

University of Crete  
School of Sciences and Engineering  
Computer Science Department

A BYZANTINE MUSIC FRAMEWORK  
USING SEMANTICS AND  
COLLABORATIVE FILTERING  
RECOMMENDATIONS

By

ELEFThERIA RANOUTSOU

Master's Thesis

Heraklion, March 2010



University of Crete  
School of Sciences and Engineering  
Computer Science Department

A BYZANTINE MUSIC FRAMEWORK  
USING SEMANTICS AND  
COLLABORATIVE FILTERING  
RECOMMENDATIONS

By

ELEFThERIA RANOUTSOU

A thesis submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE

Author: \_\_\_\_\_  
Eleftheria Ranoutsou, Computer Science Department

Supervisory  
Committee: \_\_\_\_\_  
Dimitris Plexousakis, Professor, Supervisor

\_\_\_\_\_  
Ioannis Stylianou, Associate Professor, Member

\_\_\_\_\_  
Ioannis Tzitzikas, Assistant Professor, Member

Approved by: \_\_\_\_\_  
Dimitris Plexousakis, Associate Professor  
Chairman of the Graduate Studies Committee

Heraklion, March 2010



## Abstract

Greek music (byzantine and traditional) is an important topic in local as well as international history. There are too many characteristics of high significance that distinguish and differentiate this music from the music of the rest of the world. There are also too complex issues that make the study and organization of it very hard.

Nowadays, since the development of technology allows the management of the mass and variety of information, we try to offer a solution that respects the specificity of the music and makes easier the documentation, organization, search and promotion.

The technological approaches that are tried in this work - applications include the development of an information system with the design of a database system, the usage of web services, the semantic notation, analysis and publicity of data and the utilization of recommendation algorithms.

The development and usage of services by enterprises and academic organizations has increased exponentially in recent years. However, heterogeneity and scalability lead to a new architecture. Web services allow service integration in a fast, effective way, as they are software component that are independent of implementation and platform.

Humans are capable of using the Web in order to carry out tasks such as finding information. However, these tasks cannot be accomplished by a computer, without human interference, since web pages are designed to be read by people. The goal of the Semantic Web is to make web content meaningful, so that data is suitable for automatic reasoning. This leads to the use of some descriptive data (meta-data) in machine readable form for automatic reasoning.

The Resource Description Framework (RDF) is a family of W3C specifications originally designed as a metadata data model. An extension of RDF is RDF Schema, which is a vocabulary for describing properties and classes of RDF-based resources, with semantics for generalized-hierarchies of such properties and classes.

Another semantic web tool is D2RQ, which is a declarative language for describing mappings between relational database schemata and OWL/RDFS ontologies.

Modern digital age requires more automated recommendation systems. It is essential that they may be able to filter out information not useful to the user, and be able to recommend something

according to user's habits and patterns. The approach we are going to use is collaborative filtering. Collaborative filtering takes into account both active user's information and similar users' preferences. The goal is to find users with similar tastes to the active user and to provide recommendation based on the analysis of the preferences from those other users.

We will utilize the technologies discussed above, in order to develop a progressive application that will take advantage from them. We will also show the results of the system we have developed. How it finally works, what test users see, the collaborative filtering efficiency and efficacy, and whether it meets users' needs. As the results have shown, the system comes close to users' preferences. However, several improvements could be made to the system, which we will be discussed.

*Supervisor:* Dimitris Plexousakis

Professor

## Περίληψη

Η ελληνική μουσική (βυζαντινή και παραδοσιακή) αποτελεί ένα μεγάλο κεφάλαιο της τοπικής αλλά και της παγκόσμιας ιστορίας. Τα στοιχεία που τη διακρίνουν και τη διαφοροποιούν από τις μουσικές του κόσμου είναι πάρα πολλά και ιδιάζουσας σημασίας. Η πολυπλοκότητα αυτή όμως της μουσικής καθιστά τη μελέτη και την οργάνωσή της από τους μουσικούς μια ιδιαίτερα επίπονη διαδικασία.

Στη σύγχρονη πραγματικότητα, που η ανάπτυξη της τεχνολογίας επιτρέπει τη διαχείριση του όγκου και της ποικιλομορφίας των πληροφοριών, επιχειρούμε να δώσουμε μια λύση η οποία θα σέβεται την ιδιαιτερότητα αυτής της μουσικής και παράλληλα θα καθιστά ευκολότερη την αρχειοθέτηση, οργάνωση, αναζήτηση και προβολή των πληροφοριών.

Οι τεχνολογικές προσεγγίσεις οι οποίες δοκιμάζονται στη συγκεκριμένη εργασία - εφαρμογή περιλαμβάνουν ανάπτυξη πληροφοριακού συστήματος και βάσης δεδομένων, διαδικτυακές υπηρεσίες (web services), σημασιολογική ανάλυση και προβολή των πληροφοριών, και αλγορίθμους συστάσεων (recommendation system).

Η ανάπτυξη και χρήση των υπηρεσιών από οργανισμούς και ακαδημαϊκά ιδρύματα έχει σημειώσει σημαντική αύξηση τα τελευταία χρόνια. Παρόλ' αυτά η ετερογένεια και η μεταβλητότητα αυτών των συστημάτων, οδηγεί σε μια καινούρια αρχιτεκτονική. Οι ηλεκτρονικές υπηρεσίες επιτρέπουν την γρήγορη και αποδοτική ενοποίηση αυτών των υπηρεσιών, αφού πρόκειται για τμήματα λογισμικού τα οποία μπορούν να λειτουργούν ανεξάρτητα από περιβάλλοντα υλοποίησης.

Οι χρήστες του διαδικτύου, χρησιμοποιούν τον παγκόσμιο ιστό για διάφορες εργασίες, όπως την αναζήτηση πληροφοριών. Ωστόσο, αυτές οι εργασίες δεν μπορούν να πραγματοποιηθούν από τον υπολογιστή, χωρίς την ανθρώπινη παρέμβαση, επειδή οι ιστοσελίδες κατασκευάζονται για να αναγινώσκονται από τους χρήστες. Ο στόχος του Σημασιολογικού Ιστού είναι να δώσει σημασιολογία στο διαδικτυακό περιεχόμενο, έτσι ώστε να υποστηρίζεται η αυτόματη συλλογιστική (automatic reasoning). Αυτό οδηγεί στη χρήση περιγραφικών δεδομένων (μεταδεδομένα), σε τέτοια μορφή, ώστε αυτά να είναι αναγνώσιμα και κατανοήσιμα από μηχανές.

Το RDF (Resource Description Framework) ανήκει στις προδιαγραφές του W3C, και έχει σχεδιαστεί για την μοντελοποίηση των μεταδεδομένων. Μια επέκτασή του, το RDF Schema, είναι ένα λεξικό για την περιγραφή των ιδιοτήτων και κλάσεων των RDF πόρων.

Ένα άλλο εργαλείο του Σημασιολογικού Ιστού, είναι το D2RQ, το οποίο είναι μια δηλωτική γλώσσα για την αντιστοίχιση σχημάτων σχεσιακών βάσεων δεδομένων σε OWL/ RDF οντολογίες.

Η σύγχρονη ψηφιακή εποχή απαιτεί περισσότερο αυτοματοποιημένα συστήματα συστάσεων. Είναι απαραίτητο να μπορούν να φιλτράρουν την άχρηστη για το χρήστη πληροφορία, και να μπορούν να του συστήσουν κάτι το οποίο θα συμφωνεί με τον τύπο και τις συνήθειες του. Η τεχνική που χρησιμοποιούμε, είναι η συνεργατική διήθηση. Η συνεργατική διήθηση προσμετρά τις πληροφορίες που αφορούν στον χρήστη και τις προτιμήσεις ομοίων χρηστών. Ο στόχος είναι να βρει χρήστες με παρόμοια γούστα με τον εξεταζόμενο και να προτείνει αντικείμενα με βάση την ανάλυση των προτιμήσεων αυτών των χρηστών.

Θα χρησιμοποιήσουμε τις παραπάνω τεχνολογίες, με σκοπό να αναπτύξουμε μια προοδευτική εφαρμογή, η οποία θα επωφελείται από αυτές. Θα παρουσιάσουμε επίσης τα αποτελέσματα του συστήματος που θα αναπτύξουμε. Πώς λειτουργεί, τι αντιμετώπιση έχει από τους τελικούς χρήστες, αν ο αλγόριθμος της συνεργατικής διήθησης που χρησιμοποιείται αποδίδει ικανοποιητικά, και αν καλύπτει τις ανάγκες των χρηστών. Από την ανάλυση των αποτελεσμάτων προκύπτει ότι, το σύστημα πλησιάζει αρκετά τις προτιμήσεις των χρηστών. Παρόλα αυτά, υπάρχουν αρκετές βελτιώσεις που μπορούν να πραγματοποιηθούν και τις οποίες θα συζητήσουμε.

*Επόπτης: Δημήτρης Πλεξουσάκης*

Καθηγητής



*Στον Φάνη μου για την αγάπη, την υποστήριξη και τη συμπόρευση του μαζί μου  
στην πνευματική ανάπτυξη*



## Ευχαριστίες

Ιδιαίτερες ευχαριστίες αξίζουν στον επόπτη μου κ. Δημήτρη Πλεξουσάκη, για όσα μου προσέφερε στο διάστημα της συνεργασίας μας και για τις ευκαιρίες που μου έδωσε. Χωρίς την ουσιαστική του καθοδήγηση και τις επισημάνσεις του η ολοκλήρωση αυτής της εργασίας θα ήταν αδύνατη.

Ακόμα θα ήθελα να ευχαριστήσω τον κ. Ιωάννη Στυλιανού καθώς και τον κ. Ιωάννη Τζίτζικα για την προθυμία τους να συμμετάσχουν στην επιτροπή για την αξιολόγηση της εργασίας αυτής καθώς και για τις επισημάνσεις τους πάνω σε αυτή.

Ένα επίσης ευχαριστώ θα ήθελα να απευθύνω στην κ. Ρένα Καλαϊτζάκη, γραμματέα μεταπτυχιακών σπουδών, για την ευγένεια και κυρίως την υπομονή της απέναντι μου όλα αυτά τα χρόνια.

Ένα μεγάλο ευχαριστώ ανήκει επίσης στους δασκάλους μου της μουσικής οι οποίοι με μύησαν στα μυστικά της, μου μετέδωσαν την αγάπη για αυτήν και με παρακίνησαν να τολμήσω να συνδυάσω την αιώνια αυτή τέχνη με την σύγχρονη επιστήμη της Πληροφορικής.

Θερμές ευχαριστίες εκφράζω επίσης σε όλους όσους συνέβαλαν στην ολοκλήρωση αυτής της εργασίας, για την συμβολή και συνεργασία τους, την υπομονή τους, την πρόθυμη συμμετοχή τους στην πειραματική χρήση και αξιολόγηση του συστήματος, τον πολύτιμο χρόνο τους, τις επισημάνσεις και τις ουσιαστικές παρατηρήσεις τους.

Άπειρη αγάπη κι ευγνωμοσύνη ανήκει στην οικογένειά μου και πιο συγκεκριμένα στους γονείς μου Κωνσταντίνο και Ειρήνη καθώς και στα αδέρφια μου και ιδιαίτερα στον αδερφό μου Χαράλαμπο που ήταν πάντα δίπλα μου και με στήριξαν σε όλες τις δυσκολίες. Η δύναμη που μου έδωσε η συνεχής υποστήριξή τους όλα αυτά τα χρόνια δεν μπορεί να περιγραφεί σε λίγες γραμμές.

Τέλος, εκφράζω την άπειρη αγάπη κι ευγνωμοσύνη μου προς τον Φάνη, χωρίς την παρουσία, την συμμετοχή, την καθοδήγηση, την υπομονή και επιμονή του οποίου η εργασία αυτή δεν θα είχε ολοκληρωθεί.



## Contents

### Chapter 1: Introduction

1.1	Preface.....	1
1.2	Contributions.....	2
1.3	Structure.....	2

### Chapter 2: The World of Byzantine Music

2.1	Introduction.....	4
2.2	Byzantine Music Chants (Music Types) .....	4
2.2.1	The Irmologiko.....	5
2.2.1	The Stichirariko.....	6
2.2.1	The Papadiko .....	7
2.3	Sounds (Makams) .....	7
2.4	Rythm .....	9

### Chapter 3: Supporting Technologies

3.1	Web Services .....	11
3.2	What Is a Web Service? .....	11
3.2.1	Web Services Architecture .....	12
3.2.2	Web Services Layers .....	13
3.3	Standard Technologies .....	14
3.3.1	Web Services Description Language (WSDL).....	14
3.3.1.1	Service Interface Description Layer .....	14
3.3.1.2	Service Implementation Description Layer .....	15
3.3.2	Simple Object Access Protocol (SOAP) .....	15
3.3.3	Universal Description Discovery and Integration (UDDI) .....	17
3.3.4	Publishing and Finding WSDL Service Descriptions in the UDDI registry .....	17
3.4	Semantic Web.....	18
3.5	Semantic Web Technologies - The Semantic Web Stack.....	18
3.5.1	Uniform Resource Identifier (URI).....	19
3.5.2	eXtensible Markup Language (XML).....	19
3.5.3	RDF and Ontology Languages .....	20
3.5.3.1	OWL-Lite .....	21

3.5.3.2	OWL-DL.....	21
3.5.3.3	OWL-Full .....	21
3.5.4	Rule Layer .....	21
3.5.5	The Top Layers.....	22
3.5.6	Semantic Web Query Languages.....	23
3.5.6.1	SPARQL .....	24
3.5.6.2	TRIPLE .....	24
3.5.6.3	ICS-FORTH RQL.....	25
3.5.7	Semantic Web Database Storage .....	25
3.5.7.1	ICS-FORTH RDFSuite .....	25
3.5.7.2	Sesame.....	25
3.5.7.3	Jena.....	26
3.5.8	Semantic Web and Relational Databases.....	26
3.5.8.1	D2RQ.....	26

## Chapter 4: Recommendations

4.1	Introduction.....	28
4.2	Content-based Recommendation Systems .....	28
4.3	Collaborative filtering Recommendation Systems .....	29
4.3.1	User-Based Filtering.....	29
4.3.2	Item-Based Filtering .....	31
4.3.3	Hybrid Method .....	32
4.3.4	Ratings .....	32
4.3.5	Computing Similarity .....	33
4.3.5.1	Cosine Based Similarity.....	33
4.3.5.2	Correlation Based Similarity .....	33
4.3.5.3	Mean Squared Difference.....	34
4.3.6	Limitations .....	34
4.4	The Usage of Technology in Byzantine Music Area.....	35

## Chapter 5: The Byzantine Music Framework

5.1	Introduction.....	36
5.2	Architecture.....	36
5.3	Implementation.....	37
5.3.1	Database Storage.....	37

5.3.2	Web Services Implementation .....	37
5.4	Semantic Analysis of Byzantine Music - RDF Schema Production.....	44
5.5	Recommending Music Files Based on Collaborative Filtering Method.....	52
5.5.1	Principal Component Analysis (PCA) .....	53
5.5.2	User Mapping .....	53
5.5.3	Personality Diagnosis.....	53
5.5.4	Normalization .....	55
5.5.5	Web Service Support .....	55

## **Chapter 6: Experimental Results**

6.1	Introduction.....	57
6.2	The final application .....	57
6.2.1	Content Management System.....	57
6.2.2	The Web Site.....	61
6.3	Extracting the Semantic Data .....	66
6.4	Dataset.....	68
6.5	Collaborative Filtering Algorithm Evaluation .....	68
6.6	User Evaluation.....	78
6.6.1	The Questionnaire .....	78
6.6.2	Users' Opinion .....	80

## **Chapter 7: Conclusions**

7.1	Summary.....	81
7.2	Future Extensions .....	82

## **Bibliography**

## **Appendix**

## List of Figures

Figure 1 Roles, operations and crafts of Web Services .....	12
Figure 2 Web Services Layers .....	13
Figure 3 SOAP Envelope structure.....	16
Figure 4 XML message using SOAP.....	16
Figure 5 The Semantic Web Stack .....	19
Figure 6 The user-item matrix representing the ratings users give to items .....	30
Figure 7 (a) The user-item matrix, showing how similar users are found (b) The ordering of similar users is then used to predict the rating of the test item with some weighting.....	30
Figure 8 (a) The user-item matrix, showing how similar items are found (b) The ordering of similar items is then used, to predict the rating of the test item with some weighting. ....	31
Figure 9 Using both similar items and similar users to predict the missing rating values .....	32
Figure 10 Web Framework Architecture .....	36
Figure 11 Database Schema .....	37
Figure 12 Music Types rdf-schema extension graph.....	47
Figure 13 Makams rdf-schema extension graph .....	49
Figure 14 Musical instruments rdf-schema extension graph .....	51
Figure 15 CMS logon screen .....	58
Figure 16 Menu bar .....	58
Figure 17 Music files search results.....	59
Figure 18 Music file edit form .....	60
Figure 19 Music file to albums correlation.....	61
Figure 20 The search engine first page.....	62
Figure 21 Makams' selection tree .....	63
Figure 22 Results page.....	64
Figure 23 Rating a file .....	64
Figure 24 Presentation page with the list of recommendations.....	65
Figure 25 Predicted value computation .....	65
Figure 26 D2RQ start page.....	66
Figure 27 Music files list .....	67
Figure 28 Music file #100 particular page .....	67
Figure 29 Predicted value distribution for item #1 .....	75
Figure 30 Predicted value distribution for item #76 .....	77





# Chapter 1

---

## *Introduction*

### **1.1 Preface**

Greek music (byzantine and traditional) is an important topic in local as well as international history. Research done in this section has proved that byzantine music has its root in ancient Greek music and consequently is of high importance.

There are too many characteristics of high significance that distinguish and differentiate this music from the music of the rest of the world. There are also too complex issues that make the study and organization of it very hard.

Nowadays, since the development of technology allows the management of the mass and variety of information, we try to offer a solution that respects the specificity of the music and makes easier the documentation, organization, search and promotion.

The technological approaches that are tried in this work - applications include the development of an information system with the design of a database system, the usage of web services [1], the semantic notation [15], analysis and publicity of data and the utilization of recommendation algorithms [64].

Analytically, the development of the information and database system includes the basic organization and management system. Data and metadata of each music file are analyzed, classified and form the structure of the database system, so that to retain data completeness and on the other hand provide improved search and management techniques. The Content Management System (CMS) enables the user to edit information elements dynamically.

The above system is based and operates on web services. Thus, a quick and interoperable application is provided, that is able to be adapt to every new technological request.

In parallel, we are also attempting to provide a semantic approach to the data, in order to make information more understandable and accessible by the machines (computer systems). In this direction, we have extended the music ontology, developed by kanzaki [80] and make use of D2RQ server [87] [88] to convert data from their relational form to semantics and make them accessible to the public in the web.

Finally, a smart algorithm has been developed in order to achieve efficient and quick search results, based on collaborative filtering techniques [75] [76] [81] [83]. This algorithm utilizes users' preferences data, so that to reason an additional information (music file) as "interesting" and recommend it to the user.

## 1.2 Contributions

The primary contributions of this thesis are:

- A formal notation of the web services theory, the basic technologies of the semantic web, the recommendation framework and a relevant bibliography on the current state-of-the-art technologies that have recently emerged
- The development of an active real-world framework that takes advantage of the above technologies in order to promote byzantine music, a field in which these issues have not yet been technologically studied
- The development and assessment of a methodology to improve our online application based on collaborative filtering
- A roadmap for future extensions in recommendation technologies, music's semantic notation, and the usage of web services in system development

## 1.3 Structure

In this thesis we study three different technologies: web services, semantic web and recommendations, and attempt to join them into a musicological system, specialized in byzantine music.

Chapter 2 introduces byzantine music theory, and analyzes the basic points that differentiate this music from the rest and the need for a deep study of this music, in order to develop the right supporting application.

Chapter 3 overviews web services technologies, their supporting framework and architecture and the components which are essential for the application and use of the web services.

Chapter 4 covers semantic web theory. It presents the basic architecture, the supporting technologies, and the approaches developed to promote and establish semantic web.

Chapter 5 discusses recommendation issues, focuses on collaborative filtering techniques and presents the work that has been done till now, and its limitations as well.

Chapter 6 explains how we have worked in order to combine the technologies discussed above, so that we could develop our byzantine music framework. It presents analytically, with code examples, the design of the system, the role of web services, the semantic notation of the data and the extension that has been done to the current music schema, and finally the development and usage of our collaborative filtering algorithm on our selected dataset.

Chapter 7 presents the results of our work, the final application, evidence of the efficiency of our algorithm and users' opinion about the application tested.

Chapter 8 summarizes the contribution of this thesis to the study of computer technologies as applied to musical content, discusses ways to extend the capabilities of the framework used and suggests directions for further research.

Finally, there is a chapter containing the details of bibliographic references given throughout this thesis.

# Chapter 2

---

## *The world of Byzantine Music*

### **2.1 Introduction**

Music belongs to Fine Arts and it is the oldest of all Fine Arts. It is the means through which people worshiped their Gods, expressed their feelings and ideas, and made them feel in full their joyful and sad events of their lives [102].

Music took its name from the Mousess of Ancient Greeks, who seemed to have cultivated this art more than all the other ancient people. The great ancient philosophers, mathematicians and music, such as Pythagoras, Plato, Aristotle, Aristoksenos, Tarantinos, Euclid and many other set the bases of the theory of music. All these great people dealt with the theory of music and many composed notable opuses about it. In practice, ancient Greeks' music was expressed through the various religious events and through the plays of the great ancient composers, tragedies and comedies. Through Alexander the Great campaign the whole Greek music culture was spread across Asia [101].

Byzantine music has a long-term course and evolution throughout the centuries as well as its roots stem undoubtedly from the ancient Greek music. 'Byzantine Music' is prevalently characterized the music of the Orthodox Greek church for the reason that this type of music was primarily grown and developed in Byzantium during the prime of Byzantine Empire. However, it should be mentioned that apart from the ecclesiastic music, it was also played the secular Byzantine music which was used for various other events (or celebrations) in Byzantium. This type of music was also spread to other Balkan and Russian countries together with the spread of Christianity [97].

### **2.2 Byzantine Music Chants (Music Types)**

Throughout the centuries the needs for the Christian worship resulted in a considerable variety of chants composed in many different forms and ways, some of them in prose style and some other in the form of a poem, all of them written and set to music by various poets and musicians.

The Byzantine ecclesiastic chants or hymns are divided into many particular categories and each one has a characteristic name such as kerkagaria, stichira, prosomia, doxastika, apolitikia, kathismata, theotokia, evlogitaria, antifona, polielei, kanones, hermi, katavasies, martirika,

megalitaria, exapostilaria, eni, doxologies, typical, makarismi, kontakia, Cherouvika, kinonika, asmata, trisagia, kratimata, kalofoniki hermi etc. All these names were given to the hymns and the chants

- by specific and characteristic words which are included in them; these are the names of evlogitaria, cherouvika, doxologies kekragaria, pasapnoaria, megalitaria trisagia allilouaria etc.
- by the names of people the hymns refer to; these result to the names of Theotokia, martirika etc.
- by the rhythms which are chanted before them; these are the names of stichira, doxastika, megalitaria etc.
- by the order of the hymn chanted during the service; these are the names of the stichira and doxastika of the Evening service, the heni or the aposticha which are namely chanted during the Evening service, the heni or the aposticha respectively, as well as the litourgika, the apolitikia etc.
- by the movements, the holy acts and the ritual which is performed by priests and the people during their psalmody; this is the result of the names of isodika, kinonika, kathismata, katavasis etc [101].

Other reasons that the chants of the Byzantine music are divided, include that some have their own melody which is special and distinctive and these are called idiomela, while others are accompanied by a melody (that) they lend to other chants and are called aftomela. The chants which borrow their melody are called prosomia. To explain further, prosomia are chanted by the melody of aftomela according to which they are composed [97] [98] [99] [101].

In Byzantine Music, melopiia (musical composition) is categorised in three basic kinds, which differ to each other on the melodic line (development, plot, extent) and the temporal conduction. All the chants and the musical parts belong to these kinds and are named as following:

### 2.2.1 The Irmologiko

It is called by that name because in this kind the hermi (aftomela) are basically stressed, that is to say that the first hymns of each chant of the multi-versed poetic ecclesiastic kind, the Kanonas.

The irmologiko form, because of its short and simple melody lacking melodic extension, is considered the oldest and most authentic Byzantine chant in which most elements of ancient Greek music are saved. This "is further supported by the fact that through this form the hermi and

the prosomia are basically produced, whose music was probably old, widely spread and known to people before Christianity; when Christianity appeared the Fathers of the Church started using it with new texts for obvious reasons" [102].

This form is distinguished in short, slow and sweet-voiced chant.

The first one is the shortest in melodic and temporal conduction of all the other forms of melopiia, and the rate of musical notes and text is usually a syllabus per a note (beat).

The slow heermologiko kinds almost by the rule temporal extention of the short one approximately double in size.

The sweet-voiced hermologiko kind was named as such because it is a very elaborate and completely artistic composition requiring knowledge and a perfect sweet voice. It is a kind of melopiia which was developed later, from the 17<sup>th</sup> century [102].

The short hermologiko kind includes kanones, hermi, katavasis, stichira, prosomia, apolitikia, kathismata, antifona, megalinaria, exapostilaria, doxologies, typika, makarismi, kontakia etc.

The slow katavasis belong to the slow hermologiko chant.

Finally, the sweet-voiced chant includes only hermi of kanones [101].

### **2.2.1 The Stichirariko**

It was named as such because of the Stichirari, the old book. The origin of this name is based on the fact that in front of these hymns, which the specific book included, there was a psalmody-like line. It is divided into new short, new slow and old chant.

In the new short stichirariko the rate between the musical notes and the syllables of the poetic text is approximately the same with the short hermologiko.

In the new slow stichirariko the rate is two or three approximately musical notes in one syllable of the poetic text.

The old stichirariko differs from the new one in extent – since each syllable spends four up to eight musical notes -, in plot, in composition and in the positions, but share with each other the rest characteristics [102].

The stichirariko kind is composed doxastika, stichira, idiomela, heni, kathismata, exapostilaria, antifona, hesodika, etc. which are set to music [101].

### 2.2.1 The Papadiko

The papadiko kind was named as such probably because its chants (melodies), due to their temporal length, give time to the priest to read his secret wishes in the sanctum. Also -probably because we find it in the old books- the Papadikes.

The papadiko is the most extensive of the other kinds. In this kind only one syllable can match with a whole musical phrase or period.

It is distinguished in a definite papadiko chant (melody) in which they are set in music asmatika, allillouaria, cherouvika, kinonika etc., in a chant of Kontakati or Hikimatario, in which Kontakia and Hiki were set in music, in a chant of Mathimatario, which includes anagrammatizations, stichira theotokia, kalofoniki verses of polielei, Makarios Anir, Amomos etc., and finally in a chant of Kratimatario with Kratimata [102].

Another kind which is considered coming from ancient origins and is used in worship –because of its simplicity- since the first period of Christianity, is called the emmelis apaggelia (recitation). Another name of this kind is the ekfonitiko or logaidiko chant (melody).It is chanted in the diatoniko family and usually during the klition chant (with the kliton tone in a stable tonal basis).

The evaggelika and apostolika reading passages, the kontakio of the Akathistos hymn, the Aspile Amolinte, the "Ke dos imin Despota" as well as the deliveries of the priests in various services are delivered according to it [101] [102].

## 2.3 Sounds (Makams)

The musical tradition of the East has its roots in the 'Tropous' (Modes/Makams) of the ancient Greeks' music and it is promoted mainly by the people who live in a geographical area covering the eastern Mediterranean up to Persia, India and China. Main characteristic of this music is the 'tropikotita' (modality) contrary to the Western music whose feature is 'tonikotita' (tonality). "Modality" indicated that a chant presents a particular musical behavior; the melody is developed in a particular 'mode', whereas "tonality" is based on the harmonious thought, a sequence of chords.



The various chants and hymns of the so-called Eastern Music are chanted, sung and played in various modes. The freedom of expression led this music to the creation of many modes or 'roads' contrary to the West where only two modes remained, the major and the minor one [102].

The Byzantine Music follows the eastern modality. The modes in Byzantine Music are called Echi (Sounds). As a result, sound is "the progress of a chant which is developed as a particular melodious mode – a mode on the basis of which the various chants are sung" [102].

The sounds that are officially used by the Byzantine ecclesiastic music are eight. The main four ones (First, Second, Third, Fourth) and the indirect four (Indirect of the First, Indirect of the second, Strong, Indirect of the Fourth).

Each sound has some characteristic features which differentiate it from the other ones. These characteristics are called 'systatikas (components) of the sound and are the following according to Chrysanthos: 1) Apichima, 2) Scale, 3) dominant notes and 4) endings [103].

1. The apichima: it is called a short musical phrase which is said before chanting a chant and it is characteristic for each sound. The used words sometimes give the name of the sound that uses it; as a result, there is the sound D' Agia (apichima Agia), or the sound legatos (apichima legatos).
2. The scale: each sound uses a musical scale whose extent, the whole or a part of it, is used to sing a chant. This scale, depending on the kind in which the chant belongs to, can be diatonic, chromatic or harmonious. Each scale has a first basic note which is called the basis of the sound. However, the sounds can use more than one note as a basis, so each sound is divided to sub-sounds. The note that a sound uses as a basis is in accordance with its name. For example, if the sound uses as a basis a note which is in a distance of three voices (= intermediate spaces between the notes) above the ordinary basis of the sound, it is called three-part sound, four, four-part sound, five, five-part sound etc., or reversely if it uses a note which is in a distance of two voices below the ordinary basis of the sound, it is called medium, three ring finger, four indirect. [102] [95]

A sound (sub-sound) can be also named by the name of the note which uses as a basis (the Byzantine music uses seven notes which are named by the letters of the Greek alphabet: πΑ, Βου, Γα, Δι, κΕ, Ζω, Νη) e.g. First of the κΕ (Based on κΕ), First of the πΑ

(based on  $\pi A$ ), Indirect of the Fourth from the  $N\eta$  (based on  $N\eta$ ), Indirect of the Fourth from the  $\Gamma\alpha$  (based on  $\Gamma\alpha$ ) etc.

3. The dominant notes: Dominant (main) notes in a scale are called the notes which are heard more and distinguish themselves during the performance of a chant. Each sound –and its sub-sound- has its own dominant notes creating by this way its own modal behaviour.
4. The endings: Endings in a sound are the end of a musical phrase or period, and correspond to the punctuation of the verbal speech. So, as in verbal speech, we use the comma, the semicolon, the full stop, correspondingly in music we use endings which are incomplete, complete, final and final to pause. These endings basically appear in the note of the basis and in the dominant sounds. The incomplete endings appear in one of the dominant notes, the complete endings appear in between the chant on the basis of the sound, while the final and the final to pause endings appear in the end of the chant and also on the basis of the sound, unless the sound mediates, is a duet, a trio or in four parts [102] [93] [94] [99] [100] [101] [103].

In accordance (approximately) with the sounds of the Byzantine Music are the makamia or roads (which) is used by the musicians of Arab-persian as well as Greek traditional music (folk, island, Asian minor, and of rebetis). There is a huge number of makamia named such as Usak, Chuzam, Chuseini, Sabach, Rast, Chitzaz, Chitzazkar, Nichavend etc [94] [96].

## 2.4 Rythm

In music each melody is divided into smaller sections which help us count easily the beats and mainly to perceive its rhythm. These sections are called bars. Rhythm in music is called the division of the chant in a specific order into parts which always consist of equivalent number of beats [101].

