίας.

Τα παραπάνω γνωρίσματα θα παίξουν σημαντικό ρόλο στην κατανόηση τόσο του συνόλου των λειτουργιών που θα πρέπει να ικανοποιεί μια υπηρεσία πρόσβασης σε κλινικές παρατηρήσεις, όσο και του μοντέλου αναφοράς που θα αναλυθεί στην συνέχεια.

3.1.1. Απαιτήσεις

Το πρώτο και υποχρεωτικό σύνολο λειτουργιών των υπηρεσιών πρόσβασης σε κλινικές παρατηρήσεις αποτελείται από τις ακόλουθες δεσμεύσεις.

1. Οι κλινικές παρατηρήσεις πρέπει να μπορούν να *αναζητούνται* και να *μεταφέρονται*.
2. Οι κλινικές παρατηρήσεις πρέπει να *φιλτράρονται*, όπως για παράδειγμα βάση ενός ασθενή, βάση του τύπου της παρατήρησης, βάση της κατάστασης και/ή του χρόνου.
3. Πρέπει να υπάρχει *μηχανισμός επερώτησης* για τις διαθέσιμες παρατηρήσεις.
4. Πρέπει να παρέχεται *πρόσβαση* σε πληροφορίες που αφορούν το "πλαίσιο" (context) μιας παρατήρησης.
5. Πρέπει να υπάρχει ένα προκαθορισμένο σύνολο από *τύπους* παρατηρήσεων.
6. Πρέπει να υπάρχει η ικανότητα να χρησιμοποιούνται κοινά αποδεκτά και δημόσια διαθέσιμα *λεξικά*.

Το δεύτερο και τελευταίο σύνολο λειτουργιών είναι προαιρετικό:

7. Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν ένα μηχανισμό ο οποίος θα παρέχει πρόσβαση σε μελλοντικές παρατηρήσεις.
8. Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν την ικανότητα υποστήριξης δυναμικής ανακάλυψης των υποστηριζόμενων τύπων παρατηρήσεων και δεδομένων.
9. Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν την ικανότητα γενικού φιλτραρίσματος των επερωτήσεων.
10. Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν την ικανότητα για χρήση των υπηρεσιών Trader (Trader services).
11. Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν την ικανότητα πρόσβασης στο ιστορικό αναπροσαρμογής των δεδομένων.
12. Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν ένα πληροφοριακό μοντέλο αναφοράς και τα αντίστοιχα IDL αρχεία.
13. Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν την ικανότητα χρήσης τόσο τοπικών και περιορισμένων λεξικών, όσο κοινά αποδεκτών και διαθέσιμων.

Οι υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις μπορούν να παρέχουν την ικανότητα χρήσης των υπηρεσιών επερωτήσεων με βάση λεξικά (LQS), έτσι ώστε *να υποστηρίζονται πολλά λεξικά*.

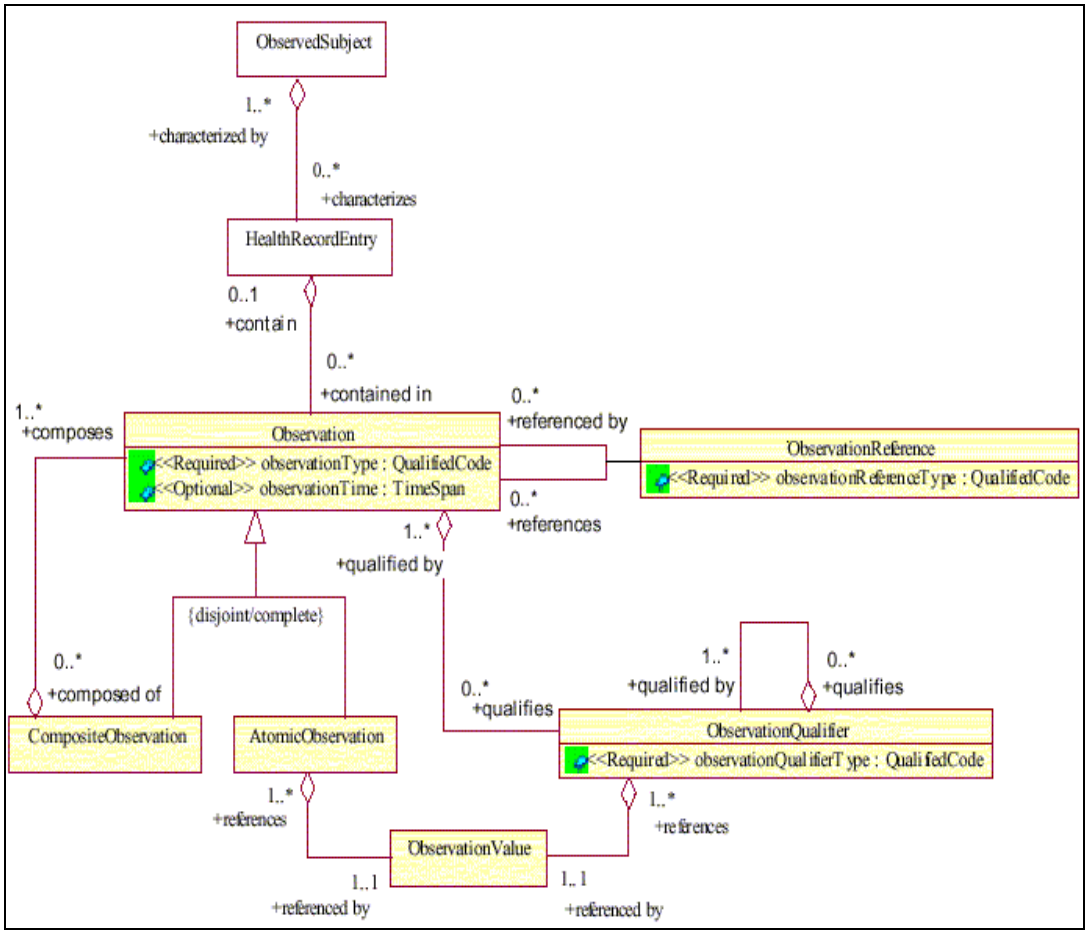
Οι παραπάνω λειτουργίες καλύπτουν σχεδόν όλο το φάσμα των λειτουργιών που μια υπηρεσία πρόσβασης σε κλινικές παρατηρήσεις θα μπορούσε να παρέχει. Αυτό που πρέπει να τονιστεί είναι ότι οι εφαρμογές οι οποίες διατηρούν κλινικές παρατηρήσεις συνήθως υλοποιούν ένα μόνο μέρος αυτών των λειτουργιών, πράγμα απολύτως φυσιολογικό, διότι κάθε εφαρμογή προσπαθεί να καλύψει τις ανάγκες των χρηστών μέσα στα πλαίσια του περιβάλλοντος που αυτή λειτουργεί. Το COAS δεν προσδιορίζει τον τρόπο με τον οποίο πρέπει να δομούνται τέτοιου είδους εφαρμογές, αλλά καθορίζει τον τρόπο με τον οποίο αυτές οι εφαρμογές

επικοινωνούν με τον υπόλοιπο κόσμο. Το ίδιο ισχύει και για το μοντέλο αναφοράς που αναλύεται στην συνέχεια.

3.1.2. Περιγραφή του Μοντέλου Αναφοράς

Σε αυτό το εδάφιο θα περιγραφεί το πληροφοριακό μοντέλο αναφοράς για τις υπηρεσίες πρόσβασης σε κλινικές παρατηρήσεις. Το μοντέλο, όπως φαίνεται και στο σχήμα 5, είναι αρκετά απλό. Παρόλα αυτά είναι αρκετά ισχυρό, ώστε να προσφέρει την *επεκτασιμότητα*, στοιχείο απαραίτητο για εφαρμογές στο χώρο της παροχής ιατρικής φροντίδας. Είναι το *μοντέλο αναπαράστασης* στο οποίο βασίζονται οι διαδικασίες *ανακάλυψης* και *εκμαίευσης* γνώσης που αναπτύσσουμε.

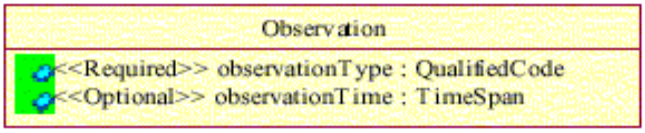
Οι βασικές οντότητες του μοντέλου, καθώς και οι σχέσεις ανάμεσά τους, παρουσιάζονται στο σχήμα 5, ενώ στην συνέχεια ακολουθεί μια σύντομη περιγραφή, για την καλύτερη κατανόηση τους. Για περισσότερες λεπτομέρειες, ο αναγνώστης παραπέμπεται στο [19].



Σχήμα 5. *Το COAS μοντέλο*

Οι οντότητες `Health Record Entry` και `Observed Subject` αναπαρίστανται στο μοντέλο για να δειχθεί ότι μπορούν να ταιριάξουν σε αυτό, όμως δεν υποστηρίζονται από το υπάρχον μοντέλο. Συνεπώς μια υπηρεσία πρόσβασης σε κλινικές παρατηρήσεις αποτελείται από τις εξής οντότητες:

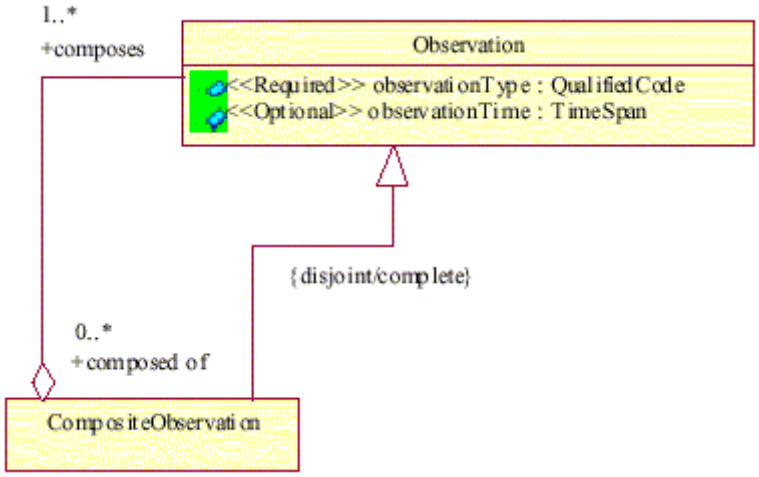
Observation



Σχήμα 6. *Αναπαράσταση του* `Observation`

Το Observation είναι μια αφηρημένη κλάση και τα attributes που περιέχει τα κληρονομούν οι οντότητες `Composite Observation` και `Atomic Observation`, οι οποίες είναι υποκλάσεις του `Observation`. Επίσης είναι *πλήρες* (complete) και *διακριτό* (disjoint). Ο χαρακτηρισμός 'πλήρες' σημαίνει ότι δεν υπάρχουν άλλες υποκλάσεις αυτού, εκτός από τις δύο που προαναφέραμε, και ο χαρακτηρισμός 'διακριτό' σημαίνει ότι οι περιπτώσεις αυτού του τύπου μπορούν να έχουν μόνο μία από τις δύο υποκλάσεις ως τύπο. Όσον αφορά τις σχέσεις του με άλλες οντότητες, το `Observation` σχετίζεται με το `Composite Observation` (ένα ή περισσότερα `Observations` συνιστούν μηδέν ή περισσότερα `Composite Observations`), με το `Observation Reference` (μηδέν ή περισσότερα `Observations` σχετίζονται με μηδέν ή περισσότερα `Observations`), και με το `Observation Qualifier` (ένα ή περισσότερα `Observations` προσδιορίζονται από μηδέν ή περισσότερα `Observation Qualifiers`).

Composite Observation

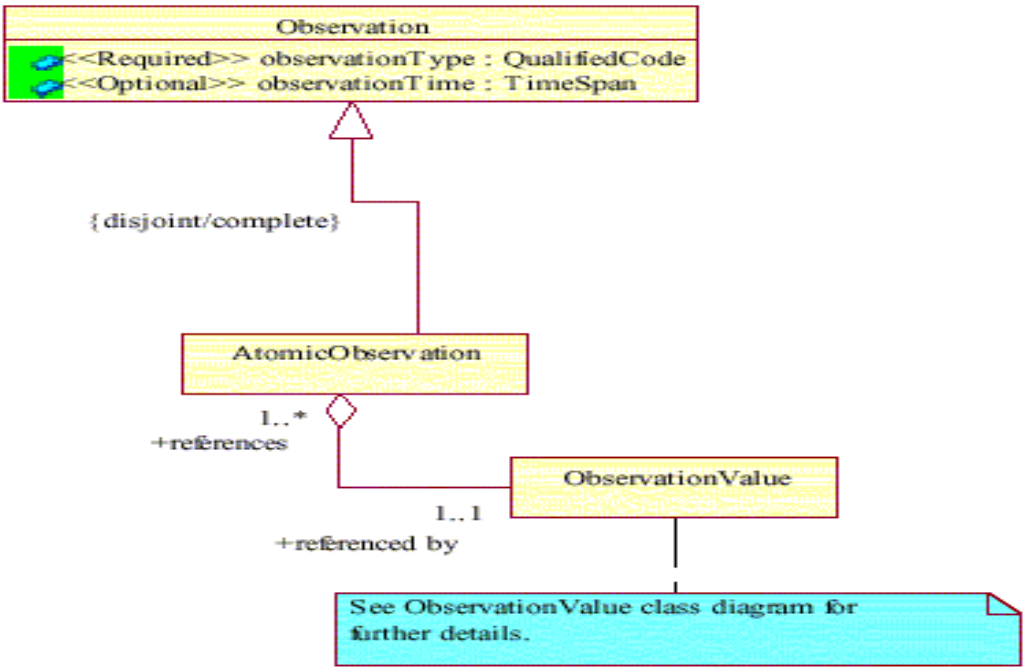


Σχήμα 7. *Αναπαράσταση του* `Composite Observation`

Το `Composite Observation` αναπαριστά ένα σύνολο από `Observations`. Ένα τέτοιο σύνολο μπορεί να είναι για παράδειγμα μια πλήρης εξέταση αίματος. Στην ουσία πρόκειται για `Observations` τα οποία αποτελούνται από άλλα πιο απλά `Observations`. Αυτό φαίνεται και στο σχήμα 7, δηλαδή, μηδέν ή περισσότερα `Composite Observations` αποτελούνται από ένα ή περισσότερα `Observations`, ενώ ένα η περισσότερα `Observations` συνθέτουν μηδέν ή περισσότερα `Composite Observations`. Το `Composite Observation` είναι υποκλάση του `Observation` και συνεπώς κληρονομεί τα χαρακτηριστικά ή ιδιότητες του (attributes). Τέλος όπως βλέπουμε το `Composite Observation`

δεν σχετίζεται με κάποια τιμή, αλλά χρησιμοποιείται για να δώσει σημασιολογικό νόημα στις οντότητες που περιέχει (Atomic Observations). Έτσι η εξέταση αίματος (Composite Observation) όπως αναφέραμε, αποτελείται από επιμέρους Atomic Observations, όπως είναι τα λευκά και ερυθρά αιμοσφαίρια, η αιμοσφαιρίνη κ.τ.λ., τα οποία συνοδεύονται από κάποιες *τιμές μέτρησης*.

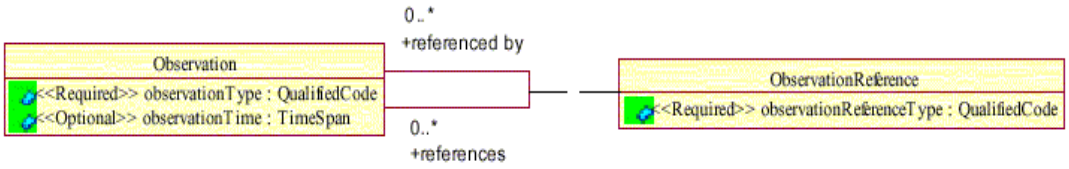
Atomic Observation



Σχήμα 8. *Αναπαράσταση του* Atomic Observation

Το Atomic Observation αναπαριστά ένα απλό Observation και σχετίζεται με κάποια τιμή (Observation Value). Κάθε Atomic Observation σχετίζεται με μία και μόνο μία τιμή, ενώ μια τιμή αναφέρεται από ένα ή περισσότερα Atomic Observations. Τέλος, επειδή το Atomic Observation είναι υποκλάση του Observation προφανώς κληρονομεί τα χαρακτηριστικά του. Παραδείγματα όπως ήδη αναφέραμε τέτοιων εξετάσεων είναι , η αιμοσφαιρίνη, τα λευκά και ερυθρά αιμοσφαίρια.

Observation Reference

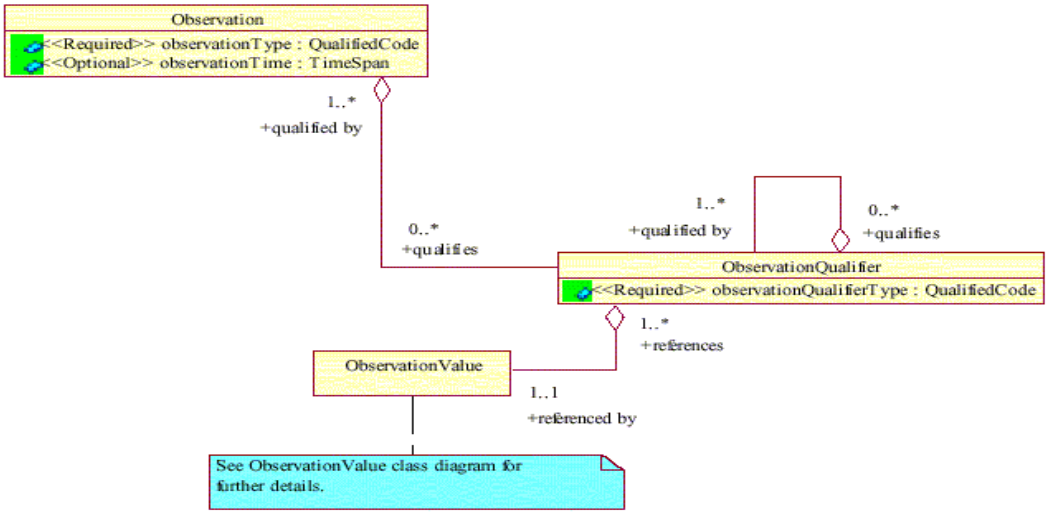


Σχήμα 9. *Αναπαράσταση του* Observation Reference

Το Observation Reference είναι μία κλάση, η οποία προσδιορίζει τον τύπο των σχέσεων ανάμεσα στα Observations και το χαρακτηριστικό της

"observationReferenceType" πρέπει να προέρχεται από ένα καλά ορισμένο σύστημα ορολογίας.

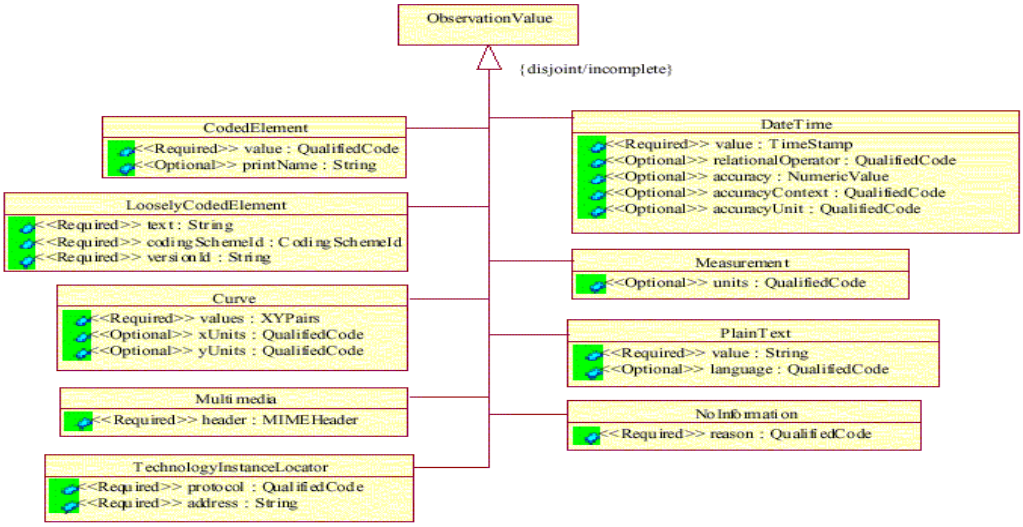
ObservationQualifier



Σχήμα 10. *Αναπαράσταση του* Observation Qualifier

Το Observation Qualifier έχει ως σκοπό να προσδιορίσει περαιτέρω το Observation καταγράφοντας το "*περιβάλλον*" του. Το χαρακτηριστικό του "observationQualifierType" προέρχεται από ένα καλά ορισμένο σύστημα ορολογίας και αναλαμβάνει να διεκπεραιώσει αυτό το σκοπό. Το ίδιο ισχύει και για το Observation Value με το οποίο συσχετίζεται. Όσον αφορά τις σχέσεις του, μηδέν ή περισσότερα ObservationQualifiers μπορούν να προσδιορίσουν ένα ή περισσότερα Observations, ενώ μηδέν ή περισσότερα Observation Qualifiers προσδιορίζουν ένα ακριβώς ObservationValue. Τέλος, μηδέν ή περισσότερα ObservationQualifiers μπορούν να προσδιορίσουν και να τροποποιήσουν ένα ή περισσότερα ObservationQualifiers.

Observation Value



Σχήμα 11. *Αναπαράσταση του Observation Value*

Το `Observation Value` είναι ένας 'αφηρημένος' τύπος. Αυτό είναι απόλυτα λογικό, καθώς η τιμή ενός `Observation` μπορεί να έχει οποιαδήποτε μορφή. Τέτοιες μορφές είναι το απλό κείμενο, ένα νούμερο, κάποια εικόνα κλπ. Το COAS ορίζει ένα σύνολο από τύπους, ως υποκλάσεις του `Observation Value`, το οποίο ναι μεν είναι διακριτό, αλλά όχι πλήρες ακόμα. Όσον αφορά τις σχέσεις του με άλλες οντότητες του μοντέλου, ένα και μόνο ένα `Observation Value` σχετίζεται με ένα ή περισσότερα `Atomic Observations`, και προσδιορίζεται επιπλέον από ένα ή περισσότερα `Observation Qualifiers`.

Για τις αναλυτικές περιγραφές ο αναγνώστης παραπέμπεται στο [19].

3.2. Οντολογία Ιατρικού Πεδίου

Βασικό κριτήριο για την επιτυχία μιας υπηρεσίας πληροφόρησης η οποία προσπελαύνει και ανακλά δεδομένα από κατανεμημένες πηγές πληροφόρησης, είναι η ικανότητα της να χειρίζεται αποδοτικά και έξυπνα την ετερογενή φύση της εκεί αποθηκευμένης πληροφορίας. Ας θεωρήσουμε ως παράδειγμα, στα πλαίσια μιας εφαρμογής, μια ιατρική βάση δεδομένων η οποία βρίσκεται σε διαφορετικές γεωγραφικές περιοχές. Εάν η εφαρμογή δεν διαθέτει ένα *μοναδικό και ενοποιημένο σχήμα κωδικοποίησης* για διαγνώσεις, τότε υπάρχει η περίπτωση οι ανεξάρτητες και αυτόνομες πηγές αποθήκευσης να έχουν καταγράψει με διαφορετικό όνομα και κωδικό την ίδια διάγνωση! Για παράδειγμα σε μια καρδιολογική μονάδα το όνομα για την 'πίεση' μπορεί να έχει εγγραφεί ως "DIASTOLICPRESSURE", ενώ σε μια άλλη ως "PRESSURE". Σε μια τέτοια περίπτωση, μολονότι έχουμε πρόσβαση και στις δυο πηγές πληροφόρησης, δεν είναι καθόλου προφανές πως θα λειτουργήσει ένας αλγόριθμος εξαγωγής κανόνων συσχέτισης, προκειμένου να ανακαλύψει συσχετίσεις, για παράδειγμα, μεταξύ διαγνώσεων και συμπτωμάτων (θα τις θεωρήσει ως μια κοινή διάγνωση ή ως δύο διαφορετικές;).

Από την άλλη πλευρά, μπορεί να αντιμετωπίσουμε την περίπτωση ολοκλήρωσης δυο (ή και περισσότερων) βάσεων δεδομένων. Ας θεωρήσουμε για παράδειγμα την περίπτωση όπου ένα κλινικό πληροφοριακό σύστημα εγγράφει και αποθηκεύει *κλινική πληροφορία* (clinical findings) μιας ομάδας ασθενών, ενώ ένα άλλο εγγράφει *εργαστηριακές εξετάσεις* (laboratory examination findings) για την ίδια ομάδα ασθενών. *Πως* θα μπορέσουμε σε αυτήν την περίπτωση να συσχετίσουμε και τις δυο βάσεις ώστε να ανακαλύψουμε ενδιαφέρουσες συσχετίσεις ανάμεσα σε διαγνώσεις/ ασθένειες και σε εργαστηριακά ευρήματα;

<ICD10-code> :: <***MedicalReferenceTerm***>-1: <Synonym-1$_1$, Synonym-1$_2$, …, Synonym-1$_{n1}$>

<ICD10-code> :: <***MedicalReferenceTerm***>-2: <Synonym-2$_1$, Synonym-2$_2$, …, Synonym-2$_{n2}$>

…… more medical terms ……

<ICD10-code> :: <***MedicalReferenceTerm***>-k: <Synonym-k$_1$, Synonym-k$_2$, …, Synonym-k$_{nk}$>

Σχήμα 12. *Η μορφή του CCTR: Common Clinical Term Representation file (ICD10-code: The International Coding for Diseases -release 10- code)*

Ο μοναδικός τρόπος για να αντιμετωπίσουμε προβλήματα τέτοιας προέλευσης, είναι να συμπεριλάβουμε και να ενσωματώσουμε στο σύστημα μας ένα

εξειδικευμένο (ιατρικό στην περίπτωση μας) και αρμόδιο *πεδίο οντολογίας*. Κινούμενοι λοιπόν σε αυτήν την κατεύθυνση έχουμε σχεδιάσει την μορφή ενός αρχείου (σε XML), στο οποίο αποθηκεύονται κοινά και διεθνώς αποδεκτά ονόματα και κωδικοί ιατρικών όρων. Το αρχείο αυτό ονομάζεται CCTR (*Common Clinical Term Representation*). Η γενική μορφή του CCTR φαίνεται στο σχήμα 12.

Η κατασκευή του CCTR βασίζεται και είναι σύμφωνη με τα πρότυπα του UMLS: *Unified Medical Language System* [36] και του ICD: *International Coding for Diseases* (version 10). Με την αναφορά στον ICD κωδικό, διαφορετικά λεξικά, για διαφορετικές γλώσσες και από διαφορετικά κλινικά πληροφοριακά συστήματα, μπορούν εύκολα να αντιστοιχηθούν και να προσαρμοστούν. Στην πραγματικότητα το σύστημα *μετά-θησαυρών* που χρησιμοποιεί το UMLS προσφέρει τέτοιους μηχανισμούς, και έχοντας δημιουργήσει ένα λεξικό όρων με χρήση του UMLS, μπορούμε εύκολα να το μετασχηματίσουμε στο CCTR format.

Έτσι, στο σύστημα μας κάθε όρος (δηλαδή, `Atomic Observation`) προτού καταχωρηθεί στις δομές μας και επιτελεστούν πάνω του όλες οι KDD διεργασίες, αναζητείται στο CCTR και αν είναι κάποιος *Synonym Term*, αντικαθιστάται από τον αντίστοιχο *Main/ Reference Term*, παρέχοντας με αυτόν τον τρόπο στο σύστημα μας την επιδιωκόμενη *ομογενοποίηση*.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--                    Last    VERSION
        XML Use For   Clinical Observations Model
            Programmer : Kwstas Christofis
 -->
<!DOCTYPE Query [
                <!ELEMENT Query (Observation)*>
                <!ATTLIST Query TimeOfQuery CDATA #REQUIRED
    WhoAsk CDATA #REQUIRED SelectedQuery CDATA #REQUIRED  GeographicRegion  CDATA #REQUIRED TimeRange CDATA #REQUIRED
Gender CDATA #REQUIRED  Age CDATA #REQUIRED>
                <!ELEMENT Observation (AtomicObservation | CompositeObservation)>
                <!ATTLIST Observation Patient_Id CDATA #REQUIRED  Information_System CDATA #REQUIRED Visit_Id  CDATA #REQUIRED>
                <!ELEMENT CompositeObservation ((AtomicObservation | CompositeObservation)* ,ObservationReference*, ObservationQualifier*)>
                <!ATTLIST CompositeObservation ObservationType CDATA #REQUIRED
   ObservationTime  CDATA #IMPLIED>
                <!ELEMENT AtomicObservation (ObservationValue,ObservationReference*, ObservationQualifier*)>
                <!ATTLIST AtomicObservation ObservationType CDATA #REQUIRED
   ObservationTime  CDATA #IMPLIED>
                <!ELEMENT ObservationValue ((PlainText | NoInformation | CodeElement | LooselyCodeElement | Curve | MultiMedia | DateTime |
Measurement |TechnologyInstanceLocator), ObservationQualifier*)>
                <!ELEMENT PlainText EMPTY>
                <!ATTLIST PlainText Value CDATA #REQUIRED
    language CDATA #IMPLIED>
                <!ELEMENT NoInformation EMPTY>
                <!ATTLIST NoInformation reason CDATA #REQUIRED>
                <!ELEMENT CodeElement EMPTY>
                <!ATTLIST CodeElement value CDATA #REQUIRED
   printName CDATA #IMPLIED>
                <!ELEMENT LooselyCodeElement EMPTY>
                <!ATTLIST LooselyCodeElement text  CDATA #REQUIRED
   codingSchemeID CDATA #REQUIRED
   versionID CDATA #REQUIRED>
                <!ELEMENT Curve EMPTY>
                <!ATTLIST Curve values CDATA #REQUIRED
   xUnits CDATA #IMPLIED
   yUnits CDATA #IMPLIED>
                <!ELEMENT Multimedia EMPTY>
                <!ATTLIST Multimedia header CDATA #REQUIRED>
                <!ELEMENT DateTime EMPTY>
                <!ATTLIST DateTime value CDATA #REQUIRED
   relationalOperator  CDATA #IMPLIED
   accuracy  CDATA #IMPLIED
   accuracycontext  CDATA #IMPLIED
   accuracyUnit  CDATA #IMPLIED>
                <!ELEMENT Measurement EMPTY>
                <!ATTLIST Measurement NumericValue CDATA #REQUIRED  units CDATA #IMPLIED>
                <!ELEMENT TechnologyInstanceLocator EMPTY>
                <!ATTLIST TechnologyInstanceLocator protocol CDATA #REQUIRED
   address CDATA #REQUIRED>
                <!ELEMENT ObservationQualifier (QualifiedBy | EMPTY)*>
                <!ATTLIST ObservationQualifier ObservationQualifierType CDATA #REQUIRED>
                <!ELEMENT QualifiedBy (ObservationQualifier)+>
                <!ELEMENT ObservationReference EMPTY>
                <!ATTLIST ObservationReference ObservationReferenceType CDATA  #REQUIRED ObservationReferenceName CDATA  # REQUIRED
Patient_Id CDATA  # REQUIRED Information_System CDATA  # REQUIRED Visit_Id CDATA  #REQUIRED ObservationTime CDATA  # IMPLIED  >
]>
```

Σχήμα 13. *Η εξαγόμενη COAS Compatible DTD*

3.3. Ενιαία και Ομογενοποιημένη Αναπαράσταση Κατανεμημένης και Ετερογενούς Ιατρικής Πληροφορίας

Προκειμένου λοιπόν να αξιοποιηθεί και να επεξεργαστεί η πληροφορία που λαμβάνεται κάθε φορά (μέσω των COAS διεπαφών) από τα πληροφοριακά μας συστήματα (Κέντρα Υγείας) και με τη βοήθεια των δεικτών που παρέχονται από το PCDD (όπως αναφέραμε στο προηγούμενο κεφάλαιο), χρειάζεται και απαιτείται ο ορισμός κάποιας καλά ορισμένης και γενικά αποδεκτής μορφής *σύνταξης* και *αναπαράστασης* της εξαγόμενης πληροφορίας.

Με δεδομένο ότι η σύγχρονη τεχνολογία αναδεικνύει ως βέλτιστη λύση- στην κατεύθυνση της περιγραφής και αναπαράστασης πληροφοριών, την χρήση της XML [37,38], προτείνουμε και υλοποιούμε την κατασκευή μιας DTD γραμματικής, συμβατής με το COAS μοντέλο, με σκοπό την παραγωγή συμβατών με αυτήν *COAS/XML αρχείων***.**

Η DTD η οποία έχει αναπτυχθεί όπως θα δειχθεί παρακάτω, βρίσκεται σε πλήρη αντιστοιχία με το COAS μοντέλο που περιγράψαμε στην προηγούμενη ενότητα, και έχει ενσωματωμένα κάποια επιπρόσθετα χαρακτηριστικά, απαραίτητα για την επίτευξη των στόχων μας (εξαγωγή στατιστικών αποτελεσμάτων, κανόνων αλλησυσχέτισης, μελλοντικών θεμάτων user-profiling κ.τ.λ)

Στο σχήμα 13 (στη προηγούμενη σελίδα), παραθέτουμε την μορφή της εξαγόμενης DTD. Στο σχήμα 14 παρακάτω, δίνουμε ένα ενδεικτικό *instance* της DTD το οποίο αντιστοιχεί σε επερώτηση που έχει γίνει στο πληροφοριακό σύστημα του Σπηλίου και με το οποίο ζητούνται να *ανακληθούν* όλες οι εκεί παρατηρούμενες βιοχημικές εξετάσεις.

```
<Query WhoAsk = "Kwstas Christofis " TimeOfQuery= " "  SelectedQuery = "1" GeographicRegion =  " "      Range = "
" Gender = " " Age = " ">
    <Observation Patient_Id = "87" Information_System = "spili" Visit_Id = "248">
      <CompositeObservation ObservationType="BIOCHEMICAL_EXAM"          ObservationTime="1997-10-06
11:41:15.291000">
          <AtomicObservation ObservationType="ΣΑΚΧΑΡΟ" ObservationTime="1997-10-06 11:41:15.291000">
                                  <ObservationValue>
                                          <PlainText Value = "222.000000"/>
                                  </ObservationValue>
          </AtomicObservation>

