



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Σύστημα Αυτοματοποιημένης Μοντελοποίησης και
Διαχείρισης XML Schemas στην Ηλεκτρονική Διακυβέρνηση**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ειρήνη Κ. Ματζάκου

Επιβλέπων : Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Σύστημα Αυτοματοποιημένης Μοντελοποίησης και
Διαχείρισης XML Schemas στην Ηλεκτρονική Διακυβέρνηση**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ειρήνη Κ. Ματζάκου

Επιβλέπων : Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 15^η Οκτωβρίου 2008.

.....
Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

.....
Δημήτριος Ασκούνης
Επικουρος Καθηγητής Ε.Μ.Π.

.....
Γ. Μέντζας
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2008

.....
Ειρήνη Κ. Ματζάκου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ειρήνη Ματζάκου, 2008

Με επιφύλαξη παντός δικαιώματος. All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Πρόλογος

Η διπλωματική εργασία που τεκμηριώνεται στο παρόν έγγραφο έχει ως αντικείμενο τον σχεδιασμό, την ανάπτυξη και την υλοποίηση ενός συστήματος λογισμικού που μοντελοποιεί δεδομένα και παράγει τα αντίστοιχα XML Σχήματα τα οποία αποτελούν τη βάση για την επίτευξη διαλειτουργικότητας στη διακίνηση πληροφορίας Ηλεκτρονικής Διακυβέρνησης μέσω ετερογενών πληροφοριακών συστημάτων. Το τελικό πρόγραμμα υλοποιήθηκε στην πλατφόρμα ανάπτυξης Microsoft Visual Studio .NET, ενώ οι κανόνες που διέπουν τον σχεδιασμό του και η μεθοδολογία με την οποία έχει υλοποιηθεί, ακολουθούν το πρότυπο UN/CEFACT CCTS και το Ελληνικό Μοντέλο Τεκμηρίωσης.

Ολοκληρώνοντας τη διπλωματική εργασία, θα ήθελα να ευχαριστήσω θερμά τον κ. Ιωάννη Ψαρρά, επιβλέποντα καθηγητή, για την ανάθεση της εργασίας και για τη δυνατότητα που μου δόθηκε μέσω αυτής να αποκτήσω ποικίλες εμπειρίες πάνω στα θέματα με τα οποία ασχολείται το Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης. Τέλος, θα ήθελα να ευχαριστήσω και τον Σπύρο Μουζακίτη, Υποψήφιο Διδάκτορα του προαναφερθέντος εργαστηρίου για την άριστη συνεργασία και πολύτιμη συμβουλευτική και καθοδήγηση καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Οκτώβριος 2008
Ματζάκου Ειρήνη

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός, η ανάπτυξη και η υλοποίηση ενός συστήματος λογισμικού το οποίο βασιζόμενο σε συγκεκριμένους κανόνες και μεθοδολογία έχει ως στόχο την αποθήκευση, επεξεργασία και μοντελοποίηση ενός αριθμού δεδομένων καθώς επίσης και την παραγωγή ενός τελικού εγγράφου, το οποίο περιέχει μοντελοποιημένη πληροφορία με σκοπό την αποτελεσματική επικοινωνία ανάμεσα στα προηγμένης τεχνολογίας πληροφοριακά συστήματα Ηλεκτρονικής Διακυβέρνησης, τα οποία διασυνδέονται μεταξύ τους για την ανταλλαγή συγκεκριμένων δεδομένων με απώτερο σκοπό τη διευκόλυνση της παροχής υπηρεσιών κάθε είδους στους πολίτες.

Οι κανόνες που διέπουν τη διαδικασία μοντελοποίησης της πληροφορίας στηρίζονται στη μεθοδολογία UN/CEFACT CCTS (Core Components Technical Specification). Η μεθοδολογία αυτή προσδίδει στα δεδομένα που απαρτίζουν την πληροφορία ένα κοινό σημασιολογικό περιεχόμενο μέσω της χρήσης των Core Components και με αυτό τον τρόπο επιτυγχάνει να γίνεται αντιληπτή η μεταδιδόμενη πληροφορία με τον ίδιο τρόπο από κάθε εμπλεκόμενο στη μεταφορά της, παρόλη την ετερογένεια των πληροφοριακών συστημάτων που μπορεί να χρησιμοποιεί ο κάθε εμπλεκόμενος.

Η παραγωγή των τελικών επιχειρησιακών εγγράφων βασίζεται στην παραγωγή XML σχημάτων (XML Schema) και συμμορφώνεται με την προτυποποίηση του CCTS. Τα XML σχήματα χρησιμεύουν στον ορισμό της δομής και του τύπου των μεταφερόμενων δεδομένων, ενώ η μεταφορά των δεδομένων αυτών και η περαιτέρω χρήση τους μέσω διαφόρων πληροφοριακών συστημάτων γίνεται μέσω του προτύπου της XML.

Για την υλοποίηση του συστήματος χρησιμοποιήθηκε η πλατφόρμα προγραμματισμού Microsoft Visual Studio .NET και το σύστημα διαχείρισης βάσεων δεδομένων (DBMS) Microsoft SQL Server.

Λέξεις – Κλειδιά: Ηλεκτρονική Διακυβέρνηση, Διαλειτουργικότητα, Ελληνικό Πλαίσιο Ηλεκτρονικής Διακυβέρνησης, XML/XML Σχήμα, Βάση Δεδομένων, Visual Studio .NET, SQL Server.

Abstract

The objective of this diploma thesis is the design, development and deployment of a software system which, following specific rules and methodology, aims to store, process and model a certain amount of data and produce a final document which contains modeled information. This kind of documents are used to model data for the efficient communication among the high technology information systems, that are being used in Electronic Government and are interconnected in order to exchange these data for the facilitation of any kind of service offered to the citizens.

The principles that rule the data modelling procedure are based on the UN/CEFACT CCTS (Core Components Technical Specification) methodology. This methodology, assigns to the data that constitute the information, common semantics, through the use of certain Core Components. This fact renders the transmitted information perceptible in the same way by every system during its transfer, despite the heterogeneity of the involved information systems.

The business documents that carry the transmitted information are written in XML and must comply with the normalization of CCTS. For that reason, we create and use XML Schemata which define the structure and the type of the transmitted data in the business documents. The transfer of these business documents is realized according to the XML standard among several information systems.

The development and deployment of this system was realized using the Microsoft Visual Studio .NET 2005 programming platform, and the database management system (DBMS) Microsoft SQL Server 2005.

Keywords: e-government, interoperability, Ηλεκτρονική Διακυβέρνηση, Διαλειτουργικότητα, Greek Standard for e-government, XML/XML Schema, Database (DBMS), Visual Studio .NET, SQL Server.

Πίνακας Περιεχομένων

1 ΕΙΣΑΓΩΓΗ.....	13
1.1 Αντικείμενο Διπλωματικής.....	13
1.2 Οργάνωση του τόμου.....	17
2 ΠΕΡΙΓΡΑΦΗ ΘΕΜΑΤΟΣ.....	18
2.1 Ηλεκτρονική Διακυβέρνηση	18
2.2 Διαλειτουργικότητα.....	22
2.2.1 Πρωτοβουλίες Διαλειτουργικότητας σε Διεθνές Επίπεδο.....	24
2.2.1.1 Ευρωπαϊκή Ένωση – EIF (European Interoperability Framework).....	24
2.2.1.2 Ηνωμένο Βασίλειο – UK Schema Library.....	27
2.2.1.3 Ηνωμένες Πολιτείες Αμερικής – NIEM (National Information Exchange Model).....	28
2.2.2 Ελλάδα – Πλαίσιο Ηλεκτρονικής Διακυβέρνησης (ΠΗΔ)	29
2.3 Ο Οργανισμός UN/CEFACT.....	32
2.3.1 CCTS (Core Component Technical Specification) και Core Components.....	34
2.3.2 Business Information Entities.....	38
2.4 XML.....	40
2.4.1 Ορθότητα XML εγγράφων.....	45
2.4.2 ebXML (electronic business eXtensible Markup Language).....	49
2.4.3 UBL (Universal Business Language).....	50
2.5 Στόχος.....	52
2.5.1 Φυσικό Πρόβλημα.....	52
2.5.2 Περιγραφή συστήματος.....	55
3 ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ.....	56
3.1 Εισαγωγή.....	56
3.1.1 UML (Unified Modeling Language).....	56
3.1.2 Microsoft Office Visio 2007.....	57
3.2 Έγγραφο Προδιαγραφών των Απαιτήσεων από το Λογισμικό Σύστημα.....	57
3.2.1 Εισαγωγή.....	57
3.2.1.1 Σκοπός.....	57
3.2.1.2 Εμβέλεια.....	57
3.2.2 Μοντέλο Περιπτώσεων Χρήσης.....	58
3.2.2.1 Στίγμα.....	58
3.2.2.2 Προοπτική.....	59
3.2.2.3 Γενική εικόνα.....	59
3.2.2.4 Περιορισμοί Σχεδίασης – Συμμόρφωση με τα πρότυπα.....	59
3.2.2.5 Επίδοση.....	59
3.2.3 Ανάλυση περιπτώσεων χρήσης.....	60
3.2.3.1 Περίπτωση Χρήσης 1.....	60
3.2.3.2 Περίπτωση Χρήσης 2.....	70
3.2.3.3 Περίπτωση Χρήσης 3.....	75
3.3 Έγγραφο Περιγραφής Αρχιτεκτονικής Συστήματος.....	78
3.3.1 Εισαγωγή.....	78
3.3.1.1 Σκοπός.....	78
3.3.1.2 Ενδιαφερόμενοι.....	78

3.3.2 Αρχιτεκτονικές αποφάσεις.....	78
3.3.2.1 Πλατφόρμα (υλισμική, λογισμική).....	78
3.3.2.2 Σχολή / Γλώσσα προγραμματισμού.....	79
3.3.2.3 Κωδική Γλώσσα.....	79
3.3.3 Αρχιτεκτονικές όψεις.....	79
3.3.3.1 Περιβαλλοντική/Υπηρεσιακή.....	79
3.3.3.2 Πλατφορμική.....	80
3.3.3.3 Δομική.....	80
3.3.3.4 Εκτελεστική.....	81
3.4 Έγγραφο Περιγραφής Λεπτομερούς Σχεδίου.....	82
3.4.1 Εισαγωγή.....	82
3.4.1.1 Σκοπός.....	82
3.4.1.2 Ενδιαφερόμενοι.....	82
3.4.2 Σχεδιαστικές Αποφάσεις.....	82
3.4.3 Σχεδιαστικές Όψεις.....	87
3.4.3.1 Αποσυνθετική.....	87
3.4.3.2 Δεδομενική.....	90
3.4.3.3 Συνεργατική.....	91
4 ΥΛΟΠΟΙΗΣΗ – ΕΛΕΓΧΟΣ.....	93
4.1 Εργαλεία υλοποίησης.....	93
4.1.1 Visual Studio .NET 2005.....	93
4.1.2 VISUAL BASIC .NET.....	95
4.1.3 .NET Framework 2.0.....	96
4.1.4 ADO .NET.....	97
4.1.5 Microsoft SQL Server 2005.....	99
4.2 Λεπτομέρειες Υλοποίησης.....	101
4.3 Έλεγχος εφαρμογής – Έγγραφο Περιγραφή Δυναμικού Ελέγχου.....	107
4.3.1 Οντότητα προς έλεγχο.....	107
4.3.2 Είδος ελέγχου.....	107
4.3.3 Περιπτώσεις δοκιμής.....	107
5 ΕΠΙΛΟΓΟΣ.....	136
5.1 Σύνοψη.....	136
5.2 Συμπεράσματα.....	138
5.3 Μελλοντικές Επεκτάσεις.....	139
6 ΒΙΒΛΙΟΓΡΑΦΙΑ.....	141
7 ΠΑΡΑΡΤΗΜΑ.....	143
7.1 FamilyStatusDocument.xsd.....	144
7.2 IdentityCardDocument.xsd.....	147
7.3 TaxClearanceApplicationDocument.xsd.....	149
7.4 TaxClearanceDocument.xsd.....	155

1

Εισαγωγή

Ο παρόν τόμος αποτελεί την τεκμηρίωση της σχεδίασης και της ανάπτυξης ενός προγράμματος το οποίο παράγει XML Σχήματα τα οποία μοντελοποιούν δεδομένα Ηλεκτρονικής Διακυβέρνησης. Στο εισαγωγικό αυτό κεφάλαιο παρουσιάζεται η γνωστική περιοχή και το θέμα στο οποίο στηρίζεται η διπλωματική εργασία, καθώς και η οργάνωση του τόμου σε κεφάλαια.

1.1 Αντικείμενο Διπλωματικής

Αντικείμενο αυτής της διπλωματικής εργασίας είναι ο σχεδιασμός, η ανάπτυξη και υλοποίηση ενός λογισμικού συστήματος για την αποθήκευση, επεξεργασία και μοντελοποίηση πληροφορίας που περιέχεται στα διάφορα έγγραφα Ηλεκτρονικής Διακυβέρνησης καθώς και η παραγωγή των κατάλληλων XML Σχημάτων που ορίζουν τον τύπο και τη δομή αυτών των εγγράφων. Η μοντελοποιημένη πληροφορία διακινείται μέσω διαφόρων πληροφοριακών συστημάτων τα οποία χρησιμοποιούνται από φορείς δημόσιας διοίκησης σε εγχώριο επίπεδο, όπως υπουργεία και οργανισμοί τοπικής αυτοδιοίκησης, από κυβερνητικούς οργανισμούς και επιχειρήσεις.

Απώτερος στόχος είναι η πρακτική υλοποίηση της μεθοδολογίας που έχει αναπτυχθεί για τη σημασιολογική Διαλειτουργικότητα, η οποία θα διέπει όλα τα πληροφοριακά συστήματα που θα εμπλέκονται στη μετάδοση της πληροφορίας Ηλεκτρονικής Διακυβέρνησης. Η επίτευξη διαλειτουργικότητας συνιστά θεμελιώδη

στόχο του Ελληνικού Πλαισίου Διαλειτουργικότητας Ηλεκτρονικής Διακυβέρνησης. Το πλαίσιο αυτό αποτελεί δράση του επιχειρηματικού προγράμματος «Κοινωνία της Πληροφορίας» και κύριος σκοπός του είναι η εξασφάλιση συμβατότητας και επικοινωνίας μεταξύ των συστημάτων Πληροφορικής του Δημοσίου Τομέα.

Είναι γνωστό ότι στη σύγχρονη «Κοινωνία της Πληροφορίας» γίνεται καθημερινά διακίνηση μεγάλου όγκου δεδομένων μεταξύ διαφόρων φορέων δημόσιας διοίκησης, επιχειρήσεων και άλλων υπηρεσιών. Το γεγονός αυτό, οφείλεται στην ταχύτατη εξάπλωση του διαδικτύου και του παγκόσμιου ιστού που συνέβαλλαν στην ολοένα αυξανόμενη παροχή ηλεκτρονικών υπηρεσιών εκ μέρους των φορέων της πολιτείας προς τους πολίτες με τη μορφή της Ηλεκτρονικής Διακυβέρνησης. Η Ηλεκτρονική Διακυβέρνηση δεν αφορά μόνο στην παροχή online υπηρεσιών και στην καλύτερη εξυπηρέτηση πολιτών και επιχειρήσεων, αλλά συνιστά ένα νέο σύνολο τεχνολογικά υποβοηθούμενων διαδικασιών, που προάγουν την αλληλεπίδραση ανάμεσα στην κυβέρνηση και τους πολίτες, τους διεθνείς οργανισμούς, τις επιχειρήσεις ή τις κυβερνήσεις άλλων κρατών. Συνεπώς, είναι φανερό ότι η διακίνηση της πληροφορίας Ηλεκτρονικής Διακυβέρνησης θα πρέπει να είναι όσο το δυνατό πιο αποδοτική όσον αφορά τόσο το χρόνο όσο και το κόστος.

Παρόλα αυτά, η τρέχουσα κατάσταση που επικρατεί σε τέτοια συστήματα για τη διακίνηση της πληροφορίας δεν είναι αρκετά ικανοποιητική καθώς δεν υπάρχει κοινό πρότυπο και κοινοί κανόνες αναπαράστασης, κωδικοποίησης και τελικά μοντελοποίησης της μεταδιδόμενης πληροφορίας. Αυτό έχει ως αποτέλεσμα τα δεδομένα που αποστέλλονται ή λαμβάνονται από τον ένα φορέα στον άλλο ή από τη μια υπηρεσία στην άλλη, να πρέπει να αποκωδικοποιηθούν από τη μορφή που τα έστειλε ο ένας φορέας και να «μεταφρασθούν» στη μορφή που «καταλαβαίνει» και χρησιμοποιεί το πληροφοριακό σύστημα του παραλήπτη. Έτσι, παρατηρείται μεγάλο κόστος σε χρόνο για τη μετάφραση της πληροφορίας από το ένα πληροφοριακό σύστημα στο άλλο, ενδεχόμενη μεταβολή ή απώλεια δεδομένων, καθώς επίσης και κόστος στους πόρους του συστήματος της κάθε πλευράς (αύξηση λειτουργικών χαρακτηριστικών του κάθε συστήματος). Όταν μάλιστα, αυξάνεται πολύ το πλήθος των πληροφοριακών συστημάτων τα οποία εμπλέκονται στη διακίνηση της πληροφορίας, τότε γίνεται αντιληπτό ότι το κόστος σε χρόνο και δαπάνες αλλά και σε δεδομένα, από την ελλιπή και ενδεχομένως μη ευσταθή επικοινωνία ανάμεσα στα συστήματα, αυξάνεται δραματικά.

Κατά συνέπεια, βασικό ζητούμενο όσον αφορά στη μετάδοση δεδομένων στην εποχή μας μεταξύ ετερογενών οντοτήτων έχει αναχθεί η επίτευξη της

Διαλειτουργικότητας, της ικανότητας δηλαδή διαφορετικών συστημάτων να χρησιμοποιούν μία μορφή «κοινής» γλώσσας επικοινωνίας, στα πλαίσια της οποίας θα αντιλαμβάνονται με τον ίδιο τρόπο τις βασικές σημασιολογικές οντότητες, γεγονός που μπορεί να επιτευχθεί μέσω της προτυποποίησης της μορφής των ανταλλασσομένων πληροφοριών και δεδομένων.

Προκειμένου να υλοποιηθεί αυτή η χρήση της «κοινής» γλώσσας, είναι αναγκαία η μοντελοποίηση της μεταδιδόμενης πληροφορίας με συγκεκριμένο τρόπο. Βασικό μέρος αυτής της διαδικασίας μοντελοποίησης είναι η μεθοδολογία UN/CEFACT CCTS (Core Components Technical Specification) η οποία προσδίδει στα δεδομένα που απαρτίζουν την πληροφορία ένα κοινό σημασιολογικό περιεχόμενο μέσω της χρήσης συγκεκριμένων δομικών συστατικών που ονομάζονται Core Components. Τα δομικά αυτά συστατικά αντιπροσωπεύουν τις οντότητες που απαρτίζουν την πληροφορία που αποτυπώνεται σε κάποιο έγγραφο και αποτελούν τη βάση για να αντιπροσωπεύσουν από πρωτογενείς οντότητες, (όπως για παράδειγμα την οντότητα «Επώνυμο» ενός πολίτη που εμπλέκεται στη διαδικασία που περιγράφει το έγγραφο), έως και πιο σύνθετες οντότητες σε ένα ανώτερο αφαιρετικό επίπεδο, (όπως για παράδειγμα το σύνολο ενός πλήθους πληροφοριών σχετικά με τον πολίτη, που αποτυπώνεται στην οντότητα «Πολίτης»), χάρη στο σαφές ιεραρχικό μοντέλο δόμησης της πληροφορίας το οποίο ακολουθούν. Τα Core Components αποτελούν τη βάση για τη δημιουργία των Business Information Entities (BIEs) τα οποία δημιουργούνται από την τοποθέτηση των Core Components σε ορισμένο επιχειρηματικό περιβάλλον και αποτελούν τις ουσιαστικές μονάδες δόμησης της πληροφορίας.

Κατ' αυτό τον τρόπο, η πληροφορία έχει συγκεκριμένη κωδικοποίηση και η μετάδοσή της από το ένα πληροφοριακό σύστημα στο άλλο δεν απαιτεί κάποια «μετάφραση» ή κάποια άλλη διαδικασία μιας και το πρότυπο κωδικοποίησης της πληροφορίας θα είναι ενιαίο για όλα τα συστήματα.

Έπειτα από την απόδοση σημασιολογικού περιεχομένου στη διαχειριζόμενη πληροφορία, ενδιάμεσο και απαραίτητο στάδιο είναι η αποθήκευσή της σε μια βάση δεδομένων. Η βάση αυτή, θα πρέπει να είναι οργανωμένη με συγκεκριμένο τρόπο και συγκεκριμένες σχέσεις μεταξύ των δεδομένων από τα οποία αποτελείται, όπως αυτές υπαγορεύονται από το προαναφερθέν πρότυπο για την επίτευξη Διαλειτουργικότητας.

Τέλος, με ανάκτηση των κατάλληλων δεδομένων από αυτή τη βάση, ακολουθεί η δημιουργία των αντίστοιχων XML σχημάτων τα οποία περιέχουν τους κανόνες για την κωδικοποίηση της πληροφορίας. Με αυτά τα XML Σχήματα πρέπει να συμμορφώνονται

τα εκάστοτε XML αρχεία που περιέχουν τα τελικά δεδομένα του εκάστοτε επιχειρησιακού εγγράφου και αποτελούν τα έγγραφα ηλεκτρονικής διακυβέρνησης τα οποία μεταδίδονται ανάμεσα στα διάφορα συστήματα με βάση το XML πρότυπο. Κατ' αυτό τον τρόπο, η προτυποποιημένη πληροφορία καθίσταται μεταφέρσιμη, εύληπτη και επεξεργάσιμη από τα πληροφοριακά συστήματα τα οποία αποτελούν και τον τελικό αποδέκτη της εργασίας μας.

Τα XML Σχήματα περιγράφουν ένα σύνολο κανόνων με τους οποίους ένα XML έγγραφο θα πρέπει να συμμορφώνεται προκειμένου να θεωρείται έγκυρο (valid). Η γλώσσα της οποίας μέρος αποτελούν τα XML Σχήματα ονομάζεται XSD (XML Schema Definition) και κύριο χαρακτηριστικό της είναι το γεγονός ότι υποστηρίζει τύπους δεδομένων, γεγονός που απλοποιεί την πραγματοποίηση διαδικασιών όπως η περιγραφή επιτρεπτού περιεχομένου ενός αρχείου, καθώς και ο έλεγχος ορθότητάς του. Κατά τη διαδικασία δημιουργίας τους, τα XML Σχήματα συμμορφώθηκαν πλήρως στο πρότυπο XML Schema, ενώ ακολουθήθηκαν και οι προδιαγραφές που θέτει το ελληνικό e-GIF (electronic Government Interoperability Framework), όπως αυτές αποτυπώνονται στο Μοντέλο Τεκμηρίωσής του, για τη σχεδίαση των XML Σχημάτων.

Στα πλαίσια υλοποίησης της παρούσας διπλωματικής εργασίας, αναπτύχθηκε μία σαφώς καθορισμένη και με διακριτά βήματα διαδικασία σχεδιασμού, ανάπτυξης και υλοποίησης της προαναφερθείσας πλατφόρμας προτυποποίησης και διαχείρισης δεδομένων Ηλεκτρονικής Διακυβέρνησης και στη συνέχεια παρήχθησαν μερικά έγγραφα από διάφορους τομείς της καθημερινής ζωής που εμπειρείχαν τους κανόνες με τους οποίους πρέπει να συμμορφώνεται η προς προτυποποίηση πληροφορία.

Για τη συγγραφή κώδικα με στόχο την ανάπτυξη και υλοποίηση του παραπάνω λογισμικού συστήματος, χρησιμοποιήθηκε η γλώσσα Visual Basic .NET η οποία είναι μια πολύ ισχυρή γλώσσα προγραμματισμού που επιτρέπει τη σύνταξη αξιόπιστου και ισχυρού λογισμικού. Παρέχει τα εργαλεία που είναι απαραίτητα για τη δημιουργία εφαρμογών οι οποίες θα ενισχύσουν τις σύγχρονες εταιρίες και θα προωθήσουν την επόμενη γενιά λογισμικού για Windows, Web και φορητές συσκευές.

Ως πλατφόρμα ανάπτυξης του όλου έργου, χρησιμοποιήθηκε το Visual Studio 2005, ένα ισχυρό εργαλείο το οποίο επιτρέπει τη γρήγορη δημιουργία μιας μεγάλης ποικιλίας εφαρμογών και επαγγελματικού λογισμικού για τα Microsoft Windows και για το Web σε χρόνο ρεκόρ. Το Visual Studio .NET 2005 έχει δημιουργηθεί για να αντιμετωπίσει τις μεγαλύτερες σημερινές προκλήσεις όσον αφορά την ανάπτυξη λογισμικού.

Τέλος, για την αποθήκευση, διαχείριση και ανάκτηση πληροφορίας σχεδιάστηκε και αναπτύχθηκε μια βάση δεδομένων. Για το σκοπό αυτό, έγινε χρήση του Microsoft SQL Server 2005 που αποτελεί μια ολοκληρωμένη προσφορά βάσης δεδομένων και ανάλυσης για τη γρήγορη παράδοση της επόμενης γενιάς επαγγελματικών λύσεων, λύσεων ηλεκτρονικού εμπορίου και αποθήκευσης δεδομένων με δυνατότητα κλιμάκωσης.

1.2 Οργάνωση του τόμου

Ο τόμος της παρούσας διπλωματικής εργασίας έχει οργανωθεί σε 7 κεφάλαια:

Το **1^ο κεφάλαιο** είναι εισαγωγικό και παρουσιάζει το αντικείμενο που πραγματεύεται η διπλωματική εργασία καθώς και την οργάνωση του τόμου.

Το **2^ο κεφάλαιο** θέτει το θεωρητικό υπόβαθρο για τις βασικές έννοιες που διέπουν τη διπλωματική εργασία, καταγράφει το φυσικό πρόβλημα και καταλήγει με μια συνοπτική παρουσίαση του συστήματος που θα υλοποιηθεί.

Το **3^ο κεφάλαιο** αναφέρεται στην Ανάλυση και το Σχεδιασμό του συστήματος ο οποίος τεκμηριώνεται με τα εξής έγγραφα: Προδιαγραφών Απαιτήσεων από το Λογισμικό Σύστημα (ΠΑΛΣ), Περιγραφή του Αρχιτεκτονικού Σχεδίου (ΠΑΣ) και Περιγραφή του Λεπτομερούς Σχεδίου (ΠΑΣ).

Το **4^ο κεφάλαιο** παρουσιάζει την υλοποίηση και τον έλεγχο της εφαρμογής. Μετά από μια θεωρητική ανασκόπηση των εργαλείων υλοποίησης, περιγράφονται ορισμένες λεπτομέρειες στην υλοποίηση και το κεφάλαιο ολοκληρώνεται με την τεκμηρίωση του ποιοτικού ελέγχου του συστήματος.

Το **5^ο κεφάλαιο** αποτελεί μια σύνοψη της παρούσας διπλωματικής εργασίας και παρατίθενται συμπεράσματα και πιθανές μελλοντικές επεκτάσεις του συστήματος που υλοποιήθηκε.

Στο **6^ο κεφάλαιο** γίνεται μια παρουσίαση της βιβλιογραφίας που χρησιμοποιήθηκε κατά την εκπόνηση του τόμου και της εφαρμογής.

Στο **7^ο κεφάλαιο** είναι το παράρτημα της παρούσας εργασίας όπου παρατίθενται μερικά από τα valid XML Schemas που μοντελοποιήθηκαν στα πλαίσια της διπλωματικής εργασίας.

2

Περιγραφή Θέματος

Στο παρόν κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο για έννοιες όπως η Ηλεκτρονική Διακυβέρνηση, η Διαλειτουργικότητα και η γλώσσα XML. Ακολουθεί ανάλυση των εννοιών αυτών, προσπάθειες που έχουν επιτελεσθεί στην Ελλάδα αλλά και στο εξωτερικό για επίτευξη Διαλειτουργικότητας στο χώρο της Ηλεκτρονικής Διακυβέρνησης και τέλος, γίνεται αναφορά στην XML και περαιτέρω ανάλυση της έννοιας των XML σχημάτων και πώς αυτά συμβάλλουν στον παραπάνω στόχο. Κλείνοντας, δίνεται μια συνοπτική περιγραφή του συστήματος που υλοποιήθηκε.

2.1 Ηλεκτρονική Διακυβέρνηση

Στην εποχή μας, η ραγδαία ανάπτυξη και εμφάνιση νέων ψηφιακών τεχνολογιών στο προσκήνιο έχει μεταβάλλει το τοπίο και στον τομέα της Διακυβέρνησης και Διοίκησης του κράτους. Έχει γίνει λοιπόν επιτακτική η ανάγκη ενσωμάτωσης των νέων αυτών τεχνολογιών στο συγκεκριμένο τομέα με απώτερο σκοπό τον εκσυγχρονισμό, την απλούστευση και το υψηλό επίπεδο λειτουργικότητας των παρεχόμενων προς τους πολίτες δημοσίων υπηρεσιών, γεγονός που θα συνάδει και με τις επιταγές της νέας διαμορφούμενης κατάστασης.

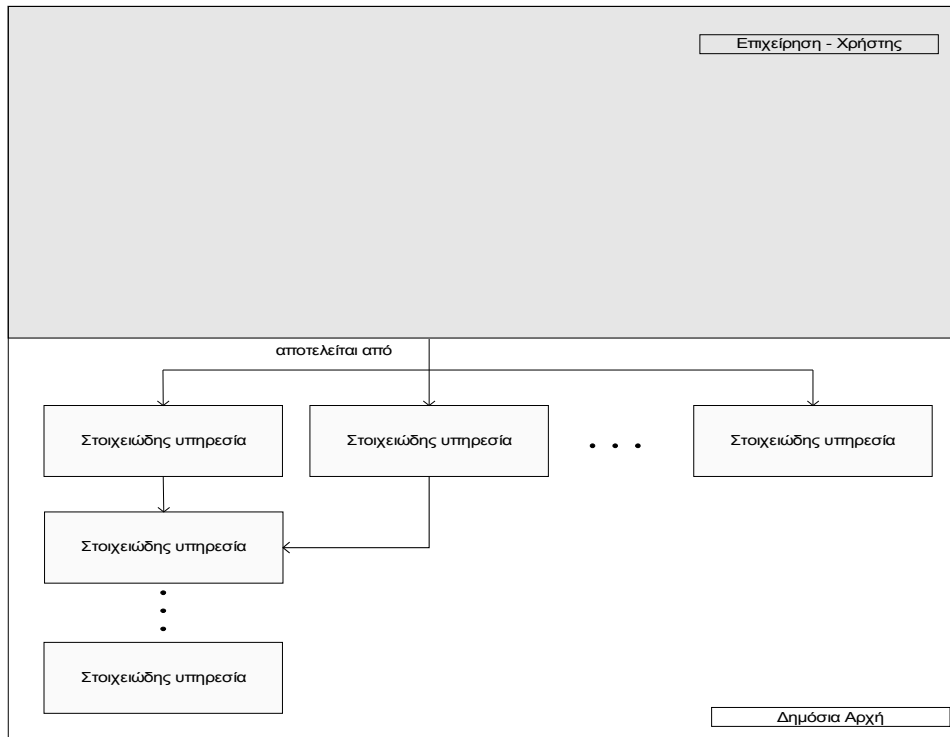
Προς την κατεύθυνση αυτή κινείται και το πλαίσιο της Ηλεκτρονικής Διακυβέρνησης (e-Government), ή Ηλεκτρονικής Διοίκησης όπως αλλιώς ονομάζεται,

πρωτεύον σκοπός του οποίου είναι η εφαρμογή των Τεχνολογιών Πληροφορικής και Επικοινωνίας στον τομέα της δημόσιας διοίκησης. Ο όρος Ηλεκτρονική Διακυβέρνηση αναφέρεται στη χρήση και την αξιοποίηση των τεχνολογιών του Διαδικτύου ως μία πλατφόρμα για την ανταλλαγή πληροφοριών, την παροχή υπηρεσιών και την επικοινωνία μεταξύ πολιτών, επιχειρήσεων και γενικά κάθε οντότητας που αποτελεί κομμάτι της διακυβέρνησης.

Κύριο μέλημα της Ηλεκτρονικής Διακυβέρνησης είναι η ποιοτική εξυπηρέτηση και η παροχή ολοκληρωμένων υπηρεσιών προς τους πολίτες και τις επιχειρήσεις. Οι υπηρεσίες αυτές θα χαρακτηρίζονται από αξιοπιστία και συνέπεια ως προς τον τρόπο που παρέχονται, ενώ θα είναι σε θέση να προσφέρουν στον άμεσα ενδιαφερόμενο σαφή διευκόλυνση σχετικά με τις υπηρεσίες που τον αφορούν. Ο ηλεκτρονικός αυτός τρόπος παροχής υπηρεσιών είναι σε θέση να γλιτώσει τόσο τους πολίτες όσο και τις επιχειρήσεις από χρονοβόρες γραφειοκρατικές διαδικασίες, δίνοντάς τους τη δυνατότητα να απέχουν ένα μόνο «κλικ» από την άμεση και απρόσκοπτη εξυπηρέτησή τους.

Για την ακρίβεια, ο εκάστοτε χρήστης μιας ηλεκτρονικής υπηρεσίας της δημόσιας διοίκησης δε χρειάζεται να γνωρίζει ούτε την λειτουργία αλλά ούτε και την οργανωτική δομή των μονάδων δημόσιας διοίκησης που εμπλέκονται στην εξυπηρέτησή του. Αρκεί απλά να έρχεται σε επαφή με το σημείο εκκίνησης της υπηρεσίας, το οποίο μπορεί να είναι για παράδειγμα ένα δημόσιο πληροφοριακό σύστημα, και να λαμβάνει το αποτέλεσμα της υπηρεσίας από ένα σημείο εξόδου, χωρίς να χρειάζεται να περιπλέκεται με τα ενδιάμεσα στάδια που οδήγησαν στην περάτωση της συγκεκριμένης υπηρεσίας. Και αυτό, διότι κάθε σύνθετη υπηρεσία στα πλαίσια της οποίας εμπλέκεται ο πολίτης ή η επιχείρηση, μπορεί να αναλυθεί σε επιμέρους στοιχειώδεις υπηρεσίες, την ακριβή δομή και περάτωση των οποίων δεν είναι αναγκαίο να γνωρίζει ο ενδιαφερόμενος προς εξυπηρέτηση.

Ο τρόπος με τον οποίο οι επιμέρους διαδικασίες απαρτίζουν την αρχική σύνθετη υπηρεσία εμφανίζεται στο παρακάτω σχήμα:



Εικόνα 2.1-1 Σύνθεση μιας σύνθετης υπηρεσίας από στοιχειώδεις υπηρεσίες.

Γίνεται λοιπόν αντιληπτό ότι η δημιουργία ενός πλαισίου ηλεκτρονικής διακυβέρνησης που θα εναρμονίζεται με τις επιταγές των νέων τεχνολογιών και που θα έχει προσαρμοστεί στις ανάγκες των πολιτών και των επιχειρήσεων, μπορεί να βελτιώσει σε σημαντικό βαθμό το επίπεδο εξυπηρέτησης και παροχής υπηρεσιών σε αυτούς. Πιο συγκεκριμένα, η εφαρμογή του έργου Ηλεκτρονικής Διακυβέρνησης παρουσιάζει πολλά πλεονεκτήματα όπως τα ακόλουθα:

- Υποστήριξη των πολιτών και των επιχειρήσεων στην ανεύρεση και αξιοποίηση δημόσιας πληροφορίας και στην ηλεκτρονική διάδρασή τους με τη Δημόσια Διοίκηση, μέσα από τον καθορισμό των προτύπων και των προδιαγραφών που θα συμβάλλουν στην ομογενοποίηση της παρουσίας της Δημόσιας Διοίκησης στο Διαδίκτυο.
- Υποστήριξη της ανταλλαγής περιεχομένου μεταξύ πληροφοριακών συστημάτων και της παροχής ασφαλών και ολοκληρωμένων υπηρεσιών Ηλεκτρονικών Συναλλαγών προς τους πολίτες και τις επιχειρήσεις

- Μείωση του κόστους επεξεργασίας της πληροφορίας. Κάτι τέτοιο επιτυγχάνεται μέσω της αυτοματοποίησης διαδικασιών ρουτίνας, της ηλεκτρονικοποίησης αρχείων και εντύπων και της μείωσης του κόστους ταξινομήσεων, εκτυπώσεων και αποστολής εντύπων.
- Μείωση του κόστους παροχής υπηρεσιών με online υπηρεσίες. Οι online υπηρεσίες έχουν ως αποτέλεσμα την ύπαρξη λιγότερων τοπικών γραφείων και τη μείωση ανθρωποδύναμης ανά παρεχόμενη υπηρεσία, καθώς επίσης και τη δημιουργία πιο εξειδικευμένων υπηρεσιών που θα οδηγήσουν στη μείωση του χρόνου αναμονής των χρηστών και θα λειτουργούν καθημερινά σε εικοσιτετράωρη βάση.
- Μείωση των διοικητικών επιβαρύνσεων μέσα από αλλαγές στους ισχύοντες νόμους και κανονισμούς και μέσω της μείωσης του χρόνου, των διαδικασιών και της απαιτούμενης προσπάθειας και εξειδίκευσης.

Ωστόσο, επειδή οι ηλεκτρονικές υπηρεσίες θα παρέχονται από ένα πλήθος φορέων της δημόσιας διοίκησης (όπως για παράδειγμα Οικονομικές υπηρεσίες, Φορολογικές υπηρεσίες, Οργανισμοί Τοπικής Αυτοδιοίκησης - ΟΤΑ) θα πρέπει να υπάρχει ένα πληροφοριακό σύστημα το οποίο θα επιβλέπει τη διαδικασία ανταλλαγής και μετάδοσης της πληροφορίας διαμέσου όλων αυτών των φορέων. Για το λόγο αυτό, στα πλαίσια της Ηλεκτρονικής Διακυβέρνησης γίνεται μία οργανωμένη προσπάθεια δημιουργίας διαδικτυακών πυλών που θα καλύπτουν ένα ευρύ φάσμα εξυπηρέτησης φορέων. Οι διαδικτυακές αυτές πύλες είναι γνωστές με το όνομα Κυβερνητικές Δικτυακές Πύλες (Government Portals) και μπορούν αφενός μεν να χρησιμοποιηθούν για την υιοθέτηση λειτουργικών πρακτικών όσον αφορά στις ηλεκτρονικές υπηρεσίες, ενώ μπορούν αφετέρου να αποτελέσουν την πεμπουσία της αρχιτεκτονικής δόμησης των υπηρεσιών αυτών που προαναφέρθηκε και που παρέχει το αφαιρετικό αυτό interface προς το χρήστη.

2.2 Διαλειτουργικότητα

Τα διοικητικά έγγραφα τα οποία θα διακινούνται μέσω των προαναφερθέντων διαδικτυακών πυλών και θα υλοποιούν την Ηλεκτρονική Διακυβέρνηση όπως περιγράφηκε παραπάνω, έχουν ένα κύκλο ζωής κατά τη διάρκεια του οποίου καθοριστικό ρόλο κατέχει η δυνατότητα επικοινωνίας των υπηρεσιών που εμπλέκονται στη ζωή αυτού του εγγράφου και η ανταλλαγή δεδομένων από ετερογενή πληροφοριακά συστήματα διαφορετικών υπηρεσιών.

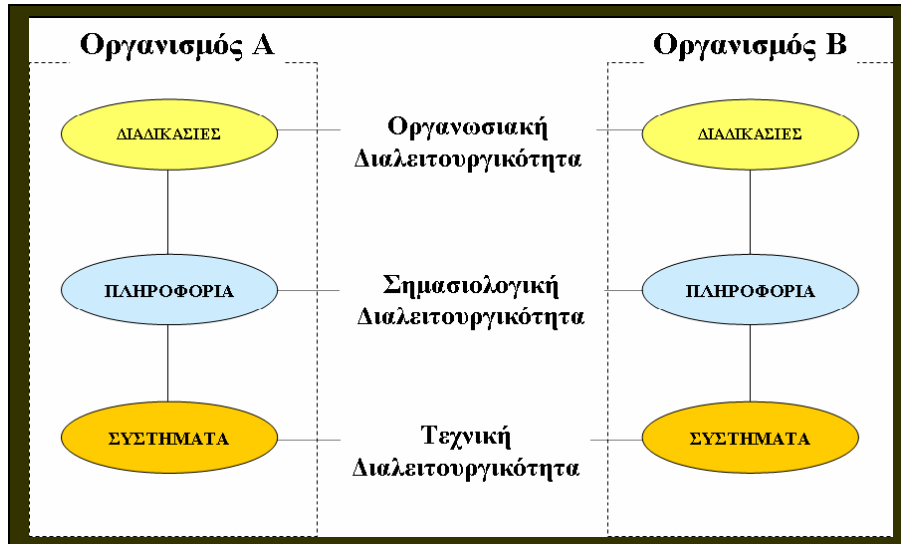
Καθίσταται λοιπόν σαφές το γεγονός ότι, στα πλαίσια της ανάπτυξης του εγχειρήματος της Ηλεκτρονικής Διακυβέρνησης και στον καθορισμό των κατάλληλων προτύπων, πρωτεύοντα ρόλο διαδραματίζει η έννοια της Διαλειτουργικότητας.

Διαλειτουργικότητα είναι η δυνατότητα διακίνησης και χρήσης της πληροφορίας με ενιαίο και αποδοτικό τρόπο από διαφορετικούς φορείς και πληροφοριακά συστήματα καθώς επίσης και η δυνατότητα ανταλλαγής και ενοποίησης δεδομένων που προέρχονται από διαφορετικά πληροφοριακά περιβάλλοντα μέσω της υιοθέτησης κοινών προτύπων. Με τη Διαλειτουργικότητα, δίνεται η εντύπωση στο χρήστη του εξωτερικού περιβάλλοντος πως έχει να κάνει με ένα ενιαίο σύστημα από το οποίο και εξυπηρετείται και όχι από διαφορετικά και ανομοιογενή συστήματα.

Γίνεται επομένως αντιληπτό πως η Διαλειτουργικότητα είναι μία θεμελιώδης έννοια και μία βασικότερη επιδίωξη στα πλαίσια της Ηλεκτρονικής Διακυβέρνησης. Ωστόσο η αποτελεσματική της υλοποίηση μόνο εύκολη υπόθεση δεν μπορεί να θεωρηθεί, από τη στιγμή κιόλας που είναι αναγκαίο να οριστεί σε τρία επίπεδα:

- Στο επίπεδο διαδικασιών, το οποίο είναι υπεύθυνο για τον ορισμό των κατάλληλων σημείων επαφής των διαδικασιών ανάμεσα στους φορείς που επικοινωνούν
- Σε επίπεδο πληροφορίας, όπου θα πρέπει η πληροφορία που ανταλλάσσεται μεταξύ των δύο φορέων όπως επίσης και τα δεδομένα που την απαρτίζουν να έχουν κοινή δομή και
- Σε επίπεδο συστημάτων που υλοποιούν συγκεκριμένα τμήματα των συνολικών διαδικασιών, όπου συμφωνημένα υποσυστήματα θα υλοποιούν τη διασύνδεση σε διαδικαστικό και πληροφοριακό επίπεδο

Επομένως, και σε πλήρη αντιστοιχία με τα επίπεδα που προαναφέρθηκαν, έχουμε τρεις μορφές διαλειτουργικότητας: την Οργανωσιακή, τη Σημασιολογική και την Τεχνική, οι οποίες απεικονίζονται στο παρακάτω σχήμα:



Εικόνα 2.2-2: Επίπεδα Διαλειτουργικότητας ανάμεσα σε δύο φορείς.

Μπορεί λοιπόν η επίτευξη διαλειτουργικότητας να αποτελεί μία επίπονη και απαιτητική διαδικασία, ωστόσο είναι σε θέση να προσφέρει πολλά πλεονεκτήματα και οφέλη τα οποία αναπτύσσονται παρακάτω.

Αρχικά, η πληροφόρηση των πολιτών και η διακίνηση πληροφοριών Ηλεκτρονικής Διακυβέρνησης θα παρέχεται ανεξαρτήτως χρόνου και φυσικού χώρου. Αυτό σημαίνει αυτομάτως βελτίωση της ποιότητας των παρεχομένων υπηρεσιών από τους διάφορους φορείς Δημόσιας Διοίκησης καθώς ο χρήστης επιλέγοντας τον κατάλληλο διάλογο επικοινωνίας μέσω της χρήσης του διαδικτύου θα έχει σε σύντομο χρόνο και απαλλαγμένος από περιττές μετακινήσεις και γραφειοκρατία τα δεδομένα, τις πληροφορίες ή τα έγγραφα που ενδεχομένως χρειάζεται.

Επιπλέον, χάρη στη Διαλειτουργικότητα, η ανταλλαγή δεδομένων θα γίνεται με πιο αποδοτικό τρόπο, αφού οι παρεχόμενες υπηρεσίες θα είναι πλήρως αυτοματοποιημένες αλλά και διασυνδεδεμένες, γεγονός που θα οδηγήσει στην εξοικονόμηση χρόνου και χρημάτων αφού, στην ουσία, θα έχουμε εκ νέου επεξεργασία της ίδιας πληροφορίας που έχει προηγουμένως επανεξεταστεί.

Τέλος, εξαιτίας της Διαλειτουργικότητας θα μπορεί να επιτευχθεί η διαφάνεια στο Δημόσιο Τομέα μέσα από την εξαφάνιση ποιοτικής διακίνησης και διαχείρισης της μεταδιδόμενης πληροφορίας.

2.2.1 Πρωτοβουλίες Διαλειτουργικότητας σε Διεθνές Επίπεδο

Εξαιτίας λοιπόν της ιδιαίτερης σημασίας που καταλαμβάνει η Διαλειτουργικότητα στην Ηλεκτρονική Διακυβέρνηση, οι περισσότερες από τις χώρες που έχουν θέσει ως στόχο την υιοθέτηση και την εφαρμογή των πρακτικών της Ηλεκτρονικής Διακυβέρνησης στη Δημόσια Διοίκηση έχουν δημιουργήσει και θέσει σε εφαρμογή ειδικά πλαίσια τα οποία ασχολούνται κυρίως με την επίτευξη διαλειτουργικότητας και τη θέσπιση κατάλληλων προτύπων που θα εφαρμοστούν στις τεχνολογίες πληροφορικής και επικοινωνιών για να επιτευχθεί Διαλειτουργικότητα.

Οι πιο βασικές και περισσότερο επιτυχημένες προσπάθειες επίτευξης διαλειτουργικότητας είναι οι ακόλουθες:

- Ευρωπαϊκή Ένωση – EIF (European Interoperability Framework)
- Ηνωμένο Βασίλειο – UK Schema Library
- Ηνωμένες Πολιτείες Αμερικής – NIEM (National Information Exchange Model)

Επιλεγμένες προσπάθειες θα αναπτυχθούν συνοπτικά παρακάτω:

2.2.1.1 Ευρωπαϊκή Ένωση – EIF (European Interoperability Framework)

Αρχικά, θα γίνει λόγος για την οργανωμένη προσπάθεια της Ευρωπαϊκής Ένωσης, η οποία ξεκίνησε αρχικά με την εφαρμογή του πλαισίου EU IDA (Electronic Interchange of Data between Administrators). Στο πλαίσιο αυτό, εφαρμόζονταν οι εξελίξεις στους τομείς των Τηλεπικοινωνιών και της Πληροφορικής, για την υποστήριξη της ανταλλαγής ηλεκτρονικών πληροφοριών και δεδομένων μεταξύ των Κυβερνήσεων των Μελών Χωρών. Αντικειμενικός σκοπός κατά τη δημιουργία του ήταν η βελτίωση στη διαδικασία των αποφάσεων και στη διαδικασία λειτουργίας των εσωτερικών αγορών. Κύρια επιδίωξη του IDA στάθηκε η υλοποίηση ενός διευρωπαϊκού δικτύου τηλεματικής μέσω των εξής ενεργειών:

- Ανάπτυξη δικτύων Διαλειτουργικότητας
- Διάθεση των πλεονεκτημάτων του δικτύου στους πολίτες και στη βιομηχανία της Ευρωπαϊκής Ένωσης
- Συνεργασία με τα Κράτη Μέλη και τις Κοινωνικές Υπηρεσίες τους

- Σύγκλιση προς μία κοινή διεπαφή (interface)

Στις μέρες μας, το IDA έχει αντικατασταθεί από το IDABC (Interoperable Delivery of European e-Government Services to Administrations, Businesses and Citizens). Το IDABC εκμεταλλεύεται τις ευκαιρίες που προσφέρει η ευρεία διάδοση των τεχνολογιών πληροφορικής και τηλεπικοινωνιών, με σκοπό να επιτύχει τα ακόλουθα:

- Να ενθαρρύνει και να υποστηρίξει τη διανομή υπηρεσιών προερχόμενων από δημόσιους παράγοντες σε διεθνές επίπεδο, οι οποίες θα απευθύνονται σε Πολίτες και σε Επιχειρήσεις σε ολόκληρη την Ευρώπη.
- Να βελτιώσει την αποδοτικότητα και τη συνεργασία μεταξύ των Ευρωπαϊκών Οργανισμών Δημόσιας Διοίκησης.
- Να αυξήσει το βιοτικό επίπεδο της Ευρώπης και να συμβάλλει στη μετατροπή της σε ένα μέρος που θα είναι ελκυστικό για εργασία και επένδυση.