In Byzantine Music the division of the bars is done in a particular way. The rhythm of the ecclesiastic hymns is tonal. That is to say, during the division of the bars the stress of the words of the text has to coincide with the strong part of the bar. This way of musical bars division has as a result a "creation of bars of different temporal value, since also the words of the text do not always have equal syllables, same stress and beats. So, contrary to what happens to the European music, the rhythm of the ecclesiastic chants is not homogeneous, but mainly mixed. Besides, if the

words have big melodious or syllabic length, then it is impossible for us to apply the tonal rhythm. In that case, it is applied the musical or the chant stress" [102].

Contrary to our ecclesiastic music, the songs of our traditional and folk culture keep stable rhythm from the beginning to its end, for that they can be danced. Depending on how many beats each bar lasts, according to which a melody of a song is divided, there are rhythms which are 2-beat, 3-beat, 4-beat, 5-beat, 6-beat, 7-beat, 12-beat etc.

# Chapter 3

---

## *Supporting Technologies*

### **3.1 Web Services**

The development and usage of services by enterprises and academic organizations has increased exponentially in recent years, enabled by factors such as the maturity and ubiquity of the HTTP protocol, and the improvement in secure communication technology. However, if somebody would like to use a service, one would need to study the way in which he/she is going to call it, to check the communication protocol, and to adjust his own system so that it is compatible with the one of the service provider.

Heterogeneity and scalability lead to a new architecture. There has already been an improvement on these issues recently, since the majority of developers make use of a common architecture of development, publication and utilization of the services [1].

### **3.2 What Is a Web Service?**

Web services are distributed to software components that provide information to applications rather than to human beings, via an application-oriented interface. They describe a collection of operations that are accessible via the network by throwing standard XML [2] messages. Web services allow service integration in a fast, effective way [3], [4].

A Web Service is a software component that is independent of implementation and platform. It can be described as using a service description language, published in a registry and discovered by a standard mechanism. Thus, it can be invoked by an Application Program Interface (API) via the network, built in round of other services.

Web Services make use of the best characteristics of the Web-based component development. They contain reusable black-box functionalities implemented independently of the programming language used [5].

Developers release their services via Internet, so that to be used by the community.

In order to make a service available on the Web the following are usually necessary:

- Publishing the service interface, thus allowing users to find and use it.

- Accepting requests and sending responses using the XML message protocol.
- Establishing a binding between external requests and business functions implementation.

A message-form document is used by the program in order to send a request to a Web Service via the network and (optionally) it receives some response, also in a XML-based message-form document. The usage of XML raises the importance of providing such extensible mechanism to describe the messages and its contents. Furthermore, Web-based services standards define the format of the message and the mechanisms to publish and discover Web Services interfaces.

### 3.2.1 Web Services Architecture

Web Service architecture meets the following issues: service interaction, message format, transport protocols and physical location.

The interface hides service implementation details, so that the web service can be re-used independently of the underlying hardware and language.

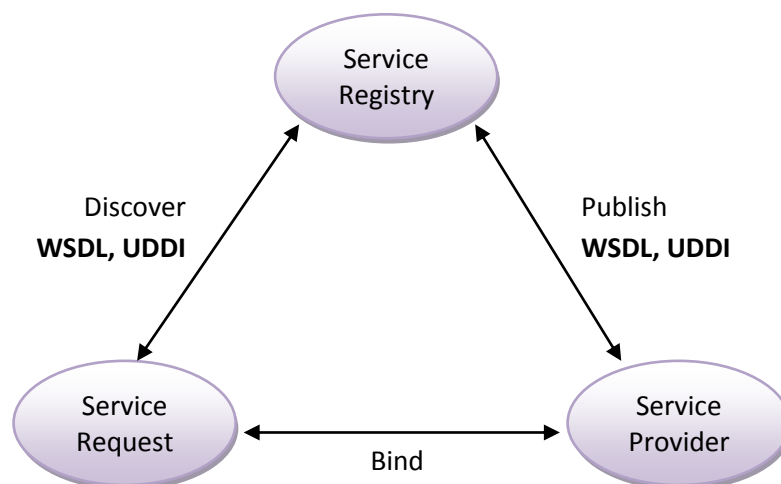


Figure 1 Roles, operations and crafts of Web Services

The figure above shows the web services architecture [6]. The basic elements of this include:

- **Service Provider:** Namely, the service owner. In architectural terms, this is the platform that receives the service requests. It creates the web service and it is responsible for its description, according to some standards, and its publication to a central registry.
- **Service Requestor:** It is any application that asks for an interaction with the web service. It could be a web browser program or another web service. It discovers and invokes web services based on service description.

- **Services Registry:** This is where service descriptions are being published by the developers. Search Requestors search Registries during development time (static binding) or run time (dynamic binding).

As mentioned above, the Service Description provides interface and implementation details, including data structures, operations and network binding information. It also contains data to accommodate search process by Service Requestors.

The main operations that are used in this architecture are the following:

- **Binding:** This operation assumes to invoke and initialize the interaction of a service with its caller, using binding information provided by the service description to locate and contact the appropriate service.
- **Publishing:** This operation is performed by the service provider, in order to contact the service registry and publish the service.
- **Discovering:** A service requestor finds a description of a service or queries a service registry for the required service type. This can be done in either runtime or development time. Following, the necessary information to invoke the service is contacted.

### 3.2.2 Web Services Layers

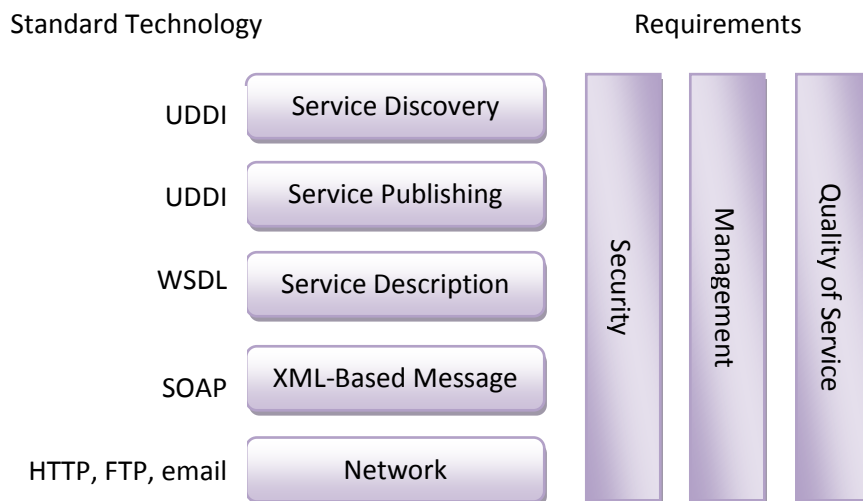


Figure 2 Web Services Layers

Figure 2 displays the web services conceptual layers described below [6], [7]:

- **Network Tier:** This is the base protocol layer. It is used to meet the needs of the application such as security, availability, performance, reliability.

- **XML-Based Message:** The standard technology used in this layer is SOAP protocol which stands for exchange information.
- **Service Description:** This is done using Web Services Description Language (WSDL). This language is the appropriate one for the definition of the interface and the interaction mechanisms of the service.
- **Service Publishing and Discovery:** These two layers use the UDDI standard in order to discover and publish information regarding Web Services.

### 3.3 Standard Technologies

In this section we discuss the basic technologies regarding web services structure, thus WSDL, SOAP and UDDI.

#### 3.3.1 Web Services Description Language (WSDL)

WSDL is an XML-based language used to describe services as a collection of accessible operations through messages [8]. It defines interfaces and service interaction mechanisms, binding protocols and network services details. It also represents additional descriptions such as context, Quality of Service and service-to-service relationship. It enables one to describe an object in a transparent, implementation-independent way.

WSDL allows service description to be done in two parts: Firstly, in the Service Interface Description Layer and, secondly, in the Service Implementation Description Layer [6] [8].

##### 3.3.1.1 Service Interface Description Layer

In this layer a WSDL service definition is provided. Thus, the service can be used, instantiated and referenced by other services [9]. Analytically, there are the following definitions:

The WSDL: *binding* which describes some basic elements such as the protocol, data format, security, etc. The WSDL: *port Type* informs the operation elements. The WSDL: *message* tag defines input and output data. Complex data types inside a message are being defined by the WSDL: *type* tag.

Finally, the role of an interface file is to describe the web service, the methods used, the specific parameters and the utilized codification.

### 3.3.1.2 Service Implementation Description Layer

There is a WSDL file that describes the implementation of the service. The service contains a WSDL: *port elements* collection with a WSDL: *binding* interface-definition element. There is also described the location of the web service, as well as the way it could be accessed by the requestors. Besides, the WSDL file provides information regarding the protocol bindings and the message formats, such as SOAP, HTTP GET/ POST and MIME [8].

### 3.3.2 Simple Object Access Protocol (SOAP)

SOAP is a protocol that allows the communication between various applications by exchanging information in distributed environments [10]. XML-format messages are exchanged in this communication, which include calling parameters and result requirements. As a result, the messages can be understood independently of platform, operating system, programming language or network [11]. The same protocol can also be used to publish, locate and call the service through UDDI registry.

The SOAP data model offers data type definition for the widely used structures such as string, integer, float, double, date etc. The process that translates data to XML is called codification. The protocol also supports remote procedure calls (RPC) and asynchronous messaging.

The parts of the SOAP package are the following:

- SOAP Envelope: The framework that contains the message content.
- SOAP Codification: Serialization mechanisms that could be used to exchange instances or data types defined by an application.
- RPC SOAP: The description of the way that remote procedure calls and responses inside the message are being encapsulated.
- Binding Framework and SOAP Transport: An abstract framework to exchange SOAP envelopes between envelopes.

Some other concepts that are also important include:

- SOAP Client: It is a process that defines the necessary information to invoke a method in an XML document.
- SOAP Server: It executes the SOAP messages and interprets the documents.
- Mensagem SOAP: The basic communication form between SOAP nodes.



The structure of the SOAP envelope is represented in figure below [10]. The SOAP block is a syntactic construction that encloses data which constitute one computational unit. The block is identified by a URI. The SOAP header is a collection of some or no blocks. Afterwards, they could be redirected to some known SOAP receptor inside the message's path. The SOAP body is a collection of blocks redirected to the last SOAP receptor.

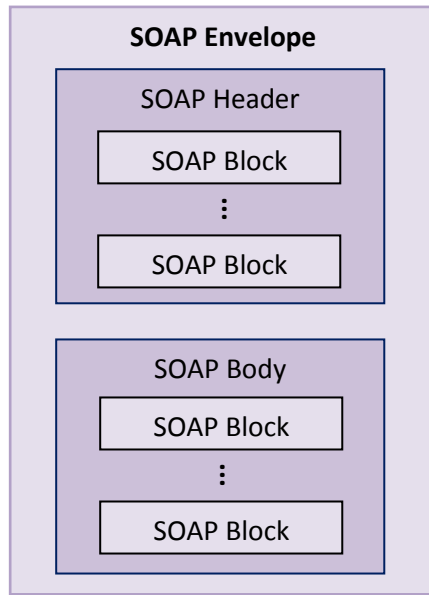


Figure 3 SOAP Envelope structure

The invocation of the service using SOAP occurs as follows [3]: The Application requests a SOAP message and invokes the service operation through Web Service Provider. The message of the Service Requestor includes the provider's network address. The network infrastructure ships the message to a SOAP server, which redirects it to the web service provider. The Web Server is responsible for processing a request message and then building a response, which is redirected through the SOAP infrastructure. When a XML message arrives at the requestor, it is converted to a programming language to be shipped to the application.



Figure 4 XML message using SOAP

### 3.3.3 Universal Description Discovery and Integration (UDDI)

The UDDI specification is a common effort to define and create a standard service registry, which can be accessed by clients and provide them with information about the requested service [12] [13].

The main element is the UDDI Project, which manages the UDDI business registry, which is public. All the information in this registry can be accessed through general queries. There can also be a private one registry, which adds security controls in order to provide data integrity and prevent non-authorized access. Therefore, it stores private information, and maybe a subset of the public registry.

The information in business registry can be separated into the following three categories:

- "white pages" - address, contact and identifiers
- "yellow pages" - industrial categorization
- "green pages" - information

The UDDI implementation is a server that applications ask to publish and locate web services. It contains information about businesses, the provided services and web services specifications. These descriptions are done in WSDL through UDDI registry [14].

The XML schema, which is used by the main UDDI information model, defines four types of information:

- Business information
- Service information
- Bindings information
- Service-specific information

### 3.3.4 Publishing and Finding WSDL Service Descriptions in the UDDI registry

The UDDI data entities provide support to describe information about providers and services. It also supports various kinds of service description.

A complete WSDL service description includes the service interface and its implementation document, which describes service's instances using a WSDL service element.

A service implementation is published in a UDDI registry as a `businessService` or within one or more `bindingTemplates`. The `businessService` is published by the service provider.

In case that the service interface represents a reusable definition, it can be published using a `tModel` and this must be done before the implementation's publication.

### 3.4 Semantic Web

The Semantic Web is the next generation of the World Wide Web in which the meaning (semantics) of information and services are defined, thus enabling computers to understand and satisfy people's requests and use of the web content. The founder of Semantic Web, Tim Berners-Lee, considers the Web as a universal medium for data, information, and knowledge exchange [15].

Humans are capable of using the Web in order to carry out tasks such as finding information about kangaroos, reserving a library book, buying a DVD at the lowest price etc. However, these tasks cannot be accomplished by a computer without human interference since web pages are designed to be read by people. Semantic web trends to make this content accessible by computers, so that they can perform finding, sharing and combining information.

Semantic Web application areas are receiving intense interest because of the rapid growth in the usage of the Web, together with information content technologies. The Semantic Web is regarded as an integrator across different content and information applications and systems. In a paper presented by Gerber, Barnard and Van der Merwe [16], the Semantic Web landscapes are charted and a brief summary of related terms and enabling technologies are presented. The architectural model proposed by Tim Berners-Lee is used as a basis to present a status model that reflects current and emerging technologies [17].

The semantic web comprises the standards and tools of XML, XML Schema, RDF, RDF Schema and OWL that are organized in the Semantic Web Stack.

### 3.5 Semantic Web Technologies - The Semantic Web Stack

The *Semantic Web Stack* is an illustration of the hierarchy of languages, where each layer exploits and uses capabilities of the layers below it. It shows how technologies, which are standardized for Semantic Web, are organized to make the Semantic Web possible. It also shows how Semantic Web is an extension (not replacement) of classical hypertext web.

The illustration was created by Tim Berners-Lee [18]. The stack is still evolving as the layers are concretized [19] [20]. Figure 5 above shows the semantic web stack.

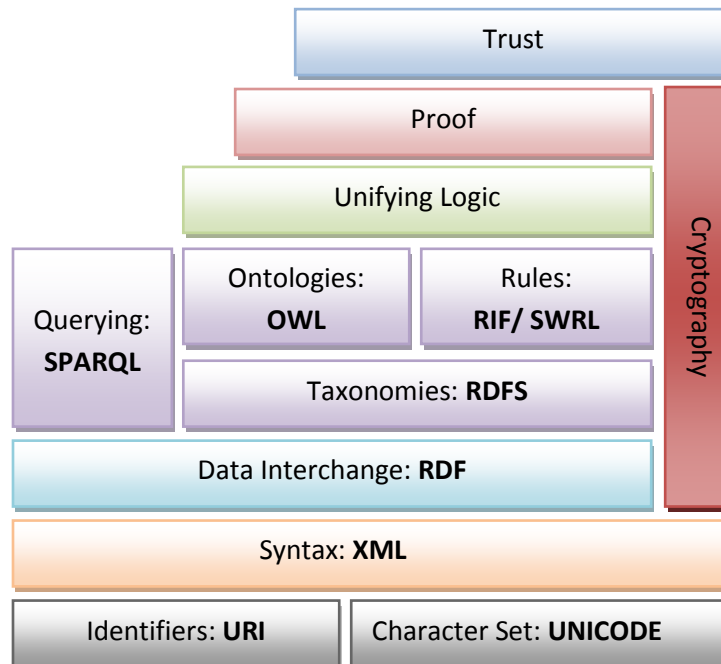


Figure 5 The Semantic Web Stack

The bottom layers contain technologies that are well known from hypertext web and that, without change, provide a basis for the semantic web.

### 3.5.1 Uniform Resource Identifier (URI)

A Uniform Resource Identifier (URI) [26] provides a simple and extensible means for identifying a resource. This specification of URI syntax and semantics is derived from concepts introduced by the World Wide Web global information initiative. The use of these identifiers dates from 1990 and is described in "Universal Resource Identifiers in WWW" [21]. The syntax is designed to meet the recommendations laid out in "Functional Recommendations for Internet Resource Locators" [22] and "Functional Requirements for Uniform Resource Names" [23].

### 3.5.2 eXtensible Markup Language (XML)

XML [24] is the base language of the Semantic Web Stack. XML is a set of rules for encoding and exchanging documents electronically. The view of an XML document includes labeled ordered trees. The labels consist of parenthesis tags. XML provides an elemental syntax for content structure within documents, yet associates no semantics with the meaning of the content contained within.

Many definitions have been developed in order to provide and restrict the structure and content of elements contained within XML documents. XML schema is a standard one [25] also known as XSD. XSD can be used to express a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema.

XML is also used for the languages syntax of the next layers of the Semantic Web. This means that the constructs of these languages are encoded as XML documents, and can be validated against the language definitions by standard validators. However, alternative syntaxes, better suitable for the human, can be provided and can be used as a starting point for defining the semantics of these languages.

### 3.5.3 RDF and Ontology Languages

The goal of the Semantic Web is to make web content meaningful, so that data is suitable for automatic reasoning. This leads to the use of some descriptive data (meta-data) in machine readable form for automatic reasoning. The simplest model of this description for expressing data models refers to objects (web resources) and their relationships. Another approach, called ontologies, is a formal representation of a set of concepts within a domain and the relationships between those concepts. This is also used to reason about the properties of that domain, and may be used to define the domain.

The Semantic Web tower introduces language layers for describing resources and for providing ontologies:

The Resource Description Framework (RDF) [27] is a family of W3C specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax formats. It is based on the idea of making statements about resources in the form of subject-predicate-object expressions. These expressions are known as triples in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion "The monkey eats fruit" in RDF is by the triple: a subject denoting "the monkey", a predicate denoting "eats", and an object denoting "fruit".

An extension of RDF is RDF Schema [28], which defines the hierarchies of concepts. Thus, RDF Schema is a vocabulary for describing properties and classes of RDF-based resources, with semantics for generalized-hierarchies of such properties and classes.

Ontologies are used to capture knowledge about some domain of interest. Ontology describes the concepts in the domain and also the relationships between these concepts. The most recent development in standard ontology languages is OWL from W3C [29]. There are three species of OWL: OWL-Lite, OWL-DL and OWL-Full. A defining feature of each sub-language is its expressiveness. OWL-Lite is the least expressive while OWL-Full is the most expressive sub-language. OWL-DL falls between OWL-Lite and OWL-DL.

#### **3.5.3.1 OWL-Lite**

OWL-Lite is the syntactically simplest sub-language. It is used when a simple class hierarchy and simple constraints are needed. For example, it is envisaged that OWL-Lite will provide a quick migration path for existing thesauri and other conceptually simple hierarchies.

#### **3.5.3.2 OWL-DL**

OWL-DL lies on the next level of sub-languages, so it is more expressive. It is based on Description Logics [29], which are a decidable fragment of First Order Logic and therefore provide the appropriate layer for automated reasoning. Consequently, it allows automatically computing the classification hierarchy and checking for inconsistencies.

#### **3.5.3.3 OWL-Full**

OWL-Full is the most expressive sub-language. It is based on a semantics different from that of OWL-Lite or OWL-DL, and is designed to preserve some compatibility with RDF Schema. For example, in OWL-Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right; this is not permitted in OWL-DL. OWL-Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for OWL-Full.

### **3.5.4 Rule Layer**

The next language level in Semantic Web Stack includes Rules which are not yet developed. The need of rules in semantic web is essential for the following reasons:

- Most of the applications are governed by rules. Therefore, a web rule language is needed to express them.
- Rules hide implementation details, since they provide a high-level description. Some of their characteristics include consistency, simplicity, good understanding, and integration.
- Rule languages overcome the inability of ontology languages to express some aspects of applications, e.g. composition of relations.

Because of the variety of needs that different applications in the Semantic Web areas have, we can conclude that a family of rule languages is needed. This effort is addressed by RuleML Initiative [31].

The Rule Interchange Format (RIF) is a standard in development within the W3C Semantic Web Activity. When completed, it will be a component of the semantic web, along with RDF and OWL. Although originally envisioned by many as a "rules layer" for the semantic web, in reality the design of RIF is based on the observation that many "rules languages" exist, and what is needed is to exchange rules between them [32].

Finally, Semantic Web Rule Language (SWRL) is a proposal for a Semantic Web rules-language, combining sub-languages of the OWL (OWL-DL and OWL-Lite) with those of the Rule Markup Language [33].

### 3.5.5 The Top Layers

As shown in figure 5, the top three layers of the Semantic Web Stack are: the unifying logic layer, the proof layer and finally the trust layer. Unifying logic and proof layers are undergoing active research. The aim is to enhance the usability and usefulness of the Web and its interconnected resources through:

- Servers, which expose existing data systems (relational dbs) using the RDF and SPARQL standards.
- Documents "marked up" with semantic information.
- Common metadata vocabularies (ontologies) and maps between vocabularies that allow document creators to know how to mark up their documents so that agents can use the information in the supplied metadata
- Automated agents to perform tasks for users of the semantic web using this data
- Web-based services to supply information specifically to agents

Finally, trust to derived statements will be supported (a) by verifying that the premises come from trusted source and (b) by relying on formal logic during deriving new information.

### 3.5.6 Semantic Web Query Languages

Several categories of query languages can be distinguished, according to the format of the Semantic Web data they can retrieve:

- Query languages for XML
- Query languages for Topic Maps
- Query languages for RDF
- Query languages for OWL

**XML Query Languages:** Although XML is not a primary format, it is possible to specify semantics using XML. Hence query languages for XML are applicable to Semantic Web data. There are two approaches that most query and transformation languages for XML use to specify the structure of the retrieved data. The first one is the navigational approach, where path-based queries over the XML data are specified. The W3C standardized languages of this approach include XPath, XSLT and XQuery. The second approach is the example-based approach. In this case, query patterns are specified as “examples” of the XML data to be retrieved. Languages of this kind are mainly research languages, with XML-QL [35] and Xcerpt [37] [38] being some well-known representative examples.

**Topic Maps Query Languages:** The most representative languages for Topic Maps data are tolog [36], AsTMA [34] and Toma [39]. Tolog is based on programming, also having SQL style constructs. AsTMA is a functional query language. Finally, Toma combines both SQL syntax and path expressions for querying.

**RDF Query Languages** can be grouped into several families, which differ in aspects such as supported ontology/metadata, data model, expressivity, support for schema information, and type of queries. The most important generation is 'SPARQL Family'. This originated with the language SquishQL [40], which evolved into RDQL [40] and was later extended to the language SPARQL [41]. These languages all 'regard RDF as triple data without schema or ontology information unless explicitly included in the RDF source'.

Another family of RDF languages, the 'RQL family', consists of the language RQL [42], and its extensions such as SeRQL [43]. The combination of both data and schema querying characterizes



this family. The RDF data model used differs from the standard data model for RDF and RDFS. The main difference is that it detects and forbids cycles in the subsumption hierarchy and on the other hand the fact that it requires both a domain and a range to be defined for each property. RQL itself has a large number of features and choices in syntactic constructs. This results in a complex, yet powerful language, which is far more expressive than other RDF query languages, especially those of the SPARQL family.

A number of other types of query languages for RDF also exist, using alternative paradigms. These include query languages using reactive rules, such as Algae [44] and deductive languages such as TRIPLE [45] and Xcerpt [37] [38].

Query languages for OWL are still in their infancy compared to those for RDF. OWL-QL (updated version of DAML) [46] is a well-known language for querying OWL. The main target of its design is to assist query-answering dialogues between computational agents on the Semantic Web. Unlike the RDF query languages, it focuses on the querying of schema rather than instance data. Although an RDF language may query OWL data, it is not designed to be aware of OWL semantics, and thus is suitable for this task.

Below, we present the most important proposals of RDF query languages.

#### **3.5.6.1 SPARQL**

SPARQL [41] has been proposed by the RDF Data Access Working Group (DAWG) of the W3C. SPARQL has facilities for: i) Extracting information from RDF sub graphs in the form of URIs, blank nodes and literals, ii) Constructing a new RDF graph using data from the queried graph, iii) Returning 'descriptions' of the resources matching a query part, iv) Specifying optional triple or graph query patterns v) Testing the absence, or non-existence, of tuples.

#### **3.5.6.2 TRIPLE**

TRIPLE [45] has been recommended by Stefan Decker and Michael Sintek. TRIPLE provides native support for resources, namespaces and abbreviations, models (sets of RDF statements) and reification. It also allows the use of rules with expressive bodies, the transformation of one or several models. TRIPLE is a syntactical extension of Horn Logic, similar to F-Logic (and SiLRI) that supports applications in need of RDF reasoning and transformation. TRIPLE provides a human-readable ASCII-syntax, as well as an RDF-based syntax for exchanging queries and rules. It supports

the definition of semantical RDF extensions in a modular way, while its main goal is to use various semantics in one inference.

### 3.5.6.3 ICS-FORTH RQL

RQL [42] is a declarative query language for querying RDF description bases. Its definition includes a set of basic queries and iterators, which constitute the necessary background for building new ones through functional decomposition. The base of RQL is a typed data model that represents properties as self-existent individuals, thus enabling the interpretation of superimposed resource descriptions. It also introduces a graph instantiation mechanism that classifies the resources in a multiple way. It adapts the functionality of semi structured or XML query languages to RDF, but it also treats properties as self-existent individuals; it exploits taxonomies of node and edge labels and allows querying of schemas as semi-structured data.

## 3.5.7 Semantic Web Database Storage

### 3.5.7.1 ICS-FORTH RDFSuite

ICS-FORTH R&D [50] [51] activities focus on high-level and scalable software tools enabling the realization of the full potential of the Semantic Web:

- The *Validating RDF Parser* (VRP): The First RDF Parser supporting semantic validation of both resource descriptions and schemas
- The *RDF Schema Specific DataBase* (RSSDB): The First RDF Store using schema knowledge to automatically generate an Object-Relational (SQL3) representation of RDF metadata and load resource descriptions.
- The *RDF Query Language* (RQL): The First Declarative Language for uniformly querying RDF schemas and resource descriptions.

### 3.5.7.2 Sesame

*Sesame* [47] [48] [49] is called an open source RDF database with support for RDF Schema inferencing and querying. Firstly, it appeared as a research prototype on behalf of the EU research project *On To Knowledge* developed by Aduna. Contemporarily, it is further developed and has become more mature. Aduna in cooperation with *NLnet Foundation*, developers from *OntoText*, and a number of volunteer developers who contribute ideas, bug reports and fixes to maintain it. Sesame has been developed and designed with flexibility in mind. Namely, its areas of use include a range of storage systems such as relational databases, in-memory, file systems, keyword

indexers, etc., offering developers a choice of tools to take advantage of the power of RDF and RDF Schema. For example, a flexible access API, which supports both local and remote (through HTTP, SOAP or RMI) access, and several query languages, among *SeRQL* being the most powerful.

### 3.5.7.3 Jena

Jena [52] is a well-known Java-based Semantic Web framework which comprises some characteristics. It supports statement centric methods for manipulating an RDF model as a set of RDF triples, centralized resource methods for manipulating an RDF model as a set of resources with properties. Moreover, it supports Cascading method calls for more convenient programming as well as built in support for RDF containers such as bag, alt and seq. Finally, another characteristic includes extensible resources, integrated parsers and writers for RDF/XML (ARP), N3 and N-TRIPLES and support for typed literals.

## 3.5.8 Semantic Web and Relational Databases

Although substantial research and development efforts have been carried out, the idea of a ubiquitous Semantic Web has not yet been fulfilled. The continuous semantic representations still come second to traditional Web pages and may be in doubt whether Semantic Web ever succeeds.

Most of the web content is being generated by Web applications having as a background databases. Scripting languages (PHP, Perl, Python, or Ruby) are used to implement these applications. Most of the time relational databases are used to store data persistently, but the basic problem with the structure and semantics, which are encoded in relational database schemes, is that the access to Web search engines and mashups is often impossible for them or they cannot be available for Web data integration. A number of components have been developed, in order to achieve the highest possible deployment of semantic technologies on the Web.

### 3.5.8.1 D2RQ

Semantic Web technologies are developing more and more. This situation results to a great necessity for RDF applications to access the content of non-RDF, legacy databases without being necessary to replicate the whole database into RDF.

D2RQ [87] [88] describes a declarative language which is used for mappings between relational database schemata and OWL/RDFS ontologies. These mappings are used by D2RQ Platforms to

give access to a RDF-view on a non-RDF database through the Jena and Sesame APIs, as well as over the Web via the SPARQL Protocol [41] and as Linked Data.

The use of D2RQ Platform may enable you to query a non -RDF database using the SPARQL query language. Also, it provides you with access to information in a non-RDF database by the use of the Jena API [52] or the Sesame API [47] [48] [49], and also to the content of the database as Linked Data over the Web. Finally, you have the possibility to ask SPARQL queries over the SPARQL Protocol against the database.

The components of the D2RQ Platform are the D2RQ Mapping Language, the D2RQ Engine and the D2R Server. More specifically, the first one is a declarative mapping language that describes the relation between an ontology and a relational data model. Following, the second one is a plug-in for the Jena and Sesame Semantic Web toolkits, which makes use of the mappings in order to rewrite Jena and Sesame API calls to SQL queries against the database. Then it passes query results up to the higher layers of the frameworks. Finally, the third one is an HTTP server that provides a Linked Data view, a HTML view for debugging and a SPARQL Protocol endpoint over the database.

There are many other active projects that provide RDF/SPARQL access to relational databases. These are analyzed as follows. Virtuoso [89] is the one that seems to make use of pretty smart rewriting algorithm and also supports Named Graphs. Triplify [90] is another active project which is defined as a small plugin for Web applications, which reveals the semantic structures encoded in relational databases by making database content available as RDF, JSON or Linked Data. SquirrelRDF [92] also supports access to LDAP directories along with SPASQL [91] which is an open-source module compiled into the MySQL server to give MySQL native support for RDF.

# Chapter 4

---

## *Recommendations*

### **4.1 Introduction**

Modern digital age requires more automated recommendation systems. People want intelligent computer systems, so that they are able to do things based on little or no input. It is essential to be able to filter out information not useful to the user, and to recommend something according to user's habits and patterns.

There are several techniques a system can use, in order to recommend items to its users. There are two main types of approaches: content-based and collaborative filtering.

### **4.2 Content-based Recommendation Systems**

Content-based recommendation systems model the content of an item and recommend items by querying a database of item properties against the preferences of the user [53]. It is based on information retrieval [54] [55] and information filtering research [56].

Traditional information retrieval approaches can be improved by the usage of user profiles (user needs, preferences, tastes). These profiles can be assumed in two ways: using questionnaires or by observing users' behavior.

There are several methods in filtering items based on content description. The words in an article [57], for instance, may constitute the content attributes when recommending articles. In another example of a movie recommendation, the attributes that are examined include genres, actors, subject matters, directors etc. of highly rated movies. So, they are recommended the movies with high degree of accordance to any of user's preferences.

Other techniques that are also being used parallel to the traditional heuristics are Bayesian classifiers, decision trees, clustering and artificial neural networks etc [58] [59]. The difference from the retrieval-based methods is that the estimation of ratings is not based on a heuristic formula, but on statistically learning the underlying data model.