          <AtomicObservation ObservationType="ΟΥΡΙΑ" ObservationTime="1997-10-06 11:41:15.291000">
                                  <ObservationValue>
                                          <NoInformation reason = "Not Asked"/>
                                  </ObservationValue>
          </AtomicObservation>
      </CompositeObservation>
    </Observation>
</Query>
```

Σχήμα 14. *Ένα instance της παραγόμενης DTD*

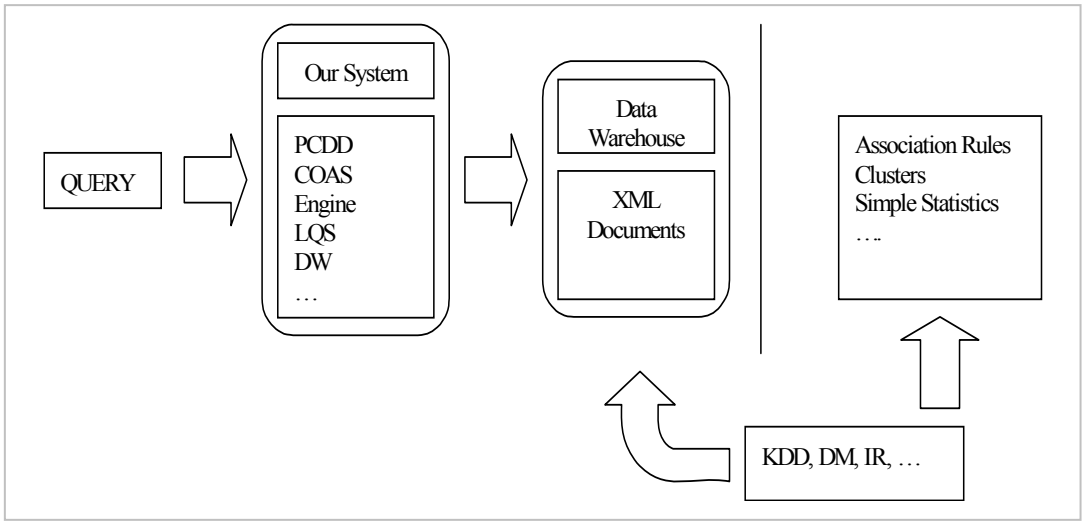
Σε αρχεία λοιπόν της παραπάνω μορφής θα βασίσουμε όλες τις λειτουργίες ανακάλυψης γνώσης που θα παρουσιάσουμε και θα αναπτύξουμε στα επόμενα κεφάλαια.

3.4. Μια αρχική εικόνα του συστήματος

Έχοντας αναφερθεί στο σημείο αυτό σε θέματα πρόσβασής, ομογενοποίησης και τυποποίησης/ αναπαράστασης της λαμβανόμενης πληροφορίας παραθέτουμε στο σχήμα 15 ένα πρώτο και απλό σενάριο χρήσης του συστήματος μας με σκοπό να γίνει κατανοητός ο τρόπος με τον οποίο συνδέονται και αλληλοσυνεργάζονται τα

επιμέρους τμήματα (όπως περιγράφηκαν έως τώρα), αλλά και για να προϊδεάσουμε το χρήστη με αυτά που πρόκειται να ακολουθήσουν.

Με την διαδικασία παραγωγής και ανακάλυψης κανόνων συσχέτισης θα ασχοληθούμε στα κεφάλαια 6-7, ενώ με την εξαγωγή απλών στατιστικών αποτελεσμάτων θα ασχοληθούμε στο επόμενο κεφάλαιο.



Σχήμα 15. *Παρουσίαση Συνολικού Συστήματος*

Κεφάλαιο 4:
Αναπαράσταση και Επεξεργασία Κλινικής Πληροφορίας – Το Πρότυπο της XML

Στο κεφάλαιο αυτό παρουσιάζουμε την μορφή των δομών που προκύπτουν από την επεξεργασία των συμβατών με την DTD μας XML αρχείων, εξηγούμε τα οφέλη (benefits) που προκύπτουν από την χρήση αυτών των δομών και κατ' επέκταση δικαιολογούμε και υποστηρίζουμε την επιλογή χρήσης του προτύπου XML. Τέλος, παρουσιάζουμε την μια από τις δύο μορφές *γνώσης* όπου αναζητούμε σε αυτήν την εργασία, τα *στατιστικά ιατρικά συμπεράσματα* τα οποία εκμαιεύονται μέσω της επεξεργασίας κατάλληλα δημιουργούμενων προτύπων δομών δεδομένων που θα παρουσιάσουμε και θα αναλύσουμε.

4.1. Parsing XML Αρχείων

Ξεκινώντας θα πρέπει να σημειώσουμε, ώστε να γίνει απόλυτα σαφές, ότι η διαδικασία επεξεργασίας της πληροφορίας (*filtering*, *indexing*, *pre-processing*, *parsing*) με σκοπό την εξαγωγή χρήσιμης κλινικής πληροφορίας, λαμβάνει χώρα εξολοκλήρου "πάνω" στα παραγόμενα XML documents, όπως αυτά προκύπτουν με τη διαδικασία που περιγράψαμε στα προηγούμενα κεφάλαια.

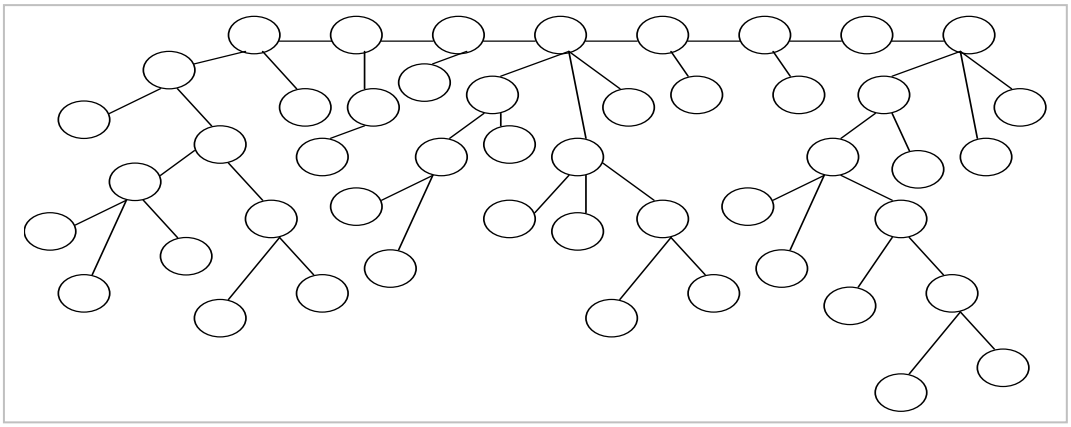
Για το λόγο αυτό έχει σχεδιαστεί και υλοποιηθεί ένας *parser* (με χρήση της Microsoft Visual C++ 6.0), που σκοπό έχει να κάνει parsing XML documents συμβατά με την σχετική DTD αναφοράς, και να τοποθετεί τις πληροφορίες σε κατάλληλες δομές. Στη συνέχεια οι δομές αυτές επεξεργάζονται και παρέχουν την ζητούμενη γνώση και πληροφορία. Στο κεφάλαιο αυτό θα περιγράψουμε την εξαγωγή απλών στατιστικών συμπερασμάτων, ενώ στο κεφάλαιο 7 θα περιγράψουμε αναλυτικά τον τρόπο εκμαίευσης και δημιουργίας κανόνων αλληλοσυσχέτισης κλινικών δεδομένων.

Σε πρώτη φάση ο Parser διαβάζοντας ένα XML document, τοποθετεί την πληροφορία σε μια δομή, η οποία έχει τη μορφή *δάσους* (σχήμα 16).Πιο συγκεκριμένα έχουμε μια συνδεδεμένη *αλυσίδα δένδρων*, όπου η ρίζα κάθε δένδρου, αντιστοιχεί σε ένα `Observation` (σχήμα 17). Θα πρέπει να υπενθυμίσουμε στο σημείο αυτό ότι ένα XML αρχείο το οποίο γίνεται parsed, αντιπροσωπεύει και είναι το αποτέλεσμα μιας επερώτησης η οποία έχει γίνει από κοινού στα κατανεμημένα πληροφοριακά συστήματα. Το γεγονός ότι η απάντηση σε μια επερώτηση αποτελείται, εν γένει, από πολλά `Observations` τα οποία ικανοποιούν τις συνθήκες-απαιτήσεις της επερώτησης, δικαιολογεί την επιλογή της προαναφερόμενης δομής δεδομένων.

4.2. Δομές και Αναπαράσταση Δεδομένων

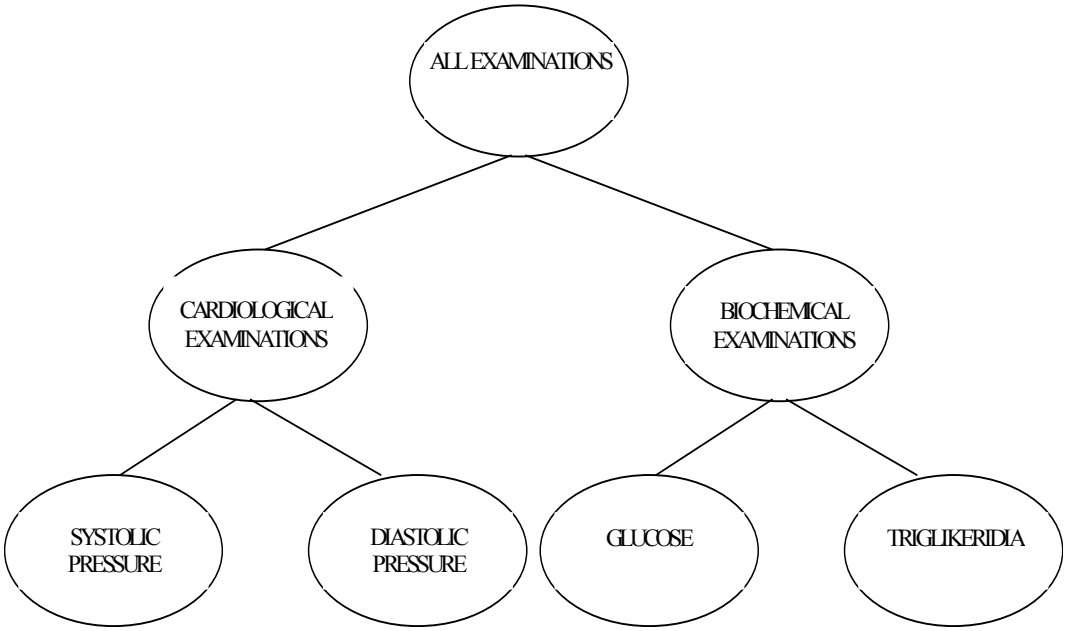
Λαμβάνοντας λοιπόν μια σειρά από `Observations`, γνωρίζουμε ότι ένα `CompositeObservation` αποτελείται εν-γένει από άλλα `Observations`, είτε `Composite`, είτε `Atomic`, γεγονός που παραπέμπει σε δενδρικές δομές. Κάθε *κόμβος* στην δομή (σχήμα 16) αντιστοιχίζεται είτε σε ένα `Atomic Observation` , είτε σε ένα `Composite Observation`. Όταν πρόκειται για `Atomic Observation` ο κόμβος εμφανίζεται ως *φύλλο* στο δένδρο. Στην αντίθετη περίπτωση κάθε `Composite Observation` αποτελεί την αφετηρία ενός νέου

υποδένδρου. Τέλος, το γεγονός ότι ένα `Composite Observation` αποτελείται από έναν οποιοδήποτε αριθμό από επιμέρους `Composite` και `Atomic Observationς`, δικαιολογεί την ασυμμετρία που μπορεί να εμφανιστεί στο δένδρο.



Σχήμα 16. *Η δομή "δάσους" που προκύπτει από το parsing των XML εγγράφων*

Χρησιμοποιώντας πραγματικά δεδομένα παρουσιάζουμε συνοπτικά την μορφή ενός δένδρου (σύνθετη εξέταση) από αυτά που συνθέτουν το τελικό "δάσος", το οποίο αποτελείται από τρία `Composite Observations` και τέσσερα `Atomic Observations` (σχήμα 17).



Σχήμα 17. *Η μορφή ενός απλού "δένδρου" αναπαράστασης κλινικών δεδομένων*

Προκειμένου να έχουμε πιο ευέλικτες δομές, στις οποίες θα μπορέσουμε να εφαρμόσουμε τις τεχνικές ανακάλυψης γνώσεων που θα ακολουθήσουν, από το αρχικό σχήμα αναπαράστασης περνάμε σε δύο νέες δομές, της μορφής που φαίνεται στα σχήματα 21, και 22. Οι δομές αυτές είναι ταχύτερα προσπελάσιμες και έξυπνα ταξινομημένες, έτσι ώστε να είναι εύκολη και γρήγορη η εξόρυξη όλων των επιθυμητών στατιστικών αποτελεσμάτων που επιδιώκουμε. Στη μεταφορά

αυτή κάνοντας κατάλληλο '*φιλτράρισμα'*, διατηρούμε μόνο την αναγκαία προς επεξεργασία πληροφορία, χωρίς να επιβαρύνουμε τις δομές με πληροφορία που δεν πρόκειται να επεξεργαστούμε. Στην δομή δάσους που αναφέραμε κρατείται όλη η πληροφορία που διαβάζεται και αποθηκεύεται στο δημιουργούμενο XML. Γνωρίζοντας ωστόσο την πληροφορία που θέλουμε να εξάγουμε εκ των προτέρων, αποδεσμεύουμε την περιττή πληροφορία, φιλτράροντας τη δομή του δάσους και κρατώντας μόνο την αναγκαία πληροφορία.

Πιο συγκεκριμένα και για να γίνει κατανοητή η δομή και ταξινόμηση που πραγματοποιούμε, αναφέρουμε σε αυτό το σημείο την μορφή της πληροφορίας που θέλουμε να εξάγουμε. Διαχωρίζουμε τα εξαγόμενα στατιστικά συμπεράσματα σε τρεις κατηγορίες.

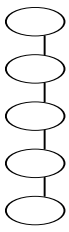
- Εξαγωγή στατιστικών αποτελεσμάτων/συμπερασμάτων ***ανά ασθενή***
- Εξαγωγή στατιστικών αποτελεσμάτων/συμπερασμάτων ***ανά ιατρική εξέταση/ διάγνωση***, η οποία έχει την μορφή του `Atomic Observation`
- Σύντομη και συνοπτική εξαγωγή αποτελεσμάτων/συμπερασμάτων από το ***σύνολο της ανακληθείσας ανά επερώτηση πληροφορίας***

Κινούμενοι λοιπόν σε αυτήν την κατεύθυνση δημιουργούμε δυο δομές, όπου η κάθε μια βοηθά στην γρήγορη εξαγωγή συμπερασμάτων, για την αντίστοιχη από τις δυο πρώτες κατηγορίες που αναφέραμε.

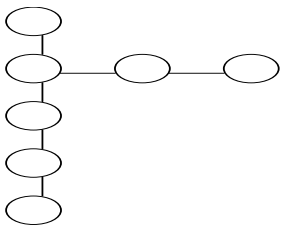
- Η πρώτη δομή (σχήμα 21) είναι ταξινομημένη ως εξής: Η οριζόντια (επάνω μέρος) αναπαριστώμενη αλυσίδα αντιστοιχίζεται στο σύνολο των διαφορετικών παρατηρούμενων (στην εκάστοτε ερώτηση) `Atomic Observations` (μορφή σχήματος 18).

Σχήμα 18. *Οριζόντια αλυσίδα σχημάτων 21 και 22*

Έτσι κάθε κόμβος αντιπροσωπεύει και ένα διαφορετικό `Atomic Observation`. Κάθε κόμβος που συμμετέχει στην οριζόντια αλυσίδα αποτελεί αφετηρία μια νέας κατακόρυφης αλυσίδας (σχήμα 19). Στην αλυσίδα αυτή κάθε κόμβος αντιπροσωπεύει και αντιστοιχίζεται σε ένα διαφορετικό ασθενή, ο οποίος έχει κάνει την συγκεκριμένη εξέταση. Τώρα, κάθε ένας από αυτούς τους κόμβους αποτελεί αφετηρία μιας νέας αλυσίδας (πχ. σχήμα 20 δεύτερος κατακόρυφος κόμβος), όπου εκεί οι κόμβοι αντιπροσωπεύουν διαφορετικές επισκέψεις του ίδιου ασθενή για την συγκεκριμένη εξέταση (ίδιο `Atomic Observation`).

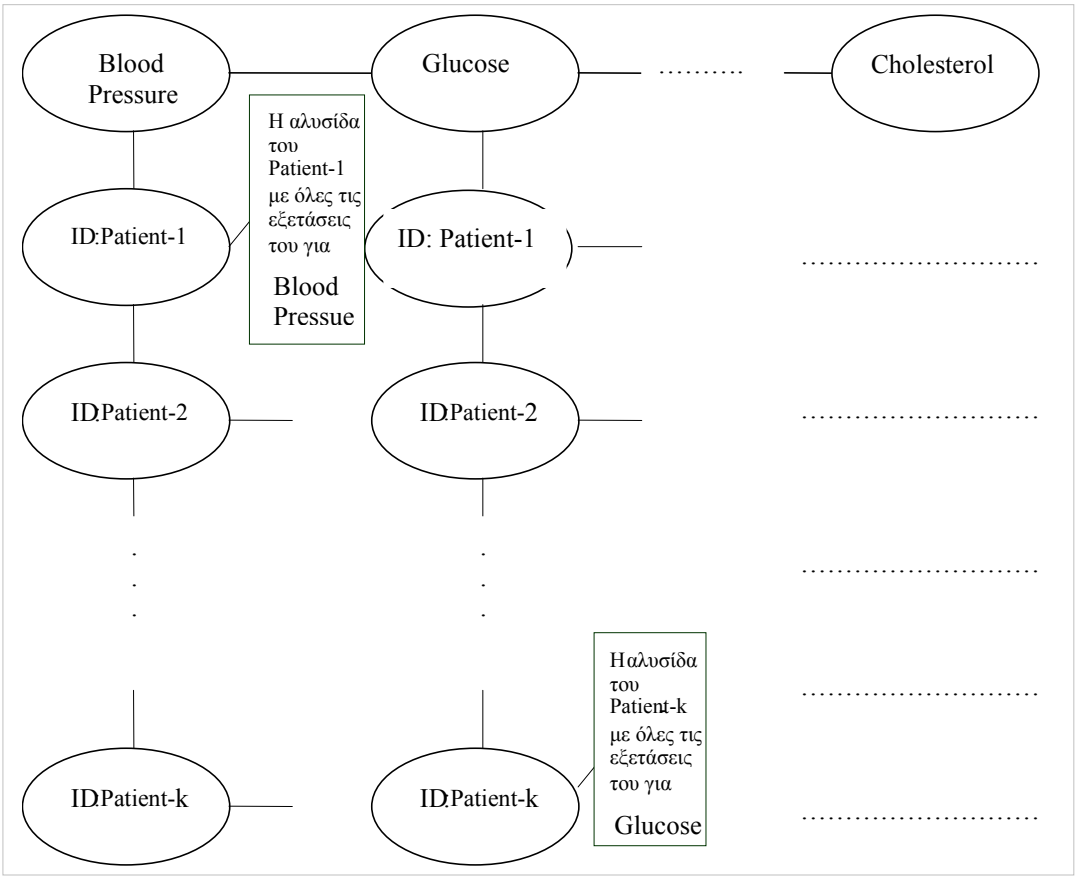


Σχήμα 19. *Κατακόρυφη αλυσίδα σχημάτων 21 και22*



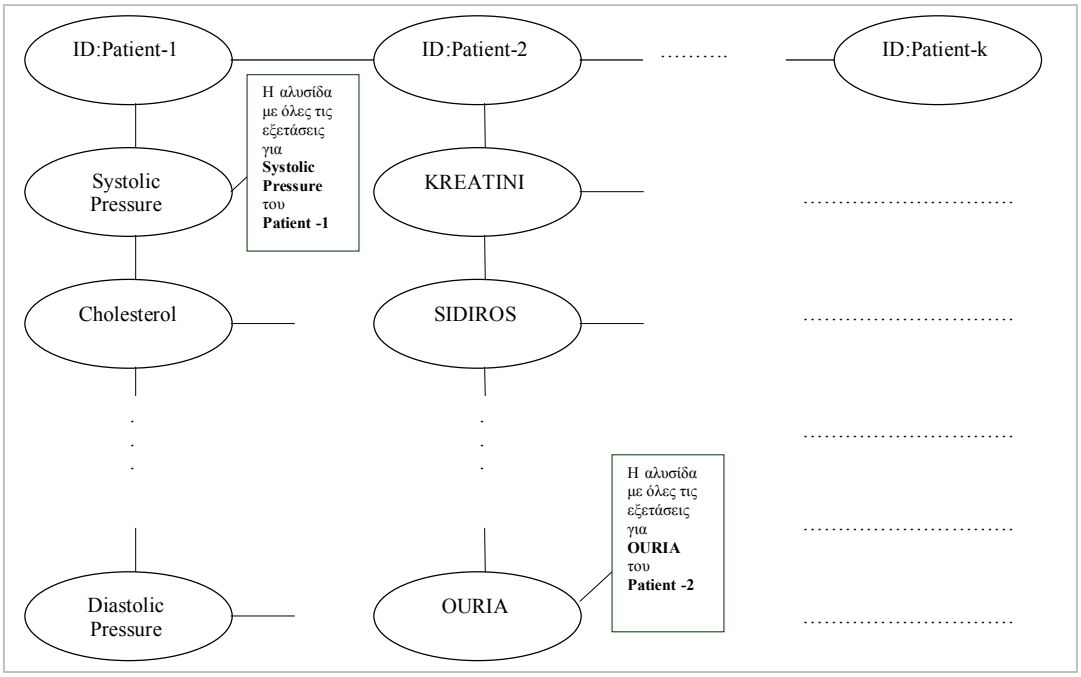
Σχήμα 20. *Αλυσίδα ΔιαφορετικώνΕπισκέψεων*

Στο σχήμα 21 παρουσιάζουμε την μορφή της αναπαράστασης που προκύπτει χρησιμοποιώντας τα όσα αναφέραμε παραπάνω:



Σχήμα 21. *Η δομή που προκύπτει με ταξινόμηση ανά Atomic Observation*

- Με ακριβώς αντίστοιχο τρόπο και χρησιμοποιώντας την μορφή του σχήματος 22, διατηρούμε μια αντίστοιχη δομή, η οποία διαφέρει μόνο στον τρόπο ταξινόμησης των κόμβων. Σ' αυτή τη περίπτωση, η οριζόντια αναπαριστώμενη αλυσίδα (σχήμα 18), αντιστοιχίζεται στο σύνολο των διαφορετικών ασθενών (διαφορετικό `Patient_Id`). Έτσι, κάθε κόμβος αντιπροσωπεύει και ένα διαφορετικό ασθενή.
  Κάθε κόμβος που συμμετέχει στην οριζόντια αλυσίδα, αποτελεί αφετηρία μια νέας κατακόρυφης αλυσίδας (σχήμα 19). Στην αλυσίδα αυτή κάθε κόμβος αντιπροσωπεύει και αντιστοιχίζεται σε μία διαφορετική εξέταση (π.χ `χοληστερίνη`) για κάθε ασθενή. Κάθε ένας από αυτούς τους κόμβους αποτελεί αφετηρία μιας άλλης οριζόντιας αλυσίδας (σχήμα 20), όπου εκεί οι κόμβοι αντιπροσωπεύουν διαφορετικές επισκέψεις για το ίδιο `Atomic Observation` και τον ίδιο ασθενή. Ακολουθεί ένα ενδεικτικό σχηματικό παράδειγμα, στο σχήμα 22.



Σχήμα 22. *Η δομή που προκύπτει με ταξινόμηση ανά Patient-Id*

4.3. Χρήση και Οφέλη του Σχήματος Αναπαράστασης Δεδομένων: Εξαγωγή Επιδημιολογικών Αποτελεσμάτων και Στατιστικών Αποτελεσμάτων

Πλέον είναι εύκολο να διαπιστώσουμε τα πλεονεκτήματα και τα οφέλη που προκύπτουν από μια τέτοια ταξινόμηση των δομών μας, στην διαδικασία εξόρυξης των ζητούμενων στατιστικών αποτελεσμάτων.

4.3.1. Εξαγωγή στατιστικών αποτελεσμάτων/ συμπερασμάτων ανά κλινική εξέταση/ διάγνωση / `Atomic Observation`

Αναλύοντας/ περιγράφοντας την δομή που προκύπτει από την ταξινόμηση, χρησιμοποιώντας τα `AtomicObservations` ως κλειδιά (σχήμα 21), προκύπτει ένας πίνακας της μορφής του σχήματος 23, από όπου εύκολα μπορούμε να εξαγάγουμε στοιχεία της μορφής:

Patient ID	Visit Id	Information System	Date of Observation	Composite Observation	Atomic Observation	Value
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	SACHARO	"112.000000"
"87"	"248"	"spili"	"1997-10-06 11:41:15.291000"	"BIOCHEMICAL_EXAM"	SACHARO	"222.000000"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	SACHARO	"143.000000"
"243"	"139"	"spili"	"1997-06-11 00:00:00.000000"	"BIOCHEMICAL_EXAM"	SACHARO	"83.000000"
"257"	"168"	"spili"	"1997-09-10 10:09:15.227000"	"BIOCHEMICAL_EXAM"	SACHARO	"282.000000"
"277"	"145"	"spili"	"1997-06-03 00:00:00.000000"	"BIOCHEMICAL_EXAM"	SACHARO	"101.000000"

Σχήμα 23. *Ο πίνακας στοιχείων που προκύπτει με ταξινόμηση ανά* `Atomic Observation`

- ✓ Πόσες και ποιες είναι οι διαφορετικές εξετάσεις που έχουν γίνει και καταγραφεί στα

πληροφοριακά μας συστήματα.

- ✓ Κάθε μια σε πόσους και ποιους διαφορετικούς ασθενείς έχει παρατηρηθεί/καταγραφεί
- ✓ Πόσες φορές έχει γίνει στον καθ' ένα από αυτούς;
- ✓ Μπορούμε να δούμε την εξέλιξη(evolution) και το εύρος των τιμών μιας συγκεκριμένης εξέτασης ενός συγκεκριμένου ασθενή, ή μιας ομάδας ασθενών(ανάλογα την επερώτηση που έχει προηγηθεί).
- ✓ Μπορούμε να δούμε σε ποια πληροφοριακά συστήματα(Κέντρα Υγείας) έχει πραγματοποιηθεί και με τι συχνότητα γίνεται η συγκεκριμένη εξέταση στον αντίστοιχο πληθυσμό.

4.3.2. Εξαγωγή στατιστικών αποτελεσμάτων/ συμπερασμάτων ανά ασθενή

Αναλύοντας/ περιγράφοντας την δομή που προκύπτει από την ταξινόμηση χρησιμοποιώντας τα `Patient Id`s (σχήμα 22) ως κλειδιά, προκύπτει ένας πίνακας της μορφής του σχήματος 24, από όπου εύκολα μπορούμε να εξαγάγουμε στοιχεία της μορφής:

Patient Id	Visit Id	Information System	Date Of Observation	Composite Observation	Atomic Observation	Value
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	SACHARO	"112.000000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	OURIA	"25.000000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	KREATINI	"0.770000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	CHOLESTEROL	"355.000000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	HDLCHOLESTEROL	"90.500000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	TRIGLIKERIDIA	"145.000000"
"87"	"248"	"spili"	"1997-10-06 11:41:15.291000"	"BIOCHEMICAL_EXAM"	SACHARO	"222.000000"

Σχήμα 24. *Ο πίνακας στοιχείων που προκύπτει με ταξινόμηση ανά* `Patient-Id`

- ✓ Πόσους και ποιους διαφορετικούς ασθενείς έχουμε καταχωρήσει στα πληροφοριακά μας συστήματα;
- ✓ Καθένας από αυτούς σε πόσες και ποιες διαφορετικές εξετάσεις έχει υποβληθεί;
- ✓ Πόσες φορές έχει πραγματοποιήσει την κάθε μια από αυτές.
- ✓ Πως έχουν μεταβληθεί οι τιμές της κάθε εξέτασης με το χρόνο; Έτσι ο γιατρός μπορεί να διαπιστώσει ίσως, εάν και πόσο αποδοτική ήταν η προτεινόμενη θεραπεία του.
- ✓ Ποιες εξετάσεις και πόσες επισκέψεις συνολικά έχει πραγματοποιήσει στο εκάστοτε πληροφοριακό σύστημα;

Οι δυο πρώτες κατηγορίες που αναφέραμε(Εξαγωγή στατιστικών αποτελεσμάτων/ συμπερασμάτων ανά κλινική εξέταση/ διάγνωση και ανά ασθενή) έχουν ιδιαίτερη σημασία όταν τα ερωτήματα που γίνονται, και κατ' επέκταση τα XML αρχεία που παράγονται, δεν είναι γενικής φύσεως. Για παράδειγμα, αναφέρονται σε ειδικές κατηγορίες ασθενών (π.χ διαβητικοί, καρδιοπαθείς), σε συγκεκριμένες εξετάσεις (Βιοχημικές, Αιματολογικές κ.τ.λ) ή και σε συγκεκριμένους ασθενείς. Όταν οι ερωτήσεις είναι γενικής μορφής, ανακαλούνται δηλαδή σύνθετες εξετάσεις (π.χ `FULL EXAMS`, οι οποίες περιέχουν περισσότερες από μια επιμέρους σύνθετες εξετάσεις) επεξεργαζόμαστε από κοινού

επιμέρους στοιχεία (`Atomic Observations`) πιθανά ασύμβατων ιατρικών εξετάσεων (π.χ μετρήσεις που έγιναν για κάποια `Atomic Observations` και για διαφορετικούς λόγους; δηλαδή διαφορετικά `Composite Observations`). Έτσι η κλινική αξία των παρουσιαζόμενων στοιχείων αφήνεται στην κρίση των ειδικών.

4.3.3. Εξαγωγή αποτελεσμάτων/ συμπερασμάτων από το σύνολο της ανακληθείσας πληροφορίας

Παρουσιάζουμε συνοπτικά τα συμπεράσματα που προκύπτουν από την χρήση και των δυο προαναφερθέντων δομών (μέσω αντίστοιχων links, αναλυτικά κεφάλαιο 10), προκειμένου να ενημερωθεί ο ενδιαφερόμενος, με μια γρήγορη ματιά, για θέματα όπως:

- ✓ πόσες και ποιες είναι οι διαφορετικές εξετάσεις που έχουν γίνει και καταγραφεί στα πληροφοριακά μας συστήματα; Με μια γρήγορη ματιά ο ενδιαφερόμενος μπορεί να παρατηρήσει όλες τις επιμέρους εξετάσεις που απαρτίζουν την ερώτηση που έκανε (σχήμα 25)

SACHARO	OURIA	KREATINI	CHOLESTEROL	HDLCHOLESTEROL
TRIGLIKERIDIA	OURIKO	NA	ALKALIKIFOSFATASI	SIDIROS

Σχήμα 25. *Ο πίνακας που προκύπτει με όλα τα διαφορετικά* `Atomic Observations`

- ✓ πόσους και ποιους διαφορετικούς ασθενείς έχουμε καταχωρήσει στα πληροφοριακά μας συστήματα; Παρατηρώντας έναν πίνακα της μορφής του σχήματος 26, λαμβάνει σύντομα τη ζητούμενη πληροφορία

Patient Id	Information System	Patient Id	Information System	Patient Id	Information System
"64"	"spili"	"87"	"spili"	"243"	"spili"
"257"	"spili"	"277"	"spili"	"336"	"spili"
"420"	"spili"	"428"	"spili"	"466"	"spili"

Σχήμα 26. *Ο πίνακας που προκύπτει με όλους τους διαφορετικούς ασθενείς*

- ✓ τι εξετάσεις έχει κάνει ο καθένας από αυτούς, πόσες φορές κ.τ.λ, με σκοπό την *σύντομη* και όχι τόσο την αναλυτική παρουσίαση των πιο βασικών χαρακτηριστικών (στο σχήμα 27 παρατηρούμε για παράδειγμα πόσες εξετάσεις έχει κάνει ένας συγκεκριμένος ασθενής και πόσες φορές έχει πραγματοποιήσει την κάθε μια από αυτές). Για αναλυτική περιγραφή, καταφεύγουμε στην μια από τις δυο προηγούμενες παρουσιάσεις.

Observation Examined	Number of Times
SACHARO	1
OURIA	1
KREATINI	1
CHOLESTEROL	1
HDLCHOLESTEROL	1
TRIGLIKERIDIA	1

Σχήμα 27. Πίνακας παρουσίασης όλων των εξετάσεων στις οποίες έχει υποβληθεί ένας ασθενής

Σημειώνουμε ότι, σε αυτήν την περίπτωση, διαφορετικοί πληθυσμοί ασθενών έχουν πιθανά εμπλακεί, και διαφορετικές εξετάσεις έχουν γίνει και έχουν επεξεργαστεί από κοινού, όπως βιοχημικές, αιματολογικές, γυναικολογικές κ.τ.λ

Σημείωση-1. Σε μια επόμενη και πιο ολοκληρωμένη έκδοση υλοποίησης του COAS, θα μπορούσαμε να εξάγουμε πρόσθετα συμπεράσματα. Η DTD που έχουμε δημιουργήσει είναι γενική, και ο parser που έχει αναπτυχθεί λαμβάνει υπόψη του πρόσθετες περιπτώσεις. Έτσι, ενώ για την ώρα δεν είναι δυνατή η ανάκληση και *αλληλοσυσχέτιση φαρμάκων* που έχουν δοθεί ως φαρμακευτική αγωγή σε ασθενείς, εάν αυτό καταστεί δυνατό (ο κώδικας στον parser ήδη το επιτρέπει), εύκολα θα μπορούμε να εξάγουμε πληροφορία της μορφής:

- ✓ Για μια διάγνωση πόσα και ποια διαφορετικά φάρμακα έχουν δοθεί;
- ✓ Ποιό/ά είναι το επικρατέστερο/α; κ.τ.λ

Σημείωση-2. Με αυτό τον τρόπο αναπαράστασης της πληροφορίας έχουμε προετοιμάσει, και επεξεργαστεί ήδη, τα απαραίτητα και αναγκαία στοιχεία για την επίτευξη της ανεύρεσης όλων των ενδιαφερόντων κανόνων αλλησυσχέτισης, όπως θα φανεί στο κεφάλαιο 7.

Κεφάλαιο 5:
Τεχνολογία Ανακάλυψης Γνώσεων από Βάσεις Δεδομένων – Η Προτεινόμενη Αρχιτεκτονική

Η τεχνολογία της ***Ανακάλυψης Γνώσεων*** από βάσεις δεδομένων, γνωστή ως **KDD** (*Knowledge Discovery from Data*), εισαγωγικά θα μπορούσαμε να πούμε ότι φέρνει σε επαφή πολλά αυτόνομα ερευνητικά πεδία όπως είναι η *Μηχανική Μάθηση*, η *Αναγνώριση Προτύπων*, η *Στατιστική*, η *Τεχνητή Νοημοσύνη,* η *Ψυχολογία*, η *Επιστήμη των Υπολογιστών* και αρκετά άλλα ακόμη εξίσου σημαντικά ερευνητικά πεδία. Ο κοινός στόχος όλων αυτών των αυτόνομων πεδίων είναι η εξαγωγή νέας και υψηλού επιπέδου γνώσης, από τα κατά περίπτωση και ερευνητικό πεδίο δεδομένα.

Το βασικό επιμέρους διαδικαστικό τμήμα του KDD, γνωστό ως ***Data Mining***, χρησιμοποιεί όπως θα δούμε εργαλεία από την στατιστική, την μηχανική μάθηση και από άλλα πεδία που αναφέραμε, προκειμένου να εξάγει χαρακτηριστικά από τα δεδομένα, τα οποία θα βοηθήσουν στην παραπέρα χρήση τους με σκοπό την ανακάλυψη γνώσης. Το KDD ως συνολική παραγωγική διαδικασία, επικεντρώνεται σε θέματα: αποθήκευσης και πρόσβασης στα δεδομένα, ελάττωσης του μεγέθους τους με ταυτόχρονη απαλλαγή από περιττή πληροφορία, αναγνώρισης βασικών χαρακτηριστικών και ιδιοτήτων τους, παρουσίαση τους στον χρήστη, εύρεση τρόπων επέμβασης και αλληλεπίδρασης του χρήστη με τον υπολογιστή για βέλτιστη εξαγωγή αποτελεσμάτων, μοντελοποίηση αλγορίθμων και μετάφραση αποτελεσμάτων.

Το αυξανόμενο ενδιαφέρον, κυρίως από επιχειρήσεις, για data mining και ανακάλυψη γνώσης από βάσεις δεδομένων, με την ξαφνική διείσδυση πολλών σχετικών εργαλείων στην αγορά, ανέδειξε και επισημοποίησε το σύγχρονο ενδιαφέρον για θέματα που σχετίζονται με αυτό που αποκαλούμε KDD. Ο W. Frawley ορίζει στο [15] το data mining ως την "*μη τετριμμένη και προφανή εξαγωγή, προηγούμενα άγνωστης και πιθανά χρήσιμης πληροφορίας"*. Δυστυχώς τα βασικά χαρακτηριστικά των βάσεων δεδομένων (αναλυτικά παρακάτω), όπως το μέγεθος τους, κάνουν αδύνατη πολλές φορές την εφαρμογή των αναπτυσσόμενων εργαλείων για την εξαγωγή του επιθυμητού και επιδιωκόμενου αποτελέσματος. Επιβάλλεται, λοιπόν να σχεδιασθούν νέοι αλγόριθμοι οι οποίοι θα λάβουν υπόψη τους αυτές τις ιδιαιτερότητες και θα παρέχουν στους χρήστες αποτελεσματικούς και πειστικούς τρόπους για την ανακάλυψη της γνώσης που επιθυμούν και χρειάζονται.

Το πρόβλημα της εξαγωγής ***κανόνων αλλησυσχέτισης*** (*Association Rule Mining*- ARM), είναι ένα από τα πιο σοβαρά προβλήματα στη διαδικασία ανακάλυψης γνώσης, και έχει τύχει ιδιαίτερης προσοχής τα τελευταία χρόνια, γεγονός που επιβεβαιώνεται από το πλήθος των σχετικά πρόσφατων δημοσιεύσεων [1, 2, 3, 5, 52]. Ένα σύνηθες πεδίο εφαρμογής τους είναι μια βάση δεδομένων με συναλλαγές καταναλωτών, σε διάφορες αγορές, προκειμένου να εξαχθούν συμπεράσματα που χαρακτηρίζουν από κοινού τη συμπεριφορά τους. Οι εξαγόμενοι κανόνες μπορεί να είναι αποκαλυπτικοί όπως για παράδειγμα ότι *«το 75% των ανθρώπων που αγοράζουν σάλτσα για μακαρόνια αγοράζουν ζυμαρικά και καπνιστό κρέας»*!. Και ενώ αυτό το παράδειγμα φαίνεται μάλλον διαισθητικά αναμενόμενο, υπάρχουν αρκετές περιπτώσεις όπου οι εξαγόμενοι κανόνες δεν είναι

προφανείς και απαιτείται ένας τρόπος εύρεσης και ανακάλυψης τους. Επιπρόσθετα, εργαλεία εξόρυξης γνώσης και κανόνων βοηθούν τους χρήστες να προσδιορίσουν ποσοτικά τις υποθέσεις τους και να τις ανασκευάσουν εάν χρειαστεί ή να τις επιβεβαιώσουν, διαμέσου των επερωτήσεων που υποβάλλονται στα αντίστοιχα εργαλεία.

Η συσχέτιση αυτών των κανόνων με εφαρμογές όπως σχεδίαση προϊόντων, marketing, διαφήμιση, προώθηση προϊόντων, μπορεί εύκολα να γίνει κατανοητή. Ωστόσο οι ARM αλγόριθμοι μπορούν να εφαρμοστούν σε ένα ευρύτερο πεδίο εφαρμογών και προβλημάτων και να λάβουν χώρα όπως θα δείξουμε ακόμα και σε *ιατρικά πεδία εφαρμογών* και σε ανάλυση και επεξεργασία κλινικών στοιχείων και δεδομένων. Στους κανόνες αλλησυσχέτισης στο ιατρικό πεδίο, θα αναφερθούμε σε μεγαλύτερη λεπτομέρεια στα επόμενα κεφάλαια.

Μια βασική παρατήρηση. Δυστυχώς, ο αριθμός όλων των υποθετικά δυνατών κανόνων αλληλοσυσχέτισης αυξάνει εκθετικά με τον αριθμό των στοιχείων (items) που συμμετέχουν στη διαδικασία. Για 1000 items για παράδειγμα, περισσότεροι από 2^{1000} κανόνες πρέπει να θεωρηθούν και να επεξεργαστούν σε μια απλοϊκή προσέγγιση. Διάφοροι αλγόριθμοι έχουν προταθεί στη διεθνή βιβλιογραφία, όπως ο *Apriori* [4] και ο *PARTITION* [34], για να κάνουν την αναζήτηση αυτή ταχύτερη και πιο έξυπνη. Όλοι αυτοί οι αλγόριθμοι διαφέρουν κυρίως στην *μορφή* αναπαράστασης και τον τρόπο αποθήκευσης των δεδομένων, την ενδιάμεση αναπαράσταση των αποτελεσμάτων κατά την διάρκεια της επεξεργασίας και το κόστος Input/Output και CPU (overhead) που προκαλούν. Χαρακτηριστικό τους είναι ότι τις περισσότερες φορές υπάρχει ανάγκη η αρχική βάση δεδομένων να διαβαστεί περισσότερες από μια φορές, γεγονός που αναπόφευκτα προκαλεί πρόσθετες καθυστερήσεις.

Έτσι, παρά τις συνεχώς αυξανόμενες και εντεινόμενες προσπάθειες για βελτίωση των εν-χρήση σειριακών αλγορίθμων, στην πράξη οι ARM αλγόριθμοι παραμένουν χρονοβόροι και όχι ιδιαίτερα ικανοποιητικοί για άμεση αλληλεπίδραση με τα αντίστοιχα εργαλεία εξόρυξης κανόνων και εκμαίευσης γνώσης στη γενικοτερή της μορφή. Η *δειγματοληψία* (*sampling*) είναι μια λύση του προβλήματος [28], αλλά οπωσδήποτε η επεξεργασία ενός μόνο μέρους από το σύνολο της βάσης δεν μπορεί να δώσει τελείως σωστά και ακριβή αποτελέσματα.

5.1. Ανακάλυψη Γνώσεων από Δεδομένα: Μια Σύντομη Επισκόπηση

Προκειμένου να δώσουμε μια εποπτική εικόνα του πεδίου ανακάλυψης γνώσης από βάσεις δεδομένων, παραθέτουμε κάποιους προβληματισμούς για το *τι είναι γνώση*, ποιες μορφές της ξεχωρίζουμε, *τι είναι εξόρυξη γνώσης* από βάσεις δεδομένων, τι χρειαζόμαστε το KDD, και ποια είναι η διαδικασία περάτωσης του. Συνεχίζουμε με θέματα που αφορούν τη διαδικασία του data mining, την παράθεση μιας λίστας από δυσκολίες που προκύπτουν έχοντας ως πηγές δεδομένων απλές και ενδεικτικές βάσεις δεδομένων (ανεξάρτητα από την μορφή τους), και κάποια θέματα γενικής φύσεως που αφορούν τη συμμετοχή του ίδιου του χρήστη στην όλη διαδικασία. Τέλος κλείνουμε παραθέτοντας την *αρχιτεκτονική* που χρησιμοποιούμε, καθώς και ένα *σενάριο χρήσης* που την υποστηρίζει. Για περισσότερες λεπτομέρειες αναφορικά με το KDD παραπέμπουμε τον αναγνώστη στα [21,40].

5.1.1. Τι είναι γνώση ;

Τα *δεδομένα* μπορούμε να ισχυριστούμε ότι αποτελούν μια *ακατέργαστη μορφή πληροφορίας* την οποία πρέπει να *επεξεργαστούμε* περαιτέρω, πολλές φορές με την

βοήθεια του υπολογιστή. Πληροφορία λοιπόν με την ορθή και ουσιαστική ερμηνεία του όρου, είναι τα δεδομένα τα οποία έχουν *οργανωθεί* με τέτοιο τρόπο (από τον άνθρωπο ή τον υπολογιστή), ώστε να είναι *σημαντικά, αξιόλογα* και *χρήσιμα*. Παραδοσιακές βάσεις δεδομένων αντιπροσωπεύουν απλούς τύπους δεδομένων, όπως αριθμούς, συμβολοσειρές (strings) και λογικές τιμές (Boolean). Η γνώση θα μπορούσαμε να πούμε ότι είναι μια μορφή πληροφορίας, η οποία βρίσκεται ένα *επίπεδο* παραπάνω από αυτήν την *"ωμή και ακατέργαστη"* ροή δεδομένων και είναι κάτι το τελείως διαφορετικό από την απλή μετάφραση αυτών των απλών μορφών πληροφορίας. Οι τρέχουσες εφαρμογές απαιτούν πιο σύνθετες δομές, όπως διαδικασίες, λειτουργίες, χρονικές ακολουθίες, στόχους, κίνητρα κ.τ.λ. Ο όρος γνώση λοιπόν περιγράφει την ευρύτερη κατηγορία της πληροφορίας που απορρέει και συνεπάγεται από όλα τα παραπάνω.

5.1.2. Μορφές – Τύποι Γνώσης

Επαναφέροντας τον προαναφερθέντα ορισμό για το KDD από τον W. Fraweley, η προς ανακάλυψη γνώση είναι *"σιωπηρά εννοούμενη, προηγουμένως άγνωστη και υποθετικά χρήσιμη"*. Με το να είναι *εννοούμενη* η γνώση, ο ορισμός της επεκτείνεται πέρα από την κλασσική προσέγγιση, σύμφωνα με την οποία η παρουσιαζόμενη γνώση είναι άμεσα κατανοητή και απλά κρατείται αποθηκευμένη και διαχειρίζεται επιτυχώς από τα *DBMS* (Data Base Management Systems). Το γεγονός ότι *«ο κ. Χ εργάζεται στην εταιρία Υ και κερδίζει $40.000 ετησίως»*, παρά το πόσο αποκαλυπτικό μπορεί να είναι για το χρήστη, δεν είναι το επιθυμητό αποτέλεσμα για έναν KDD αλγόριθμο.

Το γεγονός αυτό δείχνει ότι αποκαλώντας τη γνώση "*προηγουμένως άγνωστη"*, ο χαρακτηρισμός αναφέρεται τόσο στην προοπτική προσέγγισης από την πλευρά του συστήματος, όσο και στο τρέχων επίπεδο γνώσης του χρήστη. Αυτή η μορφή γνώσης μπορεί να χαρακτηριστεί και ως *μετά- γνώση (*meta-knowledge) και μπορεί να χαρακτηρίζει κρυμμένους νόμους και εμφανίσεις μορφών (structures), οι οποίες δεν χαρακτηρίζονται από ισχυρές συναρτησιακές εξαρτήσεις, αλλά απλά εμφανίζονται με κάποια *πιθανότητα*. Ο Fraweley στο [15], απαιτεί να λαμβάνει χώρα η διαπίστωση ότι: η πληροφορία που συνάγεται από τα εκμαιευμένα χαρακτηριστικά είναι απλούστερη από το υποσύνολο της πληροφορίας που εκμαιεύεται από τα ίδια τα αντικείμενα τα οποία την περιγράφουν. Τώρα, το πώς ερμηνεύεται το απλούστερο σε θέματα γνώσης, αφήνεται ασαφές.

Η τελευταία απαίτηση *"υποθετικά* χρήσιμη*"*, έχει να κάνει με την εκάστοτε εφαρμογή και εξαρτάται από το πόσο και που εστιάζει την προσοχή της η τρέχουσα διαδικασία data mining.

Ο *R. Agrawal* στο [2] καθορίζει τρεις τύπους γνώσης προς ανακάλυψη και έρευνα σε βάσεις δεδομένων: (1) την *ταξινόμηση (*classification*)*, (2) την *αλληλοσυσχέτιση* (associations) και την *ακολουθία περιοδικά επαναλαμβανόμενων γεγονότων* (sequences of frequent events). Η *ταξινόμηση* όπως θα δούμε αναλυτικά παρακάτω προσπαθεί να χωρίσει τα δεδομένα εισόδου σε ξεχωριστές κλάσεις χρησιμοποιώντας τόσο '*supervised'*, όσο και '*unsupervised'* (γνωστό ως clustering) μεθόδους μάθησης [51]. Ο στόχος είναι η εύρεση βασικών εννοιών, με όσο το δυνατόν περισσότερο σαφή διαχωριστικά εννοιολογικά σύνορα τα οποία χαρακτηρίζουν μια κλάση αντικειμένων. Έτσι, για μη προκαθορισμένα αντικείμενα σχετικά με την εννοιολογική τους κατηγοριοποίηση, χάρη στην ταξινόμηση μπορεί να προβλεφθεί η εννοιολογική κατηγορία και κλάση στην οποία ανήκουν. Μια τράπεζα για παράδειγμα μπορεί να επιθυμεί την κατηγοριοποίηση των πελατών της, προκειμένου να αποφασίζει σε ποιους πρέπει και σε ποιους όχι να παρέχεται

δάνειο (credit risk analysis εφαρμογές). Στο [15] οι W. Frawley και G.Piatetski-Shapiro υποδιαιρούν τη παρούσα λειτουργία σε δυο επιμέρους διαδικασίες:

- ✓ *Περίληψη-σύνοψη* (summarization), όπου αναζητούνται κοινά χαρακτηριστικά για μια κλάση μόνο.
- ✓ *Διακριτοποίηση* (discretization), όπου ο στόχος είναι η εύρεση χαρακτηριστικών, τα οποία βοηθούν στο διαχωρισμό διαφορετικών κλάσεων ή εναλλακτικά το διαχωρισμό μιας κλάσης από όλες τις υπόλοιπες.

Όταν ανακαλύπτονται *χρονικές ακολουθίες* (sequences), ο χρόνος, όπως είναι κατανοητό, αποτελεί ένα επιπρόσθετο χαρακτηριστικό. Παραδείγματα τέτοιων εφαρμογών μπορούν να βρεθούν σε αγορές ή σε συμπεριφορές καταναλωτών. Για εξειδικευμένους αλγόριθμους σε ανακάλυψη ακολουθιών που χρησιμοποιούν το διακριτό μετασχηματισμό Fourier, παραπέμπουμε τον χρήστη στο [33]. Ένα ενδιαφέρον πρόβλημα που αφορά χρονικές ακολουθίες είναι η ανακάλυψη *επεισοδίων* (episodes), δηλαδή συχνά εμφανιζόμενων γεγονότων σε ένα δοθέν διάστημα χρόνου (time window) [29]. Οι αλγόριθμοι για εύρεση επεισοδίων μοιάζουν κατά πολύ με αυτούς που χρησιμοποιούνται για ARM, τους οποίους και θα παρουσιάσουμε στη συνέχεια.

Η τρίτη κατηγορία γνώσης είναι οι ***κανόνες αλλησυσχέτισης***. Οι συσχετίσεις μπορεί να είναι αυθαίρετοι κανόνες της μορφής *"X =>Y"*. Στην κατηγορία αυτή θα αναφερθούμε εκτενώς παρακάτω.

5.1.3. Τι είναι εξαγωγή γνώσης από βάσεις δεδομένων;

Σε ένα αφηρημένο επίπεδο η ανακάλυψη γνώσης από βάσεις δεδομένων(KDD) σχετίζεται με την ανακάλυψη και εύρεση μεθόδων και τεχνικών ώστε να δημιουργείται και να εξάγεται *νόημα* από τα δεδομένα. Το KDD είναι ιδιαίτερα χρήσιμο σε περιπτώσεις όπου τα χαμηλού επιπέδου δεδομένα είναι δύσκολο να κατανοηθούν ή και να μεταφραστούν, λόγω είτε του τεράστιου όγκου τους, είτε της αυξημένης πολυπλοκότητας τους. Εάν τα δεδομένα εξάγονται από ένα ιδιαίτερα σύνθετο πεδίο, η διαδικασία του KDD συνήθως λαμβάνει χώρα και εκτελείται σε μικρού μεγέθους σύνολα δεδομένων, ανάλογα με την πολυπλοκότητα της λειτουργίας/ διαδικασίας που δημιούργησε τα δεδομένα. Στον πυρήνα και στο επίκεντρο της διαδικασίας του KDD, βρίσκεται η εφαρμογή εξειδικευμένων μεθόδων data mining για την ανακάλυψη και εξαγωγή συμπερασμάτων, προτύπων και αλληλοσυσχετίσεων.

5.1.4. Γιατί χρειαζόμαστε τις διαδικασίες ανακάλυψης γνώσεων (KDD)

Ο Fayyad το 1996 διαπίστωσε ότι *ο παραδοσιακός τρόπος μετατροπής δεδομένων σε πραγματική γνώση βασίζεται στην χειρονακτική ανάλυση και επεξεργασία τους*. Η προσέγγιση αυτή είναι γνωστή σε συμπερασματικές βάσεις δεδομένων όπου οι κανόνες εξάγονται και μαθαίνονται από ειδικούς μελετητές (Zeleznikow και Hunter, 1994: κεφ. 8). Η κλασσική προσέγγιση στην επεξεργασία δεδομένων βασίζεται ουσιαστικά σε έναν ή περισσότερους ειδικούς αναλυτές, οι οποίοι αποκτούν στενή σχέση με τα δεδομένα και χρησιμεύουν ως ένα είδος *'διεπαφής'* μεταξύ των δεδομένων, των χρηστών και των προϊόντων. Ωστόσο γίνεται εύκολα κατανοητό ότι η χειρονακτική επεξεργασία δεδομένων είναι ιδιαίτερα αργή, ακριβή και υποκειμενική. Με τα μεγέθη μάλιστα των δεδομένων να αυξάνονται με δραματικούς ρυθμούς, η χειρονακτική επεξεργασία καθίσταται αδύνατη. Τα μεγέθη των βάσεων δεδομένων αυξάνονται για δυο λόγους. Πρώτον, αυξάνεται ο αριθμός των εγγραφών, *N*, ή των αντικειμένων, και δεύτερον αυξάνεται ο αριθμός, *d*, των πεδίων ή των χαρακτηριστικών τους αντίστοιχα. Στο πεδίο της Αστρονομίας για

παράδειγμα, βάσεις δεδομένων με μεγέθη της τάξεως των $N = 10^9$ αντικειμένων είναι πολύ συνηθισμένες. Ανάλογα και στο χώρο της ιατρικής, υπάρχουν εφαρμογές ιατρικών διαγνώσεων όπου στις αντίστοιχες βάσεις υπάρχουν μεγέθη πεδίων της τάξεως των $d = 10^3$. Βιολογικές βάσεις δεδομένων είναι ακόμα πιο πολύπλοκες καθώς μπορεί να σχετίζονται δεδομένα από ετερογενείς και γεωγραφικά κατανεμημένες βάσεις δεδομένων. Όταν λοιπόν έχουμε να κάνουμε με επεξεργασία εκατομμυρίων εγγραφών, με δεκάδες ή και εκατοντάδες χιλιάδες πεδίων, η αυτόματη επεξεργασία τους κρίνεται κάτι παραπάνω από αναγκαία.

5.1.5. Η διαδικασία του KDD

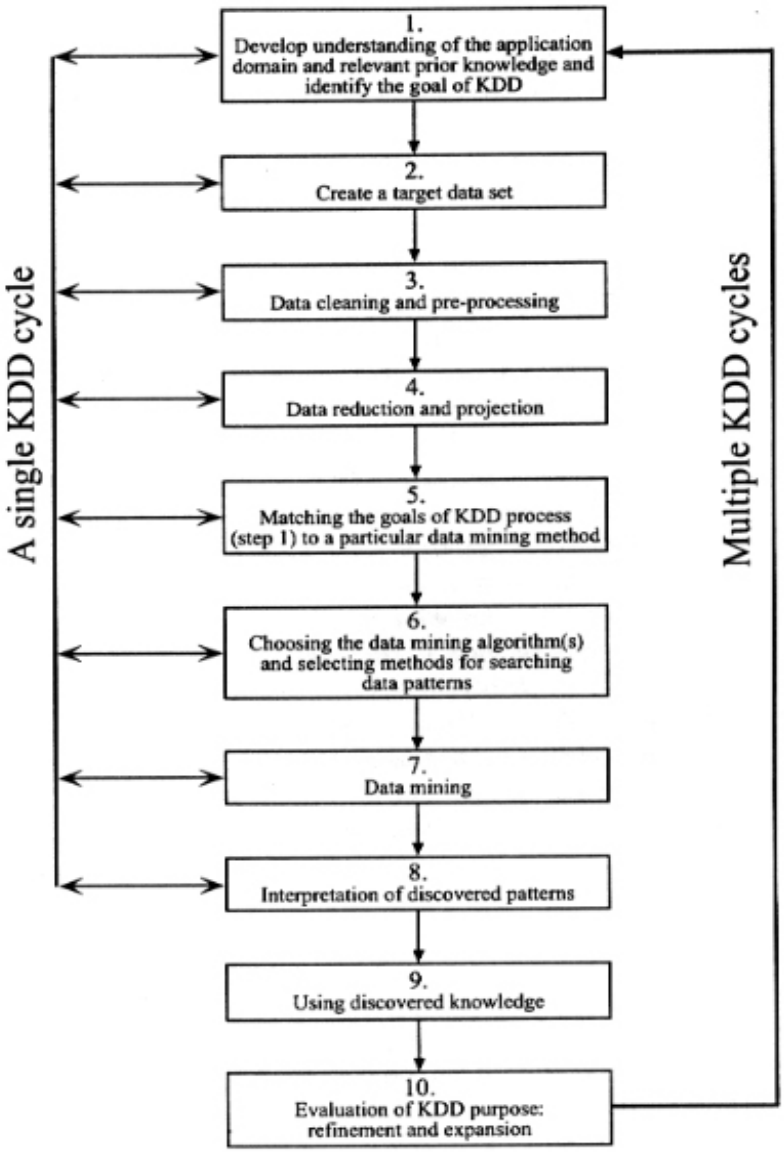
Η διαδικασία του KDD περιλαμβάνει δέκα βήματα [50].

---

1. *Μάθηση* και *κατανόηση* του πεδίου εφαρμογής. Αυτό περιλαμβάνει και υποδηλώνει την μελέτη, εκμάθηση και ανάπτυξη σχετικής με το θέμα προηγούμενης γνώσης του χρήστη. Τον ξεκάθαρο προσδιορισμό στόχου και αρχικού σκοπού της διεργασίας του KDD, έτσι όπως αυτός διαμορφώνεται και προσδιορίζεται από την πλευρά του χρήστη.
2. Την *δημιουργία* ενός συνόλου *δεδομένων* (data set), το οποίο περιλαμβάνει ένα σύνολο από επιλεγμένα δεδομένα και μεταβλητές πάνω στα οποία θα επικεντρωθεί η διαδικασία της ανακάλυψης και εξόρυξης.
3. *Προ-επεξεργασία* (pre-processing) και "καθαρισμός" των δεδομένων. Στο βήμα αυτό συμπεριλαμβάνονται λειτουργίες (όπως θα δείξουμε σε ειδική ενότητα) όπως απαλλαγή της πληροφορίας από θόρυβο, συγκέντρωση της αναγκαίας και χρήσιμης πληροφορίας και αποφάσεις διαχείρισης και αντιμετώπισης περιπτώσεων όπου λείπουν κάποιες τιμές πεδίων.
4. *Ελάττωση* και *προβολή δεδομένων* (data reduction and projection). Η λειτουργία αυτή περιλαμβάνει την εύρεση χρήσιμων χαρακτηριστικών για την αναπαράσταση των δεδομένων. Με την ελάττωση των διαστάσεων ή με μεθόδους μετασχηματισμού, ο αναγκαίος αριθμός μεταβλητών που πρέπει να ληφθούν υπόψη για την αναπαράσταση των δεδομένων μπορεί να ελαττωθεί.
5. Επιλογή της *μεθόδου* που θα χρησιμοποιηθεί για την εκπόνηση του data mining. Η επιλογή έχει να κάνει με την απόφαση του χρήστη για το εάν θα χρησιμοποιηθεί ταξινόμηση, ομαδοποίηση, σύνοψη ή κάποιο άλλο μοντέλο.
6. Επιλογή των *αλγορίθμων* που θα επιτελέσουν το data mining. Η λειτουργία αυτή είναι υπεύθυνη για την επιλογή των μεθόδων που θα χρησιμοποιηθούν για να αναζητήσουν *"πρότυπες αλληλοσυσχετίσεις"* (patterns) ανάμεσα στα δεδομένα και θα συνδυάσουν/ ταιριάξουν μια συγκεκριμένη μέθοδο εξόρυξης, με το γενικότερο πνεύμα της διαδικασίας του KDD.
7. *Data mining* – Περιλαμβάνει την αναζήτηση χαρακτηριστικών του ενδιαφέροντος του χρήστη, είτε σε μια ειδική μορφή αναπαράστασης της πληροφορίας, είτε σε ένα σύνολο από τέτοιες αναπαραστάσεις της, συμπεριλαμβάνοντας την ταξινόμηση κανόνων / δένδρων απόφασης, ομαδοποίηση κ.τ.λ.
8. *Επεξήγηση-ερμηνεία* (interpretation). Το στάδιο αυτό περιλαμβάνει την πιθανή επανάληψη των βημάτων 1-7. Επίσης καλύπτει θέματα όπως παρουσίαση των εξαγόμενων χαρακτηριστικών και μοντέλων, αλλά και την παρουσίαση(visualisation) των ίδιων των δεδομένων εισόδου, από τα οποία προκλήθηκαν τα εξαγόμενα αποτελέσματα.
9. *Χρήση* της παραγόμενης *γνώσης*. Το βήμα αυτό περιλαμβάνει απευθείας επαφή

και χρήση της προκύπτουσας γνώσης. Αυτό μπορεί να σημαίνει ότι η εξαγόμενη γνώση μπορεί να γίνει είσοδος σε κάποιο άλλο σύστημα για περαιτέρω επεξεργασία (π.χ., σε διαδικασίες multi-strategy learning). Ανάλογα μπορεί να σημαίνει τη μορφοποίηση, δόμηση και παρουσίαση της ληφθείσας γνώσης στο χρήστη με κάποιο έξυπνο και όμορφο τρόπο.

10. *Αξιολόγηση* του σκοπού που συντελέστηκε το KDD. Η νέα εκμαιευμένη γνώση συχνά χρησιμοποιείται για να τυποποιήσει και να επισημοποιήσει υποθέσεις. Επίσης νέες ερωτήσεις μπορούν να "*γεννηθούν*", χρησιμοποιώντας την επεκταμένη και διευρυμένη πλέον γνώση μας πάνω στο αντικείμενο που μελετάμε. Στο σημείο αυτό μπορούμε να επιστρέψουμε σε κάποιο προηγούμενο βήμα, εάν αυτό κριθεί αναγκαίο και να το βελτιστοποιήσουμε προσδοκώντας καλύτερα αποτελέσματα. Η σχηματική αναπαράσταση των παραπάνω όπου φαίνονται όλα τα βήματα της KDD διεργασίας, καθώς και η αλληλουχία τους φαίνεται στο σχήμα 28.



Σχήμα 28. *Τα 10 βήματα της διαδικασίας του KDD[50]*

5.2. Τι είναι το Data Mining

Το *data mining* είναι μια μεθοδολογία επίλυσης προβλημάτων, η οποία στόχο έχει να βρει μια τυπική και επίσημη περιγραφή χαρακτηριστικών κάποιων αντικειμένων, η οποία προκύπτει από ένα σύνολο δεδομένων εν γένει σύνθετης και πολύπλοκης φύσης. Οι Decker και Focardi θεωρούν ότι υπάρχουν διάφορα πεδία αντιπροσωπευτικά για να εφαρμοστούν σε αυτά οι μεθοδολογίες του data mining, και αναφέρουν ενδεικτικά την *ιατρική* και τις *επιχειρήσεις* ως δυο τέτοια πεδία. Ισχυρίζονται ότι, σε πρακτικές εφαρμογές, το data mining βασίζεται σε δυο υποθέσεις.

1. *Πρώτον,* ότι οι λειτουργίες που κάποιος θέλει να επιτελέσει και να γενικεύσει, μπορούν να προσεγγιστούν από κάποια *απλά υπολογιστικά μοντέλα*, σε κάποιο βασικό επίπεδο ακρίβειας.
2. *Δεύτερον,* τα *δειγματοληπτικά* δεδομένα περιέχουν σε ικανοποιητικό βαθμό την απαιτούμενη πληροφορία για να επιτελεστεί η προσδοκούσα γενίκευση. Ο Fayyad θεωρεί το data mining ως την εφαρμογή ειδικών αλγορίθμων για την εξαγωγή ομοιοτήτων και συσχετίσεων από το σύνολο των δεδομένων. Τα πρόσθετα βήματα της διαδικασίας του KDD υπάρχουν βασικά για να σιγουρεύουν και να εγγυώνται, όσο το δυνατόν περισσότερο, ότι η πληροφορία που εξάγεται από τα δεδομένα είναι και χρήσιμη. "*Τυφλή*" εφαρμογή του data mining γνωστή ως "*data dredging*", μπορεί εύκολα να οδηγήσει σε παραπλανητική και χωρίς νόημα πληροφορία!

5.2.1. Το data mining στην διαδικασία του KDD

Το τμήμα του data mining, μέρος της συνολικής διαδικασίας του KDD όπως αναφέραμε, συνήθως περιλαμβάνει την επαναλαμβανόμενη εφαρμογή ειδικών και εξειδικευμένων μεθόδων και λειτουργιών. Περιλαμβάνει το *αρμονικό ταίριασμα* ποικίλλων μοντέλων, προκειμένου να παρατηρηθούν αναλυτικά και προσεκτικά τα δεδομένα, ώστε να εκμαιευτούν διάφοροι *τύποι περιγραφών* των χαρακτηριστικών τους. Αυτό το ταίριασμα των μοντέλων σε τελική ανάλυση περιγράφει και την παραγόμενη και συναγόμενη γνώση. Πολύ συχνά απαιτείται η *ανθρώπινη κρίση* για να αποφασιστεί εάν τα μοντέλα δείχνουν και μπορούν να παράγουν χρήσιμη και ενδιαφέρουσα γνώση. Δυο μαθηματικοί φορμαλισμοί χρησιμοποιούνται για την αναπαράσταση των μοντέλων, η στατιστική και η λογική. Ένα μη ντετερμινιστικό μοντέλο υιοθετείται στην στατιστική προσέγγιση, ενώ η χρήση λογικής συνεπάγεται την χρήση ενός καθαρά ντετερμινιστικού μοντέλου. Η στατιστική προσέγγιση στο χώρο του data mining είναι περισσότερο διαδεδομένη κυρίως για πρακτικές εφαρμογές, καθώς τα πραγματικά δεδομένα είναι συνήθως συσχετισμένα και ταυτισμένα με έναν σημαντικό βαθμό αβεβαιότητας. Οι περισσότερες λειτουργίες του data mining είναι βασισμένες σε καλά τεκμηριωμένες και αναπτυγμένες τεχνικές από το χώρο και το πεδίο της Μηχανικής Μάθησης, της Αναγνώρισης Προτύπων και της Στατιστικής (όπως ομαδοποίηση, ταξινόμηση κ.τ.λ).

Στο σημείο αυτό θα περιγράψουμε δυο βασικούς και πρακτικούς στόχους του data mining: την *πρόγνωση/ πρόβλεψη* (*prediction*) και την *περιγραφή* (*description*). Αυτοί οι στόχοι μπορούν να επιτευχθούν με την χρήση διαφόρων γενικών μεθόδων, όπως θα περιγράψουμε παρακάτω.

Η *περιγραφή* επικεντρώνεται στην εύρεση ερμηνεύσιμων χαρακτηριστικών τα οποία είτε καθορίζουν ποσοτικά τα υπάρχοντα δεδομένα, είτε ανακαλύπτουν βασικές ιδιότητες ανάμεσα στα δεδομένα. Η *πρόγνωση* αναφέρεται στην

αντιστοίχηση μιας τιμής με μια μεταβλητή του ενδιαφέροντος μας, για μια μελλοντικά παρουσιαζόμενη εμφάνισή της. Αν και τα όρια ανάμεσα σε αυτές τις δυο λειτουργίες δεν είναι ξεκάθαρα και ακριβή, ο διαχωρισμός τους είναι αρκετά βοηθητικός στην κατανόηση του συνολικού στόχου και σκοπού της ανακάλυψης γνώσης. Οι στόχοι τους μπορούν να υλοποιηθούν χρησιμοποιώντας μια ποικιλία από μεθόδους, ειδικές για data mining, όπως είναι η ομαδοποίηση, η ταξινόμηση, η σύνοψη και η αλληλεξάρτηση μοντέλων.

Η *ταξινόμηση* (classification) είναι μια τεχνική μάθησης, η οποία αντιστοιχίζει ένα αντικείμενο εισόδου σε μια ή περισσότερες προκαθορισμένες ομάδες (classes).

Η *ομαδοποίηση* (clustering) χρησιμοποιείται για να καθοριστούν επακριβώς, είτε σαφή και ακριβή, είτε επικαλυπτόμενα υποσύνολα ανάμεσα στα δεδομένα, γεγονός που οδηγεί προφανώς σε βέλτιστη περιγραφή. Η *παλινδρόμηση* (regression) έχει να κάνει με την αντιστοίχηση ενός δεδομένου (data item), σε μια πραγματικά μετρούμενη μεταβλητή. Η *σύνοψη* αποτελείται από διάφορες μεθόδους με σκοπό την ανακάλυψη ενιαίων και "συμπυκνωμένων" χαρακτηριστικών των δεδομένων.