Για να επιτύχει τους στόχους του, το IDABC εκδίδει προτυποποιήσεις, αναπτύσσει λύσεις και παρέχει υπηρεσίες που διευκολύνουν την ηλεκτρονική επικοινωνία μεταξύ διεθνών και ευρωπαϊκών δημόσιων οργανισμών, ενώ παράλληλα προσφέρει σύγχρονες υπηρεσίες όσον αφορά στο δημόσιο τομέα σε πολίτες και επιχειρήσεις σε όλη την Ευρώπη. Το πρόγραμμα IDABC επιπλέον παρέχει κεφάλαια χρηματοδότησης σε ερευνητικά έργα τα οποία απευθύνονται στις απαιτήσεις της ευρωπαϊκής κυβερνητικής πολιτικής συμβάλλοντας με αυτόν τον τρόπο στη βελτίωση της συνεργασίας μεταξύ των κυβερνήσεων κατά μήκος της Ευρώπης. Μέσω της χρήσης ανώτατου επιστημονικού επιπέδου τεχνολογιών πληροφορικής και τηλεπικοινωνιών, μέσω της ανάπτυξης κοινά αποδεκτών λύσεων και υπηρεσιών και μέσω της παροχής μίας πλατφόρμας για την ανταλλαγή αποτελεσματικών πρακτικών μεταξύ δημόσιων κυβερνήσεων το IDABC συμβάλλει στο πρόγραμμα i2000, το οποίο αποτελεί μία πρωτοβουλία για τον εκμοντερνισμό του ευρωπαϊκού δημόσιου τομέα.

Το IDABC, στα πλαίσια της συνολικής προσπάθειας για Διαλειτουργικότητα υποστηρίζει τη χρησιμοποίηση Open Source Software όσον αφορά στην υλοποίηση εφαρμογών δημοσίου οι οποίες θα μπορούν να αναπτύσσονται, να επαναχρησιμοποιούνται και να διατίθενται μεταξύ των μελών της Ευρωπαϊκής Κοινότητας. Επιπλέον, προς την κατεύθυνση της Διαλειτουργικότητας, έχει δημιουργηθεί το European Interoperability Framework (EIF), ένα πλαίσιο σκοπός του οποίου είναι να θέσει τις βασικές αρχές και κατευθύνσεις για την επίτευξη διαλειτουργικότητας στα τρία επίπεδά της και να προτείνει τεχνικές πολιτικές και

πρότυπα ώστε να βοηθήσει τη διασύνδεση των δημόσιων διοικητικών πληροφοριακών συστημάτων σε όλη την επικράτεια της Ευρωπαϊκής Ένωσης.

Έτσι, γίνεται έρευνα για να βρεθούν τρόποι να αναπαρασταθεί και να δοθεί ένα πλαίσιο στην πληροφορία, έτσι ώστε να, μπορούν αυτοματοποιημένα εργαλεία να μοιράζονται και να επεξεργάζονται την πληροφορία αυτή με κοινό τρόπο, ακόμα και αν αυτά έχουν σχεδιαστεί ανεξάρτητα το ένα από το άλλο. Συνεπώς, στα πλαίσια των στόχων των υπηρεσιών ηλεκτρονικής διακυβέρνησης η σημασιολογική Διαλειτουργικότητα εκφράζει την ανάγκη για συμφωνία σε μία κοινή βάση όσον αφορά στα τμήματα δεδομένων που θα ανταλλάγουν σε πανευρωπαϊκή βάση.

Για την επίτευξη της σημασιολογικής διαλειτουργικότητας, το EIF προωθεί τον καθορισμό ενός κοινού συνόλου τμημάτων δεδομένων, των δομικών στοιχείων δεδομένων ηλεκτρονικής διακυβέρνησης (Core e-Government data elements) όπως για παράδειγμα τα βασικά εθνικά αναγνωριστικά για επιχειρήσεις, πολίτες και κυβερνήσεις, σε πανευρωπαϊκό επίπεδο. Απαραίτητη προϋπόθεση για τον καθορισμό αυτό είναι η υλοποίηση οργανωτικών και τεχνικών υποδομών. Αναγκαία απαίτηση για την ανταλλαγή πληροφοριών είναι η χρησιμοποίηση μίας ανεξάρτητης γλώσσας, η οποία θα καθιστά δυνατή την περιγραφή του νοήματος και της δομής των υποκείμενων δεδομένων, όπως για παράδειγμα μία mark-up γλώσσα.

Η γλώσσα αυτή που χρησιμοποιείται από το EIF είναι η XML. Επειδή όμως η XML από μόνη της δεν μπορεί να εξασφαλίσει σημασιολογική Διαλειτουργικότητα, γίνεται χρήση ειδικών δομών όπως τα XML Σχήματα (XML Schemas), τα οποία επιτρέπουν την ενοποίηση υπηρεσιών οι οποίες είχαν αναπτυχθεί με διαφορετικά λεξιλόγια και με διαφορετική οπτική όσον αφορά στα δεδομένα. Οι πρωτοβουλίες σε πανευρωπαϊκό επίπεδο για την ανάπτυξη κοινών σημασιολογικών δομών στη βάση της XML πρέπει να εκτελεστούν κατά ένα συντονισμένο τρόπο και πρέπει να λαμβάνουν υπόψιν τους τη συνεργασία και τη συμβατότητα με τα ήδη υπάρχοντα πρότυπα τυποποίησης.

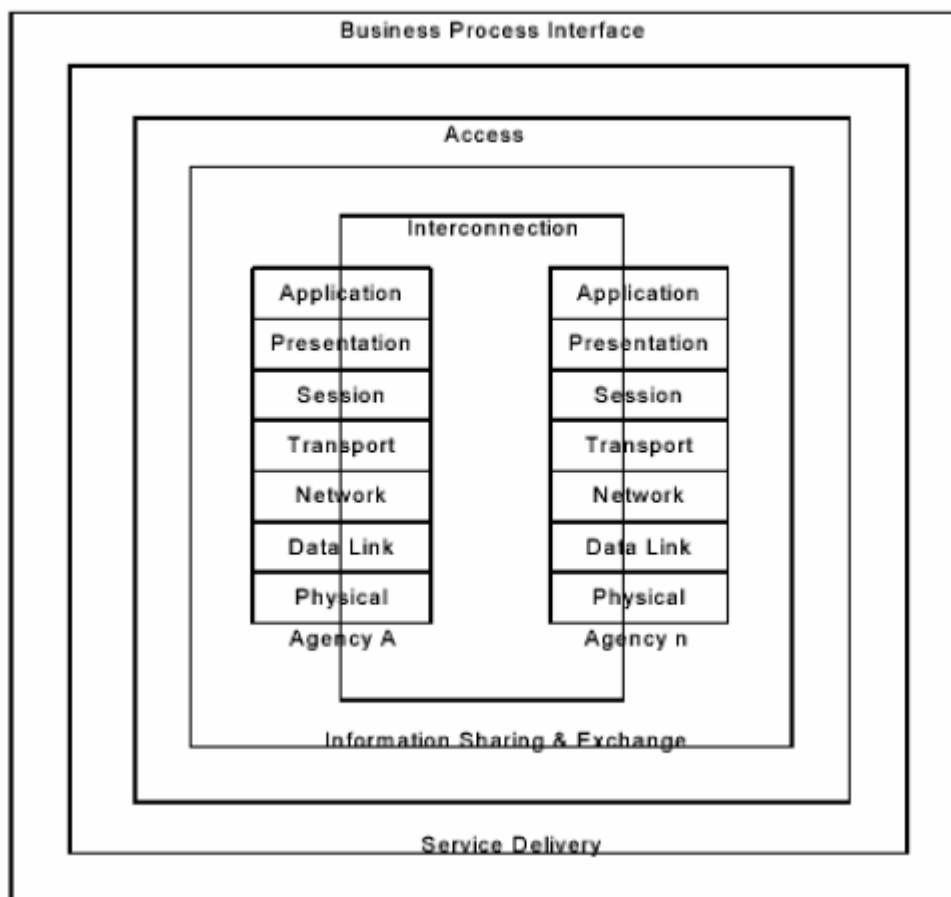
Έτσι, το EIF αναπτύσσει ειδικά λεξιλόγια XML, λαμβάνοντας υπόψιν τα ήδη συμφωνημένα Core/specific e-Government data elements, με βάση τα οποία δημιουργεί συγκεκριμένα ευρωπαϊκά σχήματα και ορισμούς τα οποία τα διαθέτει σε όλους τους πανευρωπαϊκούς φορείς μέσω κοινών υποδομών.

2.2.1.2 *Ηνωμένο Βασίλειο – UK Schema Library*

Η περισσότερο επιτυχημένη προσπάθεια επίτευξης διαλειτουργικότητας γίνεται από τη Μεγάλη Βρετανία με τη χρήση του πλαισίου UK e-GIF (Electronic Government Interoperability Framework) του οποίου τα αποτελέσματα έχουν ήδη αρχίσει να εφαρμόζονται στην κυβερνητική πολιτική του κράτους. Το UK e-GIF της Βρετανικής Κυβέρνησης έθεσε το πλαίσιο για την αποτελεσματική επικοινωνία σε επίπεδο δεδομένων μεταξύ κυβερνητικών οργανισμών και υπηρεσιών ώστε να μπορεί να υλοποιηθεί ενιαία πρόσβαση πολιτών, επιχειρήσεων και οργανισμών στις δημόσιες υπηρεσίες. Από άποψη τεχνολογικών προδιαγραφών, το e-GIF δίνει έμφαση σε τεχνολογίες Internet, World Wide Web (WWW) και XML, ενώ σκοπός του είναι η αποτελεσματικότερη επικοινωνία σε επίπεδο δεδομένων των κυβερνητικών οργανισμών, όπως επίσης και να διευκολύνει στους πολίτες αλλά και στις επιχειρήσεις την πρόσβασή τους στις δημόσιες υπηρεσίες.

Στα πλαίσια του UK e-GIF ορίζεται ένα ελάχιστο σύνολο από τεχνικούς κανόνες και προδιαγραφές, πάνω στις οποίες βασίζεται η ροή της πληροφορίας μεταξύ της κυβέρνησης και του δημόσιου τομέα. Το σύνολο αυτό κανόνων και προδιαγραφών τεκμηριώνεται στο εγχειρίδιο Technical Standards Catalogue (TSC), το οποίο έχει εκδοθεί με τη μέριμνα του UK e-GIF και παρέχει πρότυπα που καλύπτουν τη Διασύνδεση, την Ολοκλήρωση Δεδομένων και τη Διαχείριση Περιεχομένου Μεταδεδομένων. Επίσης παρέχει πρότυπα αναφορικά με τρόπους πρόσβασης σε ηλεκτρονικές υπηρεσίες όπως σταθμούς εργασίας (υπολογιστικά συστήματα), κινητά τηλέφωνα και τεχνολογίες VoIP (Voice over Internet Protocol), καθώς επίσης και αναφορικά με επιχειρηματικές περιοχές όπως το ηλεκτρονικό εμπόριο, η ηλεκτρονική μάθηση (e-learning) και η οικονομία (finance).

Στο σχήμα που ακολουθεί, περιγράφονται τα διάφορα επίπεδα στα οποία εκτείνεται η λειτουργικότητα του e-GIF πάνω από το γνωστό μοντέλο στρωμάτων του ISO.



Εικόνα 2.2-3: Μοντέλο Στρωμάτων του e-GIF.

2.2.1.3 Ηνωμένες Πολιτείες Αμερικής – NIEM (National Information Exchange Model)

Στις ΗΠΑ, οι προσπάθειες σε σχέση με τη Διαλειτουργικότητα εκτείνονται σε ένα μεγάλο αριθμό τομέων της Κυβέρνησης (State και Federal) και Ιδιωτικών Επιχειρήσεων. Αξίζει να σημειωθεί ότι η Διαλειτουργικότητα στα πλαίσια του e-Government είναι ένας από τους στρατηγικούς στόχους της Αμερικάνικης κυβέρνησης υποστηριζόμενη απ' ευθείας από το Πρόεδρο.

Στα πλαίσια των προσπαθειών για την επίτευξη διαλειτουργικότητας έχει δημιουργηθεί το NIEM (National Information Exchange Model). Το NIEM αποτελεί μία πρωτοβουλία που έχει εισαχθεί από κρατικές και ομοσπονδιακές υπηρεσίες, παρέχοντας τη θεμελίωση για την απρόσκοπτη ανταλλαγή πληροφοριών.

Η θεμελιώδης δομική μονάδα του NIEM είναι το data Component. Τα data Components είναι τα βασικά αντικείμενα επιχειρηματικών δεδομένων, τα οποία περιγράφουν κοινές πρακτικές χρησιμοποιούμενες γενικά σε επιχειρηματικές δραστηριότητες. Η πληροφορία που ανταλλάσσεται ανάμεσα σε υπηρεσίες μπορεί να διασπαστεί σε συνιστώσες (για παράδειγμα να διασπαστεί σε πληροφορία για

ανθρώπους, για μέρη, για αντικείμενα και για γεγονότα). Τα data Components, (στα πλαίσια της ανταλλαγής πληροφοριών), τα οποία είναι κοινώς διαμοιραζόμενα και αντιλαμβανόμενα μεταξύ όλων των τομέων, (Δικαιοσύνη, Διεθνές Εμπόριο κ.α.), ονομάζονται Global Components (για παράδειγμα Άτομο, Διεύθυνση, Οργανισμός), ενώ τα Components που χρησιμοποιούνται σε ανταλλαγές μεταξύ πολλαπλών τομέων αλλά δεν είναι καθολικώς διαμοιραζόμενα ονομάζονται Common Components (για παράδειγμα παράβαση, ποινή και διαθεσιμότητα).

Το NIEM παρέχει μία βιβλιοθήκη από επαναχρησιμοποιήσιμα αντικείμενα για τη δόμηση προτύπων (templates) πακέτων ανταλλαγής πληροφοριών στη μορφή των XML Σχημάτων, καθορίζοντας έτσι το περιεχόμενο της ανταλλαγής μηνυμάτων. Η ανταλλαγή πακέτων περιλαμβάνει την ανταλλαγή εγγράφων, για παράδειγμα, μία φόρμα συμπλήρωσης από αλλοδαπούς ή μία αναφορά καταγραφής γεγονότος, καθώς επίσης και τη δημιουργία ερωτημάτων (queries) και απαντήσεων (responses) σε μία βάση δεδομένων, για παράδειγμα μία αναζήτηση άδειας οχήματος σε μία βάση δεδομένων για κλεμμένα οχήματα. Τα επαναχρησιμοποιήσιμα αντικείμενα του NIEM μπορούν να χρησιμοποιηθούν και για δομημένα και για μη δομημένα ερωτήματα/απαντήσεις (queries/responses).

2.2.2 Ελλάδα – Πλαίσιο Ηλεκτρονικής Διακυβέρνησης (ΠΗΔ)

Οι επιδιώξεις λοιπόν για μια, σε παγκόσμιο επίπεδο, προσπάθεια εκσυγχρονισμού και ψηφιοποίησης των παρεχόμενων από το κράτος σε πολίτες και επιχειρήσεις υπηρεσιών, δεν θα μπορούσαν παρά να βρουν εφαρμογή και στον ελλαδικό χώρο. Συγκεκριμένα, για το λόγο αυτό τέθηκε σε εφαρμογή το Ελληνικό Πλαίσιο Παροχής Υπηρεσιών Ηλεκτρονικής Διακυβέρνησης και Πρότυπα Διαλειτουργικότητας (ή Πλαίσιο Ηλεκτρονικής Διακυβέρνησης – ΠΗΔ), το οποίο άλλωστε αποτελεί και τον ακρογωνιαίο λίθο της Ψηφιακής Στρατηγικής της Ελληνικής Δημόσιας Διοίκησης. Κύριο μέλημα του ΠΗΔ είναι η αποτελεσματική υποστήριξη υπηρεσιών Ηλεκτρονικής Διακυβέρνησης σε Τοπικό και Περιφερειακό επίπεδο και η Διαλειτουργικότητα σε επίπεδο πληροφοριακών συστημάτων, που θα οδηγήσει στην εξασφάλιση της ομοιογένειας των συστημάτων ώστε να είναι σε θέση να διαχειρίζονται με αποδοτικό τρόπο τις νέες Τεχνολογίες Πληροφορικής και Επικοινωνιών (ΤΠΕ). Το Πλαίσιο Παροχής Υπηρεσιών Ηλεκτρονικής Διακυβέρνησης απευθύνεται σε φορείς της Δημόσιας Διοίκησης, όπως Δήμους, Κοινότητες, Νομαρχίες και ΟΤΑ, σε εγχώριο επίπεδο καθώς και σε Κυβερνητικούς Οργανισμούς σε πανευρωπαϊκό και σε παγκόσμιο επίπεδο, καλύπτοντας

έτσι τον τύπο Υπηρεσιών Κυβέρνηση – Προς – Κυβέρνηση (Government – to – Government – G2G), στους πολίτες, καλύπτοντας έτσι τον τύπο Υπηρεσιών Κυβέρνηση – Προς – Πολίτη (Government – to – Citizen – G2C) και τέλος, στις επιχειρήσεις, καλύπτοντας έτσι τον τύπο Υπηρεσιών Κυβέρνηση – Προς – Επιχείρηση (Government – to – Business – G2B). Η συνολική αναπαράσταση των οντοτήτων στις οποίες απευθύνεται το ΠΗΔ, καθώς επίσης και ο τρόπος συσχέτισης με αυτές εμφανίζεται στην εικόνα 2.2-3 με λεζάντα «**Εφαρμογή του ΠΗΔ στις οντότητες που απευθύνεται.**»

Στον ελλαδικό χώρο, όπου γίνεται επίσης μία οργανωμένη προσπάθεια για την εφαρμογή ενός πλήρους, ολοκληρωμένου και λειτουργικού Προγράμματος, που θα αφορά στην Ηλεκτρονική Διακυβέρνηση, τις προσπάθειες επίτευξης του τομέα της Διαλειτουργικότητας τις έχει αναλάβει το ελληνικό e-GIF ή αλλιώς το Πλαίσιο Ηλεκτρονικής Διακυβέρνησης (ΠΗΔ) όπως είναι η πλήρης ονομασία του. Το ΠΗΔ έχει δεχθεί επιρροές από το Αγγλικό e-GIF (όπως άλλωστε έχουν κάνει και τα e-GIF της Αυστραλίας και της Νέας Ζηλανδίας) επειδή εκείνο παρέχει συμβατές με το Δημόσιο Ελληνικό Τομέα τεχνικές προδιαγραφές, που μπορούν να καλύψουν τις ελληνικές ανάγκες διαλειτουργικότητας, επειδή χαρακτηρίζεται από ομοιογένεια και απλότητα, αλλά και επειδή υπάρχουν αρκετά υψηλές πιθανότητες να αποτελέσει το κοινώς αποδεκτό πρότυπο διαλειτουργικότητας της Ευρωπαϊκής Ένωσης, παραγκωνίζοντας το IDABC. Το ΠΗΔ αποτελείται από τα εξής πλαίσια:

- Το Πλαίσιο Πιστοποίησης Δημόσιων Δικτυακών Τόπων. Σκοπός του πλαισίου αυτού, είναι η θέσπιση προτύπων και προδιαγραφών για την ανάπτυξη δημόσιων διαδικτυακών τόπων που αφορούν στην εικαστική παρουσίαση, στην κατηγοριοποίηση και διαχείριση περιεχομένου και στη διαθεσιμότητα και προσβασιμότητα του διαδικτυακού τύπου, καθώς επίσης και ο καθορισμός της μεθοδολογίας για τη διαχείριση και την εφαρμογή του Πλαισίου.
- Το Πλαίσιο Διαλειτουργικότητας και Υπηρεσιών Ηλεκτρονικών Συναλλαγών. Σκοπός του πλαισίου αυτού, είναι η ανάλυση και αποτύπωση της υφιστάμενης κατάστασης στον “Χάρτη Παρεχόμενων Υπηρεσιών” από τους Δημόσιους Φορείς προς τους Πολίτες και τις Επιχειρήσεις, η κατηγοριοποίηση των παρεχόμενων υπηρεσιών και εγγράφων σε πολλαπλές όψεις και η πρότυπη αποτύπωση κύριων υπηρεσιών προς πολίτες και επιχειρήσεις. Επίσης, το πλαίσιο αυτό στοχεύει στον καθορισμό τεχνικών προτύπων και προδιαγραφών που αφορούν στη μορφή των πληροφοριών, στον τρόπο ανταλλαγής των πληροφοριών (επικοινωνία / πρωτόκολλα), στον τρόπο πρόσβασης στις

πληροφορίες (ασφάλεια / έλεγχος πρόσβασης) και στον τρόπο αναζήτησης των πληροφοριών (υπηρεσίες καταλόγου).

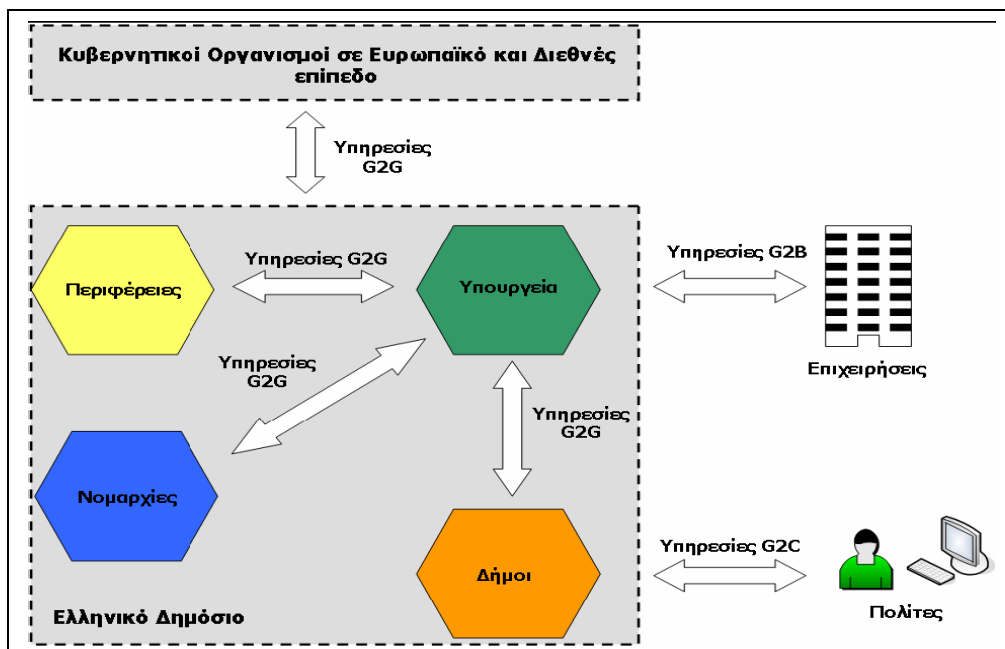
- Το Πλαίσιο Ψηφιακής Αυθεντικοποίησης. Σκοπός του πλαισίου αυτού, είναι ο καθορισμός των επιπέδων ιδιωτικότητας της πληροφορίας και των επιπέδων εμπιστοσύνης ανά είδος πληροφορίας και ανά τύπο υπηρεσίας, καθώς επίσης και η μελέτη επικινδυνότητας και αποτίμησης των επιπτώσεων στην ιδιωτικότητα, για ομάδες χρηστών και κατηγορίες πληροφορίας. Επιπλέον, στόχος του πλαισίου είναι ο σχεδιασμός Πολιτικής Αυθεντικοποίησης ανά τύπο υπηρεσίας και ανά ομάδα χρηστών και η θέσπιση ενός πλαισίου Πολιτικής Πιστοποιητικών και αναγνωριστικών Πολιτών και Επιχειρήσεων, όπως επίσης και η θέσπιση προτύπων και προδιαγραφών για την προστασία των δικαιωμάτων των πολιτών και την εξασφάλιση διαφάνειας στην επεξεργασία των δεδομένων.

Στα πλαίσια του ΠΗΔ έχει θεσπιστεί ακόμα το Μοντέλο Τεκμηρίωσης. Σε γενικές γραμμές, στόχος του Μοντέλου Τεκμηρίωσης είναι η τεκμηρίωση Διαδικασιών (ηλεκτρονικών ή μη), Web Services και Εγγράφων με σκοπό τη διευκόλυνση ανάπτυξης ελληνικών υποδομών Ηλεκτρονικής Διακυβέρνησης, που θα είναι σε θέση να προσφέρουν αποτελεσματικότερες και σε χαμηλότερο κόστος δημόσιες υπηρεσίες. Προς την κατεύθυνση επιδίωξης όλων αυτών των τεκμηρίωσης, το Μοντέλο Τεκμηρίωσης αποτελεί μία πρόταση τεκμηρίωσης για τις Διαδικασίες, τα Έγγραφα και τα XML Σχήματα που χρησιμοποιούνται από το ΠΗΔ για την κωδικοποίηση της πληροφορίας που μεταφέρεται μέσω των εγγράφων, ενώ θέτει και συμβάσεις ονοματολογίας για τις οντότητες που θα αποτελούν τα κωδικοποιημένα έγγραφα. Επομένως, τα μεταδεδομένα που θα χρησιμοποιηθούν για την αναπαράσταση της πληροφορίας των εγγράφων, θα πρέπει αφενός μεν να συμμορφώνονται στις αρχές και τις οδηγίες του προτύπου Dublin Core (ISO 15836) και αφετέρου στο Μοντέλο Τεκμηρίωσης του ΠΗΔ.

Αναφορικά με την παρούσα διπλωματική, οι ενότητες του Μοντέλου Τεκμηρίωσης που σχετίζονται με αυτήν, είναι η πρόταση τεκμηρίωσης εγγράφων και η πρόταση τεκμηρίωσης των XML Σχημάτων που πρόκειται να μοντελοποιηθούν και να σχεδιαστούν για τις ανάγκες του ΠΗΔ. Κατά συνέπεια, αρχικά γίνεται εφαρμογή του προτεινόμενου προτύπου (template) μοντελοποίησης των στοιχείων και των μεταδεδομένων του ζητούμενου εγγράφου και μετέπειτα γίνεται η μετατροπή των στοιχείων αυτών στα κατάλληλα XML Σχήματα, σύμφωνα με τις προδιαγραφές των XML Core Components, των XML Σχημάτων και των μεταδεδομένων που

καταγράφονται στο μοντέλο τεκμηρίωσης, σύμφωνα πάντα με τις συμβάσεις ονοματολογίας που προτείνονται από αυτό.

Το Μοντέλο Τεκμηρίωσης αποτελεί τη βάση στην οποία εκπονείται το Πλαίσιο Διαλειτουργικότητας και Υπηρεσιών Ηλεκτρονικών Συναλλαγών (ΠΔ&ΥΗΣ) και δομείται το Ληξιαρχείο Διαλειτουργικότητας του ΠΗΔ, κυρίως όσον αφορά στις Αποθήκες Διαδικασιών, Δεδομένων και Μεταδεδομένων που περιλαμβάνονται σε αυτό. Όσον αφορά στο Ληξιαρχείο Διαλειτουργικότητας, το Μοντέλο Τεκμηρίωσης συνεισφέρει στην αναπαράσταση του πραγματικού κόσμου, που εμπλέκει Φορείς, Υπηρεσίες, Φόρμες Δεδομένων και Πληροφοριακά Συστήματα με ακρίβεια και σαφήνεια, χωρίς ενδεχόμενες παραλείψεις ή παραδοχές.



Εικόνα 2.2-4: Εφαρμογή του ΠΗΔ στις οντότητες που απευθύνεται.

2.3 Ο Οργανισμός UN/CEFACT

Η Διαλειτουργικότητα και η επίτευξή της είναι βαρύνουσας σημασίας για μία επιτυχημένη εφαρμογή Ηλεκτρονικής Διακυβέρνησης. Για το σκοπό αυτό, ήταν επιβεβλημένη η σύσταση ενός οργανισμού σε παγκόσμιο επίπεδο, πρωταρχικός ρόλος του οποίου είναι να εισαγάγει ορισμένα κοινά και ευρέως αποδεκτά πρότυπα τεκμηρίωσης είτε διαδικασιών είτε εγγράφων είτε ονοματολογίας, τα οποία θα αποτελέσουν το αρχικό στάδιο και η περαιτέρω επεξεργασία τους θα προσαρμοστεί στις ανάγκες του κάθε κράτους ξεχωριστά. Ένας τέτοιος οργανισμός είναι ο UN /CEFACT

(United Nations Centre for Trade Facilitation and Electronic Business), κύριος στόχος του οποίου είναι η βελτίωση της ποιότητας των διαδικασιών που έχουν να κάνουν με επιχειρήσεις, το εμπόριο και τους δημόσιους διοικητικούς οργανισμούς, καθώς επίσης και η ανταλλαγή, με λειτουργικό και αποδοτικό τρόπο, πληροφοριών και δεδομένων στη βάση των σχετικών με αυτά παρεχόμενων υπηρεσιών. Το UN/CEFACT θεωρείται απόγονος του UN/EDIFACT και ουσιαστικά αποτελεί τη μετατροπή των EDI (Electronic Data Interchange) μηνυμάτων του UN/EDIFACT σε αντίστοιχα XML. Ικανοποιώντας τον αρχικό στόχο για τον οποίο και δημιουργήθηκε, ο UN/CEFACT διευκολύνει την ανάπτυξη επιχειρηματικών διαδικασιών, αλλά και διαδικασιών που έχουν να κάνουν με τη Δημόσια Διοίκηση, θεσπίζοντας πρότυπα που βρίσκουν εφαρμογή σε παγκόσμιο επίπεδο και συμβάλλουν στην απλοποίηση της μεταφοράς δεδομένων, ενώ βοηθούν στη μείωση του κόστους της και στην απαλλαγή, κατά μεγάλο ποσοστό, των χρηστών από τις γραφειοκρατικές διαδικασίες που διέπουν τις υπηρεσίες και ειδικά αυτές που έχουν να κάνουν με τη Δημόσια Διοίκηση. Ωστόσο, παρόλο που διευκολύνει την εφαρμογή διαδικασιών, επικεντρώνεται κυρίως στα έγγραφα και όχι στις διαδικασίες αυτές καθαυτές, ενώ χρησιμοποιεί τη γλώσσα XML για τη μετάδοση και την αποθήκευση των δεδομένων που απαρτίζουν τα έγγραφα αυτά. Για τη μεταφορά και τη μοντελοποίηση των διεργασιών προτείνει τη χρήση συγκεκριμένα του προτύπου ebXML (electronic business XML), το οποίο έχει τμηματική αρχιτεκτονική και δεν παρέχει άμεσα τις δομές των ηλεκτρονικών εγγράφων, αλλά επιτρέπει τον καθορισμό τους ανά περίπτωση.

2.3.1 CCTS (Core Component Technical Specification) και Core

Components

Ακρογωνιαίος λίθος της προσπάθειας προτυποποίησης που καταβάλλεται από τον Οργανισμό UN/CEFACT είναι η δημιουργία της μεθοδολογίας CCTS (Core Components Technical Specification). Η μέθοδος προτυποποίησης που εισάγεται από το CCTS καθορίζει μία νέα προσέγγιση στο πρόβλημα έλλειψης διαλειτουργικότητας μεταξύ των εφαρμογών που τοποθετούνται κατά κύριο λόγο στον επιχειρησιακό στίβο. Κύριος σκοπός του είναι η δημιουργία ευέλικτων προτύπων για την προτυποποίηση της επικοινωνίας υπηρεσιών και εφαρμογών.

Αυτή τη μεθοδολογία CCTS την πετυχαίνει κάνοντας χρήση των Core Components. Τα Core Components είναι δομικά συστατικά τα οποία έχουν τη δική τους αυτόνομη σημασιολογία και λειτουργικότητα και είναι σε θέση να αντιπροσωπεύουν τις διάφορες οντότητες που συμμετέχουν κατά την ανταλλαγή μίας πληροφορίας ή κατά την εφαρμογή κάποιας υπηρεσίας με τέτοιο τρόπο ώστε να εξασφαλίζουν σε μεγάλο βαθμό τόσο την επιζητούμενη Διαλειτουργικότητα όσο και την ευελιξία των πληροφοριακών συστημάτων που αλληλεπιδρούν. Η μεθοδολογία που βασίζεται στα Core Components μπορεί να χρησιμοποιηθεί για την ανάπτυξη ενός κοινού συνόλου σημασιολογικών δομικών μονάδων που θα αντιπροσωπεύουν σε ένα γενικότερο αφαιρετικό επίπεδο τους γενικούς τύπους δεδομένων που ανταλλάσσονται στο πλαίσιο επιχειρηματικών διαδικασιών. Επομένως, με τη σωστή και αποτελεσματική χρήση των Core Components, σημαντικές βελτιώσεις μπορούν να επιτευχθούν στον τομέα της ευέλικτης επεξεργασίας δεδομένων όπως επίσης και σε εκείνον της διαλειτουργικότητας μεταξύ των δεδομένων και οι βελτιώσεις αυτές μπορούν να επιτευχθούν διαμέσου της κατανομημένης μοντελοποίησης των επιχειρησιακών μοντέλων δεδομένων, της κοινής απεικόνισης των μοντέλων δεδομένων που χρησιμοποιούνται σε επιχειρησιακές εφαρμογές και τέλος διαμέσου της υιοθέτησης και ταξινόμησης των δομικών μονάδων σχετικά με το συγκεκριμένο περιβάλλον στο οποίο δραστηριοποιούνται.

Ωστόσο, τα Core Components δεν χρησιμοποιούνται αυτούσια για την αναπαράσταση της πληροφορίας που εμπεριέχεται στα έγγραφα που μοντελοποιούνται και που αφορούν επιχειρησιακούς τομείς. Το ρόλο αυτό αναλαμβάνουν κάποια άλλα δομικά συστατικά τα οποία θα μπορούσε να θεωρηθεί πως αποτελούν συγκεκριμένα «στιγμιότυπα» των Core Components, ονομάζονται Business Information Entities (BIEs) και βρίσκονται σε άμεση συσχέτιση με το επιχειρησιακό περιβάλλον (Context) στο οποίο

εντάσσονται τα έγγραφα. Τα συγκεκριμένα αυτά Δομικά Συστατικά, δεν είναι προκαθορισμένα και δεν προϋπάρχουν, παρά μόνο δημιουργούνται την εκάστοτε στιγμή ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής. Αντιθέτως, τα Core Components είναι συγκεκριμένα και ορισμένα εκ των προτέρων και βρίσκονται αποθηκευμένα στο σύνολό τους στην CCL (Core Components Library) στο Ληξιαρχείο Διαλειτουργικότητας και αποτελούν και τη βάση για τη δημιουργία των κατά περίπτωση BIEs.

Τα Core Components χωρίζονται σε τέσσερις κατηγορίες ανάλογα με τη δομή και τη σημασιολογία τους:

1. Τα Συγκεντρωτικά Δομικά Συστατικά (Aggregate Core Component – ACC)
2. Τα Βασικά Δομικά Συστατικά (Basic Core Component – BCC)
3. Τα Σύνθετα Δομικά Συστατικά (Association Core Component – ASCC)
4. Οι Τύποι Δομικών Συστατικών (Core Component Type).

Οι κατηγορίες αυτές αναλύονται ως εξής:

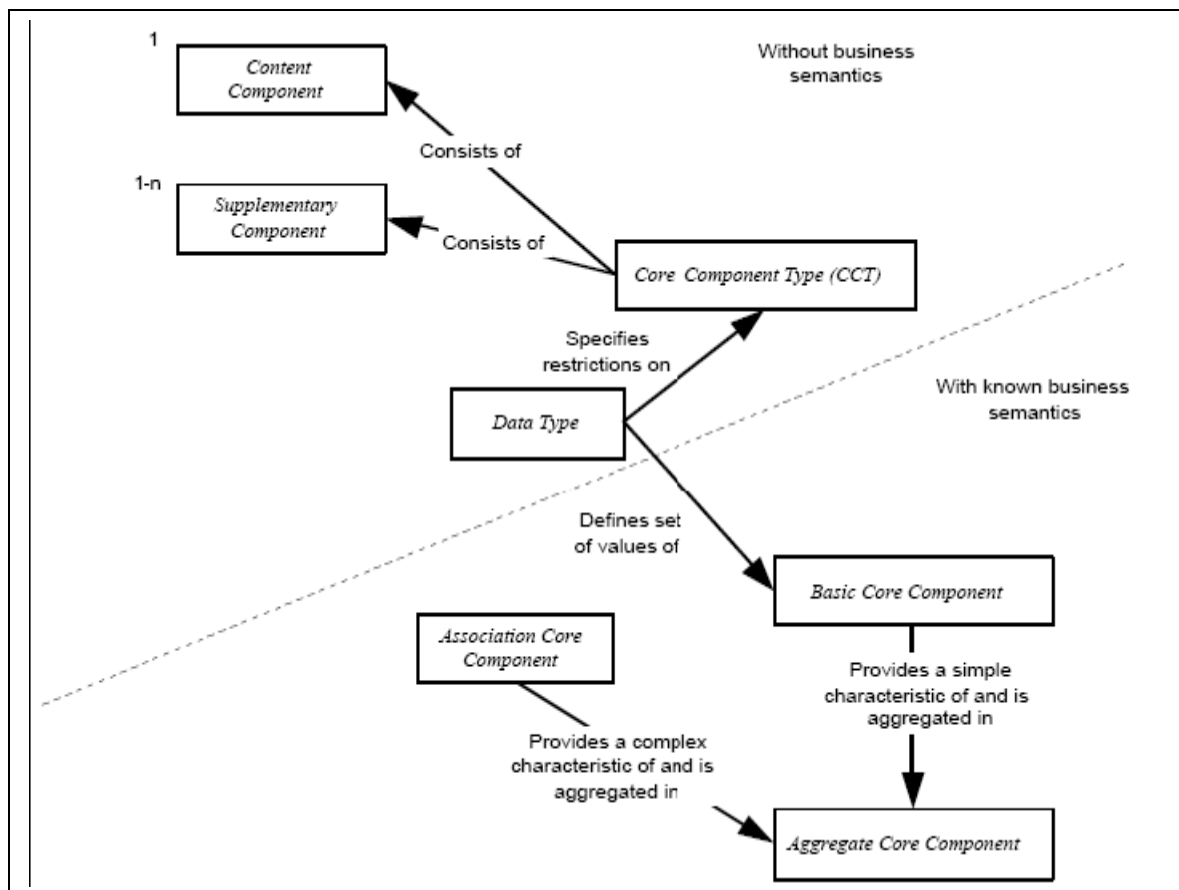
1. Συγκεντρωτικά ονομάζονται τα Δομικά Συστατικά εκείνα που αποτελούν ένα σύνολο δεδομένων που σχετίζονται νοηματικά μεταξύ τους και που όλα μαζί αναπαριστούν μία επιχειρησιακή πληροφορία. Σε επίπεδο μοντελοποίησης, θα μπορούσαμε να θεωρήσουμε πως τα ACC αναπαριστούν κλάσεις αντικειμένων, ανεξάρτητες από την ένταξή τους σε οποιοδήποτε επιχειρηματικό περιβάλλον.
2. Βασικά ονομάζονται τα Δομικά Συστατικά εκείνα τα οποία αναπαριστούν τα απλά χαρακτηριστικά από τα οποία αποτελείται ένα Συγκεντρωτικό Δομικό Συστατικό. Τα BCCs ορίζονται κατά μοναδικό τρόπο όσον αφορά στην επιχειρηματική τους σημασιολογία. Επιπλέον τα Βασικά Δομικά Συστατικά προσομοιώνουν τις ιδιότητες (Properties) που χαρακτηρίζουν τα αντίστοιχα ACCs των οποίων και αποτελούν μέρος, γι' αυτό και χαρακτηρίζονται από προκαθορισμένους τύπους δεδομένων (Data Types), οι οποίοι καθορίζουν το σύνολο των επιτρεπόμενων τιμών που είναι σε θέση να πάρουν αυτές οι ιδιότητες.
3. Σύνθετα ονομάζονται τα Δομικά Συστατικά εκείνα που αναπαριστούν τα σύνθετα χαρακτηριστικά από τα οποία συντίθεται ένα Συγκεντρωτικό Δομικό Συστατικό. Και

αυτά όπως και τα Βασικά Δομικά Συστατικά ορίζονται κατά μοναδικό τρόπο αναφορικά με την επιχειρηματική τους σημασιολογία. Σε αντίθεση όμως με τα BCCs, είναι συνδεδεμένα με κάποιο άλλο Συγκεντρωτικό Δομικό Συστατικό και το Δομικό Συστατικό αυτό είναι που καθορίζει τη δομή του Σύνθετου Δομικού Συστατικού.

4. Τέλος, έχουμε τους Τύπους Δομικών Συστατικών. Κάθε Τύπος Δομικών Συστατικών αποτελείται από ένα και μόνο ένα Συστατικό Περιεχομένου (Content Component), το οποίο είναι αυτό που μεταφέρει το κυρίως περιεχόμενο της πληροφορίας που μεταφέρει με μοναδικό τρόπο, ενώ υπάρχουν επίσης και μηδέν ή περισσότερα Συμπληρωματικά Συστατικά (Supplementary Components) τα οποία μεταφέρουν όποια επιπλέον λεπτομέρεια απαιτείται για το σαφή προσδιορισμό της πληροφορίας.

Θεμελιώδης έννοια όσον αφορά στα Core Components είναι και αυτή των Τύπων Δεδομένων (Data Types) που προαναφέρθηκε. Τα Data Types λοιπόν απεικονίζουν το σύνολο των τιμών που επιτρέπεται να χρησιμοποιηθεί για την απεικόνιση μίας ιδιότητας των Δομικών Συστατικών. Ένας Τύπος Δεδομένων πρέπει απαραίτητως να βασίζεται σε κάποιον Τύπο Δομικών Συστατικών, αλλά μπορεί να προκύπτει ύστερα από επιβολή περιορισμών είτε στο Συστατικό Περιεχομένου (Content Component) είτε σε κάποιο από τα Συμπληρωματικά Συστατικά (Supplementary Components) του CCT από τον οποίο προέρχεται. Οι Τύποι Δεδομένων χρησιμοποιούνται για να καθορίσουν το σύνολο των τιμών που λαμβάνει ένα Βασικό Δομικό Συστατικό ή μία Βασική Επιχειρηματική Οντότητα (BIE), η έννοια της οποίας θα αναλυθεί παρακάτω.

Στο σχήμα που ακολουθεί, απεικονίζεται πλήρως η δομή των Core Components, οι σχέσεις που συνδέουν τα Core Components μεταξύ τους, καθώς και ο τρόπος με τον οποίο συσχετίζονται το ένα με το άλλο:



Εικόνα 2.3-5: Δομή των Core Components

Για παράδειγμα, μία Κατάσταση (Status) αναπαρίσταται από ένα Aggregate Core Component που αποτελείται από μία σειρά ιδιοτήτων για την αναπαράσταση δεδομένων σε κατώτερο επίπεδο, όπως το Αναγνωριστικό Κατάστασης (Status ID), το οποίο χαρακτηρίζει κατά μοναδικό τρόπο μία συγκεκριμένη κατάσταση, το Όνομα Κατάστασης (Status Name), το οποίο περιγράφει το όνομα της συγκεκριμένης κατάστασης και την Ένδειξη Κατάστασης (Status Indicator), η οποία περιγράφει μία ένδειξη για τη συγκεκριμένη κατάσταση. Οι ιδιότητες αυτές, δεν μπορούν να αναλυθούν περαιτέρω, γι' αυτό και αποτελούν Basic Core Components. Το BCC Status Name μπορεί να έχει ως Τύπο Δεδομένων τη Συμβολοσειρά (String). Εκτός όμως από αυτές τις ιδιότητες, υπάρχουν και άλλες, όπως η Περίοδος Ισχύος Κατάστασης (Status Validity Period), η οποία υποδηλώνει την περίοδο κατά την οποία η συγκεκριμένη κατάσταση θεωρείται ότι βρίσκεται σε ισχύ και η Τοποθεσία Σχετιζόμενη με την Κατάσταση (Status Specified Location), η οποία περιγράφει μία τοποθεσία που σχετίζεται με τη συγκεκριμένη κατάσταση. Οι δύο αυτές ιδιότητες θεωρούνται ως Association Core Components επειδή μπορούν να αναλυθούν σε άλλες χαμηλότερου επιπέδου οντότητες.

Για παράδειγμα, η Status Validity Period μπορεί να αναλυθεί σε Ημερομηνία Έναρξης Περιόδου (Period Start Date Time), σε Ημερομηνία Λήξης Περιόδου (Period End Date Time) και σε άλλες οντότητες. Εάν θέλαμε να ορίσουμε τους τύπους των δύο αυτών ASCCs, θα λέγαμε πως το πρώτο είναι του τύπου Period, ενώ το δεύτερο είναι του τύπου Location. Βλέπουμε λοιπόν πως τα Σύνθετα Δομικά Συστατικά λαμβάνουν τον τύπο τους από το έτερο Συγκεντρωτικό Δομικό Συστατικό που συσχετίζονται με το αρχικό Συγκεντρωτικό Δομικό Συστατικό, του οποίου και αποτελούν ιδιότητα. Όλα τα συστατικά που αποτελούν ένα Συγκεντρωτικό Δομικό Συστατικό, είτε αυτά είναι Βασικά Δομικά Συστατικά είτε αυτά είναι Σύνθετα Δομικά Συστατικά και προέρχονται από κάθε περιβάλλον στο οποίο θα μπορούσε να δραστηριοποιηθεί το συγκεκριμένο Συγκεντρωτικό Δομικό Συστατικό καθώς και ο τρόπος με τον οποίο διασυνδέονται με αυτό, βρίσκονται στη Βιβλιοθήκη Δομικών Συστατικών (Core Component Library – CCL). Μέσα στη CCL υπάρχουν αποθηκευμένα αναλυτικά όλα τα Συγκεντρωτικά Δομικά Συστατικά που αναπαριστούν τις εμπλεκόμενες οντότητες κατά τη μεταφορά κάποιας πληροφορίας, ενώ παρατίθενται λεπτομερώς και όλα τα BCCs και ASCCs από τα οποία συντίθενται.