## 4.3 Collaborative filtering Recommendation Systems

Collaborative filtering techniques were first introduced by Goldberg, who used this method to sort mailing list of Tapestry (a mail filtering system), so that users would only see the most interesting lists [60]. Collaborative filtering differs from content-based techniques as it takes into account both active user's information and similar users' preferences [61] [62] [63]. "Word of mouth" is the basic idea of these approaches; that is, users are more likely to trust another user's recommendations, with whom they know that they share common tastes or they had agreed in their evaluations of the items [63] in the past. The goal is to find users with similar tastes to the active user and to provide recommendation based on the analysis of the other users preferences.

Collaborative filtering drives all drawbacks that content-based filtering methods face and Shardanand spotted [53].

The characteristics that fix these problems are:

- Content-based filtering techniques can analyse information, such as text documents using a semantic model, and thus determine whether the documents are similar or not. However, it is difficult to analyse an audio file and determine information such as genre. This drawback is addressed by collaborative filtering methods since they support the filtering of items whose content is not easily analysed by automated processes.
- Content-based techniques ignore elements such as quality or taste, while collaborative filtering takes them into account in determining recommendations.
- Content-based filtering will only suggest items which the user is definitely interested in, based on the keywords used to filter [53]; collaborative filtering, on the other hand, could yield results which suggest an item the user would never have originally considered [63].

Many commerce retailers [64] such as Amazon [65] make use of collaborative filtering techniques in order to provide personalized pages to each user. Recommendation of particular items a user would like increases the possibility of selling them.

There are two main types of collaborative filtering: user-based filtering and item-based filtering.

### 4.3.1 User-Based Filtering

User-based collaborative filtering uses rating information from similar users in order to predict active user's interest in an item [66] [67] [68]. The  $M \times N$  matrix shown below represents the ratings of  $M$  users (rows) for  $N$  items (columns).

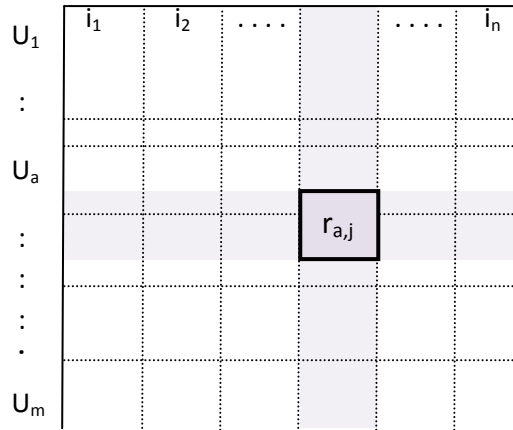


Figure 6 The user-item matrix representing the ratings users give to items

OWL-DL lies on the next level of sub-languages, so it is more expressive. It is based on Description Logics [29], which are a decidable fragment of First Order Logic and therefore provide the appropriate layer for automated reasoning. Therefore, it allows automatically computing the classification hierarchy and checking for inconsistencies.

Collaborative filtering can then be represented as the problem of predicting the missing values in the user-item matrix.

The aim of a user-based collaborative filtering is to find similar users (rows); that is, users that have rated items in a respective way. Afterwards, the technique orders users (rows) based on the level of similarity to the requested user. One can now make a prediction for the value of the unrated items based on the ratings of similar users to same items [66] [67] [69].

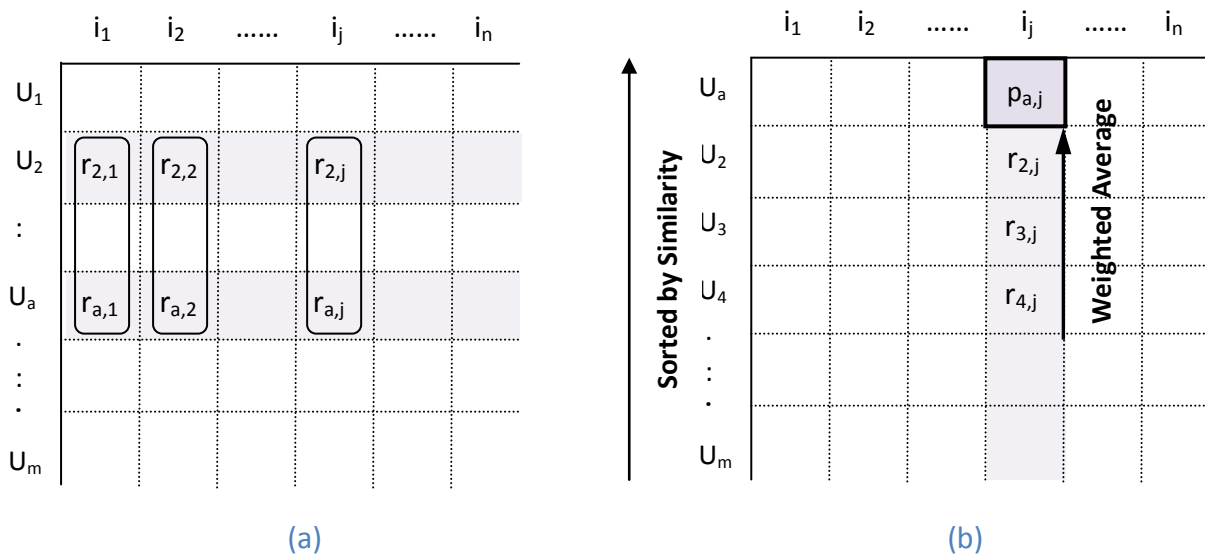


Figure 7 (a) The user-item matrix, showing how similar users are found (b) The ordering of similar users is then used to predict the rating of the test item with some weighting.

Figure 7 shows the exact process performed on the user-item matrix. Firstly, we find similar users, by detecting users that have rated the same items as the current user, as shown in (a). These users are then ordered based on the similarity value. A prediction on the rating of the test item is made by taking the average of the ratings from the similar users with some weighting based on their respective values, as shown in (b) [69].

### 4.3.2 Item-Based Filtering

Item-based approaches such as [65] [70] [71] apply the same idea as user-based collaboration, but they use the similarity between items rather than the similarity between users. Item-based filtering will take the items from the active user's profile, and find similar items to the ones the user has rated. It will then find items which have been rated similarly by other users, and then using the way those users have rated the similar items, it will attempt to predict the missing values in the user-item matrix.

As shown in Figure 8, item-based collaboration takes the ratings from the columns of the user item matrix, orders the items by similarity, and then attempts to predict the missing rating by taking a weighted average of the similar ratings [69] as with user-based collaboration.

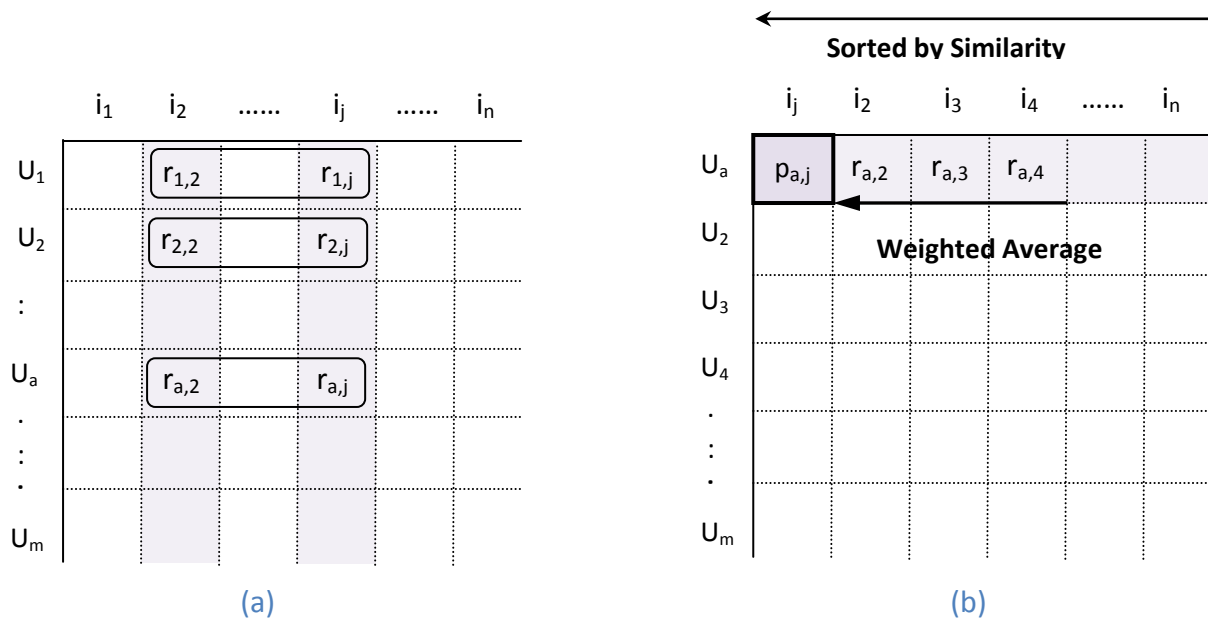


Figure 8 (a) The user-item matrix, showing how similar items are found (b) The ordering of similar items is then used, to predict the rating of the test item with some weighting.



### 4.3.3 Hybrid Method

A hybridization of the above techniques has been proposed by Wang [69]. This technique uses both user and item similarities to provide recommendations. The basic idea of this method is based on the fact that similar items rated by similar users may provide an extra source for prediction. Therefore the above collaborative filterings are combined and both pieces of information are taken into account.

Figure 9 below presents how this system works to predict ratings on the user-item matrix.

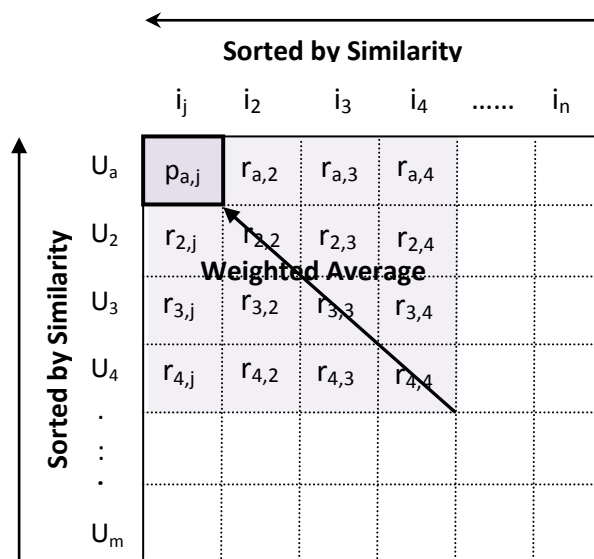


Figure 9 Using both similar items and similar users to predict the missing rating values

The columns and rows are ordered based on the similarity, so the best recommendation can be found by taking a weighted average. This also opens up the possibility that new items may be discovered, which would not have been found by performing each method on its own.

### 4.3.4 Ratings

It has already been presented that missing ratings –which users would make for items- are predicted using the user-item matrix. The user can perform explicit or implicit [63] [71] [79] gathering of rating; this can be achieved when the users either provide a rating on some numerical scale as with the GroupLens system [63], or delivers a rating from some other form of data such as purchase records with Amazon3 [65]. However, this important information is often ignored.

An accurate representation of user’s tastes is apparent when the user explicitly provides a rating on an item, given some numerical scale. On the contrary, if the user is asked to explicitly rate large databases, it is defined impractical and thus impossible to do so; not only to explicitly rate large

datasets but even a small selection of them. The best recommendation systems for the users should be those in which rating can be gathered rather implicitly, for example in Amazon4 [65]. Nevertheless, the accurate ratings for each user are necessary to achieve accurate results supplied through the already discussed collaborative filtering methods.

### 4.3.5 Computing Similarity

As we have shown above, a certain process is needed in order to compute the similarity between two users or two items. Throughout the literature, several statistical methods for computing the similarity between users/items have been described and experimented with. We will describe the main of these methods and explain them, using an example of user collaborative filtering, since computing the similarities between items is commensurate. The similarity between users  $i$  and  $j$  can be found by using item ratings from co-rated items.

#### 4.3.5.1 Cosine Based Similarity

Two users can be considered to be two vectors in the  $N$ -dimensional item space, where the similarity between them is the cosine angle between the two vectors [70] [71]. Formally, considering two users  $i$  and  $j$ , with  $\cdot$  denoting the dot product, the similarity  $\text{sim}(i, j)$  can be computed by the following relation:

$$\text{sim}(i, j) = \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{\|\mathbf{r}_i\| \|\mathbf{r}_j\|} \quad (4.1)$$

#### 4.3.5.2 Correlation Based Similarity

The Pearson  $r$  correlation coefficient is a standard correlation calculation. However, it only provides accurate result in the case of co-rated items, that is items that both users  $i, j$  have rated. The relation that gives the similarity  $\text{sim}(i, j)$  in this case is the following [53] [66] [71] [72]:

$$\text{sim}(i, j) = \frac{\sum_{t \in I} (r_{i,t} - \bar{r}_i)(r_{j,t} - \bar{r}_j)}{\sqrt{\sum_{t \in I} (r_{i,t} - \bar{r}_i)^2} \sqrt{\sum_{t \in I} (r_{j,t} - \bar{r}_j)^2}} \quad (4.2)$$

where  $I$  denotes the set of items rated by both users  $i$  and  $j$ ,  $\bar{r}_i$  is the overall average rating user  $i$  has given, and  $r_{i,t}$  denotes the rating given by user  $i$  on item  $t$ .

### 4.3.5.3 Mean Squared Difference

Mean Squared Difference is used as a measure of the difference between two vectors that may vary, and in the case of user-based filtering, can be calculated with [53],

$$sim(i, j) = \frac{\sum_{l \in I} (r_{il} - r_{jl})^2}{|I|} \quad (4.3)$$

where  $I$  is the set of common rated items by users  $i$  and  $j$ .

### 4.3.6 Limitations

There are several issues that need to be researched more, since they limit the usage and the performance of collaborative filtering techniques despite their advantages. Such an issue is the time complexity. Although, collaborative filtering methods can provide accurate recommendations, the user is likely not to make use of the system in case it is too time-consuming. This is a reality in commercial systems [64][65].

Another crucial problem in collaborative filtering is scalability [65] [71]. Commercial systems in their majority include large datasets, so that a recommendation system has to filter information from millions of items. It is often also the case that a user has rated a minimum percentage of the total number of items, therefore creating very sparse user-matrix. Consequently, the algorithm faces important problems in calculating users' neighborhood, and latens. However, this functionality can be improved by applying dimensionality reduction techniques in very large datasets [73].

Another issue that comes up is related with new items or users insertions. In this case there is no available information about a user who begins to use the system, so we cannot predict interesting items and provide accurate recommendations. In the case of new items insertion, there is no rating information, so it will take some time for them to be recommended to users. This problem is called Cold-start problem [74] or the day one [75].

Frequency of rating can also present problems, since any system using passive filtering to produce ratings, based on quantity data only, will always recommend items that are frequently purchased by users. The converse of this is that any item that is purchased very rarely will never be recommended. This problem, known as the banana problem, was indicated by Burke [76].

Users with unusual tastes will also take poor accurate recommendations, due to the reduced possibility to have any other users in common, in user-based collaborative filtering. Similarly, the same problem occurs in the case of item-based collaborative filtering, when an item is inserted in the system to which no users have rated.

Privacy risk [77] is also an important issue in collaborative filtering techniques, since personal information about a user could be revealed unintentionally. Systems that use information about a user in order to provide information to another user face potential privacy risks.

Many of the above challenges are discussed in [75] [76] [65][74] [71] [69] suggesting various solutions such as graph modelling of collaborative filtering [78] or technique clustering [66] [79].

#### **4.4 The Usage of Technology in Byzantine Music Area**

A lot of work has been done, in the technology areas discussed above. There are several applications, including electronic commerce, blogs, bioinformatics, multimedia data management, digital libraries, machine learning, social networks or wikis. There are also several approaches [105] [106] [6] to apply them in the field of music. However, no work has been done regarding Byzantine Music, in particular.

Therefore, we decided to use these technologies to develop our different system that would support byzantine music's special features. The way we have worked to achieve our goal is extensively described in the following section.

# Chapter 5

## *Byzantine Music Framework*

### 5.1 Introduction

As we have shown in previous sections there is no development with the usage of new technologies to support Byzantine Music. In this chapter we will present the system designed and developed in order to organize, document and store byzantine music files.

We will utilize the technologies discussed above, in order to develop a progressive application that will take advantage from them.

### 5.2 Architecture

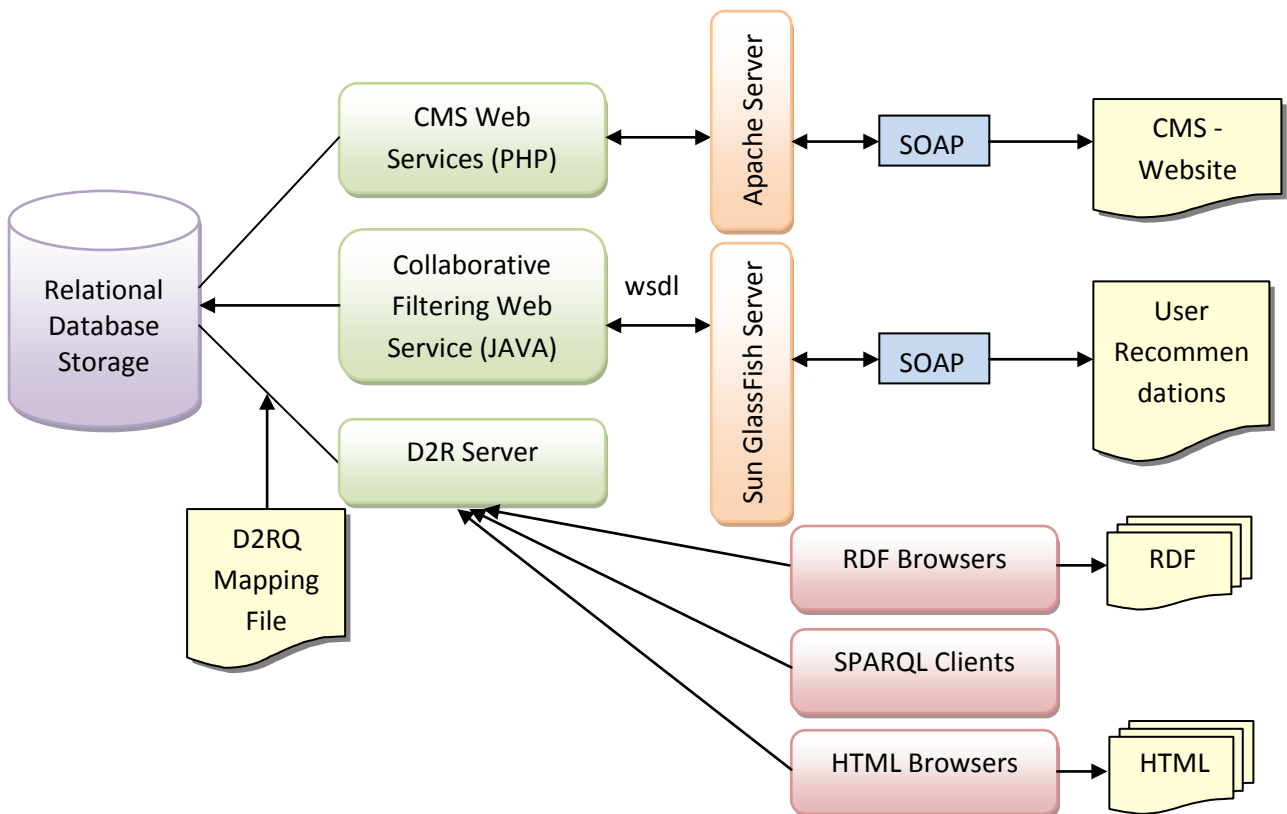


Figure 10 Web Framework Architecture

## 5.3 Implementation

### 5.3.1 Database Storage

The above figure illustrates the basic architecture elements of our implementation. Data are organized in a database schema that keeps detailed information for each music file. The structure of the database is presented to the following figure:

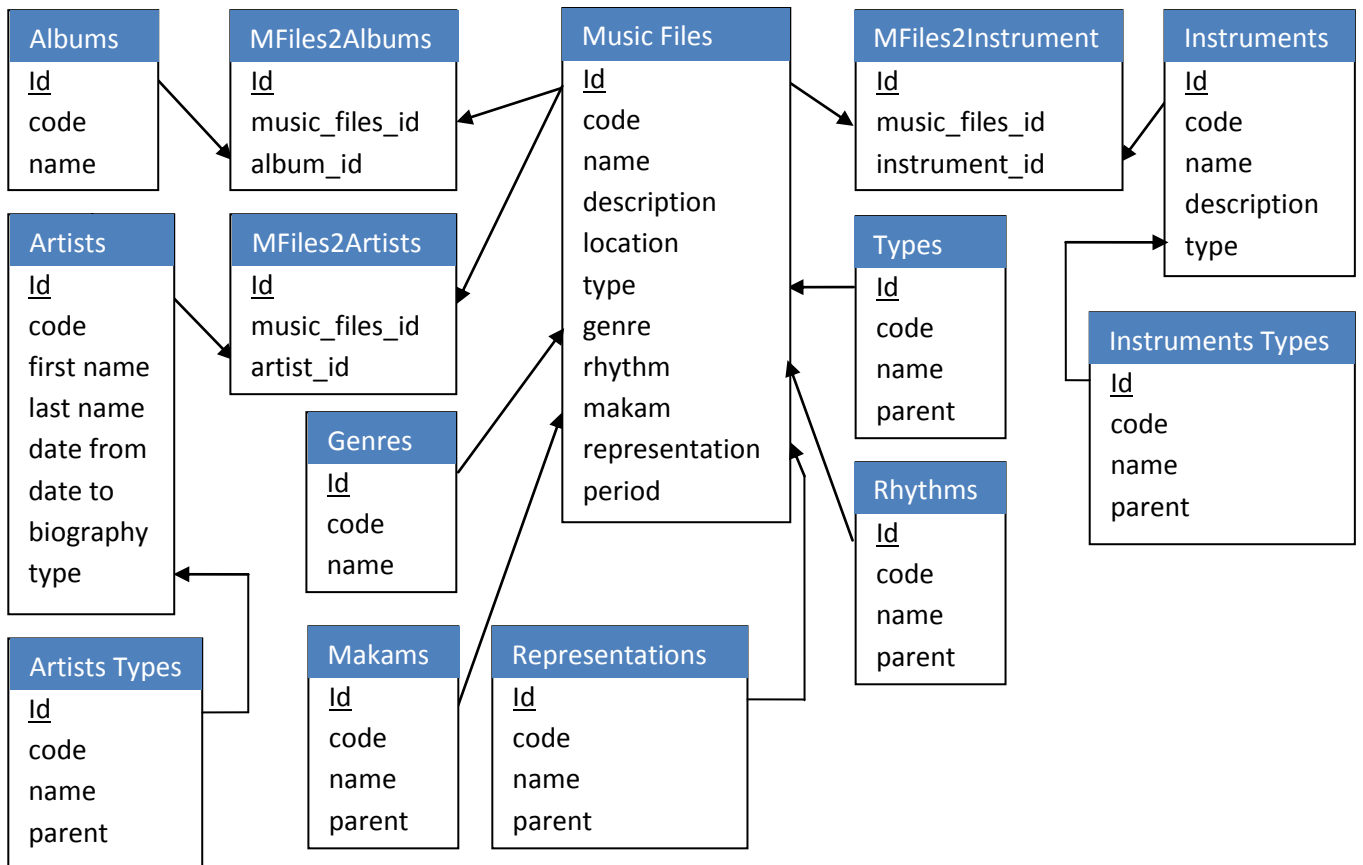


Figure 11 Database Schema

### 5.3.2 Web Services Implementation

These tables are accessed through the appropriate web services in order to manage and show data. Web services are implemented using PHP soap extension. An example of part that the wsdl file describes the service 'FindMusicFilesRequest' is shown below.

1. <?xml version='1.0' encoding='UTF-8' ?>
2. <definitions name='Master'
3.     targetNamespace='urn:EriMaster'
4.     xmlns:tns='urn:EriMaster'
5.     xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
6.     xmlns:xsd='http://www.w3.org/2001/XMLSchema'

```

7.   xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
8.   xmlns:wSDL='http://schemas.xmlsoap.org/wSDL/'
9.   xmlns='http://schemas.xmlsoap.org/wSDL/'>
10.
11.  <message name='FindMusicFilesRequest'>
12.    <part name='Request' type='xsd:ItemRequest'/>
13.  </message>
14.  <message name='FindMusicFilesResponse'>
15.    <part name='Result' type='xsd:ItemResponse'/>
16.  </message>
17.
18.  <portType name='MasterPortType'>
19.    <operation name='FindMusicFiles'>
20.      <input message='tns:FindMusicFilesRequest'/>
21.      <output message='tns:FindMusicFilesResponse'/>
22.    </operation>
23.  </portType>
24.
25.  <binding name='MasterBinding' type='tns:MasterPortType'>
26.    <soap:binding style='rpc'
27.      transport='http://schemas.xmlsoap.org/soap/http'/>
28.      <operation name='FindMusicFiles'>
29.        <soap:operation soapAction='urn:xmethods-delayed-quotes#FindMusicFiles'/>
30.        <input>
31.          <soap:body use='encoded' namespace='urn:xmethods-delayed-quotes'
32.            encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
33.        </input>
34.        <output>
35.          <soap:body use='encoded' namespace='urn:xmethods-delayed-quotes'
36.            encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
37.        </output>
38.      </operation>
39.    </binding>
40.
41.  <service name='MasterService'>
42.    <port name='MasterPort' binding='MasterBinding'>
43.      <soap:address location='http://127.0.0.1/soapmaster2/server.php'/>
44.    </port>
45.  </service>
46.
47. </definitions>

```

A WSDL document is simply a set of definitions [8]. There is a definitions element at the root, and definitions inside.

This file has some basic elements which include the definitions, the types, the messages, the prototype, the binding and finally the service.

All WSDL documents have a definitions element (line 2) which is the root element. The element defines the web service name, declares multiple namespaces used throughout the remainder of the document, and contains all the service elements described.

As client and server send data types to each other this types element (missing) describes the data types used between the client and server. As WSDL uses the W3C XML Schema specification (default choice) it is not tied exclusively to a specific typing system. It is not required for the service to use types element when XML Schema built-in simple types, such as strings and integers, is used.

Message element (line 11) describes a one-way message; irrespectively it is a single request or response. It describes the name of the message and contains zero or more message part elements, which can refer to message parameters or message return values.

The combination of multiple message elements to form a complete one-way or round-trip operation is done by the prototype element. For example, a portType (line 18) can combine one request and one response message into a single request/response operation, most commonly used in SOAP services. It should be noted that a portType can define multiple operations.

The binding element (line 25); this is used to describe the very basics of how the service will be implemented on the wire.

The service element (line 41) is used to describe the address structure that is used to invoke the specified service. This is done by using a URL for notifying the SOAP service.

The server.php file defined above is structured as shown in the following example.

```
1. <?php
2. require 'site_functions.php';
3.
4. ini_set("soap.wsdl_cache_enabled", "0");
5.
6. $server = new SoapServer("soapmaster.wsdl");
7.
8. $server->addFunction("FindMusicFiles");
9. $server->handle();
10. ?>
```



In this file PHP soap extension adds the specified functions to the registry, so that clients can make calls and use them. Function specification is described in the required file 'site\_functions.php'.

```

1. <?php
2.
3. require 'init.inc';
4. header('Content-Type: text/html; charset=utf-8');
5.
6. function FindMusicFiles($args)
7. {
8.     global $con, $makams_array, $repres_array, $mtypes_array;
9.
10.    $code = $args['code'];
11.    $name = $args['name'];
12.    $mtype = $args['mtype'];
13.    $genre = $args['genre'];
14.    $representation = $args['representation'];
15.    $makam = $args['makam'];
16.    $rythm = $args['rythm'];
17.    $period = $args['period'];
18.    $albums = $args['albums'];
19.    $artists = $args['artists'];
20.    $instruments = $args['instruments'];
21.
22.    //create makams tree table
23.    $makams_array = array();
24.    create_makam_table_tree($makam);
25.
26.    $repres_array = array();
27.    create_repre_table_tree($representation);
28.
29.    $mtypes_array = array();
30.    create_mtype_table_tree($mtype);
31.
32.    $query = "select mf.id, mf.name, mf.code, mf.location, mf.type, mf.genre, mf.makam,
mf.representation, mf.rythm, mf.period, mf.description, mf.on_line, mf.priority, mf.creation_user,
mf.creation_date, mf.last_update_user, mf.last_update_date, m.name as makamname, g.name as
genrename, mt.name mtypename, rp.name as reprename, r.name as rythmname from music_files mf
left outer join music_types mt on mt.id = mf.type left outer join makams m on m.id=mf.makam left
outer join genres g on g.id=mf.genre left outer join representations rp on rp.id=mf.representation left
outer join rythms r on r.id=mf.rythm ";
33.    if($instruments)
34.        $query .= ", music_files2musical_instruments m2i ";
35.    if($artists)
36.        $query .= ", music_files2artists m2ar ";
37.    if($albums)

```

```

38.         $query .= ", music_files2albums m2al ";
39.     if($code || $name || $mtype || $genre || $representation || $makam || $rythm || $period ||
    $albums || $artists || $instruments)
40.         $query .= "where ";
41.     if($code)
42.         $query .= "mf.code like '%".$code."%' ";
43.     if($name)
44.     {
45.         if($code)
46.             $query .= "and ";
47.         $query .= "mf.name like '%".$name."%' ";
48.     }
49.     if($mtype)
50.     {
51.         if($code || $name)
52.             $query .= "and (";
53.         else
54.             $query .= "(";
55.
56.         for($i = 0; $i < count($mtypes_array); $i++)
57.         {
58.             $query .= "mf.type='".$mtypes_array[$i]."' ";
59.             if(($i+1) < count($mtypes_array))
60.                 $query .= "or ";
61.         }
62.         $query .= ") ";
63.     }
64.     if($genre)
65.     {
66.         if($code || $name || $mtype)
67.             $query .= "and ";
68.         $query .= "mf.genre='".$genre.' ' ";
69.     }
70.     if($representation)
71.     {
72.         if($code || $name || $mtype || $genre )
73.             $query .= "and (";
74.         else
75.             $query .= "(";
76.
77.         for($i = 0; $i < count($repres_array); $i++)
78.         {
79.             $query .= "mf.representation='".$repres_array[$i]."' ";
80.             if(($i+1) < count($repres_array))
81.                 $query .= "or ";
82.         }

```

```

83.     $query .= ") ";
84. }
85. if($makam)
86. {
87.     if($code || $name || $mtype || $genre || $representation)
88.         $query .= "and ";
89.     else
90.         $query .= "(";
91.
92.     for($i = 0; $i < count($makams_array); $i++)
93.     {
94.         $query .= "mf.makam='".$makams_array[$i]."' ";
95.         if(($i+1) < count($makams_array))
96.             $query .= "or ";
97.     }
98.     $query .= ") ";
99.     //$query .= "mf.makam='$makam' ";
100. }
101. if($rythm)
102. {
103.     if($code || $name || $mtype || $genre || $representation || $makam)
104.         $query .= "and ";
105.     $query .= "mf.rythm='$rythm' ";
106. }
107. if($period)
108. {
109.     if($code || $name || $mtype || $genre || $representation || $makam || $rythm)
110.         $query .= "and ";
111.     $query .= "mf.period='$period' ";
112. }
113. if($instruments)
114. {
115.     if($code || $name || $mtype || $genre || $representation || $makam || $rythm
    || $period)
116.         $query .= "and ";
117.     $query .= "m2i.musical_instrument_id='$instruments' and m2i.music_file_id =
    mf.id ";
118. }
119. if($albums)
120. {
121.     if($code || $name || $mtype || $genre || $representation || $makam || $rythm
    || $period || $instruments)
122.         $query .= "and ";
123.     $query .= "m2al.album_id='$albums' and m2al.music_file_id = mf.id ";
124. }
125. if($artists)

```

```

126.     {
127.         if($code || $name || $mtype || $genre || $representation || $makam || $rythm
    || $period || $instruments || $albums)
128.             $query .= "and ";
129.             $query .= "m2ar.artist_id='$artists' and m2ar.music_file_id = mf.id ";
130.     }
131.
132.     $query .= "order by mf.priority asc";
133.
134.     $result = mysql_query($query);
135.
136.     if (!$result) {
137.         die('Invalid query: ' . mysql_error());
138.     }
139.
140.     $z = 0;
141.     while($rec = mysql_fetch_assoc($result))
142.         $array[$z++] = $rec;
143.
144.     return $array;
145. }
146. ?>

```

We call the web services using SOAP. An example client call is the following:

```

1. <?
2. $serverfile = "http://127.0.0.1/soapmaster2/soapmaster.wsdl";
3.
4. function find_music_files($serverfile, $code, $name, $mtype, $genre, $representation, $makam,
    $rythm, $period, $albums, $artists, $instruments)
5. {
6.     ini_set("soap.wsdl_cache_enabled", "0"); // disabling WSDL cache
7.     $client = new SoapClient($serverfile);
8.     $return = $client->FindMusicFiles(array('code' => $code, 'name' => $name, 'mtype' => $mtype,
    'genre' => $genre, 'representation' => $representation, 'makam' => $makam, 'rythm' => $rythm,
    'period' => $period, 'albums' => $albums, 'artists' => $artists, 'instruments' => $instruments));
9.
10.     return $return;
11. }
12.
13. $results = find_music_files($serverfile, $code, $title, $mtype, $genre, $representation, $makam,
    $rythm, $period, $albums, $artists, $instruments);
14.
15. ?>

```

## 5.4 Semantic Analysis of Byzantine Music - RDF Schema Production

Our goal in this step is to extend music rdf schema defined by kanzaki [80]. As we have discussed in chapter 2, byzantine music is an extension of the general music schema that has specific features needing some special description.