Η *μίξη-μοντέλων* έχει να κάνει με την αναζήτηση ενός μοντέλου ή μιας περιγραφής, έτσι ώστε να εξηγούνται ικανοποιητικά οι σχέσεις ανάμεσα στις διάφορες μεταβλητές (π.χ μοντελοποίηση του συνόλου των ανθρωπίνων χρωμοσωμάτων).

5.2.2. Συνοπτική αναφορά σε εργαλεία και τεχνικές Data mining

Περιληπτικά και συνοπτικά θα αναφέρουμε μερικές δημοφιλείς τεχνικές mining, όπως είναι (α) τα δένδρα απόφασης (decision trees and rules), (β) οι μέθοδοι ομαδοποίησης (linear regression and classification methods), (γ) οι μέθοδοι βασισμένοι σε ομοιότητες (similarity-based methods), (δ) τα πιθανοκρατικά μοντέλα (probabilistic models) και (ε) τα σχεσιακά μοντέλα εκμάθησης (relational learning models). Η κατανόηση τους σκοπό έχει να βοηθήσει το χρήστη στην βέλτιστη επιλογή μοντέλου ανά πρόβλημα και περίσταση.

- Τα *δένδρα απόφασης* αποτελούνται από κόμβους και ακμές. Κάθε κόμβος περιέχει ένα τεστ από κάποια χαρακτηριστικά των δεδομένων. Τα δένδρα απόφασης παράγουν ομαδοποιήσεις οι οποίες γίνονται εύκολα κατανοητές και παράγουν σύντομα περιεκτικά μοντέλα. Ωστόσο ο περιορισμός σε ένα συγκεκριμένο δένδρο μπορεί να περιορίσει τη λειτουργικότητα του μοντέλου. Τα δένδρα απόφασης και οι αντίστοιχα παραγόμενοι κανόνες συνήθως χρησιμοποιούνται για λειτουργίες πρόγνωσης, ομαδοποίησης και σύνοψης.
- Οι βασισμένες σε μετρικές *ομοιότητας* μέθοδοι χρησιμοποιούν αντιπροσωπευτικά παραδείγματα για να πιστοποιήσουν και να αποδείξουν την ισχύ ενός μοντέλου. Οι ιδιότητες και τα χαρακτηριστικά νέων παραδειγμάτων προβλέπονται και επιβεβαιώνονται, από τα προηγουμένως γνωστά χαρακτηριστικά γνωστών παραδειγμάτων. Αυτή η μέθοδος έχει αποδειχτεί ιδιαίτερα χρήσιμη στο πεδίο της βιολογίας.
- Οι βασισμένες στο νόμο του *Bayes* μέθοδοι παρέχουν ένα φορμαλισμό για την υποψία ισχύος κάποιων θεωρήσεων υπό συνθήκες. Οι μέθοδοι στηρίζονται βασικά στο θεώρημα και τον τύπο του Bayes, ο οποίος είναι θεμελιώδης στην θεωρία των πιθανοτήτων και χρησιμοποιεί δεσμευμένες πιθανότητες. Καθώς είναι ιδιαίτερα σημαντικό να επεξεργαζόμαστε αβέβαια γεγονότα και ενδεχόμενα, οι μέθοδοι αυτές είναι αρκετά χρήσιμες.
- Τα *σχεσιακά μοντέλα εκμάθησης* (Relational Learning; Συμπεριλαμβανομένων των συστημάτων Επαγωγικής Λογικής- Inductive Logic Programming – ILP)

συνδυάζουν την λογική πρώτης τάξης με μεθόδους αυτόματου προγραμματισμού και μηχανικής μάθησης. Τα σχεσιακά μοντέλα μπορεί να έχουν μεγάλη ισχύ στον τομέα της αναπαράστασης της πληροφορίας, ωστόσο το γεγονός αυτό αντισταθμίζεται από το σημαντικά αυξημένο κόστος αναζήτησης λύσεων.

5.3 Ειδικά Θέματα Για Βάσεις Δεδομένων

Θα μπορούσε να υποστηρίξει κάποιος ότι πολλές από τις προσεγγίσεις πού γίνονται σε βασικά θέματα του data mining δεν διαφέρουν δραματικά από τα πρότυπα προβλήματα της Μηχανικής Μάθησης. Ωστόσο, το γεγονός ότι χρησιμοποιείται μια βάση δεδομένων -ανεξάρτητα από την μορφή της- ως πηγή δεδομένων, δημιουργεί πρόσθετες δυσκολίες.

5.3.1. Όγκος Δεδομένων

Ενώ στην Μηχανική Μάθηση το μέγεθος των προς επεξεργασία δεδομένων ανά περίπτωση και πρόβλημα, σπάνια ξεπερνά τις μερικές χιλιάδες στοιχεία, οι σύγχρονες βάσεις δεδομένων στις περισσότερες περιπτώσεις ξεπερνούν τις εκατοντάδες χιλιάδες ή και τα εκατομμύρια διαφορετικών καταγραφών στοιχείων/ δεδομένων, με συνεχώς αυξανόμενη τάση. Το γεγονός αυτό δημιουργεί τεράστιο πρόβλημα στη διαχείριση τόσο των ίδιων δεδομένων, αλλά πολύ περισσότερο και των ενδιάμεσων αποτελεσμάτων που προκύπτουν κατά την επεξεργασία τους. Επιπρόσθετα, οι περισσότεροι από τους πρόσφατα προτεινόμενους σχετικούς αλγόριθμους και τεχνικές αντιμετωπίζουν μια βάση δεδομένων σαν ένα *καθολικό πίνακα* (universal relation). Αυτή η υπόθεση επιβαρύνει ακόμη περισσότερο το δημιουργούμενο πρόβλημα λόγο μεγέθους των δεδομένων. Επειδή οι βάσεις οι οποίες έχουν διαχωριστεί σε επιμέρους υπό-πίνακες υφίστανται την εφαρμογή κάποιας κανονικής μορφής με σκοπό την ελάττωση του χώρου αποθήκευσης τους, μέσω την ένωσης τους (join) δημιουργούνται ακόμη μεγαλύτεροι πίνακες, κάνοντας το πρόβλημα ακόμη μεγαλύτερο και μάλλον δισεπίλυτο!

5.3.2. Θόρυβος, Ελλιπή Και Αντιφατικά Δεδομένα

Είναι γεγονός ότι οι βάσεις δεδομένων δεν δημιουργούνται και πολύ περισσότερο δεν συντηρούνται για να εξυπηρετήσουν τις προσδοκίες και τους σκοπούς του data mining. Τα δεδομένα τους σκοπό έχουν να εξυπηρετήσουν το λόγο ύπαρξης της εκάστοτε εφαρμογής και όχι να διευκολύνουν τις εργασίες που αφορούν τη δυνατότητα μετέπειτα επεξεργασίας τους. Σε περιπτώσεις όπου κάποια ουσιώδη χαρακτηριστικά για τη διαδικασία της ανακάλυψης λείπουν, είναι δυνατόν να προκύψουν άσχημα ή ακόμα και λάθος αποτελέσματα. Μεγάλο είναι το δίλημμα που προκύπτει όταν πρέπει να ληφθούν αποφάσεις για τον χειρισμό *'NULL'* (*άγνωστων*, μη καταχωρημένων) τιμών, είτε ακαθόριστων για κάποιο λόγο τιμών. Οι εναλλακτικές λύσεις είναι, είτε να τις *αντικαταστήσουμε* με κάποιες εξορισμού τιμές που καθορίζονται από κάποιες πιθανότητες, βασισμένες στις ήδη υπάρχουσες και διαθέσιμες τιμές, είτε να *αδιαφορήσουμε* τελείως γι' αυτές. Τόσο στη μια όσο και στην άλλη περίπτωση υπάρχει ο κίνδυνος να οδηγηθούμε σε λανθασμένα αποτελέσματα.

Τα πραγματικά δεδομένα πολλές φορές είναι εμπλουτισμένα με *θόρυβο*, ή περιέχουν *αντιφατικές* πληροφορίες που οφείλονται είτε σε λανθασμένες καταχωρήσεις/ εισαγωγές (data entry), είτε στην ίδια την φύση των δεδομένων. Κάτι τέτοιο όμως δεν είναι επιθυμητό και δεν αποτελεί την καλύτερη δυνατή είσοδο για τους πρότυπους αλγόριθμους μάθησης. Έτσι πολλές φορές χρειάζονται

πιθανοκρατικές λύσεις για την αντιμετώπιση τέτοιων δυσκολιών.

Ανάλογο πρόβλημα στην περίπτωση μας αποτελεί η αντιμετώπιση περιπτώσεων στις οποίες, για κάποιο λόγο, δεν καθορίζεται η τιμή (`Observation Value`) μίας εξέτασης (`Atomic Observation`). Ο αλγόριθμος θα πρέπει να αποφασίσει σε ποιο διάστημα τιμών ανήκει ή μέτρηση-ένδειξη. Οι εναλλακτικές και πάλι προσεγγίσεις είναι δυο. *Είτε αγνοούμε* αυτές τις περιπτώσεις, είτε *αποφασίζουμε αυθαίρετα* τον τρόπο αντίδρασης του αλγορίθμου. Σ' αυτήν τη περίπτωση θεωρούμε απλά ότι οι μετρήσεις αυτές ανήκουν στα *φυσιολογικά* όρια τιμών για τις αντίστοιχες εξετάσεις.

5.3.3. Περιττή Πληροφορία

Ενώ οι αλγόριθμοι αναζήτησης υποτίθεται ότι παρατηρούν και διακρίνουν σχέσεις σε μια βάση δεδομένων, δεν είναι επιθυμητό αυτή η ανακάλυψη να σχετίζεται με *προηγούμενα γνωστή* και κατοχυρωμένη γνώση, καθότι δεν προσφέρει ουσιαστική βοήθεια. Στο [30] παρουσιάζονται δυο περιπτώσεις, στις οποίες ασήμαντες και τετριμμένες σχέσεις αναφέρονται ως εξαγωγή γνώσης. Η πρώτη περίπτωση έχει να κάνει με *ισχυρά εξαρτημένες σχέσεις*, όπως είναι η περίπτωση όπου ένα πεδίο είναι μια συνάρτηση ενός ή περισσοτέρων άλλων πεδίων (π.χ `Profit = Sales – Expenses`). Η άλλη περίπτωση είναι όταν οι τιμές ενός πεδίου *περιορίζονται* από κάποια άλλα πεδία. Τέλος, μια άλλη και δυστυχώς συχνά παρουσιαζόμενη περίπτωση σχετίζεται με την *εννοιολογική ιεράρχηση*. Όποτε γίνεται μια αναφορά σε μια υψηλού επιπέδου ιεραρχική έννοια, με την αποθήκευση κάποιου αντικειμένου, αυτό αμέσως συνεπάγεται την δημιουργία από τον αλγόριθμο της αντίστοιχης σχέσης, αν και αυτή η γνώση δεν είναι καθόλου του ενδιαφέροντος του χρήστη. Ένα τέτοιο παράδειγμα είναι οι σχέσεις `City ⇒ State, Department ⇒ Division` για μεγάλες εταιρίες, ή ότι `AIDS ⇒ ViralInfection`. Αυτές οι δυσκολίες υποδεικνύουν ότι η διαδικασία εξόρυξης πρέπει να κατευθύνεται και να καθοδηγείται τόσο από προηγούμενη γνώση, όσο και από εξειδικευμένη πάνω στο πεδίο της έρευνας γνώση.

5.4. Γενικά Θέματα

5.4.1. Οι Χρήστες στην Διαδικασία Εξόρυξης Γνώσης

Τα εργαλεία και οι εφαρμογές ανακάλυψης γνώσης δεν είναι, και ίσως ούτε ποτέ θα μπορέσουν να είναι σε θέση να καθορίσουν εάν η γνώση που εκμαιεύεται είναι πραγματικά του ενδιαφέροντος του χρήστη. Δεν είναι λίγες οι φορές όπου τα συστήματα αυτά "*τροφοδοτούν*" το χρήστη με προφανή και χωρίς κανένα ουσιαστικό νόημα πληροφορία. Δεν αποτελεί έκπληξη για παράδειγμα η εξαγωγή της πληροφορίας «*όλες οι έγκυες ασθενείς είναι θηλυκού γένους*». Κρίνεται λοιπόν αναγκαία και επιθυμητή η αλληλεπίδραση του χρήστη με το σύστημα ανακάλυψης γνώσης, προκειμένου τα αποτελέσματα να είναι πιο ουσιώδη και αποτελεσματικά. Δεν έχει ωστόσο πλήρως αποσαφηνιστεί πότε και με ποιο τρόπο κρίνεται αναγκαία η παρέμβαση του χρήστη και η απάντηση σε αυτό το ερώτημα είναι κατά πολύ εξαρτώμενη από το πεδίο όπου επιδιώκεται η εξόρυξη της γνώσης και η εφαρμογή των μεθόδων data mining.

Το βέβαιο ωστόσο είναι ότι η συμμετοχή του χρήστη απαιτεί *γρήγορους* χρόνους απόκρισης. Οι απαιτητικοί χρήστες προωθούν και επιβάλουν νέα ζητήματα στα εργαλεία ανακάλυψης γνώσης και οι σχετικοί KDD αλγόριθμοι αποκτούν και τυγχάνουν πρόσθετης προσοχής. Οι χρήστες για παράδειγμα μπορεί να επιθυμούν καταρχήν γρήγορα και λιγότερο ακριβή αποτελέσματα για μια πρώτη ματιά. Στη

συνέχεια μπορεί να εστιάζουν την προσοχή τους σε κάποια επιμέρους σημεία που σίγουρα ανταποκρίνονται στα ενδιαφέροντος τους, και να απαιτούν για τα συγκεκριμένα θέματα μεγαλύτερη ακρίβεια και επεξεργασία. Η αποθήκευση και μελλοντική χρήση της εκμαιευμένης πληροφορίας, ως σημείο αναφοράς σε δυναμικά και ολοένα αυξανόμενα περιβάλλοντα παροχής και εξαγωγής γνώσης, είναι μερικά παραδείγματα από τα προβλήματα που σύντομα θα προστεθούν στην "ατζέντα" των θεμάτων σχετικά με τα σύγχρονα και αλληλεπιδρόντα συστήματα εξαγωγής γνώσης.

5.4.2. Συμμετοχή Προηγούμενης Γνώσης Στο Ερευνητικό Πεδίο

Είναι προφανής η διαπίστωση ότι υπάρχει ανάγκη το σύστημα, που αναλαμβάνει να εκμαιεύσει νέα γνώση, να τροφοδοτείται από ήδη *προηγούμενη* πληροφορία και γνώση σχετική με το αντικείμενο της έρευνας και το πεδίο εφαρμογής.

Παρέχοντας στο σύστημα και στο εργαλείο ανακάλυψης, γνώση σχετικά με το αντικείμενο και πεδίο της εκάστοτε έρευνας, οι διαδικασίες ανακάλυψης μπορούν να επιταχυνθούν, καθώς ο αλγόριθμος μπορεί να επωφεληθεί από την προηγούμενα γνωστή γνώση και να αποφύγει λανθασμένες είτε περιττές ενέργειες. Είναι φανερό για παράδειγμα ότι το όνομα ενός ασθενή δεν αποτελεί αποφασιστικής σημασίας χαρακτηριστικό για την διάγνωση της ασθένειας του. Έτσι το συγκεκριμένο '*attribute'* μπορεί να παραλειφθεί εξολοκλήρου προτού ο αλγόριθμος data mining εμπλακεί στην όλη διαδικασία. Το πρόβλημα της επιλογής και εκμετάλλευσης της γνώσης βρίσκεται ακόμα στο στάδιο της αναζήτησης και της έρευνας.

5.5. Μια Ολοκληρωμένη Αρχιτεκτονική για την Εξαγωγή Γνώσεων απο Κατανεμημένες και Ετερογενείς Βάσεις Δεδομένων

Στο σχήμα 29 παρουσιάζεται η γενικότερη αρχιτεκτονική εξόρυξης (mining) και ανακάλυψης γνώσης από κατανεμημένα και ετερογενή πληροφοριακά συστήματα - στην οποία εντάσσεται και προσαρμόζεται η παρούσα μεταπτυχιακή εργασία. Παρουσιάζονται όλες οι λειτουργίες που επιτελούνται καθώς και οι μεταξύ τους σχέσεις (workflow). Το προτεινόμενο σχήμα μοιάζει πολύ με ένα σύστημα *πληροφοριακής μεσιτείας* (information brokering system), το οποίο μπορεί να γίνει πραγματικότητα από μία σχετική υποστηρικτική *υπηρεσία διαμεσολάβησης* (mediating service). Οι επιμέρους εμπλεκόμενες διαδικασίες και λειτουργίες μπορεί να εκπονηθούν από αντίστοιχους *αυτόνομους πράκτορες* (autonomous agents), με τελικό παραγόμενο αποτέλεσμα ένα *ολοκληρωμένο σύστημα συνεργαζόμενων πρακτόρων*(integrated and co-operating agents) [45, 46].

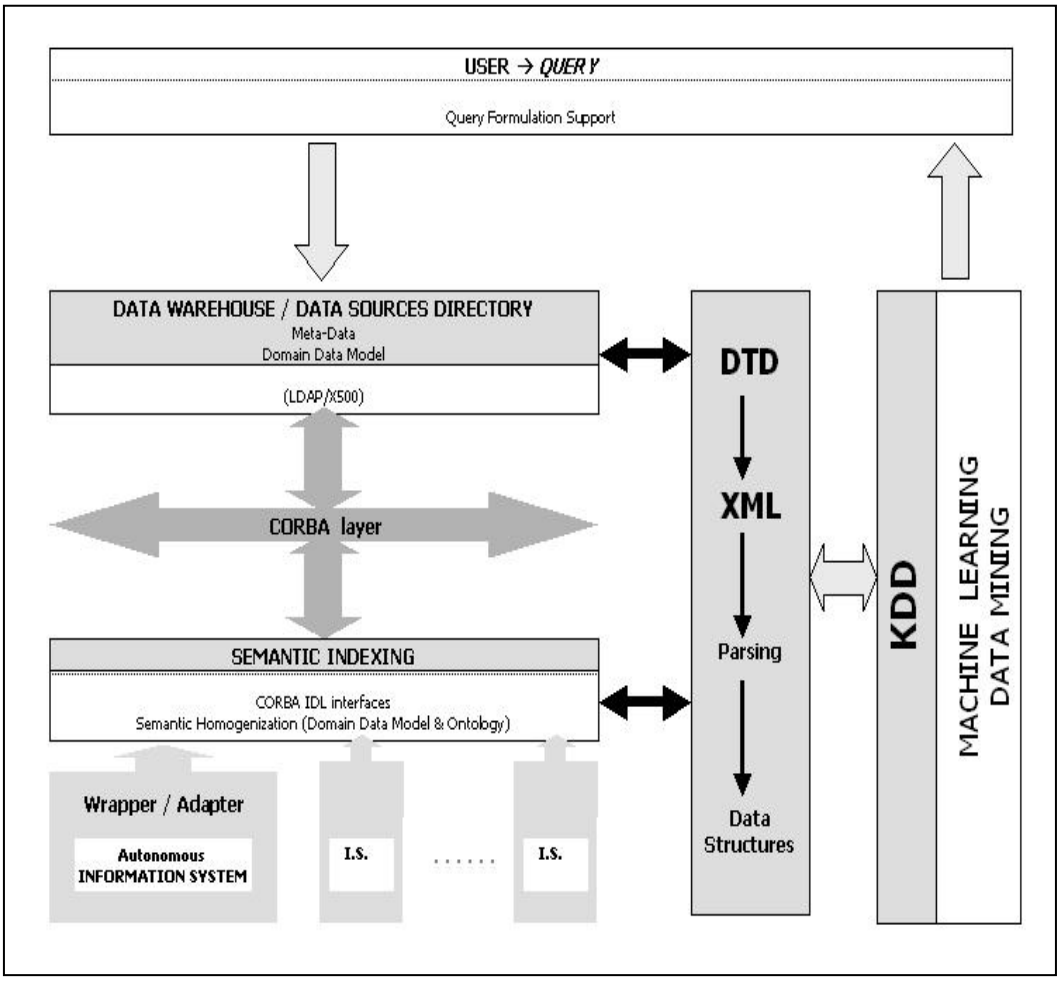
Η πρόσβαση στο σύστημα επιτυγχάνεται και προσφέρεται μέσω των CORBA IDL προσαρμοστών (adapters/ wrappers) και των σχετικών διεπαφών ανάμεσα στα αυτόνομα ιατρικά πληροφοριακά συστήματα και τον *διαμεσολαβητή*, ενώ η "*αποθήκη & οργάνωση δεδομένων*" (data warehouse) παρέχεται μέσω του Patient Clinical Data Directory και PCDD εξυπηρετητή (κεφάλαιο 2). Οι λειτουργίες αυτές όπως αναφέραμε εντάσσονται στο γενικότερο πλαίσιο στήριξης των ολοκληρωμένων δικτυακών τηλεματικών υπηρεσιών και εφαρμογών στην περιφέρειας της Κρήτης. Η εργασία μας επεκτείνει αυτήν την αρχιτεκτονική μέσω των παρακάτω νέο-σχεδιασμένων και υλοποιημένων λειτουργιών.

1. Σημασιολογική ομογενοποίηση.
2. Parsing των XML (COAS compatible) εγγράφων με όλες τις προαναφερθείσες λειτουργίες στις παραγόμενες δομές (κεφάλαια 3-4).
3. Όλες τις KDD/ARM λειτουργίες που θα περιγράψουμε στα επόμενα κεφάλαια.

Οι υπεύθυνοι agents για τα παραπάνω έχουν ήδη υλοποιηθεί

5.5.1. Ένα Ενδεικτικό Σενάριο Χρήσης

Στο σημείο αυτό έχοντας παραθέσει την αρχιτεκτονική στήριξης της δουλειάς μας, προκειμένου να γίνει περισσότερο κατανοητή και επιχειρώντας μια πρώτη εισαγωγή στους κανόνες συσχέτισης που θα μας απασχολήσουν στα επόμενα κεφάλαια, παρουσιάζουμε ένα σενάριο χρήσης όπου συνοψίζονται όλα τα παραπάνω, καθώς και πολλά από αυτά που πρόκειται να ακολουθήσουν και να επεξηγηθούν αναλυτικά στα επόμενα κεφάλαια.



Σχήμα 29. *Η γενικότερη αρχιτεκτονική εξόρυξης και ανακάλυψης γνώσης από κατανεμημένα και ετερογενή πληροφοριακά συστήματα*

---

i. Ο χρήστης μέσω του PCDD, δημιουργεί ένα συγκεκριμένο ερώτημα(query). Για παράδειγμα αυτός/ή μπορεί να ενδιαφέρεται για όλα τα encounters (present in the federation) με τιμές για κλινικές εξετάσεις(clinical findings – CF), στο σύνολο: {CF1, …, CF12), τιμές για εργαστηριακές εξετάσεις(laboratory results - LR) στο σύνολο: {LR13, …, LR21}, και τιμές για διαγνώσεις(diagnoses) στο σύνολο: {D22, …, D30}. Επιπρόσθετα καθορίζει τα επιθυμητά όρια 'minsup' και 'minconf' για τους

αναζητούμενους κανόνες συσχέτισης(αναλυτική επεξήγηση κεφάλαιο 6).

i. Ο PCDD server καθορίζει τα links στους encounters που σχετίζονται με τα ζητούμενες κλινικές εξετάσεις, εργαστηριακές εξετάσεις και τις τιμές για τις διαγνώσεις.
ii. Τα αντίστοιχα αυτόνομα ιατρικά πληροφοριακά συστήματα προσπελαύνονται (μέσω των CORBA IDL wrappers και interfaces), και οι λεπτομέρειες για τα αντίστοιχα encounters των ασθενών που ικανοποιούν τις προκαθορισμένες τιμές (CF, LR, and D) καλούνται και ανακλώνται.
iii. Η λειτουργία δημιουργίας του συμβατού με την ορισμένη DTD(κεφάλαιο 3) XML αρχείου τίθεται σε εφαρμογή και το αντίστοιχο *XML query-specific*- αρχείο δημιουργείται.
iv. Το παραγόμενο XML αρχείο περνάει από τον parser, όπως περιγράφηκε στο κεφάλαιο 4 ομογενοποιείται σημασιολογικά(κεφ. 3), ενώ παράλληλα δημιουργούνται οι δομές που περιγράφηκαν, με χρήση των απαιτούμενων κατά περίσταση KDD διεργασιών.
v. Οι KDD/ARM λειτουργίες (όπως θα περιγραφούν αναλυτικά στα επόμενα κεφάλαια) επιτελούνται για την ανακάλυψη των ζητουμένων κανόνων συσχέτισης(association rules) ανάμεσα στα ιατρικά δεδομένα εισόδου.

5.5.2. Ένα υποθετικό παράδειγμα

Στο σημείο αυτό και για τον σκοπό επίδειξης του παραπάνω σεναρίου (κάνοντας μια πρώτη εισαγωγή και για τους κανόνες που επιδιώκουμε να δημιουργήσουμε) και αποφεύγοντας για την ώρα να χρησιμοποιήσουμε ιατρική ορολογία, δημιουργούμε μια τεχνική βάση από 10 *transactions* (δηλαδή encounters/ επισκέψεις ασθενών), και ένα μεταβλητό αριθμό από items σε καθένα από αυτά (π.χ, CF1-12, LR13-21, D22-30). Τα παραπάνω απεικονίζονται στον πίνακα 1 και αποτελούν συνοπτικά το υποθετικό αποτέλεσμα στην ερώτηση που υποβάλαμε.

Πίνακας 1. *Η τεχνητή κλινική βάση (transactions and items)*

Transaction	Items
1	CF3
2	CF3, CF6, CF8, CF12
3	CF2, CF3, CF6, CF8, LR15, LR17
4	CF6, CF8, LR15, LR21
5	CF3, CF6, CF8, LR15, D23
6	CF1, CF6, CF8, LR15, D26, D29
7	CF3, CF6, CF8, CF12
8	CF3, CF6, CF8, CF12
9	CF3, CF6, CF8, CF12
10	CF3, CF6, CF8, CF12, LR15, LR17

Υποθέτοντας ότι τα προκαθορισμένα όρια είναι: minsup=70% και minconf=70%, έξι- 6, κανόνες συσχέτισης δημιουργούνται, όπως φαίνεται στον παρακάτω πίνακα 2.

Οι κανόνες που παράγονται - και θα προσπαθήσουμε να εξηγήσουμε αναλυτικά την όλη διαδικασία παραγωγής τους στα επόμενα κεφάλαια – για το συγκεκριμένο παράδειγμα είναι οι εξής:

Πίνακας 2. *Οι παραγόμενοι κανόνες για τα transactions του πίνακα 5.1. (minsup=70%, minconf=10%)*

	Support %	**Confidence** %
CF3 ⇒ CF6	70	87.5
CF8 ⇒ CF3	70	77.8
CF8 ⇒ CF6	90	100.0
CF6, CF8 ⇒ CF3	70	77.8
CF3, CF8 ⇒ CF6	70	100.0
CF3, CF6 ⇒ CF8	70	100.0

Με το παράδειγμα αυτό επιδιώξαμε μια πρώτη εισαγωγή στο κεφάλαιο των κανόνων συσχέτισης με σκοπό να προϊδεάσουμε τον χρήστη με αυτά που πρόκειται να ακολουθήσουν.

Κεφάλαιο 6:
Ανακάλυψη Κανόνων Αλληλοσυσχέτισης
(Association Rules Mining)

Στο κεφάλαιο αυτό παρουσιάζονται σε μεγαλύτερη λεπτομέρεια θέματα σχετικά με την εξόρυξη και *ανακάλυψη κανόνων συσχέτισης* (*Association Rule Mining* - ARM). Ξεκινάμε με την κατανόηση και παρουσίαση των απλών και βασικών εννοιών για το ARM, περιγράφοντας τα *συχνά εμφανιζόμενα σύνολα*, όπου στο εξής θα αναφέρουμε ως *frequent/ large sets*, και ορίζοντας το ARM. Στη συνέχεια παρουσιάζουμε τις βασικές ιδιότητες των ARM και το βασικό αλγοριθμικό σχήμα που ακολουθείται σε τέτοιες περιπτώσεις και κλείνουμε με μια ικανοποιητική αναφορά στους βασικότερους προηγούμενους αλγόριθμους στο πεδίο της έρευνας μας.

6.1. Ορισμός των Frequent/ Large Sets

Έστω $I = \{\iota_1,\iota_2,\ldots,\iota_m\}$ ένα σύνολο από literals, τα οποία στο εξής θα αναφέρουμε ως ***items***. Έστω τώρα ***D*** ένα σύνολο από ***transactions***, όπου κάθε transaction ***T*** είναι ένα σύνολο από items, τέτοιο ώστε $T \subseteq I$. Θεωρούμε ότι ένα transaction δεν περιέχει όμοια items(duplicates) και υποθέτουμε ότι τα στοιχεία που ανήκουν σε αυτό, είναι ταξινομημένα.

Θεωρούμε και αντιστοιχούμε σε κάθε transaction της βάσης (από τα n συνολικά), ένα μοναδικό αύξοντα αριθμό "*ταυτότητας*" (*TID*).

Λέμε ότι ένα transaction *T* υποστηρίζει(support) ένα σύνολο από items *X*, εάν $X \subseteq T$. Μερικές φορές υπάρχει ανάγκη να αναφερθούμε στο σύνολο το transactions τα οποία υποστηρίζουν το *X*. Σ' αυτήν την περίπτωση χρησιμοποιούμε το συμβολισμό T(X) για να αναφερθούμε στο σύνολο των TIDs αυτών των transactions.

Ορίζουμε το ***support*** ενός συνόλου *X*, και το συμβολίζουμε ως ***supp(X)***, το λόγο όλων των transactions στο *D* που υποστηρίζουν το X, προς το συνολικό αριθμό των transactions στο D. Στην περίπτωση όπου $supp(X) \geq S_{min}$, όπου S_{min} μια προκαθορισμένη ελάχιστη τιμή για το support(***minsup***), το σύνολο *X* θεωρείται ***large/ frequent.*** Το κίνητρο και ο λόγος ύπαρξης του ελάχιστου support, είναι ότι θέλουμε να ασχολούμαστε με itemsets, τα οποία εμφανίζονται αρκετά συχνά, ώστε να θεωρούνται ενδιαφέροντα (*interesting*). Κατ' επέκταση, itemsets τα οποία δεν είναι frequent/large, δεν είναι και ενδιαφέροντα για περαιτέρω επεξεργασία. Τέλος θα αναφερόμαστε και θα αποκαλούμε ένα itemset με πληθυσμό $k = |X|$, ως ένα *k-itemset*.

6.2. Ορισμός Των Κανόνων Συσχέτισης(Association Rules)

Ένας κανόνας συσχέτισης (association rule), είναι μια παραγωγή της μορφής $\boldsymbol{X \Rightarrow Y}$ όπου $X \subset I$, $Y \subset I$ και $X \cap Y = \varnothing$, (η απαίτηση τα *X, Y* να είναι χωρίς κοινά μέλη δεν είναι απολύτως αναγκαία, καθότι δεν οδηγούμαστε σε κανόνες χωρίς νόημα, αλλά σε κανόνες περιττούς και ασήμαντους). Επιπρόσθετα, απαιτείται να είναι $Y \neq \varnothing$. Ένας τέτοιος κανόνας μπορεί να θεωρηθεί ως η *πρόβλεψη* ότι: εάν ένα transaction υποστηρίζει το itemset *X*, τότε θα υποστηρίζει και το itemset *Y* με ένα μέτρο ***confidence*** του κανόνα, και συμβολίζεται ως ***conf(R)***.

Το confidence ενός κανόνα *R*, ορίζεται ως η *δεσμευμένη πιθανότητα*, τέτοια ώστε δοθέντος ότι το transaction *T* υποστηρίζει το *X*, τότε θα υποστηρίζει και το *Y*. Σε επίσημη μαθηματική μορφή:

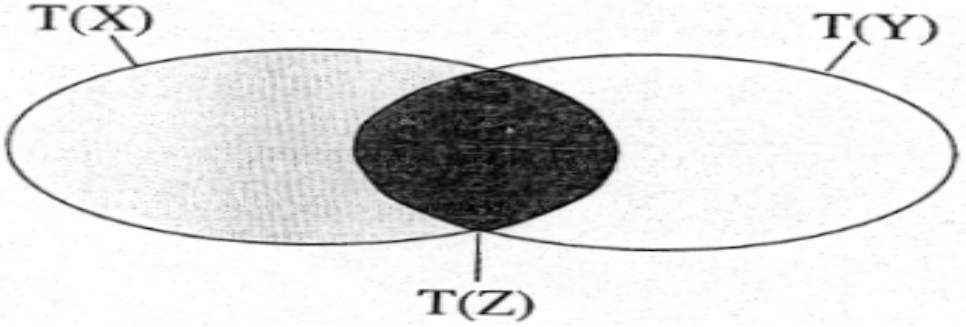
$$conf(R) = p(Y \subseteq T \mid X \subseteq T) = p(Y \subseteq T \cap X \subseteq T) / p(X \subseteq T) = supp(X \cup Y) / supp(X).$$

Το support ενός κανόνα *R* στο *D*, ορίζεται ως $supp(X \cup Y)$. Το confidence του κανόνα φανερώνει πως συχνά αναμένεται να εμφανισθεί, ενώ το support του φανερώνει κατά κάποιο τρόπο το πόσο *αξιόπιστος* είναι αυτός ο κανόνας. Για να είναι σημαντικός και ενδιαφέρων ένας κανόνας θα πρέπει να έχει αυξημένο support και ικανοποιητικό confidence. Θα ισχυριζόμαστε λοιπόν ότι ένας κανόνας *R*, λαμβάνει χώρα στο *D*, εάν για τις προκαθορισμένες από το χρήστη ελάχιστες τιμές C_{min} και S_{min}, όπου C_{min} το ελάχιστο confidence(***minconf***) και S_{min} το ελάχιστο support του κανόνα, ισχύει $conf(R) \geq C_{min}$ και $supp(R) \geq S_{min}$. Στο σημείο αυτό σημειώνουμε ότι τόσο το ηγούμενο μέρος του κανόνα, όσο και το ακόλουθο θα πρέπει να είναι frequent/large.

Για να δείξουμε την σημαντικότητα και των δυο απαιτήσεων (ελάχιστο support και ελάχιστο confidence), ας υποθέσουμε καταρχήν ότι βασίζουμε τον κανόνα μας στην εμφάνιση ενός μόνο, όχι ιδιαίτερα συχνά εμφανιζομένου item. Αυτός ο κανόνας θα έχει προφανώς *maximum confidence* (100%), ωστόσο δεν περιγράφει κάποιο συχνά εμφανιζόμενο itemset και κατά συνέπεια θα πρέπει να "*πεταχτεί*", καθότι δεν πληροί το κριτήριο $supp(R) \geq S_{min}$. Ας υποθέσουμε τώρα ότι ένας κανόνας έχει υψηλό support, αλλά χαμηλό confidence. Ας δανειστούμε το παράδειγμα από μια βάση πωλήσεων ενός supermarket και ας θεωρήσουμε ότι 2% των πελατών που αγοράζουν σαπούνι, αγοράζουν και τομάτες. Μολονότι τα επιμέρους items υποστηρίζονται αρκετά στη βάση, δεν είναι σχετικά μεταξύ τους, καθότι δεν εκφράζουν μια ισχυρή σχέση.

6.3. Ιδιότητες των Κανόνων Συσχέτισης

Όπως αναφέραμε, η απαίτηση τα *X*, *Y* να είναι χωρίς κοινά μέλη δεν είναι απολύτως αναγκαία, καθότι δεν οδηγούμαστε σε κανόνες χωρίς νόημα, αλλά σε κανόνες περιττούς και ασήμαντους. Το $X \rightarrow X$ για παράδειγμα είναι τετριμμένα αληθές και το $X \rightarrow X \cup Y$ είναι ισοδύναμο με το $X \rightarrow Y$ και επομένως όχι ενδιαφέρον. Το ηγούμενο μέρος ενός κανόνα μπορεί να είναι κενό. Έτσι κάθε transaction θεωρείται ότι υποστηρίζει το κενό itemset και κατ' επέκταση ολόκληρη η βάση ικανοποιεί τον ηγούμενο όρο. Το confidence ενός τέτοιου κανόνα είναι ίσο με τη σχετική συχνότητα εμφάνισης του ακόλουθου μέρους. Απαιτούμε ο ακόλουθος *Y* να μην είναι κενός, για τον ίδιο λόγο που απαιτούμε ακόλουθο και ηγούμενο μέρος να μην έχουν κενά στοιχεία.



Σχήμα 30. *Παράδειγμα Σύνθεσης Κανόνων*

Παραθέτουμε εφτά- 7 βασικές ιδιότητες, από τις οποίες οι τρεις- 3 πρώτες ιδιότητες

των *Frequent/Large sets*, θα φανούν καταρχήν πολύ βοηθητικές όπως θα δούμε, κυρίως στην εύρεση των ARM αλγορίθμων.

- **Ιδιότητα 1 - Συνθήκη Support για Υποσύνολα**.
  Εάν για τα itemsets *A, B* ισχύει $A \subseteq B$, τότε $supp(A) \geq supp(B)$, επειδή όλα τα transactions στο *D* που περιέχουν το *B*, αναγκαστικά περιέχουν και το *A*.
- **Ιδιότητα 2 - Τα Υπερσύνολα όχι αποδεδειγμένων Large/ Frequent (Infrequent) Συνόλων, είναι επίσης Infrequent**.
  Στην περίπτωση όπου για ένα itemset *A* ισχύει $supp(A) < Smin$, τότε κανένα υπερσύνολο *B* του *A* δεν θα είναι frequent/ large, λόγω του ότι ισχύει $supp(B) \leq supp(A) < Smin$ από την προηγούμενη ιδιότητα.
- **Ιδιότητα 3 - Τα Υποσύνολα Frequent Sets είναι επίσης Frequent**.
  Εάν ένα itemset *B* είναι frequent/ large στο *D* (δηλαδή $supp(B) \geq Smin$), τότε και κάθε υποσύνολο *A* του *B* είναι επίσης frequent/large στο *D*, καθώς $supp(A) \geq supp(B) \geq Smin$ σύμφωνα με την 1. Στην πράξη εάν το itemset $A = \{\iota_1, \iota_2, \ldots \iota_\kappa\}$ είναι large, τότε και όλα τα κ $(\kappa\text{-}1)$ υποσύνολα του θα είναι επίσης large**.** <u>Δεν ισχύει το αντίθετο!</u>
- **Ιδιότητα 4 - Δεν Επιτρέπεται η Σύνθεση των Κανόνων**.
  Εάν οι κανόνες $X \rightarrow Z$ και $Y \rightarrow Z$ λαμβάνουν χώρα στο *D*, αυτό δεν σημαίνει απαραίτητα ότι και ο κανόνας $X \cup Y \rightarrow Z$ είναι αληθής στο *D*. Ας θεωρήσουμε την περίπτωση όπου $X \cap Y = \varnothing$ και το *Z* συνεπάγεται(υποστηρίζεται) στο *D*, εάν και μόνο αν είτε το *X*, είτε το *Y* υποστηρίζεται από τα transactions της βάσης. Στην περίπτωση αυτή το itemset $X \cup Y$ έχει support 0 και κατ' επέκταση ο κανόνας $X \cup Y \rightarrow Z$ έχει *0%* confidence. Το ίδιο ισχύει και για την σύνθεση κανόνων με το ίδιο ηγούμενο μέρος: $X \rightarrow Y \wedge X \rightarrow Z \not\Rightarrow X \rightarrow Y \cup Z$
- **Ιδιότητα 5 - Διαχωρισμός Των Κανόνων.**
  Εάν ισχύει ο κανόνας $X \cup Y \rightarrow Z$, <u>δεν</u> είναι σίγουρο ότι ισχύουν και οι κανόνες $X \rightarrow Z$ και $Y \rightarrow Z$. Το γεγονός αυτό εμφανίζεται στην περίπτωση όπου για παράδειγμα το Z εμφανίζεται σε ένα transaction, εάν και μόνο αν εμφανίζονται σε αυτό τόσο το *X* όσο και το *Y*, δηλαδή εάν $supp(X \cup Y) = supp(Z)$. Εάν τα support για το *X* και το *Y* είναι πολύ μεγαλύτερα από το $supp(X \cup Y)$, οι δυο κανόνες $(X \rightarrow Z$ και $Y \rightarrow Z)$ δεν έχουν το απαιτούμενο confidence. Η περίπτωση αυτή σχηματικά αποδίδεται στο σχήμα 30. Οι κύκλοι αντιστοιχούν στο σύνολο των transactions που υποστηρίζουν τα αντίστοιχα itemset. Εντούτοις το αντίθετο ισχύει, δηλ. $X \rightarrow Y \cup Z \Rightarrow X \rightarrow Y \wedge X \rightarrow Z$, λόγω του ότι $supp(XY) \geq supp(XYZ)$ και $sup(XZ) \geq supp(XYZ)$. Έτσι οι τιμές τόσο του support, όσο και του confidence μικρότερων κανόνων, αυξάνονται συγκριτικά με τις αντίστοιχες τιμές του original κανόνα. Δυστυχώς αυτό δεν βοηθάει ιδιαίτερα κατά τη διάρκεια κατασκευής των εξαγόμενων κανόνων, διότι εμείς επιθυμούμε τη κατασκευή μεγαλύτερων κανόνων από άλλους μικρότερους και όχι το αντίστροφο.
- **Ιδιότητα 6 - Δεν ισχύει η Μεταβατικότητα.**
  Εάν ισχύουν οι κανόνες $X \rightarrow Y$ και $Y \rightarrow Z$ δεν μπορούμε να ισχυριστούμε ότι ισχύει και ο $X \rightarrow Z$. Υποθέτουμε για παράδειγμα ότι ισχύει $T(X) \subset T(Y) \subset T(Z)$ και ότι το ελάχιστο αποδεκτό confidence είναι C_{min}. Ας θεωρήσουμε ότι $conf(X \rightarrow Y) = conf(Y \rightarrow Z) = C_{min}$. Βασισμένοι τώρα στις σχετικές τιμές των support

έχουμε ότι: $conf(X \rightarrow Z) = C^2_{min} < C_{min} < 1$, το οποίο σημαίνει ότι το confidence δεν είναι αρκετό για να ισχύσει ο κανόνας.

- **Ιδιότητα 7 - Συμπέρασμα για το εάν ισχύει ένας Κανόνας.**
  Η δεύτερη ιδιότητα δείχνει ότι εάν ένας κανόνας της μορφής $A \rightarrow (L - A)$ δεν έχει το ελάχιστο confidence, τότε ούτε και ο κανόνας $B \rightarrow (L - B)$ πρόκειται να το έχει, για τα itemsets L, A, B και $B \subseteq A$. Χρησιμοποιώντας τη σχέση $supp(B) \geq supp(A)$ (ιδιότητα 1) και τον ορισμό του confidence διαπιστώνουμε ότι $conf(B \rightarrow (L - B)) = supp(L)/supp(B) \leq supp(L)/supp(A) < c$. Ανάλογα εάν ισχύει ο κανόνας $(L - C) \rightarrow C$, τότε θα ισχύουν και όλοι οι κανόνες της μορφής $(L - D) \rightarrow D$, όπου $D \subseteq C$ και $D \neq 0$, διότι το ακόλουθος μέρος απαιτείται να είναι μη κενό. Η παρούσα ιδιότητα χρησιμεύει, όπως θα τονίσουμε σε επόμενο κεφάλαιο, στην *επιτάχυνση* της διαδικασίας ανακάλυψης των κανόνων, όταν όλα τα frequent sets και τα support τους έχουν καθοριστεί.

6.4. Το βασικό αλγοριθμικό σχήμα για Ανακάλυψη Κανόνων Αλληλοσυσχέτισης

Έχοντας περιγράψει τα παραπάνω, είμαστε σε θέση πλέον να παρουσιάσουμε και να αναλύσουμε την μορφή του βασικού αλγορίθμου για την ανακάλυψη των ζητούμενων κανόνων. Αν και οι αλγόριθμοι που θα παρουσιάσουμε παρακάτω διαφέρουν μεταξύ τους, όλοι τους χρησιμοποιούν ένα βασικά κοινό σχήμα. Για την κατασκευή των κανόνων αλληλοσυσχέτισης απαιτείται η μέτρηση και ο υπολογισμός των *support* όλων των *large* συνόλων. Για το λόγο αυτό, όλοι οι αλγόριθμοι διακρίνονται από δυο βασικές φάσεις: (**i**) δημιουργούνται όλα τα *frequent/ large σύνολα*, και (**ii**) παράγονται οι επιτρεπτοί κανόνες με τα αντίστοιχα *confidence*, χρησιμοποιώντας τα ήδη υπολογισμένα frequent/ large σύνολα.

I. Εύρεση/ Ανακάλυψη όλων των Large/Frequent Sets
Εν γένει οι αλγόριθμοι που χρησιμοποιούνται για την ανακάλυψη των *large items* κάνουν πολλαπλά περάσματα στα αρχικά δεδομένα εισόδου. Στο πρώτο πέρασμα μετριέται το support του κάθε item χωριστά και καθορίζεται ποια items έχουν minimum support και κατ' επέκταση είναι large. Σε κάθε επόμενο βήμα, ξεκινάμε με ένα αρχικό σύνολο από itemsets, τα οποία βρέθηκαν large στο αμέσως προηγούμενο βήμα. Χρησιμοποιούμε αυτό το σύνολο για τη δημιουργία των νέων υποθετικά large itemsets, τα οποία στο εξής θα αναφέρουμε ως *υποψήφια* (candidate itemsets). Κατά τη διάρκεια της νέας ανάγνωσης των δεδομένων μας, μετράμε το πραγματικό support των υποψηφίων itemsets και στο τέλος αποφασίζουμε ποια από τα υποψήφια itemsets είναι και large. Αυτή η διαδικασία συνεχίζεται έως ότου δεν μπορούν να δημιουργηθούν άλλα large itemsets.

II. Κατασκευή/ Παραγωγή Κανόνων
Εφόσον έχουν υπολογιστεί και είναι διαθέσιμες όλες οι τιμές των support των itemsets, καθορίζονται τα αντίστοιχα confidence και δημιουργούνται οι πιθανοί κανόνες. Για κάθε frequent/large set X, καθένα από τα πιθανά υποσύνολα του επιλέγεται ως πρώτο μέρος του κανόνα και τα υπόλοιπα items διαμορφώνουν το δεύτερο μέρος. Καθώς το X είναι frequent θα πρέπει και όλα τα υποσύνολα του να είναι frequent σύμφωνα με την ιδιότητα 3 και τα support τους είναι ήδη γνωστά. Στη συνέχεια υπολογίζεται το confidence του κανόνα και σε σχέση με το προκαθορισμένο ελάχιστο όριο C_{min}, ο κανόνας είναι αποδεκτός ή όχι.

Βασισμένοι στην ιδιότητα 7 μπορούμε να επιτύχουμε βελτιώσεις καθώς, εάν κάποιος κανόνας δεν γίνει αποδεκτός τότε δεν είναι ανάγκη να γίνει έλεγχος και για κανένα άλλο κανόνα που θα έχει ως πρώτο μέρος του κάποιο από τα υποσύνολα του. Τα παραπάνω θα περιγραφούν αναλυτικά παρακάτω σε συγκεκριμένο υποκεφάλαιο του επόμενου κεφαλαίου.

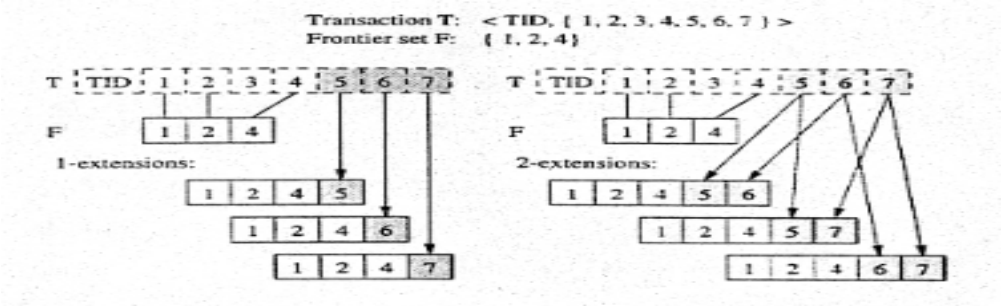
6.5. Προηγούμενοι Αλγόριθμοι

Όλοι οι προτεινόμενοι στην βιβλιογραφία ARM αλγόριθμοι διαχωρίζουν την κατασκευή των κανόνων, από την εύρεση των large/frequent συνόλων [47]. Το δεύτερο μέρος είναι και το πιο ενδιαφέρον και αυτό που λύνεται διαφορετικά από τους επιμέρους αλγορίθμους. Έτσι στις παρακάτω ενότητες θα επικεντρωθούμε στην ανακάλυψη των συχνά εμφανιζομένων συνόλων και θα περιγράψουμε τις τρεις μορφές αναπαράστασης των δεδομένων που χρησιμοποιούν οι αλγόριθμοι και είναι οι: (α) item-lists, (β) candidate-lists και (γ) TID-lists. Θα περιγράψουμε τα μειονεκτήματα και τα πλεονεκτήματα τους και θα συζητήσουμε πως η επιλογή τους επηρεάζει την απόδοση των αλγορίθμων που τις χρησιμοποιούν.

6.5.1. AIS

Το πρόβλημα των κανόνων συσχέτισης πρωτοεμφανίστηκε στο [3], με έναν αλγόριθμο ο οποίος στην συνέχεια ονομάστηκε *AIS* από τα ονόματα των συγγραφέων του [4].

Προκειμένου ο AIS να βρει τα frequent sets, δημιουργεί δυναμικά (on-the-fly) υποψηφίους όρους ενώ διαβάζει τη βάση. Αρκετά περάσματα της βάσης κρίνονται αναγκαία, και κατά τη διάρκεια καθενός από αυτά όλα τα transactions της βάσης διαβάζονται το ένα μετά το άλλο. Ένας υποψήφιος δημιουργείται με την προσθήκη νέων όρων σε σύνολα που έχουν ήδη διαπιστωθεί και χαρακτηριστεί ως frequent σε προηγούμενες εξετάσεις της βάσης. Τέτοια σύνολα αποκαλούνται ως *frontier sets*. Ο υποψήφιος που δημιουργείται από την προσθήκη ενός item σε ένα frontier set *F* ονομάζεται *1-extension* του *F*, επειδή ένα στοιχείο έχει προστεθεί στο *F*. Για να αποφευχθούν επανεμφανίσεις υποψηφίων, το item που προστίθεται πρέπει να είναι λεξικογραφικά μεγαλύτερο, από το μεγαλύτερο item του *F*, εφόσον αναφερόμαστε σε 1-extensions. Προκειμένου να αντιμετωπιστεί και να αποφευχθεί η περιττή δημιουργία υποψηφίων, τα οποία δεν εμφανίζονται και παρουσιάζονται στη βάση, ο AIS δεν δημιουργεί κάποιον υποψήφιο εάν πρώτα δεν τον εντοπίσει κατά τη διάρκεια ανάγνωσης της βάσης. Το σχήμα 31 υποδεικνύει ποια *1*-extensions δημιουργούνται όταν διαβάζεται ένα transaction το οποίο υποστηρίζει ένα frontier set *F*. Το σχήμα επεξηγεί πως η ιδέα μπορεί να επεκταθεί στην δημιουργία *k*-extensions, με την αντίστοιχη προσθήκη *k* items στο frontier set.



Σχήμα 26. *Απεικόνιση των 1 και 2 extensions του AIS*

Κάθε υποψήφιος είναι συσχετισμένος με έναν *μετρητή* ο οποίος μετρά την

συχνότητα εμφάνισης του. Όταν δημιουργείται για πρώτη φορά ένας υποψήφιος ο μετρητής του είναι 1, ενώ αυξάνεται προοδευτικά και κατά μια μονάδα κάθε φορά που επανεμφανίζεται σε κάποιο άλλο transaction.

Στις περισσότερες περιπτώσεις οι απαιτήσεις αποθήκευσης τόσο για τα υποψήφια sets όσο και για τα frontier sets, υπερβαίνουν το μέγεθος της κύριας μνήμης, έτσι τα frontier sets πρέπει να γράφονται και να διαβάζονται αντίστοιχα από περιφερειακή μνήμη (πχ., σκληρό δίσκο).

Εφόσον καθοριστούν τα νέα frontier sets, ξεκινά η επόμενη φάση επέκτασης/ μέτρησης (extension/ counting). Η όλη διαδικασία σταματά όταν δεν υπάρχουν πλέον frontier sets, που σημαίνει ότι κανένας από τους προηγούμενους υποψηφίους δεν ήταν frequent. Αρχικά το μόνο frontier set είναι το $\varnothing$, το οποίο επεκτείνεται σε όλα τα *1*-itemsets στο πρώτο βήμα, στη συνέχεια προκύπτουν με ανάλογο τρόπο τα *2*-itemsets και ούτω καθεξής.

Δυστυχώς αυτή η στρατηγική δημιουργίας υποψηφίων, προκαλεί την εμφάνιση ενός μεγάλου αριθμού από υποψήφιους και πρόσθετες *τεχνικές κλαδέματος* (pruning) κρίνονται αναγκαίες για την απόφαση της περαιτέρω επέκτασης ή όχι κάποιων υποψήφιων συνόλων. Οι τεχνικές αυτές υπολογίζουν και εκτιμούν κατά προσέγγιση το support κάποιου μελλοντικού υποψηφίου, βασισμένες στις σχετικές συχνότητες εμφάνισης των υποσυνόλων του. Τέτοιες αποφάσεις όμως προκαλούν πρόσθετο κόστος σε χρόνο και μνήμη, καθώς γίνονται επαναλαμβανόμενα σε πολλά υποσύνολα ενός transaction.

6.5.2. SETM

Ο *SETM* [22] σχεδιάστηκε για να εκτελεί μόνο βασικές λειτουργίες βάσεων δεδομένων, προκειμένου να βρίσκει τα frequent sets. Για το λόγο αυτό χρησιμοποιεί τη δικιά του αναπαράσταση, με βάση την οποία αποθηκεύει κάθε itemset συνδυασμένο με το αντίστοιχο TID του transaction που το υποστηρίζει. Το σχήμα 32 δείχνει ένα παράδειγμα από την εκτέλεση του αλγορίθμου σε μια μικρή βάση δεδομένων, όπου και φαίνεται η μορφή αποθήκευσης της βάσης με τη μορφή *: <TID, itemlist>* εγγραφών (records). Ο SETM επαναλαμβανόμενα τροποποιεί εξολοκλήρου τη βάση, εκτελώντας τις λειτουργίες δημιουργίας υποψηφίων, μέτρησης των support και διαγραφής των μη συχνά εμφανιζομένων itmes.

Κάνοντας χρήση του σχήματος παραθέτουμε και την εξήγηση του αλγορίθμου. Υποθέτουμε ότι όλα τα itemsets που δεν υπερβαίνουν το όριο του *minimum support* έχουν διαγραφεί, γι' αυτό και το *<1,{5}>* δεν είναι μέρος του L_1. Για να υπολογίσουμε το C_2, το L_1 ενώνεται εκ νέου με το L_1, εκεί όπου υπάρχουν κοινά TID, όπως φαίνεται στο σχήμα. Για κάθε transaction το C_2 περιέχει όλους τους *2*-υποψηφίους που αυτό υποστηρίζει συνοδευμένους με το αντίστοιχο TID. Στο επόμενο βήμα διαγράφονται όλοι οι μη-συχνά εμφανιζόμενοι υποψήφιοι και από το *C2* περνάμε στο L_2, όπου εκεί υπάρχουν όλα τα large *2*-itemsets. Αυτό γίνεται αφού πρώτα ταξινομηθούν τα itemsets στο C_2. Ας υποθέσουμε τώρα ότι θέλουμε να υπολογίσουμε όλους τους *3*-υποψηφίους. Αυτό γίνεται με την παρακάτω join λειτουργία *(κ==2)*:

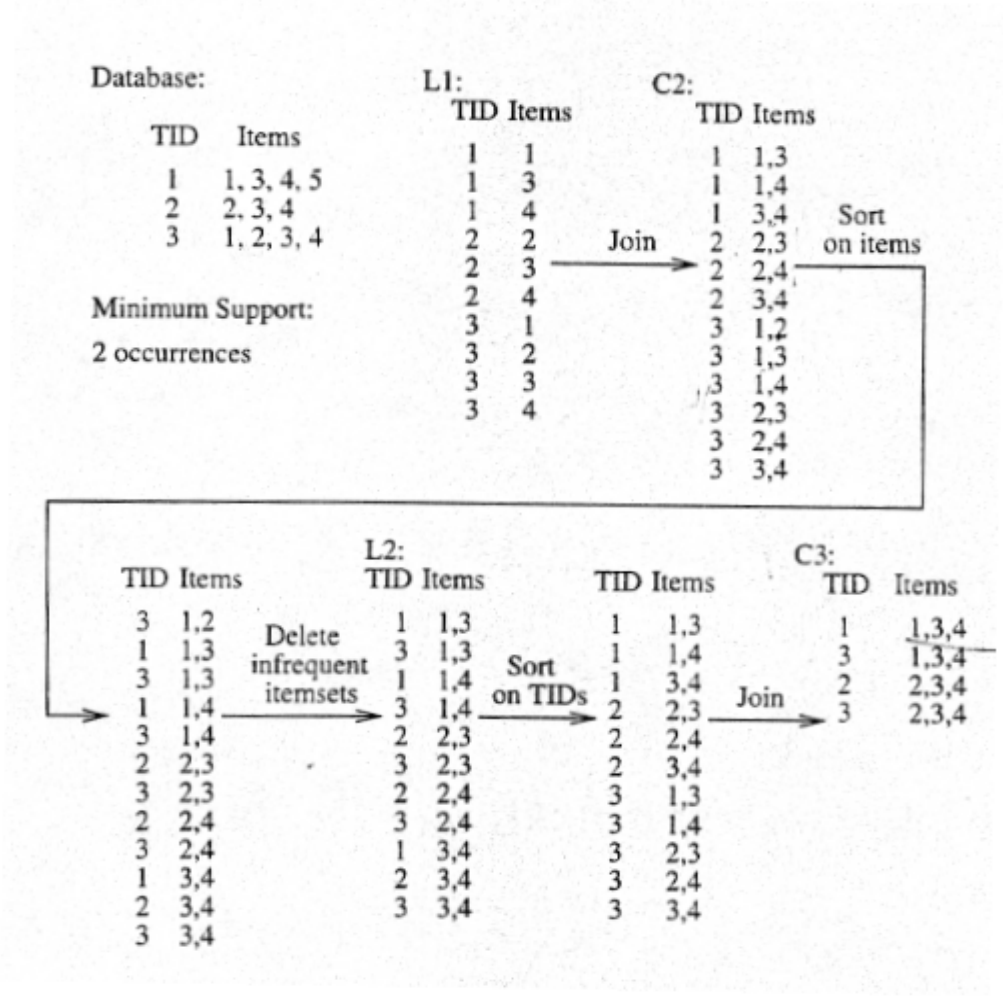
```
insert into C(k+1)
select a.TID, a. item1, a. item2, …,a. itemk, b. itemk
from  Lk a, Lk b
where a.TID = b.TID, a.item1 = b.item1,…, a.itemk-1 = b. itemk-1,   a. itemk< b. itemk
```

Δυστυχώς το L_2 είναι ταξινομημένο με βάση τα items και όχι με βάση τα TIDs, όπως χρειάζεται για να γίνει σωστά και αποτελεσματικά το join. Έτσι το L_2

πρέπει να ταξινομηθεί εκ νέου, βάση των TIDs πριν εκτελεστεί το join. Στη συνέχεια δημιουργείται το C_3, όπου και ταξινομείται ως προς τα στοιχεία, διαγράφονται τα μη συχνά εμφανιζόμενα sets και η διαδικασία προχωράει έως ότου δημιουργηθεί κάποιο κενό L_k.

Το πρόβλημα με αυτόν τον αλγόριθμο είναι ότι δημιουργεί ίδιους υποψηφίους προερχόμενους από διαφορετικά transactions με αποτέλεσμα να δημιουργούνται τεράστιος αριθμός ενδιάμεσων αποτελεσμάτων. Επιπρόσθετα τα itemsets θα πρέπει να είναι *ταξινομημένα*. Το χειρότερο όμως, όπως αναφέραμε, είναι ότι αυτοί οι τεράστιοι πίνακες που προκύπτουν θα πρέπει να ταξινομούνται δυο φορές προκειμένου να προκύπτουν τα επόμενα κάθε φορά large itemsets.

Το βασικό πλεονέκτημα και η καινοτομία του αλγορίθμου είναι ότι ο SETM δημιουργεί λιγότερους υποψηφίους από τον AIS. Για του λόγου το αληθές ας υποθέσουμε ότι ο AIS διαβάζει το παραπάνω παράδειγμα και βρίσκεται στο τρίτο πέρασμα έχοντας όλα τα L_2 itemsets ως frontier sets. Όταν θα επεξεργάζεται το πρώτο transaction ο AIS θα διαπιστώσει ότι υποστηρίζει τα frontier sets *{1,3}, {1,4}* και *{3,4}* και ότι περιέχεται το item 5. Τίποτα δεν θα τον σταματήσει από το να δημιουργήσει τα υποψήφια *{1,3,5}, {1,4,5} και {3,4,5},* αν και το *5* δεν είναι ούτε καν frequent. Ο SETM δεν θα θεωρήσει ποτέ αυτά τα σύνολα ως υποψηφίους.



Σχήμα 32. *Μέρος εκτέλεσης του παραδείγματος από τον SETM*

6.5.3. *Apriori, AprioriTid και AprioriHybrid*

Ο μεγάλος αριθμός των υποψηφίων που δημιουργούσε ο AIS ώθησε τους δημιουργούς του να αναπτύξουν μια καινούρια στρατηγική δημιουργίας υποψηφίων, η οποία ονομάστηκε `Apriori-gen` και αποτέλεσε μέρος των αλγορίθμων *Apriori* και *AprioriTid* [4].

Η βασική του αρχή βασίζεται στην ιδιότητα 7. Σύμφωνα με την ιδιότητα αυτή ο αλγόριθμος *δημιουργεί έναν υποψήφιο εάν και μόνο εάν όλα τα υποσύνολα του έχουν προηγουμένως κριθεί ως frequent*. Συγκεκριμένα ένας *(k+1)* υποψήφιος θα γίνει αποδεκτός εάν και μόνο αν όλα τα *k*-itemsets υποσύνολα του έχουν κριθεί ως frequent.

Προκειμένου να γίνουν περισσότερο κατανοητά τα παραπάνω ας υποθέσουμε ότι βρισκόμαστε στο στάδιο δημιουργίας των *k+1* υποψηφίων. Όπως φαίνεται στον SQL-like αλγόριθμο στο σχήμα 33**,** ο Apriori-gen παίρνει ως είσοδο όλα τα large/ frequent *k*-itemsets L_k και αναζητά ζεύγη από σύνολα στοιχείων τα οποία έχουν κοινά τα *k-1* μικρότερα στοιχεία τους. Παίρνοντας τα *k-1* κοινά στοιχεία σε συνδυασμό με τα δυο υπόλοιπα, τα δυο αυτά σύνολα ενώνονται για την δημιουργία του υποτιθέμενου υποψηφίου. Επανεμφανίσεις στοιχείων αποκλείονται από την απαίτηση το τελευταίο στοιχείο του δεύτερου συνόλου να είναι μεγαλύτερο. Για την ώρα η παρουσία μόνο δυο υποσυνόλων έχει συντελέσει στην δημιουργία του υποτιθέμενου υποψηφίου. Σημειώνουμε ότι μέχρι αυτό το σημείο ο Apriori-gen ακολουθεί παρόμοια στρατηγική δημιουργίας υποψηφίων με αυτήν του SETM.

Η καινοτομία του Apriori-gen έγκειται στο γεγονός ότι ελέγχεται η παρουσία όλων των εναπομεινάντων *k*-υποσυνόλων του υποψηφίου στο δεύτερο μέρος του αλγορίθμου (σχήμα 33).

Πρώτο μέρος