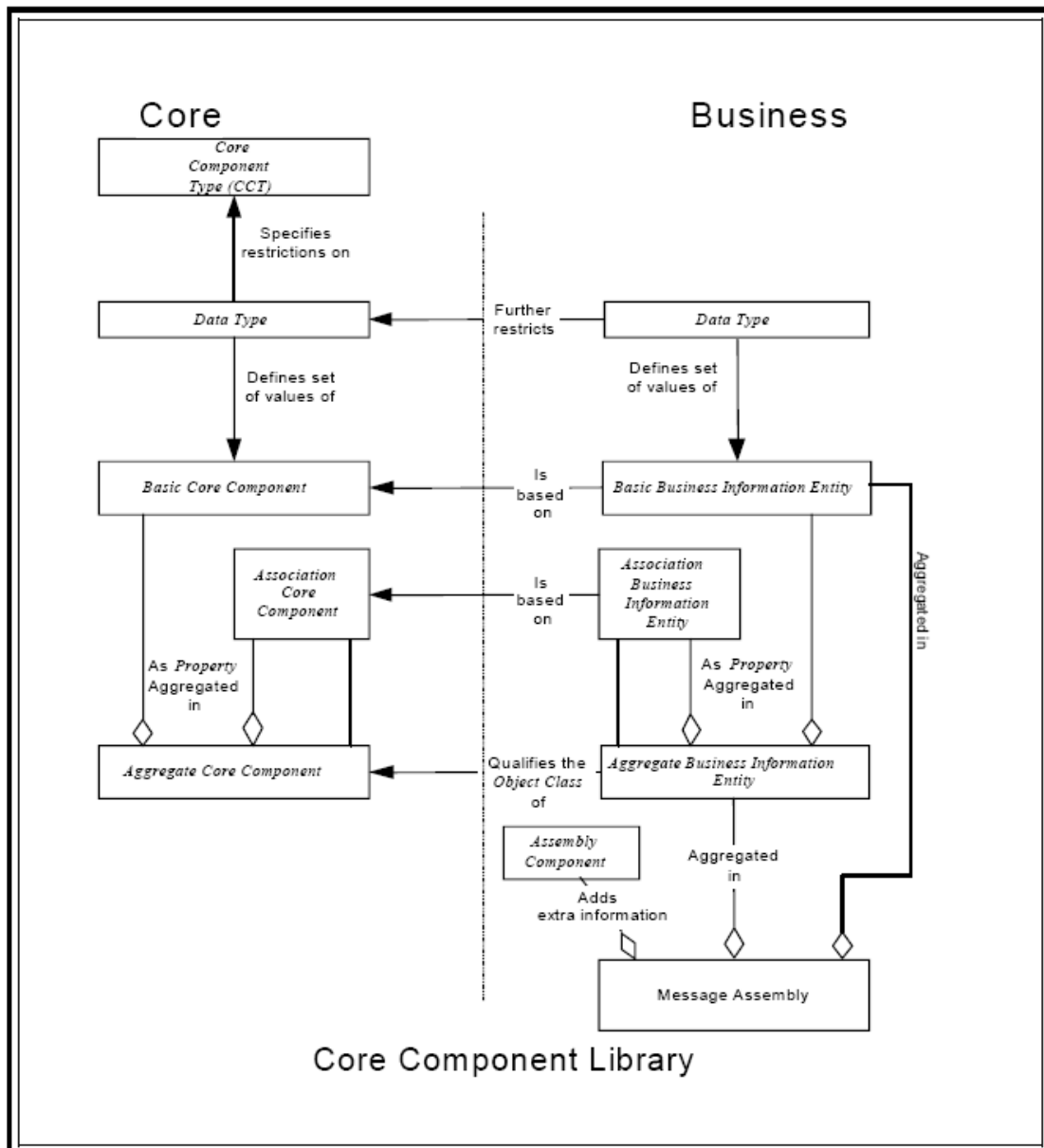
2.3.2 Business Information Entities

Όπως έχει ήδη αναφερθεί, τα Core Components δεν μπορούν να χρησιμοποιηθούν αυτούσια για την αναπαράσταση μίας πληροφορίας που εμπεριέχεται στα προς μοντελοποίηση έγγραφα και που αφορούν επιχειρησιακούς τομείς. Για το ρόλο αυτό έχουν δημιουργηθεί κάποια άλλα δομικά συστατικά τα οποία κατά κάποιο τρόπο αποτελούν συγκεκριμένα “στιγμιότυπα” των Core Components ανάλογα με το συγκεκριμένο περιβάλλον στο οποίο αναφερόμαστε, τα οποία ονομάζονται Business Information Entities (BIEs) και βρίσκονται σε άμεση συσχέτιση με το επιχειρησιακό περιβάλλον (Context) στο οποίο εντάσσονται οι προς μοντελοποίηση διεργασίες και υπηρεσίες. Υπάρχουν τριών ειδών BIEs: Το Aggregate Business Information Entity (ABIE), το Basic Business Information Entity (BBIE) και το Association Business Information Entity (ASBIE). Τα BIEs αυτά βρίσκονται σε πλήρη αντιστοιχία με τα Aggregate Core Component, Basic Core Component και Associative Core Component αντίστοιχα. Κάθε Business Information Entity προκύπτει από το αντίστοιχο του Core Component, εάν τοποθετηθεί το Core Component αυτό στο συγκεκριμένο περιβάλλον στο οποίο τοποθετείται η εργασία μας και διατηρηθούν μόνο τα Basic και Association Core Components εκείνα που είναι σχετικά με το περιβάλλον αυτό. Έτσι δημιουργούνται

38

τα Basic και Association Information Entities του συγκεκριμένου Aggregate Business Information Entity, τα οποία μπορούν να ενισχυθούν σημασιολογικά προσθέτοντάς τους ορισμένους χαρακτηρισμούς (qualifiers).

Επιστρέφοντας στο προηγούμενο παράδειγμα, παίρνοντας το Core Component Status και τοποθετώντας το στο Εργασιακό Περιβάλλον, παίρνουμε το Business Information Entity Working Status. Έτσι λοιπόν, αφού εφαρμοστεί στο BCC ID το qualifier Working Department προκύπτει πλέον το BBIE Working Department ID, το οποίο εκφράζει τον αριθμό του παραρτήματος της επιχείρησης όπου εργάζεται ο εργαζόμενος. Επιπλέον, αφού εφαρμοστεί στο BCC Indicator το qualifier Every Workday προκύπτει πλέον το BBIE Every Workday Condition Indicator, το οποίο δηλώνει εάν ο εργαζόμενος έχει εργαστεί κατά τη διάρκεια όλων των εργάσιμων ημερών. Δουλεύοντας κατ' αντιστοιχία και όσον αφορά στα ASCCs, αφού εφαρμοστεί στο ASCC Validation Period το qualifier Working προκύπτει το ASBIE Working Validation Period, το οποίο δηλώνει την περίοδο απασχόλησης ενός εργαζομένου. Γίνεται αντιληπτό λοιπόν, πως κάθε Business Information Entity πρέπει να προέρχεται από την εφαρμογή περιορισμών σε κάποιο Core Component και πως είναι άρρηκτα συνδεδεμένο με το αντίστοιχο Core Component από το οποίο προέρχεται και με το περιβάλλον στο οποίο τοποθετείται. Γι' αυτό εξάλλου και τα Aggregate Business Information Entities δεν προϋπάρχουν (σε αντίθεση με τα Aggregate Core Components τα οποία προϋπάρχουν και βρίσκονται στην Core Component Library), αλλά δημιουργούνται κάθε φορά εκ νέου, ανάλογα με το περιβάλλον στο οποίο τοποθετούνται και ανάλογα με τις απαιτήσεις του χρήστη αναφορικά με τη μοντελοποίηση. Στο παρακάτω σχήμα απεικονίζονται οι συσχετίσεις που υπάρχουν μεταξύ των Business Information Entities και των Core Components από τα οποία προκύπτουν.



Εικόνα 2.3-6: Σχέσεις μεταξύ Core Components και Business Information Entities.

2.4 XML

Η XML (eXtensible Markup Language) είναι μία δηλωτική γλώσσα, η οποία προέρχεται από την SGML (Standard Generalized Markup Language), από την οποία έχει κληρονομήσει την ισχύ και την επεκτασιμότητά της όντας ωστόσο πιο κανονικοποιημένη και πολύ πιο εύχρηστη. Μια θεμελιώδης διαφορά τους είναι το γεγονός πως ενώ η SGML χρησιμοποιείται κυρίως για τεχνική τεκμηρίωση και σε μικρότερο ποσοστό για δεδομένα διαφορετικού είδους, με την XML συμβαίνει ακριβώς το αντίθετο. Η XML εμφανίστηκε στο προσκήνιο ύστερα από την ASN.1 (Abstract

Syntax Notation One). Κύρια διαφορά μεταξύ των δύο αυτών προτύπων είναι πως η ASN.1 χρησιμοποιεί bits για την αναπαράστασή της (είναι bit coded), σε αντίθεση με την XML η οποία είναι μία γλώσσα η οποία βασίζεται αποκλειστικά σε χαρακτήρες (character based). Από το Φεβρουάριο του 1998 αποτελεί σύσταση του W3C (World Wide Web Consortium).

Η XML δεν είναι μία γλώσσα προγραμματισμού, είναι μία γλώσσα που χρησιμοποιείται για τη δόμηση και τη μεταφορά δεδομένων. Παρά τις αρκετές ομοιότητες της με την HTML (Hyper Text Markup Language) θα ήταν εσφαλμένο να τις ταυτίσουμε. Η XML δεν αποτελεί ένα πρότυπο που προορίζεται για την αντικατάσταση της HTML. Για την ακρίβεια, XML και HTML δημιουργήθηκαν με διαφορετικούς στόχους. Η HTML σχεδιάστηκε με στόχο την απεικόνιση των δεδομένων, επικεντρώνοντας στο πως εμφανίζονται τα δεδομένα. Από την άλλη, η XML σχεδιάστηκε με στόχο τη μεταφορά και την αποθήκευση των δεδομένων, επικεντρώνοντας στη δομή των δεδομένων. Μία επιπλέον διαφορά μεταξύ των δύο γλωσσών είναι αυτή που αφορά στο περιεχόμενο των ετικετών (tags) τους. Οι ετικέτες αυτές εξάλλου είναι που προσδίδουν τον όρο Markup στην ονομασία των XML. Οι ετικέτες είναι λέξεις μέσα σε γωνιακές αγκύλες ('<' και '>') και μέσα στις ετικέτες αυτές, περικλείονται οντότητες οι οποίες κωδικοποιούν μία περιγραφή της διάταξης και της λογικής δομής του εγγράφου. Μέσα στις ετικέτες μπορεί επιπλέον να περικλείονται γνωρίσματα (attributes), τα οποία είναι της μορφής: όνομα = "τιμή" (name = "value"). Οι οντότητες λοιπόν που περικλείονται μέσα στις ετικέτες, στην περίπτωση των HTML εγγράφων είναι συγκεκριμένες και προκαθορισμένες και συχνά προσδιορίζουν πως θα εμφανιστεί σε κάποιο φυλλομετρητή (browser) το δεδομένο που βρίσκεται μεταξύ ενός ζεύγους opening και closing tags. Αντιθέτως, η XML χρησιμοποιεί ετικέτες (tags) μόνο και μόνο για να οριοθετήσει τα δεδομένα, αφήνοντας την ερμηνεία τους στην εκάστοτε εφαρμογή που χρησιμοποιείται για να τα διαβάσει και να τα επεξεργαστεί. Αυτό έχει ως αποτέλεσμα οι οντότητες που βρίσκονται μέσα στις ετικέτες των εγγράφων XML να μην είναι προκαθορισμένες, αλλά να ορίζονται από το δημιουργό του εγγράφου ανάλογα με τον τρόπο με τον οποίο θέλει να ομαδοποιήσει τα δεδομένα του.

Ένα παράδειγμα XML εγγράφου το οποίο θα βοηθήσει στην απεικόνιση της δομής που χρησιμοποιείται από την XML, καθώς επίσης και στην περαιτέρω κατανόηση των χαρακτηριστικών γνωρισμάτων της XML παρατίθεται στην παρακάτω εικόνα:

```
<? xml version="1.0" encoding="ISO-8859-1"?>
<!-- An XML example -->
<shopping_mall
  xmlns: ellgram="http://www.ellinika_grammata.gr"
  xmlns: eleft="http://www.elefteroudakis.gr">
  <shoe_store name="NineWest">
    <shoe_1 type="high_hills" avail_num="42" price="120$"/>
    <shoe_2>
      <type> athletic </type>
      <avail_num> 40 </avail_num>
      <price> 80$ </price>
    </shoe_2>
  </shoe_store>
  <ellgram: bookstore>
    <ellgram name="Bible"/>
    <ellgram category="historical"/>
    <ellgram price="25$"/>
    <availability> This book is no longer available </availability>
  </ellgram: bookstore>
  <eleft: bookstore>
    <eleft name="The Shining"/>
    <eleft category="fiction"/>
    <eleft price="25$"/>
    <availability> Three copies available </availability>
  </eleft: bookstore>
</shopping_mall>
```

Εικόνα 2.4-7: Παράδειγμα XML εγγράφου.

Το παραπάνω αποτελεί ένα τυπικό παράδειγμα εγγράφου XML, στο οποίο εμφανίζονται τα χαρακτηριστικά γνωρίσματα της XML. Καταρχάς, είναι προφανής η δενδρική δομή που διέπει ένα έγγραφο XML και στην οποία θα αναφερθούμε αναλυτικά παρακάτω. Στην πρώτη σειρά του εγγράφου έχουμε την προαιρετική δήλωση της XML, η οποία καθορίζει αφενός την έκδοση της XML (στη συγκεκριμένη περίπτωση έχουμε χρήση της έκδοσης 1.0) και αφετέρου την κωδικοποίηση που χρησιμοποιείται και αφορά στους χαρακτήρες του εγγράφου (στη συγκεκριμένη περίπτωση έχουμε την κωδικοποίηση ISO-8859-1, η οποία αντιστοιχεί στο σύνολο χαρακτήρων Latin-1/West European). Έπειτα ακολουθεί η χρήση ενός XML σχολίου. Τα σχόλια στην XML ξεκινούν με τους χαρακτήρες “<!--“ και τελειώνουν με τους χαρακτήρες “-->”. Μπορούν να τοποθετηθούν σε οποιοδήποτε σημείο του εγγράφου και με την παρουσία τους διευκολύνουν στην κατανόηση και στην επεξήγηση ορισμένων σημείων του εγγράφου που υπό άλλες συνθήκες θα ήταν δυσνόητα. Επιπλέον, προσθέτουν περαιτέρω πληροφορίες αναφορικά με το έγγραφο. Τα σχόλια και το περιεχόμενό τους αγνοούνται από τον XML Validator, το πρόγραμμα που σκανάρει το έγγραφο με σκοπό να ελέγξει την ύπαρξη συντακτικών σφαλμάτων που δεν συνάδουν με την προτυποποίηση που έχει θεσπίσει το W3C, κατά τη σάρωση του εγγράφου.

Κύριο χαρακτηριστικό ενός XML εγγράφου είναι η δενδρική δομή που το διέπει. Κάθε έγγραφο περιλαμβάνει ένα στοιχείο ρίζα (root element), το οποίο περιλαμβάνει όλα τα υπόλοιπα στοιχεία, που ονομάζονται στοιχεία παιδιά (child elements). Όλα τα στοιχεία στην XML περικλείονται σε ζεύγη ετικετών (opening και closing tags) και ανάμεσα στο ζεύγος ετικετών που περικλείει κάθε στοιχείο βρίσκεται το περιεχόμενο του στοιχείου αυτού, το οποίο μπορεί να είναι είτε ένα σύνολο χαρακτήρων είτε κάποιο άλλο στοιχείο, προσδίδοντας έτσι στη δομή του εγγράφου ένα επιπλέον επίπεδο εμφώλευσης. Θα πρέπει ωστόσο, για κάθε opening tag ενός στοιχείου να υπάρχει και το αντίστοιχο closing tag του στοιχείου αυτού, διότι σε διαφορετική περίπτωση θα έχουμε την εμφάνιση συντακτικού σφάλματος. Στο σημείο αυτό αξίζει να αναφερθεί πως στη δήλωση XML που έχουμε στην αρχή του εγγράφου δεν έχουμε κάποιο αντίστοιχο closing tag, γεγονός το οποίο αποτελεί εξαίρεση. Θα πρέπει λοιπόν όλα τα στοιχεία να είναι ορθώς εμφωλευμένα το ένα μέσα στο άλλο και να ακολουθούν μία δομή του τύπου:

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Εικόνα 2.4-8: Εμφώλευση

Το opening tag ενός στοιχείου είναι δυνατόν να περιλαμβάνει και ορισμένες ιδιότητες (attributes) που αφορούν στο στοιχείο αυτό και οι οποίες εμφανίζονται πάντα στη μορφή: όνομα = "τιμή". Ένα τέτοιο attribute στο παράδειγμά μας έχουμε για το στοιχείο "shoe_store", όπου κατά τον ορισμό του έχουμε το attribute: name = "NineWest", που μας δίνει πληροφορία σχετικά με το όνομα του συγκεκριμένου καταστήματος υποδημάτων. Εξετάζοντας το στοιχείο "shoe_store" διαπιστώνουμε πως αυτό αποτελείται από δύο child attribute, τα "shoe_1" και "shoe_2". Παρατηρούμε λοιπόν πως οι πληροφορίες που αφορούν στο child element "shoe_1" παρατίθενται με τη μορφή ιδιοτήτων (attributes) του στοιχείου αυτού, ενώ αντίθετα οι πληροφορίες που αφορούν στο child element "shoe_2" παρατίθενται με τη μορφή subchild attributes του στοιχείου αυτού. Βλέπουμε λοιπόν πόσο ευέλικτη γλώσσα είναι η XML, αφού μας επιτρέπεται να περιγράψουμε πληροφορίες για τα στοιχεία του XML εγγράφου μας με περισσότερους από ένα συντακτικά ισοδύναμους τρόπους. Θα πρέπει ωστόσο να αναφερθεί πως από τους δύο προαναφερθέντες τρόπους παράθεσης πληροφοριών για ένα στοιχείο στις περισσότερες των περιπτώσεων προτιμότερος είναι ο τρόπος με τη χρήση των subchild elements, διότι τα attributes, σε αντίθεση με τα elements, δεν μπορούν να περιέχουν ούτε πολλαπλές τιμές αλλά ούτε και δενδρικές δομές (για να επιτύχουμε το nesting) και επιπλέον τα attributes δεν είναι εύκολα επεκτάσιμα και ευέλικτα στην επεξεργασία τους για αλλαγές στο μέλλον.

Στην XML, εξαιτίας του γεγονότος πως τα ονόματα των στοιχείων που περικλείονται στα tags δεν είναι προκαθορισμένα (σε αντίθεση με την HTML), αλλά ορίζονται από το δημιουργό του κάθε XML εγγράφου, στις περιπτώσεις που είναι αναγκαίο στα πλαίσια μίας εφαρμογής να συνδυαστούν XML έγγραφα που έχουν δημιουργηθεί από διαφορετικούς χρήστες έχουμε το πρόβλημα σύγκρουσης ονομάτων (name conflict). Έχουμε λοιπόν τη χρήση του ίδιου ονόματος για στοιχεία τα οποία αντιπροσωπεύουν διαφορετικές οντότητες. Για να αποφύγουμε τη σύγκρουση ονομάτων

που προκύπτει από μία τέτοια κατάσταση κάνουμε χρήση του χαρακτηριστικού της XML που ονομάζεται “χώροι ονομάτων” (namespaces). Τα namespaces αντιστοιχίζουν στο κάθε στοιχείο ένα χώρο ονομάτων μοναδικό για το στοιχείο αυτό προκειμένου να το διαφοροποιήσουν από τα υπόλοιπα στοιχεία με το ίδιο όνομα. Ο συγκεκριμένος αυτός χώρος ονομάτων συμβολίζεται με ένα πρόθεμα το οποίο ισούται με το Uniform Resource Identifier (URI) που περιγράφει αυτό το namespace και μπορεί να έχει είτε τη μορφή ενός Uniform Resource Locator (URL) είτε τη μορφή ενός Uniform Resource Name (URN). Στο παράδειγμά μας ως namespaces ορίζονται τα URLs http://www.ellinika_grammata.gr και “<http://www.elefteroudakis.gr>” και αντιστοιχίζονται στα προθέματα χώρου ονομάτων “ellgram” και “eleft” αντίστοιχα.

2.4.1 Ορθότητα XML εγγράφων

Υπάρχουν δύο επίπεδα ορθότητας για ένα XML έγγραφο. Ένα έγγραφο XML για να είναι ορθό θα πρέπει καταρχάς να είναι καλοσχηματισμένο (well formed). Αυτό σημαίνει πως το έγγραφο θα πρέπει να είναι συντακτικώς ορθό αναφορικά με τους συντακτικούς κανόνες που θεσπίζει η XML. Συγκεκριμένα οι συντακτικοί κανόνες στους οποίους θα πρέπει να υπακούει κάθε έγγραφο είναι οι ακόλουθοι:

- Κάθε XML έγγραφο θα πρέπει να περιέχει ένα στοιχείο ρίζα (root element). Όλα τα υπόλοιπα στοιχεία θα εμφανίζονται ως child elements του στοιχείου αυτού.
- Όλα τα στοιχεία στο XML έγγραφο θα πρέπει να είναι κατάλληλα εμφωλευμένα μεταξύ τους
- Κάθε στοιχείο, εκτός από ένα opening tag στο οποίο και θα δηλώνεται, θα πρέπει να περιλαμβάνει και ένα closing tag (της μορφής “</element_name>”), το οποίο θα σηματοδοτεί το τέλος του στοιχείου και των περιεχομένων του. Του closing tag αυτού θα πρέπει να προηγούνται όλα τα closing tags των στοιχείων που είναι εμφωλευμένα στο στοιχείο αυτό
- Το περιεχόμενο των ετικετών είναι case sensitive, δηλαδή υπάρχει σαφής διαχωρισμός μεταξύ πεζών και κεφαλαίων γραμμάτων. Κατά συνέπεια τα στοιχεία “<table>” και “<Table>” θεωρούνται δύο διαφορετικά στοιχεία.
- Μέσα στις ετικέτες, όταν δηλώνουμε μία ιδιότητα (element) για κάποιο στοιχείο, η τιμή (value) της ιδιότητας αυτή θα πρέπει να περικλείεται ανάμεσα σε μονά ή σε διπλά εισαγωγικά.

Προκειμένου να ελεγχθεί η συντακτική ορθότητα ενός XML εγγράφου και να διαπιστωθεί εάν αυτό είναι καλοσηματισμένο, υπάρχουν ειδικά προγράμματα τα οποία διενεργούν τον έλεγχο αυτό και ονομάζονται XML Validators. Σε περίπτωση ύπαρξης συντακτικών σφαλμάτων στο έγγραφο, ο XML Validator προκαλεί τη διακοπή της επεξεργασίας του XML εγγράφου από τις εφαρμογές που χειρίζονται τα δεδομένα του.

Ωστόσο, για να θεωρηθεί ένα XML έγγραφο ορθό, εκτός από καλοσηματισμένο θα πρέπει επιπλέον να είναι και έγκυρο (valid). Ένα XML έγγραφο ονομάζεται έγκυρο όταν συμμορφώνεται στους κανόνες που ορίζουν ειδικά έγγραφα XML όπως οι Ορισμοί Τύπων Εγγράφων (Document Type Definitions – DTD) ή εναλλακτικά τα XML Σχήματα (XML Schemas).

Η XML λοιπόν είναι μία γλώσσα δόμησης δεδομένων η οποία είναι ευρέως διαδεδομένη σε πλήθος εφαρμογών αναφορικά με το Διαδίκτυο, ενώ έχει επικρατήσει και ως το κυρίως πρότυπο για την αποθήκευση και ανταλλαγή δεδομένων μεταξύ των εφαρμογών αυτών. Η επικράτησή της αυτή οφείλεται σε ένα πλήθος χαρακτηριστικών γνωρισμάτων της, τα οποία συνοπτικά είναι τα ακόλουθα:

Με την XML, δυναμικά δεδομένα μπορούν να αποθηκευτούν σε ξεχωριστά XML αρχεία, επιτρέποντας έτσι στο χρήστη να χρησιμοποιεί την HTML μόνο για την παρουσίαση των δεδομένων αυτών γνωρίζοντας πως οποιεσδήποτε αλλαγές από πλευράς χρήστη στα δεδομένα δε θα επισύρουν αλλαγές και στα HTML έγγραφα. Με αυτόν τον τρόπο επιτυγχάνεται ο διαχωρισμός μεταξύ XML και HTML.

Η XML είναι μία γλώσσα η οποία εμφανίζεται με τη μορφή απλού κειμένου, γεγονός που της παρέχει ανεξαρτησία συστήματος υλικού (hardware) αλλά και λογισμικού (software) όσον αφορά στην αποθήκευση των δεδομένων. Έτσι λοιπόν αποφεύγονται τυχόν ασυμβατότητες μεταξύ των δεδομένων που μπορεί να προκύψουν εξαιτίας του γεγονότος πως τα διάφορα υπολογιστικά συστήματα και οι βάσεις δεδομένων εμπεριέχουν δεδομένα με διαφορετικές κωδικοποιήσεις. Επιπλέον με την XML μπορεί να επιτευχθεί η απλοποίηση στη μεταφορά και την ανταλλαγή δεδομένων μεταξύ ανομοιογενών συστημάτων και ασύμβατων εφαρμογών λόγω της ανεξαρτησίας που προαναφέραμε.

Η XML μπορεί να αποτελέσει τη βάση και να χρησιμοποιηθεί για τη δημιουργία πλήθους γλωσσών που σχετίζονται με το δικτυακό προγραμματισμό (Internet Languages). Μία από αυτές είναι η XHTML (eXtensible Hyper Text Markup Language), η οποία αποτελεί τη νεότερη έκδοση της HTML και έχει πολλά κοινά χαρακτηριστικά με αυτήν, μόνο που το συντακτικό της έχει αλλάξει ώστε να συμβαδίζει με τους κανόνες της

XML. Άλλες τέτοιες γλώσσες είναι η WSDL (Web Services Description Language), η οποία χρησιμοποιείται για την περιγραφή διαθέσιμων διαδικτυακών υπηρεσιών, καθώς επίσης και η RDF (Resource Definition Framework – Πλαίσιο Περιγραφής Πόρων), η οποία υποστηρίζει περιγραφή πόρων και εφαρμογές μεταδεδομένων όπως οι κατάλογοι μουσικής και οι συλλογές φωτογραφιών.

2.4.2 XML Schema

Τα XML Schemas προτυποποιήθηκαν από το W3C το Μάιο του 2001 και αποτελούν μία εναλλακτική λύση των Document Type Definitions (DTDs) όσον αφορά στη θέσπιση ενός συνόλου κανόνων με τους οποίους θα πρέπει να συμμορφώνεται ένα XML έγγραφο για να μπορεί να θεωρηθεί έγκυρο. Η γλώσσα που χρησιμοποιείται στα XML Schemas συχνά αναφέρεται και ως XML Schema Definition (XSD). Στα πλαίσια θέσπισης των κανόνων αυτών οι κυριότερες αρμοδιότητες των XML Schemas είναι οι εξής:

Ορίζουν το σύνολο των στοιχείων που μπορούν να εμφανίζονται σε ένα XML έγγραφο, καθώς επίσης και τις ιδιότητες (attributes) που μπορούν να εμφανιστούν στο έγγραφο αυτό.

Καθώς συμμορφώνονται στη δενδρική δομή που επιβάλλει το πρότυπο της XML, τα XML Schemas ορίζουν ποια από τα στοιχεία που εμφανίζονται σε ένα XML έγγραφο είναι child elements και ποιο από αυτά είναι το root element, πόσα είναι τα στοιχεία που είναι child elements, καθώς επίσης και την ακριβή σειρά με την οποία εμφανίζονται στο έγγραφο.

Καθορίζουν τους τύπους δεδομένων (data types) για τα στοιχεία και τις ιδιότητές τους, καθώς επίσης και επιπλέον πληροφορίες όσον αφορά στα στοιχεία αυτά όπως τυχόν προκαθορισμένες τιμές που μπορούν να λάβουν και τη συχνότητα εμφάνισής τους μέσα στο έγγραφο.

Όπως προαναφέρθηκε, τα XML Schemas αποτελούν μία εναλλακτική προτυποποίηση των κανόνων εγκυρότητας ενός XML εγγράφου και χρησιμοποιούνται στη θέση των DTD. Η ευρεία χρήση τους στις μέρες μας οφείλεται στο γεγονός πως παρέχουν πληθώρα συγκριτικών πλεονεκτημάτων σε σχέση με τον προκάτοχό τους. Καταρχάς είναι γραμμένα σε XML γεγονός που τους επιτρέπει να κληρονομούν όλα τα πλεονεκτήματα της τεχνολογίας αυτής. Έπειτα υποστηρίζουν χαρακτηριστικά όπως οι τύποι δεδομένων (data types) και οι χώροι ονομάτων (namespaces) που καθιστούν

λειτουργικά και σαφώς ορισμένα τα XML έγγραφα, ενώ είναι και επεκτάσιμα σε μελλοντικές προσθήκες με γνώμονα τη βελτίωσή τους.