The rdf-owl extension to the initial schema produced to describe music file types is provided below:

```

1. <owl:Class rdf:ID="byz_music_type_18">
2.   <rdfs:label>Βυζαντινή Μουσική</rdfs:label>
3.   <rdfs:comment></rdfs:comment>
4.   <rdfs:subClassOf rdf:resource="#Religious_Music"/>
5. </owl:Class>
6. <owl:Class rdf:ID="byz_music_type_19">
7.   <rdfs:label>Ειρμολογικά Μέλη</rdfs:label>
8.   <rdfs:comment></rdfs:comment>
9.   <rdfs:subClassOf rdf:resource="#byz_music_type_18"/>
10. </owl:Class>
11. <owl:Class rdf:ID="byz_music_type_32">
12.   <rdfs:label>Προσόμοια</rdfs:label>
13.   <rdfs:comment></rdfs:comment>
14.   <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
15. </owl:Class>
16. <owl:Class rdf:ID="byz_music_type_33">
17.   <rdfs:label>Στιχηρά Προσόμοια του Εσπερινού</rdfs:label>
18.   <rdfs:comment></rdfs:comment>
19.   <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
20. </owl:Class>
21. <owl:Class rdf:ID="byz_music_type_34">
22.   <rdfs:label>Στιχηρά Προσόμοια των Αίνων</rdfs:label>
23.   <rdfs:comment></rdfs:comment>
24.   <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
25. </owl:Class>
26. <owl:Class rdf:ID="byz_music_type_35">
27.   <rdfs:label>Στιχηρά Προσόμοια Δοξαστικά των Αίνων</rdfs:label>
28.   <rdfs:comment></rdfs:comment>
29.   <rdfs:subClassOf rdf:resource="#byz_music_type_34"/>
30. </owl:Class>
31. <owl:Class rdf:ID="byz_music_type_36">
32.   <rdfs:label>Και Νυν</rdfs:label>
33.   <rdfs:comment></rdfs:comment>
34.   <rdfs:subClassOf rdf:resource="#byz_music_type_35"/>
35. </owl:Class>
36. <owl:Class rdf:ID="byz_music_type_37">

```

```

37. <rdfs:label>Προσόμοια των αποστίχων του Εσπερινού</rdfs:label>
38. <rdfs:comment></rdfs:comment>
39. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
40. </owl:Class>
41. <owl:Class rdf:ID="byz_music_type_38">
42. <rdfs:label>Προσόμοια των αποστίχων του Όρθρου</rdfs:label>
43. <rdfs:comment></rdfs:comment>
44. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
45. </owl:Class>
46. <owl:Class rdf:ID="byz_music_type_39">
47. <rdfs:label>Προσόμοια Δοξαστικά του Εσπερινού</rdfs:label>
48. <rdfs:comment></rdfs:comment>
49. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
50. </owl:Class>
51. <owl:Class rdf:ID="byz_music_type_40">
52. <rdfs:label>Και Νυν</rdfs:label>
53. <rdfs:comment></rdfs:comment>
54. <rdfs:subClassOf rdf:resource="#byz_music_type_39"/>
55. </owl:Class>
56. <owl:Class rdf:ID="byz_music_type_41">
57. <rdfs:label>Προσόμοια Δοξαστικά Εωθινά</rdfs:label>
58. <rdfs:comment></rdfs:comment>
59. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
60. </owl:Class>
61. <owl:Class rdf:ID="byz_music_type_42">
62. <rdfs:label>Προσόμοια Δοξαστικά των αποστίχων του Εσπερινού</rdfs:label>
63. <rdfs:comment></rdfs:comment>
64. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
65. </owl:Class>
66. <owl:Class rdf:ID="byz_music_type_43">
67. <rdfs:label>Και Νυν</rdfs:label>
68. <rdfs:comment></rdfs:comment>
69. <rdfs:subClassOf rdf:resource="#byz_music_type_42"/>
70. </owl:Class>
71. <owl:Class rdf:ID="byz_music_type_44">
72. <rdfs:label>Προσόμοια Δοξαστικά των αποστίχων του Όρθρου</rdfs:label>
73. <rdfs:comment></rdfs:comment>
74. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
75. </owl:Class>
76. <owl:Class rdf:ID="byz_music_type_45">
77. <rdfs:label>Και Νυν</rdfs:label>
78. <rdfs:comment></rdfs:comment>
79. <rdfs:subClassOf rdf:resource="#byz_music_type_44"/>
80. </owl:Class>
81. <owl:Class rdf:ID="byz_music_type_46">
82. <rdfs:label>Προσόμοια Ν Ψαλμού</rdfs:label>

```

46

83. <rdfs:comment></rdfs:comment>

84. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_32"/>

85. </owl:Class>

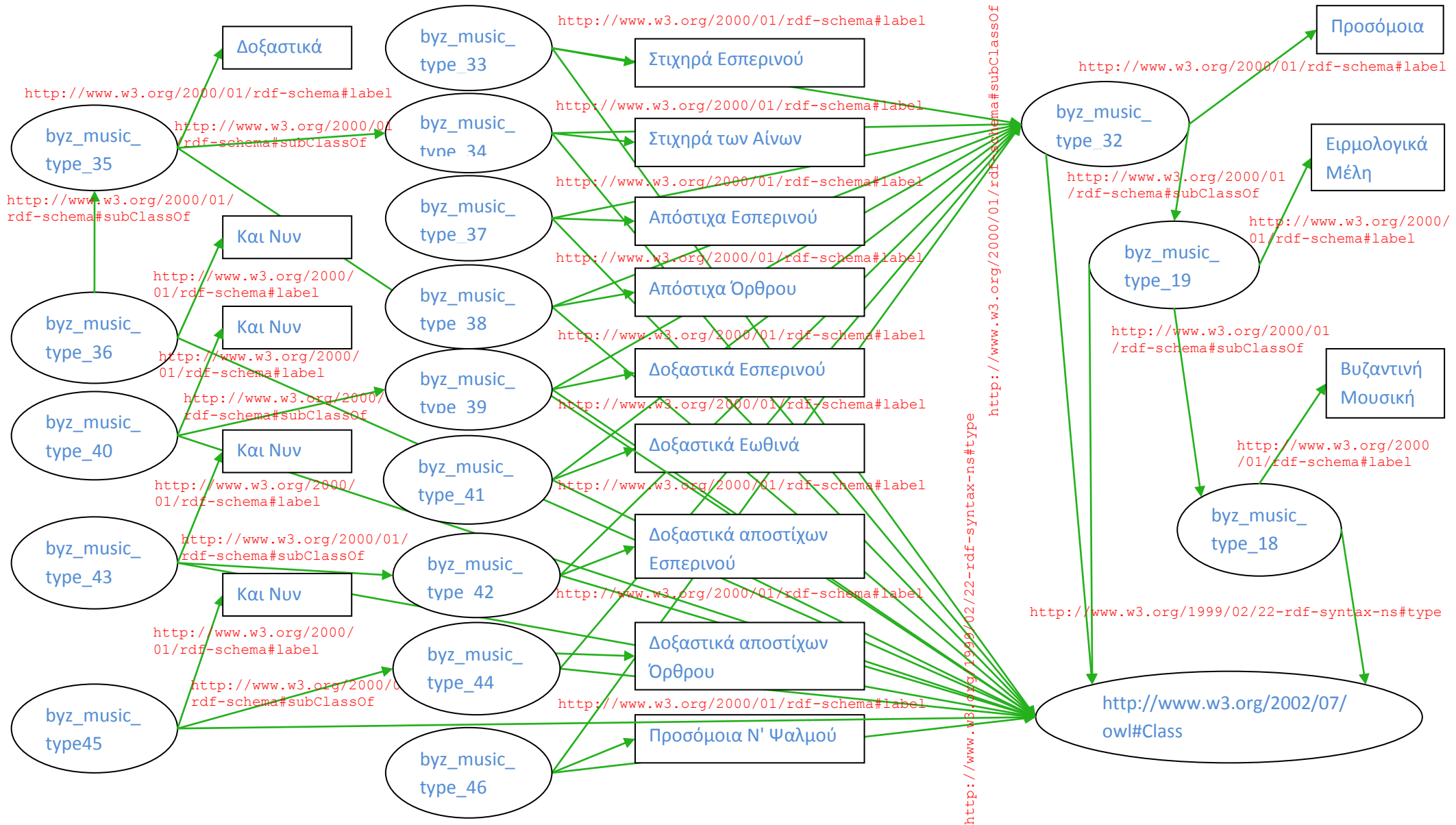


Figure 12 Music Types rdf-schema extension graph



We have also added an extension related to the makams of Byzantine Music, shown in the following lines. We provide the first makam example code:

```

1. <owl:Class rdf:ID="makam_1">
2.   <rdfs:label>Βυζαντινοί Ήχοι</rdfs:label>
3.   <rdfs:comment></rdfs:comment>
4. </owl:Class>
5. <owl:Class rdf:ID="makam_2">
6.   <rdfs:label>Διατονικό Γένος</rdfs:label>
7.   <rdfs:comment></rdfs:comment>
8.   <rdfs:subClassOf rdf:resource="#makam_1"/>
9. </owl:Class>
10. <owl:Class rdf:ID="makam_5">
11.   <rdfs:label>Πρώτος Ήχος</rdfs:label>
12.   <rdfs:comment></rdfs:comment>
13.   <rdfs:subClassOf rdf:resource="#makam_2"/>
14. </owl:Class>
15. <owl:Class rdf:ID="makam_13">
16.   <rdfs:label>Ειρμολογικός Πρώτος</rdfs:label>
17.   <rdfs:comment></rdfs:comment>
18.   <rdfs:subClassOf rdf:resource="#makam_5"/>
19. </owl:Class>
20. <owl:Class rdf:ID="makam_14">
21.   <rdfs:label>Στιχηραρικός Πρώτος</rdfs:label>
22.   <rdfs:comment></rdfs:comment>
23.   <rdfs:subClassOf rdf:resource="#makam_5"/>
24. </owl:Class>
25. <owl:Class rdf:ID="makam_15">
26.   <rdfs:label>Παπαδικός Πρώτος</rdfs:label>
27.   <rdfs:comment></rdfs:comment>
28.   <rdfs:subClassOf rdf:resource="#makam_5"/>
29. </owl:Class>
30. <owl:Class rdf:ID="makam_16">
31.   <rdfs:label>Παθητικός Πρώτος (νάος)</rdfs:label>
32.   <rdfs:comment></rdfs:comment>
33.   <rdfs:subClassOf rdf:resource="#makam_5"/>
34. </owl:Class>
35. <owl:Class rdf:ID="makam_17">
36.   <rdfs:label>Πρώτος εκ του Κε</rdfs:label>
37.   <rdfs:comment></rdfs:comment>
38.   <rdfs:subClassOf rdf:resource="#makam_5"/>
39. </owl:Class>
40. <owl:Class rdf:ID="makam_18">
41.   <rdfs:label>Χρωματικός Πρώτος</rdfs:label>
42.   <rdfs:comment></rdfs:comment>
43.   <rdfs:subClassOf rdf:resource="#makam_5"/>
44. </owl:Class>

```

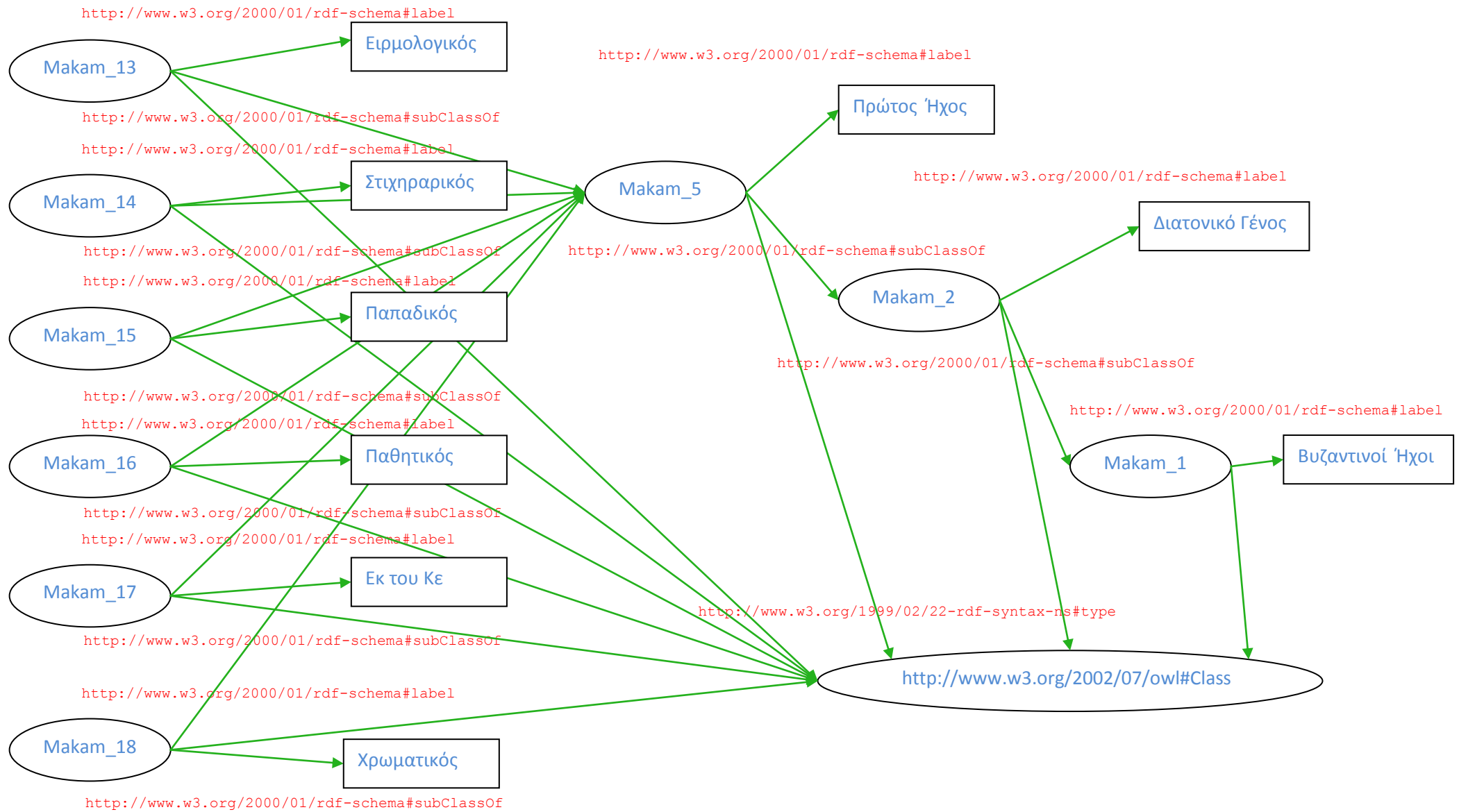


Figure 13 Makams rdf-schema extension graph

We have also extended the musical instruments class designed by kanzaki, so that it includes Greek traditional musical instruments. A subpart of the whole document is shown in the following lines:

```

1. <owl:Class rdf:ID="musical_instrument_cat_1">
2.   <rdfs:label>Παραδοσιακά Μουσικά Όργανα</rdfs:label>
3.   <rdfs:comment></rdfs:comment>
4.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_0"/>
5. </owl:Class>
6. <owl:Class rdf:ID="musical_instrument_cat_4">
7.   <rdfs:label>Αερόφωνα</rdfs:label>
8.   <rdfs:comment></rdfs:comment>
9.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_1"/>
10. </owl:Class>
11. <owl:Class rdf:ID="musical_instrument_19">
12.   <rdfs:label>Φλογέρα</rdfs:label>
13.   <rdfs:comment></rdfs:comment>
14.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
15. </owl:Class>
16. <owl:Class rdf:ID="musical_instrument_20">
17.   <rdfs:label>Ζουρνάς</rdfs:label>
18.   <rdfs:comment></rdfs:comment>
19.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
20. </owl:Class>
21. <owl:Class rdf:ID="musical_instrument_21">
22.   <rdfs:label>Τσαμπούνα</rdfs:label>
23.   <rdfs:comment></rdfs:comment>
24.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
25. </owl:Class>
26. <owl:Class rdf:ID="musical_instrument_22">
27.   <rdfs:label>Γκάιντα</rdfs:label>
28.   <rdfs:comment></rdfs:comment>
29.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
30. </owl:Class>
31. <owl:Class rdf:ID="musical_instrument_23">
32.   <rdfs:label>Κλαρίνο</rdfs:label>
33.   <rdfs:comment></rdfs:comment>
34.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
35. </owl:Class>
36. <owl:Class rdf:ID="musical_instrument_24">
37.   <rdfs:label>Φύλλο του δέντρου</rdfs:label>
38.   <rdfs:comment></rdfs:comment>
39.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
40. </owl:Class>

```

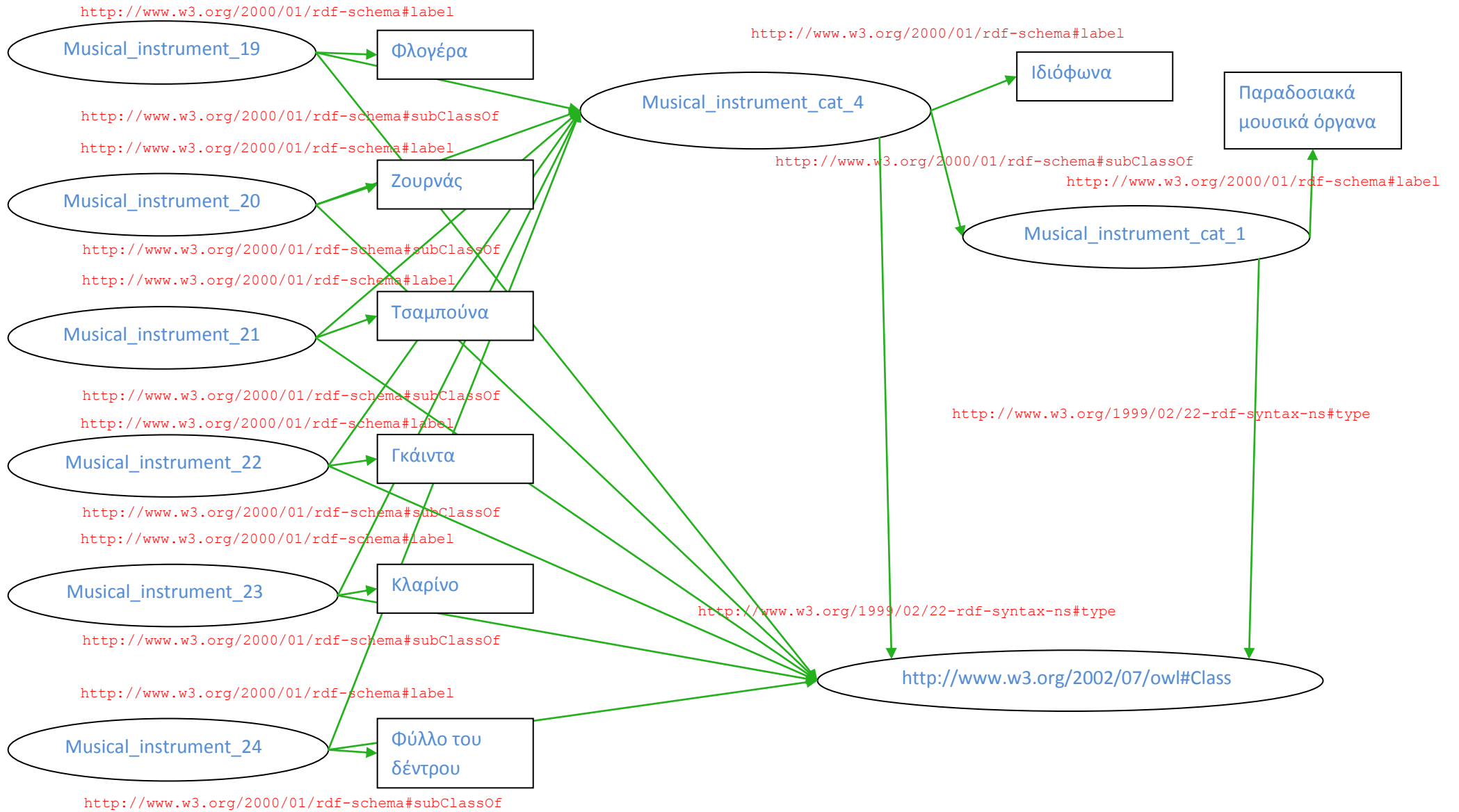


Figure 14 Musical instruments: rdf-schema extension graph

We have also made the appropriate additions in the artists schema, adding the Ιεροψάλτης owl:Class and the corresponding players to the above musical instruments. See Appendix A for more details.

## 5.5 Recommending Music Files Based on Collaborative Filtering Method

Nowadays, most of the popular commercial systems use collaborative filtering (CF) techniques to efficiently provide recommendations to users based on opinions of other users [65] [82] [83]. The algorithm used in music files recommendations of this system is based on [81]. Our algorithm is a simplified version of the framework described, which is based on user trends in order to predict user values on unrated items.

The pseudocode of the algorithm is provided below:

### **Recommendations Algorithm**

#### **Preparation:**

- Connect to the database
- Select music files ordered by id
- Select users ordered by username
- Create the array of users' ratings to items

#### **Use of PCA to reduce users - items subspace**

- Find the Covariance Matrix
- Find the EigenVectors, EigenValues Matrices
- Find the most significant Eigenvectors and Eigenvalues Matrices
- Map users into the new subspace

#### **Diagnose personality similarity**

- Create personality similarity propabilities array using Pennock's algorithm [84] for the active user  $u_a$

#### **Predict values on unrated items**

- Find propability distribution for active user's  $u_a$  for unobserved items
- Find the rating value that maximizes the above distribution for each unobserved item

#### **Recommend highly predicted items**

- Sort items by asceding predicted value
- Return 10 most highly rated items

### 5.5.1 Principal Component Analysis (PCA)

PCA is a method to reduce a high dimensional vector to a small number of orthogonal basis vectors, i.e. the principal components. The process includes the following steps:

- Firstly, we compute the covariance matrix (C) of the training set,
- Secondly we Eigen-decomposing the covariance matrix to produce two matrices: a diagonal eigenvalues matrix ( $\Lambda_{mm}$ ) and an eigenvectors matrix ( $U_{mm}$ ). These matrices give the initial covariance matrix by performing the following Equation:

$$C = U_{mm} \times \Lambda_{mm} \times U_{mm}^T \quad (5.1)$$

- We preserve the p principal eigenvectors and their corresponding eigenvalues, since covariance matrix C can also be approximated by the p largest components, according to the Equation:

$$= U_{mp} \times \Lambda_{mp} \times U_{mp}^T \approx C \quad (5.2)$$

### 5.5.2 User Mapping

The next step is to map user vectors to the reduced p-dimensional item subspace. To this aim, given the current ratings of each user we use Equation

$$= u \times U_{mp} \quad (5.3)$$

This  $m \times p$  "thin" matrix is the p-dimensional representation of the m users for the n items. So far, complexity computation failed to give real-time performance and behavior of the memory-based algorithms. From now on, the quantity of the items is no more an important factor.

### 5.5.3 Personality Diagnosis

Personality Diagnosis [84] states that user's  $u_i$  personality type can be described as a vector

$$(5.4)$$

Where  $u_i$  denotes user's  $u_i$  true ratings over  $i_j$  items. These ratings encode user's underlying internal preferences. As Pennock observes there is a critical distinction between true ratings and reported ratings. True ratings cannot be accessed directly by the system. Indeed the reported ratings, which users provide to the system, constitute the only accessible information. So, Pennock assumes that these ratings include Gaussian noise based on the fact that the same user may

report different ratings depending on different occasions, such as the mood, the context of other ratings provided in the same session or on any other reason - external factor. All these factors are summarized as Gaussian noise. That is, user's  $u_i$  reported rating for the item  $i_j$  is drawn from an independent normal distribution with mean  $y + \epsilon$ . Specifically,

$$r_{i,j} = y + \epsilon \tag{5.5}$$

where  $x$  is the reported rating value,  $y$  is the true rating value (noise-free), and  $\sigma$  is a free parameter.

It is further assumed that the distribution of rating vectors (personality types), which are contained in the rating matrix of the database, is representative of the distribution of personalities in the target population of users. Based on this, the prior probability  $\Pr(\kappa = \kappa)$  that the active user  $u_a$  rates items according to a vector  $\kappa$ , is given by the frequency that other users rate according to  $\kappa$ . So, instead of counting occurrences explicitly, a random variable  $\kappa$  is defined which takes one out of  $m$  possible values,  $\kappa_1, \dots, \kappa_m$ , each one with equal probability  $1/m$ .

We can now compute the probability that the active user  $u_a$  is of the same personality type as any other user  $u_i$ , by applying Bayes' rule.

$$\Pr(\kappa = \kappa_i \mid r_{a,1} = x_1, \dots, r_{a,n} = x_n) = \frac{\Pr(\kappa = \kappa_i) \cdot \prod_{j=1}^n \Pr(r_{a,j} = x_j \mid \kappa = \kappa_i)}{\sum_{i=1}^m \Pr(\kappa = \kappa_i) \cdot \prod_{j=1}^n \Pr(r_{a,j} = x_j \mid \kappa = \kappa_i)} \tag{5.6}$$

where  $m$  is the number of users and  $n$  the number of items.

Once we compute this quantity for each user  $u_i$ , we can compute a probability distribution for the active user's rating of any unseen object, using the following Equation:

$$\tag{5.7}$$

Finally the prediction  $p_{a,j}$  for a user  $u_a$  on target item  $i_j$  is computed as follows:

$$\tag{5.8}$$

The recommended items to the user are selected randomly from those with the highly predicted value.

#### 5.5.4 Normalization

The Nr factor shown in relation 5.6 is a parameter used to normalize the user similarity probability. The probabilities computed are multiplied to this quantity, so that to take the value in a normalized form.

#### 5.5.5 Web Service Support

The algorithm described above is implemented in Java, since the rest code of the system is written in php. Thus, we have developed one more web service in order to make the collaborative filtering results available by the appropriate php application.

We have used Sun GlassFish Enterprise Server v2.1.1 [85]. The Java code that implements the collaborative filtering algorithm is stored as an application in this server. The wsdl file that publishes the service is the following:

```

1. <definitions targetNamespace="http://endpoint.master/" name="MasterService">
2.     <types>
3.         <xsd:schema>
4.             <xsd:import namespace="http://endpoint.master/" schema
5.                 Location="http://localhost:8080/master/MasterService?xsd=1"/>
6.         </xsd:schema>
7.     </types>
8.     <message name="master">
9.         <part name="parameters" element="tns:master"/>
10.    </message>
11.    <message name="masterResponse">
12.        <part name="parameters" element="tns:masterResponse"/>
13.    </message>
14.    <portType name="Master">
15.        <operation name="master">
16.            <input message="tns:master"/>
17.            <output message="tns:masterResponse"/>
18.        </operation>
19.    </portType>
20.    <binding name="MasterPortBinding" type="tns:Master">
21.        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
22.        <operation name="master">
23.            <soap:operation soapAction=""/>
24.            <input>
25.                <soap:body use="literal"/>
26.            </input>

```



```

27.         <output>
28.             <soap:body use="literal"/>
29.         </output>
30.     </operation>
31. </binding>
32. <service name="MasterService">
33.     <port name="MasterPort" binding="tns:MasterPortBinding">
34.         <soap:address location="http://localhost:8080/master/MasterService"/>
35.     </port>
36. </service>
37. </definitions>

```

The xsd file that defines the input and output of the service is the one shown below:

```

1. <xs:schema version="1.0" targetNamespace="http://endpoint.master/">
2.     <xs:element name="master" type="tns:master"/>
3.     <xs:element name="masterResponse" type="tns:masterResponse"/>
4.     <xs:complexType name="master">
5.         <xs:sequence>
6.             <xs:element name="active_user" type="xs:string" minOccurs="0"/>
7.         </xs:sequence>
8.     </xs:complexType>
9.     <xs:complexType name="masterResponse">
10.        <xs:sequence>
11.            <xs:element name="return" type="xs:string" nillable="true" minOccurs="0"
12.                maxOccurs="unbounded"/>
13.        </xs:sequence>
14.    </xs:complexType>
15. </xs:schema>

```

The java code that implements the process can be seen in Appendix A.

The php call of the web service is made in the way we have described above for the rest web services implemented in php, using php\_soap extension, with the difference that service file points to `http://127.0.0.1:8080/master/MasterService?wsdl`.

# Chapter 6

---

## *Experimental Results*

### **6.1 Introduction**

In this section we will show the results of the system we have developed. How it finally works, what test users see, the collaborative filtering efficiency and efficacy, and whether it meets users' needs. How they judge the system and how useful they find it. The above questions are crucial, since they provide evidence of what the system offers.

### **6.2 The final application**

The application developed is separated in two sub-applications. One, which is used for the management and organization of the data, and a second one that is addressed to the general public, so that they can search, view, hear and rate items.

We start by presenting the content management system and the basic functions that are supported.

#### **6.2.1 Content Management System**

The Content Management System (CMS) is a collection of procedures used to manage work flow in a collaborative environment. These procedures can be manual or computer-based. The procedures are designed to:

- Allow for a large number of people to contribute to and share stored data
- Control access to data view or edit
- Aid in easy storage and retrieval of data
- Reduce repetitive duplicate input
- Additions or deletions of items
- Organization of the data and meta-data
- Documentation of the music files

The access to the cms is not free as for the public one. It is protected by username and password that users (administrators) should know, so that they can gain access to the data.

Below we can view the starting page of the cms. There is a form where the user should complete his username and password, in order to authenticate his login.

Figure 15 CMS logon screen

After the logon to the system, users can see the menu bar on the top of the web application where they can select which data category to manage.