```
    insert into C(k+1)
    select  a.item1, a.item2, …,a.itemk, b.itemk
    from  Lk a, Lk b
    where a.item1 = b. item1,…, a. itemk-1 = b. itemk-1, a. itemk < b.
itemk
```

Δεύτερο μέρος

```
    forall itemset c ∈ Ck+1 do
       forall k-subsets s of c do
              if(s ∉Lk) then
                     delete c from Ck+1
    end
```

Σχήμα 33. *Αλγόριθμος για τη συνάρτηση Apriori-gen*

Η δεύτερη καινοτομία του Apriori-gen και αυτή που εν γένει οδηγεί στο όνομα του, είναι ότι η φάση της δημιουργίας των υποψηφίων γίνεται πριν και χωριστά από την φάση μέτρησης των συχνοτήτων εμφάνισης τους, και ο αλγόριθμος καλείται μια φορά για την δημιουργία των υποψηφίων ενός δοθέντος μεγέθους.

Έτσι οι βελτιώσεις που προκύπτουν από τον Apriori-gen σχετικά με την στρατηγική δημιουργίας υποψηφίων συγκριτικά με τον AIS συνοψίζονται σε δυο σημεία: (1) δημιουργούνται λιγότεροι υποψήφιοι, και (2) δεν δημιουργούνται επαναληπτικά για κάθε transaction αλλά μόνο μια φορά. Όσον αφορά στην σύγκρισή του με τον SETM και αυτός δημιουργεί περισσότερους υποψηφίους και τους επανα-δημιουργεί για κάθε transaction χωριστά.

Σημείωση. Υποθέτουμε ότι τα items σε κάθε transaction κρατούνται

λεξικογραφικά ταξινομημένα. Χρησιμοποιούμε το συμβολισμό `c[1] ·c[2] ·… ·c[k]` για την αναπαράσταση ενός `itemset c`, το οποίο αποτελείται από τα items `c[1] ,c[2] ,… ,c[k]`, όπου `c[1] < c[2] <… < c[k]`. Σε κάθε itemset αντιστοιχούμε ένα μετρητή, στον οποίο αποθηκεύουμε το αντίστοιχο support. Με την δημιουργία ενός itemset ο μετρητής αρχικοποιείται στην τιμή *0*.

Apriori. Το σχήμα 34 δείχνει τον Apriori αλγόριθμο. Στο πρώτο πέρασμα του αλγορίθμου απλά μετριούνται οι εμφανίσεις των στοιχείων για να καθοριστεί ποια από αυτά θα αποτελέσουν τα large *1*-temsets. Κάθε επόμενο πέρασμα, ας πούμε *k*, αποτελείται από δυο φάσεις. Στη πρώτη, τα large L_{k-1} itemsets που βρέθηκαν στο *k-1* πέρασμα χρησιμοποιούνται για να δημιουργήσουν τα υποψήφια itemsets C_k, χρησιμοποιώντας τη συνάρτηση `apriori-gen`, όπως αυτή περιγράφηκε παραπάνω. Στη δεύτερη φάση επανεξετάζεται η βάση, καθορίζεται το support των υποψηφίων και κατ' επέκταση ευρίσκονται ποια itemsets είναι και large.

Για θέματα σχετικά με την ορθότητα, τις χρησιμοποιούμενες δομές και τον τρόπο αποθήκευσης της πληροφορίας παραπέμπουμε τον αναγνώστη στο [4].