Στο σχήμα 2.4.2-1 που ακολουθεί, απεικονίζεται ένα XML έγγραφο και το αντίστοιχο XML Schema που ορίζει τη δομή και τον τύπο των δεδομένων του εγγράφου αυτού.

```
<? xml version="1.0" encoding="ISO-8859-1"?>
<!-- An XML example -->
<Person>
  <Surname> Papanikolas </Surname>
  <Job> Doctor </Job>
  <Male_Reg_ID> 47258 </Male_Reg_ID>
  <Birth_date> 1984/12/10 </Birth_date>
</Person>
```

```
<? xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Surname" type="xs:string" minOccurs="1"
maxOccurs="unbounded" />
        <xs:element name="Job" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Male_Reg_ID" type="xs:integer"
maxOccurs="1"/>
        <xs:element name="Birth_date" type="xs:DateTime"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Εικόνα 2.4-9: XML έγγραφο και το αντίστοιχο XML Schema

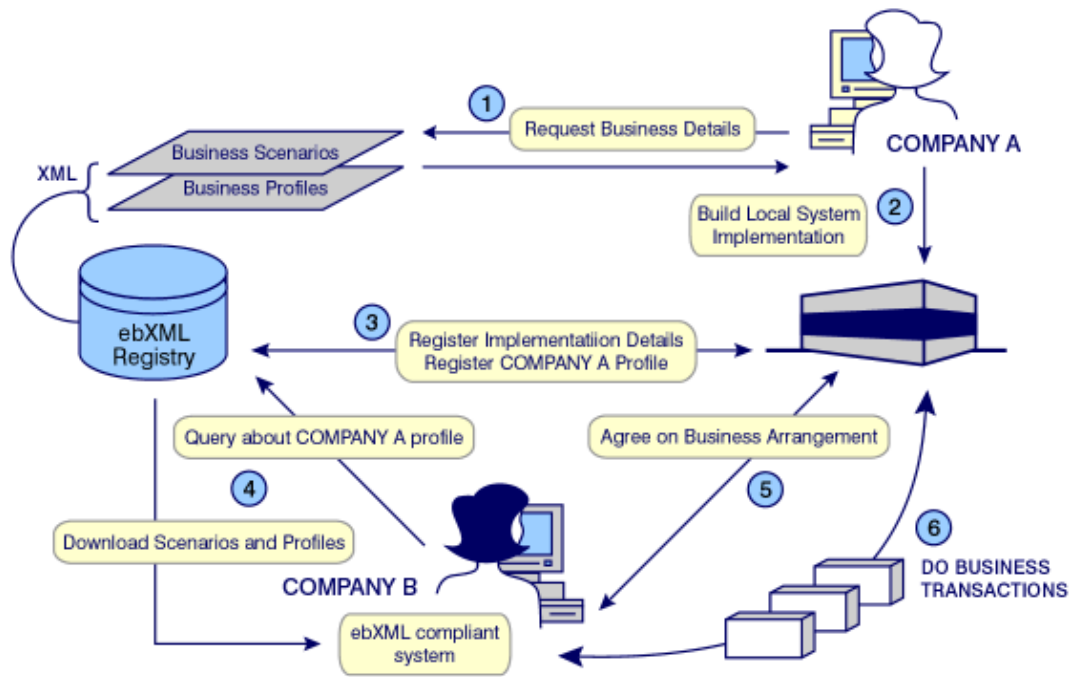
Όπως γίνεται αντιληπτό, η δομή του XML Schema είναι η εξής: Έχουμε ένα στοιχείο ρίζα (root element), του οποίου το όνομα είναι “Person” και ο τύπος είναι Complex Type και ως παιδιά αυτού ορίζονται τα <xs:element> με όνομα “Surname”, “Job”, “Male_Reg_ID” και “Birth_date” με τη σειρά με την οποία αναφέρθηκαν, η οποία είναι και η σειρά με την οποία εμφανίζονται στο αρχικό μας XML έγγραφο. Για κάθε ένα από τα στοιχεία παιδιά (child elements) αναφέρεται ο τύπος δεδομένων του που είναι και αυτός που καθορίζει το σύνολο των τιμών που μπορεί να πάρει το στοιχείο αυτό. Επίσης, για κάθε ένα τέτοιο στοιχείο αναφέρεται και ο ελάχιστος και ο μέγιστος αριθμός των φορών που συναντάται το στοιχείο αυτό μέσα στο έγγραφο. Εάν η τιμή του “minOccurs” ενός στοιχείου είναι “0”, αυτό σημαίνει πως το στοιχείο αυτό είναι προαιρετικό, ενώ εάν η τιμή του “maxOccurs” είναι “unbounded”, τότε το στοιχείο αυτό θα μπορεί να εμφανίζεται απεριόριστες φορές μέσα στο έγγραφο.

2.4.2 ebXML (electronic business eXtensible Markup Language)

Η ebXML (electronic business eXtensible Markup Language) αποτελεί ένα σύνολο κανόνων για την προτυποποίηση της XML παγκοσμίως, με σκοπό να διευκολύνουν το εμπόριο μεταξύ διεθνών οργανισμών ανεξαρτήτως του όγκου της μεταδιδόμενης πληροφορίας. Οι κανόνες αυτοί παρέχουν στις επιχειρήσεις μία πρότυπη μέθοδο για να ανταλλάσσουν επιχειρηματικά μηνύματα, να καθοδηγούν τις διεθνείς εμπορικές σχέσεις μεταξύ τους και να επικοινωνούν κάνοντας χρήση δεδομένων που διέπονται από μία κοινή σημασιολογία. Ως υποστηρικτές της ebXML φέρονται να είναι όλες οι επιχειρήσεις και οι κυβερνήσεις των συμβαλλόμενων σε αυτήν χωρών, ωστόσο δύο είναι οι θεμελιώδεις υποστηρικτές της, οι οποίοι και εισηγήθηκαν τη δημιουργία της: ο OASIS (Organization for the Advancement of Structured Information Standards) και το UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business). Αρκετά σημαντική επίσης ήταν και η συμμετοχή στη διαδικασία δημιουργίας και του NIST (National Institute of Standards and Technology), καθώς επίσης και του W3C (World Wide Web Consortium).

Κύριος στόχος που οδήγησε στη διαδικασία ανάπτυξης της ebXML ήταν η προσπάθεια επίτευξης διαλειτουργικότητας, η οποία και στέφθηκε από ένα μεγάλο ποσοστό επιτυχίας. Σε αυτό συνέβαλε η δημιουργία εγγράφων όπως το CCTS(Core Component Technical Specification), το οποίο αρχικά ήταν μέρος της ebXML και το οποίο έθετε τα πρότυπα για την ανταλλαγή και μετάδοση της επιχειρηματικής

πληροφορίας μεταξύ δύο επιχειρήσεων κάνοντας χρήση των Core Components, δομικών συστατικών με τη δική τους αυτόνομη σημασιολογική λειτουργικότητα, καθώς επίσης και των Business Information Entities που προέρχονται από την επιβολή περιορισμών στα Core Components ανάλογα με το επιχειρηματικό περιβάλλον στο οποίο αυτά δραστηριοποιούνται. Στο παρακάτω σχήμα, παρουσιάζεται αναλυτικά, σε αφαιρετικό επίπεδο, ο τρόπος που αλληλεπιδρούν δύο επιχειρήσεις κατά την ανταλλαγή δεδομένων μεταξύ τους.



Εικόνα 2.4-10: Αναπαράσταση της διαδραστικότητας μεταξύ δύο επιχειρήσεων.

2.4.3 UBL (Universal Business Language)

Από τη στιγμή που η XML αποτέλεσε σύσταση του οργανισμού W3C έχει υιοθετηθεί από πλήθος βιομηχανιών ως το πλαίσιο εργασίας για τον καθορισμό μηνυμάτων που ανταλλάσσονται στο ηλεκτρονικό εμπόριο. Η ευρεία χρήση της XML οδήγησε στην ανάπτυξη πολλαπλών εκδόσεων XML προσαρμοσμένων στις βιομηχανίες για την ανταλλαγή σχετικών εγγράφων. Ωστόσο, η ύπαρξη πολλών διαφορετικών προτύπων δεδομένων (data formats) για την ανταλλαγή πληροφοριών ανάμεσα σε διαφορετικούς επιχειρηματικούς τομείς χαρακτηρίζεται από ένα σημαντικό αριθμό προβλημάτων, όπως είναι τα εξής:

- Η ανάπτυξη και συντήρηση πολλαπλών εκδόσεων από κοινώς χρησιμοποιούμενα επιχειρηματικά έγγραφα απαιτεί τεράστια προσπάθεια.
- Η δημιουργία και συντήρηση πολλαπλών προσαρμογών για να αναπτυχθούν σχέσεις εμπορίου ανάμεσα στα όρια διαφορετικών τομέων απαιτεί ακόμη μεγαλύτερη προσπάθεια.
- Η ύπαρξη πολλαπλών προτύπων XML δυσκολεύει σε μεγάλο βαθμό την ενοποίηση των επιχειρηματικών μηνυμάτων σε XML με τα συστήματα που τα επεξεργάζονται.
- Η ανάγκη για υποστήριξη ενός αυθαίρετου αριθμού προτύπων XML καθιστά τα απαιτούμενα εργαλεία πιο ακριβά και τους κατάλληλα εκπαιδευμένους εργαζόμενους δύσκολο να βρεθούν.

Η UBL (Universal Business Language), που δημιουργήθηκε από τον οργανισμό OASIS, σκοπεύει να βοηθήσει στην επίλυση αυτών των προβλημάτων καθορίζοντας ένα γενικής χρήσης πρότυπο XML για τα επιχειρηματικά έγγραφα, το οποίο έχει τη δυνατότητα να επεκτείνεται για να καλύπτει τις ανάγκες συγκεκριμένων βιομηχανιών. Συγκεκριμένα, η UBL παρέχει τα εξής:

- Μία βιβλιοθήκη XML Σχημάτων για τα επαναχρησιμοποιήσιμα συστατικά δεδομένων (data Components) όπως “Διεύθυνση”, “Αντικείμενο”, “Πληρωμή”, τα οποία αποτελούν data elements των καθημερινών επιχειρηματικών εγγράφων.
- Ένα μικρό σύνολο από XML Σχήματα για κοινά επιχειρηματικά έγγραφα όπως “Παραγγελία” και “Τιμολόγιο”, τα οποία κατασκευάζονται από τα library Components της UBL και μπορούν να χρησιμοποιηθούν σε γενικής χρήσης εμπορικό πλαίσιο.
- Υποστήριξη για την προσαρμογή της UBL σε συγκεκριμένες εμπορικές συναλλαγές.

Η σταθερή βάση την οποία προσφέρει η UBL για τα επιχειρηματικά XML Σχήματα παρουσιάζει τα ακόλουθα πλεονεκτήματα:

- Χαμηλό κόστος ενοποίησης μεταξύ των επιχειρήσεων που θα επιτυγχάνεται μέσω της επαναχρησιμοποίησης κοινών δομών δεδομένων (data structures)
- Μείωση του κόστους του εμπορικού λογισμικού
- Χαμηλό κόστος εισόδου στη βάση από επιχειρήσεις μικρού και μεσαίου μεγέθους

- Προτυποποιημένη εκπαίδευση που θα οδηγεί σε πολλούς ικανούς εργαζομένους
- Μία παγκοσμίως διαθέσιμη κοινοπραξία από ενοποιητές συστημάτων
- Προτυποποιημένα και φθηνά εργαλεία εισαγωγής και εξαγωγής δεδομένων

Η UBL έχει σχεδιαστεί για να παρέχει ένα παγκοσμίως αντιληπτό και αναγνωρίσιμο συντακτικό για το νόμιμο “δέσιμο” των επιχειρηματικών εγγράφων, για να λειτουργεί εντός ενός πρότυπου επιχειρηματικού πλαισίου όπως είναι το ISO 15000 (ebXML) και για να παρέχει μία ολοκληρωμένη και βασισμένη σε πρότυπα υποδομή η οποία θα μπορεί να επεκτείνει τα οφέλη των ήδη υπάρχοντων συστημάτων στις επιχειρήσεις κάθε μεγέθους.

2.5 Στόχος

Στην παρούσα ενότητα, θα περιγραφεί το φυσικό πρόβλημα που καλούμαστε να αυτοματοποιήσουμε και θα δοθεί μια γενική περιγραφή του συστήματος

2.5.1 Φυσικό Πρόβλημα

Το φυσικό πρόβλημα στο οποίο στηρίζεται η παρούσα διπλωματική αναφέρεται στην κάθε είδους διακίνηση δεδομένων των πολιτών τα οποία χρησιμοποιούνται από διάφορους δημόσιους και ιδιωτικούς φορείς και από άλλες υπηρεσίες. Τα δεδομένα αυτά μπορεί να αποτελούν συστατικό ενός πιστοποιητικού, μιας βεβαίωσης, μιας αίτησης ή ενός άλλου παρόμοιου εγγράφου.

Σε γενικές γραμμές, η διακίνηση εγγράφων ανάμεσα στους κυβερνητικούς οργανισμούς πραγματοποιείται με τη διακίνηση εγγράφων. Παρόλα αυτά, τα δεδομένα που διακινούνται δεν έχουν συγκεκριμένη προτυποποίηση και μοντελοποίηση και η αποστολή, παραλαβή και επεξεργασία τους από τον ένα φορέα στον άλλο είναι μια διαδικασία επίπονη, χρονοβόρα και ενέχει πολλούς κινδύνους απώλειας δεδομένων κατά τη μεταφορά.

Για παράδειγμα, κατά την αποστολή ενός Πιστοποιητικού Οικογενειακής Κατάστασης δεν υπάρχει τρόπος ανάμεσα στα διάφορα ετερογενή συστήματα που ανταλλάσσουν αυτό το πιστοποιητικό να το αναγνωρίζουν απευθείας και να μεταφέρονται τα στοιχεία αυτόματα στο σύστημα. Υπάρχει η ενδιάμεση επέμβαση του ανθρώπου ο οποίος τα στοιχεία που θα παραλάβει από ένα σύστημα αποστολέα, θα

πρέπει να τα επεξεργαστεί και να τα αποθηκεύσει στο σύστημά του σε μια άλλη ενδεχομένως μορφή που καταλαβαίνει και χρησιμοποιεί το σύστημα παραλήπτης. Όπως αναφέρθηκε και πιο πάνω αυτή είναι μια επίπονη και χρονοβόρα διαδικασία η οποία ενέχει τον κίνδυνο ακόμα και να χαθούν κάποια δεδομένα αν για παράδειγμα κάποια πεδία του αρχικού εγγράφου δεν έχουν άμεση αντιστοιχία στα πεδία του συστήματος – παραλήπτη.

Γι αυτό το λόγο, κρίνεται απαραίτητη η χρήση ενός κοινού μοντέλου για την αναπαράσταση και μοντελοποίηση της πληροφορίας Ηλεκτρονικής Διακυβέρνησης ώστε τα διάφορα ετερογενή συστήματα να «καταλαβαίνουν» με τον ίδιο τρόπο τα διάφορα επιχειρησιακά έγγραφα και να «μεταφράζεται» η πληροφορία και τα δεδομένα από το ένα σύστημα στο άλλο αυτόματα, γρήγορα, αποτελεσματικά και με ασφάλεια.

Σαν **πρώτο βήμα** λοιπόν, κατασκευάζονται τα Core Components όπως αυτά έχουν αναλυθεί σε προηγούμενο εδάφιο, τα οποία αποτελούν τις βασικές δομές για τα στοιχεία του κάθε εγγράφου ξεκινώντας από το πώς θα μοντελοποιηθεί το όνομα και η διεύθυνση μέχρι το πώς θα μοντελοποιηθεί ένα φυσικό αντικείμενο όπως ένα σπίτι, ένα αυτοκίνητο ένα κτήριο ή μια εικόνα (ζωγραφιά, σχέδιο, φωτογραφία).

Δεύτερο βήμα και πιο σημαντικό, είναι η παραγωγή των Business Information Entities (BIEs) όπως αυτά επίσης έχουν αναλυθεί προηγουμένως, τα οποία προκειμένου να συσταθούν, έχουν ως πρότυπο κάποια από τα ήδη υπάρχοντα Core Components. Ένα BIE μπορεί να είναι κομμάτι ενός πιστοποιητικού, μιας βεβαίωσης ή μιας αίτησης. Αυτή η δομή, με βάση τις σχέσεις της με τις άλλες δομές και το περιεχόμενό της θα δώσει το τελικό αρχείο – XML σχήμα που επιθυμούμε το οποίο θα είναι well formed, σύμφωνα με το πρότυπο της XML και valid, σύμφωνα με τους κανόνες που υποβάλλει το Ελληνικό Πλαίσιο Διαλειτουργικότητας Ηλεκτρονικής Διακυβέρνησης (ΠΔΗΔ) όπως αυτό αναλύθηκε παραπάνω.

Παρακάτω, παρατίθενται τα στοιχεία τα οποία πρέπει να δοθούν σε κάθε ένα από τα τρία παραπάνω βήματα, τα οποία όπως είδαμε είναι:

- | | |
|---------|---|
| Βήμα 1° | Δημιουργία και διαχείριση των Core Components |
| Βήμα 2° | Δημιουργία και διαχείριση των Business Information Entities |
| Βήμα 3° | Δημιουργία τελικού XML σχήματος |

Απαιτούμενα στοιχεία για τη δημιουργία Core Components.

- UniqueID
- Name (eGIF XML Term)
- Dictionary Entry Name (DEN)
- Version
- Definition
- Object Class Term
- Property Term
- Associated Object Class Term
- Representation Term
- Cardinality Min
- Cardinality Max
- Greek Related Term(s)

Απαιτούμενα στοιχεία για τη δημιουργία BIEs.

- UniqueID
- Name (eGIF XML Term)
- Dictionary Entry Name (DEN)
- Version
- Definition
- Object Class Term Qualifier
- Object Class Term
- Property Term Qualifier
- Property Term
- Associated Object Class Term Qualifier
- Associated Object Class Term
- Representation Term
- Primitive Type
- Qualified Data Type
- Cardinality Min
- Cardinality Max
- Context: Business Process
- Context: Organization
- Context: Region
- Related Term(s)

2.5.2 Περιγραφή συστήματος

Το προς υλοποίηση σύστημα λογισμικού αναλαμβάνει να αυτοματοποιήσει τη δημιουργία όλων εκείνων των δομών που είναι απαραίτητες ώστε να αποδίδεται σημασιολογικό περιεχόμενο στα διάφορα δεδομένα Ηλεκτρονικής Διακυβέρνησης καθώς επίσης και των τελικών XML Σχημάτων που μοντελοποιούν τα διάφορα επιχειρησιακά έγγραφα Ηλεκτρονικής Διακυβέρνησης.

Το σύστημα λογισμικού που θα δημιουργηθεί θα αυτοματοποιεί τρεις βασικές λειτουργίες με βάση καθένα από τα τρία προαναφερθέντα βήματα που πρέπει να ακολουθηθούν προκειμένου να παραχθεί ένα τελικό XML Σχήμα:

- Δημιουργία και διαχείριση των Core Components. Η λειτουργία αυτή περιέχει την εισαγωγή των επιθυμητών χαρακτηριστικών του κάθε Core Component και την αντίστοιχη δημιουργία και αποθήκευσή του σε μια βάση δεδομένων. Στη συνέχεια ο χρήστης μπορεί να δει τα υπάρχοντα Core Components, να τα επεξεργαστεί, να δημιουργήσει καινούρια ή να τα διαγράψει.
- Δημιουργία και διαχείριση των Business Information Entities (BIEs). Η λειτουργία αυτή περιέχει την εισαγωγή των επιθυμητών χαρακτηριστικών του κάθε BIE και την αντίστοιχη δημιουργία του και αποθήκευσή του σε μια βάση δεδομένων. Το διαφορετικό στοιχείο σε αυτή τη διαδικασία είναι ότι κάθε BIE βασίζεται και έχει ως πρότυπο ένα από τα ήδη υπάρχοντα Core Components. Οπότε, κάποια πεδία και κάποια χαρακτηριστικά του κάθε BIE συμπληρώνονται αυτόματα ανάλογα με το ποιο Core Component έχει ως template του.
- Δημιουργία του τελικού XML Σχήματος. Σε αυτό το σημείο ο χρήστης επιλέγει ένα ABIE, το οποίο αναπαριστά ένα σύνολο από BIEs τα οποία αποτελούν τη βάση για ένα επιχειρησιακό έγγραφο, για ένα πιστοποιητικό για παράδειγμα, και με ένα κλικ παράγει το αντίστοιχο XML Σχήμα το οποίο είναι well formed όπως επιτάσσει η XML και valid σύμφωνα με τις επιταγές του ΠΔΗΔ.

Τα παραπάνω Core Components και BIEs αποθηκεύονται όπως αναφέρθηκε σε μια βάση δεδομένων ώστε να μπορεί ο χρήστης να ανατρέξει σε αυτά όποτε το επιθυμεί, να τα επεξεργαστεί, να τα διαγράψει και να τα χρησιμοποιήσει ανά πάσα στιγμή για τη δημιουργία νέων BIEs και νέων XML Σχημάτων ανάλογα με τις ανάγκες για προτυποποίηση και αναπαράσταση πληροφορίας.

3

Ανάλυση και Σχεδιασμός

Στο παρόν κεφάλαιο αναλύεται η διαδικασία σχεδιασμού και υλοποίησης του πληροφοριακού συστήματος η οποία τεκμηριώνεται με τα έγγραφα Προδιαγραφών Απαιτήσεων από το σύστημα, με το έγγραφο Περιγραφής της Αρχιτεκτονικής και με το έγγραφο Περιγραφής Λεπτομερούς Σχεδίου για το Σύστημα. Έστω ότι το σύστημα έχει το όνομα “EGODAM” από τα αρχικά: E-GOvernment Data Modeling. Αυτή η συμβολική ονομασία θα χρησιμοποιείται σε όλο το κείμενο ως όνομα του προς υλοποίηση λογισμικού συστήματος.

3.1 Εισαγωγή

3.1.1 UML (Unified Modeling Language)

Η Ενοποιημένη Γλώσσα Μοντελοποίησης (UML) είναι μια οπτική (visual) γλώσσα μοντελοποίησης με εκφραστικές δυνατότητες για την απεικόνιση (visualization), προδιαγραφή (specification), τεκμηρίωση (documentation) και κατασκευή (construction) των προϊόντων της διαδικασίας ανάπτυξης συστημάτων. Αποτελεί σήμερα βιομηχανικό πρότυπο το οποίο αναπτύχθηκε στο πλαίσιο του οργανισμού OMG και βρίσκεται σε συνεχή εξέλιξη. Ανήκει δε στην κατηγορία των αντικειμενοστρεφών γλωσσών μοντελοποίησης. Παρατίθεται εδώ, γιατί αποτέλεσε το εργαλείο με βάση το οποίο έχουν

σχεδιαστεί διάφορες όψεις του λογισμικού συστήματος που υλοποιήθηκε προκειμένου να δοθούν και πιο παραστατικές περιγραφές του συστήματος και όχι μόνο περιγραφικές.

3.1.2 *Microsoft Office Visio 2007*

Το Visio 2007 είναι ένα πρόγραμμα δημιουργίας διαγραμμάτων που διευκολύνει τη δημιουργία επιχειρηματικών και τεχνικών διαγραμμάτων για την τεκμηρίωση και την οργάνωση σύνθετων ιδεών, διαδικασιών και συστημάτων και το οποίο αποτέλεσε πολύτιμο εργαλείο στην εκπόνηση της παρούσας διπλωματικής εργασίας προκειμένου να σχεδιαστούν, με βάση πάντα τους κανόνες της UML, οι διάφορες όψεις του πληροφοριακού συστήματος.

3.2 *Έγγραφο Προδιαγραφών των Απαιτήσεων από το Λογισμικό Σύστημα*

3.2.1 *Εισαγωγή*

3.2.1.1 *Σκοπός*

Σκοπός του παρόντος εγγράφου είναι η παρουσίαση των προδιαγραφών των απαιτήσεων από το λογισμικό σύστημα το οποίο θα αναπτυχθεί στα πλαίσια μιας προσπάθειας αυτοματοποίησης της μοντελοποίησης των δεδομένων Ηλεκτρονικής Διακυβέρνησης και δημιουργίας των κατάλληλων XML Σχημάτων που για την προτυποποιημένη διαχείριση αυτών των δεδομένων.

3.2.1.2 *Εμβέλεια*

Με τη βοήθεια των διαγραμμάτων σε UML, το παρόν έγγραφο παρουσιάζει τις περιπτώσεις χρήσης του συστήματος. Απευθύνεται σε όλους τους δημόσιους και ιδιωτικούς φορείς που προσφέρουν ποικίλες υπηρεσίες στους πολίτες και που επιθυμούν να ανταλλάσσουν μεταξύ τους δεδομένα και έγγραφα των πολιτών σε μια «κοινή» γλώσσα. Απευθύνεται επίσης στον αρχιτέκτονα και κατασκευαστή του συστήματος και στους τελικούς χρήστες σε κάθε ένα από τους παραπάνω φορείς οι οποίοι θα αναλάβουν τη διαδικασία αυτοματοποιημένης μοντελοποίησης των δεδομένων για λογαριασμό του

κάθε ένα φορέα Ηλεκτρονικής Διακυβέρνησης και την περαιτέρω συντήρηση του λογισμικού συστήματος.

3.2.2 Μοντέλο Περιπτώσεων Χρήσης

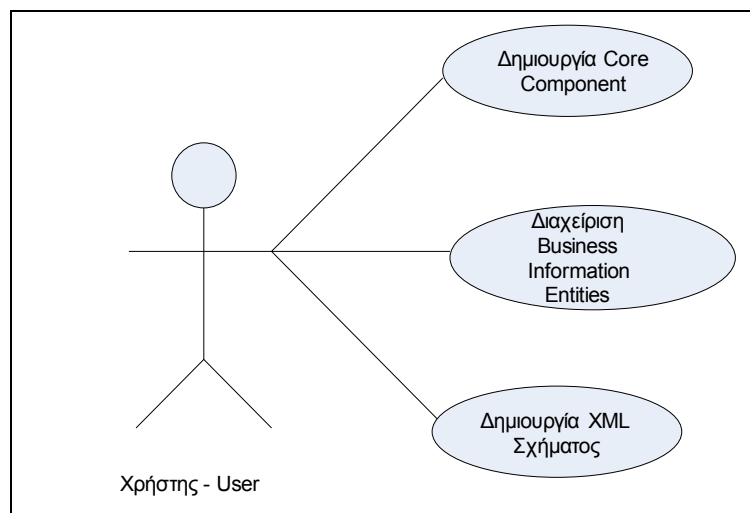
3.2.2.1 Στίγμα

Το μοντέλο περιπτώσεων χρήσης περιγράφει μια σειρά από ενέργειες που εκτελεί το σύστημα ώστε να αποδώσει χρήσιμα αποτελέσματα σε ένα ρόλο του συστήματος. Με τον όρο ρόλος του συστήματος αναφέρεται ένας χρήτης ή ένα εξωτερικό σύστημα (λογισμικό) που αλληλεπιδρά με το παρόν σύστημα λογισμικού για να λάβει κάποιες πληροφορίες.

Το παρόν σύστημα λογισμικού υλοποιεί τρεις λειτουργίες οι οποίες δημιουργούν αντίστοιχα τρεις περιπτώσεις χρήσης:

- Δημιουργία και διαχείριση Core Component
- Δημιουργία και διαχείριση Business Information Entity
- Δημιουργία XML Σχήματος

Κάθε περίπτωση χρήσης θα αναλυθεί σε ξεχωριστό έγγραφο στην επόμενη παράγραφο. Πιο αναλυτικά, το διάγραμμα περιπτώσεων χρήσης παρουσιάζεται στο σχήμα που ακολουθεί:



Εικόνα 3.2-11: Διάγραμμα Περιπτώσεων Χρήσης

3.2.2.2 Προοπτική

Το μοντέλο περιπτώσεων χρήσης αποβλέπει στη δημιουργία ενός εύκολα κατανοητού τρόπου παρουσίασης των απαιτήσεων.

3.2.2.3 Γενική εικόνα

Μια γενική εικόνα του προς υλοποίηση λογισμικού συστήματος δόθηκε στο τέλος του 2^{ου} κεφαλαίου ενώ η διεπαφή χρήστη (User Interface) αναλύεται στην επόμενη παράγραφο, για κάθε περίπτωση χρήσης ξεχωριστά.

3.2.2.4 Περιορισμοί Σχεδίασης – Συμμόρφωση με τα πρότυπα