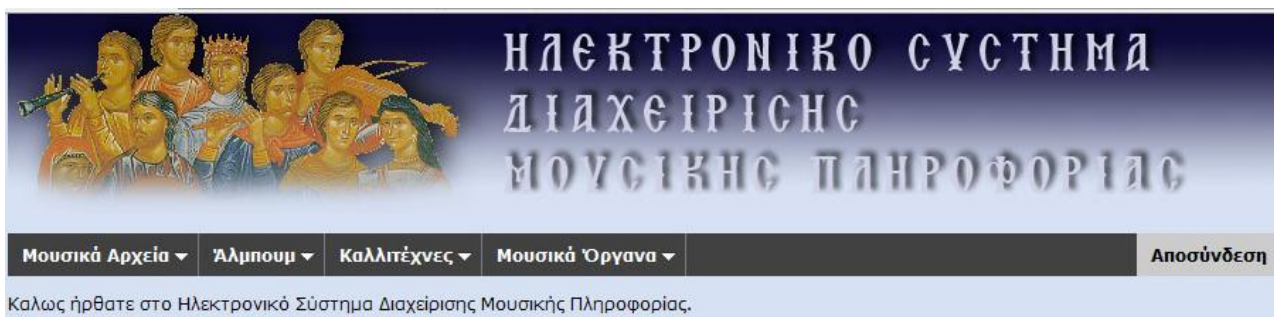


Figure 16 Menu bar

As we can see in the picture above, there are several categories that a user is able to work out.

These include the following elements:

- Album
  - Artists
    - Artists' categories
- Musical Instruments
  - Musical Instruments' types
- Music Files
  - Genre
  - Makam
  - Rhythm
  - Representation
  - Music Type

The supported actions include addition, search, deletion and editing of the above instances.

By selecting 'search a category' we can view a list of the items inserted in the system.

The screenshot displays a web interface for searching music files. At the top, there are navigation tabs: 'Μουσικά Αρχεία', 'Άλμπουμ', 'Καλλιτέχνες', 'Μουσικά Όργανα', and 'Αποσύνδεση'. Below the tabs is a search bar with the text 'Αναζήτηση Μουσικού Αρχείου' and a search button labeled 'Αναζήτηση'. The main content area shows a table of search results. The table has four columns: 'Μουσικού Αρχείου', 'Κατηγορία', 'Δημοσιευμένο', and 'Κατάταξη'. The first column contains 10 numbered items, each with a checkbox. The second column lists categories for each item. The third column shows green checkmarks, indicating that all items are published. The fourth column shows dropdown menus for sorting, with numbers 1 through 10. Below the table, there is a pagination control showing '1-10 από 208 αποτελέσματα' and a 'Διαγραφή Επιλεγμένων' button.

Μουσικού Αρχείου	Κατηγορία	Δημοσιευμένο	Κατάταξη
<input type="checkbox"/> 1. Αινείτε τον Κύριον	Κοινωνικά	✓	1
<input type="checkbox"/> 2. Χερουβικό	Χερουβικά	✓	2
<input type="checkbox"/> 3. Αινείτε τον Κύριον	Κοινωνικά	✓	3
<input type="checkbox"/> 4. Χαίρε Νύμφη Φαεινή	Καλοφωνικοί Ειρμοί	✓	4
<input type="checkbox"/> 5. Δούλοι Κύριον	Πολυέλεσι	✓	5
<input type="checkbox"/> 6. Ποτήριο σωτηρίου λήψομαι	Κοινωνικά	✓	6
<input type="checkbox"/> 7. Εις το όρος	Ιδιόμελα Δοξαστικά Εωθινά	✓	7
<input type="checkbox"/> 8. Δοξολογία	Αργές Δοξολογίες	✓	8
<input type="checkbox"/> 9. Δεύτε χριστοφόροι λαοί	Ιδιόμελα Δοξαστικά των Ωρών	✓	9
<input type="checkbox"/> 10. Ανάγνωσμα - Αλληλούια - Ιδού ο Νυμφίος	Απολυτικά	✓	10

Figure 17 Music files search results

The results are presented separated into pages. Each row is divided into five columns. The first one includes the serial number of the resulted record with a checkbox on its left side, where the

user checks in order to delete an item. The second column refers to the music file category in which the record belongs. By clicking on a category name, the system filters the results, so that it now includes only the files corresponding to the category. The next column indicates whether a file is on-line (available to the public) or not. By clicking on the symbol or the state shifts to the opposite one. Finally, the last column shows/alters the sorting order of the file in the results list.

To choose an element for editing we can click on its file name. The next page is shown below:

Μουσικό Αρχείο | Άλμπουμ | Καλλιτέχνες | Μουσικά Όργανα

Η συμπλήρωση των πεδίων με αστερίσκο (\*) είναι υποχρεωτική.

**Στοιχεία Μουσικού Αρχείου**

\* Κωδικός: Ανείτε τον Κύριον  
 \* Όνομασία: Ανείτε τον Κύριον  
 Περιγραφή:  
 Αλλαγή Αρχείου: Αναζήτηση...  
 Αρχείο: AF\_Track01.mp3

**Χαρακτηριστικά Μουσικού Αρχείου**

Είδος Μουσικής: Κοινωνικά  
 Genre: Επιλέξτε Genre  
 Αναπαράσταση: Ήχογράφηση σε ψηφιακό δίσκο  
 Ήχος: Άγια  
 Ρυθμός: Τονικός  
 Χρονική Περίοδος:

**Στοιχεία Δημοσίευσης**

Δημοσιευμένο: Ναι  Όχι   
 Κατάταξη: 1  
 Ημ. Δημιουργίας: 2009-09-29  
 Χρήστης: ranutsu  
 Ημ. Τελευταίας Τροποποίησης: 2009-09-29  
 Χρήστης: ranutsu

Επεξεργασία Καθαρισμός

Figure 18 Music file edit form

This page contains four sheets, the main sheet where the user can edit music file's information, the album-music files sheet where the file is associated to one or more albums, the artists sections where we can define the artists that take part, and the last one, where the musical instruments used can be indicated.

As we can see in the above picture, there are several fields that we can edit to change the underlying information of the file. Synoptically, they include the code of the file (it is used as the item id in rdf schema production), the title (label), the mp3 file, a short description (comment) of the file, the chronological period and some extra details that characterize the file such as genre, music type, makam, rhythm and representations (drop down lists). When we have finished with the changes needed, we press on the Edit (Επεξεργασία) button to save them.

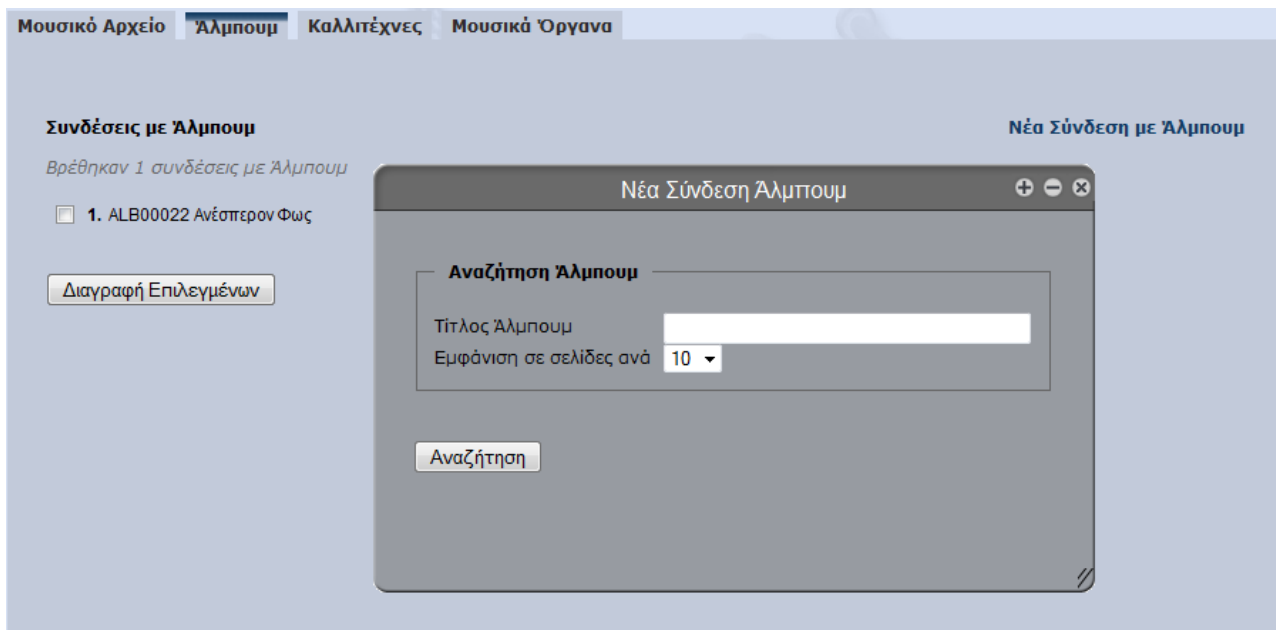


Figure 19 Music file to albums correlation

Afterwards, we can work up with the albums that may contain the file. By clicking the link on the upper left side we can search through the albums inserted in the system and select the appropriate titles. Note that a file may belong in more than one album. The same process is used to manage the artists and musical instruments of a music file.

## 6.2.2 The Web Site

The information stored in the system using the cms is then provided to the public from a different location, so that users can search, view or listen to the files. This is an application designed to support advanced searches based on several criteria, which the user can combine in different ways in order to locate the files he/she is more interested in. Users can also signup to the system, in order to be able to rate the items and get recommendations.

We will show these actions in the following pages with some examples explaining the usage framework. Firstly, we present the starting page of the system.

**ΗΛΕΚΤΡΟΝΙΚΗ ΜΟΥΣΙΚΗ ΣΥΛΛΟΓΗ**  
Καλωσήρθατε **rmanos**, Έξοδος

**Αναζήτηση Μουσικού Αρχείου**

Κωδικός:   
Τίτλος:

**Χαρακτηριστικά Μουσικού Αρχείου**

Είδος Μουσικής:   
Genre:   
Αναπαράσταση:   
Ήχος:   
Ρυθμός:   
Χρονική Περίοδος:

**Επιπρόσθετα Στοιχεία**

Άλμπουμ:   
Καλλιτέχνες:   
Μουσικά Όργανα:

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Figure 20 The search engine first page

As it is illustrated in the above figure, the starting page contains an advanced search form. The user can select none or more criteria to filter the results and get the files he/she is interested in. The criteria include the following fields:

- Code
- Music file name
- Music Type
- Genre
- Representation
- Makam
- Rhythm

- Chronological Period
- Album
- Artist by category
- Musical instrument by category

These elements are dynamically exported from the database, and structured search is supported. This means that if a makam category is selected and includes subcategories in the information tree, the system is going to return files that belong to the category, along with these in the subcategories grouped. E.g. if the user selects "Protos Echos" (Πρώτος Ήχος), the results will also contain files from "Stixirarikos Protos Echos" (Στιχηραρικός Πρώτος Ήχος), "Eirmologikos Protos Echos" (Ειρμολογικός Πρώτος Ήχος) etc.

An example of the categories tree is shown below:

**Χαρακτηριστικά Μουσικού Αρχείου**

Είδος Μουσικής: Επιλέξτε Είδος Μουσικής

Genre: Επιλέξτε Genre

Αναπαράσταση: Επιλέξτε Αναπαράσταση

Ήχος: Επιλέξτε Ήχο

Ρυθμός: Βυζαντινοί Ήχοι

Χρονική Περίοδος: Διατονικό Γένος

**Επιπρόσθετα Στοιχεία**

Άλμπουμ:

Καλλιτέχνες:

Μουσικά Όργανα:

Αναζήτηση

Πρώτος Ήχος

- Ειρμολογικός Πρώτος
- Στιχηραρικός Πρώτος
- Παπαδικός Πρώτος
- Παθητικός Πρώτος (νάος)
- Πρώτος εκ του Κε
- Χρωματικός Πρώτος
- Τέταρτος Ήχος
- Λέγετος
- Άγια
- Στιχηραρικός Τέταρτος
- Κλίτιον Μέλος
- Χρωματικός Τέταρτος
- Ήχος Πλάγιος του Πρώτου
- Στιχηραρικός Πλάγιος του Πρώτου (εκ του Πα)
- Ειρμολογικός Πλάγιος του Πρώτου (εκ του Κε)
- Πλάγιος του Πρώτου με χρόα σπάθη
- Εναρμόνιος Πλάγιος του Πρώτου

Figure 21 Makams' selection tree

By clicking the search button at the bottom of the form, we get the results list, separated into several pages.



**Νέα Αναζήτηση**  
◀ ▶ ||

*Βρέθηκαν 208 αποτελέσματα σύμφωνα με τα κριτήρια αναζήτησης.*

**1. Αινείτε τον Κύριον** Ακούστε

Κατηγορία: Κοινωνικά Ήχος: Άγια  
Άλμπουμ: Ανέσπερον Φως  
Καλλιτέχνες: Βασίλειος Νόνης, Δανιήλ Πρωτοψάλτης  
Μουσικά Όργανα:

**2. Χερουβικό** Ακούστε

Κατηγορία: Χερουβικά Ήχος: Άγια  
Άλμπουμ: Ανέσπερον Φως  
Καλλιτέχνες: Βασίλειος Νόνης, Πέτρος Βυζάντιος  
Μουσικά Όργανα:

**3. Αινείτε τον Κύριον** Ακούστε

Κατηγορία: Κοινωνικά Ήχος: Διατονικός Βαρύς  
Άλμπουμ: Ανέσπερον Φως  
Καλλιτέχνες: Βασίλειος Νόνης, Πέτρος Λαμπαδάριος  
Μουσικά Όργανα:

Figure 22 Results page

As it is shown, the results are presented with some information about the category, the makam, the album, the artists and the musical instruments. These elements are active links that filter the results by their label. The user can also listen to the file by clicking on the right link named "Ακούστε" (Akouste) or rate the file, in case he is already logged in the system. The title of each record is also a link that leads to the specific file location, where the user can view information about its details, listen to the music file, rate it, and finally get his personalized recommendations.

**9. Δεύτε χριστοφόροι λαοί** Ακούστε

Κατηγορία: **Ιδιόμελα Δοξαστικά των Ωρών** Ήχος: Σπχηρατικός Πλάγιος του Πρώτου (εκ του Πα)  
Άλμπουμ: Ανέσπερον Φως  
Καλλιτέχνες: Βασίλειος Νόνης, Μανουήλ Πρωτοψάλτης  
Μουσικά Όργανα:

★ ★ ★ ★ ★ ★ ★ ★ ★ ★

9 Rating: 7.1/10 (9 votes)

Figure 23 Rating a file

The above picture shows an example of rating a file. As we can see, there are 10 stars, some of which are transparent, some have brown color, and the rest are yellow. Brown stars indicate the total value of the file, while yellow stars show the active user's time rating. Under the stars we show the average rating value (indicated also by the brown stars), as well as the number of voters.

An example of a presentation page is shown in figure 24. The details of the file are displayed at the top of the page. Each title opens a description window of the category, the makam, the album or the artist's biography. Under the file information the star rating bar appears. The link that plays the file is located at the left side. Finally, the personalized recommendations are presented as "see also" files. Each recommendation is a link that goes to the file's page.

**Δείτε χριστοφόροι λαοί** Ακούστε

Κατηγορία: **Ιδιόμελα Δοξαστικά των Ωρών**  
 Ήχος: **Σπηραρικός Πλάγιος του Πρώτου (εκ του Πα)**  
 Άλμπουμ: **Ανέσπερον Φως**  
 Καλλιτέχνες: **Βασίλειος Νόνης, Μανουήλ Πρωτοψάλτης**

★ ★ ★ ★ ★ ★ ★ ☆ ☆ ☆

9 Rating: **7.1/10** (9 votes cast)

**Δείτε Επίσης**

1. Το προσταχθέν Predicted value **10.0/10.0** [Λεπτομέρειες](#)
2. Τα πάθη τα σεπτά Predicted value **10.0/10.0** [Λεπτομέρειες](#)
3. Προ της γεννήσεως... Predicted value **9.0/10.0** [Λεπτομέρειες](#)
4. Κεκραγάριον αργοσύντομον Predicted value **9.0/10.0** [Λεπτομέρειες](#)
5. Πόρνη προσήλθε σοι Predicted value **9.0/10.0** [Λεπτομέρειες](#)
6. Προκαθάρωμεν εαυτούς... Predicted value **9.0/10.0** [Λεπτομέρειες](#)
7. Χαίρε Νύμφη - Αλληλούια Predicted value **9.0/10.0** [Λεπτομέρειες](#)
8. Αλληλουάριον Predicted value **9.0/10.0** [Λεπτομέρειες](#)
9. Ε Ωδή κανόνος Αγ. Γεωργίου, "Ορθρίσωμεν... Χριστός Ανέστη... Αναστάς..." Predicted value **9.0/10.0** [Λεπτομέρειες](#)
10. "Δόξα... Ανέτειλε το έαρ..." Predicted value **9.0/10.0** [Λεπτομέρειες](#)
11. Μαρτυρικόν - Θεοτοκίον Predicted value **9.0/10.0** [Λεπτομέρειες](#)
12. "Ατενίσαι το όμμα..." Predicted value **9.0/10.0** [Λεπτομέρειες](#)
13. "Χαίρε Κεχαριτωμένη..." Predicted value **9.0/10.0** [Λεπτομέρειες](#)
14. Αύτη ημέρα Κυρίου Predicted value **8.0/10.0** [Λεπτομέρειες](#)
15. Θεαρχίω νεύματι Predicted value **8.0/10.0** [Λεπτομέρειες](#)

Figure 24 Presentation page with the list of recommendations

Shown next to the title is the predicted value, estimated by the collaborative filtering algorithm. The red link to the right of each recommendation opens an explanation window, which shows the algorithmic process that computes the predicted value.

**Διαδικασία υπολογισμού της προβλεπόμενης τιμής για το Τη Υπερμάχω**

**Εξεταζόμενη πηγή 1**

Ομοιότητα με χρήστη giota = 0,00036136  
 Τιμή χρήστη giota = 10.0  
 $\text{distribution}[34] += e^{-\frac{(1-10.0)^2}{2 \cdot \sigma^2}} \cdot 0,00036136 = 0,00032656$

Ομοιότητα με χρήστη koutenp = 0,00000004  
 Τιμή χρήστη koutenp = 7.0  
 $\text{distribution}[34] += e^{-\frac{(1-7.0)^2}{2 \cdot \sigma^2}} \cdot 0,00000004 = 0,00032660$

Ομοιότητα με χρήστη mike1821 = 0,00156185  
 Τιμή χρήστη mike1821 = 10.0  
 $\text{distribution}[34] += e^{-\frac{(1-10.0)^2}{2 \cdot \sigma^2}} \cdot 0,00156185 = 0,00173805$

Ομοιότητα με χρήστη dimboug = 0,00002304  
 Τιμή χρήστη dimboug = 2.0  
 $\text{distribution}[34] += e^{-\frac{(1-2.0)^2}{2 \cdot \sigma^2}} \cdot 0,00002304 = 0,00176107$

Ομοιότητα με χρήστη boutsika = 0,00000043  
 Τιμή χρήστη boutsika = 6.0  
 $\text{distribution}[34] += e^{-\frac{(1-6.0)^2}{2 \cdot \sigma^2}} \cdot 0,00000043 = 0,00176149$

Ομοιότητα με χρήστη skoridak = 0,00051730

Figure 25 Predicted value computation

As we have explained in the previous section, each value between 1 and 10 is examined to find the one, which maximizes the probability distribution for active user's rating of an unobserved object. To do this, we apply 5.7 relation for every user who has rated the item. The similarity between each user and the active, as we have found it by using Pennock's personality diagnosis (5.6), is also displayed together with each user's rating value. We will analytically discuss algorithm's efficiency below.

## 6.3 Extracting the Semantic Data

In order to export data from the relational database we use D2RQ [87][88].

**D2R Server**  
Running at <http://localhost:2020/>

Home | [albums](#) [artists](#) [artists\\_types](#) [genres](#) [itemratings](#) [makams](#) [music\\_files](#) [music\\_files2albums](#) [music\\_files2artists](#) [music\\_files2musical\\_instruments](#) [music\\_files2playlists](#) [music\\_types](#) [musical\\_instruments](#) [musical\\_instruments\\_types](#) [playlists](#) [representations](#) [rythms](#) [totalratings](#) [users](#)

This is a database published with D2R Server. It can be accessed using

1. your plain old web browser
2. Semantic Web browsers
3. SPARQL clients.

**1. HTML View**

You can use the navigation links at the top of this page to explore the database.

**2. RDF View**

You can also explore this database with **Semantic Web browsers** like [Tabulator](#) or [Disco](#). To start browsing, open this entry point URL in your Semantic Web browser:

**<http://localhost:2020/all>**

**3. SPARQL Endpoint**

SPARQL clients can query the database at this SPARQL endpoint:

**<http://localhost:2020/sparql>**

The database can also be explored using [this AJAX-based SPARQL Explorer](#).

Generated by [D2R Server](#)

Figure 26 D2RQ start page

Start page contains all tables' labels, as we have described them in section 5.3. By selecting a table we can view all instances inserted in it and select one for more details. The table instances are characterized by the serial number (id) noted in the database table. A particular link is also created for each item, formed like this: <http://servername:2020/resource/databasetable/id>, where servername is denoted as localhost for our example, databasetable could be one of the table names e.g. music\_files and id is the serial number of the item e.g. #100.

All music\_files

[Home](#) | [albums](#) [artists](#) [artists\\_types](#) [genres](#) [itemratings](#) [makams](#) [music\\_files](#) [music\\_files2albums](#) [music\\_files2artists](#) [music\\_files2musical\\_instruments](#) [music\\_files2playlists](#) [music\\_types](#) [musical\\_instruments](#) [musical\\_instruments\\_types](#) [playlists](#) [representations](#) [rythms](#) [totalratings](#) [users](#)

- [music\\_files #1](#)  
http://localhost:2020/resource/music\_files/1
- [music\\_files #10](#)  
http://localhost:2020/resource/music\_files/10
- [music\\_files #100](#)  
http://localhost:2020/resource/music\_files/100
- [music\\_files #101](#)  
http://localhost:2020/resource/music\_files/101

Figure 27 Music files list

The particular page of each item shows its private information in a form of triplets (predicate, property, value), which is table fields and its relations to other tables also. Studying the example below, we observe that the following issues are represented:

- The music file relations to albums (by clicking the link we will see which album collections include this file)
- The music file relations to artists (by following the link we will view the artists who take part in this file production)
- The fields makam, representation, rhythm, and type are links that also go to the particular semantic pages of the assigned values (according to the foreign keys relations explained in section 6.3)
- Some of the table columns that are not filled (like genre) do not appear on this page
- File's ratings are also displayed

Property	Value
is vocab:itemratings_item_id of	db:itemratings
rdfs:label	music_files #100
is vocab:music_files2albums_music_file_id of	<http://localhost:2020/resource/music_files2albums/100>
is vocab:music_files2artists_music_file_id of	<http://localhost:2020/resource/music_files2artists/110>
vocab:music_files_code	Ο ευσχημων Ιωσήφ
vocab:music_files_creation_date	2009-09-29 (xsd:date)
vocab:music_files_creation_user	ranutsu
vocab:music_files_description	&nbsp;
vocab:music_files_id	100 (xsd:int)
vocab:music_files_last_update_date	2009-09-29 (xsd:date)
vocab:music_files_last_update_user	ranutsu
vocab:music_files_location	PS1_IN_Track28.mp3
vocab:music_files_makam	<http://localhost:2020/resource/makams/21>
vocab:music_files_name	Ο ευσχημων Ιωσήφ
vocab:music_files_on_line	true (xsd:boolean)
vocab:music_files_priority	100 (xsd:int)
vocab:music_files_representation	<http://localhost:2020/resource/representations/8>
vocab:music_files_rythm	<http://localhost:2020/resource/rythms/1>
vocab:music_files_size	795837 (xsd:double)
vocab:music_files_type	<http://localhost:2020/resource/music_types/22>
is vocab:totalratings_id of	<http://localhost:2020/resource/totalratings/100>
rdf:type	vocab:music_files

Figure 28 Music file #100 particular page

## 6.4 Dataset

The dataset we used in our experiments contains 1.500 ratings, scaling from 0 to 10, derived from 63 users on 208 music files (items). The music files have been collected from several sources (CDs, personal collections, live recordings etc), documented, organised and stored in the database schema described above.

The system users are musicians from the [86] <http://www.analogion.com> community. This is an international musicological forum about byzantine music. Its members are exclusively chanters that are experts in byzantine music theory and practice. They have been asked to access the trial site, make multiple searches on the files stored in the system and rate them according to their interests.

The ratings have been collected in a 20-days period, and used to examine the performance of the system. The results are discussed in the next section.

## 6.5 Collaborative Filtering Algorithm Evaluation

To examine the efficiency of our algorithm, we will proceed to an example. Assuming that user rmanos is logged in, the following recommendations' list he will get by the system will be displayed.

### Δείτε Επίσης

1. [Το προσταχθέν](#) Predicted value **10.0**/10.0 [Λεπτομέρειες](#)
2. [Τα πάθη τα σεπτά](#) Predicted value **10.0**/10.0 [Λεπτομέρειες](#)
3. [Προ της γεννήσεως...](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
4. [Κεκραγάριον αργοσύντομον](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
5. [Πόρνη προσήλθε σοι](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
6. [Προκαθάρωμεν εαυτούς...](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
7. [Χαίρε Νύμφη - Αλληλούια](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
8. [Αλληλουάριον](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
9. [Ε Ωδή κανόνος Αγ. Γεωργίου, "Ορθρίσωμεν... Χριστός Ανέστη... Αναστάς..."](#) Predicted value **9.0**/10.0  
[Λεπτομέρειες](#)
10. ["Δόξα... Ανέτειλε το έαρ..."](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
11. [Μαρτυρικόν - Θεοτοκίον](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
12. ["Ατενίσαι το όμμα..."](#) Predicted value **9.0**/10.0 [Λεπτομέρειες](#)
13. [Αύτη ημέρα Κυρίου](#) Predicted value **8.0**/10.0 [Λεπτομέρειες](#)

14. [Θεαρχίω νεύματι](#) Predicted value 8.0/10.0 [Λεπτομέρειες](#)
15. [Εν τη γεννήσει την Παρθενίαν](#) Predicted value 8.0/10.0 [Λεπτομέρειες](#)
16. [Προς τίνα καταφύγω άλλην αγνή](#) Predicted value 8.0/10.0 [Λεπτομέρειες](#)
17. [Δεύτε πιστοί... Δόξα πατρί... Δεύτε χριστοφόροι λαοί...](#) Predicted value 8.0/10.0 [Λεπτομέρειες](#)
18. [Επεσκέψατο ημάς](#) Predicted value 8.0/10.0 [Λεπτομέρειες](#)
19. [Ο γλυκασμός των Αγγέλων](#) Predicted value 8.0/10.0 [Λεπτομέρειες](#)
20. [Χαίρε Νύμφη](#) Predicted value 8.0/10.0 [Λεπτομέρειες](#)

.....

75. [Ωδή Γ Παρακλητικού Κανόνος](#) Predicted value 4.0/10.0 [Λεπτομέρειες](#)
76. ["Ελαμψεν η χάρις Σου..."](#) Predicted value 3.0/10.0 [Λεπτομέρειες](#)

As we can observe from the above example, there are several items with different predicted values. We will now choose two recommendations from this list to check how the predicted value is calculated.