```
1. L1 = {large 1-itemsets}
2. for (k = 2; Lk-1  ≠ ∅; k++) do begin
3.      Ck = apriori-like-gen(Lk-1);  // Καινούριου Υποψήφιοι
4.      forall transactions t ∈ D do begin
5.          Ct = subset(Ck,t); // Οι υποψήφιοι που περιέχονται στο t
6.          forall candidates c  ∈ Ct do
7.              c.count++;
8.       end
9.      Lk = { c ε Ck | c.count >= minsup}
10.   end
11. Answer = ∪k Lk;
```

Σχήμα 34. *Ο αλγόριθμος Apriori*

Item-lists. Το βασικό πρόβλημα του Apriori (και του AIS) είναι ότι πρέπει να διαβάζει εξολοκλήρου τη βάση σε κάθε "*πέρασμα*" του, αν και πολλά από τα itmes και τα transactions της βάσης από κάποιο σημείο και μετά δεν έχουν καμία ουσιαστική αξία για τα επόμενα "περάσματα" του αλγορίθμου και μόνο επιβάρυνση σε χρόνο και μνήμη προκαλούν. Πρακτικά και ουσιαστικά items τα οποία δεν είναι frequent/ large και transactions τα οποία έχουν πλήθος στοιχείων μικρότερο από το πλήθος των στοιχείων των τρεχόντων υποψηφίων, δεν είναι αναγκαία και προκαλούν μόνο επιβάρυνση. *Η διαγραφή τους θα βοηθούσε στην αποφυγή περιττού κόστους σε χρόνο και μνήμη*, προκειμένου να μετρηθούν sets τα οποία δεν είναι δυνατόν να είναι υποψήφια.

Ο Apriori δεν περιλαμβάνει τέτοιες βελτιστοποιήσεις και επιπρόσθετα θα πρέπει να σημειώσουμε ότι κάτι τέτοιο θα ήταν ιδιαίτερα δύσκολο να προστεθεί στις δυνατότητες τόσο του Apriori, αλλά και του AIS. Η δυσκολία πηγάζει από την μορφή αναπαράστασης, η οποία χρησιμοποιείται και από τους δυο αλγορίθμους. Όπως απεικονίζεται και φαίνεται στο σχήμα 35, σε αυτήν τη μορφή αναπαράστασης τα transactions είναι αποθηκευμένα σε μια ακολουθία ταξινομημένων στοιχείων υπό την μορφή λίστας. Αυτή η μορφή των item-lists αποτελεί το πλέον χρησιμοποιούμενο format εισόδου και κάνει δύσκολη την διαγραφή και απαλλαγή από τα μη αναγκαία τμήματα της πληροφορίας. Ας υποθέσουμε ότι θέλουμε να διαγράψουμε όλα τα στοιχεία που δεν λαμβάνουν

μέρος σε κανένα από τα frequent/ large sets. Δυστυχώς η γνώση και η πληροφορία για το ποια items θα κρατήσουμε και ποια θα πετάξουμε, είναι διαθέσιμη μόνο αφότου διαβάσουμε μια ακόμα φορά τη βάση και μετρήσουμε τα supports των υποψηφίων. Έτσι μπορούμε να απομακρύνουμε τα περιττά items μόνο σε επόμενο πέρασμα της βάσης, αφού αυτά διαβαστούν για μια ακόμα φορά, χωρίς αυτό να είναι αναγκαίο.

Όπως θα δούμε αργότερα οι άλλες δυο μορφές αναπαράστασης διαγράφουν αυτά τα στοιχεία αμέσως, γεγονός που οδηγεί σε μικρότερα μεγέθη δεδομένων σε αργότερα περάσματα της βάσης. Δυστυχώς αυτό δεν ισχύει όπως θα δούμε για τα πρώτα περάσματα, όπου το μέγεθος των ενδιάμεσων αποτελεσμάτων μπορεί να υπερβεί το αρχικό μέγεθος της βάσης. Το πλεονέκτημα της αναπαράστασης των δεδομένων με τη μορφή των item-lists, είναι ότι το μέγεθος των δεδομένων δεν αυξάνει κατά τη διάρκεια εκτέλεσης του αλγορίθμου και δεν μπορεί να υπερβεί το αρχικό μέγεθος της βάσης.

TID	Items
1	{1,3,4,5}
2	{2,3,4}
3	{1,2,3,4}

Σχήμα 35. *Μορφή αναπαράστασης πληροφορίας με χρήση item-list*

AprioriTid. Το υπαρκτό πρόβλημα με τον Apriori (αδυναμία άμμεσης απαλλαγής από την περιττή, για τα επόμενα βήματα, πληροφορία) οδήγησε τους συγγραφείς του στην δημιουργία του *AprioriTid*, ο οποίος σε αντίθεση με τον Apriori, χρησιμοποιεί όπως θα δούμε μια διαφορετική αναπαράσταση της πληροφορίας από τα item-lists.

Ο AprioriTid μπορεί να θεωρηθεί ότι αποτελεί μια βελτιωμένη έκδοση του SETM, η οποία όμως δεν στηρίζεται σε βασικές λειτουργίες βάσεων δεδομένων και χρησιμοποιεί την `apriori-gen` για ταχύτερη δημιουργία υποψηφίων. Επιπρόσθετα ο AprioriTid διαβάζει τα δεδομένα μια μόνο φορά και προσπαθεί να τα αποθηκεύσει για όλα τα υπόλοιπα περάσματα. Και σε αυτήν την περίπτωση, κάθε επαναληπτική εκτέλεση του αλγορίθμου αποτελείται από την φάση δημιουργίας των υποψηφίων μέσω της `apriori-gen`, ακολουθούμενη από την φάση μέτρησης και τον προσδιορισμό των supports των τρεχόντων (υπό παραγωγή) υποψηφίων.

Το ενδιαφέρον σημείο του αλγορίθμου βρίσκεται όπως είπαμε στο γεγονός ότι η βάση *D* δεν χρησιμοποιείται μετά την πρώτη ανάγνωση της για την μέτρηση των υποψηφίων. Σε αντίθεση το σύνολο $\bar{C}_k$ χρησιμοποιείται γι΄ αυτό το σκοπό. Κάθε μέλος του set $\bar{C}_k$ είναι της μορφής $<TID, \{X_k\}>$, όπου κάθε X_k είναι ένα υποθετικά large itemset, το οποίο εμφανίζεται στο transaction με αναγνωριστικό TID. Αυτή η μορφή αναπαράστασης είναι γνωστή με το όνομα *λίστα υποψηφίων* (candidate-lists). Για $k=1$, το $\bar{C}_1$ ανταποκρίνεται στη βάση *D*, με τη διαφορά ότι κάθε item *i*, έχει αντικατασταθεί από το itemset {i}. Για $k>1$, το $\bar{C}_k$ δημιουργείται από το βήμα 10 του αλγορίθμου (σχήμα 36). Τα μέλη του $\bar{C}_k$ που αντιπροσωπεύουν ένα transaction *t*, είναι: $<t.TID, \{c \in C_k, \text{ με το } c \text{ να περιέχεται στο } t\}>$. Εάν ένα transaction δεν περιέχει κανένα *k*-itemset ως υποψήφιο, τότε το $\bar{C}_k$ δεν θα έχει καμία είσοδο για αυτό το transaction. Έτσι ο αριθμός των εγγραφών στο $\bar{C}_k$ μπορεί να είναι μικρότερος από τον αριθμό των transactions στη βάση, ειδικά για μεγάλες τιμές του k. Επιπρόσθετα, για μεγάλες τιμές του *k*, το μέγεθος κάθε εγγραφής

μπορεί να είναι πολύ μικρότερο από το μέγεθος του transaction που αντιπροσωπεύει. Ωστόσο για μικρές τιμές του *k*, μια εγγραφή στο $\bar{C}_k$ μπορεί να είναι μεγαλύτερη από το μέγεθος του αντίστοιχου transaction, διότι μια εγγραφή στο $\bar{C}_k$ περιλαμβάνει όλους του υποψηφίους μεγέθους *k*, που περιέχονται στο εν λόγω transaction.

```
1.    L1 = {large 1-itemsets}
2.    C̄1 = database D;
3.    for (k = 2; Lk-1 ≠ ∅; k++) do begin
4.       Ck = apriori-like-gen(Lk-1); // New Candidates
5.       C̄k = ∅;
6.       forall entries t ∈ C̄k-1 do begin
7.          // determine candidate itemsets in Ck contained
            // in the transaction with identifier t.TID
            Ct = { c ∈Ck | (c-c[k]) ∈ t.set-ofoitemsets ∧
                 (c-c[k-1] ∈ t.set-of-itemsets};
8.          forall candidates c  ∈ Ct do
9.            c.count++;
10.         if(Ct ≠ ∅) then  C̄k += < t.TID, Ct>;
11.      end
12.      Lk = { c ε Ck | c.count >= minsup}
13.   end
14.   Answer = ∪k Lk;
```

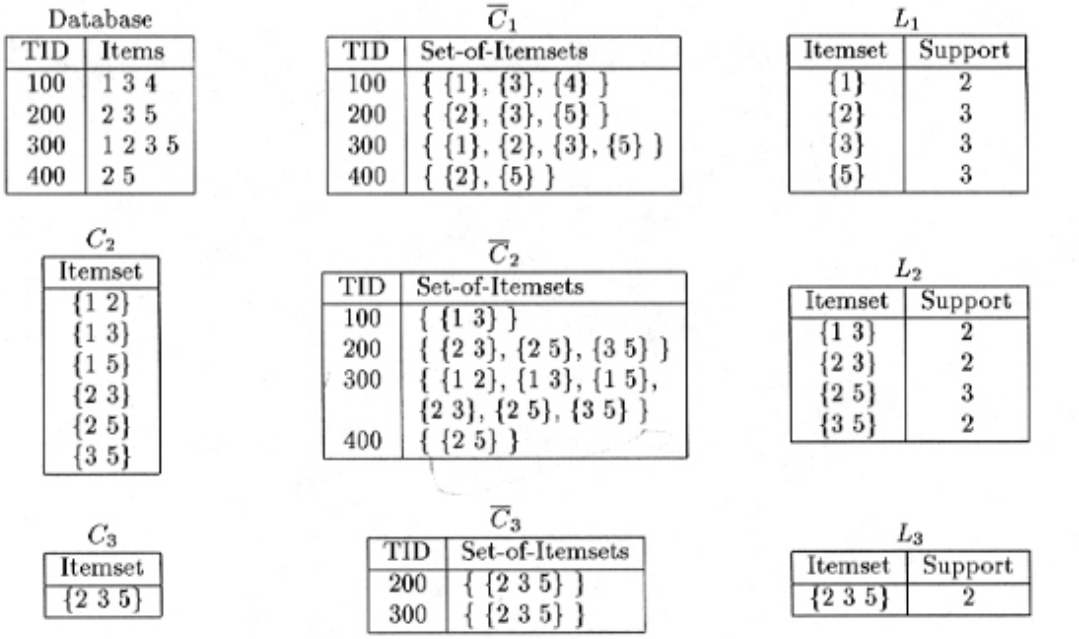
Σχήμα 36. *Ο Αλγόριθμος AprioriTid*

Παράδειγμα

Ας θεωρήσουμε τη βάση του σχήματος 37 και ας υποθέσουμε ότι το ελάχιστο support είναι η εμφάνιση ενός itemset σε 2 transactions. Καλώντας τη συνάρτηση `apriori-gen` με είσοδο το L_1, στο βήμα 4 δημιουργούνται τα υποψήφια itemsets C_2. Στα βήματα 6 έως 10 του αλγορίθμου υπολογίζεται το support των υποψηφίων στο C_2 χρησιμοποιώντας τις εγγραφές στο $\bar{C}_1$ και δημιουργώντας παράλληλα το $\bar{C}_2$. Η πρώτη εγγραφή στο $\bar{C}_1$ είναι η *{{1} {3} {4}}* και αντιπροσωπεύει το transaction με TID 100. Το set C_t στο βήμα 7 που ανταποκρίνεται σε αυτήν την εγγραφή *t* είναι το {{1 3}}, επειδή το {1,3} είναι μέλος του C_2 και τόσο τα *({1 3} – {1})* και *({1 3}-{3}* είναι μέλη του `t.set-of-itemsets`.

Καλώντας την συνάρτηση `apriori-gen`, με το L_2 ως είσοδο, παίρνουμε το C_3. Εξετάζοντας και διαβάζοντας μια ακόμα φορά τα στοιχεία στα $\bar{C}_2$ και C_3, δημιουργούμε το $\bar{C}_3$. Σημειώνουμε ότι δεν υπάρχει είσοδος στο $\bar{C}_3$ για τα transactions με TIDs 100 και 400, καθώς δεν περιέχουν κανένα από τα itemsets στο C_3. Ο υποψήφιος *{2 3 5 }* είναι τελικά το μόνο itemset που διαπιστώνεται ως large και κατ' επέκταση το μοναδικό μέλος του L_3. Όταν δημιουργείται το C_4, από το C_3, διαπιστώνεται ότι είναι άδειο και η διαδικασία σταματά.

Για θέματα σχετικά με την ορθότητα, τις χρησιμοποιούμενες δομές και τον τρόπο αποθήκευσης της πληροφορίας, παραπέμπουμε τον αναγνώστη στο [4].



Σχήμα 37. *Βρίσκοντας Large-Itemests με χρήση του αλγορίθμου AprioriTid*

AprioriHybrid: ***Συνδυασμός των Apriori & AprioriTid.*** Πειράματα που πραγματοποιήθηκαν από τον Agrawal [4] έδειξαν ότι οι Apriori και AprioriTid πραγματοποιούν καλύτερες επιδόσεις από τους AIS και SETM.

Ενδιαφέρουσα όμως είναι η σύγκριση των Apriori και AprioriTid, καθώς και οι δυο δημιουργούν τον ίδιο αριθμό υποψηφίων και διαφέρουν βασικά στην μορφή αναπαράστασης της πληροφορίας. Ενώ ο Apriori αποφεύγει να εναλλάσσει τα δεδομένα μεταξύ κύριας και περιφερειακής μνήμης, δεν απαλλάσσεται όπως δείξαμε από περιττή πληροφορία σε επόμενα περάσματα του στη βάση, με αποτέλεσμα να σπαταλά χρόνο σε ανώφελες προσπάθειες μέτρησης των support των itemsets, χωρίς καμία σημασία. Από την άλλη πλευρά ο AprioriTid, "*πετά*" όπως δείξαμε την αποδεδειγμένα περιττή πλέον πληροφορία και κατά συνέπεια επιτυγχάνει καλύτερους χρόνους σε επόμενα περάσματα. Δυστυχώς, όμως, λόγω της μορφής αναπαράστασης δεδομένων που υιοθετεί (candidate-list) είναι κυρίως αργός στο δεύτερο πέρασμα, και αν η μνήμη δεν επαρκεί για την αποθήκευση των δεδομένων, η εναλλαγή των αποτελεσμάτων με τη περιφερειακή μνήμη κρίνεται αναγκαία, γεγονός που προκαλεί πρόσθετες καθυστερήσεις και μπορεί σε αυτή τη περίπτωση να καταστήσει τον αρχικό αλγόριθμο Apriori ταχύτερο.

Για το λόγο αυτό ένας άλλος αλγόριθμος, ο *AprioriHybrid*, προτείνεται στο [4], ο οποίος χρησιμοποιεί τον Apriori για τα πρώτα περάσματα και εναλλάσσεται με τον AprioriTid, μόλις είναι βέβαιο ότι τα δεδομένα ταιριάζουν πλέον στη μνήμη. Η εναλλαγή αυτή έχει κάποιο χρονικό κόστος για το πέρασμα από τη μια μορφή αναπαράστασης στην άλλη, ωστόσο καλύπτεται σε επόμενα περάσματα. Η υβριδική έκδοση του αλγορίθμου οδηγεί σε βελτιώσεις έναντι του Apriori, όποτε ο AprioriTid μπορεί να χρησιμοποιηθεί αρκετά μετά την εναλλαγή, ώστε να μπορέσει να υπερκαλύψει το πρόσθετο κόστος της προκύπτουσας εναλλαγής στην αναπαράσταση της μορφής.

Κεφάλαιο 7:
Εξόρυξη Κατανεμημένων και Ετερογενών Κλινικών Δεδομένων: Ψάχνοντας για Ενδιαφέροντες Κανόνες Αλληλοσυσχέτισης

Στον προτεινόμενο αλγόριθμο μας υιοθετούμε σε μεγάλο βαθμό τις ιδέες του Apriori και υλοποιούμε έναν ***Apriori-like*** αλγόριθμο κάνοντας διάφορες προσθήκες, βασισμένες σε παρατηρήσεις, επανεξετάσεις και αφαιρέσεις στον αρχικό αλγόριθμο. Ακολουθούμε κάποιες διαφοροποιήσεις στις τεχνικές υλοποίησης και προσπαθούμε να εκμεταλλευτούμε τα πλεονεκτήματα που προκύπτουν από τις *δυναμικές δομές δεδομένων* που χρησιμοποιούμε, καθότι πιστεύουμε ότι αυτή η μορφή αναπαράστασης της πληροφορίας, είναι αυτή που πρόκειται να επικρατήσει στο μέλλον.

Δοθέντος ενός συνόλου *D*, κλινικών/ ιατρικών transactions, το πρόβλημα της εξόρυξης και ανακάλυψης κανόνων συσχέτισης έγκειται και πάλι στην δημιουργία εκείνων των κανόνων, οι οποίοι θα έχουν τιμές support και confidence μεγαλύτερες από κάποιες προκαθορισμένες από τον χρήστη ελάχιστες τιμές, *minimum support* (*minsup*) και *minimum confidence* (*minconf*) αντίστοιχα. Θα πρέπει να σημειώσουμε ότι δεν περιοριζόμαστε από την μορφή της αναπαράστασης του *D*. Για παράδειγμα, το *D* μπορεί να είναι ένα απλό data file ή ένα relational table (σχεσιακή βάση δεδομένων). Ωστόσο στην δική μας προσέγγιση (επεξεργασία XML αρχείων), η αναπαράσταση και επεξεργασία της πληροφορίας γίνεται εξολοκλήρου με τη χρήση δυναμικών δομών δεδομένων, όπου το κάθε transaction όπως θα περιγράψουμε, έχει τη μορφή δυναμικής αλυσίδας κόμβων πληροφορίας, και όπου ο κάθε κόμβος αντιστοιχείται σε ένα item.

Ο αλγόριθμος μας σε πλήρη αναλογία με τους Apriori και AprioriTid, δημιουργεί υποψήφια itemsets, των οποίων πρόκειται να υπολογιστεί το support σε ένα πέρασμα, χρησιμοποιώντας μόνο τα large itemsets που βρέθηκαν στο προηγούμενο βήμα και αδιαφορώντας για τα transactions της βάσης. Θα πρέπει να τονίσουμε ότι κάθε υποσύνολο ενός large itemset πρέπει να είναι και αυτό large. Έτσι τα υποψήφια itemsets τα οποία θα αποτελούνται από k items, θα δημιουργούνται μόνο από την ένωση διαπιστωμένων large *k-1*-itemsets.

Ο αλγόριθμος μας σε αντιστοιχία με τον AprioriTid έχει την επιπρόσθετη δυνατότητα, ως επέκταση και βελτίωση του Apriori, ότι η αρχική βάση δεδομένων δεν χρησιμοποιείται καθόλου για την μέτρηση των υποψηφίων itemsets μετά το πρώτο πέρασμα, όπως θα δείξουμε παρακάτω.

Δυναμικά διατηρούμε σε κάθε βήμα, μόνο το μέρος της βάσης που είναι αναγκαίο και προφανώς, κάθε φορά είναι γνήσιο υποσύνολο της, σε αντίθεση με τον AprioriTid ο οποίος χρησιμοποιεί κωδικοποίηση της βάσης που στα πρώτα βήματα μπορεί να υπερβεί το αρχικό της μέγεθος.

7.1. Παραδοχές, Παραλληλισμοί, και Συσχετίσεις Εννοιών KDD στο Ιατρικό Πεδίο Εφαρμογής

Στο σημείο αυτό θα πρέπει να γίνει κατανοητή η ερμηνεία που δίνουμε στους όρους item και transaction, καθώς επίσης ο τρόπος υιοθέτησής και αντιστοίχησης τους στη κλινική πρακτική, όπου εργαζόμαστε.

- Ένα transaction στην περίπτωση μας αντιστοιχείται με μια *επίσκεψη* (visit) ενός ασθενή σε κάποιο πληροφοριακό σύστημα (Κέντρο Υγείας). Ένα transaction χαρακτηρίζεται από τις εξής ιδιότητες (attributes ή features): `Patient_Id`, `Information_System`, `Visit_Id`, `Date`.
- Ένα item αντίστοιχα θεωρείται ως μια *εξέταση* (`Atomic Observation`), όπως αυτή περιγράφηκε στο COAS μοντέλο. Περιγράφεται συνοπτικά και παραστατικά από την τριάδα της μορφής `<Atomic_Observation, value, interval >` όπου:
  - Το `Atomic_Observation` περιγράφηκε αναλυτικά (κεφάλαιο 3) στο COAS μοντέλο (π.χ. *Χοληστερίνη*)
  - `Value` είναι η αντίστοιχη τιμή για το `AtomicObservation` στην συγκεκριμένη κάθε φορά εξέταση του ασθενή (π.χ τιμή για *Χοληστερίνη '251'*)
  - `Interval` είναι το πεδίο τιμών του, δηλαδή το διάστημα τιμών όπου η συγκεκριμένη τιμή ανήκει (π.χ. *'Interval-2':[120:200]='Normal'*; πιο αναλυτική περιγραφή παρακάτω).

7.2. Διακριτοποίηση Αριθμητικών Χαρακτηριστικών

Όπως εύκολα μπορεί να γίνει κατανοητό, από τον παραπάνω ορισμό του item στο ιατρικό πεδίο εφαρμογής, το `AtomicObservation` που χαρακτηρίζει το εκάστοτε item δεν είναι εν γένει ένα Boolean χαρακτηριστικό. Κατ' επέκταση και σύμφωνα με το [39] απαιτείται μια αντιστοίχηση του προβλήματος μας, στο πρόβλημα εξαγωγής κανόνων συσχέτισης από '*Boolean'* χαρακτηριστικά.

Συνοπτικά, αναφέρουμε ότι αυτό απαιτεί καταρχήν το διαχωρισμό του πεδίου τιμών του κάθε item, σε ένα σύνολο διαστημάτων και στη συνέχεια την αντιστοίχηση του κάθε `<attribute, interval>` σε ένα Boolean attribute [39]. Στην γενική περίπτωση στην οποία τα ποσοτικά χαρακτηριστικά έχουν λίγες αποδεκτές τιμές, η αντιστοίχηση είναι απλή. Αντί να έχουμε ένα χαρακτηριστικό (π.χ ένα πεδίο στον πίνακα μας), έχουμε τόσα Boolean χαρακτηριστικά όσες και οι επιτρεπτές τιμές του. Έτσι η τιμή του αντίστοιχου στο `<attribute1, value1>` Boolean χαρακτηριστικού, θα είναι '*1'* αν το `attribute1` έχει τιμή '*value1'* στην αρχική εγγραφή. Στην περίπτωση όπου ένα χαρακτηριστικό αντιστοιχίζεται σε ένα μεγάλο πεδίο τιμών, τότε απαιτείται η διάτμηση και αντιστοίχηση ενός διαστήματος τιμών, `<attribute, interval>` σε ένα Boolean χαρακτηριστικό. Ωστόσο, υπάρχουν δυο προβλήματα με αυτήν τη προσέγγιση.

1. *Υπολογισμός MinSup*. Εάν ο αριθμός των διαστημάτων που θα προκύψουν είναι *μεγάλος*, το support του κάθε διαστήματος που θα προκύψει μπορεί να είναι αρκετά χαμηλό. Έτσι αν δεν χρησιμοποιηθούν μεγαλύτερα διαστήματα, κάποιοι από τους κανόνες που αφορούν τα χαρακτηριστικά αυτά θα χαθούν, λόγο έλλειψης *ελάχιστου support*!
2. *Υπολογισμός MinConf*. Πάντα υπάρχει απώλεια πληροφορίας όταν χωρίζουμε τιμές σε διαστήματα. Κάποιοι κανόνες μπορεί να έχουν ελάχιστο confidence μόνο εάν το item του *ηγούμενου* μέρους ('`IF`' σκέλος του κανόνα) αποτελείται από μία μόνο τιμή, ή ένα μικρό διάστημα τιμών. Έτσι αν μεγαλώσουμε-διευρύνουμε αυτό το διάστημα τιμών μπορεί να πάψει να ισχύει η αναγκαία συνθήκη για το *minimum confidence*!

Στη δικιά μας περίπτωση χωρίζουμε το πεδίο τιμών του κάθε `AtomicObservation` σε **τρία** διαστήματα τιμών. Και αυτό γιατί στην ιατρική υπάρχουν πάντα μετρήσεις που ανταποκρίνονται σε τιμές *Μικρότερες του Κανονικού* (**`Low`**), *Κανονικές/ Φυσιολογικές τιμές* (**`Normal`**) και *Τιμές Μεγαλύτερες από το Φυσιολογικό* (**`High`**). Αυτά τα διαστήματα μπορούν να εξαχθούν είτε κατόπιν συμβουλής γιατρού, είτε από παγκόσμια αποδεκτούς και διεθνείς ιατρικούς οδηγούς και πρωτόκολλα. Σε περιπτώσεις όπου για κάποιες εξετάσεις δεν μπορεί να ευρεθεί αντίστοιχη πληροφορία, μπορούμε να αντιμετωπίσουμε το πρόβλημα αλγοριθμικά και να προχωρήσουμε σε αυτόματη *διακριτοποίηση* (*discretization*) των τιμών των αντίστοιχων χαρακτηριστικών σε τρία διαστήματα, λαμβάνοντας υπόψη τις ελάχιστες και μέγιστες τιμές εμφάνισης τους στο δείγμα που εξετάζουμε.

Παράδειγμα

Στο σημείο αυτό μπορούμε να παρουσιάσουμε ένα απλό παράδειγμα για να γίνουν κατανοητά τόσο τα παραπάνω, όσο και ο τρόπος με τον οποίο γίνεται και νοείται η κωδικοποίηση αλλά και η ταξινόμηση των δημιουργούμενων items στα transactions που περιέχονται.

Έστω λοιπόν ότι έχουμε τα `AtomicObservations`: *`Λευκά Αιμοσφαίρια`*, *`Ερυθρά Αιμοσφαίρια`*, *`Σάχαρο`*. Αφού έχουμε τρία `AtomicObservations` και για καθένα από αυτά υπάρχουν τρία διαστήματα τιμών, και όπως αναφέραμε παραπάνω χρειαζόμαστε 9 συνεχόμενους αριθμούς για την κωδικοποίηση και αντιστοίχηση του κάθε `AtomicObservation` με το αντίστοιχο διάστημα τιμών του, σε ένα κωδικοποιημένο Boolean χαρακτηριστικό. Στην αρίθμηση που προκύπτει δίνουμε διαδοχικούς αύξοντες αριθμούς, όπου ανά τρεις αντιπροσωπεύουν και ένα `AtomicObservation` με τα τρία διαστήματα τιμών του (`Low`, `Normal`, `High`). Έτσι στο εξής, όταν αναφερόμαστε σε ταξινομημένα items μέσα σε ένα transaction, αναφερόμαστε στην αριθμητική τους διάταξη με βάση την παραπάνω κωδικοποίηση. Στο τέλος των διεργασιών μας επανερχόμαστε στην αρχική μορφή και ονοματολογία από όπου και ξεκινήσαμε, χρησιμοποιώντας αντίστοιχα μια διαδικασία αποκωδικοποίησης.

Πίνακας 3. *Παράδειγμα Διακριτοποίησης Αριθμητικών Ιδιοτήτων*

Attribute - Interval	**Boolean attribute**
`Λευκά Αιμοσφαίρια [low]`	‘1’
`Λευκά Αιμοσφαίρια [normal]`	‘2’
`Λευκά Αιμοσφαίρια [high]`	‘3’
`Ερυθρά Αιμοσφαίρια[low]`	‘4’
`Ερυθρά Αιμοσφαίρια[normal]`	‘5’
`Ερυθρά Αιμοσφαίρια[high]`	‘6’
`Σάκχαρο[low]`	‘7’
`Σάκχαρο[normal]`	‘8’
`Σάκχαρο[high]`	‘9’

Η εμφάνιση δηλαδή του `AtomicObservation`: *`Ερυθρά Αιμοσφαίρια`* και ο έλεγχος - διαπίστωση ότι η τιμή του ανήκει στο κανονικό διάστημα τιμών, ισοδυναμεί με την εμφάνιση του item “*5*” στην κωδικοποιημένη βάση μας.

7.3. Ο Αλγόριθμος AprioriXML. Ένας Apriori-like ARM Αλγόριθμος

Το σχήμα 38 δείχνει τον Apriori-like αλγόριθμο μας. Το όνομα του, ***AprioriXML*** προέκυψε επειδή εφαρμόστηκε σε XML αρχεία. Στο πρώτο πέρασμα ο αλγόριθμος απλά μετρά τις εμφανίσεις του κάθε item για να αποφασίσει εάν ανήκει στα large 1-itemsets. Κάθε επόμενο βήμα, ας πούμε *k*, αποτελείται από *τέσσερις φάσεις*.

---

1. **Εύρεση/ Παραγωγή Υποψηφίων.** Τα large itemsets L_{k-1} που βρέθηκαν στο *k-1* βήμα του αλγορίθμου, χρησιμοποιούνται για τη δημιουργία των C_k itemsets υποψηφίων.
2. **Τροποποίηση της Βάσης Δεδομένων.** Κάνοντας χρήση της γνώσης που έχουμε αποκτήσει, για τα ήδη υπάρχοντα large L_{k-1} itemsets, τροποποιούμε τη βάση μας δυναμικά, *αποκόβοντας από αυτήν πληροφορία που ξέρουμε ότι είναι περιττή για το επόμενο βήμα*.
3. **Φάση Μέτρησης.** Σε αυτήν μετράμε όπως θα δούμε τα support όλων των C_k itemsets υποψηφίων, που βρέθηκαν στην πρώτη φάση. Η μέτρηση λαμβάνει όμως χώρα στη βάση που έχει προκύψει από την δεύτερη φάση και είναι απαλλαγμένη από περιττή πληροφορία.
4. **Έλεγχος Αποδοχής Itemsets.** Σε κάθε βήμα, γίνεται ο *έλεγχος* για το ποια από τα υποψήφια μέχρι τώρα itemsets είναι όντως και large ενώ παράλληλα γίνεται η *προετοιμασία* για το ποια items θα παραμείνουν στη βάση και ποια όχι στο επόμενο βήμα του αλγορίθμου.