Το σύστημα θα πρέπει να συμμορφώνεται με τις τεχνικές πολιτικές και τα πρότυπα που ορίζονται από το Ελληνικό Πλαίσιο Διαλειτουργικότητας Ηλεκτρονικής Διακυβέρνησης (ΠΔΗΔ, Greek e-Government Interoperability Framework) και να βρίσκεται σε συμφωνία με τις εξελίξεις στο σχεδιασμό πληροφοριακών συστημάτων Ηλεκτρονικής Διακυβέρνησης της Βρετανικής Κυβέρνησης (United Kingdom e-Government Interoperability Framework) και γενικότερα των χωρών της Ευρωπαϊκής Ένωσης (EU IDABC: Interoperable Delivery of European e-Government Services to public Administrations, Businesses and Citizens).

3.2.2.5 Επίδοση

Το παρόν σύστημα λογισμικού θα πρέπει να λειτουργεί σύμφωνα με τις καθορισμένες προδιαγραφές, που αναλύονται στο παρόν έγγραφο. Σε περίπτωση αποτυχίας κάποιας λειτουργίας ή κάποιας αίτησης προς τη βάση δεδομένων ή άλλης αστοχίας του προγράμματος θεωρείται απαραίτητο να υπάρχει μηχανισμός χειρισμού λαθών, που θα ενημερώνει κατάλληλα τον χρήστη.

3.2.3 *Ανάλυση περιπτώσεων χρήσης*

3.2.3.1 *Περίπτωση Χρήσης 1*

Διαχείριση των Core Components

3.2.3.1.1 *Περιγραφή Περίπτωσης Χρήσης*

Σε μια πρώτη επαφή του με το πρόγραμμα, και πριν φτάσει στη δημιουργία του Core Component, ο χρήστης έρχεται σε επαφή με τη διεπαφή (interface) του προγράμματος μέσω της οποίας θα διαχειρίζεται το πρόγραμμα. Τα βασικά χαρακτηριστικά που αποτέλεσαν κριτήριο με βάση το οποίο σχεδιάστηκε η διεπαφή ήταν το ότι πρέπει να είναι όσο το δυνατόν πιο απλή στη χρήση, να δίνει πλήρη εικόνα στον χρήστη για τα δεδομένα που βρίσκονται στη βάση, να τον ενημερώνει με κατάλληλα μηνύματα, να τον καθοδηγεί, όπου χρειαστεί, σε νέες φόρμες για περαιτέρω εισαγωγή, ανάκτηση, επεξεργασία ή διαγραφή δεδομένων και γενικά να είναι εύχρηστη και φιλική.

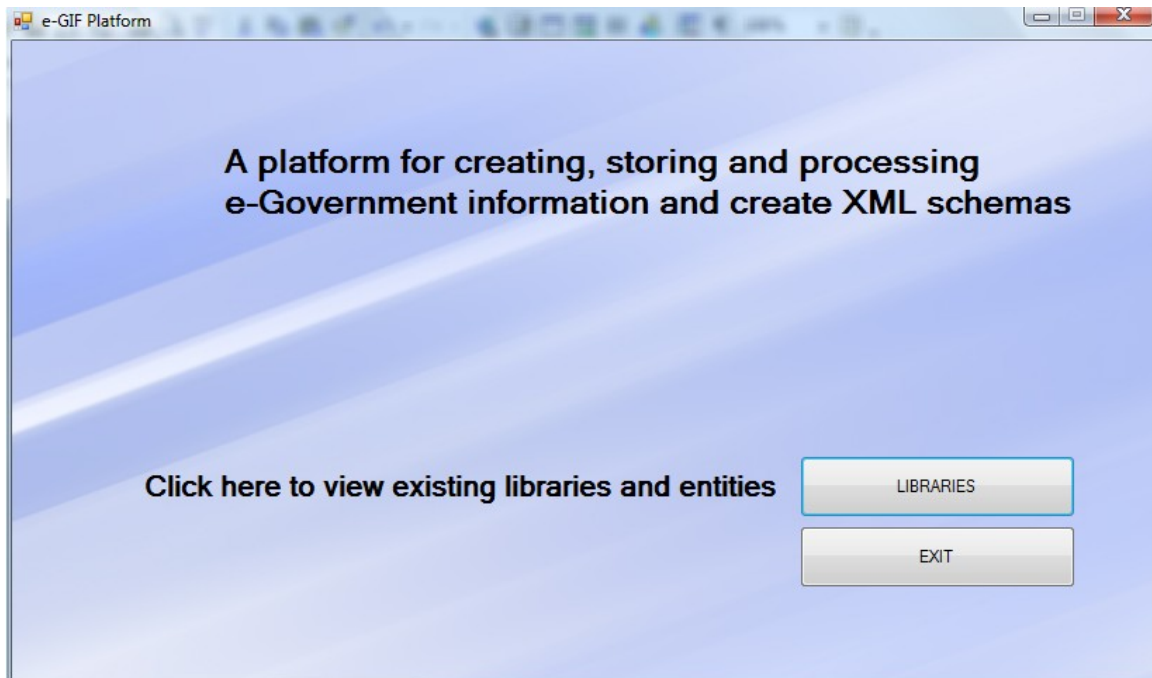
Με αυτό τον τρόπο, ο χρήστης δε θα δυσκολευτεί να εξοικειωθεί με τη χρήση του προγράμματος, ενώ οι διάφορες λειτουργίες διαχείρισης των δεδομένων σύντομα θα αποτελούν για το χρήστη μια διαδικασία αυτοματοποιημένη και αποδοτική.

Πρέπει εδώ να σημειωθεί, ότι η διεπαφή δεν είχε από την αρχή τη μορφή που θα παρουσιαστεί παρακάτω. Μέσα από μια σειρά λειτουργιών που σταδιακά προσθέτονταν, προέκυπταν συνεχώς πιο βελτιωμένες και πλήρεις μορφές διεπαφής μέχρι που τελικά, πήρε μια βέλτιστη τελική μορφή.

Στην επόμενη σελίδα παρουσιάζεται το εισαγωγικό περιβάλλον της εφαρμογής όταν κληθεί από τον χρήστη. Παρατηρούμε ότι το εισαγωγικό παράθυρο που βλέπει ο χρήστης όταν εκτελεί την εφαρμογή είναι ουσιαστικά ένα περιβάλλον καλωσορίσματος που καθοδηγεί τον χρήστη δίνοντάς του τη δυνατότητα να επιλέξει

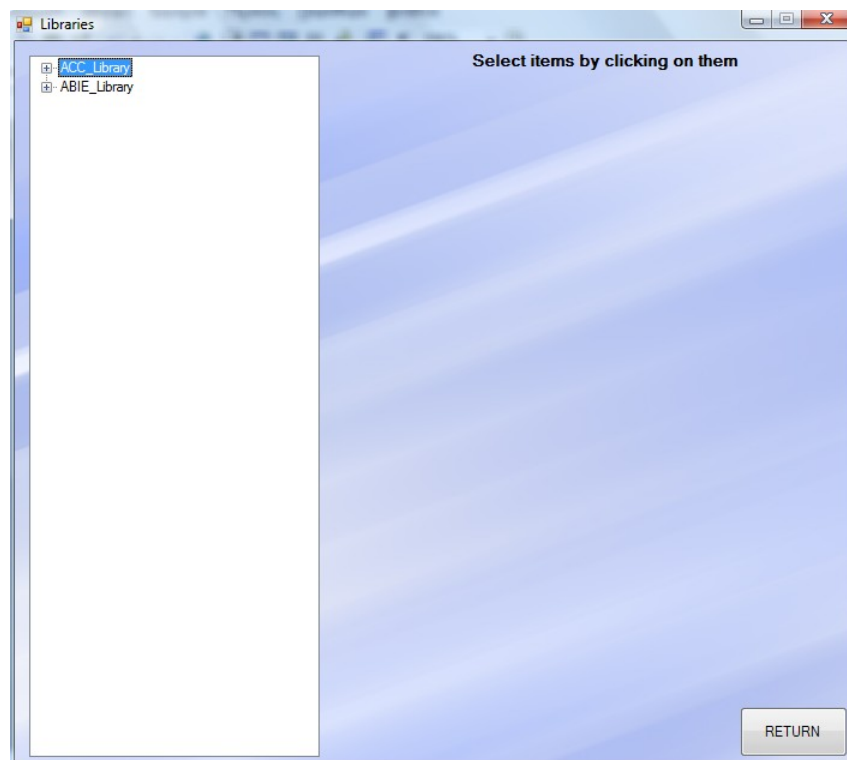
- να δει τις υπάρχουσες βιβλιοθήκες¹ με την κωδικοποιημένη πληροφορία, τα δεδομένα και τις αποθηκευμένες οντότητες και στη συνέχεια να τις διαχειριστεί όπως επιθυμεί,
- να τερματίσει την εφαρμογή

¹ Η έννοια βιβλιοθήκη εξηγείται στην αμέσως επόμενη σελίδα



Εικόνα 3.2-12: Εισαγωγικό Παράθυρο Εφαρμογής

Σε επόμενο βήμα, και αν ο χρήστης επιλέξει την πρώτη επιλογή, εμφανίζεται το δεύτερο περιβάλλον της εφαρμογής, το οποίο αποτελεί και την κυρίως διεπαφή. Η πρώτη εικόνα αυτής της διεπαφής πριν ο χρήστης κάνει οποιαδήποτε άλλη κίνηση είναι η παρακάτω:



Εικόνα 3.2-13: Παράθυρο υπαρχουσών Βιβλιοθηκών

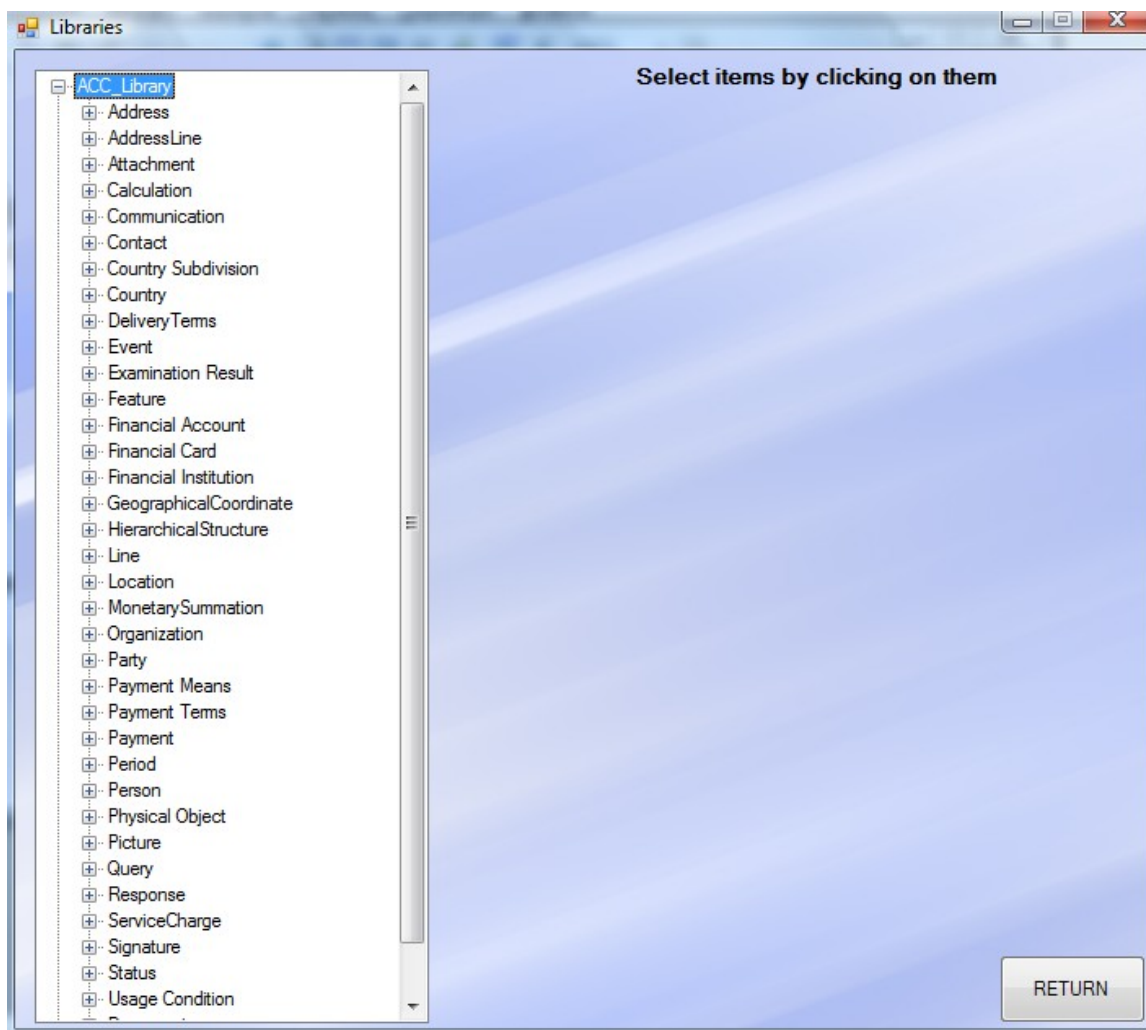
Παρατηρείται λοιπόν ότι αποτελείται από ένα παράθυρο το οποίο περιέχει τις βιβλιοθήκες με τα Aggregate Core Components (ACC_Library) και τα Aggregate Business Information Entities (ABIE_Library) καθένα από τα οποία αποτελείται από τα αντίστοιχα BCC και ASCC για τα ACCs και από τα αντίστοιχα BBIEs και ASBIEs για τα ABIEs, έννοιες και σχέσεις όπως αυτές έχουν περιγραφεί και αναλυθεί στο προηγούμενο κεφάλαιο.

Όταν αναφέρεται ο όρος «βιβλιοθήκες», ουσιαστικά πρόκειται για ένα σύνολο ομοειδών οντοτήτων. Οπότε, χρησιμοποιείται ο όρος «η βιβλιοθήκη με τα ACCs» και εννοείται το σύνολο των ACCs που είναι αποθηκευμένα στην βάση. Η παρούσα εφαρμογή έχει λοιπόν δυο «βιβλιοθήκες». Η δεύτερη είναι η βιβλιοθήκη των ABIEs και εννοείται το σύνολο των ABIEs που είναι αποθηκευμένα στη βάση. Φυσικά, υπάρχουν και κάποιες άλλες βιβλιοθήκες στην εφαρμογή μας, οι οποίες όμως υπάγονται στις δυο βασικές. Έτσι, μπορεί να γίνει λόγος για βιβλιοθήκες από BCCs οι οποίες είναι ουσιαστικά το σύνολο των BCCs ενός ACC τα οποία είναι προφανώς αποθηκευμένα στη βάση. Άρα, έχουμε κι άλλα σύνολα οντοτήτων για τα οποία όμως δε θα γίνει χρήση του όρου βιβλιοθήκη για να τα διαχωρίσουμε από τις δυο βασικές βιβλιοθήκες που αναφέρθηκαν παραπάνω.

Ο τρόπος με τον οποίο παρουσιάζονται οι δυο βιβλιοθήκες και τα περιεχόμενά τους είναι σε δενδρική μορφή καθώς είναι ένας πιο εύχρηστος και πιο παραστατικός τρόπος για τον χρήστη ώστε να έχει μια ιεραρχημένη και πλήρη εικόνα των περιεχομένων της βάσης δεδομένων κάθε στιγμή. Έτσι, δε χρειάζεται να αλλάζει παράθυρα και επιλέγοντας ποιοι κόμβοι του δέντρου θέλει να είναι ανεπτυγμένοι και ποιοι όχι κρατά μόνο την πληροφορία που τον ενδιαφέρει έχοντας παράλληλα τη δυνατότητα να έχει έλεγχο και παραστατική εικόνα των υπολοίπων πληροφοριών όποτε το θελήσει. Αυτή η αναπαράσταση δεδομένων σε δενδρική μορφή συναντάται πολύ συχνά σε σύγχρονες εφαρμογές λογισμικού εξαιτίας των χαρακτηριστικών που αναπτύχθηκαν παραπάνω.

Στην παραπάνω διεπαφή λοιπόν, δίνεται στο χρήστη η δυνατότητα να κάνει κλικ επάνω σε όποια βιβλιοθήκη θέλει, αναπτύσσοντάς την ώστε να εμφανιστούν τα περιεχόμενά της. Για την περίπτωση χρήσης που αναλύουμε σε αυτό το κομμάτι, ο χρήστης θα κάνει κλικ στην ACC_Library ώστε να αναπτυχθεί και να εμφανιστούν όλα τα Core Components που είναι αποθηκευμένα στη βάση δεδομένων. Σημειώνεται εδώ, ότι σε μια υποθετική πρώτη χρήση της εφαρμογής η βάση θα ήταν άδεια και θα δημιουργούσαμε εξ αρχής τα Core Components. Παρόλα αυτά έχουμε μια πιο «γεμάτη» εικόνα της βάσης αφού έχουν ήδη γίνει κάποιες προσθήκες για πιο ρεαλιστική παρουσίαση.

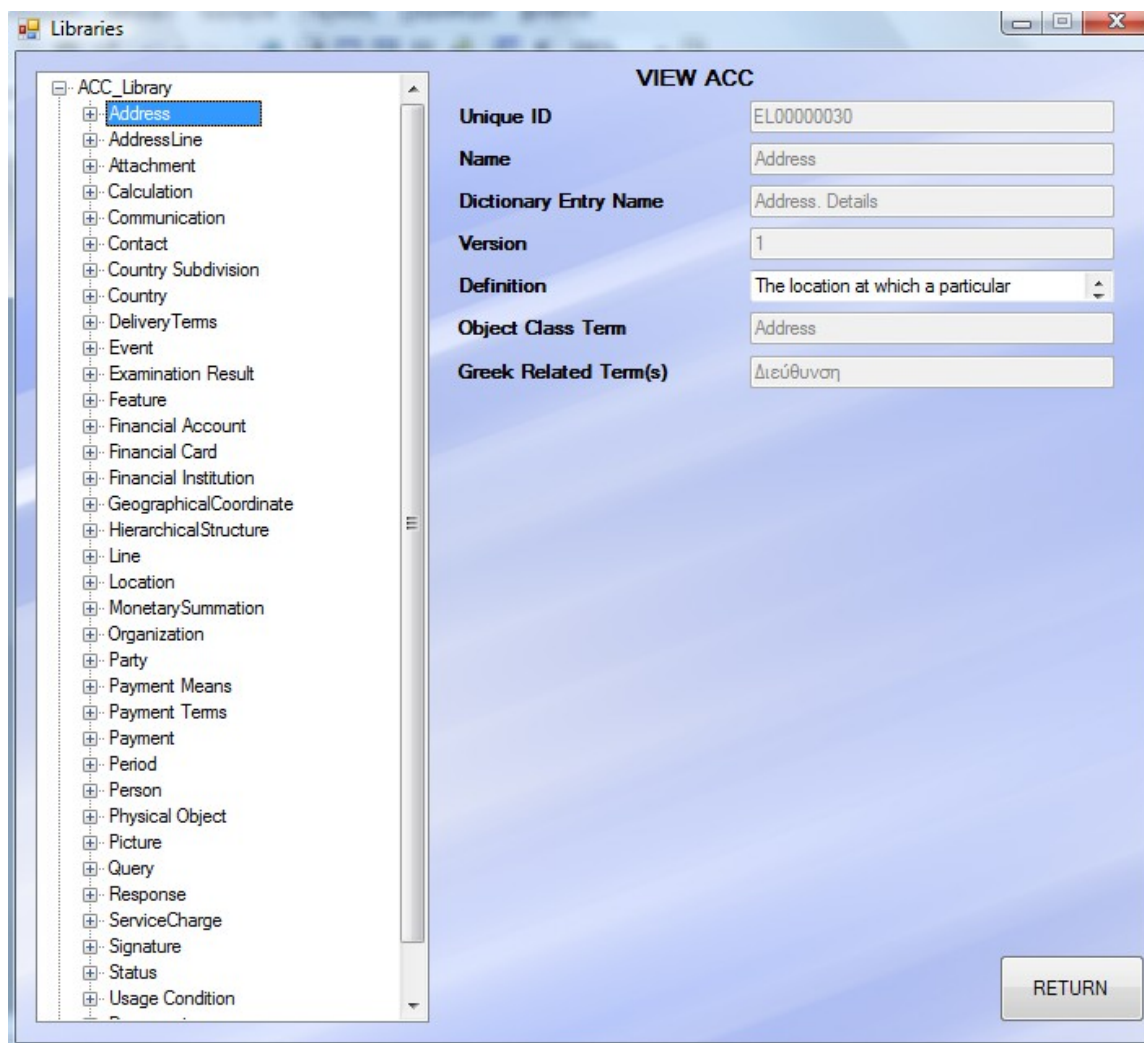
Κάνοντας κλικ λοιπόν στην ACC_Library βλέπουμε τα παρακάτω αποθηκευμένα Core Components:



Εικόνα 3.2-14: Αποθηκευμένα Core Components

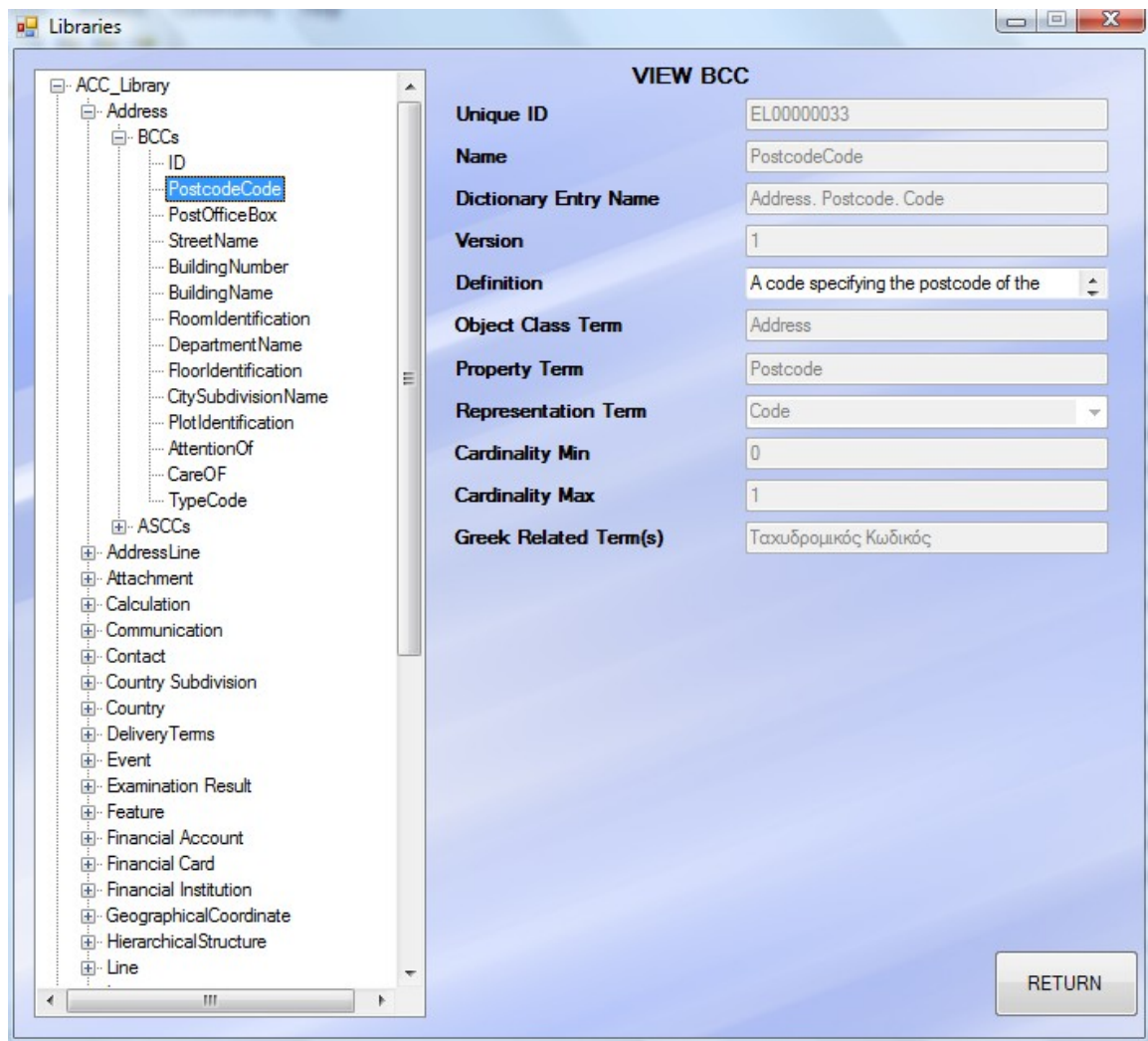
Συνεχίζοντας, ο χρήστης κάνει κλικ σε κάποιο από τα ACC που περιέχονται στη βιβλιοθήκη ώστε να αναπτυχθεί και να εμφανιστεί το περιεχόμενό του, δηλαδή τα BCC και ASCC από τα οποία αποτελείται.

Μια εικόνα που δείχνει αυτό το βήμα είναι η ακόλουθη:



Εικόνα 3.2-15: Πλήρες Παράθυρο Core Component

Σε αυτό το σημείο πια, η διεπαφή έχει μια από τις πιο χαρακτηριστικές της μορφές. Καταρχήν, όπως είναι φανερό, έχει αναπτυχθεί η βιβλιοθήκη με τα ACC, έχει επιλεγθεί ένα από αυτά που ήδη περιέχονται στην βάση (το Address), έχουν εμφανιστεί οι ετικέτες των BCC και ASCC απλά έχει αναπτυχθεί μόνο η πρώτη. Και αφού έγινε ανάπτυξη των BCC που ανήκουν στο συγκεκριμένο ACC, έχει επιλεγεί ένα από αυτά (το PostcodeCode). Αυτή η επιλογή έχει ως αποτέλεσμα να εμφανιστεί στα δεξιά του παραθύρου ένα σύνολο από δεδομένα. Αυτά τα δεδομένα είναι ουσιαστικά τα χαρακτηριστικά του BCC, όπως ο κωδικός του, το όνομά του, η έκδοσή του και διάφορα άλλα:

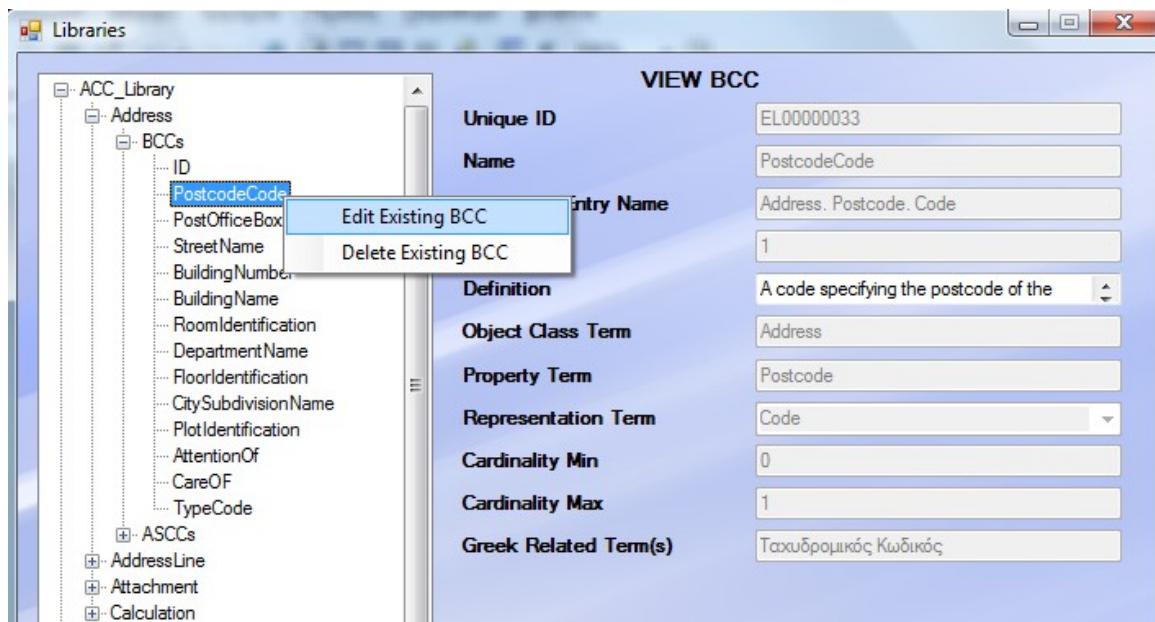


Εικόνα 3.2-16: Πλήρες Παράθυρο Core Component (2)

Όπως είναι εμφανές, η διεπαφή αποτελείται γενικά από Ετικέτες (Labels) και Textboxes (ή combo boxes σε κάποιες περιπτώσεις). Οι Labels δείχνουν το είδος των δεδομένων της οντότητας που επεξεργάζεται ο χρήστης τη δεδομένη στιγμή και τα αντίστοιχα Textboxes περιέχουν την τιμή του εκάστοτε δεδομένου. Δηλαδή, μια Label μπορεί να αναγράφει πχ το κείμενο “UniqueID” και να αναφέρεται στο δεδομένο «Μοναδική Ταυτότητα» της οντότητας και το αντίστοιχο Textbox μπορεί να περιέχει πχ την τιμή EL00000033, η οποία αποτελεί ουσιαστικά την «μοναδική ταυτότητα» της οντότητας που επεξεργάζεται το πρόγραμμα τη δεδομένη στιγμή. Αυτή η γενική μορφή της διεπαφής με τις Labels και τα Textboxes είναι παρόμοια για όλες τις διαφορετικές οντότητες που διαχειρίζεται το κυρίως πρόγραμμα αλλά μπορεί να παρουσιάσει διάφορες παραλλαγές στο είδος των δεδομένων και τις τιμές τους ανάλογα την οντότητα, καθώς κάθε τύπος οντότητας μπορεί να έχει διαφορετικά χαρακτηριστικά και άρα να αποτελείται από διαφορετικούς τύπους δεδομένων και κατά συνέπεια τα δεδομένα αυτά να είναι και

διαφορετικής φύσης. Γι αυτό και στη συνέχεια του κειμένου, οι διάφορες μορφές που μπορεί να πάρει το παράθυρο της διεπαφής ανάλογα με την οντότητα που διαχειριζόμαστε, ονομάζονται στιγμιότυπα της διεπαφής.

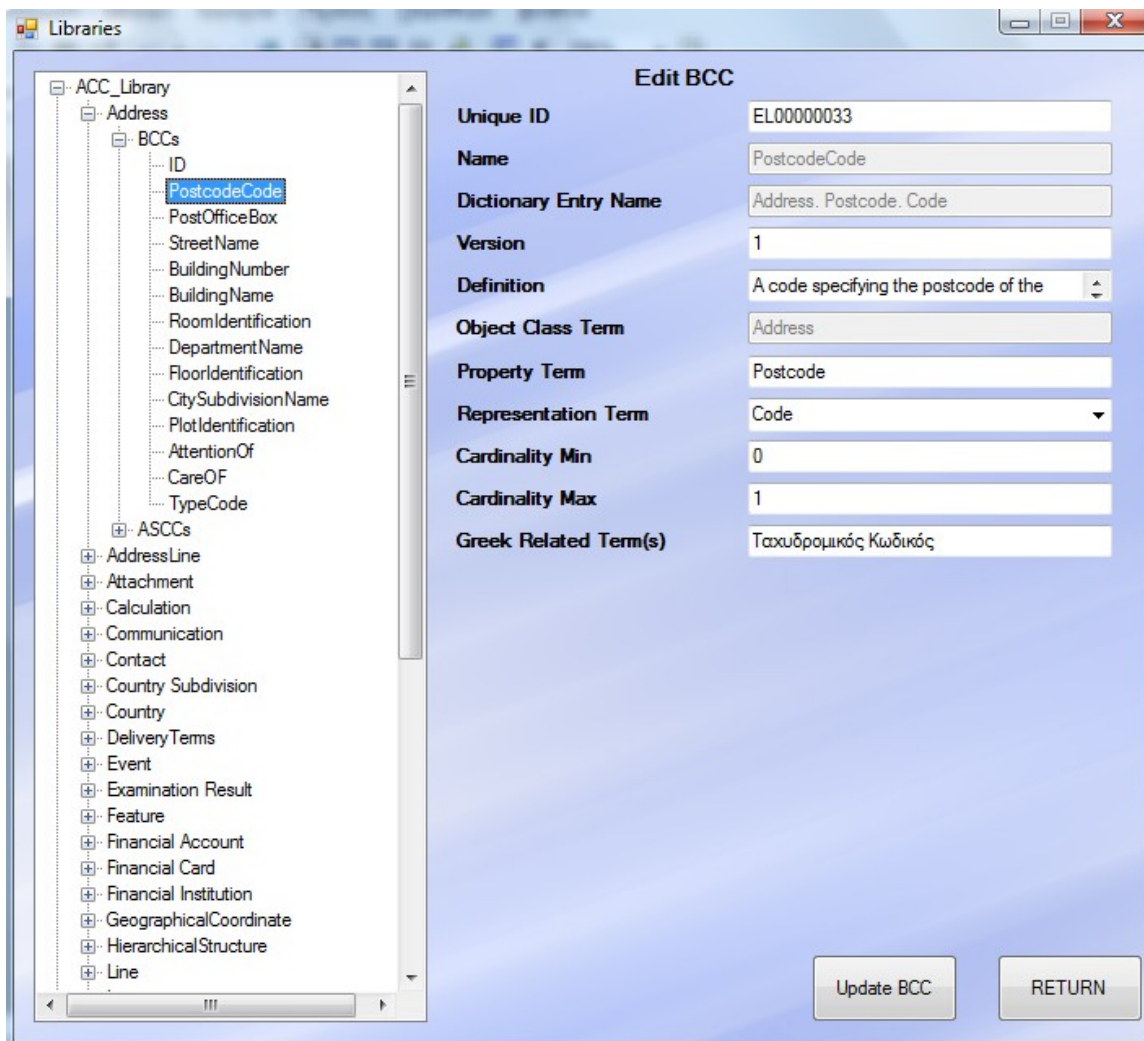
Επιστρέφοντας πάλι στο προηγούμενο παράθυρο, παρουσιάζονται και άλλες δυνατότητες που έχει ο χρήστης. Μπορεί λοιπόν να επεξεργαστεί κάποια από τις οντότητες ή και να την διαγράψει, επιλέγοντας την ενέργειά του μέσω ενός μενού που παρουσιάζεται αν κάνει δεξί κλικ επάνω στην οντότητα που επιθυμεί να επεξεργαστεί:



Εικόνα 3.2-17: Επεξεργασία Core Component

Αν επιλέξει να την επεξεργαστεί (edit), τότε θα πάρει μια εικόνα όπως αυτή που ακολουθεί στην επόμενη σελίδα. Το διαφορετικό που παρατηρείται στο συγκεκριμένο στιγμιότυπο της διεπαφής είναι το ότι τα πεδία που περιέχουν τις τιμές των εκάστοτε δεδομένων δεν είναι όλα απενεργοποιημένα όπως ήταν πιο πριν όπου απλά γινόταν εμφάνιση – επισκόπηση (View) των διαφόρων οντοτήτων. Επίσης, υπάρχει ένα ακόμα κουμπί (Update BCC) για την αποθήκευση των αλλαγών.

Σε αυτό το σημείο πρέπει να αναφερθεί ότι το ποια πεδία θα είναι ενεργοποιημένα και ποια όχι εξαρτάται από τους κανόνες του προτύπου που υλοποιείται ανάλογα με το αν κάποια πληροφορία είναι μοναδική, αν κάποια οντότητα έχει κάποια άλλη ως πρότυπο και κληρονομεί αυτόματα δεδομένα που δε μπορούν να αλλάξουν και γενικά ανάλογα με το τι κανονισμούς υπαγορεύει το πρότυπο για την διαχείριση της πληροφορίας. Η εικόνα με την επεξεργασία (edit) είναι η ακόλουθη:



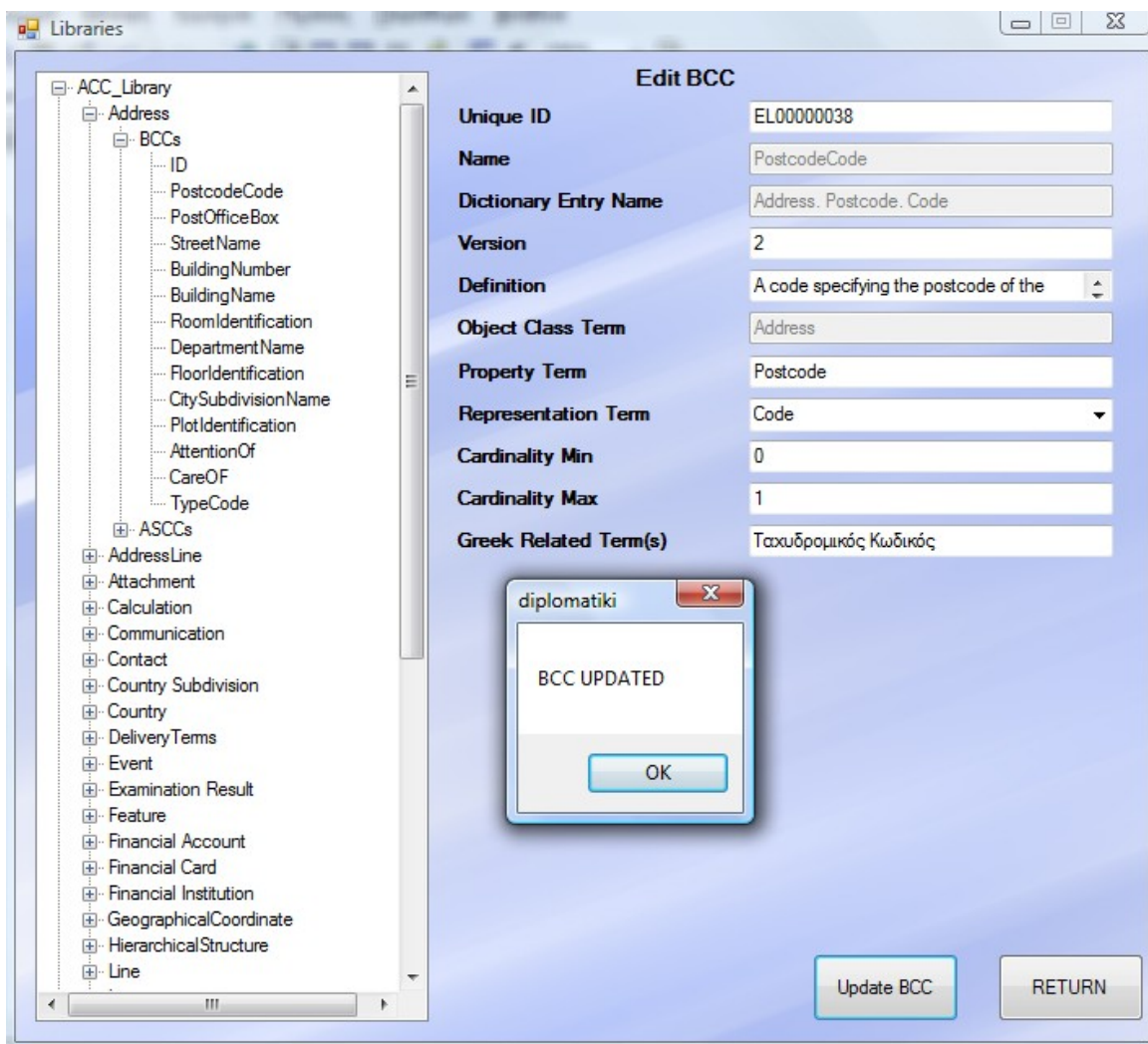
Εικόνα 3.2-18: Επεξεργασία Core Component (2)

Αν επιλεγεί η διαγραφή (delete) της οντότητας, τότε θα εκτελεσθεί η διαδικασία της διαγραφής και θα εμφανιστεί ένα μήνυμα στον χρήστη που θα τον ενημερώνει για την επιτυχία ή αποτυχία της διαγραφής.

Να αναφερθεί σε αυτό το σημείο ότι υπάρχει και η δυνατότητα δημιουργίας νέων οντοτήτων και αποθήκευσής τους στη βάση. Αυτή γίνεται εφικτή με δεξί κλικ επάνω σε ένα σύνολο οντοτήτων και όχι σε μια τυχαία οντότητα. Για παράδειγμα, για να δημιουργηθεί ένα νέο ACC πρέπει να γίνει δεξί κλικ επάνω στην ACC_Library (και όχι επάνω σε κάποιο ACC). Όταν θέλουμε να προστεθεί ένα νέο BCC, τότε κάνουμε δεξί κλικ επάνω στην ετικέτα με όνομα BCCs του εκάστοτε ACC στο οποίο θέλουμε να κάνουμε προσθήκη νέου BCC.

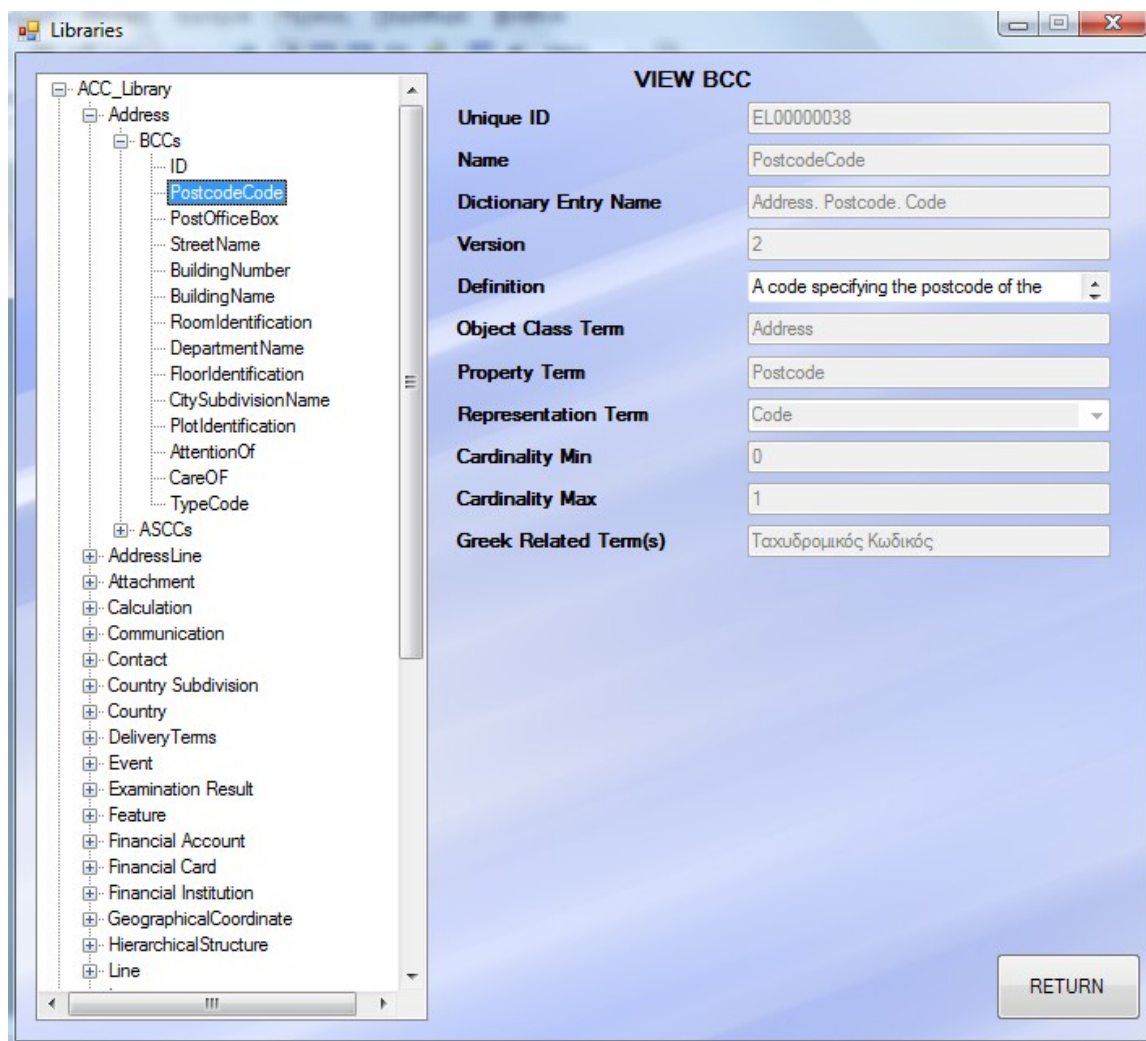
Όταν επιλεγεί όντως η δημιουργία νέας οντότητας, τότε στα δεξιά του παραθύρου εμφανίζεται ένα σύνολο από δεδομένα με τα αντίστοιχα πεδία άδεια και απενεργοποιημένα. Ο χρήστης επιλέγει τις τιμές που θέλει και τέλος επιλέγει

αποθήκευση (Store) και του εμφανίζεται ένα μήνυμα επιτυχίας ή αποτυχίας της αποθήκευσης στη βάση. Παράδειγμα δημιουργίας δε παρατίθεται στο παρόν τμήμα της εργασίας μιας και πλήρη παραδείγματα δημιουργίας και αποθήκευσης παρατίθενται στο κεφάλαιο της τεκμηρίωσης. Παρατίθεται όμως, παράδειγμα επεξεργασίας το οποίο ακολουθεί σχεδόν την ίδια διαδικασία με τη διαδικασία της καταχώρησης μόνο που κάποια πεδία είναι ενεργά για επεξεργασία και κάποια (εδώ το Object Class Term) δεν είναι μιας και συμπληρώνονται αυτόματα από το ποια οντότητα έχει ως πρότυπο η τρέχουσα οντότητα που διαχειριζόμαστε. Εδώ, το νέο BCC ανήκει στο ACC Address οπότε εξ' ορισμού και με βάση τους κανόνες προτυποποίησης, το Object Class Term του θα είναι ίδιο με αυτό του ACC στο οποίο ανήκει:



Εικόνα 3.2-19: Update Core Component

Στην επόμενη εικόνα φαίνεται και η απόδειξη του Update όπου έχει μεταβληθεί το version από 1 σε 2 και το UniqueID από EL00000033 σε EL00000038:



Εικόνα 3.2-20: Update Core Component (2)

Ροή γεγονότων

Παρακάτω παρουσιάζονται η βασική ροή γεγονότων που περιγράφει τη διαδικασία διαχείρισης των Core Components και πώς αυτή ολοκληρώνεται με επιτυχία. Στην περίπτωση που προκληθεί κάποιο σφάλμα και σε κάποιο σημείο αποτύχει αυτή η διαδικασία, έχουν προβλεφθεί και εφαρμόζονται εναλλακτικές ροές γεγονότων

Βασική Ροή:

1. Ο χρήστης επιλέγει τη βιβλιοθήκη ACC_Library και κάνοντας δεξί κλικ επιλέγει τη δημιουργία νέου Core Component (Create New Acc)
2. Συμπληρώνει όλα τα πεδία ανάλογα με τις προτιμήσεις του
3. Κάνει κλικ στο κουμπί Create ACC
4. Αν έχουν δοθεί στοιχεία τα οποία είναι αποδεκτά τότε γίνεται σύνδεση με τη βάση δεδομένων, καταχωρούνται τα δεδομένα και ενημερώνεται ο χρήστης για

την επιτυχή καταχώρηση των δεδομένων στη βάση. Αν προέκυψε κάποιο σφάλμα στην εισαγωγή δεδομένων στη βάση τότε εμφανίζεται κατάλληλο σφάλμα λάθους και επιστρέφει η ροή του προγράμματος στο βήμα 2.

5. Μετά την επιτυχή καταχώρηση του ACC γίνεται ένα refresh της οθόνης και εμφανίζεται η βασική διεπαφή του προγράμματος όπου με κλικ στην ACC_Library εμφανίζονται ξανά όλα τα αποθηκευμένα ACC μαζί με το νεοδημιουργηθέν ACC. Σειρά τώρα έχει η δημιουργία των BCC και ASCC του συγκεκριμένου ACC. Ο χρήστης σε αυτό το σημείο κάνει δεξί κλικ επάνω στο νεοδημιουργηθέν ACC και επιλέγει είτε δημιουργία BCC είτε δημιουργία ASCC ανάλογα με την επιθυμία του.
6. Συμπληρώνει όλα τα πεδία που εμφανίζονται ανάλογα με τις προτιμήσεις του.
7. Κάνει κλικ στο αντίστοιχο κουμπί δημιουργίας (Create New BCC/Create New ASCC)
8. Αν δεν υπάρχει κάποιο μη αποδεκτό δεδομένο τότε γίνεται σύνδεση στη βάση και αποθήκευση του εκάστοτε δομικού στοιχείου και ενημέρωση του χρήστη με κατάλληλο μήνυμα για την επιτυχή καταχώρηση, ενώ αν υπήρχαν μη αποδεκτά δεδομένα τότε εμφανίζεται κατάλληλο μήνυμα σφάλματος και ο ροή επιστρέφει στο βήμα 6.

Με παρόμοια ροή γίνονται και οι λειτουργίες της επεξεργασίας και διαγραφής των Core Components.

3.2.3.2 Περίπτωση Χρήσης 2

Διαχείριση Business Information Entities

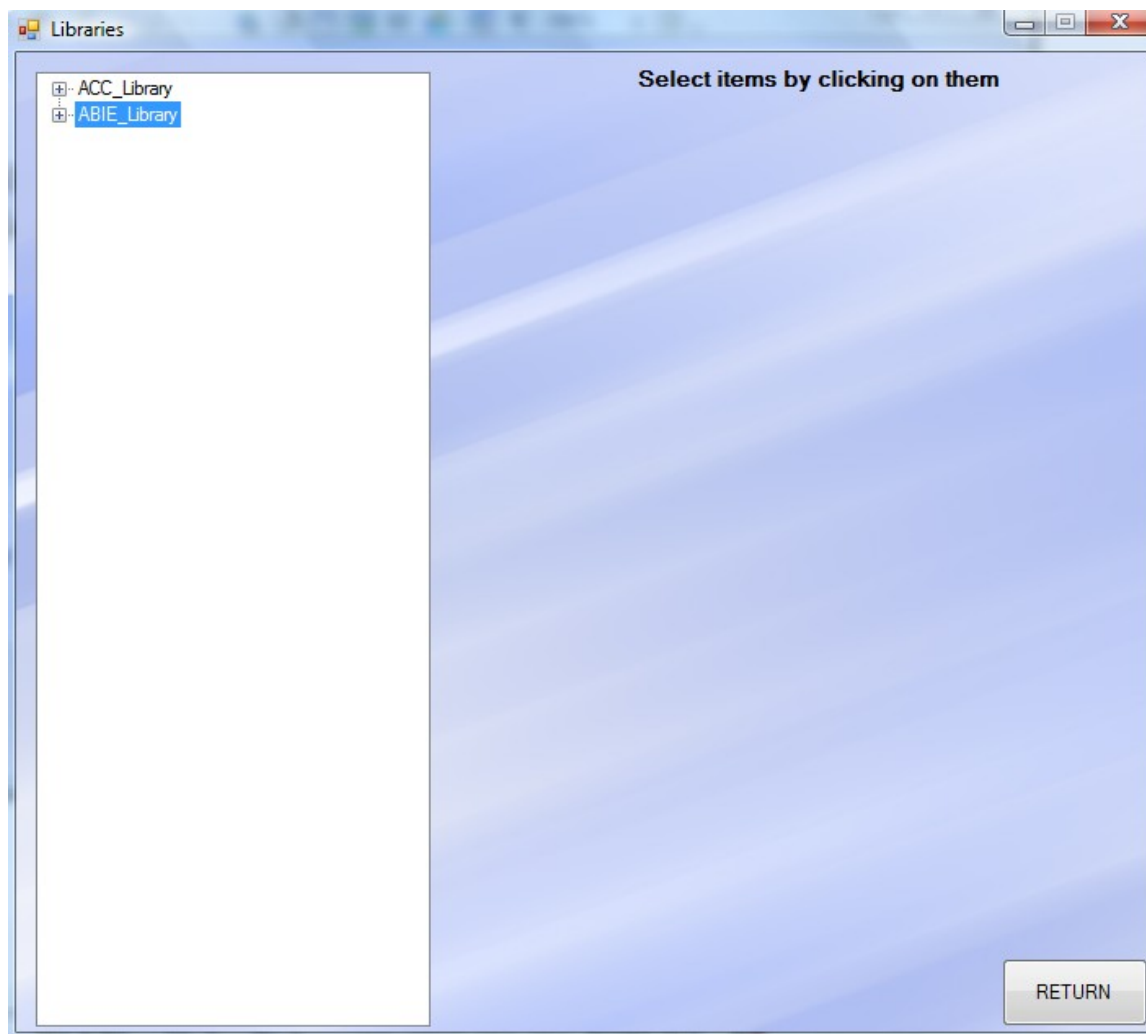
3.2.3.2.1 Περιγραφή Περίπτωσης Χρήσης

Παρόμοια είναι η χρήση της εφαρμογής και για τα BIE. Η σημαντική διαφορά στη συγκεκριμένη, εκτός από τα πιο πολλά νέα πεδία και δεδομένα, είναι το γεγονός του ότι τα BIE, έχοντας ως πρότυπο τα CC, έχουν πολλά πεδία τους απενεργοποιημένα (δεν μπορούν να επεξεργαστούν από το χρήστη) και ήδη συμπληρωμένα αυτόματα από το πρόγραμμα, κληρονομώντας κάποια χαρακτηριστικά τους από τα αντίστοιχα CC.

Σε αυτό το κομμάτι της διπλωματικής θα παρουσιαστεί η επεξεργασία και η διαγραφή ενός BIE, αφού πλήρες παράδειγμα δημιουργίας με βάση κάποιο πρότυπο δίνεται στο κεφάλαιο της τεκμηρίωσης.

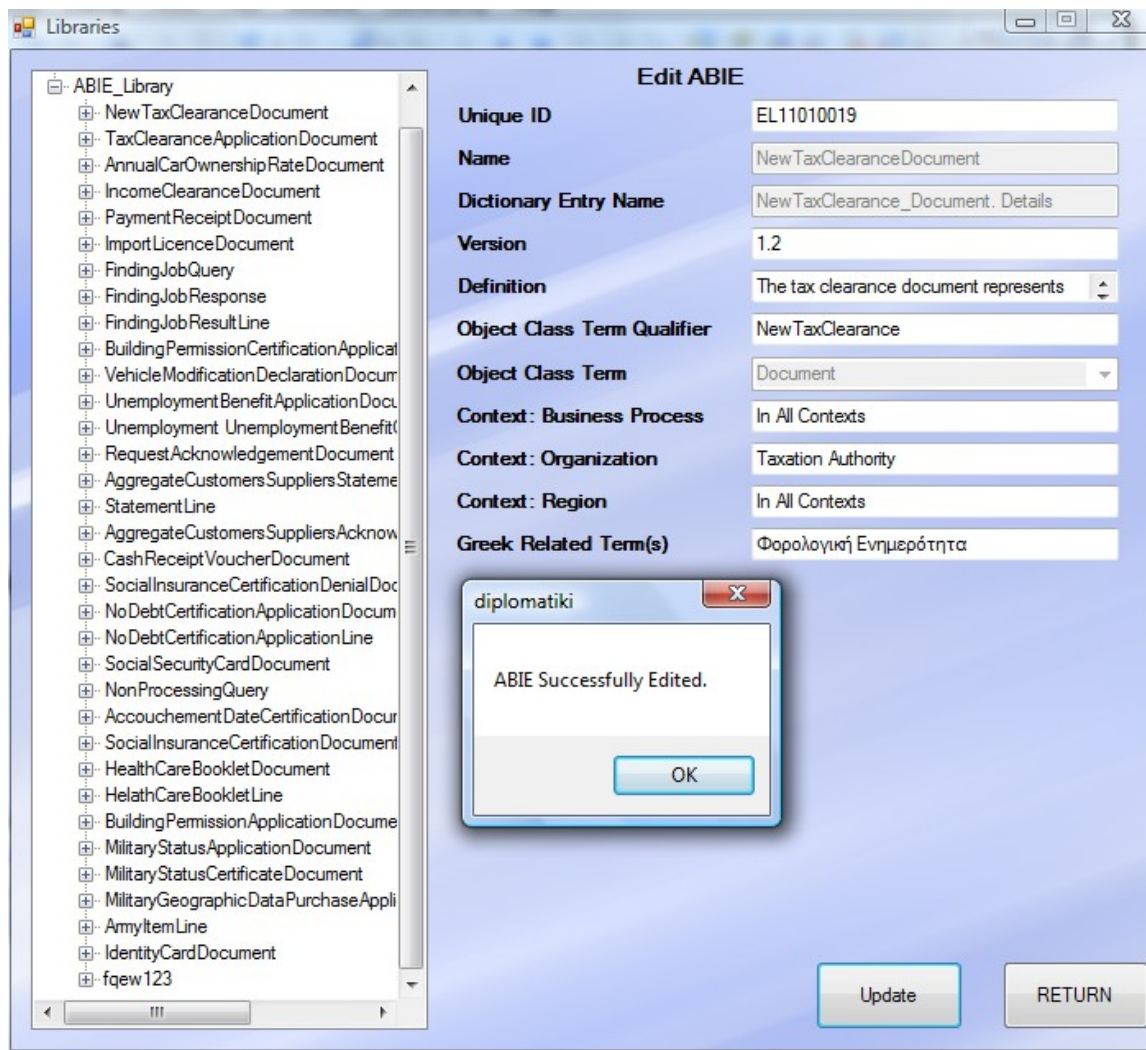
Για την επεξεργασία ενός ABIE, ο χρήστης θα πρέπει να ακολουθήσει τα παρακάτω βήματα:

Επιλογή της βιβλιοθήκης με τα ABIE:



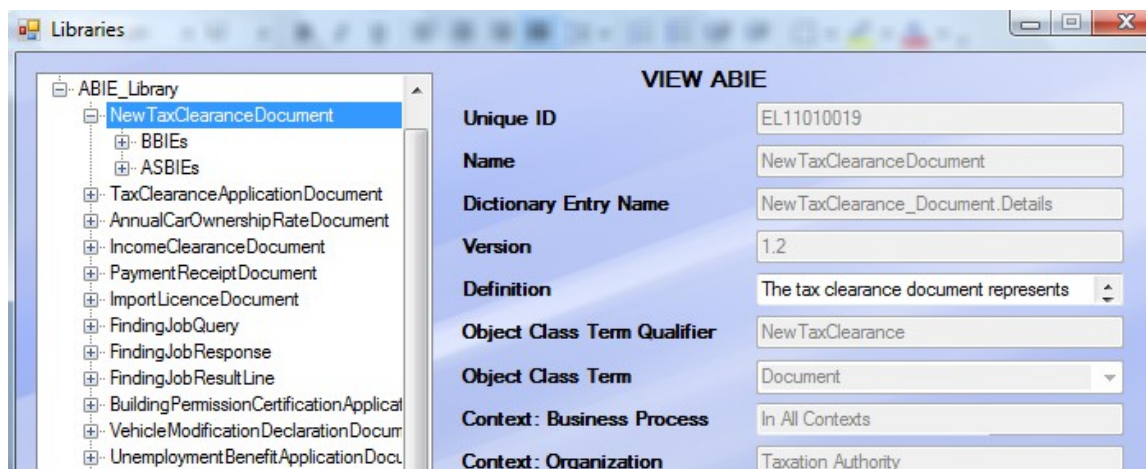
Εικόνα 3.2-21: Επιλογή Βιβλιοθήκης ABIE

Έστω ότι επιλέγεται για επεξεργασία το TaxClearanceDocument (Εγγραφο Φορολογική Ενημερότητα):



Εικόνα 3.2-22: ABIE Update

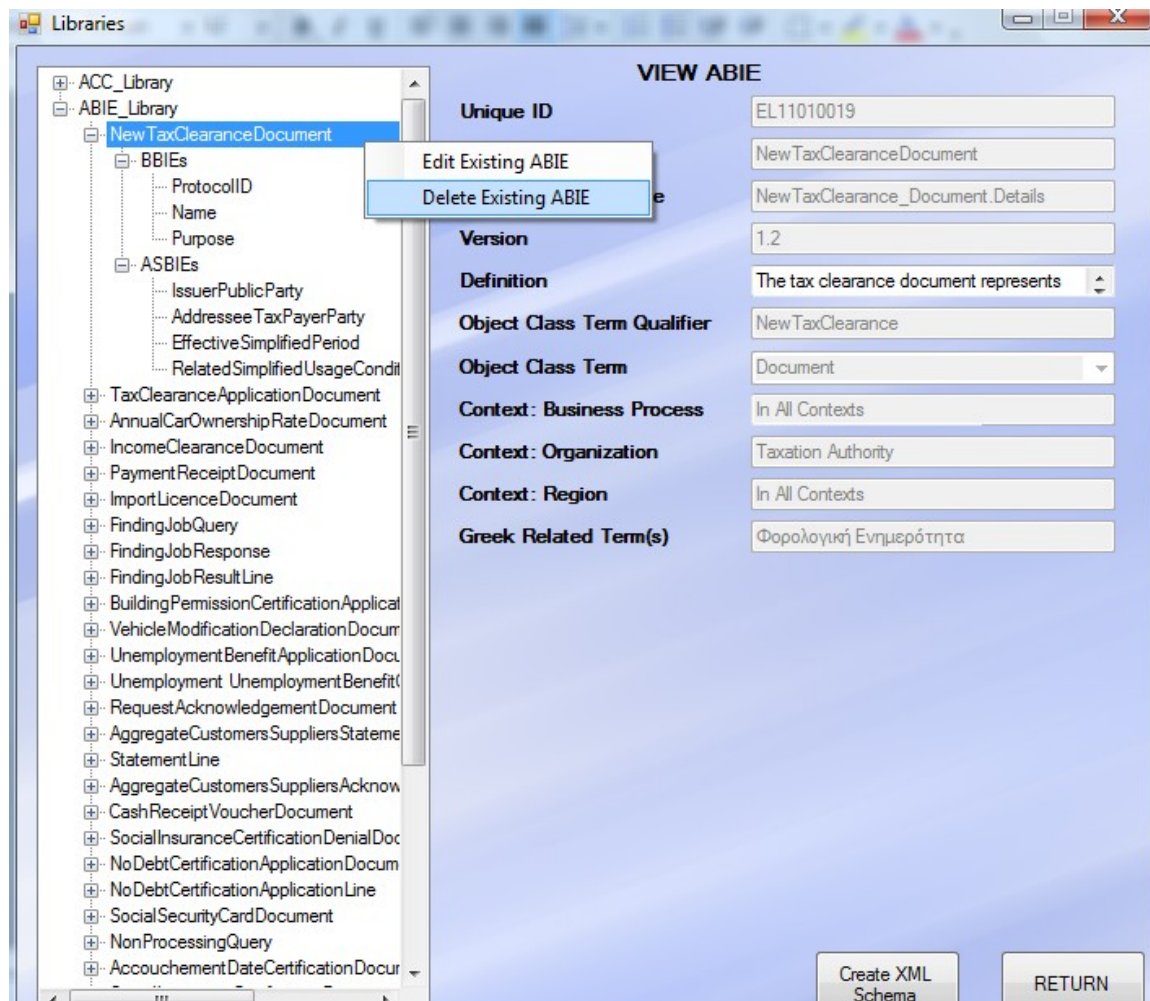
Και η τεκμηρίωση του Update:



Εικόνα 3.2-23: ABIE Update (2)

Έχει επιλεγθεί να εμφανιστούν μόνο τα σημεία που επεξεργάστηκαν και τα οποία όπως φαίνεται έχουν ενημερωθεί σωστά (η version έγινε 1.2, το Object Class Term Qualifier έχει γίνει NewTaxClearance από TaxClearance που ήταν πιο πριν και για το UniqueID άλλαξε το τελευταίο ψηφίο).

Θα παρατεθεί ακόμη, ένα παράδειγμα διαγραφής, και μάλιστα θα διαγραφεί ABIE του οποίου η διαγραφή συνεπάγεται τη διαγραφή και των αντίστοιχων BBIE και ASBIE προφανώς. Οπότε, έστω το ακόλουθο παράδειγμα:

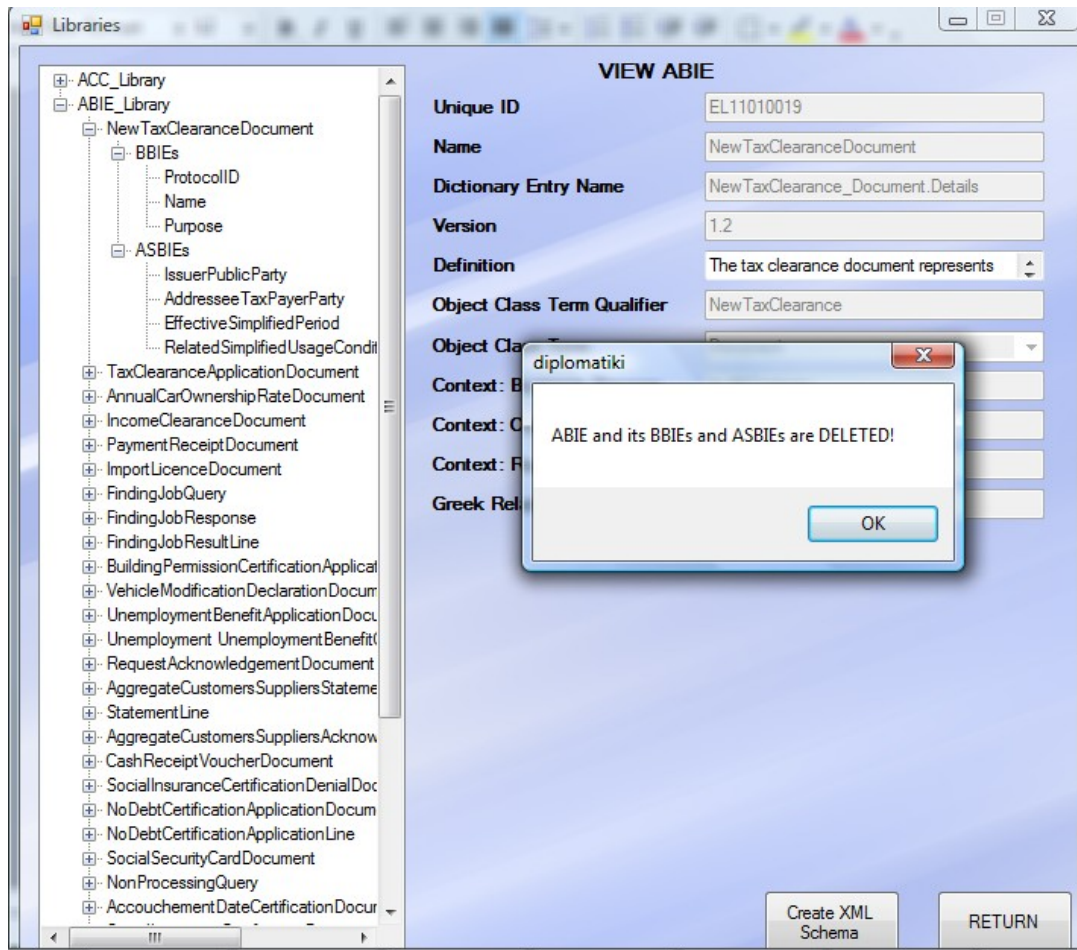


Εικόνα 3.2-24: Διαγραφή ABIE

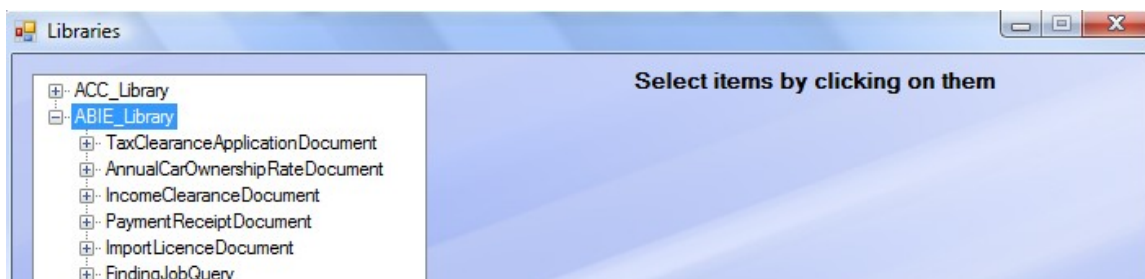
Και η διαδικασία και τεκμηρίωση της διαγραφής είναι στην επόμενη σελίδα. Για να πειστεί κανείς ότι διαγράφηκαν και τα BBIE και τα ASBIE δεν έχει παρά να ανατρέξει στη βάση δεδομένων να διαπιστώσει ότι έχουν και αυτά διαγραφεί με επιτυχία από τη βάση. Τα BBIE του NewTaxClearanceDocument δεν υπάρχουν πια στη βάση δεδομένων. Ήταν καταχωρημένα πριν τα BBIE του TaxClearanceApplication τα οποία τώρα είναι πρώτα σε καταχώρηση στη βάση δεδομένων:

ID	UniqueID	ABIEID	Name	DEN	Version	Definition
20	EL11010021	4	MultipleTypeIndi...	Tax Clearance A...	1	The indication of..
21	EL11010022	4	Name	Tax Clearance A...	1	A name, expres...
22	EL11010023	4	SubmissionDate...	Tax Clearance A...	1	The date, time, ...
23	EL11010024	4	CopyNumeric	Tax Clearance A...	1	The number of r...
24	EL11010025	4	DeclarationCont...	Tax Clearance A...	1	A textual descri...
26	EL11010008	36	Name	IdentityCard_Do...	1.0	A name, expres...
27	EL11010009	36	IssueDateTime	IdentitvCard Do...	1.0	The date. time. ...

Εικόνα 3.2-25: Στιγμιότυπο Βάσης Δεδομένων



Εικόνα 3.2-26: Διαγραφή ABIE (2)



Εικόνα 3.2-27: Τεκμηρίωση διαγραφής ABIE

3.2.3.3 Περίπτωση Χρήσης 3

Δημιουργία XML Σχήματος

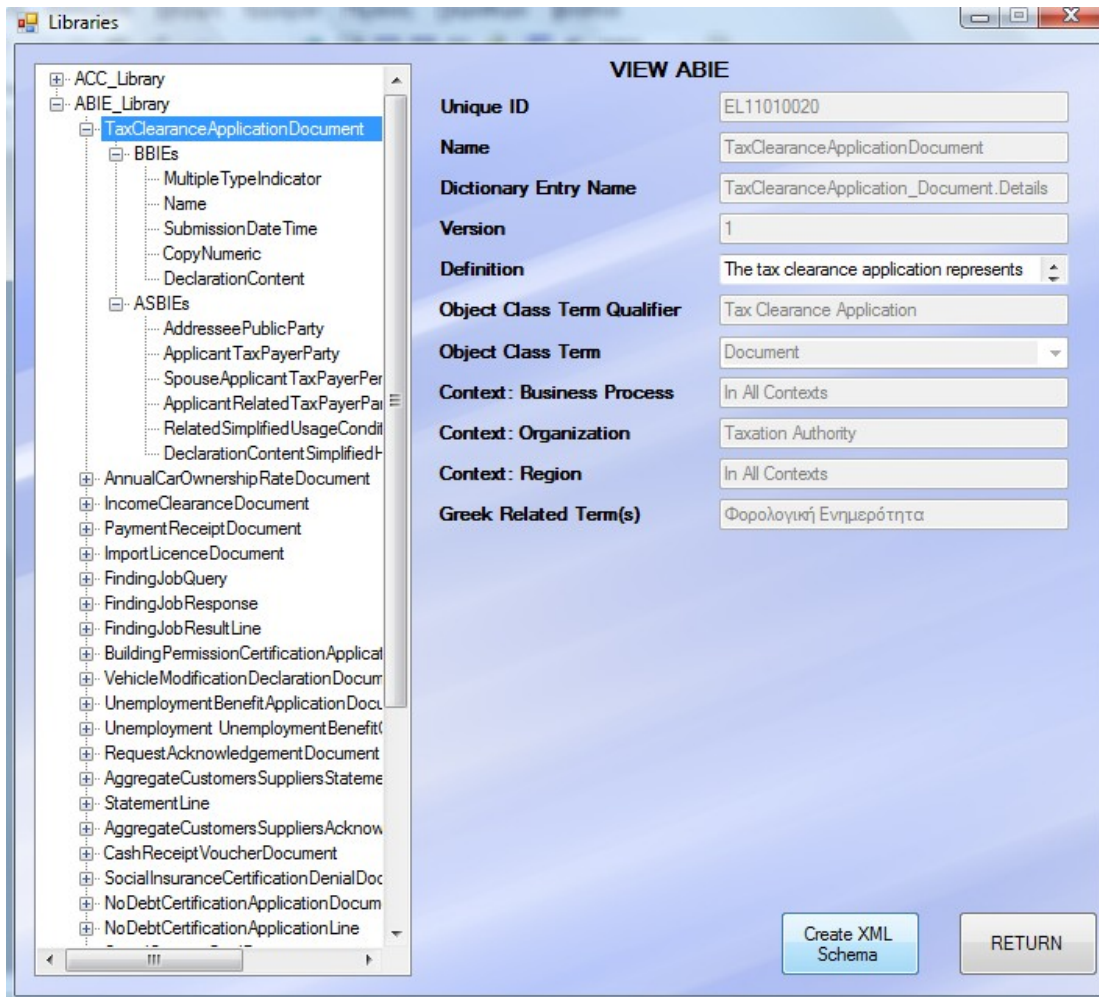
3.2.3.3.1 Περιγραφή Περίπτωσης Χρήσης

Φτάνοντας λοιπόν στον βασικό λόγο ύπαρξης του παρόντος λογισμικού συστήματος, την παραγωγή του τελικού XML Σχήματος από τα υπάρχοντα Core Components και Business Information Entities, παρατηρούμε ότι στο παράθυρο επεξεργασίας των BIE, υπάρχει άλλη μια πρόσθετη λειτουργία στο μενού που εμφανίζεται όταν γίνει δεξί κλικ επάνω σε ένα ABIE. Αυτή η λειτουργία, (εκτός των edit και delete) είναι η «create XML schema» η οποία, αν επιλεγεί, εμφανίζει μια νέα φόρμα όπου ο χρήστης επιλέγει πού θα αποθηκευθεί το παραγόμενο XML schema. Όταν δώσει την επιθυμητή διαδρομή και επιλέξει τη δημιουργία (create) του XML σχήματος, θα εμφανιστεί κατάλληλο μήνυμα που θα ενημερώνει για επιτυχή ή όχι δημιουργία του αρχείου. Ο χρήστης σε εκείνο το σημείο μπορεί να αναζητήσει στη δοθείσα διαδρομή το νέο αρχείο. Σε εκείνο το σημείο η αποστολή της παρούσας πλατφόρμας έχει εκπληρωθεί!

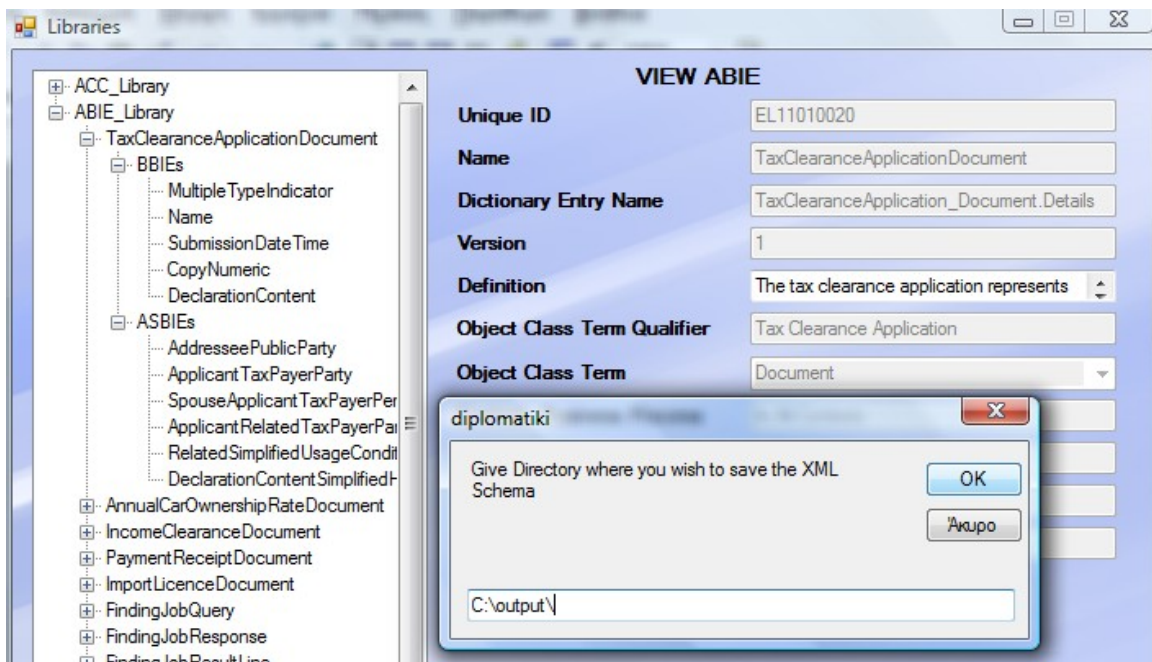
Η ροή γεγονότων έχει ως εξής:

1. Ο χρήστης επιλέγει την ABIE_Library
2. Επιλέγει ένα ABIE
3. Πατάει το κουμπί Create XML Schema
4. Δίνει την επιθυμητή διαδρομή αποθήκευσης του αρχείου