Firstly we will view the details (Λεπτομέρειες) of the No 1 item named as "Το προσταχθέν"

### Process to estimate predicted value about "Το προσταχθέν"

#### Test value 1

*Similarity probability with user babistserk = 0,3076*  
*babistserk's item value = 10.0*

$$\text{distribution}[1] += e^{-(1-10.0)^2/2*\sigma^2} * 0,3076 = 0,2780$$

*Similarity probability with user koutenp = 0,0784*  
*Koutenp's item value = 10.0*

$$\text{distribution}[1] += e^{-(1-10.0)^2/2*\sigma^2} * 0,0784 = 0,3489$$

*Similarity probability with user dimbouk = 0,0062*  
*Dimbouk's item value = 8.0*

$$\text{distribution}[1] += e^{-(1-8.0)^2/2*\sigma^2} * 0,0062 = 0,3547$$

*Similarity probability with user boutsika = 0,0642*  
*Boutsika's item value = 9.0*

$$\text{distribution}[1] += e^{-(1-9.0)^2/2*\sigma^2} * 0,0642 = 0,4140$$

*Similarity probability with user mike = 0,0224*  
*Mike's item value = 7.0*

$$\text{distribution}[1] += e^{-(1-7.0)^2/2*\sigma^2} * 0,0224 = 0,4354$$

### Test value 2

*Similarity probability with user babistserk = 0,3076*

*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(2-10.0)^2/2*\sigma^2} * 0,3076 = 0,2839$$

*Similarity probability with user koutenp = 0,0784*

*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(2-10.0)^2/2*\sigma^2} * 0,0784 = 0,3563$$

*Similarity probability with user dimbouk = 0,0062*

*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(2-8.0)^2/2*\sigma^2} * 0,0062 = 0,3623$$

*Similarity probability with user boutsika = 0,0642*

*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(2-9.0)^2/2*\sigma^2} * 0,0642 = 0,4227$$

*Similarity probability with user mike = 0,0224*

*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(2-7.0)^2/2*\sigma^2} * 0,0224 = 0,4444$$

### Test value 3

*Similarity probability with user babistserk = 0,3076*

*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(3-10.0)^2/2*\sigma^2} * 0,3076 = 0,2893$$

*Similarity probability with user koutenp = 0,0784*

*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(3-10.0)^2/2*\sigma^2} * 0,0784 = 0,3631$$

*Similarity probability with user dimbouk = 0,0062*

*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(3-8.0)^2/2*\sigma^2} * 0,0062 = 0,3691$$

*Similarity probability with user boutsika = 0,0642*

*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(3-9.0)^2/2*\sigma^2} * 0,0642 = 0,4305$$

*Similarity probability with user mike = 0,0224*

*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(3-7.0)^2/2*\sigma^2} * 0,0224 = 0,4525$$

#### **Test value 4**

*Similarity probability with user babistserk = 0,3076*

*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(4-10.0)^2/2*\sigma^2} * 0,3076 = 0,2940$$

*Similarity probability with user koutenp = 0,0784*

*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(4-10.0)^2/2*\sigma^2} * 0,0784 = 0,3690$$

*Similarity probability with user dimbouk = 0,0062*

*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(4-8.0)^2/2*\sigma^2} * 0,0062 = 0,3751$$

*Similarity probability with user boutsika = 0,0642*

*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(4-9.0)^2/2*\sigma^2} * 0,0642 = 0,4374$$

*Similarity probability with user mike = 0,0224*

*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(4-7.0)^2/2*\sigma^2} * 0,0224 = 0,4596$$

#### **Test value 5**

*Similarity probability with user babistserk = 0,3076*

*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(5-10.0)^2/2*\sigma^2} * 0,3076 = 0,2981$$

*Similarity probability with user koutenp = 0,0784*

*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(5-10.0)^2/2*\sigma^2} * 0,0784 = 0,3741$$

*Similarity probability with user dimbouk = 0,0062*



*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(5-8.0)^2/2*\sigma^2} * 0,0062 = 0,3803$$

*Similarity probability with user boutsika = 0,0642*

*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(5-9.0)^2/2*\sigma^2} * 0,0642 = 0,4433$$

*Similarity probability with user mike = 0,0224*

*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(5-7.0)^2/2*\sigma^2} * 0,0224 = 0,4656$$

### **Test value 6**

*Similarity probability with user babistserk = 0,3076*

*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(6-10.0)^2/2*\sigma^2} * 0,3076 = 0,3015$$

*Similarity probability with user koutenp = 0,0784*

*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(6-10.0)^2/2*\sigma^2} * 0,0784 = 0,3784$$

*Similarity probability with user dimbouk = 0,0062*

*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(6-8.0)^2/2*\sigma^2} * 0,0062 = 0,3846$$

*Similarity probability with user boutsika = 0,0642*

*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(6-9.0)^2/2*\sigma^2} * 0,0642 = 0,4481$$

*Similarity probability with user mike = 0,0224*

*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(6-7.0)^2/2*\sigma^2} * 0,0224 = 0,4705$$

### **Test value 7**

*Similarity probability with user babistserk = 0,3076*

*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(7-10.0)^2/2*\sigma^2} * 0,3076 = 0,3041$$

*Similarity probability with user koutenp = 0,0784*  
*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(7-10.0)^2/2*\sigma^2} * 0,0784 = 0,3817$$

*Similarity probability with user dimbouk = 0,0062*  
*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(7-8.0)^2/2*\sigma^2} * 0,0062 = 0,3879$$

*Similarity probability with user boutsika = 0,0642*  
*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(7-9.0)^2/2*\sigma^2} * 0,0642 = 0,4518$$

*Similarity probability with user mike = 0,0224*  
*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(7-7.0)^2/2*\sigma^2} * 0,0224 = 0,4743$$

### **Test value 8**

*Similarity probability with user babistserk = 0,3076*  
*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(8-10.0)^2/2*\sigma^2} * 0,3076 = 0,3060$$

*Similarity probability with user koutenp = 0,0784*  
*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(8-10.0)^2/2*\sigma^2} * 0,0784 = 0,3841$$

*Similarity probability with user dimbouk = 0,0062*  
*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(8-8.0)^2/2*\sigma^2} * 0,0062 = 0,3903$$

*Similarity probability with user boutsika = 0,0642*  
*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(8-9.0)^2/2*\sigma^2} * 0,0642 = 0,4545$$

*Similarity probability with user mike = 0,0224*  
*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(8-7.0)^2/2*\sigma^2} * 0,0224 = 0,4769$$

**Test value 9**

*Similarity probability with user babistserk = 0,3076*  
*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(9-10.0)^2/2*\sigma^2} * 0,3076 = 0,3072$$

*Similarity probability with user koutenp = 0,0784*  
*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(9-10.0)^2/2*\sigma^2} * 0,0784 = 0,3855$$

*Similarity probability with user dimbouk = 0,0062*  
*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(9-8.0)^2/2*\sigma^2} * 0,0062 = 0,3918$$

*Similarity probability with user boutsika = 0,0642*  
*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(9-9.0)^2/2*\sigma^2} * 0,0642 = 0,4560$$

*Similarity probability with user mike = 0,0224*  
*Mike's item value= 7.0*

$$\text{distribution}[1] += e^{-(9-7.0)^2/2*\sigma^2} * 0,0224 = 0,4783$$

**Test value 10**

*Similarity probability with user babistserk = 0,3076*  
*Babistserk's item value= 10.0*

$$\text{distribution}[1] += e^{-(10-10.0)^2/2*\sigma^2} * 0,3076 = 0,3076$$

*Similarity probability with user koutenp = 0,0784*  
*Koutenp's item value= 10.0*

$$\text{distribution}[1] += e^{-(10-10.0)^2/2*\sigma^2} * 0,0784 = 0,3860$$

*Similarity probability with user dimbouk = 0,0062*  
*Dimbouk's item value= 8.0*

$$\text{distribution}[1] += e^{-(10-8.0)^2/2*\sigma^2} * 0,0062 = 0,3922$$

*Similarity probability with user boutsika = 0,0642*  
*Boutsika's item value= 9.0*

$$\text{distribution}[1] += e^{-(10-9.0)^2/2*\sigma^2} * 0,0642 = 0,4564$$

*Similarity probability with user mike = 0,0224*

*Mike's item value = 7.0*

$\text{distribution}[1] += e^{-(10-7.0)^2/2*\sigma^2} * 0,0224 = 0,4785$

**Consequently the value that maximizes distribution is 10.0**

We can now make the following observations:

- By examining the similarity probabilities between user rmanos and the other users that have rated the item "Το προσταχθέν", we see that our active user is more similar to user babistserk who has rated the item with value 10.0 (0,3076)
- Probability similarity to user mike, who has rated the item with 7.0 is 0,0224, as well as probability similarity to user dimbouk (rating value 8.0) is 0,0062
- We can also view that the minimum rating value that other users have given to item is 7.0.
- The distribution follows an increasing line that is maximized when value is 10.0
- The distribution presents great increment for the first 6 values from 1.0 to 6.0 (approximately 0.8 unit per step), while it is least increased for the last 3 values, that is from 7.0 to 10.0 (as it is shown in figure 29)

From the above observations we can conclude that:

- the final predicted value is near to the values that users with highest similarity have given
- The distribution takes its highest value, in case the testing value has been given to the item by at least one more user
- Users with minimal similarity to the active do not affect meaningfully the result (they have also low impact in the estimation of distribution)

Let us examine one more recommendation to support our observations:

We take the last item from the recommendations list that is item #76 "Ελαμψεν η χάρις Σου... "

### Process to estimate predicted value about "Ελαμψεν η χάρις Σου... "

#### Test value 1

*Similarity probability with user babistserk = 0,3076*

*babistserk's item value = 3.0*

$$\text{distribution}[76] += e^{-(1-3.0)^2/2*\sigma^2} * 0,3076 = 0,3060$$

*Similarity probability with user koutenp = 0,0784*

*koutenp's item value = 5.0*

$$\text{distribution}[76] += e^{-(1-5.0)^2/2*\sigma^2} * 0,0784 = 0,3829$$

*Similarity probability with user boutsika = 0,0642*

*boutsika's item value = 2.0*

$$\text{distribution}[76] += e^{-(1-2.0)^2/2*\sigma^2} * 0,0642 = 0,4471$$

#### Test value 2

$$\text{distribution}[76] = 0,4490$$

#### Test value 3

$$\text{distribution}[76] = 0,4498$$

#### Test value 4

$$\text{distribution}[76] = 0,4494$$

#### Test value 5

$$\text{distribution}[76] = 0,4480$$

#### Test value 6

distribution[76] = 0,4454

**Test value 7**

distribution[76] = 0,4418

**Test value 8**

distribution[76] = 0,4371

**Test value 9**

distribution[76] = 0,4313

**Test value 10**

distribution[76] = 0,4246

**Consequently the value that maximizes distribution is 3.0**

**Figure 30 Predicted value distribution for item #76**

As we can see from the second example, the distribution takes its maximum value near the value that user with highest similarity (babistserk) has given. The values near this one also maximize the distribution, while the rest minimize it.

## 6.6 User Evaluation

### 6.6.1 The Questionnaire

In order to evaluate the reliability of the results, we have compiled a short questionnaire that users were asked to answer after using the system. The formed questions and their certain answers are the following:

**1. Are you satisfied with the system's results?**

- a. Absolutely satisfied
- b. Very satisfied
- c. Simply satisfied
- d. Disappointed
- e. Very disappointed

**2. How would you rate the recommended items?**

- a. Exactly the same
- b. About the same
- c. In some cases differently
- d. Absolutely differently

**3. How interesting do you find the recommended files?**

- a. Absolutely interesting
- b. Very interesting
- c. Neutral
- d. Unfavorable

**4. Are you satisfied with the ordering in the recommendations' list?**

- a. Absolutely satisfied
- b. Very satisfied
- c. Simply satisfied
- d. Disappointed
- e. Very disappointed

**5. Would you select any of the recommended files on your own?**

- a. All of them
- b. Some of them
- c. Least of them
- d. None of them



## 6. Do you find the recommendations' list helpful?

- a. Essential
- b. Helpful
- c. Neutral
- d. Lousy

As a conclusion from the above results, we can say that the majority of users find the system useful. In general, the system comes close to users' preferences, but there are some differences observed. The recommendations list is helpful to final users, since they save time from searching through the data to indicate other files of their interest.

### 6.6.2 Users' Opinion

In this section we will present what users commented on our system, and how useful they found the system in music technology and research:

*"Congratulations for your efforts. Such attempts to sort, catalog, and index electronic data are useful and help the development of computer science."*

*Evangelos Theodorou*

*"Nice work! Congratulations!"*

*Nikolaos Tachliampouris*

*"Nice work, we hope it will not stop but it will get evolved more and more."*

*Mamais Michalis*

# Chapter 7

---

## *Conclusions*

### **7.1 Summary**

In this thesis we have presented the design and implementation of a byzantine music application framework. The goal of this study was to analyze, organize, store and efficiently utilize special characteristics of byzantine music, in order to contribute to the research and study of music. Users of the system take advantage of the technologies used to document, organize and search efficiently their music files in a scientific way.

Firstly, we have designed a database system to store data and metadata that concern a music file, taking into account the excellence and variety of byzantine music. We have created the appropriate structures and their relations in order to save and restore data with respect to their sensitivity and completeness.

We have also taken advantage of the growing up of semantic web technologies to make content accessible in such a way, and provide it to a more understandable way. Doing so, we have extended the current music rdf schema to include and support special information about byzantine music, without concerning about the general musicological schema, but taking benefit of it. We have also made use of D2RQ server in order to make the relational database's content semantically available.

Besides, we have developed a recommendation algorithm based on collaborative filtering techniques to advance the functionality of our system and help the user indicate files that meet his tastes quickly. Hereupon, we have studied collaborative filtering techniques and formed a standard that would make use of users' similarities to predict one's rating values and recommend files according to his interests.

All of the above issues have been designed and implemented with the support of the appropriate development of web services, so that our system could be independent of building technologies, interoperable, stretchable and able to join different methodologies, as well as public methods and techniques for use of external systems.

We have also discussed the main issues of the state of the art literature review regarding the supporting technologies, and presented the technological approaches and theories, as well as the limitations.

Finally, in order to prove the efficiency of our application, we have tested over a specialized group, apart from musicians, researchers of byzantine music and chanters. A questionnaire has also been formed and answered by the test users of the system to draw conclusions about performance of the system. As the results have shown, several improvements could be made to the system, which we will discuss below.

## **7.2 Future Extensions**

In this thesis we have tried to join recently researched technologies with byzantine music world and introduce a system that would benefit from the former to advance the latter. However, this is just the beginning, since there is variety of subjects that should be studied in order to take full advantage of the existing technologies. We will present the ones we consider as crucial, so that to cover the discussed themes at least.

First, because of the data form, we could make use of a hybrid recommendation method that would utilize content information as well. Rousidis [81] in his work proposes the use of a content-based method to complete un-filled cells in the initial users to items ratings' matrix. This could solve the cold-start problem.

As an extension to the previous one, incremental updating techniques can be also added, in order to save time from the initial steps of recommendation algorithm, eigen-decomposing and user mapping. At a starting point, we have selected a simplified version of the algorithm described in [81], that is most commonly like Pennock's [84] suggestion, making use of PCA as well.

We could also extend the supported rdf schema more to semantically describe a wider set of music ontologies, including Eastern Music in general.

# Bibliography

---

- [1] M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, T. Newling. *Patterns: Service Oriented Architecture and Web Services*. Redbooks, 2004
- [2] T. Bray, J. Paoli, Sperberg-McQueen, C. M. and E. Maler. *Extensível Markup Language (XML) 1.0. Second Edition*. <http://www.w3.org/TR/2000/REC-xml-20001006>, October 2000
- [3] H. Kreger. *Web Services Conceptual Architecture*. IBM Software Group, May 2001
- [4] T. Clements. *Overview of SOAP Web Services-Technical Overviews*. Sun Microsystems, August 2001
- [5] S. Graham, S. Simeonov, T. Boubez, D. Davis, G. Daniels, N. Yuichi and R. Neyama. *Building Web Services With Java*. SAMS, pages 581, 2002
- [6] W. Cohen, W. Fan, *Web-collaborative filtering: recommending music by crawling the Web*, Computer Networks (Amsterdam, Netherlands: 1999), v.33 n.1–6, pages 685–698, 2000
- [7] M. Hendricks, B. Galbraith, R. Irani, J. Milbery, T. Modi, A. Tost, A. Toussaint, S. J. Basha and S. Cable. *Professional Java Web Services*. Wrox, pages 588, January 2002
- [8] E. Christensen, F. Curbera, G. Meredith and S. Weerawarana. *Web Services Description Language (WSDL)*. March 2001
- [9] P. Brittenham, F. Cubera, D. Ehnebuske, S. Graham. *Understanding WSDL in a UDDI registry, Part 1*. September 2001.
- [10] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, D. Winer, *SOAP: Simple Object AccessProtocol*, April 2000
- [11] S. Seely. *SOAP Cross Platform Web ServiceDevelopment Using XML*. Prentice Hall PTR, pages 391, 2002
- [12] ARIBA, INC., International Business Machines Corporation and Microsoft Corporation. *UDDI Data Structures Reference*. September, 2000
- [13] D. Ehnebuske, D. Rogers, C. V. Riegen. *UDDI Version 2.0 Data Structure reference*. June, 2001. <http://www.uddi.org/pubs/DataStructure-V2.03-Published-20020719.pdf>

- [14] A. Rajaram. *Overview of UDDI. Web Services - Technical Overviews*. Sun Microsystems. August, 2001
- [15] T. Berners-Lee, J. Hendler, O. Lassila. *The Semantic Web*. Scientific American. v.284 n.5, pages 34-43, 2001
- [16] A. Gerber, A. Barnard, A. Van der Merwe. *A Semantic Web Status Model, Integrated Design & Process Technology*. Special Issue: IDPT 2006
- [17] A. Gerber, A. Van der Merwe, A. Barnard. *A Functional Semantic Web architecture*. European Semantic Web Conference 2008, ESWC'08, Tenerife, June 2008
- [18] W3C. *Semantic Web - XML2000, slide 10*. <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>, 2008
- [19] W3C. *Representing Knowledge in the Semantic Web, slide 7*. <http://www.w3c.it/talks/2005/openCulture/slide7-0.html>, 2008
- [20] W3C. *Semantic Web, and Other Technologies to Watch, slide 24*. [http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#\(24\)](http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24)), 2008
- [21] T. Berners-Lee. *Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web*, RFC 1630, CERN, June 1994
- [22] J. Kunze. *Functional Recommendations for Internet Resource Locators*. RFC 1736, IS&T, UC Berkeley, February 1995
- [23] K. Sollins, L. Masinter. *Functional Requirements for Uniform Resource Names*. RFC 1737, Internet Engineering Task Force, December 1994
- [24] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler. *Extensive Markup Language (XML) 1.0 (Second Edition)*, W3C REC-xml, October 2000
- [25] S. Thompson, D. Beech, M. Maloney, N. Mendelson. *XML Schema Part 1: Structures*, W3C REC REC-xmlschema-1-20010502, May 2001

- [26] T. Berners-Lee, R. Fielding, L. Masinter. *IETF (Internet Engineering Task Force) RFC 2396: Uniform Resource Identifiers (URI):Generic Syntax*, 1998
- [27] O. Lassila, R. Swick. *Resource Description Framework (RDF) Model and Syntax Specification*, World Wide Web Consortium, Recommendation REC-rdf-syntax -19990222, February 1999
- [28] D. Brickley, R.V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema*, World Wide Web Consortium, <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>, 2004
- [29] P.F. Patel-Schneider, P. Hayes, I. Horrocks. *OWL Web OntologyLanguage Semantics and Abstract Syntax*, W3C Recommendation, 2003
- [30] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2002
- [31] RuleML. *The Rule Markup Language Initiative*. <http://www.ruleml.org/>
- [32] M. Kifer. *Rule Interchange Format: The Framework*. Web Reasoning and Rule Systems. Lecture Notes in Computer Science, 2008
- [33] P. Bijan. *Cautiously Approaching SWRL*. <http://www.mindswap.org>, 2005
- [34] R. Barta. *AsTMa 1.3 Language Specification*. Technical report, Bond University, 2003
- [35] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, D. Suci. *A Query Language for XML*. *Computer Networks*, v.31 n.11–16, pages 1155–1169, 1999
- [36] Lars Marius Garshol. *tolog - a topic maps query language*. In Proceedings of the First International Workshop on Topic Maps Research and Applications (TMRA) LNCS 3873, pages 183–196, 2005
- [37] S. Berger, F. Bry, T. Furche, B. Linse, A. Schroeder. *Beyond XML and RDF: The versatile Web query language Xcerpt*. In Proceedings of the 15th International World Wide Web Conference (WWW), pages 1053–1054, 2006
- [38] S. Schaffert, F. Bry. *Querying the Web Reconsidered: A Practical Introduction to Xcerpt*. In Proc. Extreme Markup Languages, August, 2004
- [39] M. Lacher, Stefan Decker. *RDF, Topic Maps, and the Semantic Web. Markup Languages: Theory and Practice*, v.3 n.3, pages 313–331, December 2001

- [40] Libby Miller, A. Seaborne, A. Reggiori. *Three implementations of SquishQL, a simple RDF query language*. In Proc. of the International Semantic Web Conference, pages 423–435, 2002
- [41] E. Prud'hommeaux, A. Seaborne. *SPARQL Query Language for RDF*. Candidate recommendation. W3C, <http://www.w3.org/TR/rdf-sparql-query/>, June 2007
- [42] G. Karvounarakis, A. Magkanaraki, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl, K. Tolle. *Querying the Semantic Web with RQL*. Computer Networks and ISDN Systems Journal, v.42 n.5, pages 617–640, August 2003
- [43] J. Broekstra, A. Kampman. *SeRQL: A Second Generation RDF Query Language*. In Proc. SWAD-Europe Workshop on Semantic Web Storage and Retrieval, 2003
- [44] E. Prud'hommeaux. *Algae RDF Query Language*. <http://www.w3.org/2004/05/06-Algae/>, 2004
- [45] S. Decker, M. Sintek, A. Billig, N. Henze, P. Dolog, W. Nejdl, A. Harth, A. Leicher, S. Busse, J. Luis Ambite, M. Weathers, G. Neumann, U. Zdun. *TRIPLE - an RDF rule language with context and use cases*. In Proc. of Rule Languages for Interoperability, 2005
- [46] R. Fikes, P. Hayes, I. Horrocks. *OWL-QL – A Language for Deductive Query Answering on the Semantic Web*. Journal of Web Semantics, v.2 n.1, pages 19–29, 2004
- [47] J. Broekstra, A. Kampman. *Query Language Definition*. On-To-Knowledge project deliverable 9. March 2001
- [48] A. Kampman, F. van Harmelen. *Sesame's Interpretation of RDF Schema*. Administrator Nederland bv. Version 1.2., April 2001
- [49] J. Broekstra, A. Kampman, F. van Harmelen. *Sesame: a Generic Architecture for Storing and Querying RDF and RDF Schema*. 1st International Semantic Web Conference (ISWC2002), Sardinia, Italy, June 2002
- [50] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, K. Tolle. *The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases*. 2nd International Workshop on the Semantic Web (SemWeb'01), in conjunction with Tenth International World Wide Web Conference (WWW10), pages 1-13, Hong Kong, May 2001

- [51] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis. *On Storing Voluminous RDF Descriptions: The case of Web Portal Catalogs*. In Proceedings of the 4th International Workshop on the Web and Databases (WebDB'01)- In conjunction with ACM SIGMOD/PODS, Santa Barbara, CA, May 2001
- [52] B. McBride. *Jena: Implementing the RDF Model and Syntax Specification*. In Steffen Staab et al (eds.): Proceedings of the Second International Workshop on the Semantic Web-SemWeb2001, May 2001
- [53] U. Shardanand and P. Maes. *Social information filtering: algorithms for automating "word of mouth"*. Proceedings of the SIGCHI conference on Human factors in computing systems, pages 210–217, 1995
- [54] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999
- [55] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989
- [56] N. Belkin and B. Croft. *Information Filtering and Information Retrieval*. Communications of the ACM, v.35 n. 12, pages 29-37, 1992
- [57] D.W. Oard. *The State of the Art in Text Filtering*. User Modeling and User Adapted Interaction, v.7 n.3, pages 141-178, 1997
- [58] R. J. Mooney, P. N. Bennet and L. Roy. *Book Recommending Using Text Categorization with Extracted Information*. In Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98-08. AAAI Press, 1998
- [59] M. J. Pazzani and D. Billsus. *Learning and Revising User Profiles: The identification of interesting web sites*. Machine Learning, v.27, pages 313-331, 1997
- [60] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. *Using collaborative filtering to weave an information tapestry*. Communications of the ACM, v.35 n.12, pages 61–70, 1992
- [61] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. *Recommending and evaluating choices in a virtual community of use*. Proceedings of the SIGCHI conference on Human factors in computing systems, pages 194–201, 1995



- [62] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. *GroupLens: applying collaborative filtering to Usenet news*. Communications of the ACM, v.40 n.3, pages 77–87, 1997
- [63] P. Resnick, Center for Coordination Science, and Sloan School of Management. *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. Center for Coordination Science, Sloan Massachusetts Institute of Technology, Sloan School of management, 1994
- [64] J.B. Schafer, J. Konstan, and J. Riedi. *Recommender systems in e-commerce*. Proceedings of the 1st ACM conference on Electronic commerce, pages 158–166, 1999
- [65] G. Linden, B. Smith, and J. York. *Amazon. com recommendations: item-to-item collaborative filtering*. Internet Computing, IEEE, v.7 n.1, pages 76–80, 2003
- [66] J.S. Breese, D. Heckerman, C. Kadie, et al. *Empirical analysis of predictive algorithms for collaborative filtering*. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 461, 1998
- [67] J.L. Herlocker, J.A. Konstan, and J. Riedl. *An algorithmic framework for performing collaborative filtering*. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pages 230–237, 1999
- [68] G.R. Xue, C. Lin, Q. Yang, W.S. Xi, H.J. Zeng, Y. Yu, and Z. Chen. *Scalable collaborative filtering using cluster-based smoothing*. Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 114–121, 2005
- [69] J. Wang, A.P. de Vries, and M.J.T. Reinders. *Unifying user-based and item-based collaborative filtering approaches by similarity fusion*. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 501–508, 2006
- [70] M. Deshpande and G. Karypis. *Item-based top-N recommendation algorithms*. ACM Transactions on Information Systems (TOIS), v.22 n.1, pages 143-177, January 2004
- [71] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. *Item-based collaborative filtering recommendation algorithms*. Proceedings of the 10th international conference on World Wide Web, pages 285–295, 2001

- [72] P. Melville, R.J. Mooney, and R. Nagarajan. *Content-boosted collaborative filtering*. Proceedings of the SIGIR-2001 Workshop on Recommender Systems, 9, 2001
- [73] B.M. Sarwar, G. Karypis, JA Konstan, and J. Riedl. *Analysis of recommender algorithms for e-commerce*. Proceedings of the 2nd ACM Conference on Electronic Commerce, pages 158–167, 2000
- [74] A.I. Schein, A. Popescul, L.H. Ungar, and D.M. Pennock. *Methods and metrics for cold-start recommendations*. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pages 253–260, 2002
- [75] D. Maltz and K. Ehrlich. *Pointing the way: active collaborative filtering*. Proceedings of the SIGCHI conference on Human factors in computing systems, pages 202–209, 1995
- [76] R. Burke. *Integrating Knowledge-based and Collaborative-filtering Recommender Systems*. Proceedings of the Workshop on AI and Electronic Commerce, 1999
- [77] N. Ramakrishnan, BJ Keller, BJ Mirza, AY Grama, and G. Karypis. *Privacy risks in recommender systems*. Internet Computing, IEEE, v.5 n.6, pages 54–63, 2001
- [78] C.C. Aggarwal, J.L. Wolf, K.L. Wu, and P.S. Yu. *Horning hatches an egg: a new graph-theoretic approach to collaborative filtering*. Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 201–212, 1999
- [79] A. Kohrs and B. Merialdo. *Clustering for collaborative filtering applications*. Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation, Vienna, Austria, February, 1999
- [80] The Web Kanzaki. Japan, Music, Internet & Computer, <http://www.kanzaki.com/ns/music.rdf>
- [81] J. Rousidis. *On Identifying trends in recommendation algorithms*. Master of Thesis, Computer Science Department, UOC, Heraklion, March 2007
- [82] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl. *An Algorithmic Framework for Performing Collaborative Filtering*. In Proc. of SIGIR, 1999

- [83] M. Papagelis, D. Plexousakis, T. Kutsuras. *A Method for Alleviating the Sparsity Problem in Collaborative Filtering Using Trust Inferences*. In Proceedings of the 3rd International Conference on Trust Management, 2005
- [84] D. Pennock, E. Horvitz, S. Lawrence, C. Lee Giles. *Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach*. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000), pages 473-480, Morgan Kaufmann, San Francisco, 2000
- [85] Glassfish Community, <https://glassfish.dev.java.net/>
- [86] Psaltologion, Analogion Forum, <http://www.analogion.com/>
- [87] C. Bizer, A. Seaborne. *D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs*. 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan, November 2004
- [88] C. Bizer, R. Cyganiak. *D2R Server - Publishing Relational Databases on the Semantic Web*. Poster at the 5th International Semantic Web Conference, Athens, USA, November 2006
- [89] O. Erling, I. Mikhailov. *RDF Support in the Virtuoso DBMS*, In *Conference on Social Semantic Web*, v.113GI, pages 59-68, 2007
- [90] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, D. Aumüller. *Triplify – Light-Weight Linked Data Publication from Relational Databases*. In Proceedings of Semantic Data Web Track of 18th International World Wide Web Conference (WWW 2009), Madrid, Spain, April 2009
- [91] E. Prudhommeaux. *SPASQL: SPARQL Support In MySQL*. <http://xtech06.usefulinc.com/schedule/paper/156>
- [92] SquirrelRDF. <http://jena.sourceforge.net/SquirrelRDF/>
- [93] Γρηγόριος Πρωτοψάλτης, Χουρμούζιος Χαρτοφύλακας, Χρύσανθος Προύσης, *Θεωρητικόν στοιχειώδες της Μουσικής*, Κωνσταντινούπολη 1859
- [94] Κηλτζανίδη Π., *Μεθοδική διδασκαλία Ελληνικής Μουσικής*, Κωνσταντινούπολη 1881
- [95] Καρά Σ., *Μέθοδος της ελληνικής μουσικής*, Αθήνα 1982
- [96] Κωνσταντίνου Πρωτοψάλτου, *Ερμηνεία της Εξωτερικής Μουσικής*, Κωνσταντινούπολη 1843

- [97] Μαυρουδή Π., *Κρηπίδα γενικών μαθημάτων Βυζαντινής μουσικής*, Μακεδονικό Ωδείο Βυζαντινής Μουσικής «Κοσμάς Μαδυτινός», Θεσσαλονίκη 2002
- [98] Μαυρουδή Π., *Η Ελληνορθόδοξη Θρησκευτική Μουσική Τέχνη*, Μακεδονικό Ωδείο Βυζαντινής Μουσικής «Κοσμάς Μαδυτινός», Θεσσαλονίκη 2007
- [99] Μαργαζιώτη Ι. *Θεωρητικόν Βυζαντινης Εκκλησιαστικής Μουσικής*, Αθήνα.
- [100] Οικουμενικό Πατριαρχείο, *Στοιχειώδης Διδασκαλία της Εκκλησιαστικής Μουσικής*, Κωνσταντινούπλη 1888
- [101] Παναγιωτόπουλου Δ., *Θεωρία και πράξη της Βυζαντινής Εκκλησιαστικής*, Σωτήρ, Αθήνα 1982
- [102] Χατζηθεοδώρου Γ., *Μέθοδος Διδασκαλίας της Βυζαντινής Εκκλησιαστικής Μουσικής*, Ο Μιχαήλ Πολυχρονάκης, Άγιος Νκόλαος Κρήτης 2004
- [103] Χρύσανθος Μαδύτων, *Εισαγωγή και Μέγα Θεωρητικόν της Μουσικής*, Κωνσταντινούπολη 1821
- [104] Ψάχου Κ., *Η παρασημαντική της Βυζαντινής Μουσικής*, Διόνυσος, Αθήνα 1978
- [105] A. Swartz, *MusicBrainz: A Semantic Web Service*, Intelligent Systems, v. 17, n. 1, pages 76-77, February 2002
- [106] Y. Raimond, S. Abdallah, M. Sandler, F. Giasson, *The music Ontology*, In Proceedings of the International Conference on Music Information Retrieval, 2007

# Appendix A

---

## Music Types rdf-schema extension

```

1. <owl:Class rdf:ID="byz_music_type_18">
2.   <rdfs:label>Βυζαντινή Μουσική</rdfs:label>
3.   <rdfs:comment></rdfs:comment>
4.   <rdfs:subClassOf rdf:resource="#Religious_Music"/>
5. </owl:Class>
6. <owl:Class rdf:ID="byz_music_type_19">
7.   <rdfs:label>Ειρμολογικά Μέλη</rdfs:label>
8.   <rdfs:comment></rdfs:comment>
9.   <rdfs:subClassOf rdf:resource="#byz_music_type_18"/>
10. </owl:Class>
11. <owl:Class rdf:ID="byz_music_type_22">
12.   <rdfs:label>Απολυτικά</rdfs:label>
13.   <rdfs:comment></rdfs:comment>
14.   <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
15. </owl:Class>
16. <owl:Class rdf:ID="byz_music_type_23">
17.   <rdfs:label>Δοξολογίες</rdfs:label>
18.   <rdfs:comment></rdfs:comment>
19.   <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
20. </owl:Class>
21. <owl:Class rdf:ID="byz_music_type_24">
22.   <rdfs:label>Κανόνες </rdfs:label>
23.   <rdfs:comment></rdfs:comment>
24.   <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
25. </owl:Class>
26. <owl:Class rdf:ID="byz_music_type_25">
27.   <rdfs:label>Κοντάκια</rdfs:label>
28.   <rdfs:comment></rdfs:comment>
29.   <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
30. </owl:Class>
31. <owl:Class rdf:ID="byz_music_type_26">
32.   <rdfs:label>Ωδές</rdfs:label>
33.   <rdfs:comment></rdfs:comment>
34.   <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
35. </owl:Class>
36. <owl:Class rdf:ID="byz_music_type_27">
37.   <rdfs:label>Ειρμιοί</rdfs:label>
38.   <rdfs:comment></rdfs:comment>
39.   <rdfs:subClassOf rdf:resource="#byz_music_type_26"/>
40. </owl:Class>
41. <owl:Class rdf:ID="byz_music_type_28">