```
1.  L1 = {large 1-itemsets};
2.  for (k = 2; Lk-1  ≠ ∅; k++)
    {
3.        Ck = apriori-like-gen(Lk-1); // phase 1
4.        New_D = fix_update_old_D(Lk-1); // phase 2
5.        ∀ c ε Ck  // phase 3
          {
6.          ∀ remain&&update transaction t ε New_D
              {
7.            find_itemset = look_for(c, t)
8.            if(find_itemset = = 1)
9.                  c.count ++;
              }
          }
10. Lk = { c ε Ck | c.count >= minsup} // phase 4
    }
11. Answer = ∪k Lk;
```

Σχήμα 38. *Ο AprioriXML Αλγόριθμος*

7.3.1. Ανάλυση Φάσεων και Βημάτων του AprioriXML

- **Δημιουργία Υποψηφίων.** Στην πρώτη φάση (βήμα 3 του αλγορίθμου, σχήμα 38), τα large itemsets L_{k-1} που βρέθηκαν στο *k-1* βήμα του αλγορίθμου, χρησιμοποιούνται για τη δημιουργία των C_k itemsets υποψηφίων. Στο βήμα αυτό χρησιμοποιείται και υλοποιείται η συνάρτηση *apriori-gen* των Apriori-AprioriTid (αποδεδειγμένα η καλύτερη, βλ. κεφάλαιο 6) η οποία παίρνει ως όρισμα το σύνολο των large itemsets L_{k-1} και επιστρέφει ένα υπερσύνολο των large *k*-itemsets. Η συνάρτηση λειτουργεί και υλοποιείται ως εξής:

Σε μια βοηθητική δομή δεδομένων (μορφή *αλυσίδας*), τοποθετούμε κάθε δημιουργούμενο υποψήφιο *k*-itemset, ο οποίος προκύπτει καταρχήν σύμφωνα με την διαδικασία που περιγράφεται στο σχήμα 39.

```
insert into C(k+1)
select  a.item1, a.item2, …,a. itemk, b.itemk
from  Lk a, Lk b
where a.item1 = b.item1,…, a. itemk-1 = b.itemk-1, a. itemk < b. itemk
```

Σχήμα 39. *Πρώτο μέρος της συνάρτησης Apriori-gen*

Λέμε 'καταρχήν' γιατί στην συνέχεια ακολουθεί ένα δεύτερο μέρος (prune step), στο οποίο *διαγράφονται εκείνοι οι υποψήφιοι για τους οποίους έστω και ένα υποσύνολό τους δεν κρίθηκε large, στο προηγούμενο βήμα του αλγορίθμου* (σχήμα 40).

```
Για όλα τα  itemset c ∈ Ck+1
{
       Για όλα τα k-υποσύνολα  s του c
       {
           Εάν (s ∉Lk)
           {
              Διέγραψε το c από το Ck+1
           }
       }
}
```

Σχήμα 40. *Δεύτερο μέρος της συνάρτησης* `apriori-gen`

Σημείωση. Η συνθήκη $\texttt{a.item}_k < \texttt{b.item}_k$ (σχήμα 39) απλά εξασφαλίζει ότι δεν θα δημιουργούνται επαναλαμβανόμενοι υποψήφιοι.

- **Τροποποίηση Βάσης: Απαλλαγή Απο Περιττή Πληροφορία.** Στη δεύτερη φάση (βήμα 4 του αλγορίθμου, σχήμα 38), κάνοντας χρήση της γνώσης που ήδη έχουμε αποκτήσει για τα υπάρχοντα large L_{k-1} itemsets, τροποποιούμε τη βάση μας *δυναμικά*, αποκόβοντας από αυτήν πληροφορία που ξέρουμε ότι είναι περιττή για το επόμενο βήμα της μέτρησης των τρεχόντων υποψηφίων.

  Την εργασία αυτή εκτελεί η συνάρτηση `fix_update_old_D`. Διατρέχει όλα τα εναπομείναντα transactions και κάθε κόμβο(item) αυτών. Για κάθε ένα από αυτά τα items, ελέγχει την αντίστοιχη τιμή του στον βοηθητικό πίνακα `help_support`, όπως αυτός περιγράφεται αναλυτικά στην τέταρτη φάση ελέγχου. Εάν στο *k* βήμα του αλγορίθμου βρεθεί ότι η τιμή για τον εκάστοτε κόμβο που ελέγχεται, στον πίνακα `help_support` ισούται με '*k-1*', γεγονός που σημαίνει ότι το συγκεκριμένο item, ανήκει σε τουλάχιστον ένα large *(k-1)-itemset*, τότε το συγκεκριμένο item παραμένει στην τρέχουσα δυναμική βάση διότι κρίνεται αναγκαίο. Σε αντίθετη περίπτωση διαγράφεται, καθότι είναι περιττό, και αν μείνει θα καθυστερήσει τη φάση της μέτρησης στο επόμενο βήμα, αφού θα χρειαστεί να γίνουν πολλαπλές συγκρίσεις με αυτό (μπορεί και με όλα τα items του κάθε υποψηφίου *k*-itmeset), που δεν θα έχουν κανένα νόημα.

  Έτσι, από πολύ νωρίς, η βάση (στη πραγματικότητα η δομή αποθήκευσης δεδομένων) μας *ελευθερώνεται* από items που σίγουρα δεν θα παίξουν κανένα ρόλο στη συνέχεια. Όχι μόνο items, αλλά και ολόκληρα transactions μπορεί σύντομα να διαγραφούν από τη βάση μας, χωρίς να επιβαρύνουν την περαιτέρω

επεξεργασία. Έτσι μπορεί να ξεκινήσουμε με 10, για παράδειγμα, transactions όπου καθ' ένα περιέχει 10 items και σε πολύ μικρό αριθμό βημάτων να έχουν μείνει μόλις 5 transactions με διαφορετικό αριθμό items για το καθένα και μέγιστο αριθμό items πολύ μικρότερο του 10! Με αυτόν τον τρόπο στην επόμενη φάση, που είναι η φάση μέτρησης των υποψήφιων itemsets, δεν θα χρειαστεί να κάνουμε περιττές συγκρίσεις, γεγονός που οδηγεί αφενός σε *βελτιστοποίηση σε χρόνο*, αφετέρου *ελευθερώνει χώρο* από την κύρια μνήμη, καθώς η νέα βάση γίνεται ολοένα και μικρότερη σε μέγεθος.

- **Φάση Μέτρησης των Υποψήφιων Large Itemsets.** Η τρίτη φάση (βήματα 5 - 9 του αλγορίθμου, σχήμα 38) όπως ήδη αναφέραμε ονομάζεται *Φάση Μέτρησης*. Σε αυτήν μετράμε τα support όλων των C_k υποψηφίων itemsets, που βρέθηκαν στην πρώτη φάση. Αυτό γίνεται ελέγχοντας την ύπαρξη ή μη των υποψηφίων itemsets, στη βάση που έχει προκύψει από τη δεύτερη φάση, η οποία όπως δείξαμε είναι απελευθερωμένη από όλα τα items που δεν ανήκουν στο σύνολο των διαπιστωμένων *large-(k-1)*-itemsets και κατά συνέπεια από ολόκληρα transactions που γνωρίζουμε ότι στο προηγούμενο βήμα δεν περιείχαν και δεν υποστήριζαν κανένα large itemset.

  Αυτός είναι και ο λόγος που στον ψευδοκώδικα αναφέρουμε '`remain&&update`' transaction, καθότι εν γένει σε κάθε βήμα αλλάζει και είτε απαλλάσσεται από περιττή πληροφορία, είτε διαγράφεται τελείως. Η συνάρτηση `look_for`, η οποία χρησιμοποιείται σε αυτή τη φάση, στο βήμα *k*, παίρνει ως είσοδο ένα κάθε φορά *k*-itemset (από την αλυσίδα που είναι αποθηκευμένα όλα τα υποψήφια *k*-itemsets), το αποκωδικοποιεί και βρίσκει τα επιμέρους items που το αποτελούν (πλήθους *k*).

  Στη συνέχεια τα αναζητεί σε κάθε transaction με έξυπνο και γρήγορο τρόπο, επωφελούμενη της ταξινόμησης ως εξής: τα αναζητά ένα-ένα ξεκινώντας από το λεξικογραφικά μικρότερο. Με δεδομένο ότι και τα items στο κάθε transaction είναι ταξινομημένα, εάν δεν βρει το πρώτο σταματά και επιστρέφει *0*, γεγονός που σημαίνει ότι το συγκεκριμένο transaction δεν υποστηρίζει τον τρέχον υποψήφιο. Σ' αυτήν την περίπτωση δεν έχει νόημα να αναζητήσει τα υπόλοιπα *k-1* items του itemset. Εάν το πρώτο item βρεθεί, συνεχίζεται η αναζήτηση για το δεύτερο, από το σημείο όπου βρέθηκε το πρώτο item και μετά καθώς, λόγω ταξινόμησης, δεν μπορεί να βρίσκεται σε προηγούμενη θέση. Το γεγονός αυτό, όπως είναι κατανοητό, επιταχύνει αρκετά τη διαδικασία της μέτρησης, καθότι δεν γίνονται περιττές συγκρίσεις. Η διαδικασία συνεχίζεται με ανάλογο ρυθμό και η συνάρτηση επιστρέφει *1*, μόνο εάν όλα τα επιμέρους items βρεθούν στο τρέχον transaction. Σε αυτήν την περίπτωση αυξάνεται και ο αντίστοιχος για το εν-λόγω υποψήφιο itemset μετρητής `c.count`.

  Η διαδικασία αυτή επαναλαμβάνεται για όλα τα εναπομείναντα transactions και στο τέλος της έχει υπολογιστεί το ζητούμενο support, όπου και καταχωρείται στο αντίστοιχο πεδίο του κάθε υποψηφίου και θα χρειαστεί όπως θα δείξουμε στο επόμενο βήμα.

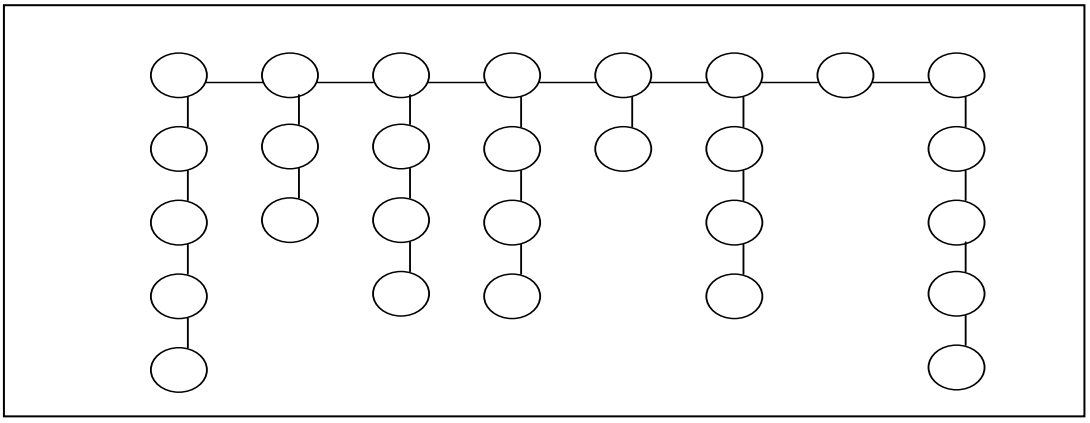
- **Φάση Ελέγχου και Προετοιμασίας της Βάσης.** Στην τέταρτη και τελευταία φάση (βήμα 10 του αλγορίθμου, σχήμα 38) του τρέχοντος κάθε φορά βήματος, γίνεται ο έλεγχος για το ποιά από τα υποψήφια μέχρι τώρα itemsets είναι όντως και large. Αυτό επιτυγχάνεται με την βοήθεια του τρίτου βήματος καθώς είναι πλέον γνωστή η συχνότητα εμφάνισης του κάθε itemset. Ελέγχοντας εάν αυτή ξεπερνά το προκαθορισμένο όριο *minsup*, αποφαινόμαστε για το εάν ένα υποψήφιο itemset, είναι ή όχι large.

Σε αυτήν την περίπτωση συντελείται και μια πρόσθετη λειτουργία *προετοιμασίας* για την διαμόρφωση της βάσης στο επόμενο βήμα του αλγορίθμου. Στο σημείο αυτό πρέπει να ορίσουμε τον βοηθητικό πίνακα `help_support`, μέσω του οποίου επιτυγχάνεται η συγκεκριμένη λειτουργία. Ο `help_support` είναι ένας πίνακας ακεραίων ο οποίος έχει τόσες θέσεις, όσες και τα παρατηρούμενα στην βάση μας διαφορετικά items (μετά τη διακριτοποίηση). Κάθε θέση του αντιστοιχεί σε ένα `Atomic Observation` και σε ένα εύρος τιμών που αυτό αντιπροσωπεύει, σύμφωνα με την κωδικοποίηση που έχει γίνει και περιγραφεί παραπάνω.

Εάν κάποιο itemset τηρεί το ζητούμενο κριτήριο και κριθεί ως large, τότε κάθε item που το αποτελεί χρειάζεται να συμμετέχει και να παραμείνει στη βάση και για το επόμενο βήμα. Για το λόγο αυτό, επιμέρους εξειδικευμένες συναρτήσεις χωρίζουν το κάθε αποδεδειγμένο large itemset στα επιμέρους items που το αποτελούν και το συνθέτουν, και θέτουν την αντίστοιχη θέση του `help_support` για το καθένα από αυτά ίση με το k (τρέχον βήμα). Έτσι οι θέσεις του `help_support`, που έχουν τεθεί ίσες με k φανερώνουν για τα αντίστοιχα items που αντιπροσωπεύουν ότι: αφενός μετέχουν στα δημιουργούμενα large k-itemsets, και αφετέρου είναι αυτά που πρέπει να παραμείνουν, καθότι θα αποτελέσουν τη βάση για το επόμενο βήμα του αλγορίθμου.

7.3.2. Χρησιμοποιούμενες Δομές

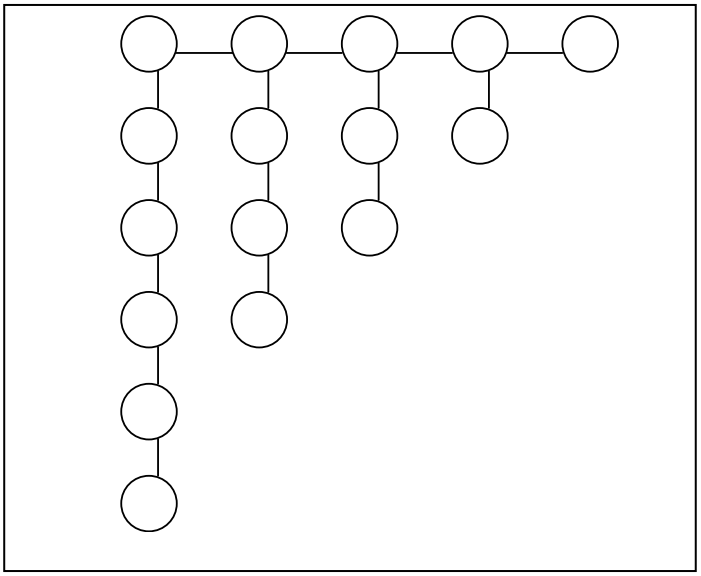
Καταρχήν, για την αναπαράσταση της βάσης χρησιμοποιούμε την μορφή του σχήματος 41. Όπως βλέπουμε έχουμε μια *αλυσίδα από αλυσίδες*. Κάθε transaction αντιστοιχίζεται σε μια κατακόρυφη αλυσίδα, ενώ όλες αυτές οι επιμέρους αλυσίδες ενώνονται μέσω της οριζόντιας αλυσίδας και δημιουργούν τη δομή όπου αναπαριστούμε την βάση μας. Κάθε κόμβος της δομής μας αναπαριστά και ένα item, όπως αυτό έχει οριστεί σε προηγούμενες ενότητες. Η βάση αυτή, όπως περιγράψαμε, τροποποιείται σε κάθε βήμα, καθώς απαλλάσσεται από κόμβους (items) και ολόκληρες αλυσίδες (transactions).



Σχήμα 41. *Η δομή 'αλυσίδας' / 'δάσους' αναπαράστασης δεδομένων*

Για τα large itemsets χρησιμοποιούμε μια συνδεδεμένη αλυσίδα από αλυσίδες, σχηματίζοντας μία δομή '*δάσους*'. Η μορφή της φαίνεται στο σχήμα 42. Η πρώτη από τα αριστερά κατακόρυφη αλυσίδα αντιπροσωπεύει τα L_1-itemsets, η δεύτερη τα L_2-itemsets και ούτω καθεξής. Εύκολα μπορεί να παρατηρήσει κανείς ότι μπορούμε να έχουμε το πολύ `max_number_size` από αλυσίδες, όπου

`max_number_size` είναι ο μέγιστος αριθμός από items που αποτελούν κάποιο transaction στην αρχική αναπαράσταση της βάσης. Οι κατακόρυφες αλυσίδες από αριστερά προς τα δεξιά θα περιέχουν, συνήθως, συνεχώς μειούμενο αριθμό κόμβων, όπως είναι φυσικό. Κάθε κόμβος της προκύπτουσας δομής θα περιέχει στοιχεία πληροφορίας της μορφής: *κωδικός αναπαράστασης* του αντίστοιχου itemset και το αντίστοιχο support. Τη δομή αυτή θα διατρέξουμε στη συνέχεια κατά τη διαδικασία της εύρεσης και ανακάλυψης των κανόνων συσχέτισης και κάθε κόμβος της όπως θα δούμε θα αποτελέσει "*πηγή*" εξαγωγής πιθανών κανόνων.



Σχήμα 42: *Δομή κράτησης των large Itemsets*

Για τα υποψήφια itemsets διατηρούμε μια απλή αλυσίδα όπως ήδη αναφέραμε, με ανάλογη μορφή με την αλυσίδα όπου κρατούνται τα large itemsets. Στο εκάστοτε *k* βήμα του αλγορίθμου ένα υποσύνολο αυτής θα αποτελεί όπως δείξαμε και την *k*-οστή κατακόρυφη αλυσίδα στη δομή των large-itemsets.

7.4. Ανακάλυψη/ Δημιουργία Κανόνων Συσχέτισης

Προκειμένου να δημιουργήσουμε κανόνες συσχέτισης ανάμεσα στα items της βάσης μας και έχοντας ακολουθήσει την προηγούμενη διαδικασία ανακάλυψης όλων των large itemsets (όπως αυτή περιγράφηκε στα προηγούμενα κεφάλαια-υποκεφάλαια), συνεχίζουμε ως εξής:

1. Για κάθε large itemset *l*, βρίσκουμε όλα τα μη κενά υποσύνολά του. Για κάθε τέτοιο υποσύνολο *α*, εξάγουμε ένα κανόνα της μορφής *α => (l - α)*, εάν ο λόγος του *support(l)* προς το *support(α)* είναι τουλάχιστον ίσος με το προκαθορισμένο από το χρήστη όριο *minconf*. Εν γένει θεωρούμε όλα τα δυνατά υποσύνολα του *l* και δημιουργούμε με αυτόν τον τρόπο κανόνες με περισσότερους από έναν ακόλουθους όρους.
2. Στη παρούσα μεταπτυχιακή εργασία έχει υλοποιηθεί ο αλγόριθμος για την κατασκευή όλων των κανόνων συσχέτισης με έναν μόνο ακόλουθο όρο (σχήματα 43, 44). Με ανάλογο τρόπο, γενικεύοντας τον ίδιο αλγόριθμο και κάνοντας χρήση μιας συνθήκης (την οποία θα αναφέρουμε και θα αναλύσουμε παρακάτω), μπορούμε πολύ εύκολα να εξάγουμε όλους τους κανόνες με περισσότερους από έναν ακόλουθους όρους (σχήμα 45).

7.4.1. Ο Αλγόριθμος Εξαγωγής Κανόνων Συσχέτισης με Έναν Ακόλουθο Όρο

Έχοντας βρει και αποθηκεύσει όλα τα large itemsets όπως δείξαμε (στη δομή του σχήματος 42), προχωράμε όπως φαίνεται το σχήμα 43. Διατρέχουμε όλες τις προκύπτουσες αλυσίδες που περιέχουν large itemsets μεγεθών *2* έως *k*. Για κάθε μια από αυτές επισκεπτόμαστε όλους τους κόμβους της και αναζητούμε εκμαίευση κανόνων από κάθε κόμβο χωριστά με χρήση της συνάρτησης `check_if_gen_rules`.

```
for(k=2;  Lk != NULL;  k++)
{
  ∀lk ε Lk
   {
     check_if_gen_rules(lk);
   }
}
```

Σχήμα 43. *Ο Ψευδοκώδικας Ανεύρεσης Κανόνων Συσχέτισης*

Η συνάρτηση `check_if_gen_rules` (σχήμα 44) αποφασίζει εάν ο εκάστοτε κόμβος δύναται να δημιουργήσει κανόνες και ποιους, με χρήση της συνάρτησης `create_rule_from`. Πιο συγκεκριμένα η `check_if_gen_rules` παίρνει ως όρισμα ένα large *k*-itemset. Μέσω της συνάρτησης `fix_without_ i` δημιουργεί τα *k* δυνατά *(k-1)* υποσύνολα του (αφαιρώντας το *ι-στό* κάθε φορά στοιχείο). Αναζητά καθένα από αυτά στην L_{k-1} αλυσίδα και υπολογίζει το confidence του κανόνα με αυτό ως ηγούμενο μέρος, γνωρίζοντας το support του. Εάν ικανοποιείται η συνθήκη *confidence ≥ minconf* καλείται η `create_rule_from` και δημιουργεί το κανόνα με το *ι-στο* item ως ακόλουθο όρο.

```
    procedure check_if_gen_rules (lk: large k-itemset )

1.  ∀ item i ε lk
2.  antecedent   =   fix_without_i();       /*   Δημιουργία   itemset
    antecedent(size = k-1)      από το lk, χωρίς το ι_οστό item */
3.  find  =  find_Litemset(antecedent);  /* Αναζήτηση και εύρεση του
    δημιουργούμενου κ-1 itemset στην δομή όπου κρατούνται τα large itemsets
    και συγκεκριμένα στην αλυσίδα όπου κρατούνται τα κ-1 large itemsets */
4.    support(antecedent) = find ->support;
5.    threshold = support(lk) / support(antecedent);
6.  if (threshold  ≥ minconf)
       create__rule_from(lk, antecedent); /* Δημιουργία και εμφάνιση
του παραγόμενου κανόνα με το itemset antecedent ως πρώτο όρο και τον(lk-
antecedent) ως ακόλουθο. */
```

Σχήμα 44. *Αλγόριθμος Παραγωγής Κανόνων Με Έναν Ακόλουθο Όρο*

7.4.2. Αλγόριθμος Για Εξαγωγή Κανόνων Με Πολλαπλούς Ακόλουθους Όρους

Όπως αναφέραμε, πολύ εύκολα μπορούμε να γενικεύσουμε τον παραπάνω αλγόριθμο ώστε το ακόλουθο μέρος του κανόνα να περιλαμβάνει περισσότερους από έναν όρους. Για παράδειγμα δοθέντος του itemset *ABCD*, θεωρούμε πρώτα το υποσύνολο *ABC*, κατόπιν το *AB* κ.τ.λ.

Συνθήκη. Στην περίπτωση όπου ένα υποσύνολο α ενός large itemset *l* δεν παράγει κάποιο κανόνα, τότε δεν είναι ανάγκη να εξετάσουμε κανένα από τα υποσύνολα του α, για το αν παράγουν κάποιο κανόνα με τη χρήση του *l*.

Έτσι εάν ο κανόνας *ABC* => D δεν έχει αρκετά υψηλό confidence, δεν είναι ανάγκη να ελέγξουμε εάν ο κανόνας *AB => CD* μπορεί να σχηματιστεί (ιδιότητα 7, κεφάλαιο 6). Με αυτόν τον τρόπο δεν υπάρχει περίπτωση να χάσουμε κάποιο κανόνα, επειδή το support κάθε υποσυνόλου (*subα*) του *α*, πρέπει να είναι τουλάχιστον όσο το support του α. Έτσι το confidence του κανόνα *subα => (l – subα)* δεν μπορεί να είναι περισσότερο από το confidence του *α => (l – α)*. Συμπεραίνουμε λοιπό ότι: εάν το α δεν παράγει ένα κανόνα που να αφορά όλα τα items του *l*, με το α ως πρώτο μέρος του (antecedent), τότε ούτε το subα πρόκειται να παράγει κάποιον κανόνα.

Ο παρακάτω αναδρομικός αλγόριθμος(σχήμα 45) υλοποιεί τα παραπάνω και επεκτείνει τον προηγούμενο αλγόριθμο.

```
    forall large itemsets l_k, k>=2 do
         call genrules(l_k,l_k);