5. Πατάει OK και το πρόγραμμα τον ενημερώνει για την επιτυχή δημιουργία του αρχείου.
6. Εναλλακτικά μπορεί να πατήσει Cancel αν δεν επιθυμεί τελικά δημιουργία αρχείου.
7. Πηγαίνει στο δοθέντα προορισμό και ανοίγει το XML Schema με έναν Validator προκειμένου να ελέγξει την ορθότητα του παραγόμενου από το πρόγραμμα XML Σχήματος.

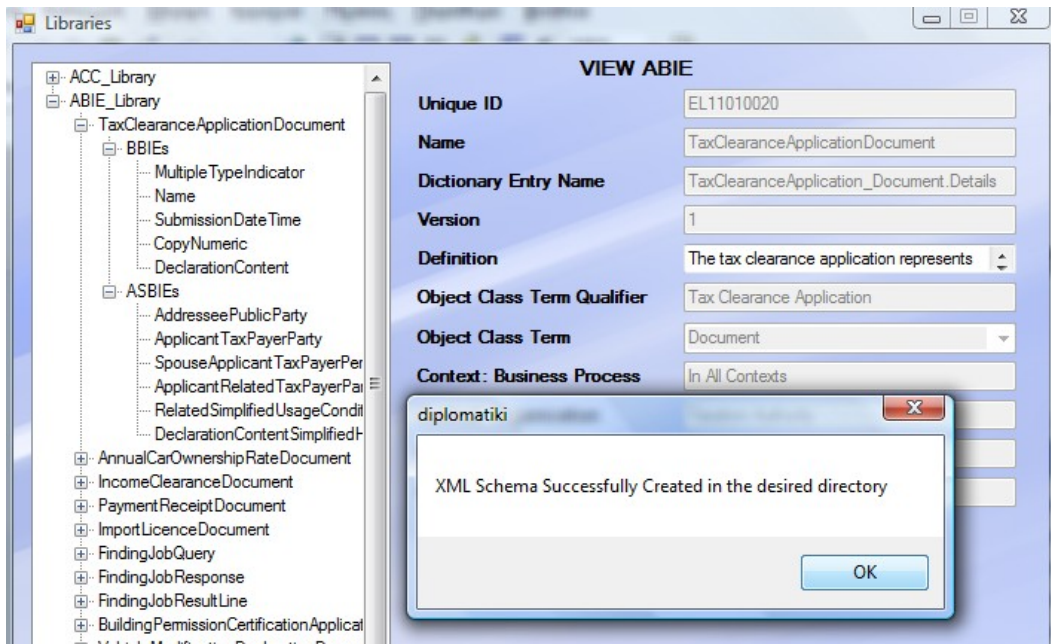
Ακολουθεί με στιγμιότυπα η εκτέλεση της παραπάνω ροής γεγονότων:



Εικόνα 3.2-28: Create XML Schema



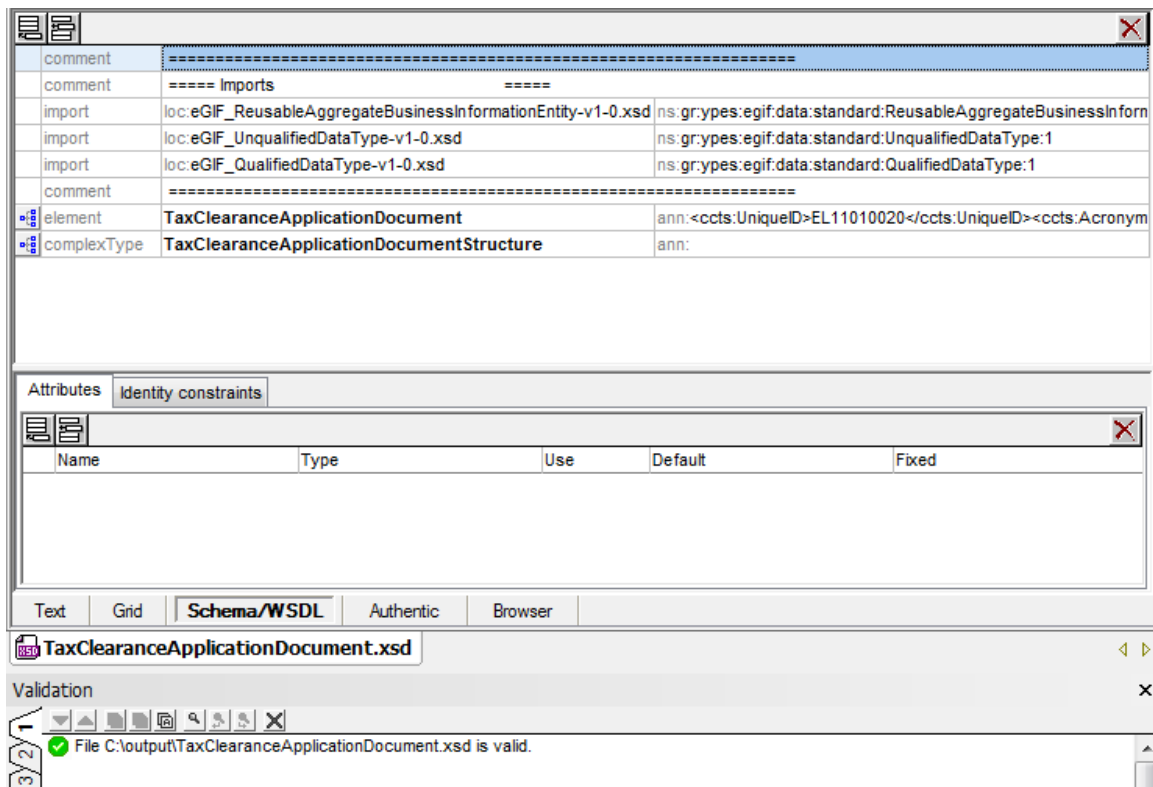
Εικόνα 3.2-29: Create XML Schema (2)



Εικόνα 3.2-30: Create XML Schema (3)

	TaxClearanceApplicationDocument.xsd	12/10/2008 5:21 μμ	W3C XML :
	fqew123.xsd	10/10/2008 12:19 μμ	W3C XML :
	NewTaxClearanceDocument.xsd	10/10/2008 12:04 πμ	W3C XML :
	IdentityCardDocument.xsd	9/10/2008 5:46 μμ	W3C XML :
	eGIF_IdentityCard-v1-0.xsd	18/2/2008 10:32 μμ	W3C XML :

Εικόνα 3.2-31: XML Schema in destination folder



Εικόνα 3.2-32: Validity of XML Schema

3.3 Έγγραφο Περιγραφής Αρχιτεκτονικής Συστήματος

3.3.1 Εισαγωγή

3.3.1.1 Σκοπός

Σκοπός του εγγράφου είναι η παρουσίαση της αρχιτεκτονικής του λογισμικού συστήματος το οποίο θα αναπτυχθεί στα πλαίσια της αυτοματοποιημένης μοντελοποίησης δεδομένων Ηλεκτρονικής Διακυβέρνησης και στην παραγωγή των αντίστοιχων XML σχημάτων.

3.3.1.2 Ενδιαφερόμενοι

Αποδέκτες του παρόντος εγγράφου είναι οι διάφοροι δημόσιοι και ιδιωτικοί φορείς που ανταλλάσσουν δεδομένα Ηλεκτρονικής Διακυβέρνησης καθώς επίσης και ο κατασκευαστής του παρόντος λογισμικού συστήματος.

3.3.2 Αρχιτεκτονικές αποφάσεις

3.3.2.1 Πλατφόρμα (υλισμική, λογισμική)

Ως υλισμική πλατφόρμα χρησιμοποιείται ένας προσωπικός υπολογιστής (PC) ο οποίος είναι εγκατεστημένος στον εκάστοτε δημόσιο φορέα ή υπηρεσία που επιθυμεί να επικοινωνεί με τους υπόλοιπους φορείς με κοινό τρόπο κατά την ανταλλαγή δεδομένων Ηλεκτρονικής Διακυβέρνησης. Αυτός ο υπολογιστής φέρει τη βάση δεδομένων στην οποία αποθηκεύονται τα εκάστοτε μοντελοποιημένα δεδομένα και σε αυτόν παράγονται και αποθηκεύονται τα τελικά XML αρχεία.

Ως λογισμική πλατφόρμα ανάπτυξης και εκτέλεσης χρησιμοποιούνται:

- Η πλατφόρμα Microsoft Visual Studio .NET 2005 όπου γίνεται η μοντελοποίηση και διαχείριση των δεδομένων Ηλεκτρονικής Διακυβέρνησης
- Ο Microsoft SQL Server 2005 ο οποίος υποστηρίζεται από το περιβάλλον Windows

Ως λογισμική πλατφόρμα συντήρησης θα χρησιμοποιηθεί η ίδια με αυτήν της ανάπτυξης και της εκτέλεσης. Στην παρούσα έκδοση, δεν προβλέπεται κάποιο ιδιαίτερο λογισμικό για τη συντήρηση.

3.3.2.2 Σχολή / Γλώσσα προγραμματισμού

Ως σχολή προγραμματισμού θα χρησιμοποιηθεί η αντικειμενοστρεφής σχολή προγραμματισμού και ως γλώσσα προγραμματισμού θα χρησιμοποιηθεί η Visual Basic η οποία αποτελεί βασική γλώσσα ανάπτυξης εφαρμογών στην πλατφόρμα .NET.

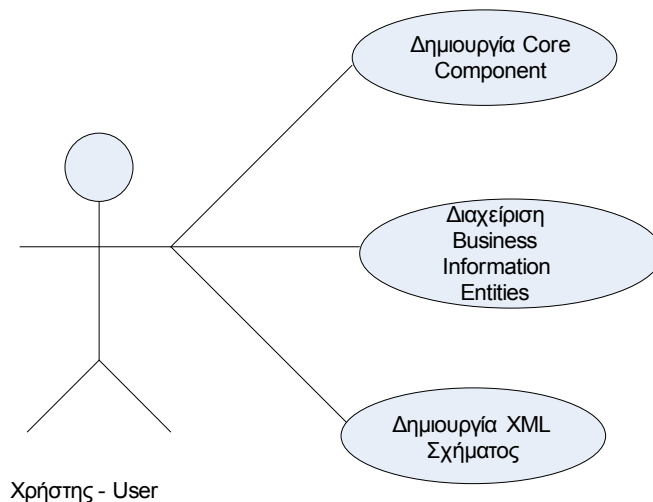
3.3.2.3 Κωδική Γλώσσα

Με βάση τη λογισμική πλατφόρμα που χρησιμοποιήθηκε, τελική γλώσσα του έργου είναι η Visual Basic, μαζί με τις διαπροσωπείες για επικοινωνία με τον SQL-Server.

3.3.3 Αρχιτεκτονικές όψεις

3.3.3.1 Περιβαλλοντική/Υπηρεσιακή

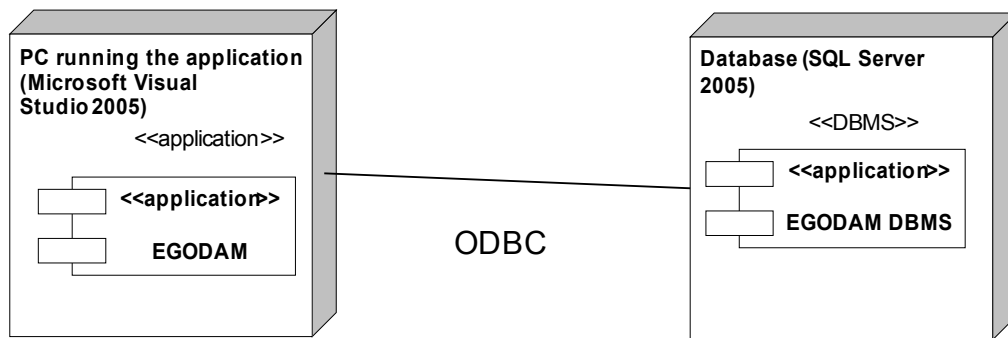
Στην περιβαλλοντική όψη φαίνονται ποιες εργασίες (υπηρεσίες) θα προσφέρει το λογισμικό σύστημα και σε ποιους οι οποίοι αποτελούν τους δράστες (actors). Περιγράφεται με ένα διάγραμμα περιπτώσεων χρήσης της UML που για το σύστημα 'EDAM' θα είναι:



Εικόνα 3.3-33: Περιβαλλοντική - Υπηρεσιακή Όψη

3.3.3.2 Πλατφορμική

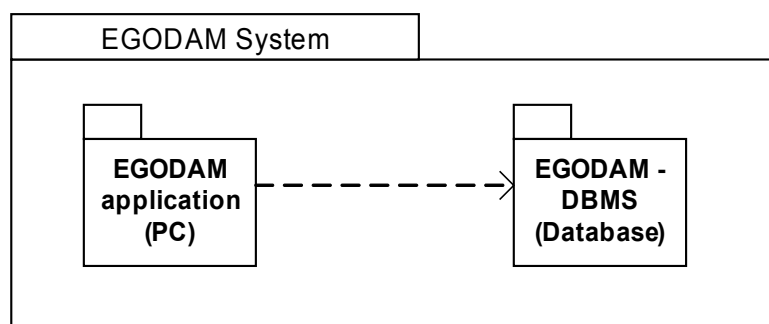
Η πλατφορμική όψη περιγράφει με ένα διάγραμμα παράταξης (deployment) της UML, την υλισμική και λογισμική πλατφόρμα στην οποία θα ενσωματωθεί για να τρέχει το λογισμικό σύστημα “EGODAM”.



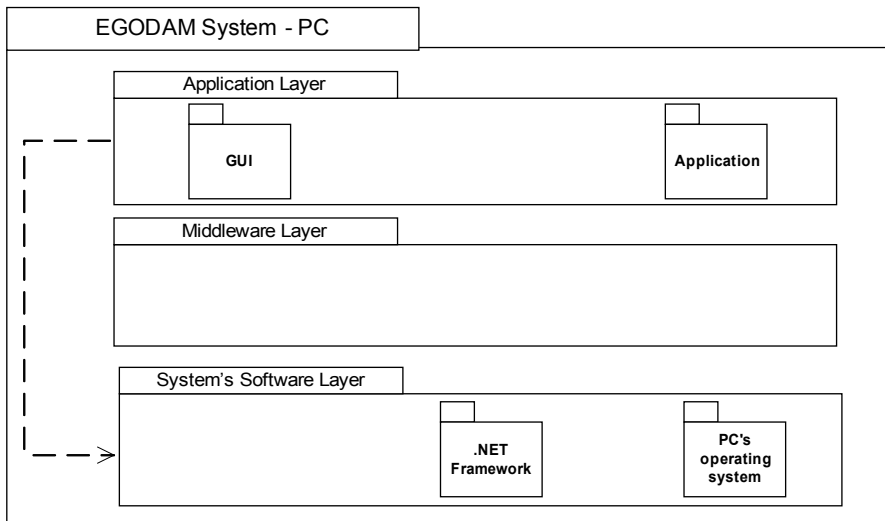
Εικόνα 3.3-34: Πλατφορμική Όψη

3.3.3.3 Δομική

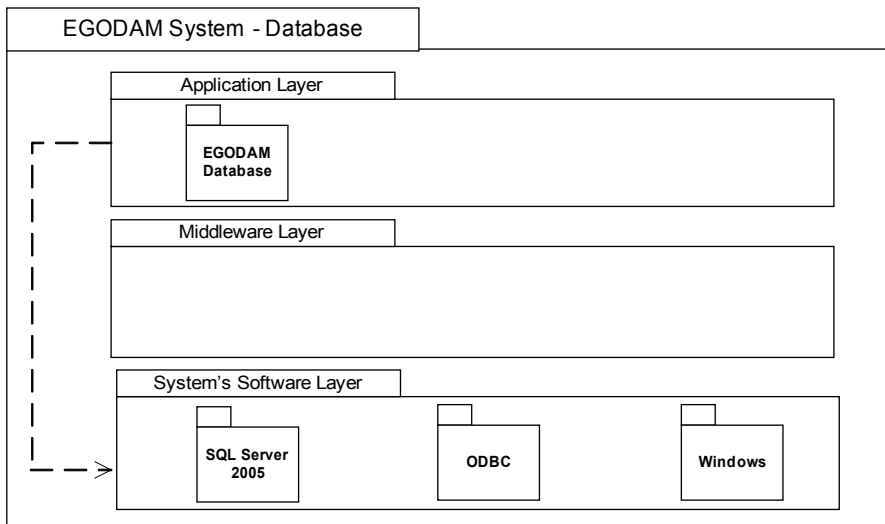
Στη δομική όψη προβάλλονται τα μέρη του συστήματος “EGODAM” και φανερώνεται ποιες είναι οι εξαρτήσεις τους. Παρατίθεται η δομική όψη του συστήματος αρχικά (επίπεδο 0) και έπειτα οι δομικές όψεις του προγράμματος και της βάσης δεδομένων από τα οποία αποτελείται το σύστημα (επίπεδο 1):



Εικόνα 3.3-35: Δομική Όψη - επίπεδο 0



Εικόνα 3.3-36: Δομική Όψη - επίπεδο 1



Εικόνα 3.3-37: Δομική όψη - επίπεδο 1 (2)

3.3.3.4 Εκτελεστική

Η εκτελεστική όψη περιγράφει τον τρόπο λειτουργίας του λογισμικού που τρέχει σε κάθε υπολογιστή. Εμφανίζεται ως ένα σύνολο εκτελέσιμων οντοτήτων (run time entities), που όλες άμεσα ή έμμεσα υποστηρίζουν την αυτοματοποίηση των δεδομενικών εργασιών του συστήματος συνεργαζόμενες μεταξύ τους, και καταγράφονται ως ένα διάγραμμα που οι κόμβοι του είναι ενεργές κλάσεις (active classes) της UML.



Εικόνα 3.3-38: Εκτελεστική Όψη

3.4 Έγγραφο Περιγραφής Λεπτομερούς Σχεδίου

3.4.1 Εισαγωγή

3.4.1.1 Σκοπός

Σκοπός του εγγράφου είναι η παρουσίαση του Λεπτομερούς Σχεδίου του λογισμικού συστήματος για την αυτοματοποιημένη μοντελοποίηση δεδομένων Ηλεκτρονικής Διακυβέρνησης.

3.4.1.2 Ενδιαφερόμενοι

Ενδιαφερόμενοι του παρόντος εγγράφου είναι οι διάφοροι δημόσιοι φορείς και υπηρεσίες που ασχολούνται με την διαχείριση και ανταλλαγή πληροφορίας Ηλεκτρονικής Διακυβέρνησης ως τελικοί χρήστες καθώς επίσης και οι αρχιτέκτονες, σχεδιαστές, προγραμματιστές και ελεγκτές ως κατασκευαστές του παρόντος λογισμικού συστήματος.

3.4.2 Σχεδιαστικές Αποφάσεις

Κατά τη λεπτομερή σχεδίαση του παρόντος λογισμικού συστήματος λήφθηκαν διάφορες αποφάσεις όσον αφορά τόσο τη διεπαφή όσο και την αναπαράσταση της πληροφορίας ανάλογα πάντα με την προς εκτέλεση λειτουργία. Παρακάτω παρατίθενται οι πιο σημαντικές σχεδιαστικές αποφάσεις που λήφθηκαν ώστε να ανταποκρίνονται στις προδιαγραφές και τα πρότυπα του συστήματος.

➤ Αποφάσεις για τη διεπαφή:

Γενικά, η δημιουργία φορμών και η προσθήκη διαφόρων εργαλείων, labels, textboxes και άλλων χαρακτηριστικών σε αυτές τις φόρμες, γίνεται με αυτόματο τρόπο από την ίδια τη γλώσσα προγραμματισμού και το περιβάλλον στο οποίο προγραμματίζουμε χωρίς γενικά να χρειάζεται να γραφεί κώδικας από τον προγραμματιστή.

Παρόλα αυτά, για την δημιουργία της συγκεκριμένης διεπαφής, η οποία υλοποιώντας ένα ευρύ σύνολο σεναρίων χρήσης, δεν είναι στατική αλλά μπορεί να έχει πολλά και διαφορετικά στιγμιότυπα και μπορεί να πάρει διάφορες μορφές, πολλές αλλαγές έπρεπε να γίνονται δυναμικά κατά τη διάρκεια εκτέλεσης του προγράμματος. Έτσι, άλλες

ετικέτες και textboxes είναι απενεργοποιημένα όταν διαχειριζόμαστε συγκεκριμένες οντότητες οι οποίες δεν έχουν κάποια συγκεκριμένα χαρακτηριστικά, ενώ όποτε χρειαστεί, αν παραδείγματος χάριν επιλεγεί η διαχείριση μιας άλλου τύπου οντότητας, τότε υπάρχει κώδικας που επανενεργοποιεί κάποια πεδία και τα κάνει ορατά στο χρήστη (visible).

Επίσης, όπως επισημάνθηκε και πιο πριν, ανάλογα με το ποια λειτουργία εκτελείται, δηλαδή create, view, edit μιας οντότητας, κάποια πεδία είναι ενεργοποιημένα και κάποια όχι. Αυτή η εναλλαγή σε ενεργοποιημένα ή όχι γίνεται επίσης δυναμικά με ειδικό κώδικα που έχει γραφτεί για την κάθε περίπτωση ώστε να μπορεί ή όχι να επέμβει ο χρήστης (enabled). Επίσης, το ποια κουμπιά θα είναι ορατά και τι κείμενο θα φέρουν είναι και αυτά προγραμματισμένο με ειδικό κώδικα έτσι ώστε να αλλάζουν δυναμικά ανάλογα με τις λειτουργίες που εκτελεί ο χρήστης.

Θα μπορούσαν βέβαια να είχαν φτιαχτεί πολλές και διαφορετικές διεπαφές, με στατικά χαρακτηριστικά και με μορφή γνωστή εκ των προτέρων, μια για κάθε διαφορετική οντότητα. Αυτό όμως θα είχε μεγάλο κόστος σε χρόνο και σε χώρο, θα ήταν πολύ δύσκολη η διαχείριση του κώδικα, θα έπρεπε με μικρές μεταβολές σε κάποιο κανόνα να αλλάζει ολόκληρη η φόρμα και γενικά το πρόγραμμα θα γινόταν δύσκολο στην συντήρηση και μελλοντική μεταβολή ή επέκτασή του, καθόλου λειτουργικό και σίγουρα με πολλές περισσότερες γραμμές επαναλαμβανόμενου κώδικα.

➤ Αποφάσεις για την υλοποίηση των κανόνων μοντελοποίησης

Το πιο σημαντικό και αυστηρό κομμάτι κώδικα όμως, που έχει γραφτεί για την συγκεκριμένη πλατφόρμα, καθώς επίσης και το πιο χρονοβόρο και αυτό που δημιούργησε τις πιο πολλές δυσκολίες όσον αφορά στη συσχέτιση των δεδομένων μεταξύ τους και στην απόδοση σημασιολογικού περιεχομένου στη διαχειριζόμενη πληροφορία, με βάση αυστηρούς και συγκεκριμένους κανόνες και σχέσεις μεταξύ των δεδομένων, ήταν αναμφισβήτητα η υλοποίηση των κανόνων μοντελοποίησης των δεδομένων.

Σε αυτό το σημείο πολύτιμος οδηγός ήταν το αρχείο e-gif repository το οποίο περιέχει όλες τις οντότητες που χειρίζεται η παρούσα πλατφόρμα με όλα τους τα χαρακτηριστικά, τις επιτρεπόμενες τους τιμές και τις σχέσεις τους με τις υπόλοιπες οντότητες με βάση το CCTS και το Ελληνικό Μοντέλο Τεκμηρίωσης. Ξεκινώντας αρχικά από τα Core Components τα οποία προκύπτουν από τα αρχικά δεδομένα με την επιβολή κατάλληλων περιορισμών, προχωρούμε έπειτα σε επίπεδο Business Information Entities, τα οποία θα προκύπτουν από τα αντίστοιχα Core Components ανάλογα με το επιχειρηματικό περιβάλλον στο οποίο εντάσσεται το έγγραφο, και τέλος, ολοκληρώνεται το έργο του λογισμικού που αναπτύσσεται, στο επίπεδο XML όπου έχουμε την τελική

δημιουργία του XML σχήματος για χρήση σε διάφορα έγγραφα Ηλεκτρονικής Διακυβέρνησης προκειμένου να ορίσει τον τύπο και τη δομή τους.

Διάφορα σημεία κώδικα που είναι ενδεικτικά για την συμμόρφωσή τους με τους κανόνες και τα πρότυπα του eGIF είναι τα εξής:

→ Ανάλογα με το τι είδους είναι η οντότητα κάποια πεδία είναι ενεργοποιημένα και κάποια όχι.

→ Ανάλογα με το αν πρόκειται για επεξεργασία ή εισαγωγή στοιχείων κάποια πεδία είναι ενεργοποιημένα και κάποια όχι.

→ Ανάλογα με τα δεδομένα που δίνει ο χρήστης, κάποια πεδία μεταβάλλονται εκείνη την ώρα και συντίθενται από τα νέα δεδομένα που εισάγει ο χρήστης.

→ Ανάλογα με το αν κάποια οντότητα σχετίζεται με κάποια άλλη (συνήθως μια οντότητα αποτελεί πρότυπο για κάποια άλλη) πρέπει να κρατούνται κάποια στοιχεία σε προσωρινές μεταβλητές σε περίπτωση που χρειαστεί να αποθηκευθεί στη βάση δεδομένων η μια οντότητα και μαζί με αυτή και το στοιχείο (συνήθως εξωτερικό κλειδί) που την συσχετίζει με την άλλη οντότητα.

→ Ανάλογα με το αν η οντότητα είναι θεμελιώδης ή όχι, κάποια πεδία συμπληρώνονται αυτόματα (η οντότητα έχει ως πρότυπό της κάποια άλλη οντότητα και κληρονομεί κάποια χαρακτηριστικά της εξ' ορισμού) ή δίνεται στον χρήστη η δυνατότητα εισαγωγής ή επεξεργασίας κάποιων πεδίων.

→ Επίσης, στο ίδιο μοτίβο, αν κάποια οντότητα έχει ως πρότυπο κάποια άλλη, κάποια πεδία της δεν είναι απαραίτητο να συμπληρώνονται αυτόματα όπως στην προηγούμενη περίπτωση αλλά μπορεί να μπορούν να πάρουν τιμές από ένα περιορισμένο σύνολο επιτρεπομένων τιμών το οποίο διαφέρει από την μια οντότητα – πρότυπο στην άλλη. Γι αυτό θα πρέπει να υπάρχει κώδικας που να ανιχνεύει ποια είναι η οντότητα πρότυπο, να ανακτά τις επιτρεπόμενες τιμές που αυτή ορίζει και να καταδεικνύει στον χρήστη από ποιο σύνολο επιτρεπομένων τιμών μπορεί να συμπληρώσει κάποιο ή κάποια πεδία της τρέχουσας οντότητας που διαχειρίζεται.

Η γενικότερη πάντως φιλοσοφία συγγραφής κώδικα σε αυτή τη φάση ανάπτυξης του προγράμματος ήταν οι αυστηροί κανόνες και η προσπάθεια να υλοποιηθούν όλες οι υποπεριπτώσεις που θα μπορούσαν να προκύψουν για εισαγωγή, επεξεργασία και διαχείριση πληροφορίας, να επιτρέπεται ή όχι η επεξεργασία κάποιων πεδίων, να παίρνουν μόνο συγκεκριμένες επιτρεπτές τιμές και κυρίως, να ανιχνεύονται, αποθηκεύονται και ανακτώνται οι σχέσεις ανάμεσα στις διάφορες οντότητες

➤ Αποφάσεις για τη βάση δεδομένων

Τέλος, αφού έγινε λόγος και για το πώς κατασκευάστηκε και χειρίζεται η διεπαφή αλλά και για το πώς εφαρμόστηκαν οι κανόνες για την προτυποποίηση της πληροφορίας, πρέπει να γίνει λόγος για το πώς γίνεται η επικοινωνία με την βάση δεδομένων η οποία χειρίζεται την προτυποποιημένη πια πληροφορία.

Σε αυτή την περίπτωση, έχει γραφεί κώδικας τέτοιος σύμφωνα με τον οποίο λειτουργούμε ως πελάτες (*clients*) ως προς τη βάση δεδομένων επιθυμώντας να ζητήσουμε σύνδεση σε αυτή (*connection*) με χρήση του ονόματος της βάσης όπως αυτό έχει δοθεί κατά τη δημιουργία και αποθήκευσή της στον υπολογιστή (*My.Settings.diplomatikiConnectionString*).

Οι δομές που είναι απαραίτητες για σύνδεση στη βάση δεδομένων και για διαχείρισή της περιέχονται στη βιβλιοθήκη System.Data.SqlClient της γλώσσας VB.NET.

Η πιο σημαντική δομή από αυτές που χρησιμοποιήθηκαν για τις συνδέσεις στην βάση δεδομένων ήταν αυτή του DataAdapter που περιέχει το query με την εντολή που θέλουμε να εκτελέσουμε στην βάση δεδομένων και μάλιστα γραμμένη σε γλώσσα sql. Η δήλωση αυτού του αιτήματος (*query*) γίνεται μέσω ενός string το οποίο περιέχει όλο τον κώδικα σε sql που χρειάζεται για την καταχώρηση νέων δεδομένων στη βάση. Έτσι, η διαχείριση της βάσης δεδομένων δεν είναι απαραίτητο να γίνεται πχ μέσω ενός Sql Server Manager, αλλά μπορεί να γίνει και μέσω μιας γλώσσας υψηλού επιπέδου, (όπως είναι η VB.NET στη συγκεκριμένη περίπτωση), αρκεί να εμφολιαστεί (*nested*) το sql query σε ένα string. Με αυτό τον τρόπο μπορεί να γίνει οποιοδήποτε είδους αίτηση (*query*) στην βάση δεδομένων.

Αυτή η δομή, μαζί με την δομή DataSet που χρησιμοποιείται σαν ενδιάμεσος buffer για την αποστολή ή ανάκτηση στοιχείων προς και από την βάση αποτελούν την πεμπουσία των συνδέσεων στη βάση δεδομένων διευκολύνοντας και απλουστεύοντας τις συναλλαγές μας με αυτή.

Ο κώδικας λοιπόν που έχει γραφεί, εκμεταλλευόμενος τις προαναφερθείσες δομές, δημιουργεί μια νέα σύνδεση στη βάση δεδομένων, χρησιμοποιώντας το όνομά της (ένα *string*). Έπειτα, ανοίγει τη σύνδεση αυτή (*open*) και έπειτα δηλώνεται ένα αίτημα καταχώρησης στη βάση.

Η καταχώρηση που ζητά λοιπόν το query, γίνεται αρχικά με συλλογή όλων των δεδομένων που υπάρχουν στα διάφορα textboxes του τρέχοντος στιγμιότυπου της διεπαφής, τα οποία ανατίθενται (*Parameters.Add*) σε συγκεκριμένες μεταβλητές οι οποίες αντιστοιχίζονται στα αντίστοιχα πεδία (*@Name*) του πίνακα της βάσης

δεδομένων όπου επιθυμούμε να αποθηκεύσουμε τα δεδομένα. Αφού γίνει αυτή η αντιστοίχιση, τότε γίνεται η εγγραφή των δεδομένων στη βάση (*fill*) και έπειτα η σύνδεση που είχε ανοιχτεί κλείνει με ειδικό κώδικα (*close*) και ο χρήστης στο τέλος ενημερώνεται για την επιτυχή εγγραφή δεδομένων στη βάση (με *MsgBox*).

Για όλες τις λειτουργίες νέας καταχώρησης στη βάση έχουν κατασκευαστεί κατάλληλες συναρτήσεις, μια για κάθε διαφορετικό είδος καταχώρησης (άλλη για τα ACC, άλλη για τα BCC κλπ) καθώς είναι διαφορετικό το είδος των παραμέτρων που πρέπει να αντιστοιχισθούν κάθε φορά και φυσικά είναι διαφορετικός και ο πίνακας της βάσης όπου θα γίνει η καταχώρηση. Η μορφή όμως και τα γενικά βήματα αυτών των συναρτήσεων είναι τα ίδια.

Για τη λειτουργία *view* της βάσης, η λογική είναι η ίδια με την προηγούμενη περίπτωση. Τα βήματα είναι σχεδόν τα ίδια για την ανάκτηση των δεδομένων, απλά αυτό που αλλάζει είναι το περιεχόμενο του *string* του *query* καθώς επίσης και το ότι τα επιστρεφόμενα δεδομένα ανατίθενται σε κάποιον πίνακα. Δηλαδή, με τη μορφή που έχει το *query*, η οποία είναι τέτοια ώστε να επιλέγονται όλα τα στοιχεία από ένα πίνακα της βάσης (*SELECT **), επιλέγονται ολόκληρες γραμμές από τους πίνακες της βάσης που ανατίθενται σε πίνακες εξωτερικούς, που ανήκουν στο κυρίως πρόγραμμα για εύκολη πρόσβαση και χρήση τους ώστε να μην ανοίγονται συνεχώς νέες συνδέσεις για ανακτήσεις δεδομένων.

Η διαδικασία της επεξεργασίας (*update*) σε γενικές γραμμές είναι όπως και στις άλλες περιπτώσεις, απλά αλλάζει πάλι το *query* και ακόμη, δεν έχουμε επιστροφή δεδομένων απλά μια πρόσβαση στη βάση για αλλαγή ή ενημέρωση όπως αλλιώς λέγεται (*update*) κάποιας καταχώρησης. Το *string* του *query* δείχνει υπό ποιες συνθήκες και σε ποιο πίνακα πρέπει να γίνει η ενημέρωση και κάνει αντιστοίχιση των νέων υπό καταχώρηση δεδομένων στις αντίστοιχες παραμέτρους που θα αντιστοιχιστούν κατά τη νέα καταχώρηση με τα κατάλληλα πεδία του πίνακα της βάσης. Μοιάζει πολύ με την νέα καταχώρηση όπως αυτή περιγράφηκε πιο πριν. Τέλος, για τη διαγραφή από τη βάση, ο κώδικας που έχει γραφτεί δημιουργεί μια νέα σύνδεση στη βάση δεδομένων και με βάση το *query* που περιέχεται στο *string* του κώδικα, γίνεται προσπέλαση της βάσης δεδομένων και εκτελείται το προαναφερθέν *query*. Οπότε, είτε θα διαγραφεί κάποιο ή κάποια στοιχεία από την βάση αν πληρούνται οι διάφορες συνθήκες που ο κώδικας επιβάλλει, αλλιώς δε θα γίνει τίποτα (αν δεν πληρούνται οι προαπαιτούμενες συνθήκες, αν το στοιχείο προς διαγραφή δεν υπάρχει ή προκύψει πχ κάποιο *exceprtion* στο πρόγραμμα).

- Χειρισμός των εξαιρέσεων – άλλων λαθών

Έχει γραφτεί ειδικός κώδικας με μπλοκ εντολών, όπως το try...catch ή το if...then...else, ο οποίος προστατεύει το πρόγραμμα από ανεπιθύμητες εξαιρέσεις (exceptions) που μπορεί να προκληθούν αν διαβαστεί κάποιο Null πεδίο από τη βάση δεδομένων ή αν γίνει κάποιο invalid cast (μετατροπή από μια μορφή δεδομένων σε μια άλλη). Προκειμένου λοιπόν να μην «κολλήσει», όπως λέμε, το πρόγραμμα και τερματιστεί μη ομαλά, χρησιμοποιείται ειδικός κώδικας αποφυγής λαθών – εξαιρέσεων για διευκόλυνση χρήσης του από τον χρήστη.

3.4.3 Σχεδιαστικές Όψεις

3.4.3.1 Αποσυνθετική

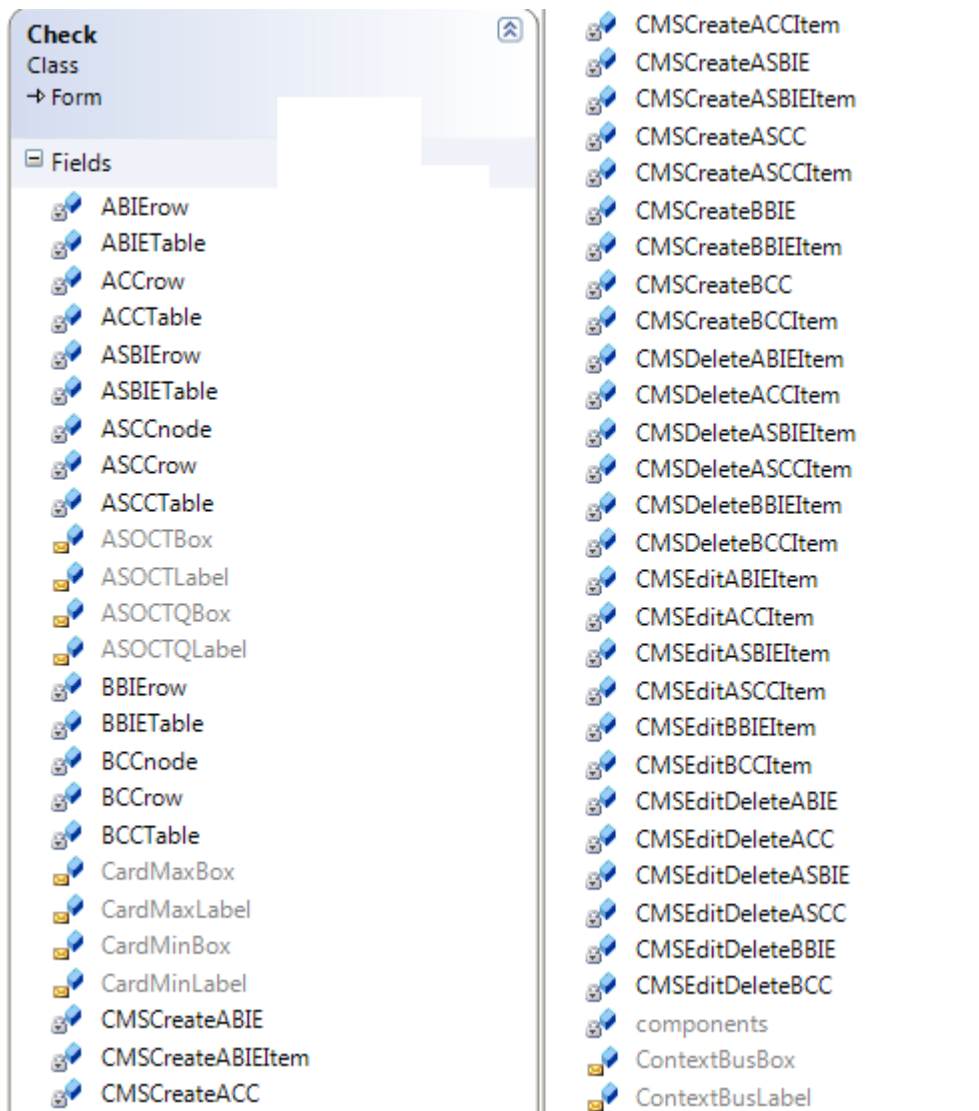
Το πρόγραμμα αποτελείται από 3 βασικές φόρμες

1. MainForm.vb
2. Check.vb
3. SelectTemplate.vb

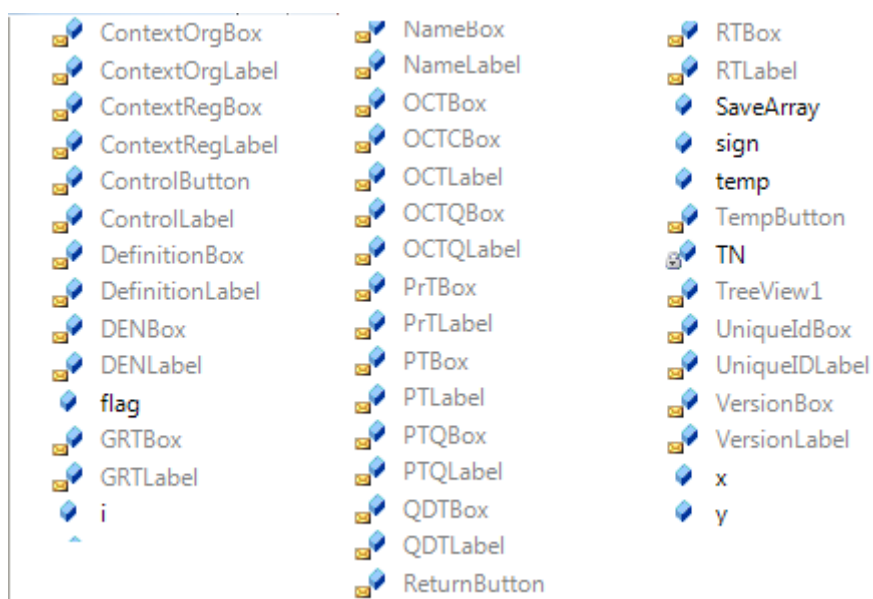
κάθε μια από τις οποίες είναι ουσιαστικά μια μεγάλη κλάση με κάποια αντικείμενα ορατά προς τις άλλες κλάσεις, κάποια άλλα μη ορατά, ενώ κάθε κλάση αποτελείται από ένα πλήθος μεθόδων οι οποίες εφαρμόζονται επάνω στα αντικείμενα (αντικειμενοστρεφής προγραμματισμός).

Διαγράμματα των κλάσεων όπως αυτά ελήφθησαν από το Visual Studio .NET όπου υλοποιήθηκε το πρόγραμμα:

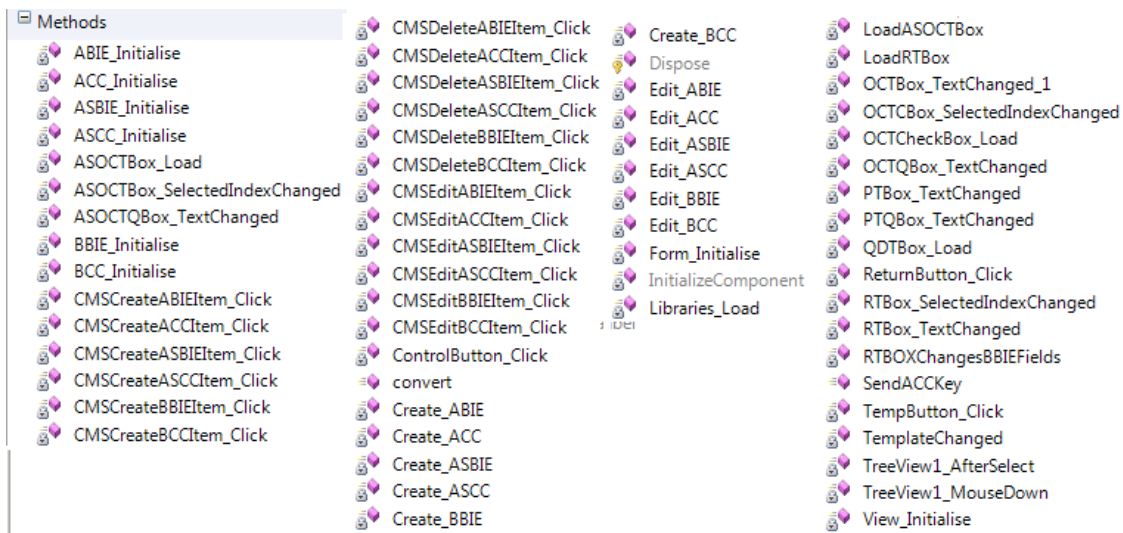
(φαίνονται οι μέθοδοι και τα αντικείμενα)



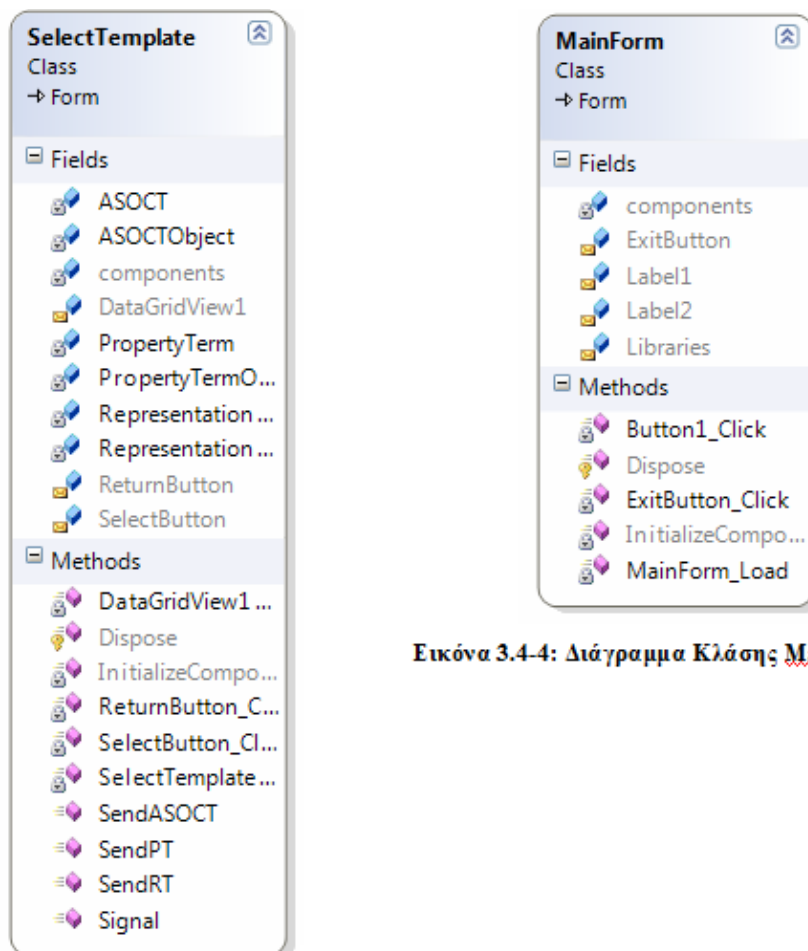
Εικόνα 3.4-39: Διάγραμμα Κλάσης Check.vb (1) - fields



Εικόνα 3.4-40: Διάγραμμα κλάσης Check.vb (2) - fields