```

42. <rdfs:label>Τροπάρια</rdfs:label>  
43. <rdfs:comment></rdfs:comment>  
44. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_26"/>  
45. </owl:Class>  
46. <owl:Class rdf:ID="byz\_music\_type\_29">  
47. <rdfs:label>Καταβασίες</rdfs:label>  
48. <rdfs:comment></rdfs:comment>  
49. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_26"/>  
50. </owl:Class>  
51. <owl:Class rdf:ID="byz\_music\_type\_30">  
52. <rdfs:label>Εξαποστειλάρια </rdfs:label>  
53. <rdfs:comment></rdfs:comment>  
54. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_19"/>  
55. </owl:Class>  
56. <owl:Class rdf:ID="byz\_music\_type\_31">  
57. <rdfs:label>Αυτόμελα</rdfs:label>  
58. <rdfs:comment></rdfs:comment>  
59. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_19"/>  
60. </owl:Class>  
61. <owl:Class rdf:ID="byz\_music\_type\_32">  
62. <rdfs:label>Προσόμοια</rdfs:label>  
63. <rdfs:comment></rdfs:comment>  
64. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_19"/>  
65. </owl:Class>  
66. <owl:Class rdf:ID="byz\_music\_type\_33">  
67. <rdfs:label>Στιχηρά Προσόμοια του Εσπερινού</rdfs:label>  
68. <rdfs:comment></rdfs:comment>  
69. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_32"/>  
70. </owl:Class>  
71. <owl:Class rdf:ID="byz\_music\_type\_34">  
72. <rdfs:label>Στιχηρά Προσόμοια των Αίνων</rdfs:label>  
73. <rdfs:comment></rdfs:comment>  
74. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_32"/>  
75. </owl:Class>  
76. <owl:Class rdf:ID="byz\_music\_type\_35">  
77. <rdfs:label>Στιχηρά Προσόμοια Δοξαστικά των Αίνων</rdfs:label>  
78. <rdfs:comment></rdfs:comment>  
79. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_34"/>  
80. </owl:Class>  
81. <owl:Class rdf:ID="byz\_music\_type\_36">  
82. <rdfs:label>Και Νυν</rdfs:label>  
83. <rdfs:comment></rdfs:comment>  
84. <rdfs:subClassOf rdf:resource="#byz\_music\_type\_35"/>  
85. </owl:Class>  
86. <owl:Class rdf:ID="byz\_music\_type\_37">  
87. <rdfs:label>Προσόμοια των αποστίχων του Εσπερινού</rdfs:label>

```
88. <rdfs:comment></rdfs:comment>
89. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
90. </owl:Class>
91. <owl:Class rdf:ID="byz_music_type_38">
92. <rdfs:label>Προσόμοια των αποστίχων του Όρθρου</rdfs:label>
93. <rdfs:comment></rdfs:comment>
94. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
95. </owl:Class>
96. <owl:Class rdf:ID="byz_music_type_39">
97. <rdfs:label>Προσόμοια Δοξαστικά του Εσπερινού</rdfs:label>
98. <rdfs:comment></rdfs:comment>
99. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
100. </owl:Class>
101. <owl:Class rdf:ID="byz_music_type_40">
102. <rdfs:label>Και Νυν</rdfs:label>
103. <rdfs:comment></rdfs:comment>
104. <rdfs:subClassOf rdf:resource="#byz_music_type_39"/>
105. </owl:Class>
106. <owl:Class rdf:ID="byz_music_type_41">
107. <rdfs:label>Προσόμοια Δοξαστικά Εωθινά</rdfs:label>
108. <rdfs:comment></rdfs:comment>
109. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
110. </owl:Class>
111. <owl:Class rdf:ID="byz_music_type_42">
112. <rdfs:label>Προσόμοια Δοξαστικά των αποστίχων του Εσπερινού</rdfs:label>
113. <rdfs:comment></rdfs:comment>
114. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
115. </owl:Class>
116. <owl:Class rdf:ID="byz_music_type_43">
117. <rdfs:label>Και Νυν</rdfs:label>
118. <rdfs:comment></rdfs:comment>
119. <rdfs:subClassOf rdf:resource="#byz_music_type_42"/>
120. </owl:Class>
121. <owl:Class rdf:ID="byz_music_type_44">
122. <rdfs:label>Προσόμοια Δοξαστικά των αποστίχων του Όρθρου</rdfs:label>
123. <rdfs:comment></rdfs:comment>
124. <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
125. </owl:Class>
126. <owl:Class rdf:ID="byz_music_type_45">
127. <rdfs:label>Και Νυν</rdfs:label>
128. <rdfs:comment></rdfs:comment>
129. <rdfs:subClassOf rdf:resource="#byz_music_type_44"/>
130. </owl:Class>
131. <owl:Class rdf:ID="byz_music_type_46">
132. <rdfs:label>Προσόμοια Ν Ψαλμού</rdfs:label>
133. <rdfs:comment></rdfs:comment>
```

```
134.         <rdfs:subClassOf rdf:resource="#byz_music_type_32"/>
135.     </owl:Class>
136.     <owl:Class rdf:ID="byz_music_type_47">
137.         <rdfs:label>Ευλογητάρια</rdfs:label>
138.         <rdfs:comment></rdfs:comment>
139.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
140.     </owl:Class>
141.     <owl:Class rdf:ID="byz_music_type_48">
142.         <rdfs:label>Στιχηρά</rdfs:label>
143.         <rdfs:comment></rdfs:comment>
144.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
145.     </owl:Class>
146.     <owl:Class rdf:ID="byz_music_type_49">
147.         <rdfs:label>Απόστιχα</rdfs:label>
148.         <rdfs:comment></rdfs:comment>
149.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
150.     </owl:Class>
151.     <owl:Class rdf:ID="byz_music_type_50">
152.         <rdfs:label>Κεκραγάρια</rdfs:label>
153.         <rdfs:comment></rdfs:comment>
154.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
155.     </owl:Class>
156.     <owl:Class rdf:ID="byz_music_type_51">
157.         <rdfs:label>Πασαπνοάρια</rdfs:label>
158.         <rdfs:comment></rdfs:comment>
159.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
160.     </owl:Class>
161.     <owl:Class rdf:ID="byz_music_type_52">
162.         <rdfs:label>Αντίφωνα</rdfs:label>
163.         <rdfs:comment></rdfs:comment>
164.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
165.     </owl:Class>
166.     <owl:Class rdf:ID="byz_music_type_53">
167.         <rdfs:label>Αναβαθμοί</rdfs:label>
168.         <rdfs:comment></rdfs:comment>
169.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
170.     </owl:Class>
171.     <owl:Class rdf:ID="byz_music_type_54">
172.         <rdfs:label>Προκείμενα</rdfs:label>
173.         <rdfs:comment></rdfs:comment>
174.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
175.     </owl:Class>
176.     <owl:Class rdf:ID="byz_music_type_55">
177.         <rdfs:label>Προκείμενα της ημέρας</rdfs:label>
178.         <rdfs:comment></rdfs:comment>
179.         <rdfs:subClassOf rdf:resource="#byz_music_type_54"/>
```



```
180. </owl:Class>
181. <owl:Class rdf:ID="byz_music_type_56">
182.     <rdfs:label>Προκείμενα εορτών Αγίων</rdfs:label>
183.     <rdfs:comment></rdfs:comment>
184.     <rdfs:subClassOf rdf:resource="#byz_music_type_54"/>
185. </owl:Class>
186. <owl:Class rdf:ID="byz_music_type_57">
187.     <rdfs:label>Προκείμενα Δεσποτικών Εορτών</rdfs:label>
188.     <rdfs:comment></rdfs:comment>
189.     <rdfs:subClassOf rdf:resource="#byz_music_type_54"/>
190. </owl:Class>
191. <owl:Class rdf:ID="byz_music_type_58">
192.     <rdfs:label>Καθίσματα</rdfs:label>
193.     <rdfs:comment></rdfs:comment>
194.     <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
195. </owl:Class>
196. <owl:Class rdf:ID="byz_music_type_59">
197.     <rdfs:label>Μεσώδια Καθίσματα</rdfs:label>
198.     <rdfs:comment></rdfs:comment>
199.     <rdfs:subClassOf rdf:resource="#byz_music_type_58"/>
200. </owl:Class>
201. <owl:Class rdf:ID="byz_music_type_60">
202.     <rdfs:label>Καθίσματα Όρθρου</rdfs:label>
203.     <rdfs:comment></rdfs:comment>
204.     <rdfs:subClassOf rdf:resource="#byz_music_type_58"/>
205. </owl:Class>
206. <owl:Class rdf:ID="byz_music_type_61">
207.     <rdfs:label>Αναστάσιμα Καθίσματα</rdfs:label>
208.     <rdfs:comment></rdfs:comment>
209.     <rdfs:subClassOf rdf:resource="#byz_music_type_58"/>
210. </owl:Class>
211. <owl:Class rdf:ID="byz_music_type_62">
212.     <rdfs:label>Λειτουργικά</rdfs:label>
213.     <rdfs:comment></rdfs:comment>
214.     <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
215. </owl:Class>
216. <owl:Class rdf:ID="byz_music_type_63">
217.     <rdfs:label>Τυπικά</rdfs:label>
218.     <rdfs:comment></rdfs:comment>
219.     <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
220. </owl:Class>
221. <owl:Class rdf:ID="byz_music_type_64">
222.     <rdfs:label>Μακαρισμοί</rdfs:label>
223.     <rdfs:comment></rdfs:comment>
224.     <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
225. </owl:Class>
```

```

226. <owl:Class rdf:ID="byz_music_type_65">
227.     <rdfs:label>Μεγαλυνάρια</rdfs:label>
228.     <rdfs:comment></rdfs:comment>
229.     <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
230. </owl:Class>
231. <owl:Class rdf:ID="byz_music_type_66">
232.     <rdfs:label>Καλοφωνικοί Ειρμοί</rdfs:label>
233.     <rdfs:comment></rdfs:comment>
234.     <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
235. </owl:Class>
236. <owl:Class rdf:ID="byz_music_type_67">
237.     <rdfs:label>Ιδιόμελα</rdfs:label>
238.     <rdfs:comment></rdfs:comment>
239.     <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
240. </owl:Class>
241. <owl:Class rdf:ID="byz_music_type_68">
242.     <rdfs:label>Στιχηρά Ιδιόμελα του Εσπερινού</rdfs:label>
243.     <rdfs:comment></rdfs:comment>
244.     <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
245. </owl:Class>
246. <owl:Class rdf:ID="byz_music_type_69">
247.     <rdfs:label>Αναστάσιμα</rdfs:label>
248.     <rdfs:comment></rdfs:comment>
249.     <rdfs:subClassOf rdf:resource="#byz_music_type_68"/>
250. </owl:Class>
251. <owl:Class rdf:ID="byz_music_type_70">
252.     <rdfs:label>Στιχηρά Ιδιόμελα των Αίνων</rdfs:label>
253.     <rdfs:comment></rdfs:comment>
254.     <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
255. </owl:Class>
256. <owl:Class rdf:ID="byz_music_type_71">
257.     <rdfs:label>Αναστάσιμα </rdfs:label>
258.     <rdfs:comment></rdfs:comment>
259.     <rdfs:subClassOf rdf:resource="#byz_music_type_70"/>
260. </owl:Class>
261. <owl:Class rdf:ID="byz_music_type_72">
262.     <rdfs:label>Στιχηρά Ιδιόμελα Δοξαστικά των Αίνων</rdfs:label>
263.     <rdfs:comment></rdfs:comment>
264.     <rdfs:subClassOf rdf:resource="#byz_music_type_70"/>
265. </owl:Class>
266. <owl:Class rdf:ID="byz_music_type_73">
267.     <rdfs:label>Και Νυν</rdfs:label>
268.     <rdfs:comment></rdfs:comment>
269.     <rdfs:subClassOf rdf:resource="#byz_music_type_72"/>
270. </owl:Class>
271. <owl:Class rdf:ID="byz_music_type_74">

```

```

272.         <rdfs:label>Ιδιόμελα των αποστίχων του Εσπερινού</rdfs:label>
273.         <rdfs:comment></rdfs:comment>
274.         <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
275.     </owl:Class>
276. <owl:Class rdf:ID="byz_music_type_75">
277.     <rdfs:label>Αναστάσιμα</rdfs:label>
278.     <rdfs:comment></rdfs:comment>
279.     <rdfs:subClassOf rdf:resource="#byz_music_type_74"/>
280. </owl:Class>
281. <owl:Class rdf:ID="byz_music_type_76">
282.     <rdfs:label>Ιδιόμελα των αποστίχων του Όρθρου</rdfs:label>
283.     <rdfs:comment></rdfs:comment>
284.     <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
285. </owl:Class>
286. <owl:Class rdf:ID="byz_music_type_77">
287.     <rdfs:label>Ιδιόμελα Δοξαστικά του Εσπερινού</rdfs:label>
288.     <rdfs:comment></rdfs:comment>
289.     <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
290. </owl:Class>
291. <owl:Class rdf:ID="byz_music_type_78">
292.     <rdfs:label>Και Νυν</rdfs:label>
293.     <rdfs:comment></rdfs:comment>
294.     <rdfs:subClassOf rdf:resource="#byz_music_type_77"/>
295. </owl:Class>
296. <owl:Class rdf:ID="byz_music_type_79">
297.     <rdfs:label>Ιδιόμελα Δοξαστικά Εωθινά</rdfs:label>
298.     <rdfs:comment></rdfs:comment>
299.     <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
300. </owl:Class>
301. <owl:Class rdf:ID="byz_music_type_80">
302.     <rdfs:label>Ιδιόμελα Δοξαστικά των αποστίχων του Εσπερινού</rdfs:label>
303.     <rdfs:comment></rdfs:comment>
304.     <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
305. </owl:Class>
306. <owl:Class rdf:ID="byz_music_type_81">
307.     <rdfs:label>Και Νυν</rdfs:label>
308.     <rdfs:comment></rdfs:comment>
309.     <rdfs:subClassOf rdf:resource="#byz_music_type_80"/>
310. </owl:Class>
311. <owl:Class rdf:ID="byz_music_type_82">
312.     <rdfs:label>Ιδιόμελα Δοξαστικά των αποστίχων του Όρθρου</rdfs:label>
313.     <rdfs:comment></rdfs:comment>
314.     <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
315. </owl:Class>
316. <owl:Class rdf:ID="byz_music_type_83">
317.     <rdfs:label>Και Νυν</rdfs:label>

```

```

318.         <rdfs:comment></rdfs:comment>
319.         <rdfs:subClassOf rdf:resource="#byz_music_type_82"/>
320.     </owl:Class>
321.     <owl:Class rdf:ID="byz_music_type_84">
322.         <rdfs:label>Ιδιόμελο Ν Ψάλμου</rdfs:label>
323.         <rdfs:comment></rdfs:comment>
324.         <rdfs:subClassOf rdf:resource="#byz_music_type_67"/>
325.     </owl:Class>
326.     <owl:Class rdf:ID="byz_music_type_85">
327.         <rdfs:label>Πολυέλεοι </rdfs:label>
328.         <rdfs:comment></rdfs:comment>
329.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
330.     </owl:Class>
331.     <owl:Class rdf:ID="byz_music_type_86">
332.         <rdfs:label>Άλλα Μέλη</rdfs:label>
333.         <rdfs:comment></rdfs:comment>
334.         <rdfs:subClassOf rdf:resource="#byz_music_type_19"/>
335.     </owl:Class>
336.     <owl:Class rdf:ID="byz_music_type_20">
337.         <rdfs:label>Στιχηραρικά Μέλη</rdfs:label>
338.         <rdfs:comment></rdfs:comment>
339.         <rdfs:subClassOf rdf:resource="#byz_music_type_18"/>
340.     </owl:Class>
341.     <owl:Class rdf:ID="byz_music_type_87">
342.         <rdfs:label>Καθίσματα</rdfs:label>
343.         <rdfs:comment></rdfs:comment>
344.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
345.     </owl:Class>
346.     <owl:Class rdf:ID="byz_music_type_88">
347.         <rdfs:label>Μεσώδια Καθίσματα</rdfs:label>
348.         <rdfs:comment></rdfs:comment>
349.         <rdfs:subClassOf rdf:resource="#byz_music_type_87"/>
350.     </owl:Class>
351.     <owl:Class rdf:ID="byz_music_type_89">
352.         <rdfs:label>Καθίσματα Όρθρου</rdfs:label>
353.         <rdfs:comment></rdfs:comment>
354.         <rdfs:subClassOf rdf:resource="#byz_music_type_87"/>
355.     </owl:Class>
356.     <owl:Class rdf:ID="byz_music_type_90">
357.         <rdfs:label>Αναστάσιμα Καθίσματα</rdfs:label>
358.         <rdfs:comment></rdfs:comment>
359.         <rdfs:subClassOf rdf:resource="#byz_music_type_87"/>
360.     </owl:Class>
361.     <owl:Class rdf:ID="byz_music_type_91">
362.         <rdfs:label>Έξαποστειλάρια </rdfs:label>
363.         <rdfs:comment></rdfs:comment>

```

```
364.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
365.     </owl:Class>
366.     <owl:Class rdf:ID="byz_music_type_92">
367.         <rdfs:label>Αντίφωνα</rdfs:label>
368.         <rdfs:comment></rdfs:comment>
369.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
370.     </owl:Class>
371.     <owl:Class rdf:ID="byz_music_type_93">
372.         <rdfs:label>Εισοδικά</rdfs:label>
373.         <rdfs:comment></rdfs:comment>
374.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
375.     </owl:Class>
376.     <owl:Class rdf:ID="byz_music_type_94">
377.         <rdfs:label>Πολυέλεοι </rdfs:label>
378.         <rdfs:comment></rdfs:comment>
379.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
380.     </owl:Class>
381.     <owl:Class rdf:ID="byz_music_type_95">
382.         <rdfs:label>Αργά Λειτουργικά</rdfs:label>
383.         <rdfs:comment></rdfs:comment>
384.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
385.     </owl:Class>
386.     <owl:Class rdf:ID="byz_music_type_96">
387.         <rdfs:label>Κεκραγάρια</rdfs:label>
388.         <rdfs:comment></rdfs:comment>
389.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
390.     </owl:Class>
391.     <owl:Class rdf:ID="byz_music_type_97">
392.         <rdfs:label>Πασαπνοάρια</rdfs:label>
393.         <rdfs:comment></rdfs:comment>
394.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
395.     </owl:Class>
396.     <owl:Class rdf:ID="byz_music_type_98">
397.         <rdfs:label>Ανοιξαντάρια</rdfs:label>
398.         <rdfs:comment></rdfs:comment>
399.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
400.     </owl:Class>
401.     <owl:Class rdf:ID="byz_music_type_99">
402.         <rdfs:label>Αργές Δοξολογίες</rdfs:label>
403.         <rdfs:comment></rdfs:comment>
404.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
405.     </owl:Class>
406.     <owl:Class rdf:ID="byz_music_type_100">
407.         <rdfs:label>Ιδιόμελα</rdfs:label>
408.         <rdfs:comment></rdfs:comment>
409.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
```

```

410. </owl:Class>
411. <owl:Class rdf:ID="byz_music_type_101">
412.     <rdfs:label>Ιδιόμελα των Ωρών</rdfs:label>
413.     <rdfs:comment></rdfs:comment>
414.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
415. </owl:Class>
416. <owl:Class rdf:ID="byz_music_type_102">
417.     <rdfs:label>Ιδιόμελα Δοξαστικά των Ωρών</rdfs:label>
418.     <rdfs:comment></rdfs:comment>
419.     <rdfs:subClassOf rdf:resource="#byz_music_type_101"/>
420. </owl:Class>
421. <owl:Class rdf:ID="byz_music_type_103">
422.     <rdfs:label>Και Νυν</rdfs:label>
423.     <rdfs:comment></rdfs:comment>
424.     <rdfs:subClassOf rdf:resource="#byz_music_type_102"/>
425. </owl:Class>
426. <owl:Class rdf:ID="byz_music_type_104">
427.     <rdfs:label>Στιχηρά Ιδιόμελα του Εσπερινού</rdfs:label>
428.     <rdfs:comment></rdfs:comment>
429.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
430. </owl:Class>
431. <owl:Class rdf:ID="byz_music_type_105">
432.     <rdfs:label>Αναστάσιμα</rdfs:label>
433.     <rdfs:comment></rdfs:comment>
434.     <rdfs:subClassOf rdf:resource="#byz_music_type_104"/>
435. </owl:Class>
436. <owl:Class rdf:ID="byz_music_type_106">
437.     <rdfs:label>Στιχηρά Ιδιόμελα των Αίνων</rdfs:label>
438.     <rdfs:comment></rdfs:comment>
439.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
440. </owl:Class>
441. <owl:Class rdf:ID="byz_music_type_107">
442.     <rdfs:label>Αναστάσιμα </rdfs:label>
443.     <rdfs:comment></rdfs:comment>
444.     <rdfs:subClassOf rdf:resource="#byz_music_type_106"/>
445. </owl:Class>
446. <owl:Class rdf:ID="byz_music_type_108">
447.     <rdfs:label>Στιχηρά Ιδιόμελα Δοξαστικά των Αίνων</rdfs:label>
448.     <rdfs:comment></rdfs:comment>
449.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
450. </owl:Class>
451. <owl:Class rdf:ID="byz_music_type_109">
452.     <rdfs:label>Και Νυν</rdfs:label>
453.     <rdfs:comment></rdfs:comment>
454.     <rdfs:subClassOf rdf:resource="#byz_music_type_108"/>
455. </owl:Class>

```

```
456. <owl:Class rdf:ID="byz_music_type_110">
457.     <rdfs:label>Ιδιόμελα των αποστίχων του Εσπερινού</rdfs:label>
458.     <rdfs:comment></rdfs:comment>
459.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
460. </owl:Class>
461. <owl:Class rdf:ID="byz_music_type_111">
462.     <rdfs:label>Αναστάσιμα</rdfs:label>
463.     <rdfs:comment></rdfs:comment>
464.     <rdfs:subClassOf rdf:resource="#byz_music_type_110"/>
465. </owl:Class>
466. <owl:Class rdf:ID="byz_music_type_112">
467.     <rdfs:label>Ιδιόμελα των αποστίχων του Όρθρου</rdfs:label>
468.     <rdfs:comment></rdfs:comment>
469.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
470. </owl:Class>
471. <owl:Class rdf:ID="byz_music_type_113">
472.     <rdfs:label>Ιδιόμελα Δοξαστικά του Εσπερινού</rdfs:label>
473.     <rdfs:comment></rdfs:comment>
474.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
475. </owl:Class>
476. <owl:Class rdf:ID="byz_music_type_114">
477.     <rdfs:label>Και Νυν</rdfs:label>
478.     <rdfs:comment></rdfs:comment>
479.     <rdfs:subClassOf rdf:resource="#byz_music_type_113"/>
480. </owl:Class>
481. <owl:Class rdf:ID="byz_music_type_115">
482.     <rdfs:label>Ιδιόμελα Δοξαστικά Εωθινά</rdfs:label>
483.     <rdfs:comment></rdfs:comment>
484.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
485. </owl:Class>
486. <owl:Class rdf:ID="byz_music_type_116">
487.     <rdfs:label>Ιδιόμελα Δοξαστικά των αποστίχων του Εσπερινού</rdfs:label>
488.     <rdfs:comment></rdfs:comment>
489.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
490. </owl:Class>
491. <owl:Class rdf:ID="byz_music_type_117">
492.     <rdfs:label>Και Νυν</rdfs:label>
493.     <rdfs:comment></rdfs:comment>
494.     <rdfs:subClassOf rdf:resource="#byz_music_type_116"/>
495. </owl:Class>
496. <owl:Class rdf:ID="byz_music_type_118">
497.     <rdfs:label>Ιδιόμελα Δοξαστικά των αποστίχων του Όρθρου</rdfs:label>
498.     <rdfs:comment></rdfs:comment>
499.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
500. </owl:Class>
501. <owl:Class rdf:ID="byz_music_type_119">
```

```

502.         <rdfs:label>Και Νυν</rdfs:label>
503.         <rdfs:comment></rdfs:comment>
504.         <rdfs:subClassOf rdf:resource="#byz_music_type_118"/>
505.     </owl:Class>
506. <owl:Class rdf:ID="byz_music_type_120">
507.     <rdfs:label>Ιδιόμελο Ν Ψαλμού</rdfs:label>
508.     <rdfs:comment></rdfs:comment>
509.     <rdfs:subClassOf rdf:resource="#byz_music_type_100"/>
510. </owl:Class>
511. <owl:Class rdf:ID="byz_music_type_121">
512.     <rdfs:label>Αυτόμελο</rdfs:label>
513.     <rdfs:comment></rdfs:comment>
514.     <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
515. </owl:Class>
516. <owl:Class rdf:ID="byz_music_type_122">
517.     <rdfs:label>Προσόμοια</rdfs:label>
518.     <rdfs:comment></rdfs:comment>
519.     <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
520. </owl:Class>
521. <owl:Class rdf:ID="byz_music_type_123">
522.     <rdfs:label>Στιχηρά Προσόμοια του Εσπερινού</rdfs:label>
523.     <rdfs:comment></rdfs:comment>
524.     <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
525. </owl:Class>
526. <owl:Class rdf:ID="byz_music_type_124">
527.     <rdfs:label>Στιχηρά Προσόμοια των Αίνων</rdfs:label>
528.     <rdfs:comment></rdfs:comment>
529.     <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
530. </owl:Class>
531. <owl:Class rdf:ID="byz_music_type_125">
532.     <rdfs:label>Στιχηρά Προσόμοια Δοξαστικά των Αίνων</rdfs:label>
533.     <rdfs:comment></rdfs:comment>
534.     <rdfs:subClassOf rdf:resource="#byz_music_type_124"/>
535. </owl:Class>
536. <owl:Class rdf:ID="byz_music_type_126">
537.     <rdfs:label>Και Νυν</rdfs:label>
538.     <rdfs:comment></rdfs:comment>
539.     <rdfs:subClassOf rdf:resource="#byz_music_type_125"/>
540. </owl:Class>
541. <owl:Class rdf:ID="byz_music_type_127">
542.     <rdfs:label>Προσόμοια των αποστίχων του Εσπερινού</rdfs:label>
543.     <rdfs:comment></rdfs:comment>
544.     <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
545. </owl:Class>
546. <owl:Class rdf:ID="byz_music_type_128">
547.     <rdfs:label>Προσόμοια των αποστίχων του Όρθρου</rdfs:label>

```



```
548.         <rdfs:comment></rdfs:comment>
549.         <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
550.     </owl:Class>
551.     <owl:Class rdf:ID="byz_music_type_129">
552.         <rdfs:label>Προσόμοια Δοξαστικά του Εσπερινού</rdfs:label>
553.         <rdfs:comment></rdfs:comment>
554.         <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
555.     </owl:Class>
556.     <owl:Class rdf:ID="byz_music_type_130">
557.         <rdfs:label>Και Νυν</rdfs:label>
558.         <rdfs:comment></rdfs:comment>
559.         <rdfs:subClassOf rdf:resource="#byz_music_type_129"/>
560.     </owl:Class>
561.     <owl:Class rdf:ID="byz_music_type_131">
562.         <rdfs:label>Προσόμοια Δοξαστικά Εωθινά</rdfs:label>
563.         <rdfs:comment></rdfs:comment>
564.         <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
565.     </owl:Class>
566.     <owl:Class rdf:ID="byz_music_type_132">
567.         <rdfs:label>Προσόμοια Δοξαστικά των αποστίχων του Εσπερινού</rdfs:label>
568.         <rdfs:comment></rdfs:comment>
569.         <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
570.     </owl:Class>
571.     <owl:Class rdf:ID="byz_music_type_133">
572.         <rdfs:label>Και Νυν</rdfs:label>
573.         <rdfs:comment></rdfs:comment>
574.         <rdfs:subClassOf rdf:resource="#byz_music_type_132"/>
575.     </owl:Class>
576.     <owl:Class rdf:ID="byz_music_type_134">
577.         <rdfs:label>Προσόμοια Δοξαστικά των αποστίχων του Όρθρου</rdfs:label>
578.         <rdfs:comment></rdfs:comment>
579.         <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
580.     </owl:Class>
581.     <owl:Class rdf:ID="byz_music_type_135">
582.         <rdfs:label>Και Νυν</rdfs:label>
583.         <rdfs:comment></rdfs:comment>
584.         <rdfs:subClassOf rdf:resource="#byz_music_type_134"/>
585.     </owl:Class>
586.     <owl:Class rdf:ID="byz_music_type_136">
587.         <rdfs:label>Προσόμοια Ν Ψαλμού</rdfs:label>
588.         <rdfs:comment></rdfs:comment>
589.         <rdfs:subClassOf rdf:resource="#byz_music_type_122"/>
590.     </owl:Class>
591.     <owl:Class rdf:ID="byz_music_type_137">
592.         <rdfs:label>Ευλογητάρια</rdfs:label>
593.         <rdfs:comment></rdfs:comment>
```

```

594.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
595.     </owl:Class>
596.     <owl:Class rdf:ID="byz_music_type_138">
597.         <rdfs:label>Άλλα Μέλη</rdfs:label>
598.         <rdfs:comment></rdfs:comment>
599.         <rdfs:subClassOf rdf:resource="#byz_music_type_20"/>
600.     </owl:Class>
601.     <owl:Class rdf:ID="byz_music_type_21">
602.         <rdfs:label>Παπαδικά Μέλη</rdfs:label>
603.         <rdfs:comment></rdfs:comment>
604.         <rdfs:subClassOf rdf:resource="#byz_music_type_18"/>
605.     </owl:Class>
606.     <owl:Class rdf:ID="byz_music_type_139">
607.         <rdfs:label>Μαθήματα</rdfs:label>
608.         <rdfs:comment></rdfs:comment>
609.         <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
610.     </owl:Class>
611.     <owl:Class rdf:ID="byz_music_type_140">
612.         <rdfs:label>Χερουβικά </rdfs:label>
613.         <rdfs:comment></rdfs:comment>
614.         <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
615.     </owl:Class>
616.     <owl:Class rdf:ID="byz_music_type_141">
617.         <rdfs:label>Κοινωνικά</rdfs:label>
618.         <rdfs:comment></rdfs:comment>
619.         <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
620.     </owl:Class>
621.     <owl:Class rdf:ID="byz_music_type_142">
622.         <rdfs:label>Αργά Τρισάγια</rdfs:label>
623.         <rdfs:comment></rdfs:comment>
624.         <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
625.     </owl:Class>
626.     <owl:Class rdf:ID="byz_music_type_143">
627.         <rdfs:label>Αργά Αλληλουάρια</rdfs:label>
628.         <rdfs:comment></rdfs:comment>
629.         <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
630.     </owl:Class>
631.     <owl:Class rdf:ID="byz_music_type_144">
632.         <rdfs:label>Ασματικά</rdfs:label>
633.         <rdfs:comment></rdfs:comment>
634.         <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
635.     </owl:Class>
636.     <owl:Class rdf:ID="byz_music_type_145">
637.         <rdfs:label>Θεοτοκία</rdfs:label>
638.         <rdfs:comment></rdfs:comment>
639.         <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>

```

```

640. </owl:Class>
641. <owl:Class rdf:ID="byz_music_type_146">
642.     <rdfs:label>Αργά Κεκραγάρια</rdfs:label>
643.     <rdfs:comment></rdfs:comment>
644.     <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
645. </owl:Class>
646. <owl:Class rdf:ID="byz_music_type_147">
647.     <rdfs:label>Αργά Πασαπινοάρια</rdfs:label>
648.     <rdfs:comment></rdfs:comment>
649.     <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
650. </owl:Class>
651. <owl:Class rdf:ID="byz_music_type_148">
652.     <rdfs:label>Κρατήματα</rdfs:label>
653.     <rdfs:comment></rdfs:comment>
654.     <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
655. </owl:Class>
656. <owl:Class rdf:ID="byz_music_type_149">
657.     <rdfs:label>Δοξαστικά Πολυελαίων</rdfs:label>
658.     <rdfs:comment></rdfs:comment>
659.     <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
660. </owl:Class>
661. <owl:Class rdf:ID="byz_music_type_150">
662.     <rdfs:label>Αργά Προκείμενα της Μεγάλης Τεσσαρακοστής</rdfs:label>
663.     <rdfs:comment></rdfs:comment>
664.     <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
665. </owl:Class>
666. <owl:Class rdf:ID="byz_music_type_151">
667.     <rdfs:label>Άλλα Μέλη</rdfs:label>
668.     <rdfs:comment></rdfs:comment>
669.     <rdfs:subClassOf rdf:resource="#byz_music_type_21"/>
670. </owl:Class>
671. <owl:Class rdf:ID="byz_music_type_152">
672.     <rdfs:label>Εμμελής Απαγγελία</rdfs:label>
673.     <rdfs:comment></rdfs:comment>
674.     <rdfs:subClassOf rdf:resource="#byz_music_type_18"/>
675. </owl:Class>

```

### Makams rdf-schema extension

```

1. <owl:Class rdf:ID="makam_1">
2.     <rdfs:label>Βυζαντινοί Ήχοι</rdfs:label>
3.     <rdfs:comment></rdfs:comment>
4.     <rdfs:subClassOf rdf:resource="#makam_0"/>
5. </owl:Class>
6. <owl:Class rdf:ID="makam_2">
7.     <rdfs:label>Διατονικό Γένος</rdfs:label>