    procedure genrules(l_k :large k-itemsets, a_m : large m-itemsets)
1.  A = {(m-1) - itemsets a_m-1 | a_m-1  ⊂ a_m };
2.  forall a_m-1  ε A do begin
3.     conf = support(lk)/support(a_m-1);
4.     if(conf>=minconf) then begin
          Output the rule a_m-1 => (l_k - a_m-1), with confidence = conf and
support = support(k);
5.        if(m-1 > 1) then
6.          call genrules(l_k, a_m-1);
7.     end
8.  end
```

Σχήμα 45. *Ο Αλγόριθμος Ανακάλυψης Κανόνων Συσχέτισης Με Πολλαπλούς Ακόλουθους Όρους*

Κεφάλαιο 8:
Παράδειγμα Εφαρμογής Αλγορίθμου με Χρήση Πραγματικών Ιατρικών Δεδομένων

Στο κεφάλαιο αυτό περιγράφουμε κάποια παραδείγματα με αποτελέσματα από κάποιες εκτελέσεις του αλγορίθμου μας σε πραγματικά ιατρικά δεδομένα. Παρουσιάζουμε τρία συνολικά παραδείγματα. Το πρώτο σκοπό έχει να δείξει αναλυτικά τον τρόπο εκτέλεσης του αλγορίθμου, γι' αυτό και παρουσιάζεται σε μεγάλη λεπτομέρεια. Στα δυο επόμενα παραδείγματα και έχοντας κατανοήσει πλήρως τον τρόπο εκτέλεσης του αλγορίθμου, παρουσιάζουμε τα αποτελέσματα από κάποιους κανόνες συσχέτισης, τα οποία εκμαιεύουμε από παραδείγματα χρήσης ***βιοχημικών*** και ***αιματολογικών*** ιατρικών εξετάσεων.

8.1 Πρώτο Παράδειγμα – Αναλυτική Επεξήγηση του AprioriXML

Στο παράδειγμα αυτό προκειμένου να γίνουν κατανοητές τόσο η ευστάθεια όσο και η εκτέλεση του αλγορίθμου, παρουσιάζουμε αναλυτικά τα δεδομένα μας, αλλά και κάποια βήματα που περιγράφηκαν στο κεφάλαιο 7. Το παράδειγμα αντιπροσωπεύει 10 *τυχαία επιλεγμένες* επισκέψεις ασθενών στο πληροφοριακό σύστημα του Σπηλίου, όπου έχουν πραγματοποιηθεί βιοχημικές εξετάσεις. Προσπαθούμε να δούμε πως σχετίζονται 10 από τα `Atomic Observations` που υπάγονται σε αυτήν την σύνθετη εξέταση, αναζητώντας αντίστοιχους κανόνες.

Σημείωση-1. Κρατάμε 10 transactions για να μπορεί να παρατηρήσει εύκολα ο αναγνώστης τις % αναλογίες.

Σημείωση-2. Για συντομία χώρου δεν παρουσιάζουμε το αντίστοιχο με την DTD μας XML αρχείο εισόδου (το οποίο έχει *δυναμικά δημιουργηθεί* από την επερώτησή μας), αλλά στον παρακάτω πίνακα 4 παρουσιάζουμε ακριβώς την αντίστοιχη πληροφορία για να μπορεί ο αναγνώστης μας να παρακολουθήσει άνετα το πείραμα μας.

Πίνακας 4. *Αναλυτική έκθεση δεδομένων εισόδου*

Composite Observation	Atomic Observation		Value	Composite Observation	Atomic Observation		Value
BIOCHEM EXAM(1)	ΣΑΚΧΑΡΟ	(1)	112.000	BIOCHEM EXAM(6)	ΣΑΚΧΑΡΟ	(1)	82.000
	ΟΥΡΙΑ	(2)	25.000		ΟΥΡΙΑ	(2)	28.000
	ΚΡΕΑΤΙΝΙΝΗ	(3)	0.770		ΚΡΕΑΤΙΝΙΝΗ	(3)	0.770
	ΧΟΛΗΣΤΕΡΙΝΗ	(4)	355.000		ΝΑ	(4)	141.000
	HDL-ΧΟΛΗΣΤΕΡΙΝΗ	(5)	90.500		ΧΟΛΗΣΤΕΡΙΝΗ	(5)	277.000
	ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(6)	145.000		HDL-ΧΟΛΗΣΤΕΡΙΝΗ	(6)	"Not Asked"
BIOCHEM EXAM(2)	ΣΑΚΧΑΡΟ	(1)	143.000		ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(7)	221.000
	ΟΥΡΙΑ	(2)	43.000		ΟΥΡΙΚΟ ΟΞΥ	(8)	3.500
	ΚΡΕΑΤΙΝΙΝΗ	(3)	0.900		ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ	(9)	"Not Asked"
	ΧΟΛΗΣΤΕΡΙΝΗ	(4)	209.000		ΣΙΔΗΡΟΣ	(10)	84.000
	HDL-ΧΟΛΗΣΤΕΡΙΝΗ	(5)	48.900	BIOCHEM EXAM(7)	ΣΑΚΧΑΡΟ	(1)	82.000
	ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(6)	441.000		ΟΥΡΙΑ	(2)	22.000
	ΟΥΡΙΚΟ ΟΞΥ	(7)	3.400		ΚΡΕΑΤΙΝΙΝΗ	(3)	0.970
BIOCHEM EXAM(3)	ΣΑΚΧΑΡΟ	(1)	83.000		ΝΑ	(4)	"Not Asked"
	ΟΥΡΙΑ	(2)	17.000		ΧΟΛΗΣΤΕΡΙΝΗ	(5)	154.000
	ΚΡΕΑΤΙΝΙΝΗ	(3)	0.450		HDL-ΧΟΛΗΣΤΕΡΙΝΗ	(6)	74.800
	ΝΑ	(4)	"Not Asked"		ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(7)	33.000
	ΧΟΛΗΣΤΕΡΙΝΗ	(5)	136.000		ΟΥΡΙΚΟ ΟΞΥ	(8)	2.500
	HDL-ΧΟΛΗΣΤΕΡΙΝΗ	(6)	"Not Asked"		ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ	(9)	105.000
	ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(7)	34.000	BIOCHEM EXAM(8)	ΣΑΚΧΑΡΟ	(1)	93.000
	ΟΥΡΙΚΟ ΟΞΥ	(8)	2.100		ΟΥΡΙΑ	(2)	29.000
BIOCHEM EXAM(4)	ΣΑΚΧΑΡΟ	(1)	282.000		ΚΡΕΑΤΙΝΙΝΗ	(3)	"Not Asked"
	ΟΥΡΙΑ	(2)	66.000		ΝΑ	(4)	"Not Asked"
	ΚΡΕΑΤΙΝΙΝΗ	(3)	1.300		ΧΟΛΗΣΤΕΡΙΝΗ	(5)	212.000
	ΝΑ	(4)	137.000		HDL-ΧΟΛΗΣΤΕΡΙΝΗ	(6)	87.100
	ΧΟΛΗΣΤΕΡΙΝΗ	(5)	233.000		ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(7)	70.000
	ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(6)	565.000		ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ	(8)	94.000

	ΟΥΡΙΚΟ ΟΞΥ (7)	6.500	BIOCHEM EXAM(9)	ΣΑΚΧΑΡΟ	(1)	184.000
	ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ (8)	97.000		ΟΥΡΙΑ	(2)	31.000
BIOCHEM EXAM(5)	ΣΑΚΧΑΡΟ (1)	101.000		ΚΡΕΑΤΙΝΙΝΗ	(3)	0.970
	ΟΥΡΙΑ (2)	20.000		ΝΑ	(4)	"Not Asked"
	ΚΡΕΑΤΙΝΙΝΗ (3)	0.900		ΧΟΛΗΣΤΕΡΙΝΗ	(5)	194.000
	ΝΑ (4)	148.000		HDL-ΧΟΛΗΣΤΕΡΙΝΗ	(6)	"Not Asked"
	ΧΟΛΗΣΤΕΡΙΝΗ (5)	159.000		ΤΡΙΓΛΥΚΕΡΙΔΙΑ	(7)	129.000
	HDL-ΧΟΛΗΣΤΕΡΙΝΗ (6)	35.900		ΟΥΡΙΚΟ ΟΞΥ	(8)	4.100
	ΤΡΙΓΛΥΚΕΡΙΔΙΑ (7)	67.000	BIOCHEM EXAM(10)	ΣΑΚΧΑΡΟ	(1)	222.000
	ΟΥΡΙΚΟ ΟΞΥ (8)	6.000		ΟΥΡΙΑ	(2)	"Not Asked"
	ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ (9)	162.000				

Έχοντας διαβάσει όλα τα δεδομένα, ο αλγόριθμος τα κωδικοποιεί παρέχοντας τους συνεχόμενους αύξοντες αριθμούς. Δεσμεύει τρεις αριθμούς για κάθε `Atomic Observation Type` και τους μοιράζει στα αντίστοιχα `Low`, `Normal` και `High` διαστήματα, λαμβάνοντας υπόψη τις *min* και *max* αντίστοιχα παρατηρούμενες τιμές. Έτσι προκύπτει ο παρακάτω πίνακας.

Πίνακας 5. *Αποτελέσματα επεξεργασίας και κωδικοποίησης*

Atomic Observation Type	Min Value	Max Value	Code For Low Values	Code For Normal Values	Code For Hight Values
ΣΑΚΧΑΡΟ	82.000000	282.000000	1	2	3
ΟΥΡΙΑ	17.000000	66.000000	4	5	6
ΚΡΕΑΤΙΝΙΝΗ	0.450000	1.300000	7	8	9
ΧΟΛΗΣΤΕΡΙΝΗ	136.000000	355.000000	10	11	12
HDL-ΧΟΛΗΣΤΕΡΙΝΗ	35.900000	90.500000	13	14	15
ΤΡΙΓΛΥΚΕΡΙΔΙΑ	33.000000	565.000000	16	17	18
ΟΥΡΙΚΟ ΟΞΥ	2.100000	6.500000	19	20	21
ΝΑ	137.000000	148.000	22	23	24
ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ	94.000000	162.000000	25	26	27
ΣΙΔΗΡΟΣ	0.000000	84.000000	28	29	30

Στο πίνακα 5α, παρουσιάζουμε την κωδικοποιημένη μορφή της αρχικής βάσης δεδομένων. Έχει προκύψει από ταυτόχρονη κωδικοποίηση και ταξινόμηση όλων των items. Στη συνέχεια ο αλγόριθμος εκτελεί δυναμικά πάνω της όλες τις λειτουργίες που έχουν περιγραφεί στο κεφάλαιο 7. Έτσι στο δεύτερο κιόλας βήμα μετά την δημιουργία του L_1, η βάση μας έχει πάρει την μορφή του πίνακα 5β. Βλέπουμε πόσο έχει απελευθερωθεί από περιττή πληροφορία, απαλλαγμένη από items και transactions. Σημειώνουμε ότι αμέσως διαγράφεται το 5^o transactions με 8 items! Έτσι οι *2*-itemsets υποψήφιοι θα αναζητηθούν στην βάση του πίνακα 5β και όχι σε αυτή του πίνακα 5α; με προφανή οφέλη. Χάριν συντομίας δεν παρουσιάζουμε αναλυτικά τα επόμενα βήματα, σημειώνουμε δε ότι δημιουργούνται δεκαέξι *3*-itemsets υποψήφιοι, και ένας *4*-itemset υποψήφιος ο οποίος τυγχάνει να είναι και large.

Πίνακας 5α. *Αρχική Βάση Δεδομένων* **Πίνακας 5β.** *Η Βάση μετά το 2^o πέρασμα*

	Original	After 2nd Scan	Status
1	< 1 , 4, 8, 12, 15, 16>	< 1 , 4, 8, 16>	***Reduced***
2	< 3 , 5>	----	*Eliminated*
3	< 1 , 5, 8, 10, 13, 18, 19>	< 1 , 8, 10, 19>	*Reduced*
4	< 1 , 4, 7, 10, 14, 16, 19, 23>	< 1 , 4, 10, 16, 19, 23>	*Reduced*
5	< 3 , 6, 9, 11, 18, 21, 22, 25>	----	*Eliminated*
6	< 1 , 4, 8, 10, 13, 16, 21, 24, 27>	< 1 , 4, 8, 10, 16>	*Reduced*
7	< 1 , 4, 8, 11, 14, 17,19, 23, 26, 30>	< 1 , 4, 8, 19, 23>	*Reduced*
8	< 1 , 4, 8, 10, 15, 16, 19, 23, 25>	< 1 , 4, 8, 10, 16, 19, 23>	*Reduced*
9	< 1 , 4, 8, 11, 15, 16, 23, 25>	< 1 , 4, 8, 16, 23>	*Reduced*
10	< 2 , 4, 8, 10, 14, 16, 20, 23>	< 4 , 8, 10, 16, 23>	*Reduced*

Οι κανόνες προφανώς προκύπτουν (χρησιμοποιώντας το παραπάνω σχήμα), σε κωδικοποιημένη μορφή. Αφού τους αποκωδικοποιήσουμε τους παρουσιάζουμε στον επόμενο πίνακα.

Πίνακας 6. *Οι κανόνες συσχέτισης που βρέθηκαν για το πρώτο παράδειγμα*

	Support %	**Confidence** %
ΟΥΡΙΑ[LOW] => ΣΑΚΧΑΡΟ[LOW]	60	85.7
ΣΑΚΧΑΡΟ[LOW] => ΟΥΡΙΑ[LOW]	60	85.7
ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΣΑΚΧΑΡΟ[LOW]	60	85.7
ΣΑΚΧΑΡΟ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	60	85.7
ΧΟΛΗΣΤΕΡΙΝΗ[LOW] => ΣΑΚΧΑΡΟ[LOW]	40	80
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΣΑΚΧΑΡΟ[LOW]	50	83.3
ΣΑΚΧΑΡΟ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	50	71.4
ΟΥΡΙΚΟ ΟΞΥ[LOW] => ΣΑΚΧΑΡΟ[LOW]	40	100
ΝΑ[NORMAL] => ΣΑΚΧΑΡΟ[LOW]	40	80
ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΟΥΡΙΑ[LOW]	60	85.7
ΟΥΡΙΑ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	60	85.7
ΧΟΛΗΣΤΕΡΙΝΗ[LOW] => ΟΥΡΙΑ[LOW]	40	80
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΟΥΡΙΑ[LOW]	60	100
ΟΥΡΙΑ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	60	85.7
ΝΑ[NORMAL] => ΟΥΡΙΑ[LOW]	50	100
ΟΥΡΙΑ[LOW] => ΝΑ[NORMAL]	50	71.4
ΧΟΛΗΣΤΕΡΙΝΗ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	40	80
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	50	83.3
ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	50	71.4
ΝΑ[NORMAL] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	40	80
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΧΟΛΗΣΤΕΡΙΝΗ[LOW]	40	66.7
ΧΟΛΗΣΤΕΡΙΝΗ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	40	80
ΝΑ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	40	80
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΝΑ[NORMAL]	40	66.7
ΟΥΡΙΑ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΣΑΚΧΑΡΟ[LOW]	50	83.3
ΣΑΚΧΑΡΟ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΟΥΡΙΑ[LOW]	50	83.3
ΣΑΚΧΑΡΟ[LOW] ΟΥΡΙΑ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	50	83.3
ΟΥΡΙΑ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΣΑΚΧΑΡΟ[LOW]	50	83.3
ΣΑΚΧΑΡΟ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΟΥΡΙΑ[LOW]	50	100
ΣΑΚΧΑΡΟ[LOW] ΟΥΡΙΑ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	50	83.3
ΟΥΡΙΑ[LOW] ΝΑ[NORMAL] => ΣΑΚΧΑΡΟ[LOW]	40	80
ΣΑΚΧΑΡΟ[LOW] ΝΑ[NORMAL] => ΟΥΡΙΑ[LOW]	40	100
ΣΑΚΧΑΡΟ[LOW] ΟΥΡΙΑ[LOW] => ΝΑ[NORMAL]	40	66.7
ΚΡΕΑΤΙΝΙΝΗ[NORMAL] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΣΑΚΧΑΡΟ[LOW]	40	80
ΣΑΚΧΑΡΟ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	40	80
ΣΑΚΧΑΡΟ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	40	66.7
ΚΡΕΑΤΙΝΙΝΗ[NORMAL] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΟΥΡΙΑ[LOW]	50	100
ΟΥΡΙΑ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	50	83.3
ΟΥΡΙΑ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	50	83.3
ΚΡΕΑΤΙΝΙΝΗ[NORMAL] ΝΑ[NORMAL] => ΟΥΡΙΑ[LOW]	40	100
ΟΥΡΙΑ[LOW] ΝΑ[NORMAL] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	40	80
ΟΥΡΙΑ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΝΑ[NORMAL]	40	66.7
ΧΟΛΗΣΤΕΡΙΝΗ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΟΥΡΙΑ[LOW]	40	100
ΟΥΡΙΑ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΧΟΛΗΣΤΕΡΙΝΗ[LOW]	40	66.7
ΟΥΡΙΑ[LOW] ΧΟΛΗΣΤΕΡΙΝΗ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	40	100
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] ΝΑ[NORMAL] => ΟΥΡΙΑ[LOW]	40	100
ΟΥΡΙΑ[LOW] ΝΑ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	40	80
ΟΥΡΙΑ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΝΑ[NORMAL]	40	66.7
ΟΥΡΙΑ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΣΑΚΧΑΡΟ[LOW]	40	80
ΣΑΚΧΑΡΟ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΟΥΡΙΑ[LOW]	40	100
ΣΑΚΧΑΡΟ[LOW] ΟΥΡΙΑ[LOW] ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] => ΚΡΕΑΤΙΝΙΝΗ[NORMAL]	40	80
ΣΑΚΧΑΡΟ[LOW] ΟΥΡΙΑ[LOW] ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	40	80

Στο σημείο αυτό θα παρουσιάσουμε δύο ολοκληρωμένα παραδείγματα, παρουσιάζοντας αποτελέσματα και σχόλια του αλγορίθμου μας, χρησιμοποιώντας *πραγματικά κλινικά δεδομένα* τα οποία είναι αποθηκευμένα στις βάσεις μας. Θα χρησιμοποιήσουμε τις *βιοχημικές* και *αιματολογικές* εξετάσεις ως πηγές εξαγωγής συμπερασμάτων και παραγωγής κανόνων.

Σημείωση. Δεν κατέστη δυνατή η χρήση άλλων σύνθετων (`Composite Observations`) ιατρικών εξετάσεων (π.χ καρδιολογικές, γυναικολογικές, κλινικές, αξονικές κ.τ.λ) ως παραδείγματα εφαρμογών μας, καθότι είτε δεν υπήρχαν στις βάσεις μας πραγματικά στοιχεία γι' αυτές , είτε δεν υπήρχαν καθόλου τιμές για τα αντίστοιχα πεδία των επιμέρους εξετάσεων τους (`Atomic Observations`).

8.2. Δεύτερο Παράδειγμα: Ενδιαφέροντες Αλληλοσυσχετίσεις Ευρημάτων Αιματολογικών Εξετάσεων

Το παράδειγμα που ακολουθεί χρησιμοποιεί ως δεδομένα, μετρήσεις και αποτελέσματα από *αιματολογικές* εξετάσεις που έγιναν σε *άνδρες* ασθενείς, στο πληροφοριακό σύστημα του *Σπηλίου*.

Παρατήρηση. Λαμβάνοντας την απάντηση από την παραπάνω ερώτηση, διαπιστώνουμε όπως θα δούμε ότι οι περισσότερες από τις τιμές των `Atomic Observations`, τα οποία συνθέτουν την αιματολογική εξέταση που μελετάμε, έχουν τιμή *"Not Asked"*. Είτε λοιπόν δεν έχει γίνει, είτε δεν έχει καταγραφεί τιμή για τις περισσότερες μετρήσεις των επιμέρους εξετάσεων. Σε πολλές από αυτές (τις περισσότερες) δεν έχει καταγραφεί ούτε μια φορά κάποια τιμή τους; σε ελάχιστες υπάρχει σχεδόν πάντα καταγραμμένη τιμή μέτρησης; ενώ στις υπόλοιπες (αρκετές) η τιμή μέτρησης της εκάστοτε υπό-εξέτασης έχει καταγραφεί στις λιγότερες από τις μισές συνολικές ανακληθείσες απαντήσεις.

Σημείωση. Το γεγονός αυτό όπως είναι λογικό, δημιουργεί και παρουσιάζει εξωγενή προβλήματα στη χρήση του αλγορίθμου. Προκειμένου λοιπόν οι δυσκολίες αυτές να γίνουν κατανοητές και να ληφθούν υπόψη στην αξιολόγηση των αποτελεσμάτων, κάνουμε μια σύνθετη παρουσίαση τους για να βοηθήσουμε στην βέλτιστη κατανόηση εξαγωγής τους.

Σημαντικό Σχόλιο. Ιδιαίτερη σημασία, λόγω της φύσης των δεδομένων, έχει η απόφαση που θα πάρουμε σχετικά με την αντιμετώπιση των περιπτώσεων/ εξετάσεων όπου δεν έχει καταγραφεί η τιμή μέτρησης τους. Δυο είναι, όπως έχουμε αναφέρει, οι δυνατοί τρόποι αντιμετώπισης και οι αντίστοιχες παραδοχές.
Παραδοχή-1: Είτε θεωρούμε ότι οι τιμές αυτές ανήκουν στα φυσιολογικά όρια μέτρησης
Παραδοχή-2: Είτε δεν τις λαμβάνουμε καθόλου υπόψη μας στην όλη ARM διαδικασία

Η πρώτη παραδοχή κρίνεται περισσότερο σωστή όταν ένα μόνο μέρος/ ποσοστό από τις συνολικές μετρήσεις μιας εξέτασης, δεν έχει τιμή. Τότε με μεγαλύτερη πιθανότητα μπορούμε να υποθέσουμε ότι η συγκεκριμένη τιμή, αντιστοιχεί στα φυσιολογικά όρια.

Στην περίπτωση όμως όπου μια εξέταση δεν παρουσιάζεται ποτέ με κάποια τιμή μέτρησης, ίσως θα πρέπει να μην ληφθεί καθόλου υπόψη, καθότι δεν παρέχει κάποια ουσιαστική πληροφορία.

Γνωρίζοντας και εκθέτοντας τα παραπάνω είμαστε σε θέση να παρουσιάσουμε τα αποτελέσματα εκτέλεσης του αλγορίθμου μας.

8.2.1. Αποτελέσματα

Λαμβάνοντας την απάντηση στην ερώτηση μας παρατηρούμε ότι τις συνθήκες της ερώτησης, ικανοποιούν *36 άνδρες ασθενείς* από το πληροφοριακό σύστημα του *Σπηλίου* με *42* καταγραμμένες *επισκέψεις* (έχουν αφαιρεθεί αυτές όπου δεν υπήρχε καμία μέτρηση με οποιαδήποτε μορφή). Λόγο έλλειψης χώρου είναι αδύνατο να παρουσιάσουμε το παράδειγμα στην λεπτομέρεια του. Θα παρουσιάσουμε όμως όλα τα αναγκαία στοιχεία ώστε να είναι δυνατή η κατανόηση του.

Από το σύνολο το επεξεργασμένων στοιχείων, διαπιστώνουμε ότι έχουμε καταρχήν *41 διαφορετικά* `Atomic Observations` (πίνακας 7), τα οποία και συνθέτουν στο σύνολο της την *Αιματολογική Εξέταση*, έτσι όπως αυτή έχει

καταγραφεί στο συγκεκριμένο πληροφοριακό σύστημα. Από τις 41 αυτές υπό-εξετάσεις- όπως διαπιστώνουμε τρέχοντας τον αλγόριθμο, οι 30 από αυτές δεν εμφανίζουν σε καμία από τις 42 εξεταζόμενες επισκέψεις ποτέ κάποια τιμή μέτρησης (*"Not Asked"*). Δεν έχει λοιπόν νόημα σε αυτό το σημείο να παρουσιάσουμε χωρίς καμία επεξεργασία (το επιβάλλει άλλωστε και η KDD διαδικασία) τους εξαγόμενους κανόνες, πολύ περισσότερο αν θεωρούμε φυσιολογικές τις τιμές που λείπουν, καθότι θα λάβουμε ένα τεράστιο αριθμό κανόνων οι οποίοι θα περιλαμβάνουν συσχετίσεις υπό-εξετάσεων με φυσιολογικές τιμές, και οι οποίες στην ουσία θα είναι *μη μετρήσιμες*!

Πίνακας 7. *Όλες οι παρατηρούμενες υπό-εξετάσεις*

WBC	LYM	GRAN	MID	RBC	HGB
HCT	MCV	MCH	MCHC	RDM	ΑΙΜΟΠΕΤΑΛΙΑ
PCT	MPV	PDV	ΛΕΜΦΟ	ΜΟΝΟ	ΙΩΣΙΝ
ΒΑΣΕΟΦ	ΧΡΟΝΟΣ ΡΟΗΣ	ΡΑΒΔΟ	ΜΕΤΑΜΥΕΛ	ΜΥΕΛΟΚΥΤ	ΠΡΟΜΥΕΛ
ΒΛΑΣΤΕΣ	Α-ΛΕΜΦΟ	ΑΤΥΠΑ	ΑΝΙΣΟΚΥΤΤΑΡΩΣΗ	ΥΠΟΧΡΩΜΙΑ	ΠΟΙΚΙΛΟΚΥΤΤΑΡΩΣΗ
ΣΤΟΧΟΚΥΤΤΑΡΩΣΗ	ΣΦΑΙΡΟΚΥΤΤΑΡΩΣΗ	ΒΑΣΕΟΦΙΛΗ ΣΤΙΞΗ	ΜΙΚΡΟΚΥΤΤΑΡΩΣΗ	ΠΟΛΥΧΡΩΜΑΤΟ	ΕΛΛΕΙΠΤΟΚΥΤΤΑΡΩΣΗ
ΤΚΕ1ΩΡΑ	ΤΚΕ2ΩΡΑ	ΔΕΚ	ΠΑΡΑΤΗΡΗΣΕΙΣ	ΧΡΟΝΟΣ ΠΗΞΗΣ	

Για το λόγο αυτό αφαιρούμε καταρχήν εκείνες τις επισκέψεις (encounters) όπου όλες οι υπό-εξετάσεις τους έχουν τιμή "*Not Asked*". Στη συνέχεια κρατάμε από το σύνολο των `Atomic Observations` (κατά κύριο λόγο) εκείνα που εμφανίζουν τουλάχιστον μια φορά κάποια τιμή μέτρησης (πίνακας 8). Έχουμε απομακρύνει δηλαδή *27* από τα *30* `Atomic Observations` που δεν έχουν λάβει ποτέ καμία τιμή. Αφήσαμε εσκεμμένα τρία τυχαία από αυτά για να δούμε πως επηρεάζουν το αποτέλεσμα του αλγορίθμου (πίνακας 9).

Πίνακας 8. *Όλες οι υπό-εξετάσεις με παρατηρούμενες αριθμητικές μετρήσεις*

WBC	LYM	GRAN	RBC	HGB	HCT
MCV	MCH	MCHC	ΑΙΜΟΠΕΤΑΛΙΑ	ΛΕΜΦΟ	

Πίνακας 9: *Οι 3 εναπομείναντες εξετάσεις με μοναδική τιμή "Not Asked"*

PCT	MPV	PDV			

Περίπτωση Α. Στην περίπτωση αυτή *τρέχουμε* τον αλγόριθμο, με τις παραπάνω παραδοχές, χωρίς να λαμβάνουμε υπόψη τις τιμές "*Not Asked*" (επεξεργαζόμαστε δηλαδή την τιμή μιας υπό-εξέτασης μόνο αν είναι διάφορη από "*Not Asked*", στην αντίθετη περίπτωση την αγνοούμε). Χάριν συντομίας στον πίνακα 10 παρουσιάζονται τα αποτελέσματα κανόνων, που σε αυτούς μετέχουν τέσσερις όροι. Έχουν παραλειφθεί όλα τα δυνατά υποσύνολα αυτών, τα οποία φυσικά τηρούν τα ζητούμενα όρια. Από τις εναπομείνασες *31* τελικές επισκέψεις ζητάμε την εξαγωγή κανόνων συσχέτισης με παρουσία τουλάχιστο *11* εμφανίσεων (*support* = 11/31 = 35.5%) και υψηλό *confidence 80%*. Έχουμε φροντίσει η διαδικασία παραγωγής να φτάνει έως και την δημιουργία *4*-large-itemsets. Ενδεικτικά αναφέρουμε για τα μεγέθη των δημιουργούμενων υποψηφίων στα επιμέρους βήματα του αλγορίθμου*:* $|C_3| = 13$, $|C_4| = 1$.

Πίνακας 10. *Οι 4 από τους παραγόμενους κανόνες*

	Support %	**Confidence** %
GRAN[LOW] HGB[HIGHT] HCT[HIGHT] => WBC[LOW]	35.5	91.7
WBC[LOW] HGB[HIGHT] HCT[HIGHT] => GRAN[LOW]	35.5	100
WBC[LOW] GRAN[LOW] HCT[HIGHT] => HGB[HIGHT]	35.5.	100
WBC[LOW] GRAN[LOW] HGB[HIGHT] => HCT[HIGHT]	35.5	91.7

Περίπτωση Β. Στην περίπτωση αυτή τρέχουμε τον αλγόριθμο, με την διαφορά ότι ο αλγόριθμος μεταχειρίζεται και θεωρεί τις τιμές των μετρήσεων για τα `Atomic Observations` που έχουν καταγραφεί ως "*Not Asked*", ως *φυσιολογικές*. Και πάλι αναζητούμε κανόνες με *confidence 80%,* αλλά με λίγο μεγαλύτερο support (support = 14/31~45%). Η επιλογή αυτή έχει γίνει σκόπιμα για να δείξει ότι ενώ οι παραγόμενοι κανόνες θα έπρεπε να είναι υποσύνολο των προηγουμένων (ίδιο πλήθος παρατηρούμενων `Atomic Observations`), είναι πολύ περισσότεροι και σημειώνουμε ότι η διαδικασία παραγωγής φτάνει έως και την δημιουργία *7-large*-itemsets. Στον πίνακα 11, παρατηρούμε κάποιους από τους παραγόμενους κανόνες (δείχνουμε κάποιους ενδεικτικούς κανόνες με συμμετοχή 2-7 όρων), όπου και βλέπουμε την *επίδραση* και (αναμενόμενη) παρουσία των τριών `Atomic Observations` που παρουσιάζονται με τιμές "*Not Asked*" (στους παραγόμενους κανόνες), αλλά και την επικράτηση των φυσιολογικών τιμών, γεγονός που έχει να κάνει με την παρουσία πολλών "*Not Asked*" τιμών και στις υπόλοιπες υπό-εξετάσεις. Ενδεικτικά αναφέρουμε για τα μεγέθη των δημιουργούμενων υποψηφίων στα επιμέρους βήματα του αλγορίθμου: $|C_3| = 110$, $|C_4| = 123$, $|C5| = 83$, $|C_6| = 31$ *και* $|C_7| = 5$.

Πίνακας 11. *Ένας μέρος των παραγόμενων κανόνων με συμμετοχή των "Not Asked" τιμών*

	Support %	**Confidence** %
GRAN[LOW] => WBC[LOW]	74.2	82.1
RBC[NORMAL] => GRAN[LOW]	77.4	92.3
WBC[LOW] RBC[NORMAL] => GRAN[LOW]	61.3	95.0
PCT[NORMAL] MPV[NORMAL] => PDV[NORMAL]	100	100
WBC[LOW] MPV[NORMAL] ΛΕΜΦΟ[NORMAL] => RBC[NORMAL]	64.5	83.3
WBC[LOW] PCT[NORMAL] PDV[NORMAL] => MPV[NORMAL]	77.4	100
GRAN[LOW] PCT[NORMAL] MPV[NORMAL]ΛΕΜΦΟ[NORMAL] => WBC[LOW]	74.2	88.5
GRAN[LOW] RBC[NORMAL] PCT[NORMAL] MPV[NORMAL] ΛΕΜΦΟ[NORMAL] => WBC[LOW]	61.3	86.4
RBC[NORMAL] MCV[NORMAL] MCH[NORMAL] PCT[NORMAL] MPV[NORMAL] PDV[NORMAL] => ΛΕΜΦΟ[NORMAL]	45.2	100

8.3. Τρίτο Παράδειγμα: Ενδιαφέρουσες Αλληλοσυσχετίσεις Ευρημάτων Βιοχημικών Εξετάσεων

Το παράδειγμα που ακολουθεί χρησιμοποιεί ως δεδομένα μετρήσεις και αποτελέσματα από *βιοχημικές* εξετάσεις που έγιναν σε *άνδρες* και *γυναίκες* ασθενείς, στα πληροφοριακά συστήματα του *Σπηλίου* και των *Ανωγείων*.

Σημείωση. Στο πείραμα που παρουσιάζουμε επεξεργαζόμαστε τα αποτελέσματα από *41* συνολικά ασθενείς και *42 επισκέψεις*. Δυστυχώς δεν υπήρχαν καταγραμμένες, πολλαπλές επισκέψεις ασθενών, χωρίς αυτό βέβαια να προκαλεί πρόβλημα στην εκτέλεση του αλγορίθμου, αφού η όλη διαδικασία είναι ανεξάρτητη από την ταυτότητα του ασθενή. Ωστόσο αυτό δεν μας επιτρέπει την

επίδειξη *στατιστικών αποτελεσμάτων*, κάτι που συνέβη και στα προηγούμενα παραδείγματα.

Παρατήρηση. Όπως και στο δεύτερο παράδειγμα, έτσι και τώρα θα χρειαστεί να κάνουμε μια σειρά από *παραδοχές*, γεγονός που επιβάλει *η φύση των καταγραμμένων πραγματικών στοιχείων*, που υπάρχουν στα πληροφοριακά μας συστήματα (Σπήλι, Ανώγεια). Και σ' αυτήν την περίπτωση, το συντριπτικά μεγαλύτερο μέρος των `Atomic Observations` (υπό-εξετάσεις), που αποτελούν την βιοχημική εξέταση (`Composite Observation`) που μελετάμε, είτε δεν έχουν καταγραφεί, είτε έχουν τιμή "*Not Asked*". Το πρόβλημα ήταν ιδιαίτερα μεγάλο, στη περίπτωση του πληροφοριακού συστήματος των Ανωγείων, όπου καταφέραμε να συγκεντρώσουμε μόλις *3* ασθενείς, και πάλι με την παρουσία *5-6* μόνο `Atomic Observation` να έχουν κανονική μέτρηση!

8.3.1. Αποτελέσματα

Προσοχή. Από το σύνολο των επεξεργαζομένων στοιχείων διαπιστώνουμε ότι έχουμε καταρχήν *40* διαφορετικά `Atomic Observations` (πίνακας 12), τα οποία και συνθέτουν στο σύνολο της την Βιοχημική Εξέταση, έτσι όπως αυτή έχει καταγραφεί στα πληροφοριακά μας συστήματα. Από τις 40 αυτές υπό-εξετάσεις όπως διαπιστώνουμε(τρέχοντας τον αλγόριθμο) οι *25* από αυτές δεν εμφανίζουν σε καμία (σχεδόν) από τις 42 εξεταζόμενες επισκέψεις, ποτέ κάποια τιμή μέτρησης ("Not Asked"). Δεν έχει λοιπόν νόημα και πάλι να λάβουμε αυτά τα χαρακτηριστικά υπόψη μας κατά την εκτέλεση του αλγορίθμου, για τους λόγους που εξηγήσαμε στο προηγούμενο παράδειγμα. Επίσης, άλλα *5* από τα εναπομείναντα `AtomicObservations (K, NA, HDL-ΧΟΛΗΣΤΕΡΙΝΗ, LDL-ΧΟΛΗΣΤΕΡΙΝΗ, ΣΙΔΗΡΟΣ)` έχουν πραγματικές τιμές μετρήσεων (όχι "*Not Asked*") σε ποσοστό μικρότερο από το *35*% των συνολικών εμφανίσεων τους. Για το λόγο αυτό δεν θα τα λάβουμε υπόψη μας στη διαδικασία που θα ακολουθήσει. Για τα *10* λοιπόν συχνότερα εμφανιζόμενα με πραγματικές αριθμητικές τιμές μέτρησης, `Atomic Observations` (πίνακας 13), εκτελούμε τον αλγόριθμο και παρουσιάζουμε τα αποτελέσματα με τις δυο διαφορετικές εναλλακτικές επιλογές (*παραμετρικοποιήσεις*) του αλγορίθμου όπως και στο προηγούμενο παράδειγμα.

Πίνακας 12. *Όλες οι παρατηρούμενες υπό-εξετάσεις για την Βιοχημική Εξέταση*

ΣΑΚΧΑΡΟ	ΟΥΡΙΑ	Κ	ΝΑ	ΧΟΛΗΣΤΕΡΙΝΗ
HDL-ΧΟΛΗΣΤΕΡΙΝΗ	LDL-ΧΟΛΗΣΤΕΡΙΝΗ	ΤΡΙΓΛΥΚΕΡΙΔΙΑ	ΟΛΙΚΑ ΛΙΠΙΔΙΑ	ΟΥΡΙΚΟ ΟΞΥ
SGOT	SGPT	ΟΛΙΚΗ ΧΟΛΕΡΥΘΡΙΝΗ	ΑΜΕΣΗ ΧΟΛΕΡΥΘΡΙΝΗ	ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ
Γ-GT	CPK	LDH	ΟΛΙΚΑ ΛΕΥΚΩΜΑΤΑ	ΑΛΒΟΥΜΙΝΗ
Α-ΑΜΥΛΑΣΗ	ΑΣΒΕΣΤΙΟ	ΣΙΔΗΡΟΣ	ΑΝΟΧΗ ΓΛΥΚΟΖΗΣ0	ΑΝΟΧΗ ΓΛΥΚΟΖΗΣ60
ΑΝΟΧΗ ΓΛΥΚΟΖΗΣ90	ΑΝΟΧΗ ΓΛΥΚΟΖΗΣ120	ΠΑΡΑΤΗΡΗΣΕΙΣ	ΚΡΕΑΤΙΝΙ	ΦΕΡΙΤΙΝΗ
ΛΟΙΠΑ	ΦΩΣΦΟΡΟΣ	ΣΦΑΙΡΙΝΕΣ	ΧΡΟΝΟΣ QUICK ΜΑΡΤΥΡΑ	ΧΡΟΝΟΣ QUICK ΑΣΘΕΝΗ
ΟΞΙΝΗ ΦΩΣΦΑΤΑΣΗ	ΠΡΟΣΤΑΤΙΚΟ ΚΛΑΣΜΑ	B12	ΦΥΛΛΙΚΟ	PSA

Πίνακας 13. *Οι 10 συχνότερα μετρήσιμες υπό-εξετάσεις για την Βιοχημική Εξέταση*

ΣΑΚΧΑΡΟ	ΟΥΡΙΑ	ΚΡΕΑΤΙΝΙ	ΧΟΛΗΣΤΕΡΙΝΗ	ΤΡΙΓΛΥΚΕΡΙΔΙΑ
ΟΥΡΙΚΟ ΟΞΥ	Γ-GT	SGOT	SGPT	ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ

Περίπτωση Α. Στην περίπτωση αυτή *τρέχουμε* τον αλγόριθμο στα δεδομένα μας, χωρίς να λαμβάνουμε υπόψη τις τιμές "*Not Asked*" (επεξεργαζόμαστε δηλαδή την τιμή μιας υπό-εξέτασης μόνο αν είναι διάφορη από "*Not Asked*", στην αντίθετη περίπτωση την αγνοούμε) . Από τις *42* τελικές ιατρικές επισκέψεις ζητάμε την εξαγωγή κανόνων συσχέτισης με παρουσία τουλάχιστον *20* εμφανίσεων (*support = 20/42 = 47.6%*) και αρκετά υψηλό *confidence 90%.* Έχουμε φροντίσει η διαδικασία παραγωγής να φτάνει μόνο μέχρι την δημιουργία *3-large*-itemsets, ώστε να είναι δυνατή η παρουσία <u>όλων</u> των εξαγόμενων κανόνων. Αναφέρουμε ότι για την απαίτηση μας δημιουργούνται μόνο $|C_3| = 5$ υποψήφιοι. Οι σχετικοί κανόνες αλληλοσυσχέτισης φαίνονται στον πίνακα 14.

Πίνακας 14. *Οι παραγόμενοι κανόνες για Βιοχημικές Εξετάσεις*

	Support %	**Confidence** %
ΣΑΚΧΑΡΟ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	59.5	96.2
ΟΥΡΙΑ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	69.0	93.5
SGPT[LOW] => SGOT[LOW]	52.4	95.7
SGOT[LOW] => SGPT[LOW]	54.8	95.8
Γ-GT[LOW] => SGOT[LOW]	54.8	95.8
SGOT[LOW] => Γ-GT[LOW]	54.8	100
Γ-GT[LOW] => SGPT[LOW]	54.8	95.8
SGPT[LOW] => Γ-GT[LOW]	54.8	100
ΣΑΚΧΑΡΟ[LOW] ΟΥΡΙΑ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	47.6	95.2
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] Γ-GT[LOW] => SGOT[LOW]	47.6	95.2
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] SGOT[LOW] => Γ-GT[LOW]	47.6	100
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] Γ-GT[LOW] => SGPT[LOW]	47.6	95.2
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] SGPT[LOW] => Γ-GT[LOW]	47.6	100
SGPT[LOW] Γ-GT[LOW] => SGOT[LOW]	52.4	95.7
SGOT[LOW] Γ-GT[LOW] => SGPT[LOW]	52.4	95.7
SGOT[LOW] SGPT[LOW] => Γ-GT[LOW]	52.4	100

Περίπτωση Β. Στην περίπτωση αυτή τρέχουμε τον αλγόριθμο πάνω ακριβώς στα ίδια δεδομένα, με την διαφορά ότι ο αλγόριθμος μεταχειρίζεται και θεωρεί τις τιμές των μετρήσεων για τα `Atomic Observations` που έχουν καταγραφεί ως "*Not Asked*", ως *φυσιολογικές*. Και πάλι αναζητούμε κανόνες με *confidence 90%* και *support 47.6%.* Στον πίνακας 15 παρατηρούμε όλους τους παραγόμενους κανόνες.