Εικόνα 3.4-41: Διάγραμμα Κλάσης Check.vb – Methods

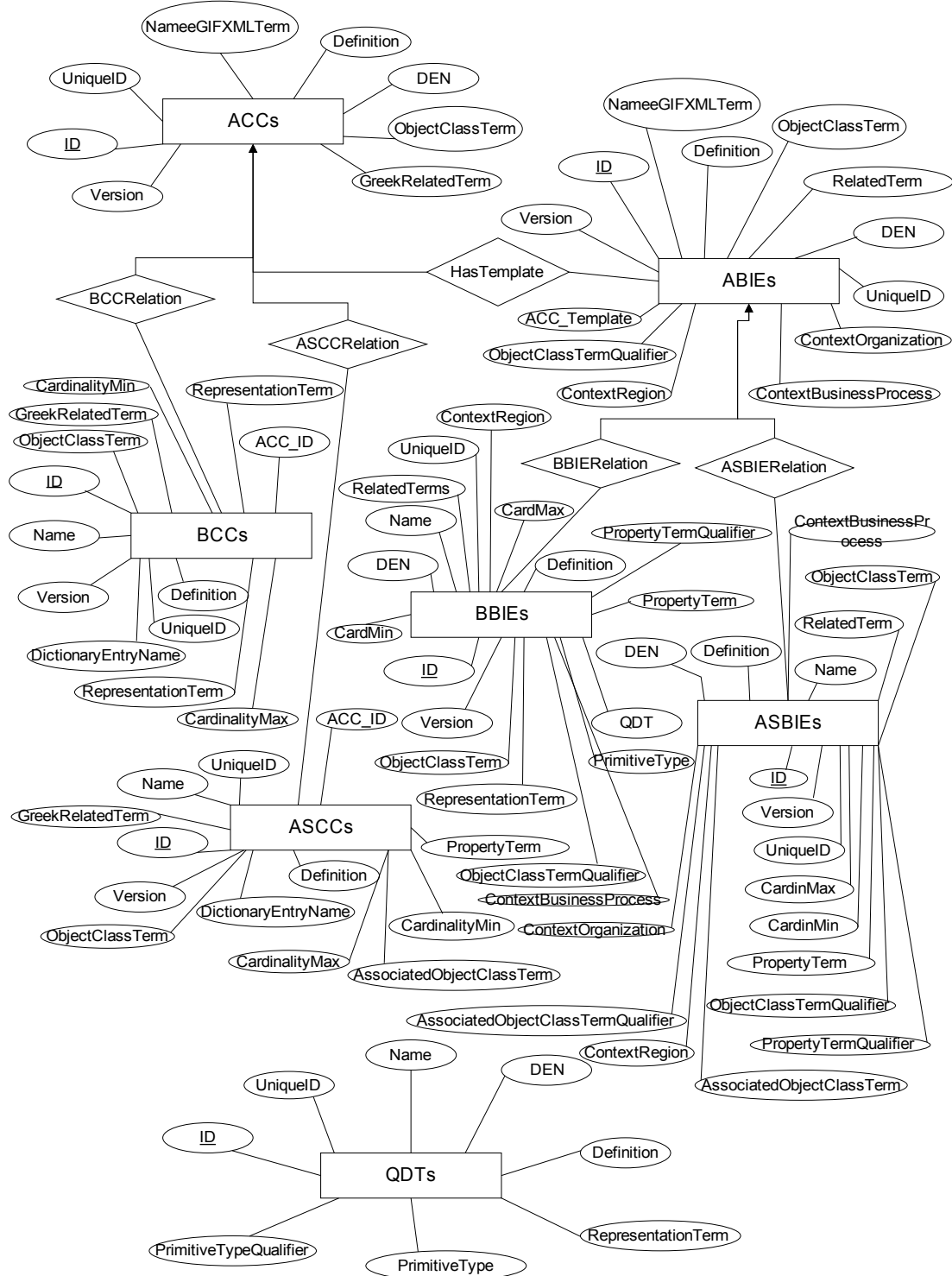


Εικόνα 3.4-4: Διάγραμμα Κλάσης MainForm.vb

Εικόνα 3.4-5: Διάγραμμα Κλάσης SelectTemplate.vb

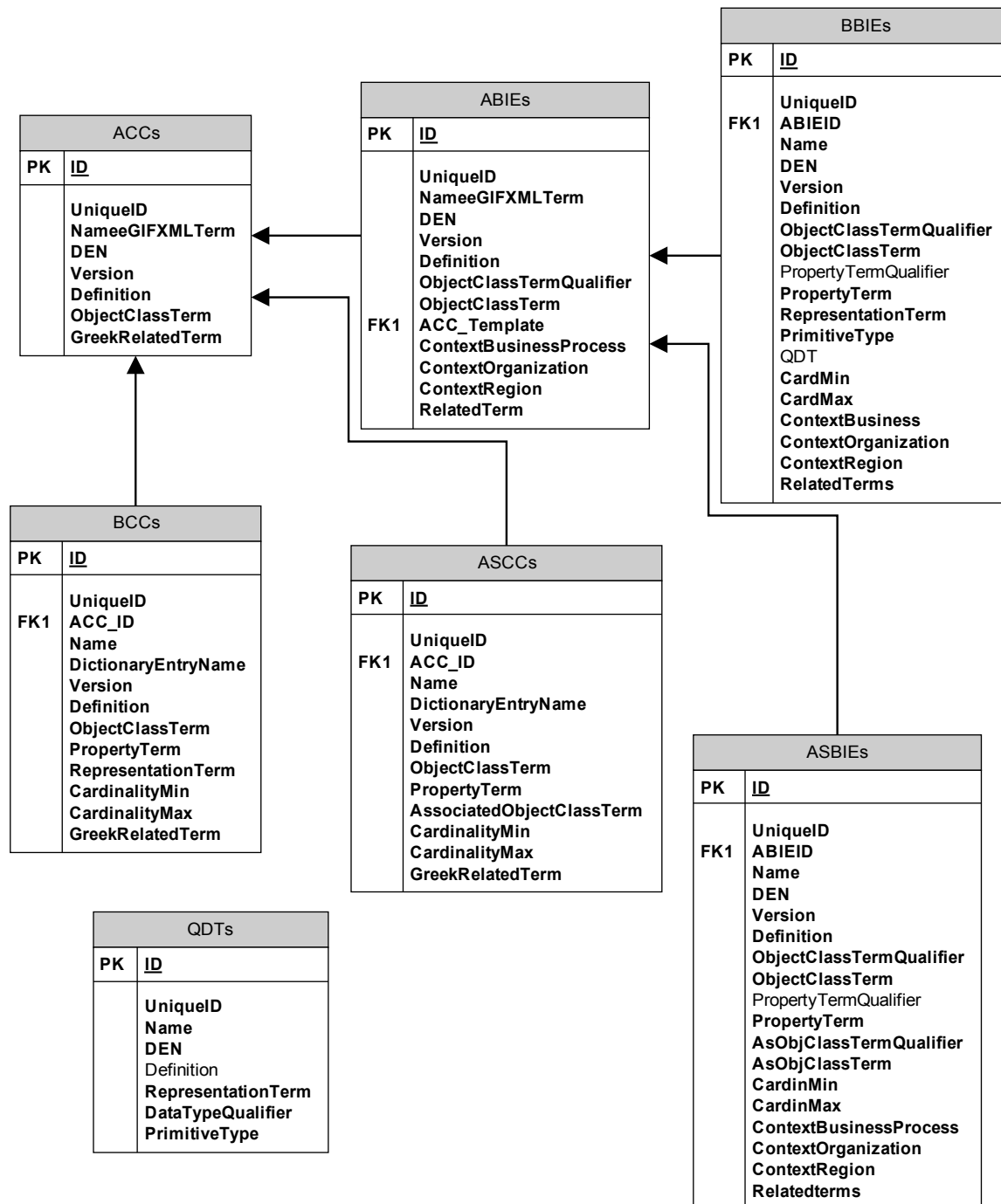
3.4.3.2 Δεδομενική

Στη δεδομενική όψη εμφανίζεται το σχήμα της βάσης δεδομένων που θα χρησιμοποιηθεί και το οποίο θα περιγραφεί επιπρόσθετα ως διάγραμμα οντοτήτων – συσχετίσεων (Entity – Relationship) στα σχήματα που ακολουθούν. Δεδομενικό Διάγραμμα – Διάγραμμα οντοτήτων – συσχετίσεων



Εικόνα 3.4-6: Διάγραμμα Οντότητας - Συσχέτισης

Δεδομενικό Διάγραμμα – Σχήμα Βάσης Δεδομένων

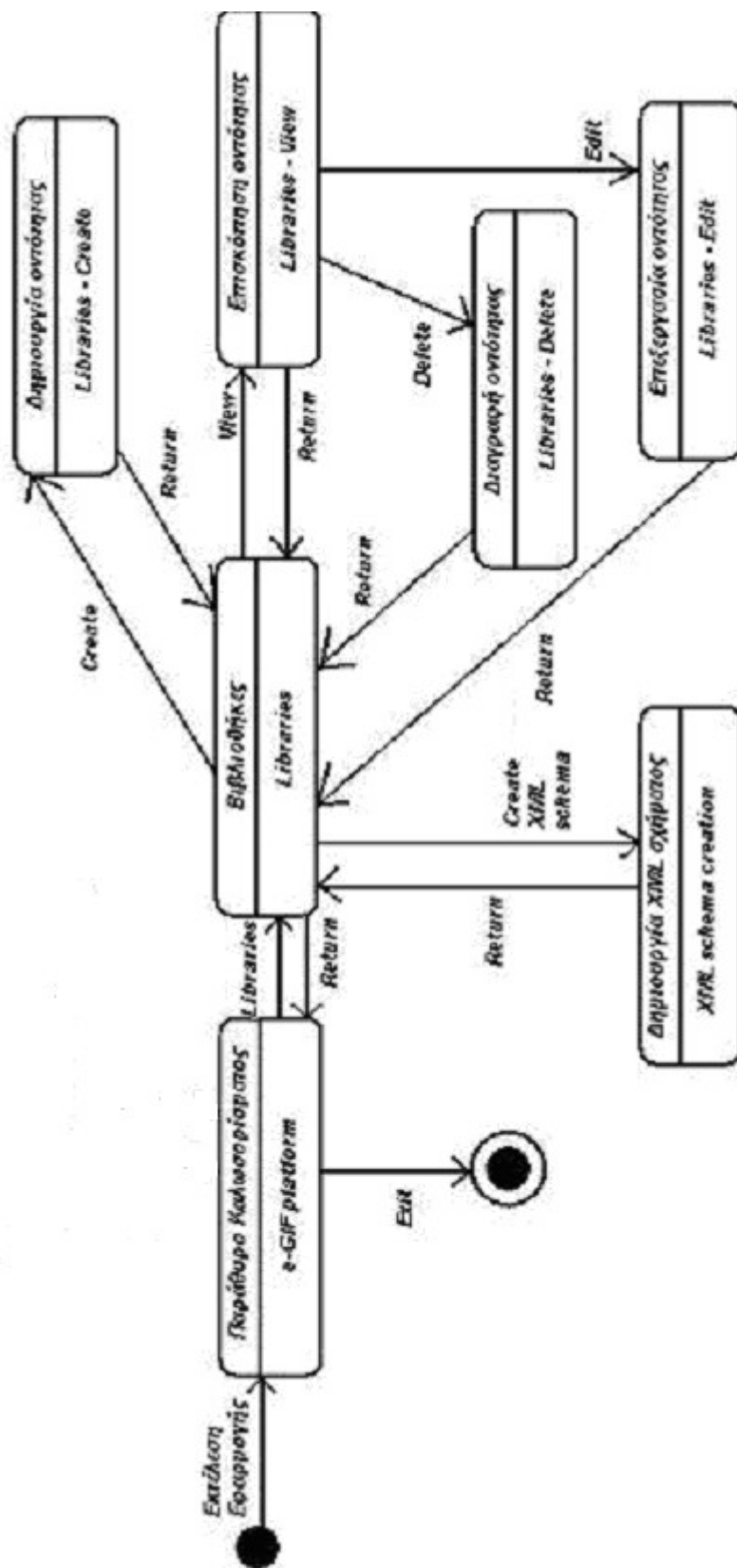


Εικόνα 3.4-7: Διάγραμμα Οντότητας – Συσχέτισης (2)

3.4.3.3 Συνεργατική

Στην παρούσα όψη φαίνεται πώς συνεργάζονται τα σχεδιαστικά συνθετήματα για την εκτέλεση των εργασιών που αυτοματοποιεί το υπό κατασκευή λογισμικό σύστημα.

Διάγραμμα καταστάσεων:



Εικόνα 3.4-8: Ακολουθιακό Διάγραμμα Συστήματος

4

Υλοποίηση – Έλεγχος

Στο κεφάλαιο 4, πραγματοποιείται μια παρουσίαση των εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη του παρόντος λογισμικού συστήματος, δηλαδή του Microsoft Visual Studio 2005, του .NET Framework 1.1, και του Microsoft SQL Server 2000. Παράλληλα, γίνεται μια σύντομη ανασκόπηση της γλώσσας προγραμματισμού Visual Basic .NET και της τεχνολογίας ADO.NET. Ακολουθούν οι λεγόμενες λεπτομέρειες υλοποίησης με την παρουσίαση των διαδικασιών που δημιουργήθηκαν. Το κεφάλαιο ολοκληρώνεται με το έγγραφο Τεκμηρίωσης Δυναμικού Ελέγχου, το οποίο εξασφαλίζει την ποιότητα του τελικού προϊόντος με την επικύρωση των προδιαγραφών των απαιτήσεων, την επαλήθευση του Αρχιτεκτονικού και Λεπτομερούς Σχεδίου και την επαλήθευση του κώδικα.

4.1 Εργαλεία υλοποίησης

4.1.1 Visual Studio .NET 2005

Το Visual Studio .NET 2005 είναι ένα ενοποιημένο περιβάλλον ανάπτυξης (IDE) και παρέχει μια συνεπή διασύνδεση για όλες τις γλώσσες, συμπεριλαμβανομένων των Microsoft Visual Basic .NET, Microsoft Visual C++ .NET, Microsoft Visual C# .NET και Microsoft Visual J# .NET.

Χρησιμοποιώντας λοιπόν τη Visual Basic .NET ως γλώσσα συγγραφής κώδικα, επωφελείται κανείς από τα παραστατικά προγράμματα σχεδίασης για τη δημιουργία εμπλουτισμένων εφαρμογών που βασίζονται στα Windows και δυναμικών εφαρμογών Web οι οποίες αποδίδουν σε οποιοδήποτε πρόγραμμα περιήγησης.

Το Visual Studio .NET 2005 περιέχει μια ενημερωμένη έκδοση του .NET Framework, την έκδοση 1.1, η οποία βασίζεται στην προηγούμενη έκδοση, με νέες δυνατότητες και βελτιωμένη δυνατότητα κλιμάκωσης, αξιοπιστία, ασφάλεια και επιδόσεις. Αυτές οι κύριες επιδόσεις στο υποκείμενο πλαίσιο ανάπτυξης συνδυάζονται με τη βελτιωμένη ανταπόκριση του IDE, ώστε να είναι δυνατή η μείωση του κόστους που έχει σχέση με τον προγραμματισμό, την ανάπτυξη και τη συντήρηση εφαρμογών.

Η αντιμετώπιση των αναδυόμενων ευκαιριών που παρέχουν οι υπηρεσίες Web της XML και η ανάπτυξη εφαρμογών για φορητούς υπολογιστές ήταν οι κύριοι στόχοι στη σχεδίαση του Visual Studio .NET 2005. Επιτρέπει τη δημιουργία και χρησιμοποίηση υπηρεσιών Web και παρέχει διαισθητικά προγράμματα σχεδίασης για γρήγορη δημιουργία εφαρμογών Web ευρείας χρήσης για φορητές συσκευές, καθώς και λογισμικού για έξυπνες συσκευές.

ΔΥΝΑΤΟΤΗΤΕΣ

➤ Σχεδίαση φορμών των Windows

Επιταχύνει και απλοποιεί τη δημιουργία ισχυρών εφαρμογών που βασίζονται στα Windows.

➤ Τεχνολογία IntelliSense

Παρέχει ολοκλήρωση εντολών μέσα στο πρόγραμμα επεξεργασίας κώδικα και το παράθυρο Immediate.

➤ Ενσωματωμένη ασφάλεια

Περιορίζει με ασφάλεια την πρόσβαση στις εφαρμογές μόνο για εξουσιοδοτημένους χρήστες, ενώ η δυνατότητα ασφάλειας της πρόσβασης σε κώδικα επιτρέπει την ασφαλή λήψη πολλών εμπλουτισμένων εφαρμογών .NET προγράμματος-πελάτη από το Internet.

➤ Σχεδίαση φορμών Web

Παρέχει παραστατική δημιουργία επαγγελματικών εφαρμογών Web.

➤ Υποστήριξη υπηρεσιών Web

Επιτρέπει στους προγραμματιστές να δημιουργήσουν γρήγορα και να χρησιμοποιήσουν υπηρεσίες Web που βασίζονται σε πρότυπα, οι οποίες προωθούν την ενοποίηση εφαρμογών.

➤ **Υπηρεσίες παροχής διαχειριζόμενων δεδομένων**

Παρέχουν στοιχεία ελέγχου δεδομένων για ομαλές συνδέσεις με προελεύσεις δεδομένων OLEDB και ODBC, όπως ο SQL Server, η Access, η Oracle, κ.λπ.

➤ **Υποστήριξη για το .NET Framework έκδοσης 1.1**

Παρέχει πλήρη, άμεση πρόσβαση στις εμπλουτισμένες βιβλιοθήκες κλάσεων που υπάρχουν στο .NET Framework.

➤ **Υποστήριξη εφαρμογών για φορητούς υπολογιστές**

Επιτρέπει τη χρησιμοποίηση των τρεχουσών δυνατοτήτων για τη δημιουργία εμπλουτισμένου λογισμικού για φορητούς υπολογιστές και εφαρμογές Web ευρείας χρήσης για φορητές συσκευές.

4.1.2 VISUAL BASIC .NET

Γενικά, η VISUAL BASIC .NET είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού που παρέχει τη δυνατότητα σύνταξης αξιόπιστου και ισχυρού λογισμικού. Ιδιαίτερα μέσα από την πλατφόρμα VISUAL STUDIO .NET που παρέχει πλήθος εργαλείων σχεδιασμού και προγραμματισμού, καθίσταται ένα εύχρηστο, γρήγορο και αποδοτικό εργαλείο για την ανάπτυξη και συντήρηση αυτόνομων και εύχρηστων εφαρμογών (Windows Application) αλλά και εφαρμογών στο Internet (Web Application) ή υπηρεσιών στο Web (Web Services).

Γίνεται λοιπόν αμέσως αντιληπτή η σημασία χρήσης μιας τέτοιας γλώσσας για την υλοποίηση του παρόντος λογισμικού συστήματος καθώς όχι μόνο παρέχεται η δυνατότητα κατασκευής μιας αυτόνομης πλατφόρμας για Windows η οποία αποθηκεύει, διαχειρίζεται πληροφορία και παράγει έγγραφα Ηλεκτρονικής Διακυβέρνησης, αλλά παρέχεται και η δυνατότητα η πλατφόρμα αυτή, με την απόδοση συγκεκριμένων χαρακτηριστικών μέσω συγγραφής συγκεκριμένου κώδικα, να γίνει μια πλατφόρμα που θα μπορεί να γίνει ενεργή και σε επίπεδο διαδικτύου ως ένα ολοκληρωμένο Web Service υλοποιώντας την έννοια της Διαλειτουργικότητας και εκπληρώνοντας πλήρως τις ανάγκες της όπως αυτές περιγράφηκαν παραπάνω.

Η επιλογή της λοιπόν ως εργαλείο για την ανάπτυξη της παρούσας πλατφόρμας ήταν πολύ σημαντική καθώς όχι μόνο έχει ως τελικό προϊόν εφαρμογές ισχυρές, εύχρηστες και αποδοτικές αλλά και σε επίπεδο προγραμματισμού και σχεδίασης, προσφέρει πολλά πλεονεκτήματα για τον προγραμματιστή, μερικά από τα οποία παρατίθενται παρακάτω:

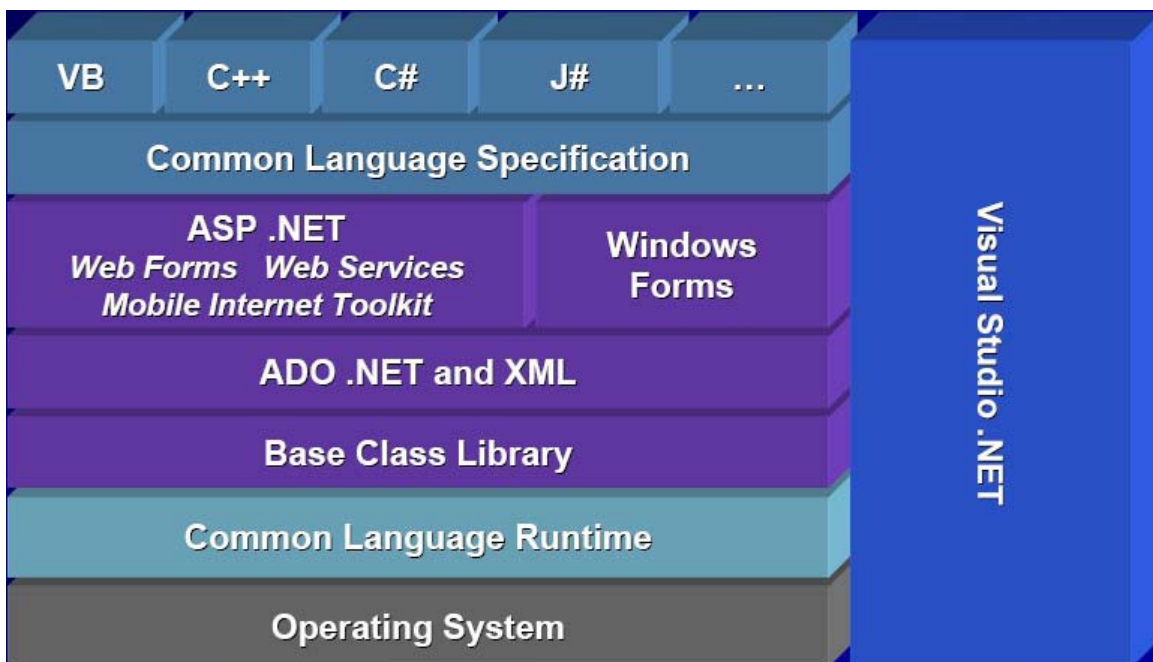
- **Προσφέρει λύση στα σύγχρονα προβλήματα ανάπτυξης πιο γρήγορα**
Επιτρέπει την γρήγορη δημιουργία δυναμικών εφαρμογών Windows και Web, την εύκολη ανάπτυξη και συντήρηση εφαρμογών ενώ παράλληλα, επιτρέπει την πρόσβαση σε δεδομένα από ένα ευρύ φάσμα σεναρίων.
- **Δίνει τη δυνατότητα δημιουργίας ισχυρών εφαρμογών των Windows σε λιγότερο χρόνο**
Δηλαδή, δημιουργούνται πιο άμεσες, πιο ισχυρές και πιο εύχρηστες εφαρμογές, με δυνατότητα αγκύρωσης στοιχείων ελέγχου, πρόγραμμα άμεσης επεξεργασίας μενού και πολλά άλλα.
- **Επιτρέπει τη γρήγορη ανάπτυξη εφαρμογών για το Web**
Η χρησιμοποίηση της σχεδίασης φορμών Web, συμβάλλει στην γρήγορη δημιουργία πραγματικών εφαρμογών Web με προγράμματα – πελάτες που επεξεργάζονται μικρή ποσότητα δεδομένων, οι οποίες αποδίδουν έξυπνα σε οποιοδήποτε πρόγραμμα περιήγησης και σε οποιαδήποτε πλατφόρμα.
- **Διευκολύνει την ανάπτυξη και συντήρηση εφαρμογών**
Οι δυσκολίες με τα αρχεία DLL έχουν πλέον ξεπεραστεί με την παράλληλη διατήρηση εκδόσεων, την ανάπτυξη XCOPY και την αυτόματη λήψη εφαρμογών Windows.
- **Εξασφαλίζει γρήγορη ανάπτυξη εφαρμογών για φορητές συσκευές Web**
Υποστηρίζει περισσότερες από 200 φορητές συσκευές Web, όπως κινητά τηλέφωνα, συσκευές τηλεϊδοποίησης και υπολογιστές PDA.

4.1.3 .NET Framework 2.0

Το Microsoft .NET Framework είναι η καρδιά της στρατηγικής .NET. Ως πλαίσιο εργασίας διαχειρίζεται και εκτελεί εφαρμογές και υπηρεσίες Web, περιέχει μια βιβλιοθήκη κλάσεων (τη Framework Class Library, FCL), ενισχύει την ασφάλεια και παρέχει πολλές άλλες προγραμματιστικές δυνατότητες. Οι λεπτομέρειες του .NET Framework δίνονται στο Common Language Specification (CLS), που παρέχει πληροφορίες μεταξύ άλλων για την αποθήκευση τύπων δεδομένων και τα αντικείμενα.

Τα προγράμματα μεταγλωττίζονται σε οδηγίες συγκεκριμένες για τη μηχανή σε δύο βήματα: πρώτα, το πρόγραμμα μεταγλωττίζεται σε ενδιάμεσο κώδικα (bytecode) γραμμένο σε Microsoft Intermediate Language (MSIL), ο οποίος ουσιαστικά ορίζει οδηγίες για το Common Language Runtime (CLR) που λειτουργεί ως εικονική μηχανή

(virtual machine). Κώδικας που έχει μετατραπεί σε MSIL από άλλες γλώσσες και πηγές μπορεί να «συνυφανθεί» με αυτό από το CLR. Κατόπιν, ένας άλλος μεταγλωττιστής στο CLR μεταφράζει τη MSIL σε κώδικα μηχανής για μια συγκεκριμένη πλατφόρμα, δημιουργώντας μια μόνο εφαρμογή. Η αρχιτεκτονική αυτή των δύο βημάτων παρέχει δυνατότητα μεταφοράς μεταξύ των λειτουργικών συστημάτων, Διαλειτουργικότητα ανάμεσα στις γλώσσες και έλεγχο πάνω στα χαρακτηριστικά διαχείρισης της εκτέλεσης, όπως στη διαχείριση μνήμης 112 και την ασφάλεια. Η αρχιτεκτονική του .NET Framework παρουσιάζεται λεπτομερώς στο επόμενο σχήμα:



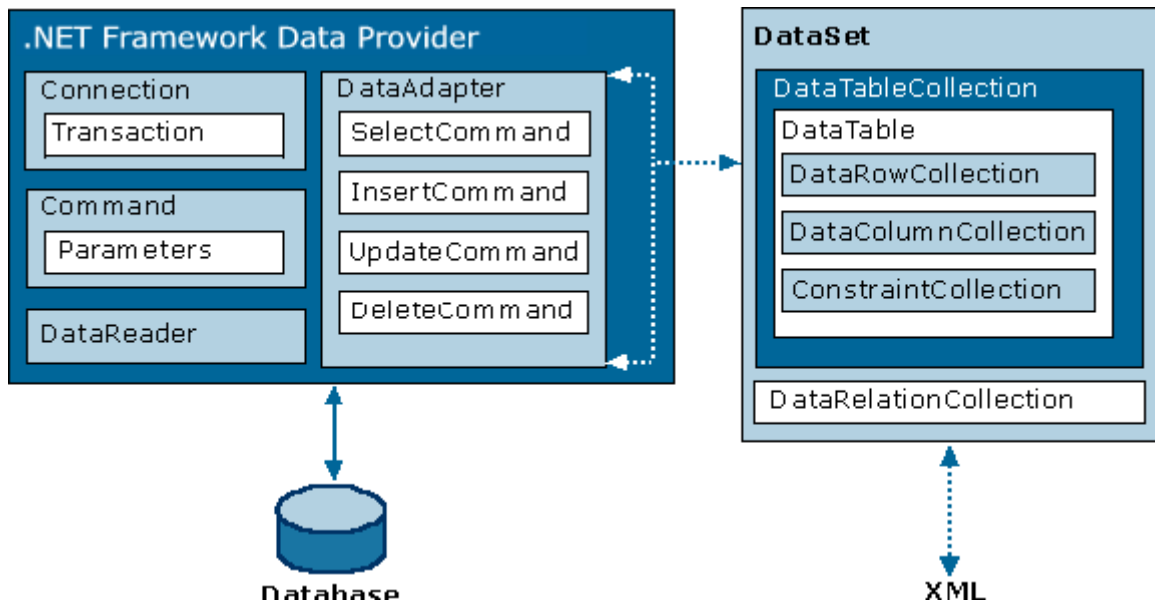
Εικόνα 4.1-42: .NET Framework

4.1.4 ADO .NET

Το μοντέλο αντικειμένων ADO .NET παρέχει ένα API (Application Programming Interface) για πρόσβαση σε συστήματα βάσεων δεδομένων είτε είναι συνδεδεμένα με το πλαίσιο εργασίας είτε όχι. Δημιουργήθηκε για το .NET Framework και είναι η επόμενη γενιά των ActiveX Data Objects (ADO), που σχεδιάστηκαν για αλληλεπίδραση με το Component Object Model (COM) της Microsoft. Γενικά, το ADO.NET παρέχει ένα επίπεδο ανάμεσα στις 113 γλώσσες προγραμματισμού ανώτερου επιπέδου και τις βάσεις δεδομένων, το οποίο επιτρέπει στους προγραμματιστές να γράφουν προγράμματα που να έχουν πρόσβαση σε βάσεις δεδομένων χωρίς να γνωρίζουν πώς αυτές υλοποιούνται. Στις καινοτομίες του ADO.NET σε σχέση με τα

ADO εντάσσονται: η υποστήριξη του n-tier μοντέλο, η πλήρης ενσωμάτωση της αναπαράστασης δεδομένων σε XML και η θεώρησή του ως βιβλιοθήκη κώδικα. Τα κυριότερα αντικείμενα που συνιστούν το ADO.NET είναι:

- *Connection*: Καθορίζει τις ρυθμίσεις για τη σύνδεση σε μια συγκεκριμένη πηγή δεδομένων που τρέχει πάνω σε οποιοδήποτε σύστημα διαχείρισης βάσης δεδομένων, όπως σε MS SQL Server, Oracle ή MS Access.
- *Command*: Περιέχει τις εντολές σε SQL που αποστέλλονται προς τη βάση δεδομένων (εντολές ανάκτησης ή ενημέρωσης δεδομένων, κλήσεις σε αποθηκευμένες διαδικασίες).
- *DataReader*: Δημιουργείται όταν εκτελεστεί μια εντολή μέσω του αντικειμένου Command και παρέχει ένα σειριακό τρόπο ανάγνωσης δεδομένων μόνο προς τη μια κατεύθυνση και χωρίς δυνατότητα τροποποίησής τους.
- *DataSet*: Αναπαριστά ένα “cache” δεδομένων, δηλαδή δεδομένα που το πρόγραμμα αποθηκεύει προσωρινά στην τοπική μνήμη. Η δομή του μιμείται τη δομή μιας σχεσιακής βάσης δεδομένων, καθώς αποτελείται από οποιονδήποτε αριθμό DataTables, που αντιπροσωπεύουν πίνακες δεδομένων, από τους περιορισμούς και τις συσχετίσεις που τους διέπουν. Τα DataSets αποθηκεύονται ως έγγραφα XML στη μνήμη και η περιγραφή της δομής τους μπορεί να πραγματοποιηθεί με XML Schemas. Ένα ακόμα πλεονέκτημα της χρήσης του αντικειμένου DataSet είναι ότι είναι αποσυνδεδεμένο – το πρόγραμμα δεν χρειάζεται μια μόνιμη σύνδεση στην προέλευση δεδομένων για να εργαστεί με δεδομένα σε ένα DataSet. Αντίθετα, το πρόγραμμα συνδέεται στην προέλευση δεδομένων μόνο κατά το αρχικό συμπλήρωμα του DataSet και κατόπιν για την αποθήκευση οποιονδήποτε αλλαγών γίνονται σε αυτό.
- *DataAdapter*: Επιτρέπει τη συμπλήρωση και την ενημέρωση των περιεχομένων ενός DataSet κρύβοντας τις λεπτομέρειες που αφορούν το σύνολο των απαραίτητων εντολών και τη σύνδεση με τη βάση δεδομένων. Τα DataAdapters περιέχουν 4 ιδιότητες τύπου Command για τη διαχείριση δεδομένων: DeleteCommand για την διαγραφή δεδομένων, InsertCommand για την εισαγωγή νέων δεδομένων, SelectCommand για την ανάκτηση δεδομένων και UpdateCommand για την ενημέρωση δεδομένων. Η αρχιτεκτονική του ADO .NET απεικονίζεται στο επόμενο σχήμα :



Εικόνα 4.1-43: ADO.NET Architecture

4.1.5 Microsoft SQL Server 2005

Ένα πολύ σημαντικό κομμάτι στην υλοποίηση της πλατφόρμας είναι η κατασκευή μιας βάσης δεδομένων στην οποία αποθηκεύονται τα κωδικοποιημένα πια δεδομένα με συστηματικό και μεθοδικό τρόπο ώστε να καθίσταται δυνατή η μεταξύ τους συσχέτιση αλλά και η ανάκτησή τους. Η ανάκτησή τους γίνεται από το κυρίως πρόγραμμα το οποίο διαχειρίζεται τη βάση δεδομένων και το οποίο επιτρέπει την αποθήκευση, ανάκτηση, επεξεργασία, ενημέρωση και διαγραφή δεδομένων και κωδικοποιημένης πληροφορίας. Τέλος, για την κατασκευή του τελικού XML σχήματος, το κυρίως πρόγραμμα προσπελαύνει τη βάση, κάνει αναζήτηση και ανάκτηση των επιθυμητών δεδομένων και παράγει το τελικό έγγραφο (XML schema).

Για να είναι οι παραπάνω διαδικασίες εύχρηστες, ασφαλείς και κυρίως αποδοτικές, χρησιμοποιήθηκε ο Microsoft SQL Server 2005 ο οποίος καλύπτει τις παραπάνω ανάγκες καθώς προσφέρει πολλές δυνατότητες κατά το σχεδιασμό, ανάπτυξη και διαχείριση βάσεων δεδομένων μερικές από τις οποίες επισημαίνονται παρακάτω:

Δυνατότητες

➤ Υποστήριξη για πρότυπα του Internet

Ενσωμάτωση υπαρχόντων συστημάτων χωρίς προγραμματισμό, μέσω ολοκληρωμένης υποστήριξης για πρότυπα W3C, συμπεριλαμβανομένων των XML, Xpath, XSL και HTTP.

➤ Μεταβίβαση έπειτα από αποτυχία συμπλέγματος

Εγκατάσταση βάσεων δεδομένων προετοιμασμένων για μεταβίβαση έπειτα από αποτυχία απευθείας από το πρόγραμμα Εγκατάστασης. Χρήση ενεργητικής-παθητικής μεταβίβασης έπειτα από αποτυχία με εφεδρικό υλικό, ή ενεργητικής-ενεργητικής μεταβίβασης έπειτα από αποτυχία σε περιβάλλοντα με περιορισμούς υλικού. Δυνατότητα μεταβίβασης έπειτα από αποτυχία των βάσεων δεδομένων σε οποιονδήποτε παραμένοντα κόμβο με σύστημα μεταβίβασης έπειτα από αποτυχία με 4 κόμβους.

➤ Ασφάλεια

Προστασία δεδομένων με υψηλότερη προεπιλεγμένη ασφάλεια κατά την εγκατάσταση. Περιλαμβάνει υποστήριξη για συνδέσεις Secure Sockets Layer (SSL) και πρωτόκολλα Kerberos. Χορήγηση πιστοποιητικού C2 σε εξέλιξη.

➤ Νέοι τύποι δεδομένων

Ευέλικτη αποθήκευση και αναφορά δεδομένων με τύπους δεδομένων bigint, sql_variant και πίνακα.

➤ Αυτο-διαχείριση και ρύθμιση

Ο SQL Server 2000 χρησιμοποιεί δυναμικούς αλγορίθμους για αυτόματη προσαρμογή σε περιβάλλοντα όπου ο χρόνος εκτέλεσης μεταβάλλεται διαρκώς.

➤ Ανάλυση ερωτημάτων SQL

Εντοπισμός σφαλμάτων σε αποθηκευμένες διαδικασίες. Ορισμός σημείων διακοπής, καθορισμός παρακολούθησης, προβολή μεταβλητών και παρέμβαση στον κώδικα. Εντοπισμός του κώδικα εκτέλεσης σε διακομιστή ή πρόγραμμα-πελάτη. Εύκολη εγγραφή T-SQL με βάση πρότυπα.

➤ Συναρτήσεις που ορίζονται από το χρήστη

Εκ νέου χρήση του κώδικα μέσω της δημιουργίας συναρτήσεων T-SQL. Ενσωμάτωση στερεότυπης λογικής για απλοποιημένη ανάπτυξη.

Η σχεσιακή γλώσσα που χρησιμοποιεί ο SQL Server για την επικοινωνία των εφαρμογών με το DBMS ονομάζεται Transact – SQL (T – SQL). Είναι μια διάλεκτος της πιο σημαντικής γλώσσας βάσεων δεδομένων που υπάρχει σήμερα : της Δομημένης Γλώσσας Ερωτημάτων SQL (Structured Query Language). Σε γλώσσα T – SQL γράφονται οι αποθηκευμένες διαδικασίες (stored procedures), οι οποίες ουσιαστικά είναι μια αλληλουχία προτάσεων και επεκτάσεων SQL που αποθηκεύονται στον διακομιστή της βάσης δεδομένων και εκτελούνται αδιαίρετα. Μια αποθηκευμένη διαδικασία μεταγλωττίζεται εκ των προτέρων πριν αποθηκευτεί σαν αντικείμενο μέσα στη βάση. Με αυτόν τον τρόπο, το σχέδιο εκτέλεσης μιας διαδικασίας αποθηκεύεται στη βάση δεδομένων και εκτελείται όταν εκτελείται η αποθηκευμένη διαδικασία. Η επαναλαμβανόμενη μεταγλώττιση μιας διαδικασίας σχεδόν πάντα εξαιρείται και η απόδοση της εκτέλεσης αυξάνεται. Επίσης, οι αποθηκευμένες διαδικασίες που παρακολουθούν τις λειτουργίες εγγραφής ή / και ανάγνωσης που αφορούν ένα πίνακα, είναι ένα πρόσθετο χαρακτηριστικό ασφάλειας της βάσης δεδομένων, καθώς ο διαχειριστής μπορεί να παρακολουθεί τροποποιήσεις από χρήστες ή από προγράμματα εφαρμογών.

Στην παρούσα εφαρμογή χρησιμοποιήθηκε ο SQL Server 2005 καθώς επίσης και ένα ακόμα εργαλείο για διαχείριση της βάσης δεδομένων, το SQL Server Management Studio Express το οποίο διευκολύνει τη δημιουργία και επεξεργασία των διαφόρων πινάκων της βάσης δεδομένων.

4.2 Λεπτομέρειες Υλοποίησης

Σε αυτό το σημείο παρατίθενται όλες αυτές οι μέθοδοι που εφαρμόζονται στα διάφορα αντικείμενα. Έχουν καταχωρηθεί στην παρακάτω λίστα και δίπλα σημειώνεται η φόρμα στην οποία ανήκουν και μια σύντομη αναφορά ανάλογα με το αν υλοποιεί ο κώδικάς τους κάποιο από τα παρακάτω:

1. Χειρισμό διεπαφής
2. Υλοποίηση κανόνων προτυποποίησης
3. Χειρισμό βάσης δεδομένων
4. Παραγωγή τελικού XML Σχήματος

Μέθοδος	Φόρμα	Ρόλος
Private Sub Libraries_Load	Check.vb	Φόρτωση βιβλιοθηκών, επεξεργάζεται διεπαφή, υλοποιεί κανόνες προτυποποίησης, διαχειρίζεται τη βάση δεδομένων
Private Sub TreeView1_AfterSelect	Check.vb	Χειρίζεται την εμφάνιση του δέντρου, επεξεργάζεται διεπαφή, υλοποιεί κανόνες προτυποποίησης
Private Sub TreeView1_MouseDown	Check.vb	Χειρίζεται τη διεπαφή
Private Sub ReturnButton_Click	Check.vb	Μέθοδος που όταν πατηθεί return, κλείνει την τρέχουσα φόρμα
Private Sub CMSCreateACCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει δημιουργία ACC – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSEditACCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει επεξεργασία ACC – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSDeleteACCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει διαγραφή ACC – συνδέεται στη βάση και διαγράφει το επιλεγμένο ACC και τα συσχετιζόμενα με αυτό BCC & ASCC
Private Sub CMSCreateBCCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει δημιουργία BCC – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSEditBCCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει επεξεργασία BCC – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSDeleteBCCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει διαγραφή BCC – συνδέεται στη βάση και διαγράφει το επιλεγμένο BCC