```

```

8.     <rdfs:comment></rdfs:comment>
9.     <rdfs:subClassOf rdf:resource="#makam_1"/>
10.  </owl:Class>
11.  <owl:Class rdf:ID="makam_5">
12.     <rdfs:label>Πρώτος Ήχος</rdfs:label>
13.     <rdfs:comment></rdfs:comment>
14.     <rdfs:subClassOf rdf:resource="#makam_2"/>
15.  </owl:Class>
16.  <owl:Class rdf:ID="makam_13">
17.     <rdfs:label>Ειρμολογικός Πρώτος</rdfs:label>
18.     <rdfs:comment></rdfs:comment>
19.     <rdfs:subClassOf rdf:resource="#makam_5"/>
20.  </owl:Class>
21.  <owl:Class rdf:ID="makam_14">
22.     <rdfs:label>Στιχηραρικός Πρώτος</rdfs:label>
23.     <rdfs:comment></rdfs:comment>
24.     <rdfs:subClassOf rdf:resource="#makam_5"/>
25.  </owl:Class>
26.  <owl:Class rdf:ID="makam_15">
27.     <rdfs:label>Παπαδικός Πρώτος</rdfs:label>
28.     <rdfs:comment></rdfs:comment>
29.     <rdfs:subClassOf rdf:resource="#makam_5"/>
30.  </owl:Class>
31.  <owl:Class rdf:ID="makam_16">
32.     <rdfs:label>Παθητικός Πρώτος (νάος)</rdfs:label>
33.     <rdfs:comment></rdfs:comment>
34.     <rdfs:subClassOf rdf:resource="#makam_5"/>
35.  </owl:Class>
36.  <owl:Class rdf:ID="makam_17">
37.     <rdfs:label>Πρώτος εκ του Κε</rdfs:label>
38.     <rdfs:comment></rdfs:comment>
39.     <rdfs:subClassOf rdf:resource="#makam_5"/>
40.  </owl:Class>
41.  <owl:Class rdf:ID="makam_18">
42.     <rdfs:label>Χρωματικός Πρώτος</rdfs:label>
43.     <rdfs:comment></rdfs:comment>
44.     <rdfs:subClassOf rdf:resource="#makam_5"/>
45.  </owl:Class>
46.  <owl:Class rdf:ID="makam_6">
47.     <rdfs:label>Τέταρτος Ήχος</rdfs:label>
48.     <rdfs:comment></rdfs:comment>
49.     <rdfs:subClassOf rdf:resource="#makam_2"/>
50.  </owl:Class>
51.  <owl:Class rdf:ID="makam_25">
52.     <rdfs:label>Λέγετος</rdfs:label>
53.     <rdfs:comment></rdfs:comment>

```

```
54. <rdfs:subClassOf rdf:resource="#makam_6"/>
55. </owl:Class>
56. <owl:Class rdf:ID="makam_26">
57. <rdfs:label>Άγια</rdfs:label>
58. <rdfs:comment></rdfs:comment>
59. <rdfs:subClassOf rdf:resource="#makam_6"/>
60. </owl:Class>
61. <owl:Class rdf:ID="makam_27">
62. <rdfs:label>Στιχηραρικός Τέταρτος</rdfs:label>
63. <rdfs:comment></rdfs:comment>
64. <rdfs:subClassOf rdf:resource="#makam_6"/>
65. </owl:Class>
66. <owl:Class rdf:ID="makam_28">
67. <rdfs:label>Κλίτιον Μέλος</rdfs:label>
68. <rdfs:comment></rdfs:comment>
69. <rdfs:subClassOf rdf:resource="#makam_6"/>
70. </owl:Class>
71. <owl:Class rdf:ID="makam_29">
72. <rdfs:label>Χρωματικός Τέταρτος</rdfs:label>
73. <rdfs:comment></rdfs:comment>
74. <rdfs:subClassOf rdf:resource="#makam_6"/>
75. </owl:Class>
76. <owl:Class rdf:ID="makam_7">
77. <rdfs:label>Ήχος Πλάγιος του Πρώτου</rdfs:label>
78. <rdfs:comment></rdfs:comment>
79. <rdfs:subClassOf rdf:resource="#makam_2"/>
80. </owl:Class>
81. <owl:Class rdf:ID="makam_30">
82. <rdfs:label>Στιχηραρικός Πλάγιος του Πρώτου (εκ του Πα)</rdfs:label>
83. <rdfs:comment></rdfs:comment>
84. <rdfs:subClassOf rdf:resource="#makam_7"/>
85. </owl:Class>
86. <owl:Class rdf:ID="makam_31">
87. <rdfs:label>Ειρμολογικός Πλάγιος του Πρώτου (εκ του Κε)</rdfs:label>
88. <rdfs:comment></rdfs:comment>
89. <rdfs:subClassOf rdf:resource="#makam_7"/>
90. </owl:Class>
91. <owl:Class rdf:ID="makam_32">
92. <rdfs:label>Πλάγιος του Πρώτου με χρώα σπάθη</rdfs:label>
93. <rdfs:comment></rdfs:comment>
94. <rdfs:subClassOf rdf:resource="#makam_7"/>
95. </owl:Class>
96. <owl:Class rdf:ID="makam_33">
97. <rdfs:label>Εναρμόνιος Πλάγιος του Πρώτου</rdfs:label>
98. <rdfs:comment></rdfs:comment>
99. <rdfs:subClassOf rdf:resource="#makam_7"/>
```

```

100. </owl:Class>
101. <owl:Class rdf:ID="makam_34">
102.     <rdfs:label>Πεντάφωνος Πλάγιος του Πρώτου</rdfs:label>
103.     <rdfs:comment></rdfs:comment>
104.     <rdfs:subClassOf rdf:resource="#makam_7"/>
105. </owl:Class>
106. <owl:Class rdf:ID="makam_35">
107.     <rdfs:label>Εναρμόνιος Πλάγιος του Πρώτου με Βου ύφεση (Φρύγιος)</rdfs:label>
108.     <rdfs:comment></rdfs:comment>
109.     <rdfs:subClassOf rdf:resource="#makam_7"/>
110. </owl:Class>
111. <owl:Class rdf:ID="makam_56">
112.     <rdfs:label>Παπαδικός Πλάγιος του Πρώτου</rdfs:label>
113.     <rdfs:comment></rdfs:comment>
114.     <rdfs:subClassOf rdf:resource="#makam_7"/>
115. </owl:Class>
116. <owl:Class rdf:ID="makam_8">
117.     <rdfs:label>Ήχος Πλάγιος του Τετάρτου</rdfs:label>
118.     <rdfs:comment></rdfs:comment>
119.     <rdfs:subClassOf rdf:resource="#makam_2"/>
120. </owl:Class>
121. <owl:Class rdf:ID="makam_43">
122.     <rdfs:label>Πλάγιος του Τετάρτου εκ του Νη</rdfs:label>
123.     <rdfs:comment></rdfs:comment>
124.     <rdfs:subClassOf rdf:resource="#makam_8"/>
125. </owl:Class>
126. <owl:Class rdf:ID="makam_44">
127.     <rdfs:label>Πλάγιος του Τετάρτου κατά τριφωνία</rdfs:label>
128.     <rdfs:comment></rdfs:comment>
129.     <rdfs:subClassOf rdf:resource="#makam_8"/>
130. </owl:Class>
131. <owl:Class rdf:ID="makam_45">
132.     <rdfs:label>Χρωματικός Πλάγιος του Τετάρτου</rdfs:label>
133.     <rdfs:comment></rdfs:comment>
134.     <rdfs:subClassOf rdf:resource="#makam_8"/>
135. </owl:Class>
136. <owl:Class rdf:ID="makam_46">
137.     <rdfs:label>Πλάγιος του Τετάρτου με φθορά Δευτέρου Ήχου στο Δι</rdfs:label>
138.     <rdfs:comment></rdfs:comment>
139.     <rdfs:subClassOf rdf:resource="#makam_8"/>
140. </owl:Class>
141. <owl:Class rdf:ID="makam_47">
142.     <rdfs:label>Εναρμόνιος Πλάγιος του Τετάρτου</rdfs:label>
143.     <rdfs:comment></rdfs:comment>
144.     <rdfs:subClassOf rdf:resource="#makam_8"/>
145. </owl:Class>

```

```
146. <owl:Class rdf:ID="makam_58">
147.     <rdfs:label>Πλάγιος του Τετάρτου Επτάφωνος</rdfs:label>
148.     <rdfs:comment></rdfs:comment>
149.     <rdfs:subClassOf rdf:resource="#makam_8"/>
150. </owl:Class>
151. <owl:Class rdf:ID="makam_3">
152.     <rdfs:label>Χρωματικό Γένος</rdfs:label>
153.     <rdfs:comment></rdfs:comment>
154.     <rdfs:subClassOf rdf:resource="#makam_1"/>
155. </owl:Class>
156. <owl:Class rdf:ID="makam_9">
157.     <rdfs:label>Δεύτερος Ήχος</rdfs:label>
158.     <rdfs:comment></rdfs:comment>
159.     <rdfs:subClassOf rdf:resource="#makam_3"/>
160. </owl:Class>
161. <owl:Class rdf:ID="makam_19">
162.     <rdfs:label>Δεύτερος εκ του Πα</rdfs:label>
163.     <rdfs:comment></rdfs:comment>
164.     <rdfs:subClassOf rdf:resource="#makam_9"/>
165. </owl:Class>
166. <owl:Class rdf:ID="makam_49">
167.     <rdfs:label>Ειρμολογικός Δεύτερος</rdfs:label>
168.     <rdfs:comment></rdfs:comment>
169.     <rdfs:subClassOf rdf:resource="#makam_19"/>
170. </owl:Class>
171. <owl:Class rdf:ID="makam_50">
172.     <rdfs:label>Παπαδικός Δεύτερος</rdfs:label>
173.     <rdfs:comment></rdfs:comment>
174.     <rdfs:subClassOf rdf:resource="#makam_19"/>
175. </owl:Class>
176. <owl:Class rdf:ID="makam_20">
177.     <rdfs:label>Δεύτερος εκ του Βου</rdfs:label>
178.     <rdfs:comment></rdfs:comment>
179.     <rdfs:subClassOf rdf:resource="#makam_9"/>
180. </owl:Class>
181. <owl:Class rdf:ID="makam_51">
182.     <rdfs:label>Ειρμολογικός Δεύτερος εκ του Βου</rdfs:label>
183.     <rdfs:comment></rdfs:comment>
184.     <rdfs:subClassOf rdf:resource="#makam_20"/>
185. </owl:Class>
186. <owl:Class rdf:ID="makam_52">
187.     <rdfs:label>Παπαδικός Δεύτερος εκ του Βου</rdfs:label>
188.     <rdfs:comment></rdfs:comment>
189.     <rdfs:subClassOf rdf:resource="#makam_20"/>
190. </owl:Class>
191. <owl:Class rdf:ID="makam_21">
```

```

192.         <rdfs:label>Δεύτερος εκ του Δι</rdfs:label>
193.         <rdfs:comment></rdfs:comment>
194.         <rdfs:subClassOf rdf:resource="#makam_9"/>
195.     </owl:Class>
196.     <owl:Class rdf:ID="makam_53">
197.         <rdfs:label>Ειρμολογικός Δεύτερος από το Δι</rdfs:label>
198.         <rdfs:comment></rdfs:comment>
199.         <rdfs:subClassOf rdf:resource="#makam_21"/>
200.     </owl:Class>
201.     <owl:Class rdf:ID="makam_54">
202.         <rdfs:label>Στιχηραρικός Δεύτερος εκ του Δι</rdfs:label>
203.         <rdfs:comment></rdfs:comment>
204.         <rdfs:subClassOf rdf:resource="#makam_21"/>
205.     </owl:Class>
206.     <owl:Class rdf:ID="makam_55">
207.         <rdfs:label>Παπαδικός Δεύτερος εκ του Δι</rdfs:label>
208.         <rdfs:comment></rdfs:comment>
209.         <rdfs:subClassOf rdf:resource="#makam_21"/>
210.     </owl:Class>
211.     <owl:Class rdf:ID="makam_22">
212.         <rdfs:label>Δεύτερος εκ του Δι με καταλήξεις στο Βου</rdfs:label>
213.         <rdfs:comment></rdfs:comment>
214.         <rdfs:subClassOf rdf:resource="#makam_9"/>
215.     </owl:Class>
216.     <owl:Class rdf:ID="makam_23">
217.         <rdfs:label>Διατονικός Δεύτερος</rdfs:label>
218.         <rdfs:comment></rdfs:comment>
219.         <rdfs:subClassOf rdf:resource="#makam_9"/>
220.     </owl:Class>
221.     <owl:Class rdf:ID="makam_24">
222.         <rdfs:label>Δευτερόπρωτος</rdfs:label>
223.         <rdfs:comment></rdfs:comment>
224.         <rdfs:subClassOf rdf:resource="#makam_9"/>
225.     </owl:Class>
226.     <owl:Class rdf:ID="makam_10">
227.         <rdfs:label>Ήχος Πλάγιος του Δεύτερου</rdfs:label>
228.         <rdfs:comment></rdfs:comment>
229.         <rdfs:subClassOf rdf:resource="#makam_3"/>
230.     </owl:Class>
231.     <owl:Class rdf:ID="makam_36">
232.         <rdfs:label>Στιχηραρικός Πλάγιος του Δευτέρου (εκ του Πα)</rdfs:label>
233.         <rdfs:comment></rdfs:comment>
234.         <rdfs:subClassOf rdf:resource="#makam_10"/>
235.     </owl:Class>
236.     <owl:Class rdf:ID="makam_37">
237.         <rdfs:label>Ειρμολογικός Πλάγιος του Δευτέρου (εκ του Δι)</rdfs:label>

```



```
238.         <rdfs:comment></rdfs:comment>
239.         <rdfs:subClassOf rdf:resource="#makam_10"/>
240.     </owl:Class>
241.     <owl:Class rdf:ID="makam_38">
242.         <rdfs:label>Επτάφωνος Πλάγιος του Δευτέρου</rdfs:label>
243.         <rdfs:comment></rdfs:comment>
244.         <rdfs:subClassOf rdf:resource="#makam_10"/>
245.     </owl:Class>
246.     <owl:Class rdf:ID="makam_39">
247.         <rdfs:label>Πλάγιος του Δευτέρου με καταλήξεις Δι (Νενανώ)</rdfs:label>
248.         <rdfs:comment></rdfs:comment>
249.         <rdfs:subClassOf rdf:resource="#makam_10"/>
250.     </owl:Class>
251.     <owl:Class rdf:ID="makam_57">
252.         <rdfs:label>Παπαδικός Πλάγιος του Δευτέρου</rdfs:label>
253.         <rdfs:comment></rdfs:comment>
254.         <rdfs:subClassOf rdf:resource="#makam_10"/>
255.     </owl:Class>
256.     <owl:Class rdf:ID="makam_4">
257.         <rdfs:label>Εναρμόνιο Γένος</rdfs:label>
258.         <rdfs:comment></rdfs:comment>
259.         <rdfs:subClassOf rdf:resource="#makam_1"/>
260.     </owl:Class>
261.     <owl:Class rdf:ID="makam_11">
262.         <rdfs:label>Τρίτος Ήχος</rdfs:label>
263.         <rdfs:comment></rdfs:comment>
264.         <rdfs:subClassOf rdf:resource="#makam_4"/>
265.     </owl:Class>
266.     <owl:Class rdf:ID="makam_12">
267.         <rdfs:label>Βαρύς Ήχος</rdfs:label>
268.         <rdfs:comment></rdfs:comment>
269.         <rdfs:subClassOf rdf:resource="#makam_4"/>
270.     </owl:Class>
271.     <owl:Class rdf:ID="makam_40">
272.         <rdfs:label>Εναρμόνιος Βαρύς</rdfs:label>
273.         <rdfs:comment></rdfs:comment>
274.         <rdfs:subClassOf rdf:resource="#makam_12"/>
275.     </owl:Class>
276.     <owl:Class rdf:ID="makam_41">
277.         <rdfs:label>Διατονικός Βαρύς</rdfs:label>
278.         <rdfs:comment></rdfs:comment>
279.         <rdfs:subClassOf rdf:resource="#makam_12"/>
280.     </owl:Class>
281.     <owl:Class rdf:ID="makam_42">
282.         <rdfs:label>Διατονικός Επτάφωνος Βαρύς</rdfs:label>
283.         <rdfs:comment></rdfs:comment>
```

```

284.         <rdfs:subClassOf rdf:resource="#makam_12"/>
285.     </owl:Class>
286.     <owl:Class rdf:ID="makam_48">
287.         <rdfs:label>Οκτάηχα Μέλη</rdfs:label>
288.         <rdfs:comment></rdfs:comment>
289.         <rdfs:subClassOf rdf:resource="#makam_1"/>
290.     </owl:Class>

```

### Musical instruments rdf-schema extension

```

1. <owl:Class rdf:ID="musical_instrument_cat_1">
2.     <rdfs:label>Παραδοσιακά Μουσικά Όργανα</rdfs:label>
3.     <rdfs:comment></rdfs:comment>
4.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_0"/>
5. </owl:Class>
6. <owl:Class rdf:ID="musical_instrument_cat_2">
7.     <rdfs:label>Ιδιόφωνα</rdfs:label>
8.     <rdfs:comment></rdfs:comment>
9.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_1"/>
10. </owl:Class>
11. <owl:Class rdf:ID="musical_instrument_1">
12.     <rdfs:label>Ζίλια</rdfs:label>
13.     <rdfs:comment></rdfs:comment>
14.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
15. </owl:Class>
16. <owl:Class rdf:ID="musical_instrument_2">
17.     <rdfs:label>Κουδούνι</rdfs:label>
18.     <rdfs:comment></rdfs:comment>
19.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
20. </owl:Class>
21. <owl:Class rdf:ID="musical_instrument_3">
22.     <rdfs:label>Μασά</rdfs:label>
23.     <rdfs:comment></rdfs:comment>
24.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
25. </owl:Class>
26. <owl:Class rdf:ID="musical_instrument_4">
27.     <rdfs:label>Τζαμάλα</rdfs:label>
28.     <rdfs:comment></rdfs:comment>
29.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
30. </owl:Class>
31. <owl:Class rdf:ID="musical_instrument_5">
32.     <rdfs:label>Τρίγωνο</rdfs:label>
33.     <rdfs:comment></rdfs:comment>
34.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
35. </owl:Class>
36. <owl:Class rdf:ID="musical_instrument_6">
37.     <rdfs:label>Παλαμάκια - Στράκες</rdfs:label>

```

```
38. <rdfs:comment></rdfs:comment>
39. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
40. </owl:Class>
41. <owl:Class rdf:ID="musical_instrument_7">
42. <rdfs:label>Νομίσματα</rdfs:label>
43. <rdfs:comment></rdfs:comment>
44. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
45. </owl:Class>
46. <owl:Class rdf:ID="musical_instrument_8">
47. <rdfs:label>Μασούρ-πλεξίδες</rdfs:label>
48. <rdfs:comment></rdfs:comment>
49. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
50. </owl:Class>
51. <owl:Class rdf:ID="musical_instrument_9">
52. <rdfs:label>Σουργούτ</rdfs:label>
53. <rdfs:comment></rdfs:comment>
54. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
55. </owl:Class>
56. <owl:Class rdf:ID="musical_instrument_10">
57. <rdfs:label>Κουτάλια</rdfs:label>
58. <rdfs:comment></rdfs:comment>
59. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
60. </owl:Class>
61. <owl:Class rdf:ID="musical_instrument_11">
62. <rdfs:label>Μαγειρικά σκέυη</rdfs:label>
63. <rdfs:comment></rdfs:comment>
64. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
65. </owl:Class>
66. <owl:Class rdf:ID="musical_instrument_12">
67. <rdfs:label>Ποτηράκια</rdfs:label>
68. <rdfs:comment></rdfs:comment>
69. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
70. </owl:Class>
71. <owl:Class rdf:ID="musical_instrument_13">
72. <rdfs:label>Κομπολόι</rdfs:label>
73. <rdfs:comment></rdfs:comment>
74. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_2"/>
75. </owl:Class>
76. <owl:Class rdf:ID="musical_instrument_cat_3">
77. <rdfs:label>Μεμβρανόφωνα</rdfs:label>
78. <rdfs:comment></rdfs:comment>
79. <rdfs:subClassOf rdf:resource="#musical_instrument_cat_1"/>
80. </owl:Class>
81. <owl:Class rdf:ID="musical_instrument_14">
82. <rdfs:label>Νταούλι</rdfs:label>
83. <rdfs:comment></rdfs:comment>
```

```

84.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_3"/>
85. </owl:Class>
86. <owl:Class rdf:ID="musical_instrument_15">
87.   <rdfs:label>Τουμπί</rdfs:label>
88.   <rdfs:comment></rdfs:comment>
89.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_3"/>
90. </owl:Class>
91. <owl:Class rdf:ID="musical_instrument_16">
92.   <rdfs:label>Ντέφι</rdfs:label>
93.   <rdfs:comment></rdfs:comment>
94.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_3"/>
95. </owl:Class>
96. <owl:Class rdf:ID="musical_instrument_17">
97.   <rdfs:label>Ταμπουτσά</rdfs:label>
98.   <rdfs:comment></rdfs:comment>
99.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_3"/>
100. </owl:Class>
101. <owl:Class rdf:ID="musical_instrument_18">
102.   <rdfs:label>Τουμπελέκι</rdfs:label>
103.   <rdfs:comment></rdfs:comment>
104.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_3"/>
105. </owl:Class>
106. <owl:Class rdf:ID="musical_instrument_cat_4">
107.   <rdfs:label>Αερόφωνα</rdfs:label>
108.   <rdfs:comment></rdfs:comment>
109.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_1"/>
110. </owl:Class>
111. <owl:Class rdf:ID="musical_instrument_19">
112.   <rdfs:label>Φλογέρα</rdfs:label>
113.   <rdfs:comment></rdfs:comment>
114.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
115. </owl:Class>
116. <owl:Class rdf:ID="musical_instrument_20">
117.   <rdfs:label>Ζουρνάς</rdfs:label>
118.   <rdfs:comment></rdfs:comment>
119.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
120. </owl:Class>
121. <owl:Class rdf:ID="musical_instrument_21">
122.   <rdfs:label>Τσαμπούνα</rdfs:label>
123.   <rdfs:comment></rdfs:comment>
124.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
125. </owl:Class>
126. <owl:Class rdf:ID="musical_instrument_22">
127.   <rdfs:label>Γκάνιτα</rdfs:label>
128.   <rdfs:comment></rdfs:comment>
129.   <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>

```

```
130. </owl:Class>
131. <owl:Class rdf:ID="musical_instrument_23">
132.     <rdfs:label>Κλαρίνο</rdfs:label>
133.     <rdfs:comment></rdfs:comment>
134.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
135. </owl:Class>
136. <owl:Class rdf:ID="musical_instrument_24">
137.     <rdfs:label>Φύλλο του δέντρου</rdfs:label>
138.     <rdfs:comment></rdfs:comment>
139.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_4"/>
140. </owl:Class>
141. <owl:Class rdf:ID="musical_instrument_cat_5">
142.     <rdfs:label>Χορδόφωνα</rdfs:label>
143.     <rdfs:comment></rdfs:comment>
144.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_1"/>
145. </owl:Class>
146. <owl:Class rdf:ID="musical_instrument_25">
147.     <rdfs:label>Ταμπουράς</rdfs:label>
148.     <rdfs:comment></rdfs:comment>
149.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
150. </owl:Class>
151. <owl:Class rdf:ID="musical_instrument_26">
152.     <rdfs:label>Λαγούτο</rdfs:label>
153.     <rdfs:comment></rdfs:comment>
154.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
155. </owl:Class>
156. <owl:Class rdf:ID="musical_instrument_27">
157.     <rdfs:label>Ούτι</rdfs:label>
158.     <rdfs:comment></rdfs:comment>
159.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
160. </owl:Class>
161. <owl:Class rdf:ID="musical_instrument_28">
162.     <rdfs:label>Λαουτοκιθάρα</rdfs:label>
163.     <rdfs:comment></rdfs:comment>
164.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
165. </owl:Class>
166. <owl:Class rdf:ID="musical_instrument_29">
167.     <rdfs:label>Μαντολίνο & κιθάρα</rdfs:label>
168.     <rdfs:comment></rdfs:comment>
169.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
170. </owl:Class>
171. <owl:Class rdf:ID="musical_instrument_30">
172.     <rdfs:label>Λύρα</rdfs:label>
173.     <rdfs:comment></rdfs:comment>
174.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
175. </owl:Class>
```

```

176. <owl:Class rdf:ID="musical_instrument_31">
177.     <rdfs:label>Κεμεντζές</rdfs:label>
178.     <rdfs:comment></rdfs:comment>
179.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
180. </owl:Class>
181. <owl:Class rdf:ID="musical_instrument_32">
182.     <rdfs:label>Κεμανές</rdfs:label>
183.     <rdfs:comment></rdfs:comment>
184.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
185. </owl:Class>
186. <owl:Class rdf:ID="musical_instrument_33">
187.     <rdfs:label>Βιολί</rdfs:label>
188.     <rdfs:comment></rdfs:comment>
189.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
190. </owl:Class>
191. <owl:Class rdf:ID="musical_instrument_34">
192.     <rdfs:label>Κανονάκι</rdfs:label>
193.     <rdfs:comment></rdfs:comment>
194.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
195. </owl:Class>
196. <owl:Class rdf:ID="musical_instrument_35">
197.     <rdfs:label>Σαντούρι</rdfs:label>
198.     <rdfs:comment></rdfs:comment>
199.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
200. </owl:Class>
201. <owl:Class rdf:ID="musical_instrument_36">
202.     <rdfs:label>Λατέρνα</rdfs:label>
203.     <rdfs:comment></rdfs:comment>
204.     <rdfs:subClassOf rdf:resource="#musical_instrument_cat_5"/>
205. </owl:Class>

```

## Master.java

```

package master.endpoint;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

import flanagan.analysis.PCA;
import flanagan.math.Matrix;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

@WebService()
public class Master {

    private static Connection sqlConnection;
    private static ResultSet result;

```

```

private static ResultSet result2;
private static double[] propabilities;
private static double[] distribution;
private static double sigma = 20;
private static String[] users;
private static int[] items;

@WebMethod(operationName = "master")
public static String[] master_alg(@WebParam(name = "active_user") String
active_user)
{
    double p = 0.2;

    int numberOfItems = 0;
    int numberOfUsers = 0;

    //Connect to the database
    try {
        Class.forName("org.gjt.mm.mysql.Driver");
    } catch (ClassNotFoundException e1) {}

    String url = "";
    try {
        url = "jdbc:mysql://domainname/database";
        sqlConnection = DriverManager.getConnection(url, "username",
"password");
    } catch (java.sql.SQLException e) {}

    //Select all items
    try {
        Statement select = sqlConnection.createStatement();
        result = select.executeQuery("SELECT id FROM music_files order
by id");
        result.last();
        int numberOfRows = numberOfItems = result.getRow();
        items = new int[numberOfRows];
        int i = 0;
        result.beforeFirst();
        while (result.next()) { //
            int key = result.getInt(1);
            items[i++] = key;
        }
    } catch (Exception e) {}

    //Select all users
    try {
        Statement select2 = sqlConnection.createStatement();
        result2 = select2.executeQuery("SELECT id, username FROM users
order by id");
        result2.last();
        int numberOfRows2 = numberOfUsers = result2.getRow();
        users = new String[numberOfRows2];

        int j = 0;
        result2.beforeFirst();
        while (result2.next()) {
            String val = result2.getString(2);
            users[j++] = val;
        }
    } catch (Exception e) {}

    //Create the array of users to items ratings

```

```

double[][] array = new double[numberOfUsers][numberOfItems];

try {
    result2.beforeFirst();
    int i = 0;
    int j = 0;

    while (result2.next()) {
        j = 0;
        result.beforeFirst();
        while (result.next()) {
            Statement select3 =
                sqlConnection.createStatement();
            ResultSet result3 = select3.executeQuery("SELECT
rating_value FROM itemratings where rater_id =
 '"+result2.getString(2)+"' and
 item_id='"+result.getInt(1)+"'");
            if(result3.next())
                array[i][j] = (double) result3.getInt(1);
            j++;
        }
        i++;
    }

} catch (SQLException e) {}

PCA pcal = new PCA();
pcal.useCovarianceMatrix();
pcal.enterScoresAsRowPerPerson(array);
Matrix cm = pcal.covarianceMatrix();
double[][] eigenv = pcal.orderedEigenvectors();
Matrix evlc = new Matrix(eigenv);

/**** TAKE MOST SIGNIFICANT EIGENVALUES - EIGENVECTORS *****/

int cols = (int) (cm.getNcol()*p);
int actrows = cm.getNrow();

Matrix mseigvec = new Matrix(evlc.getSubMatrix(0, 0, actrows-1,
cols-1));

Matrix user = new Matrix(array);

Matrix user_mapping = user.times(mseigvec);
double[][] user_mappingd = user_mapping.getArrayCopy();

/***** PREDICTION STEP *****/

int au_pos = 0;

for(; au_pos < users.length; au_pos++)
    if(users[au_pos].equals(active_user))
        break;

try {
    Statement select2 = sqlConnection.createStatement();
    ResultSet result4 = select2.executeQuery("SELECT id, username
FROM users where username != '"+active_user+"' order by id");
    result4.last();
    int usersno = result4.getRow();

    propabilities = new double[usersno+1];

```



```

int i = 0;

//initialization
propabilities[au_pos]=1.0/((double) (usersno+1));

result4.beforeFirst();

//Find other users similarity propability
while (result4.next()) {
    String val = result4.getString(2);
    int u_pos = 0;

    for(; u_pos < users.length; u_pos++)
        if(users[u_pos].equals(val))
            break;

    propabilities[u_pos] = 1.0/((double) (usersno+1));
    for(int k = 0; k < user_mappingd[u_pos].length; k++)
    {
        double arv = user_mappingd[au_pos][k];
        double rv = user_mappingd[u_pos][k];
        propabilities[u_pos] *= Math.exp(-Math.pow(arv-rv,
            2)/(2*Math.pow(sigma, 2)));
    }
    i++;
}
catch (Exception e) {}

//Create the distribution and predict the values of the unrated
items

distribution = new double[items.length];
double[] predictions = new double[items.length];

double max;

for (int j=0; j < items.length; ++j)
{
    max = 0.0;
    //Only if it is unobserved
    if (array[au_pos][j] == 0.0)
    {
        for (int m=1; m <= 10; ++m) //Check which value
            maximizes the distribution
        {
            distribution[j] = 0.0;
            //repeat for any user
            for (int i=0; i < propabilities.length; ++i)
            {
                if (array[i][j] != 0.0)
                    distribution[j] += Math.exp(-
                        Math.pow(m-array[i][j],
                            2)/(2*Math.pow(sigma, 2))) *
                    propabilities[i];
            }

            //Check if it is max. In this case keep it as the
            prediction value
            if (distribution[j] > max)
            {

```

```

        max = distribution[j];
        predictions[j] = m;
    }
    }
    distribution[j] = max;
}

String[] recommendations = new String[predictions.length];

//Sorting of the results
int maxpos;
for(int i = 0; i < predictions.length; i++)
{
    maxpos = i;
    for(int j = i; j < predictions.length; j++)
        if (predictions[j]>predictions[maxpos])
            maxpos = j;

    //switch positions i and minpos
    int tmpitem = items[i];
    double predval=predictions[i];
    items[i] = items[maxpos];
    predictions[i] = predictions[maxpos];
    items[maxpos]=tmpitem;
    predictions[maxpos]=predval;
    recommendations[i] = items[i]+"|"+predictions[i];
}
return recommendations;
}
}

```