Παρατηρούμε ότι η *επίδραση* των `Atomic Observations` που παρουσιάζονται με τιμές "*Not Asked*" δεν είναι ιδιαίτερα μεγάλη σ' αυτό το παράδειγμα. Έχουμε την παραγωγή 19 κανόνων, έναντι 16 στην προηγούμενη περίπτωση. Το γεγονός αυτό οφείλεται στο ότι σε αυτό το παράδειγμα, τα `Atomic Observations` που μετείχαν, παρουσίαζαν τις περισσότερες φορές αριθμητική τιμή μέτρησης. Έτσι ο συνυπολογισμός και των "*Not Asked*" τιμών (που ήταν οι λιγότερες), δεν άλλαξε την κατάσταση. Εξαίρεση αποτελεί το `Atomic Observation 'ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ'`, όπου παρουσιάζεται στους κανόνες λόγο επικράτησης των "*Not Asked*" τιμών. Αναφέρουμε ότι για την απαίτηση μας δημιουργούνται $|C_3| = 6$ υποψήφιοι, μόλις ένας παραπάνω από την προηγούμενη περίπτωση!.

Πίνακας 15. *Οι παραγόμενοι κανόνες για Βιοχημικές Εξετάσεις με συμμετοχή των "Not Asked" τιμών*

	Support %	**Confidence** %
ΣΑΚΧΑΡΟ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	59.5	96.2
ΟΥΡΙΑ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	69.0	93.5
ΚΡΕΑΤΙΝΙΝΗ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	52.4	91.7
ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	57.1	92.3

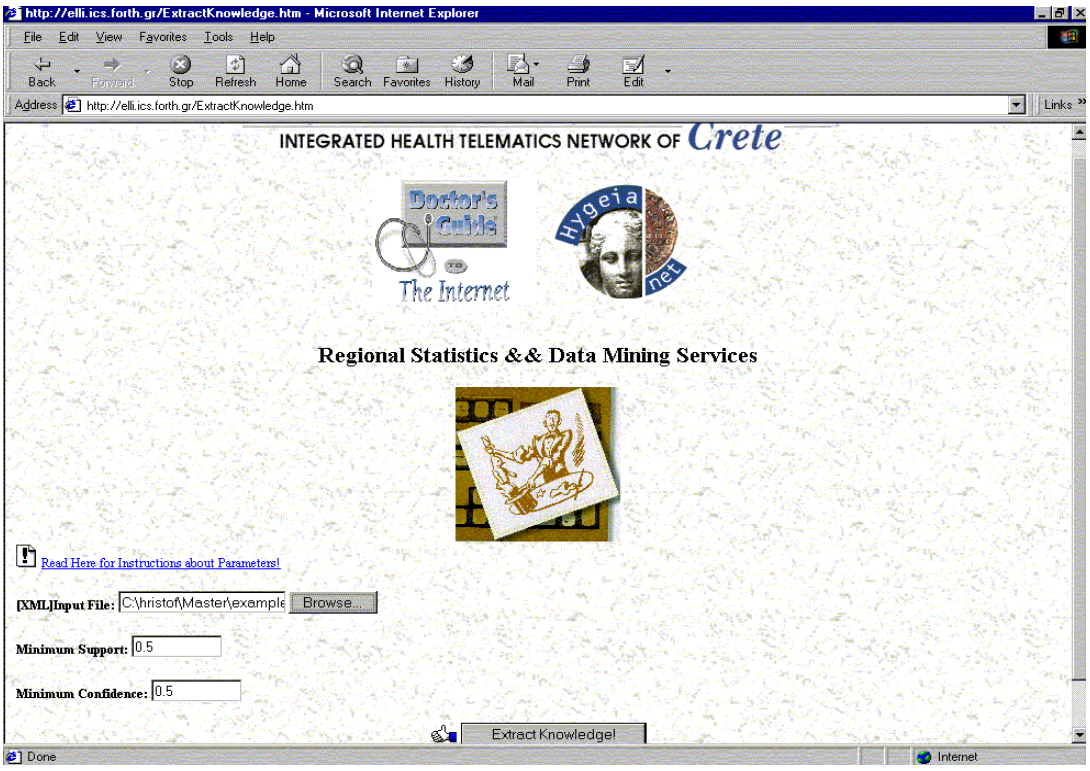
SGPT[LOW] => SGOT[LOW]	52.4	95.7
SGOT[LOW] => SGPT[LOW]	54.8	95.8
Γ-GT[LOW] => SGOT[LOW]	54.8	95.8
SGOT[LOW] => Γ-GT[LOW]	54.8	100
Γ-GT[LOW] => SGPT[LOW]	54.8	95.8
SGPT[LOW] => Γ-GT[LOW]	54.8	100
ΣΑΚΧΑΡΟ[LOW] ΟΥΡΙΑ[LOW] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	47.6	95.2
ΟΥΡΙΑ[LOW] ΑΛΚΑΛΙΚΗ ΦΩΣΦΑΤΑΣΗ[NORMAL] => ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW]	50.0	95.5
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] Γ-GT[LOW] => SGOT[LOW]	47.6	95.2
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] SGOT[LOW] => Γ-GT[LOW]	47.6	100
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] Γ-GT[LOW] => SGPT[LOW]	47.6	95.2
ΤΡΙΓΛΥΚΕΡΙΔΙΑ[LOW] SGPT[LOW] => Γ-GT[LOW]	47.6	100
SGPT[LOW] Γ-GT[LOW] => SGOT[LOW]	52.4	95.7
SGOT[LOW] Γ-GT[LOW] => SGPT[LOW]	52.4	95.7
SGOT[LOW] SGPT[LOW] => Γ-GT[LOW]	52.4	100

Κεφάλαιο 9:
Το Περιβάλλον Web του Συστήματος – Παρουσίαση Αποτελεσμάτων – Χρήση του Συστήματος

Στο κεφάλαιο αυτό επιδεικνύουμε τον τρόπο παρουσίασης του συστήματος μας στους χρήστες του. Παρουσιάζουμε κάποιες οθόνες (screen dumps) οι οποίες αντιστοιχούν στην *πλοήγηση* του χρήστη στο σύστημα μας, από την αρχική είσοδο του στο δημιουργούμενο περιβάλλον, έως την τελική παρουσίαση σε αυτόν όλων των παραγομένων αποτελεσμάτων.

9.1. Είσοδος στο Σύστημα

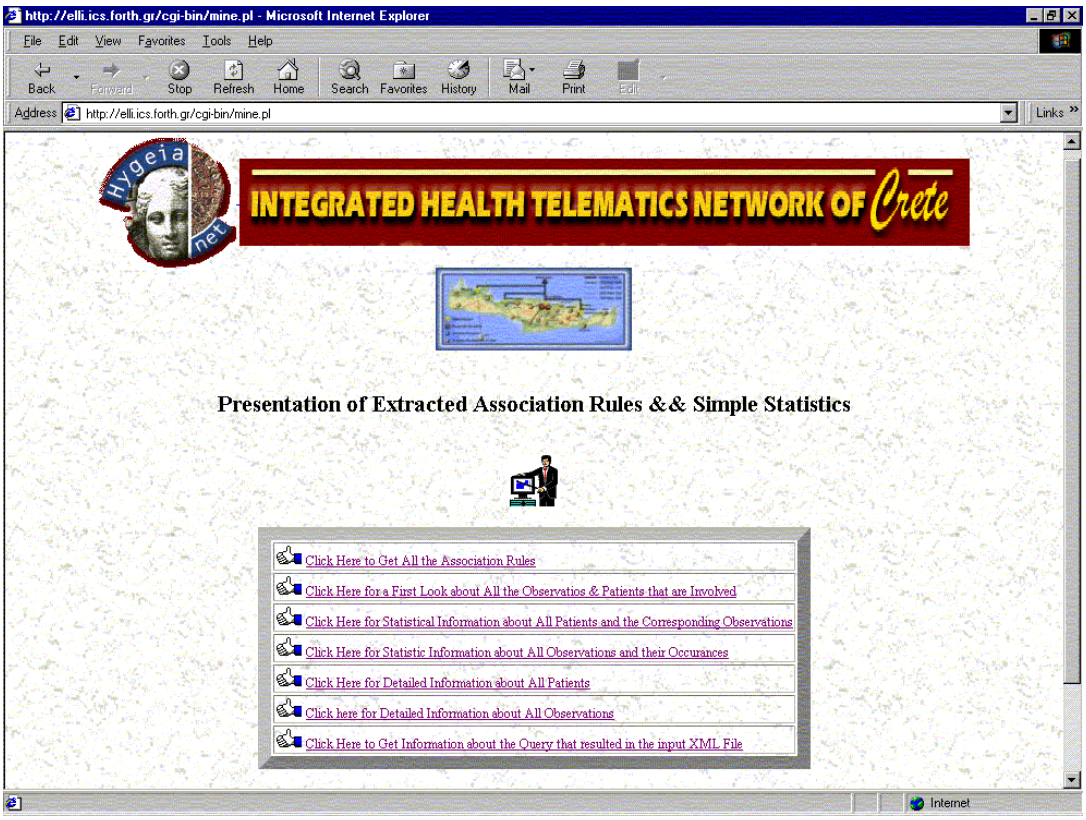
Στο σχήμα 46 παρουσιάζουμε την αρχική σελίδα υποδοχής του συστήματος μας. Όπως παρατηρούμε ο χρήστης καλείται να συμπληρώσει και να δώσει τις απαιτούμενες παραμέτρους στον αλγόριθμο μας, προκειμένου να επιτελεστούν όλες οι mining διεργασίες που αναφέραμε στα προηγούμενα κεφάλαια. Σε δυο φόρμες καλείται να συμπληρώσει τα ζητούμενα και από το χρήστη προκαθορισμένα όρια *'minconf'* και *'minsup'*. Παράλληλα του δίνεται η ευκαιρία να επιλέξει με όμορφο τρόπο το *XML αρχείο* που θα δώσει ως είσοδο στον αλγόριθμο μας, κάνοντας *'browse'* και *'double click'* στο επιλεγμένο αρχείο. Οδηγίες για όλες τις παραμέτρους παρέχονται από αντίστοιχο βοηθητικό *link*. Έχοντας δώσει όλες τις επιθυμητές παραμέτρους δεν μένει παρά να εκτελέσει το πρόγραμμα μας, κάνοντας click πάνω στο *'Extract Knowledge'* button. Με τον τρόπο αυτό οι παράμετροι περνιούνται στον αλγόριθμο μας, το πρόγραμμα εκτελείται και δυναμικά (με χρήση HTML και cgi/perl-scripts) στην επόμενη πλέον σελίδα(σχήμα 47) παρουσιάζονται τα αποτελέσματα του αλγορίθμου μας.



Σχήμα 46: *Είσοδος στο Σύστημα*

9.2. Παρουσίαση Αποτελεσμάτων

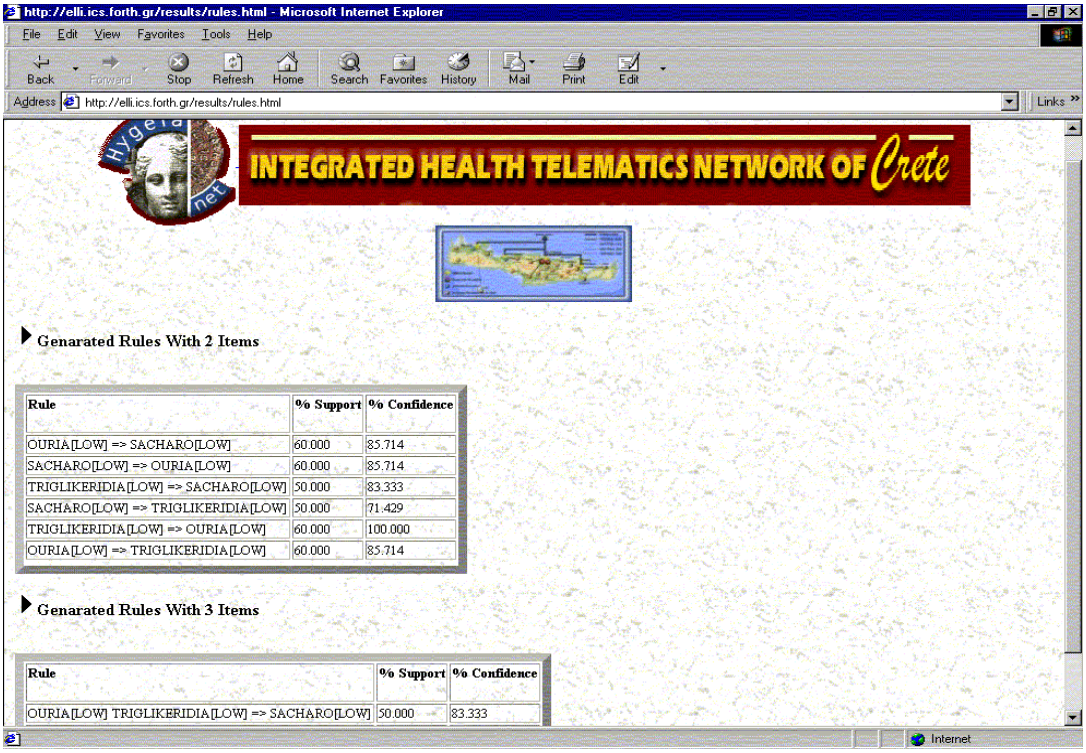
Στη σελίδα αυτή αφότου έχει εκτελεστεί ο αλγόριθμος, παρουσιάζονται με μορφή πίνακα όλα τα δυνατά αποτελέσματα. Κάθε γραμμή του πίνακα έχει την μορφή link, πάνω στο οποίο περιγράφεται και η πληροφορία που μπορεί να ανακαλέσει ο χρήστης επιλέγοντας το.



Σχήμα 47: *Παρουσίαση Αποτελεσμάτων*

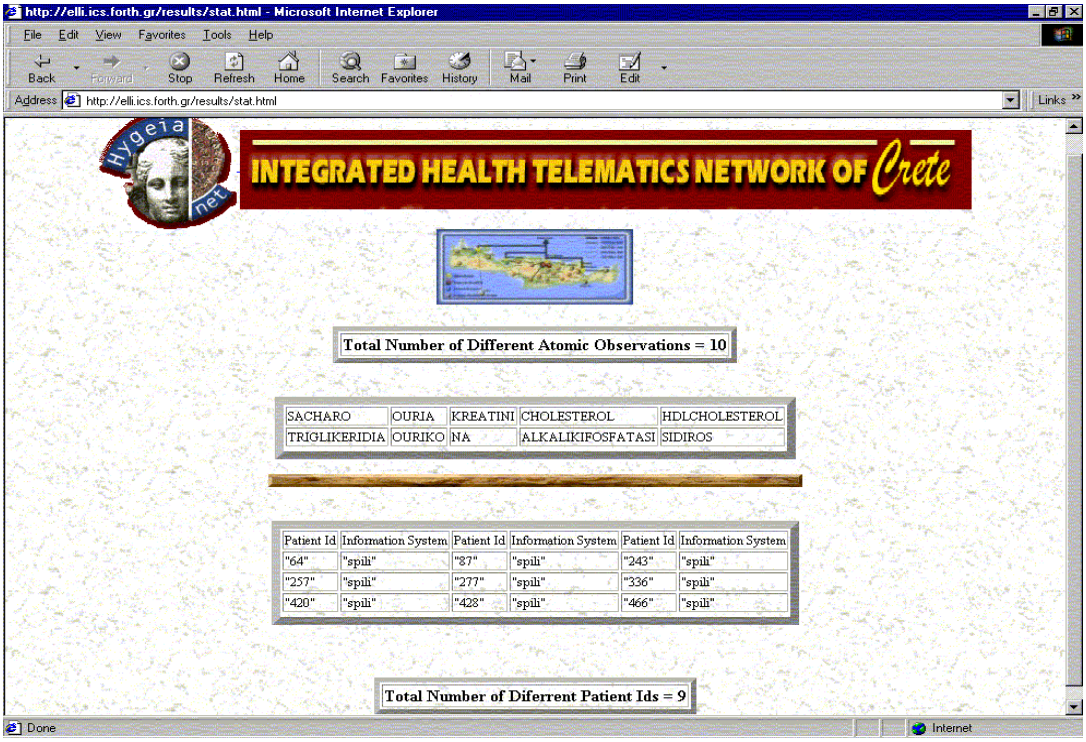
Του δίνεται η ευκαιρία να ανακλά πληροφορίες τόσο για τους παραγόμενους ιατρικούς κανόνες συσχέτισης(κεφάλαιο 7), όσο και για στατιστικά στοιχεία που αφορούν τους ασθενείς και τις εξετάσεις στις οποίες υποβλήθηκαν(αναλυτικά κεφάλαιο 4). Τα links που του παρουσιάζονται του δίνουν την ευκαιρία να ενημερωθεί για κάποια στατιστικά θέματα, τόσο με μια γρήγορη ματιά για σύντομη κατατόπιση και ενημέρωση, όσο και για μια πιο αναλυτική και σε βάθος παρουσίαση. Η εξέταση όλων των links προσδίδει μια ολοκληρωμένη και σε βάθος παρουσίαση της παραγόμενης γνώσης. Πιο αναλυτικά:

- Επιλέγοντας το πρώτο link του πίνακα μπορεί να ενημερωθεί για όλους τους παραγόμενους ιατρικούς κανόνες, οι οποίοι εκμαιεύτηκαν από το XML αρχείο εισόδου και για τα συγκεκριμένα όρια *'minconf'* και *'minsup'*. Ο χρήστης έχει τη δυνατότητα να παρατηρήσει ομαδοποιημένους τους κανόνες, με βάση το πλήθος των items που τους αποτελούν(σχήμα 48).



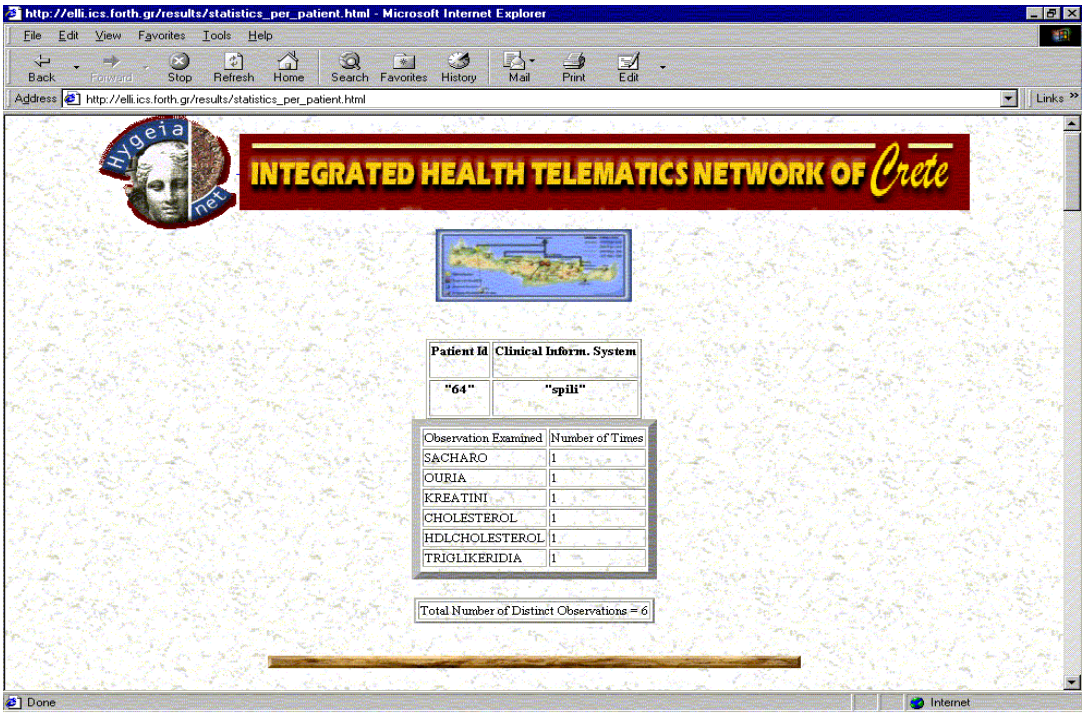
Σχήμα 48: *Εξαγόμενοι Ιατρικοί Κανόνες Συσχέτισης*

- Επιλέγοντας το δεύτερο link ο χρήστης μπορεί άμεσα να ενημερωθεί για το πλήθος τον διαφορετικών εξετάσεων(Atomic Observations), αλλά και όλων των διαφορετικών ασθενών οι οποίοι μετείχαν στην επερώτηση που έγινε(σχήμα 49). Έτσι μπορεί να γνωρίζει από τι σύνολο εξετάσεων και από τι αριθμό ασθενών προήλθαν οι εξαγόμενοι κανόνες, γεγονός ιδιαίτερα σημαντικό.

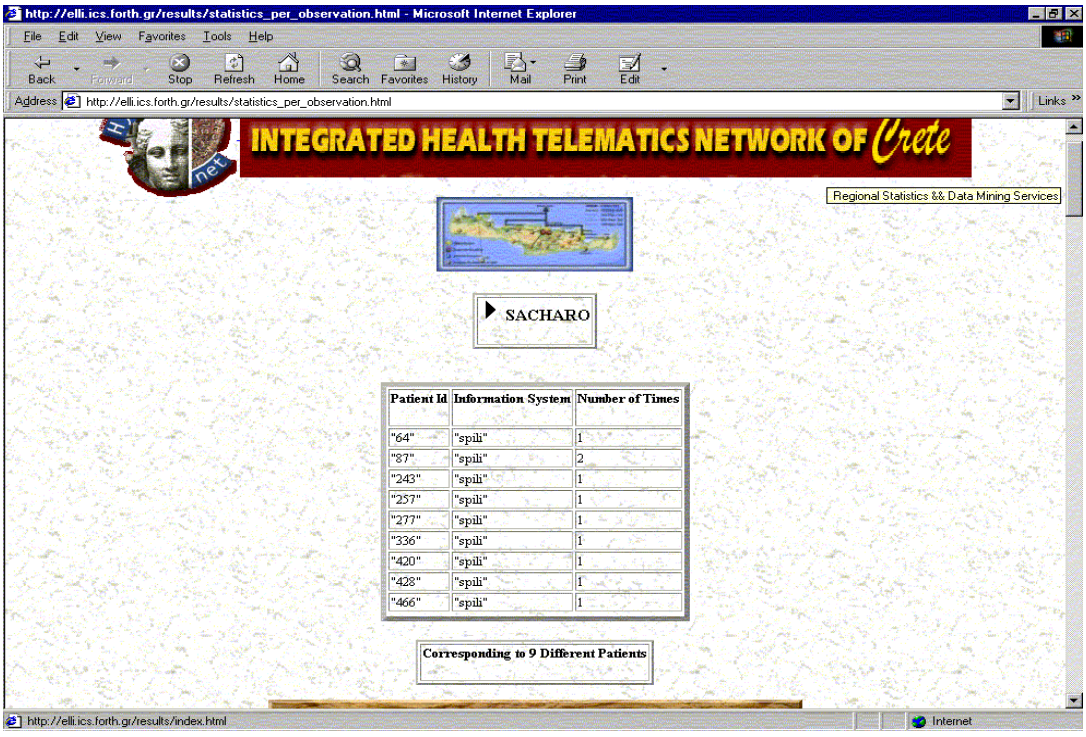


Σχήμα 49: *Όλες οι Εξετάσεις και όλοι οι Ασθενείς*

- Επιλέγοντας το τρίτο και τέταρτο link μπορεί να διαπιστώσει για κάθε ασθενή ποιες εξετάσεις έχει πραγματοποιήσει, αλλά και πόσες φορές έχει υποβληθεί στην κάθε μία από αυτές(σχήμα 50). Ανάλογα, για κάθε εξέταση μπορεί να διαπιστώσει σε πόσους και ποιους ασθενείς έχει πραγματοποιηθεί, καθώς και πόσες φορές, γεγονός που υποδηλώνει το πλήθος των επισκέψεων του εκάστοτε ασθενή(σχήμα 51).

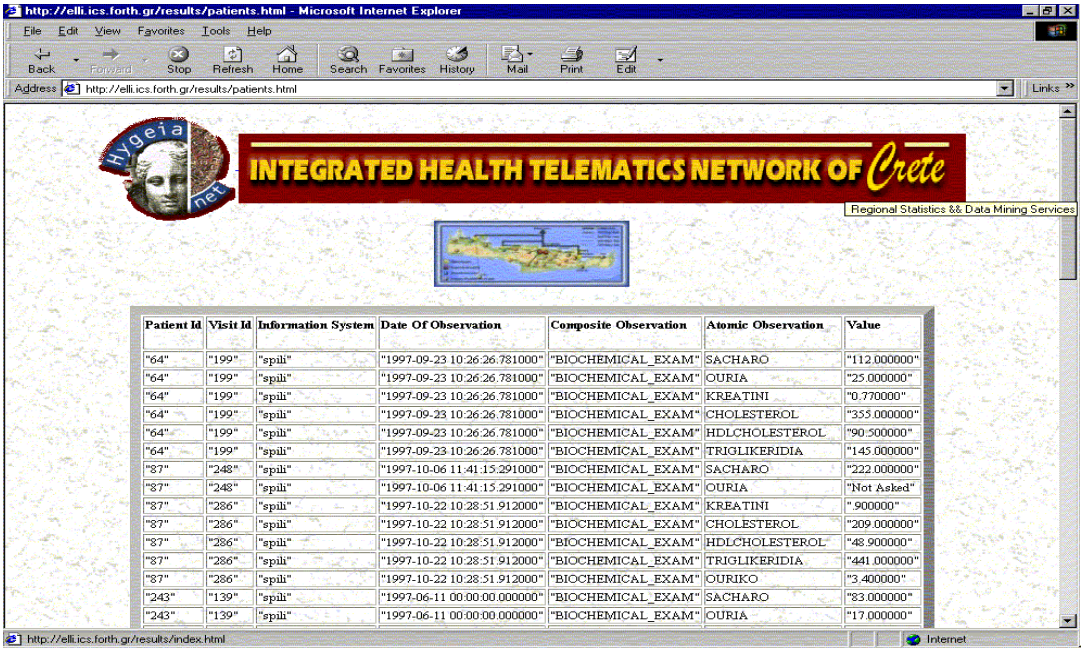


Σχήμα 50: *Όλες οι Εξετάσεις που έχει πραγματοποιήσει ένας Ασθενής*



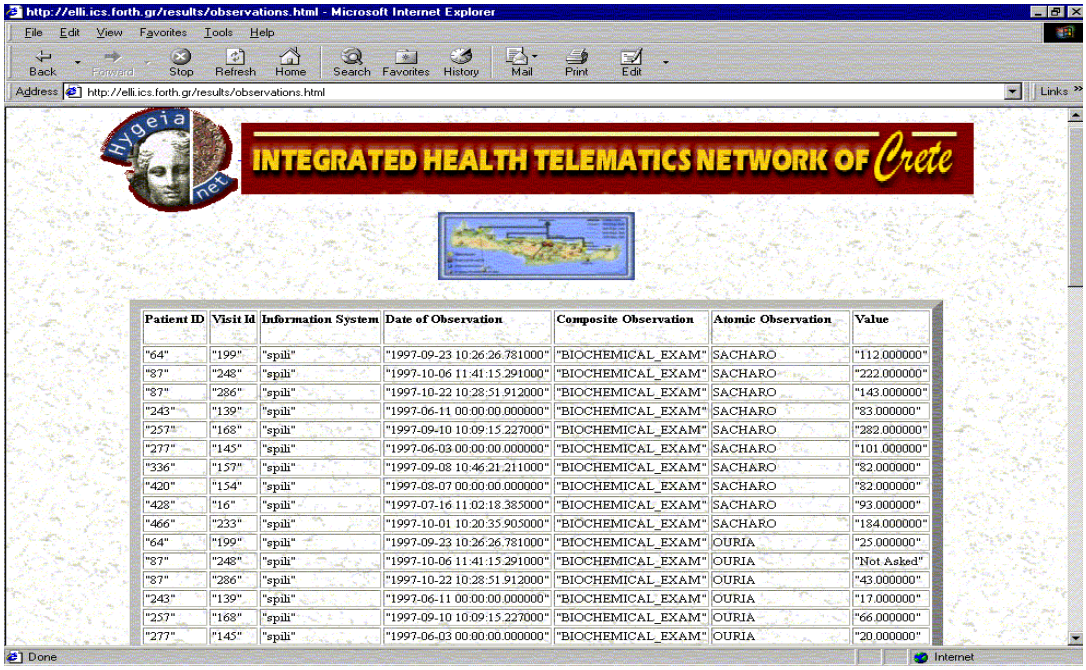
Σχήμα 51: *Όλοι οι Ασθενείς που έχουν υποβληθεί στην εκάστοτε Εξέταση*

- Επιλέγοντας το πέμπτο και έκτο link μπορεί να πάρει πρόσθετες και αναλυτικές πληροφορίες. Μπορεί να ενημερωθεί για ημερομηνίες εξετάσεων, για τις τιμές τους στην κάθε επίσκεψη, για το λόγο που έγιναν και σε ποια γενικότερη / συνθετότερη εξέταση έγκεινται, σε ποιο κλινικό πληροφοριακό σύστημα καταγράφηκαν κ.τ.λ. Τις πληροφορίες αυτές μπορεί να τις πάρει ταξινομημένες είτε ως προς την εκάστοτε εξέταση(σχήμα 53), είτε ως προς τον κάθε ασθενή(σχήμα 52), με οφέλη και στις δυο περιπτώσεις που περιγράφηκαν στο κεφάλαιο 4.



Patient Id	Visit Id	Information System	Date Of Observation	Composite Observation	Atomic Observation	Value
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	SACHARO	"112.000000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	OURIA	"25.000000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	KREATINI	"0.770000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	CHOLESTEROL	"355.000000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	HDLCHOLESTEROL	"90.500000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	TRIGLIKERIDIA	"145.000000"
"87"	"248"	"spili"	"1997-10-06 11:41:15.291000"	"BIOCHEMICAL_EXAM"	SACHARO	"222.000000"
"87"	"248"	"spili"	"1997-10-06 11:41:15.291000"	"BIOCHEMICAL_EXAM"	OURIA	"Not Asked"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	KREATINI	".900000"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	CHOLESTEROL	"209.000000"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	HDLCHOLESTEROL	"48.900000"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	TRIGLIKERIDIA	"441.000000"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	OURIKO	"3.400000"
"243"	"139"	"spili"	"1997-06-11 00:00:00.000000"	"BIOCHEMICAL_EXAM"	SACHARO	"83.000000"
"243"	"139"	"spili"	"1997-06-11 00:00:00.000000"	"BIOCHEMICAL_EXAM"	OURIA	"17.000000"

Σχήμα 52: *Αναλυτικές πληροφορίες ταξινομημένες ανα Ασθενή*



Patient ID	Visit Id	Information System	Date of Observation	Composite Observation	Atomic Observation	Value
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	SACHARO	"112.000000"
"87"	"248"	"spili"	"1997-10-06 11:41:15.291000"	"BIOCHEMICAL_EXAM"	SACHARO	"222.000000"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	SACHARO	"143.000000"
"243"	"139"	"spili"	"1997-06-11 00:00:00.000000"	"BIOCHEMICAL_EXAM"	SACHARO	"83.000000"
"257"	"168"	"spili"	"1997-09-10 10:09:15.227000"	"BIOCHEMICAL_EXAM"	SACHARO	"282.000000"
"277"	"145"	"spili"	"1997-06-03 00:00:00.000000"	"BIOCHEMICAL_EXAM"	SACHARO	"101.000000"
"336"	"157"	"spili"	"1997-09-08 10:46:21.211000"	"BIOCHEMICAL_EXAM"	SACHARO	"82.000000"
"420"	"154"	"spili"	"1997-08-07 00:00:00.000000"	"BIOCHEMICAL_EXAM"	SACHARO	"82.000000"
"428"	"16"	"spili"	"1997-07-16 11:02:18.385000"	"BIOCHEMICAL_EXAM"	SACHARO	"93.000000"
"466"	"233"	"spili"	"1997-10-01 10:20:35.905000"	"BIOCHEMICAL_EXAM"	SACHARO	"184.000000"
"64"	"199"	"spili"	"1997-09-23 10:26:26.781000"	"BIOCHEMICAL_EXAM"	OURIA	"25.000000"
"87"	"248"	"spili"	"1997-10-06 11:41:15.291000"	"BIOCHEMICAL_EXAM"	OURIA	"Not Asked"
"87"	"286"	"spili"	"1997-10-22 10:28:51.912000"	"BIOCHEMICAL_EXAM"	OURIA	"43.000000"
"243"	"139"	"spili"	"1997-06-11 00:00:00.000000"	"BIOCHEMICAL_EXAM"	OURIA	"17.000000"
"257"	"168"	"spili"	"1997-09-10 10:09:15.227000"	"BIOCHEMICAL_EXAM"	OURIA	"66.000000"
"277"	"145"	"spili"	"1997-06-03 00:00:00.000000"	"BIOCHEMICAL_EXAM"	OURIA	"20.000000"

Σχήμα 53: *Αναλυτικές πληροφορίες ταξινομημένες ανά Εξέταση*

- Στο τελευταίο link μπορεί να πάρει πληροφορίες σχετικά με τα χαρακτηριστικά της επερώτησης που δημιούργησαν το αντίστοιχο XML και προκάλεσαν όλα τα παραπάνω αποτελέσματα.

Σημείωση Από κάθε σημείο του συστήματος μας ο χρήστης έχει την δυνατότητα να *επιστρέψει* στην αρχική σελίδα και να εκτελέσει τον αλγόριθμο με νέα thresholds, εάν τα αποτελέσματα που προέκυψαν δεν ήταν διαφωτιστικά λόγο μη επιτυχημένης επιλογής στα όρια *'minconf'* και *'minsup'*.

Κεφάλαιο 10:
Συμπεράσματα και Μελλοντική Δουλειά

10.1. Συμπεράσματα

Παρουσιάσαμε μια μεθοδολογία, με το αντίστοιχο αρχιτεκτονικό υπόβαθρο και έναν λειτουργικό σκελετό εργασίας, για *εξόρυξη γνώσεων* (data mining) από κατανεμημένες και ετερογενείς βάσεις δεδομένων και τα σχετικά ιατρικά πληροφοριακά συστήματα.
Η κατανόηση της προτεινόμενης αρχιτεκτονικής δεν είναι μια εύκολη διαδικασία και μια *πολυδιάστατη* φάση ολοκλήρωσης/ επεξεργασίας απαιτείται και κρίνεται αναγκαία να ακολουθηθεί. Αυτή η προσέγγιση κρίνεται επιβεβλημένη και υπαγορεύεται από τον συνδυασμό και την παρουσία πολλών (multi disciplinary)τεχνολογιών και λειτουργιών. Ενδεικτικά αναφέρουμε τη χρήση τεχνολογίας CORBA για ομοιόμορφη πρόσβαση σε κατανεμημένα δεδομένα, λειτουργίες *σημασιολογικής ομογενοποίησης* και προηγμένες *DTD/ XML* διαδικασίες. Οι λειτουργίες αυτές, συσχετισμένες με σύγχρονες, προχωρημένες και αποτελεσματικές *αναπαραστάσεις μοντέλων*, διαμορφώνουν και προσδιορίζουν ένα σκελετό και ένα περιβάλλον, στο οποίο μπορούν να εκπονηθούν πλέον έξυπνα και αποτελεσματικά όλες οι απαιτούμενες και αναγκαίες KDD διεργασίες.

- Τα παραπάνω πιστοποιούν την *πολυσυνθετικότητα* της προσέγγισης, όσον αφορά την εξαγωγή γνώσης από κατανεμημένες και ετερογενείς βάσεις δεδομένων, η οποία συνδυασμένη με την τροποποίηση-μορφοποίηση των KDD/ARM διεργασιών, ώστε να είναι απόλυτα εφαρμόσιμες στα παραγόμενα XML έγγραφα, αποτελούν τη βασική καινοτομία της εργασίας μας. Αξίζει να αναφέρουμε ότι σε μια πρόσφατη παρουσίαση από τον Rakesh Agrawal [43] σημειώνεται η ανάγκη για DTD/XML μοντελοποίηση των σύγχρονων βάσεων δεδομένων και τονίζεται η αντίστοιχη ανάγκη επεξεργασίας και μορφοποίησης προς αυτήν την κατεύθυνση όλων των KDD διεργασιών.

- Βασισμένοι στην πρόβλεψη ότι οι μελλοντικές βάσεις θα χρησιμοποιούν XML-like αναπαραστάσεις και μορφές δεδομένων, προκειμένου να αποθηκεύεται και να εκμαιεύεται η προς επεξεργασία πληροφορία, η εργασία μας παρουσιάζει μια υποσχόμενη αρχιτεκτονική και ένα περιβάλλον εργασίας κινούμενο προς αυτήν ακριβώς την κατεύθυνση.

10.2. Μελλοντική Δουλειά

Τα μελλοντικά σχέδια επέκτασης της δουλειάς μας, συνοπτικά θα μπορούσαμε να αναφέρουμε ότι κινούνται σε τέσσερις κατευθύνσεις:

I. Εκτέλεση και εφαρμογή *μεγάλης κλίμακας πειραμάτων* (με τη χρήση όλων των κλινικών πληροφοριακών συστημάτων που μετέχουν στον ολοκληρωμένο ηλεκτρονικό φάκελο υγείας που αναπτύσσεται στην περιφέρεια της Κρήτης), με στόχο να εξετάσουμε και να επιβεβαιώσουμε την ***αποτελεσματικότητα*** της προσέγγισης μας.

II. Το σχεδιασμό και την ανάπτυξη κατάλληλων ***διεπιφανειών ανθρώπου-υπολογιστή*** (human computer interface), για την μεταφορά και παρουσίαση των

KDD αποτελεσμάτων, λαμβάνοντας υπόψη θέματα που σχετίζονται με την *προσωπικότητα του εκάστοτε χρήστη* (***user profile, personalization***). Στο σημείο αυτό δίνουμε ιδιαίτερη σημασία στην συμμετοχή του ίδιου του χρήστη στην ARM διαδικασία. Για παράδειγμα, ο χρήστης θα μπορεί να επιλέγει ένα μόνο μέρος από τους παραγόμενους κανόνες, υποδεικνύοντας και επιλέγοντας τα items (τμήματα της πληροφορίας) που τον ενδιαφέρουν. Έτσι, προτού παρουσιαστούν οι κανόνες στο σύνολο τους, θα φιλτράρονται μέσω της υπόδειξης του χρήστη και θα παρουσιάζονται μόνο αυτοί που περιέχουν τουλάχιστον ένα item από αυτά που ενδιαφέρουν το χρήστη.

III. Βελτιώσεις και προσθήκες όπου κρίνεται αναγκαίο στις χρησιμοποιούμενες ML/ ARM/ KDD διεργασίες, με ταυτόχρονη υιοθέτηση και τροποποίηση και *άλλων διεργασιών και μεθοδολογιών* (***clustering***, ***decision trees*** κ.τ.λ) στο περιβάλλον εργασίας μας.

Ενδεικτικά αναφέρουμε μια αξιόλογη βελτίωση η οποία είναι κρυμμένη σε ένα λεπτό σημείο. Γνωρίζουμε ότι ένα βασικό κλειδί στην εξόρυξη κανόνων συσχέτισης, είναι ο **προσδιορισμός** και καθορισμός του **minsup**. Χρησιμοποιείται για να ελαχιστοποιήσει το χώρο αναζήτησης, καθώς και για να περιορίσει τον αριθμό των δημιουργούμενων κανόνων. Ωστόσο, χρησιμοποιώντας και ορίζοντας ένα μόνο minsup στην εργασία μας, αυτομάτως δεχόμαστε ότι όλα τα items στη συλλογή μας, έχουν την ίδια βαρύτητα και εμφανίζονται με ανάλογες συχνότητες. Δεν ισχύει όμως κάτι ανάλογο στο χώρο της ιατρικής στον οποίο εργαζόμαστε, καθότι υπάρχουν εξετάσεις οι οποίες πραγματοποιούνται με πολύ μικρότερη συχνότητα από κάποιες άλλες. Έτσι σε περιπτώσεις όπου οι συχνότητες εμφάνισης των αντίστοιχων items σε μια συλλογή στοιχείων εμφανίζονται να διαφέρουν αρκετά, ερχόμαστε αντιμέτωποι με 2 προβλήματα:

- ✓ Εάν το *minsup* τεθεί πολύ ψηλά, δεν θα μπορέσουμε να βρούμε κανόνες που θα αφορούν, τα σπάνια ή αραιά εμφανιζόμενα items (Atomic Observations/ εξετάσεις)
- ✓ Εάν το *minsup* τεθεί πολύ χαμηλά, προκειμένου οι κανόνες μας να αφορούν τόσο τα συχνά όσο και μη-συχνά στοιχεία της συλλογής μας, το αποτέλεσμα είναι η παραγωγή ενός πολύ μεγάλου αριθμού κανόνων, πολλοί από τους οποίους δεν παρουσιάζουν κάποιο ιδιαίτερο και αξιόλογο ενδιαφέρον.

Το δίλημμα αυτό είναι γνωστό με την ονομασία *rare item problem* [41]. Η λύση στο παραπάνω πρόβλημα δίνεται, με το να επιτρέπουμε στον χρήστη να ορίζει *multiple minimum supports*. Πιο συγκεκριμένα ο χρήστης μπορεί να ορίζει διαφορετικό minsup για κάθε item. Η προσέγγιση αυτή μας δίνει τη δυνατότητα να παράγουμε "rare item rules", χωρίς να προκαλούμε την παραγωγή μεγάλου και χωρίς νόημα αριθμού κανόνων, όσον αφορά τα συχνά εμφανιζόμενα items [42]. Με αυτόν τον τρόπο, θα παράγουμε κανόνες οι οποίοι θα αφορούν εξετάσεις που σχετίζονται με σπάνιες παθήσεις και έχουν μικρή συχνότητα εμφάνισης στα πληροφοριακά μας συστήματα.

IV. Εφαρμογή της αρχιτεκτονικής μας και σε ***άλλα πεδία εφαρμογών*** (π.χ εξόρυξη γνώσης από οικονομικές και χρηματιστηριακές πηγές πληροφόρησης).

Βιβλιογραφία

[1] Rakesh Agrawal, Sakti Ghost, Tomasz Imielinski, and Arun Swami. An interval classifier for database mining applications. In 18th Int'l Conf. On Very Large Databases(VLDB), Vancouver, Canada pages 560-573, 1992.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Database mining: a performance perspective. IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 6, December 1993:914-925, 1993.

[3] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining Association Rules between sets of items in large databases. In SIGMOD, Washington D.C, pages 207-216, May 1993.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In 20th Int'l Conf. On Very Large Databases(VLDB), Santiago. Chile, Sept. 1994. Expanded version available as IBM Research Report RJ9839, June 1994.

[5] Marko Bohanec and Ivan Bratko. Trading accuracy for simplicity in decision trees. Machine Learning, 15:223-250, 1994.

[6] Forslund D., and Kilman D. The Virtual Patient Record: A Key to Distributed Healthcare and Telemedicine. *Los Alamos National Laboratory*. February 29, 1996 http://www.acl.lanl.gov/TeleMed/Papers/virtual.html.

[7] InterCare. InterCare End-user Applications, *Deliverable D4.1*, *Health Telematics program, Europe, HC 4011 project*. November 1999.

[8] Grimson W., Berry D., Grimson J., Stephens G., Felton E., Given P., and O'Moore R. Federated Healthcare Record Server - the Synapses paradigm. *Web document*, http:// www.cs.tcd.ie/synapses/public/html/ technicaldescription.html, 1997.

[9] Hsu C., Bouziane M., Cheung W., Rattner L., and Yee L. Metadatabase Modeling for Enterprise Information Integration. *Journal of Systems Integration*, 2:1, pp. 5-37, 1992.

[10] Sciore E., Siegel M., and Rosenthal A. Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems. *ACM Transactions on Database Systems*, Vol. 19, No. 2, pp. 254-290, June 1994.

[11] Baldonado Wang M.Q., and Cousins S.B. Addressing Heterogeneity in the Networked Information Environment. Technical Report, *Computer Science Department, Stanford University*, December 1996.

[12] Tsiknakis M, Chronaki C.E., Kapidakis S., Nikolaou C, and Orphanoudakis S.C., An Integrated Architecture for the Provision of Health Telematic Services based on Digital Library Technologies**.** *International Journal on Digital Libraries*, Special Issue on "Digital Libraries in Medicine", vol. 1(3), pp. 257-277, 1997.

[13] OMG group. Web site, http://www.omg.org.

[14] Shuh B. Directories and X.500: An Introduction. *Network Notes #45*, ISSN 1201-4338, Information Technology Services, National Library of Canada, http://www.nlc-bnc.ca/pubs/netnotes/notes45.htm, March 1997.

[15] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: An overview. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, Knowledge Discovery in Databases, pages 1-30,

AAAI/MIT, 1991.

[16] ITU. Recommendation X.500 (11/93) - Information technology - Open Systems Interconnection - The directory: Overview of concepts, models, and services, 1993.

[17] CORBA. Web site, http://www.corba.org.

[18] Lelis P. Data Integration in Heterogeneous, Autonomous and Distributed Clinical Information Systems. BSc Thesis, *Dept. of Computer Science, University of Crete*, Heraklion, October 1999.

[19] COAS. *Clinical Observations Access Service (COAS)*. Final Submission, OMG Document: corbamed/99-03-25, 1999.

[20] 2AB, Care Data Systems, Inc., CareFlowNet, Inc., HBO & Company, HealthMagic, Inc., HUBlink, Inc., IDX Systems Corporation, IONA Technologies PLC, Oacis Healthcare Systems, Protocol Systems, Inc., Sholink Corporation, "Person Identification Service (PIDS)", OMG CORBAmed DTF Adopted Submission, OMG TC Document corbamed/98-02-29, February 1998.

[21] Marcel Holsheimer and Arno P.J.M. Siebes. Data mining: the search for knowledge in databases. Technical Report CS-R9429, CWI, January 1994.

[22] Houtsma and Arun Swami. Set-oriented mining of association rules. Technical Report RJ 9567, IBM Research Report, Oct. 1993.

[23] CORBAmed Health Care Domain Task Force of the Object Management Group (http://www.omg/org/corbamed).

[24] Object Management Group, "The CORBAmed Roadmap", Revised Submission, OMG TC Document CORBAmed /98-02-03, February, 1998.

[25] Microsoft Corporation, "DCOM Technical Overview", PDC '97 Conference Paper, White Paper, 1996.

[26] DICOM: An Introduction to the Standard (http://www.xray.hmc.psu.edu/dicom_intro/DICOMIntro.html).

[27] 3M Health Information Systems, and Protocol Systems, Inc., "Lexicon Query Service RFP Response", Revised Submission, OMG, OMG TC Document CORMAmed/98-03-22, March, 1998.

[28] Heikki Mannila, Hannu Toivinen, and A. Inkeri Verkamo. Improved methods for finding association rules. In AAAI Workshop on Knowledge Discovery, Seattle, Washington, pages 181-192, July 1994.

[29] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovering frequent episodes in sequences. In Proc. Knowledge Discovery and Data Mining(KDD'95), (to appear), 1995. Also: Technical Report C-1995-10, University of Helsinki, Department of Computer Science, Finland, March 1995.

[30] Christopher J. Matheus, Philip K. Chan, and Gregory Piatetsky-Shapiro. Systems for knowledge discovery in databases. IEEE Transactions on Knowledge Discovery and Data Engineering, 5(6). Dec. 1993.

[31] Katehakis D.G., Chronaki C.E., et al., "Towards a Virtual Electronic HealthCare Record: The Patient Clinical Data Directory", Version 3.09, 1999.

[32] HYGEIAnet Web site. Integrated Health Care Network of Crete, http://www.hygeianet.gr.

[33] Arun Swami, Rakesh Agrawal, Christos Faloutsos. Efficient similarity search in sequence databases. In 4th Int'l Conf. On Foundations of Data Organization and Algorithms, Chicago, Oct. 1993. Also in Lecture Notes in Computer Science 730,

Springer Verlag, 1993, 69-84.

[34] Ashok Sarasere, Edward Omiecinsky, and Shamkant Navathe. An efficient algorithm for mining association rules in large databases. In 21th Int'l Conf. On Very Large Databases(VLDB), Zurich, Switzerland, Sept. 1995. Also Gatech Technical Report No. GIT-CC-95-04.

[35] Gennari J.H., Stein A.R., and Musen M.A. Reuse For Knowledge-Based Systems and CORBA Components. Proceedings of *10th Knowledge Acquisition Workshop*, Banff, Alberta, Canada, 1996.

[36] UMLS. *UMLS 2000 Documentation*. Web document, http://www.nlm.nih.gov/research/umls/UMLSDOC.HTML

[37] Simon St. Laurent. "XML: Extensible Markup Language". IDG Books, 1998.

[38] David Megginson. "Structuring XML Documents". Printice Hall, 1998.

[39] Ramakrishnan Srikant and Rakesh Agrawal. Mining Quantitative Association Rules in Large Relational Tables. In Proceedings of the ACM SIGMOD Conference on Management of Data, June 1996.

[40] Beat Wuthrich. Knowledge Discovery in databases. Draft course manuscript, Hong Kong University of Science and Technology, May 1994.

[41] Mannila, H. "Database Methods for Data Mining." KDD-98 tutorial, 1998.

[42] Bing Liu, Wynne Hsu and Yiming Ma. Mining Association Rules with Multiple Minimum Supports. In ACM SIGKDD International Conference on Knowledge Discovery & Data Mining(KDD-99), August 15-18, 1999, San Diego, CA, USA.

[43] Agrawal R. Data Mining: Crossing the Chasm. Invited talk at the *5th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining (KDD-99)*, San Diego, California, August 1999.

[44] Han J., and u Y. Discovery of multiple-level association rules from large databases. In *21st Int'l onf. On very Large Databases (VLDB)*, Zurich, Switzerland, 1995.

[45] Haverkamp D., Gauch S. Intelligent Information Agents: Review and Challenges for Distributed Information Sources. *Journal of the American Society for Information Science*, 49:4, pp. 304-311, April 1998

[46] Knoblock C.A., Yigal A. An architecture for information retrieval agents. *AAAI Spring Symposium on Software Agents*, Stanford, 1994.

[47] Mueller A. Fast Sequential and Parallel Algorithms for Association Rule Mining: A Comparision. Technical report CS-TR-3515, dept. of Computer Science, University of Maryland, Vollege Park, MD, August 1995.

[48] Wah B.W., Huang T.S., Joshi A.K., Moldovan D., Aloimonos J., Bajcy R.K., Ballard D., DeGroot D., DeJong K., Dyer C.R., Fahlman E., Grishman R., Hirschman L., Korf R.E., Levinson S.E., Miranker D.P., Morgan N.H., Nirenburg S., Poggio T., Riseman E.M., Stanfill C., Stolfo S.J., Tanimoto S.L., and Weems C. Report on Workshop on High Performance Computing and Communication for Grand Challenge Applications: Computer Vision, Speech and Natural language Processing, and Artificial Intelligence. *IEEE Transactions on Knowledge and Data Engineering*, 5:1, pp. 138-154, February 1993.

[49] Wiederhold G., Mediators in the architecture of future information systems. *IEEE Computer*, 25:3, 1992.

[50] Vladimir Brusic and John Zeleznikow. Knowledge discovery and data mining in biological databases. Cambridge University Press, 1999.

[51] Karsten M. Decker and Sergio Focardi. Technology Overview: A Report on Data Mining. CSCS TR-95-02, May 29, 1995

[52] Ramakrishnan Srikant, Quoc Vu and Rakesh Agrawal. Mining Association Rules with Item Constraints. 1997.

[53] David W. Cheung, Vincent T.Ng, Ada W. Fu and Yongjian Fu. Efficient Mining of Association Rules in Distributed Databases. IEEE Transactions and data engineering, Vol. 8, No 6, December 1996.

[54] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient Algorithms for Discovering Association Rules. Appeared in AAAI Workshop on Knowledge Discovery in Databases, Eds. Usama M. Fayyad and Ramasamy Uthurusamy, pages 181-192, Seattle, Washington, July 1994.

[55] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Improved Methods for Finding Association Rules. Helsinki, December 1993(Revised February 1994).