Private Sub CMSCreateASCCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει δημιουργία ASCC – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSEditASCCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει επεξεργασία ASCC – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSDeleteASCCItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει διαγραφή ASCC – συνδέεται στη βάση και διαγράφει το επιλεγμένο ASCC
Private Sub CMSCreateABIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει δημιουργία ABIE – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSEditABIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει επεξεργασία ABIE – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSDeleteABIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει διαγραφή ABIE – συνδέεται στη βάση και διαγράφει το επιλεγμένο ABIE και τα συσχετιζόμενα με αυτό BBIE & ASBIE
Private Sub CMSCreateBBIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει δημιουργία BBIE – προετοιμάζει

		διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSEditBBIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει επεξεργασία BBIE – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSDeleteBBIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει διαγραφή BBIE– συνδέεται στη βάση και διαγράφει το επιλεγμένο BBIE
Private Sub CMSCreateASBIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει δημιουργία ASBIE – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSEditASBIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει επεξεργασία ASBIE – προετοιμάζει διεπαφή και καλεί – αφού συμπληρωθούν τα πεδία – συνάρτηση για σύνδεση στη βάση
Private Sub CMSDeleteASBIEItem_Click	Check.vb	Η μέθοδος που ενεργοποιείται όταν ο χρήστης επιλέξει διαγραφή ASBIE – συνδέεται στη βάση και διαγράφει το επιλεγμένο ASBIE
Private Sub ACC_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για τη διαχείριση των ACC (εμφανίζει και ενεργοποιεί συγκεκριμένα πεδία) – καλείται στην αρχή των μεθόδων που διαχειρίζονται ACC
Private Sub BCC_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για τη διαχείριση των BCC (εμφανίζει και ενεργοποιεί συγκεκριμένα πεδία) – καλείται στην αρχή των μεθόδων που διαχειρίζονται BCC
Private Sub ASCC_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για τη διαχείριση των ASCC (εμφανίζει και ενεργοποιεί συγκεκριμένα πεδία) – καλείται στην αρχή των μεθόδων που διαχειρίζονται ASCC
Private Sub ABIE_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για τη διαχείριση των ABIE (εμφανίζει και ενεργοποιεί συγκεκριμένα πεδία) – καλείται στην αρχή των μεθόδων που διαχειρίζονται ABIE
Private Sub BBIE_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για τη διαχείριση των BBIE (εμφανίζει και ενεργοποιεί συγκεκριμένα πεδία) – καλείται στην αρχή των μεθόδων που διαχειρίζονται BBIE
Private Sub ASBIE_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για τη διαχείριση των ASBIE (εμφανίζει και ενεργοποιεί συγκεκριμένα πεδία) – καλείται στην αρχή των μεθόδων που διαχειρίζονται ASBIE
Private Sub View_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για τη διαδικασία view (απλή επισκόπηση των οντοτήτων δηλαδή). Ουσιαστικά κρατάει όλα τα πεδία απενεργοποιημένα εκτός του Definition
Private Sub Form_Initialise()	Check.vb	Η μέθοδος που σχεδιάζει τη διεπαφή για την πρώτη εμφάνιση της διεπαφής – ουσιαστικά καθαρίζει την οθόνη και καλείται μετά από κάθε κλήση για εισαγωγή, επεξεργασία ή διαγραφή
Private Sub ControlButton_Click	Check.vb	Η μέθοδος που καθορίζει ποια ενέργεια θα

		γίνει – και ποια μέθοδος θα κληθεί ανάλογα με την κάθε περίπτωση όταν πατηθεί το κουμπί ελέγχου – μπορεί να είναι αποθήκευση, διαγραφή, δημιουργία XML Σχήματος κλπ...
Private Sub RTBox_TextChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο Representation Term με βάση τους κανόνες CCTS
Private Sub ASOCTBox_SelectedIndexChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο ASOCT με βάση τους κανόνες CCTS
Private Sub PTBox_TextChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο Property Term με βάση τους κανόνες CCTS
Private Sub OCTCheckBox_Load()	Check.vb	«Φορτώνει» το checkbox με όλα τα ACC της βάσης και χρησιμοποιείται ως το πεδίο που καθορίζει ποιο ACC θα έχει ως template του το προς δημιουργία ABIE
Private Sub QDTBox_Load()	Check.vb	«Φορτώνει» το checkbox με όλα τα QDT της βάσης
Private Sub ASOCTBox_Load()	Check.vb	«Φορτώνει» το checkbox με όλα τα ACC της βάσης και χρησιμοποιείται ως το πεδίο που καθορίζει με ποιο ACC θα σχετίζεται το προς δημιουργία ASBIE
Private Sub OCTBox_TextChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο OCTBOX με βάση τους κανόνες CCTS (για CCs)
Private Sub NameBox_TextChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο Name
Private Sub OCTCBox_SelectedIndexChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο OCTCBox με βάση τους κανόνες CCTS (για BIEs)
Private Sub Create_ACC()	Check.vb	Η καθαυτό συνάρτηση δημιουργίας και καταχώρησης ACC στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Edit_ACC()	Check.vb	Η καθαυτό συνάρτηση επεξεργασίας και ενημέρωσης των ACC στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Create_BCC()	Check.vb	Η καθαυτό συνάρτηση δημιουργίας και καταχώρησης BCC στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Edit_BCC()	Check.vb	Η καθαυτό συνάρτηση επεξεργασίας και ενημέρωσης των BCC στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Create_ASCC()	Check.vb	Η καθαυτό συνάρτηση δημιουργίας και καταχώρησης ASCC στη βάση δεδομένων – Δημιουργεί connection στη βάση και

		αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Edit_ASCC()	Check.vb	Η καθαυτό συνάρτηση επεξεργασίας και ενημέρωσης των ASCC στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Create_ABIE()	Check.vb	Η καθαυτό συνάρτηση δημιουργίας και καταχώρησης ABIE στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Edit_ABIE()	Check.vb	Η καθαυτό συνάρτηση επεξεργασίας και ενημέρωσης των ABIE στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Create_BBIE()	Check.vb	Η καθαυτό συνάρτηση δημιουργίας και καταχώρησης BBIE στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Edit_BBIE()	Check.vb	Η καθαυτό συνάρτηση επεξεργασίας και ενημέρωσης των BBIE στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Create_ASbie()	Check.vb	Η καθαυτό συνάρτηση δημιουργίας και καταχώρησης ASbie στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub Edit_ASbie()	Check.vb	Η καθαυτό συνάρτηση επεξεργασίας και ενημέρωσης των ASbie στη βάση δεδομένων – Δημιουργεί connection στη βάση και αποθηκεύει – καλείται από άλλες μεθόδους που έχουν προετοιμάσει τη διεπαφή και τα υπόλοιπα πεδία προς αποθήκευση
Private Sub PTQBox_TextChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο Property Term Qualifier με βάση τους κανόνες CCTS
Private Sub ASOCTQBox_TextChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο ASOCT Qualifier με βάση τους κανόνες CCTS
Private Sub OCTQBox_TextChanged	Check.vb	Η μέθοδος που καθορίζει ποια πεδία και πώς θα αλλάξουν όταν αλλάξει το πεδίο OCT Qualifier με βάση τους κανόνες CCTS
Private Sub TempButton_Click	Check.vb	Η μέθοδος που καλείται και εμφανίζει τη φόρμα SelectTemplate.vb κατά τη δημιουργία BBIE και ASbie για την επιλογή των επιμέρους στοιχείων τους από επιτρεπόμενη λίστα στοιχείων με βάση το ACC που το ABIE τους έχει ως template όταν πατηθεί το αντίστοιχο κουμπί

Public Function SendACCKey() As Integer	Check.vb	Στέλνει στη φόρμα SelectTemplate.vb το πρωτεύον κλειδί του ACC που έχουν ως template τα BBIE και ASBIE για να ανακτήσει από τη βάση και εμφανίσει τα κατάλληλα στοιχεία
Private Sub TemplateChanged	Check.vb	Η μέθοδος που καλείται όταν επιλέξει ο χρήστης στοιχεία για τα BBIE και ASBIE από την επιτρεπόμενη λίστα στοιχείων που μπορεί να έχουν με βάση το ACC template του ABIE τους
Private Sub LoadRTBox()	Check.vb	«Φορτώνει» στο combo box τις τιμές που μπορεί να πάρει το πεδίο Representation Term
Public Sub convert(ByVal Destination, ByVal ABIEkey, ByVal ABIEname)	Check.vb	Η μέθοδος που αναλαμβάνει εξολοκλήρου τη δημιουργία του XML Σχήματος, επιβάλλει τη μορφή και τη σύνταξή του
Private Sub RTBox_SelectedIndexChanged	Check.vb	Καθορίζει το πώς αλλάζουν τα διάφορα πεδία που εξαρτώνται από το πεδίο Representation Term όταν αυτό μεταβληθεί
Private Sub RTBOXChangesBBIEFields()	Check.vb	Καθορίζει το πώς αλλάζουν τα διάφορα πεδία που εξαρτώνται από το πεδίο Representation Term όταν αυτό μεταβληθεί κατά τη δημιουργία BBIE
Private Sub SelectTemplate_Load	SelectTemplate.vb	Αναλαμβάνει να φορτώσει από τη βάση δεδομένων τα κατάλληλα στοιχεία για τα υπό δημιουργία BBIE και ASBIE ανάλογα με το ACC που το ABIE τους έχει ως template
Private Sub DataGridView1_CellContentClick	SelectTemplate.vb	Ανακτά από τη βάση και αναθέτει σε εξωτερικές μεταβλητές τα επιλεγμένα από το χρήστη στοιχεία
Private Sub SelectButton_Click	SelectTemplate.vb	Ανάλογα με τη λειτουργία, στέλνει στο κυρίως πρόγραμμα τα επιλεγμένα στοιχεία και το ενημερώνει για την αποστολή αυτή
Public Function SendPT() As String	SelectTemplate.vb	Αποστολή στην κυρίως φόρμα του περιεχομένου του πεδίου Property Term που επιλέχθηκε
Public Function SendRT()	SelectTemplate.vb	Αποστολή στην κυρίως φόρμα του περιεχομένου του πεδίου Representation Term που επιλέχθηκε
Public Function SendASOCT() As String	SelectTemplate.vb	Αποστολή στην κυρίως φόρμα του περιεχομένου του πεδίου ASOCT που επιλέχθηκε
Public Sub Signal()	SelectTemplate.vb	Σήμα προς την κύρια φόρμα ότι τα στοιχεία εστάλησαν
Private Sub ReturnButton_Click	SelectTemplate.vb	Εάν πατηθεί γίνεται επιστροφή στην κυρίως φόρμα χωρίς να έχει επιλεγεί κάτι
Private Sub Button1_Click	MainForm.vb	Από το παράθυρο καλωσορίσματος οδηγεί στην κυρίως φόρμα
Private Sub ExitButton_Click	MainForm.vb	Από το παράθυρο καλωσορίσματος τερματίζει την εφαρμογή
Private Sub MainForm_Load	MainForm.vb	Αρχικοποιεί τη διεπαφή καλωσορίσματος

4.3 Έλεγχος εφαρμογής – Έγγραφο Περιγραφή Δυναμικού

Ελέγχου

4.3.1 Οντότητα προς έλεγχο

Στο παρόν έγγραφο, το λογισμικό σύστημα που αναπτύχθηκε αποτελεί την οντότητα προς έλεγχο ποιότητας.

4.3.2 Είδος ελέγχου

Ο έλεγχος θα γίνει απευθείας σε ολόκληρο το σύστημα που υλοποιήθηκε.

4.3.3 Περιπτώσεις δοκιμής

Θα δημιουργηθεί ένα ελάχιστο σύνολο περιπτώσεων δοκιμής για το λειτουργικό δυναμικό έλεγχο, το οποίο εξασφαλίζει ότι ελέγχονται όλες οι εργασίες του υλοποιηθέντος λογισμικού συστήματος τουλάχιστον μια φορά η κάθε μια.

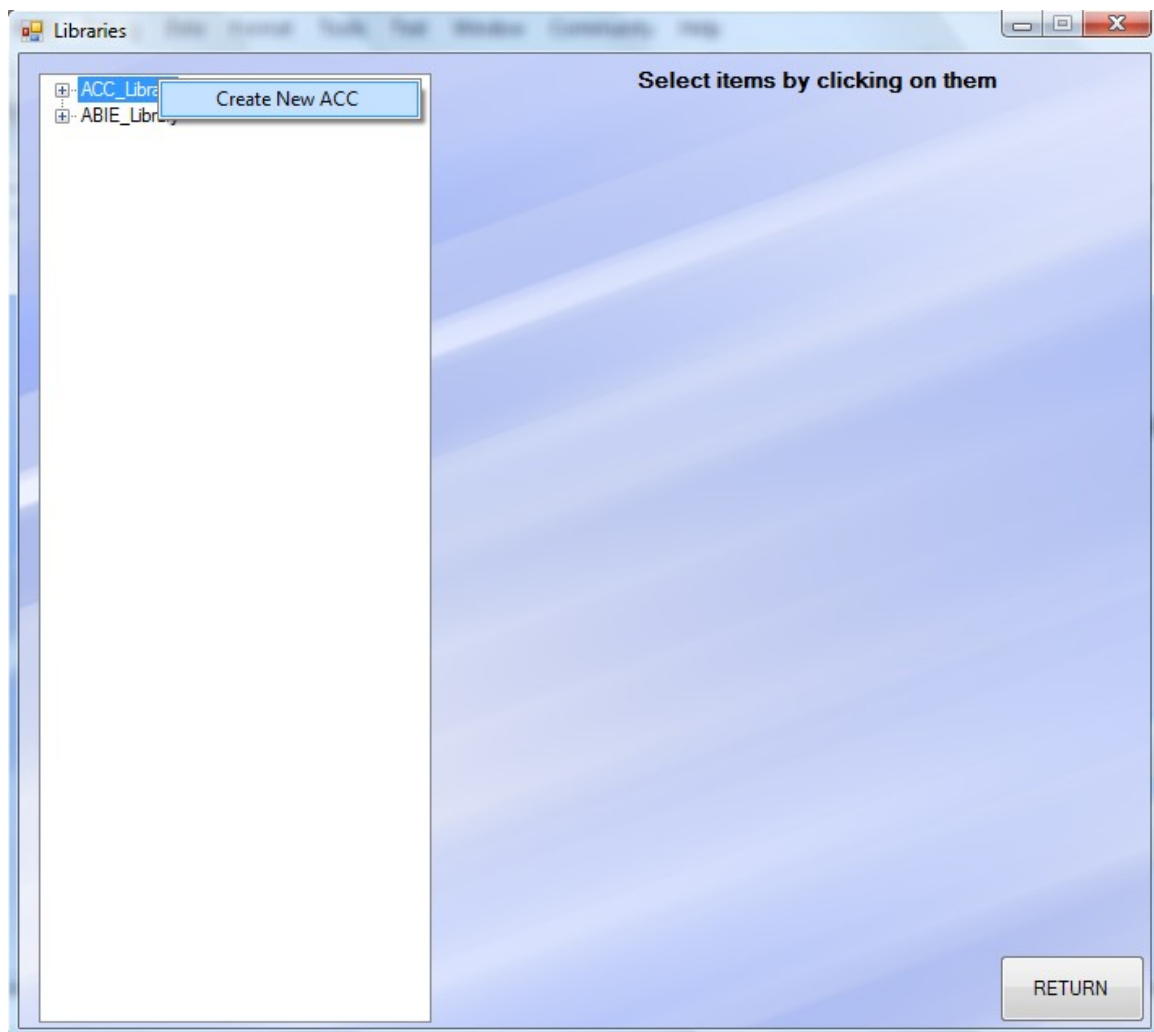
Σημειώνεται ότι οι διαφορετικές περιπτώσεις δοκιμής είναι πολλές ανάλογα με τις επιλογές του χρήστη και τις διαφορετικές ανάγκες κάθε φορά, αλλά στόχος της παρούσας ενότητας είναι να παρουσιάσει ενδεικτικά ένα πλήρες και αντιπροσωπευτικό παράδειγμα χρήσης του προγράμματος από τη δημιουργία των αρχικών Core Components μέχρι την παραγωγή του τελικού XML Σχήματος.

Με βάση τους υπάρχοντες κανόνες του προτύπου που ακολουθείται και υλοποιείται στην παρούσα διπλωματική εργασία, θα παρουσιαστεί η μοντελοποίηση των δεδομένων ενός δελτίου αστυνομικής ταυτότητας και η παραγωγή του τελικού XML Σχήματος που παρουσιάζει τη μορφή που πρέπει να έχει ένα αρχείο XML που περιέχει τα στοιχεία ενός πολίτη ώστε να αποτελεί έγκυρο αρχείο δελτίου αστυνομικής ταυτότητας

Το ABIE σε αυτή την περίπτωση είναι το δελτίο αστυνομικής ταυτότητας το οποίο αποτελείται από δυο BBIE και δυο ASBIE. Τα BBIE είναι ο Τίτλος του Εγγράφου και η Ημερομηνία Έκδοσης, ενώ τα ASBIE είναι η Εκδούσα Αρχή και ο Πολίτης. Οι οντότητες αυτές έχουν όλες τους ως template το ACC Document, δηλαδή το Aggregate Core Component με το όνομα Έγγραφο. Το ACC όπως έχει αναπτυχθεί και πιο πάνω αποτελείται από κάποια BCCs και ACCs όπως, για παράδειγμα, είναι η Ημερομηνία Έκδοσης και το Σχετιζόμενο γεγονός, αντίστοιχα.

Για λόγους οικονομίας χώρου, θα παρουσιαστεί η δημιουργία του ACC Document και από η δημιουργία ενός από τα αντίστοιχα BCC και ACC της οντότητας Έγγραφο. Όλες όμως οι οντότητες θα δημιουργηθούν και θα καταχωρηθούν με τον ίδιο τρόπο και θα ακολουθήσουν και τα αντίστοιχα στιγμιότυπα εκτέλεσης του προγράμματος που θα τεκμηριώσουν αυτό το γεγονός.

Η εκτέλεση, λοιπόν, αρχίζει με τη δημιουργία του ACC με το όνομα Document (Έγγραφο). Κατά τα γνωστά, όπως αυτά παρουσιάστηκαν στις περιπτώσεις χρήσης κατά την περιγραφή του συστήματος σε προηγούμενο κεφάλαιο, δημιουργούμε ένα ACC:



Εικόνα 4.3-44: Τεκμηρίωση (1)

Οπότε, εμφανίζεται η παρακάτω οθόνη όπου συμπληρώνουμε τα επιμέρους στοιχεία για αυτό το Core Component, ακολουθώντας πάντα τους κανόνες όπως αυτοί έχουν τεθεί από το CCTS και έχουν εφαρμοσθεί στο Ελληνικό Πλαίσιο Ηλεκτρονικής Διακυβέρνησης:

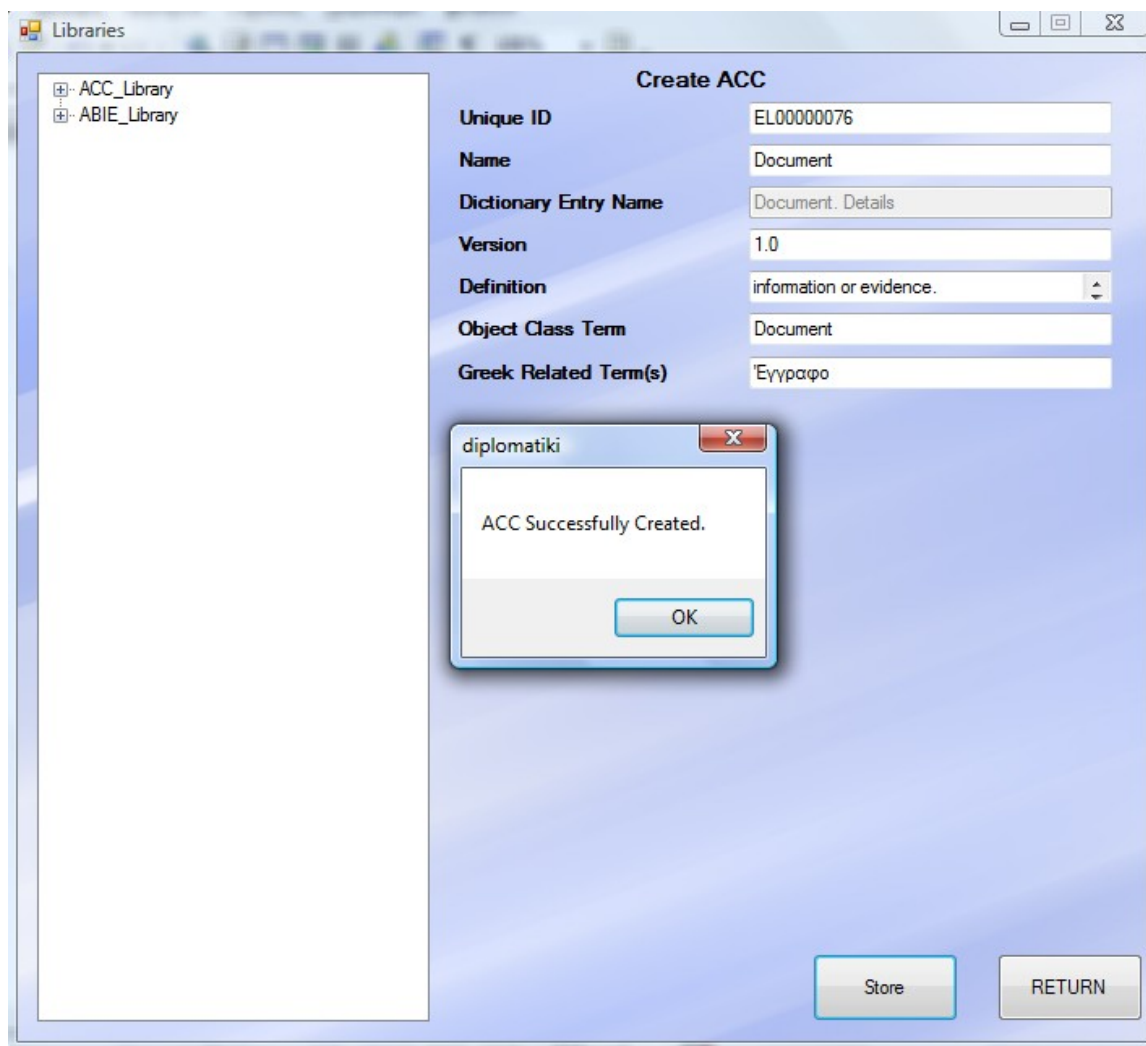
The screenshot shows a software window titled "Libraries" with a "Create ACC" form. On the left, a tree view shows "ACC_Library" and "ABIE_Library". The form fields are as follows:

Field	Value
Unique ID	EL00000076
Name	Document
Dictionary Entry Name	Document. Details
Version	1.0
Definition	information or evidence.
Object Class Term	Document
Greek Related Term(s)	Έγγραφο

At the bottom right, there are two buttons: "Store" and "RETURN".

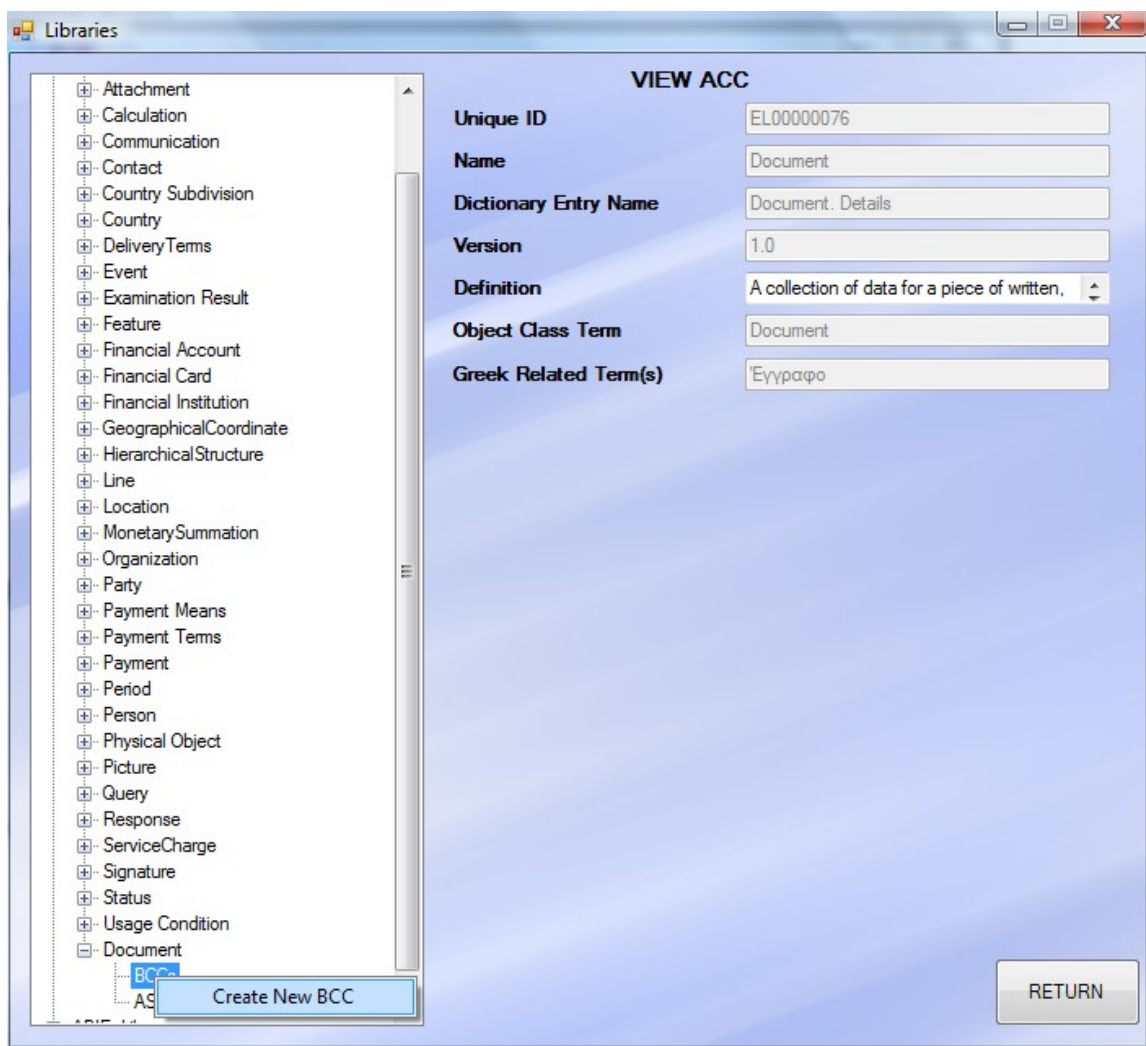
Εικόνα 4.3-45: Τεκμηρίωση (2)

Και αποθηκεύουμε το νέο αυτό ACC κάνοντας κλικ στο Store:



Εικόνα 4.3-46: Τακμηρίωση (3)

Οπότε τώρα στις βιβλιοθήκες υπάρχει και το καινούριο αυτό ACC χωρίς βέβαια BCC και ACC ακόμη. Γι' αυτό το λόγο προχωρούμε στην προσθήκη του πρώτου BCC με όνομα Αναγνωριστικό Εγγράφου:

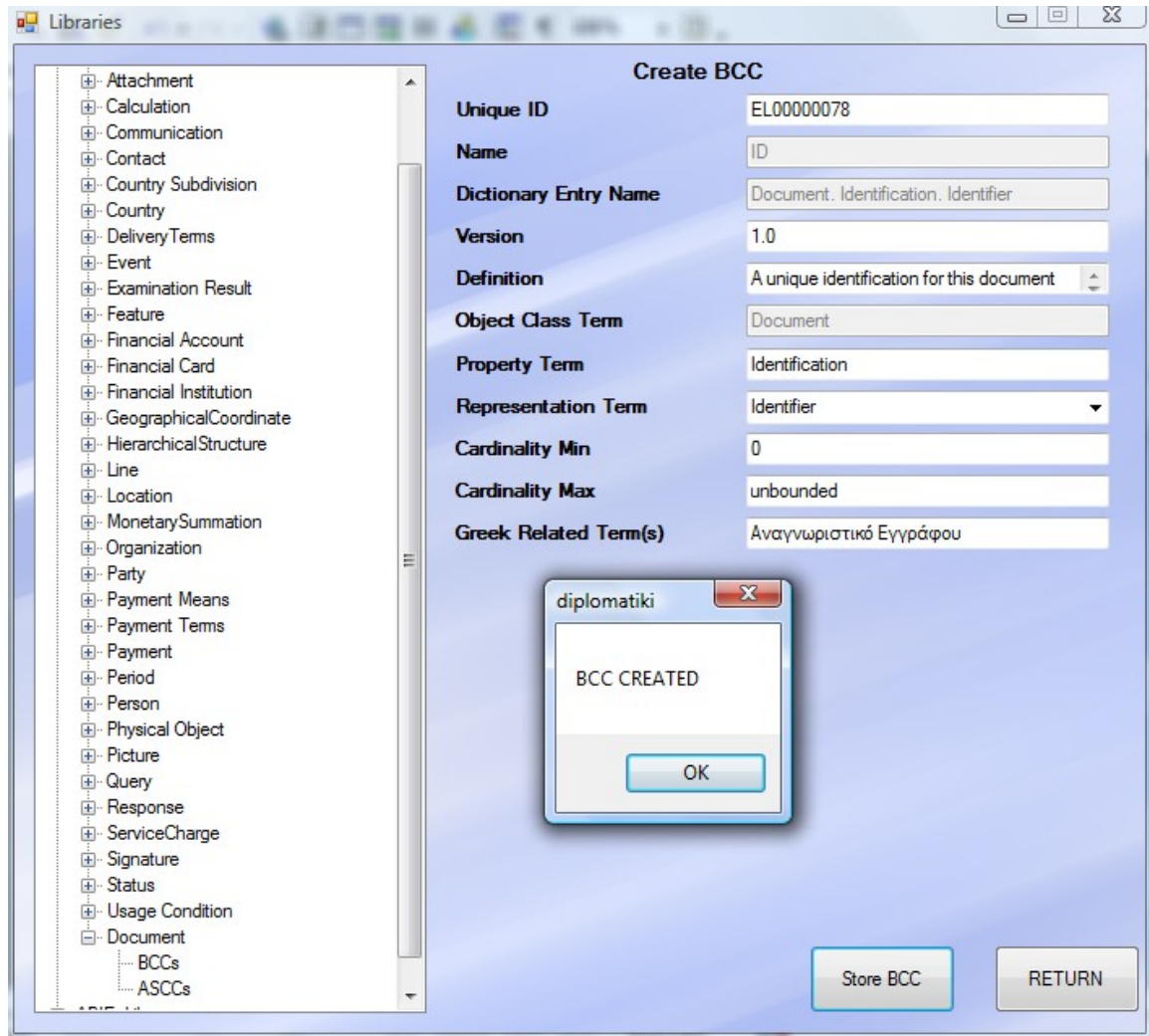


Εικόνα 4.3-47: Τεκμηρίωση (4)

Έπειτα, θα εμφανιστεί η φόρμα συμπλήρωσης των στοιχείων του BCC η οποία φαίνεται στην επόμενη σελίδα. Αξίζει σε αυτό το σημείο να παρατηρηθεί ότι τα αυτόματα πεδία τα οποία δεν είναι ενεργοποιημένα (όπως είναι για παράδειγμα το Name ή το Dictionary Entry Name) και τα οποία σχηματίζονται με βάση κάποια άλλα πεδία και πώς αυτά μεταβάλλονται (όπως τα πεδία Property Term και Representation Term) έχουν συμπληρωθεί αυτόματα ανάλογα με τις επιλογές του χρήστη και φυσικά πάντα σε συμφωνία με το πρότυπο που υλοποιείται. Το ίδιο συμβαίνει και με το πεδίο Object Class Term που έχει συμπληρωθεί αυτόματα λόγω του ότι το BCC έχει σαν Template του το ACC Document που έχει Object Class Term τον όρο Document. Ακόμη, πρέπει να σημειωθεί, ακόμα και αν δε φαίνεται επακριβώς στις εικόνες, ότι για το Representation Term υπάρχει ειδικό Combo Box για την επιλογή του αντίστοιχου όρου από τους 8 θεμελιώδεις όρους του προτύπου CCTS. Τέλος, οι cardinalities δείχνουν το πλήθος από τις συγκεκριμένες οντότητες (BCC) μπορεί να έχει η συσχετιζόμενη με αυτό οντότητα (ACC).

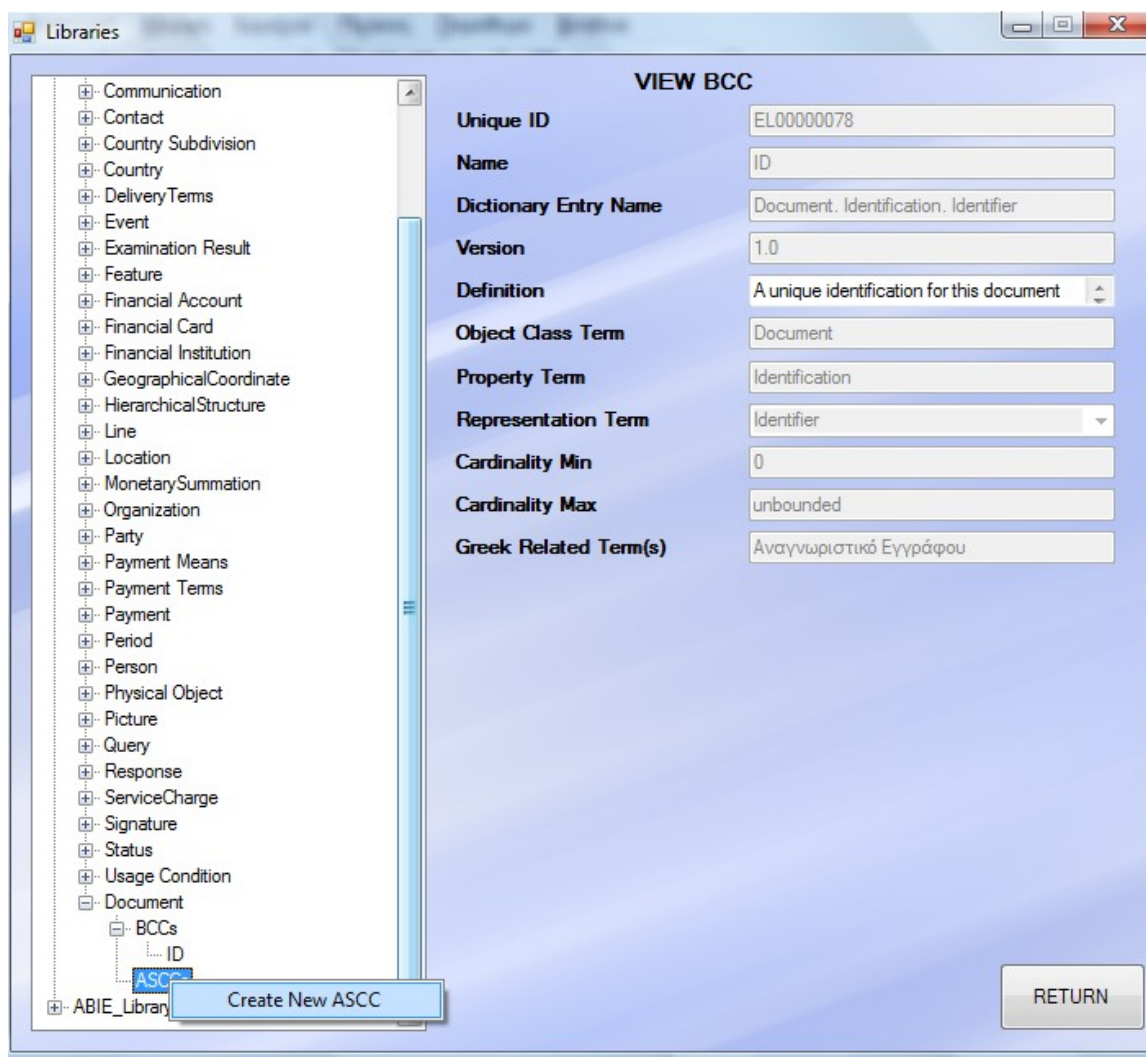
Στην προκειμένη περίπτωση, ένα Έγγραφο (ACC) μπορεί να έχει κανένα ή και περισσότερα (χωρίς όριο) Αναγνωριστικά Εγγράφου (BCC).

Με κλικ στο κουμπί Store BCC έχουμε την επιτυχή καταχώρηση του BCC στη λίστα όπως θα φανεί και στην επόμενη εικόνα όπου τεκμηριώνονται όλα τα παραπάνω:



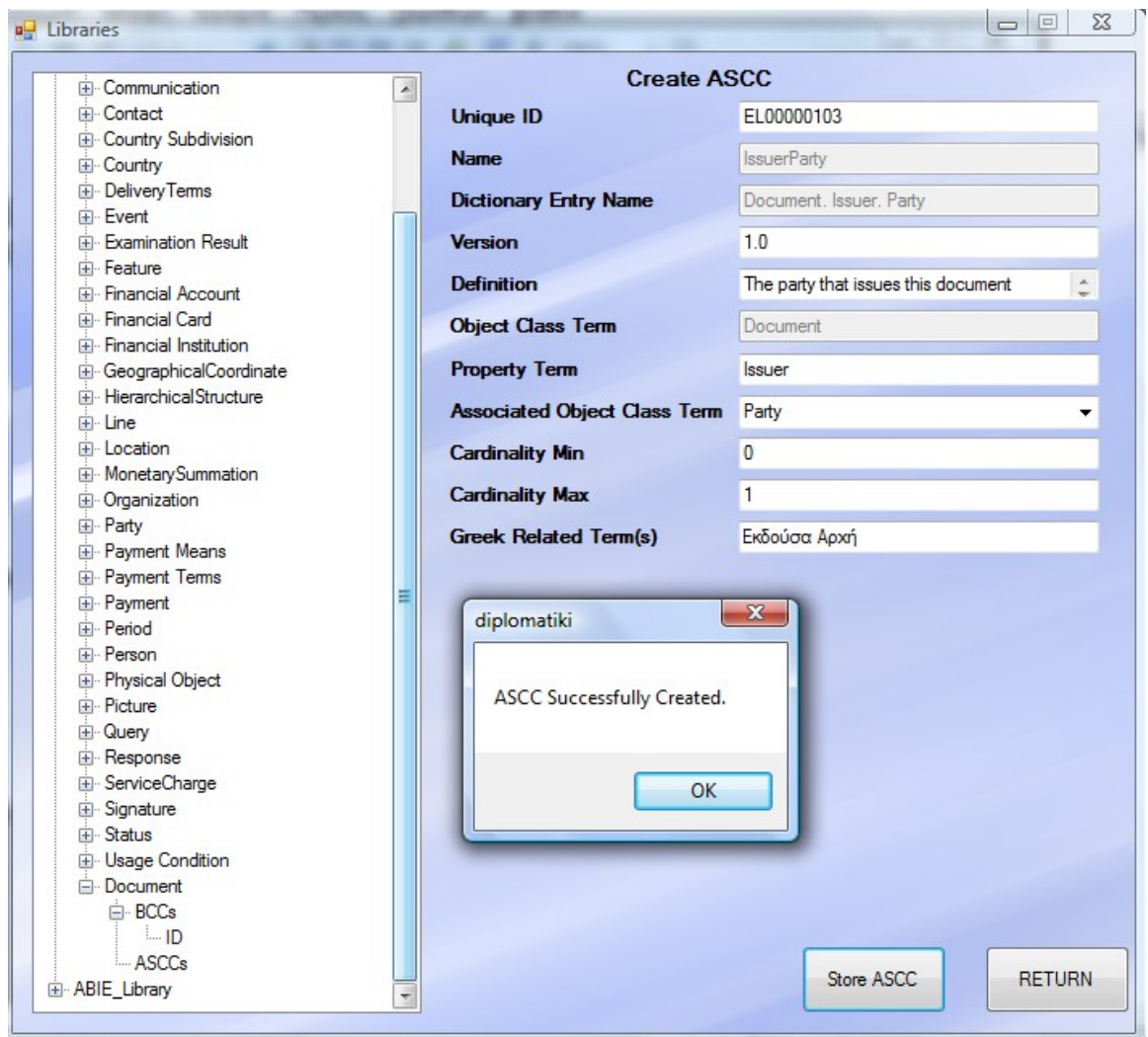
Εικόνα 4.3-48: Τεκμηρίωση (5)

Στη συνέχεια, θα προχωρήσουμε στη δημιουργία ενός ASCC, έστω εκείνου με το όνομα Εκδούσα Αρχή. Στο επόμενο παράθυρο φαίνονται πάλι οι ενέργειές μας όπου φαίνεται και η καταχώρηση που μόλις κάναμε για το BCC με όνομα ID:



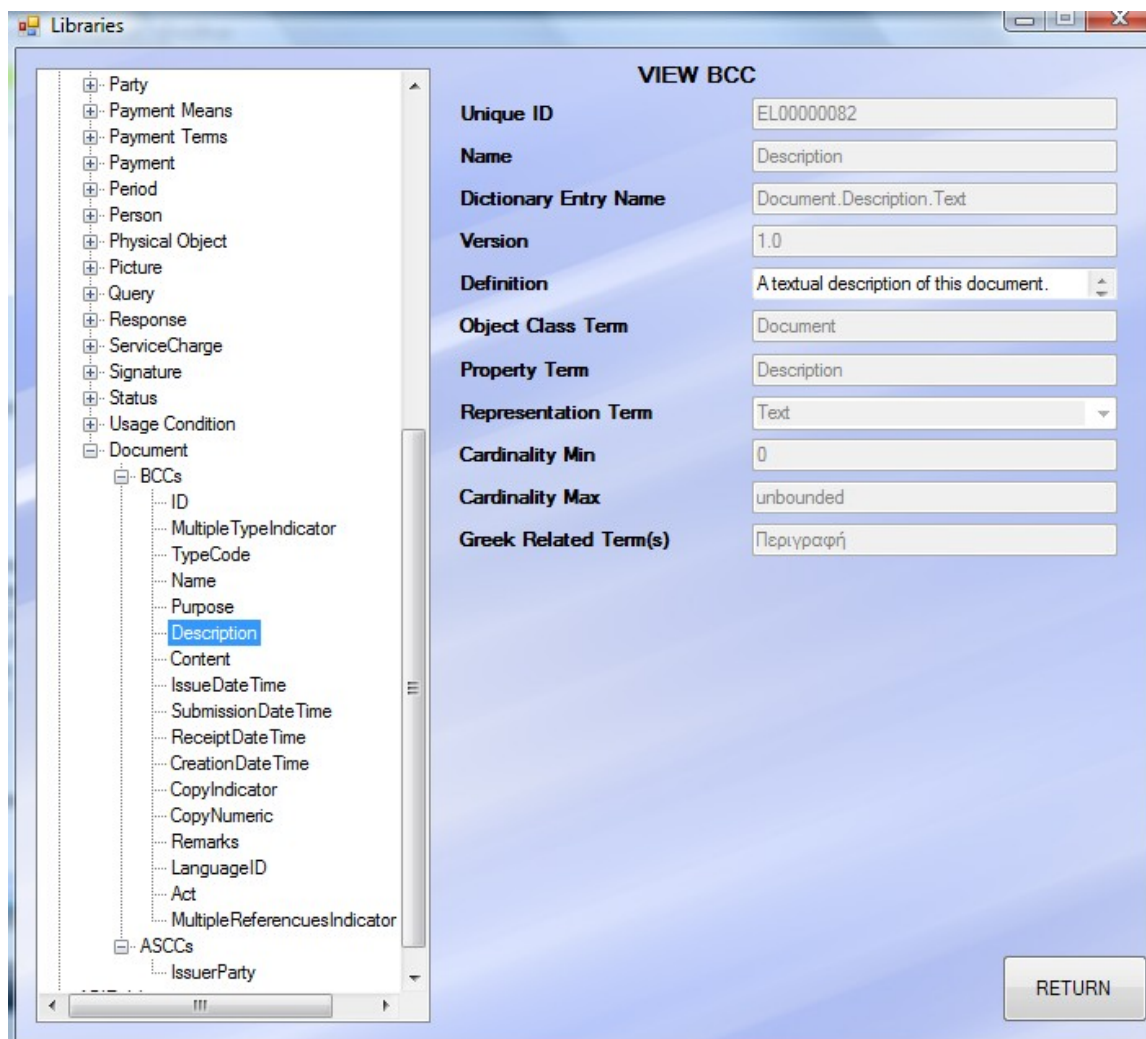
Εικόνα 4.3-49: Τεκμηρίωση (6)

Στην επόμενη σελίδα φαίνεται η συμπλήρωση των πεδίων για το ASCC και η επιτυχής καταχώρησή του στη βάση δεδομένων. Στη συνέχεια, τα υπόλοιπα BCC και ASCC της οντότητας έγγραφο θα εισαχθούν κανονικά με το παρόν πρόγραμμα και στο τέλος θα παρατεθεί μια πλήρης εικόνα για την οντότητα η οποία θα καταστήσει δυνατή για τη δημιουργία του επιθυμητού ABIE, των BBIE και ASBIE τα οποία θα έχουν ως Template το Document και από εκεί θα γίνει η δημιουργία του τελικού XML Σχήματος.



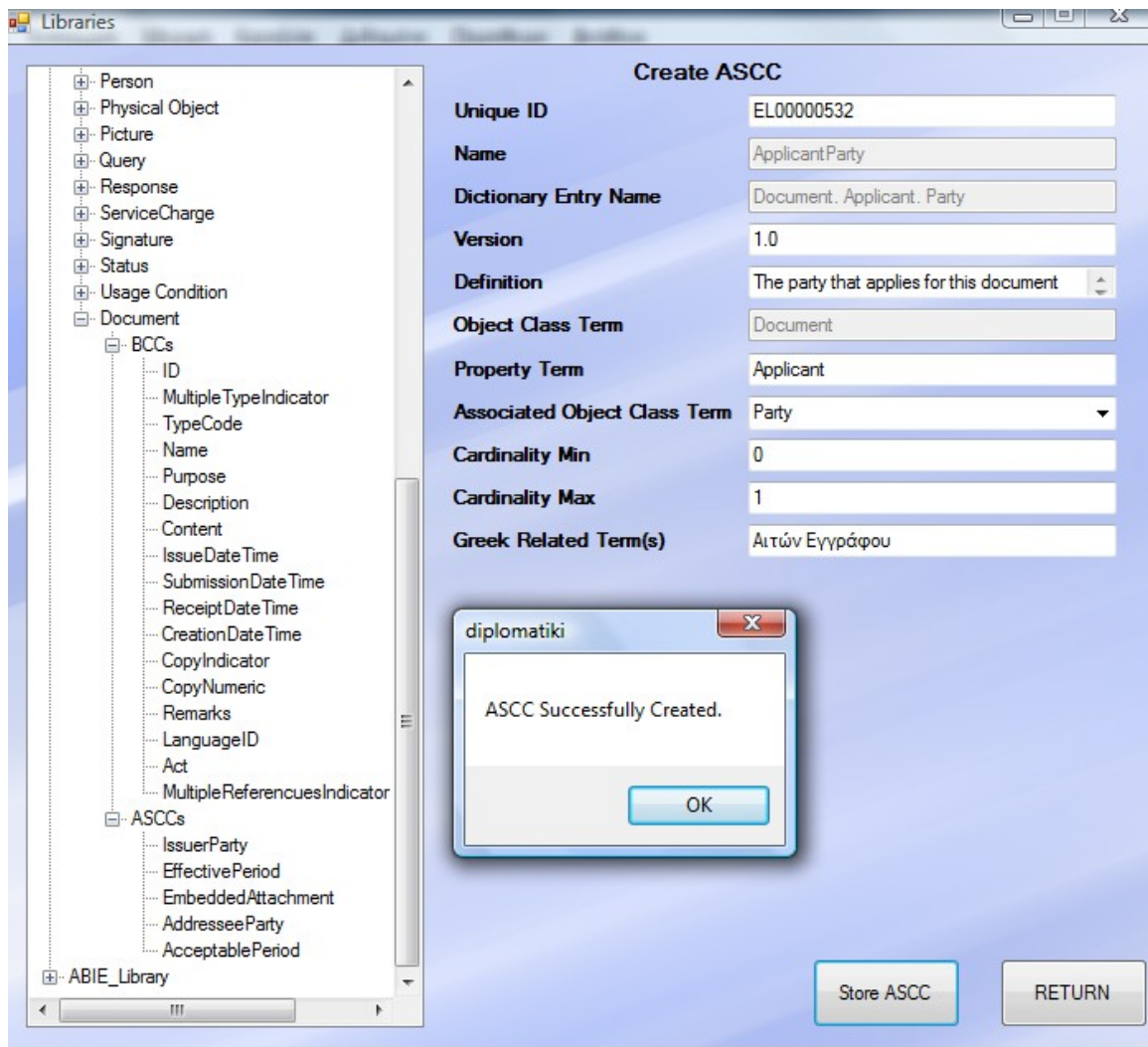
Εικόνα 4.3-50: Τεκμηρίωση (7)

Παρατίθενται παρακάτω μερικά ενδιάμεσα στιγμιότυπα (προσθήκη BCC):



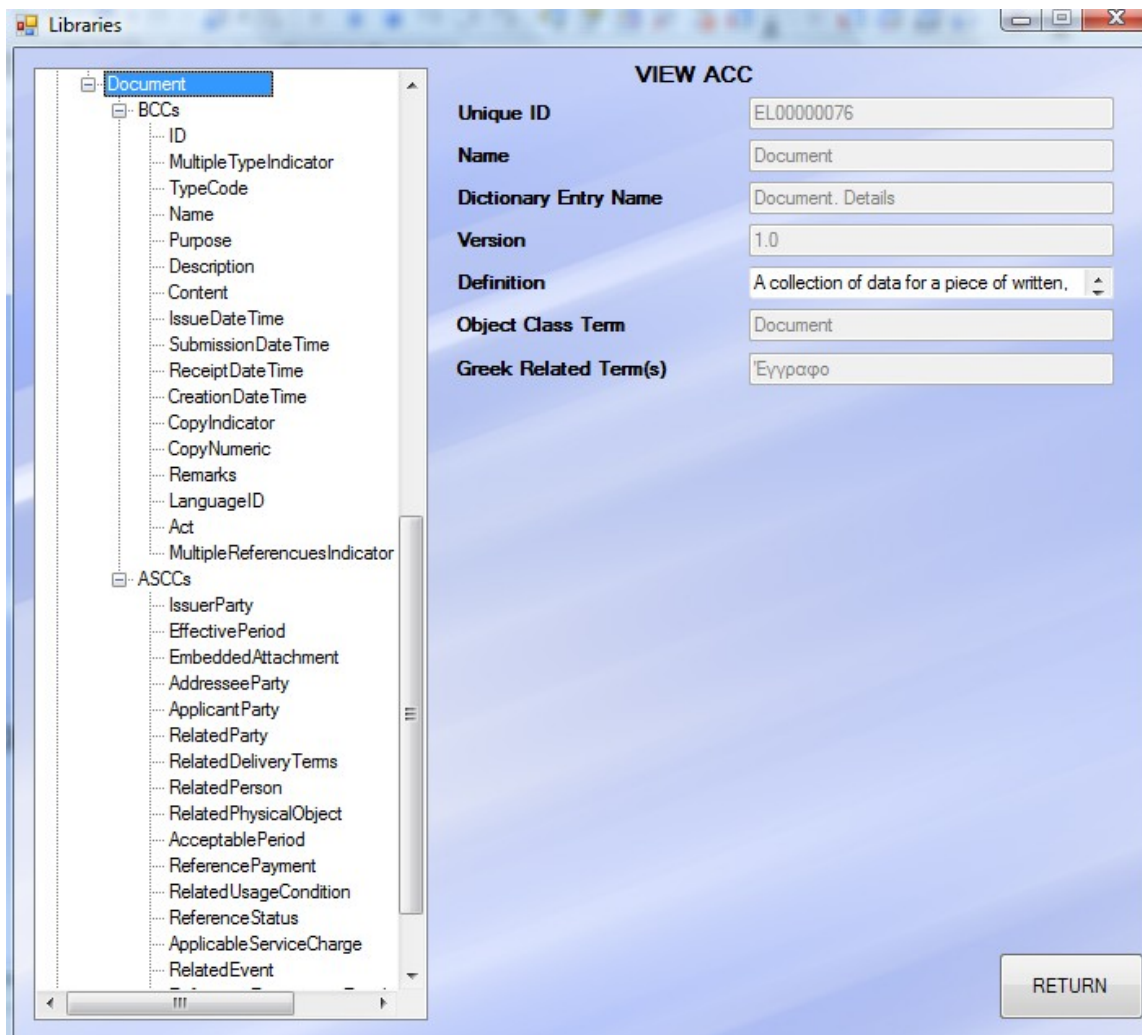
Εικόνα 4.3-51: Τεκμηρίωση (8)

Προσθήκη ASCC:



Εικόνα 4.3-52: Τεκμηρίωση (9)

Τέλος, παρατίθεται παρακάτω η πλήρης εικόνα για την οντότητα Έγγραφο:

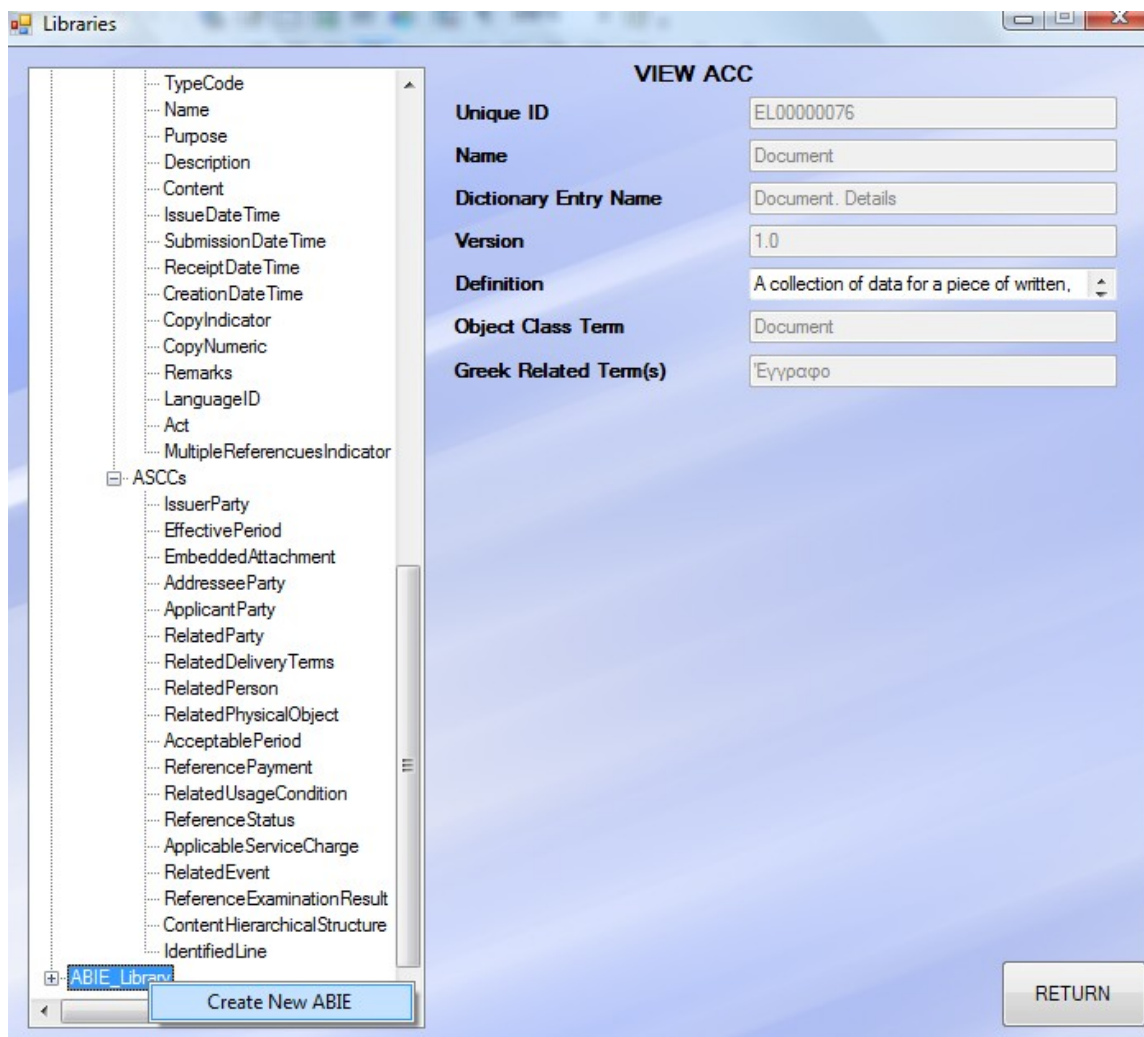


Εικόνα 4.3-53: Τεκμηρίωση (10)

Αφού ολοκληρώθηκε η διαδικασία δημιουργίας του βασικού ACC το οποίο θα αποτελέσει το Template για τη δημιουργία του ABIE Δελτίο Αστυνομικής Ταυτότητας ή IdentityCardDocument όπως είναι η μοντελοποιημένη ονομασία του με βάση το πρότυπο που χρησιμοποιείται.

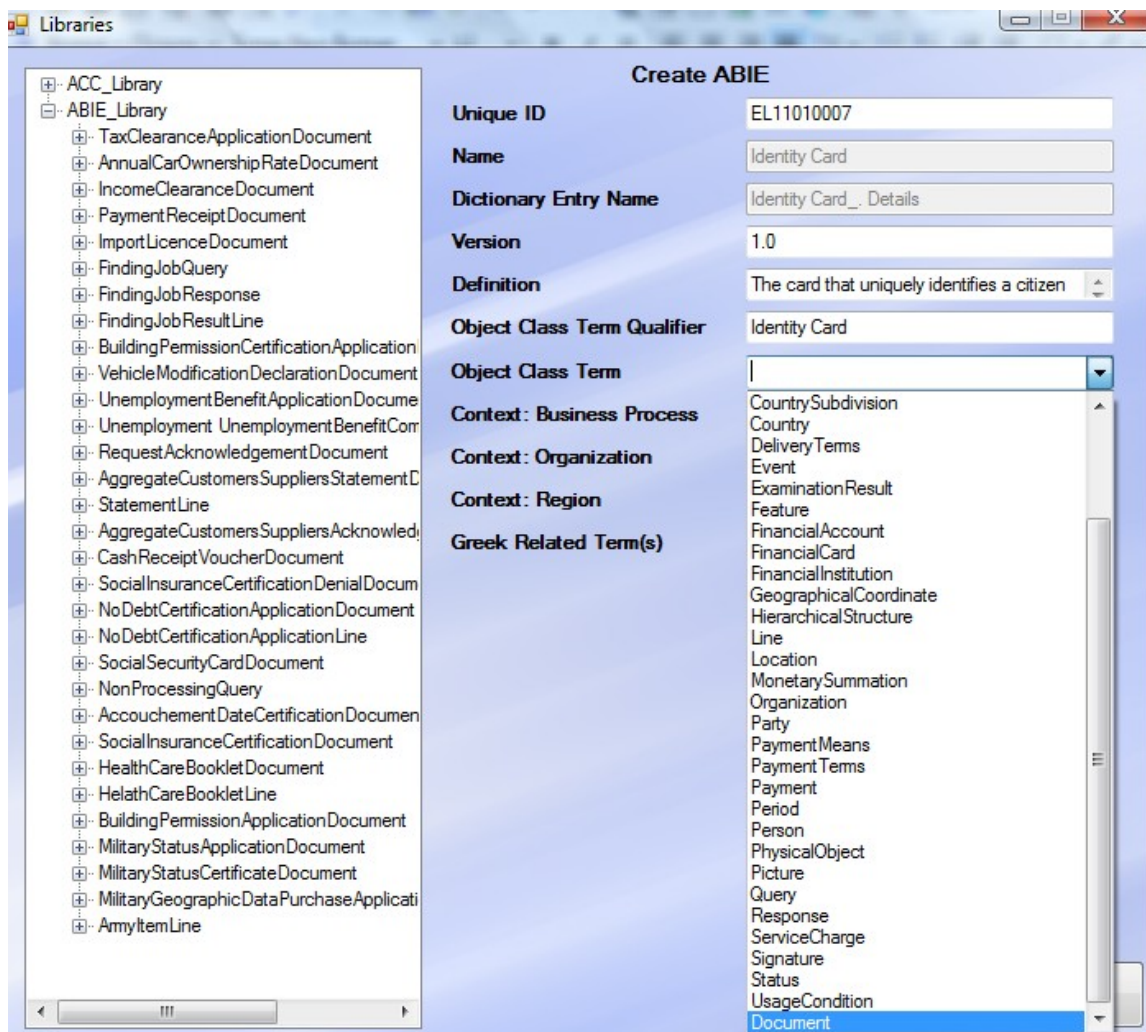
Το ABIE Δελτίο Αστυνομικής Ταυτότητας αποτελείται από δυο BBIE, τον Τίτλο Εγγράφου (Name) και την Ημερομηνία Έκδοσης (IssueDateTime) και από δυο ASBIE, την Εκδούσα Αρχή (IssuerPublicParty) και τον Πολίτη (RelatedCitizenPerson). Επιλέχθηκε να κατασκευασθεί αυτό το ABIE αφού έχει λίγα στοιχεία, γι' αυτό και θα γίνει αναλυτική παρουσίαση της κατασκευής του, και επειδή είναι ένα από τα πιο ενδεικτικά παραδείγματα εφαρμογής της παρούσας πλατφόρμας για ένα έγγραφο που χρησιμοποιείται πολύ συχνά στην Ηλεκτρονική Διακυβέρνηση.

Η αρχή θα γίνει με τη δημιουργία του ABIE:



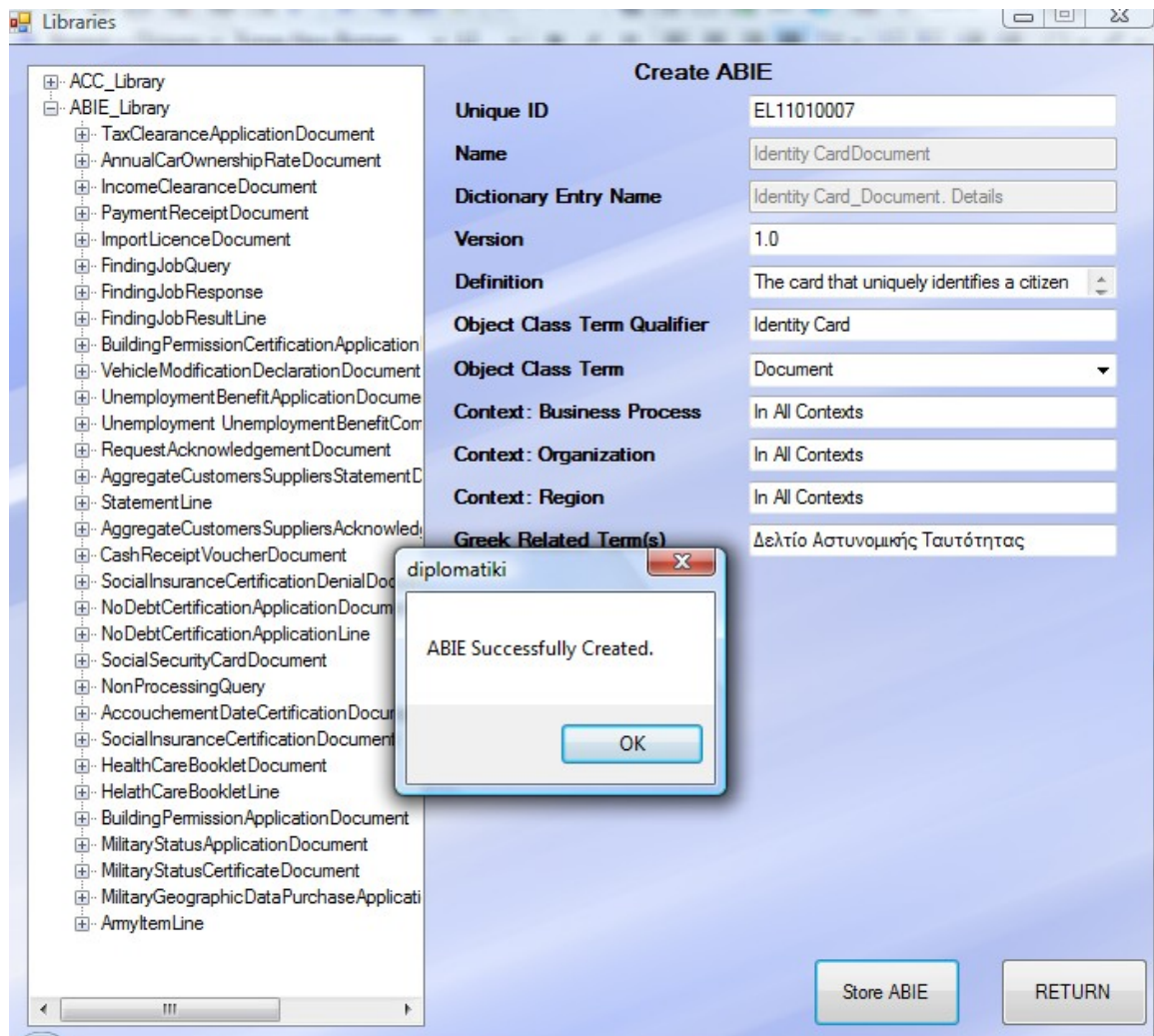
Εικόνα 4.3-54: Τεκμηρίωση (11)

Και συμπληρώνουμε με τα επιθυμητά στοιχεία. Στην επόμενη σελίδα φαίνεται ένα στιγμιότυπο της δημιουργίας όπου επιλέγεται από τη λίστα των αποθηκευμένων στη βάση ACC εκείνο το μοναδικό που θα αποτελεί Template για το ABIE που δημιουργείται. Σε αυτό το σημείο και με την ακόλουθη καταχώρηση στη βάση δεδομένων, το νεοεισαχθέν ABIE διατηρεί εσωτερικό σύνδεσμο με τη βασική οντότητα Έγγραφο και όλα της πεδία και σε επόμενα βήματα θα φανεί πάλι πώς η οντότητα – Template καθορίζει τα επιτρεπτά πεδία για τα BBIE και ASBIE του ABIE «Δελτίο Αστυνομικής Ταυτότητας».



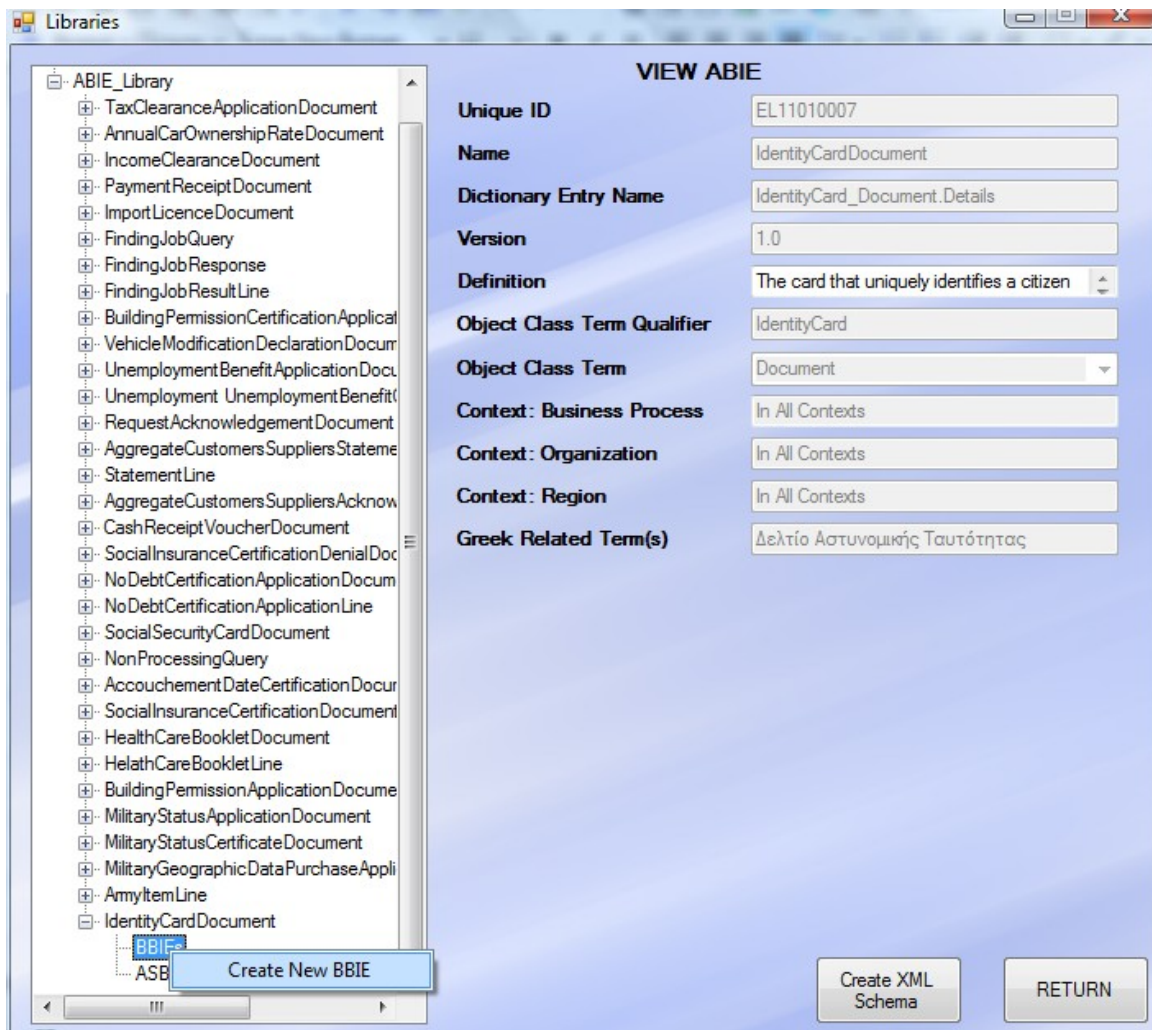
Εικόνα 4.3-55: Τεκμηρίωση (12)

Και η τελική εικόνα:

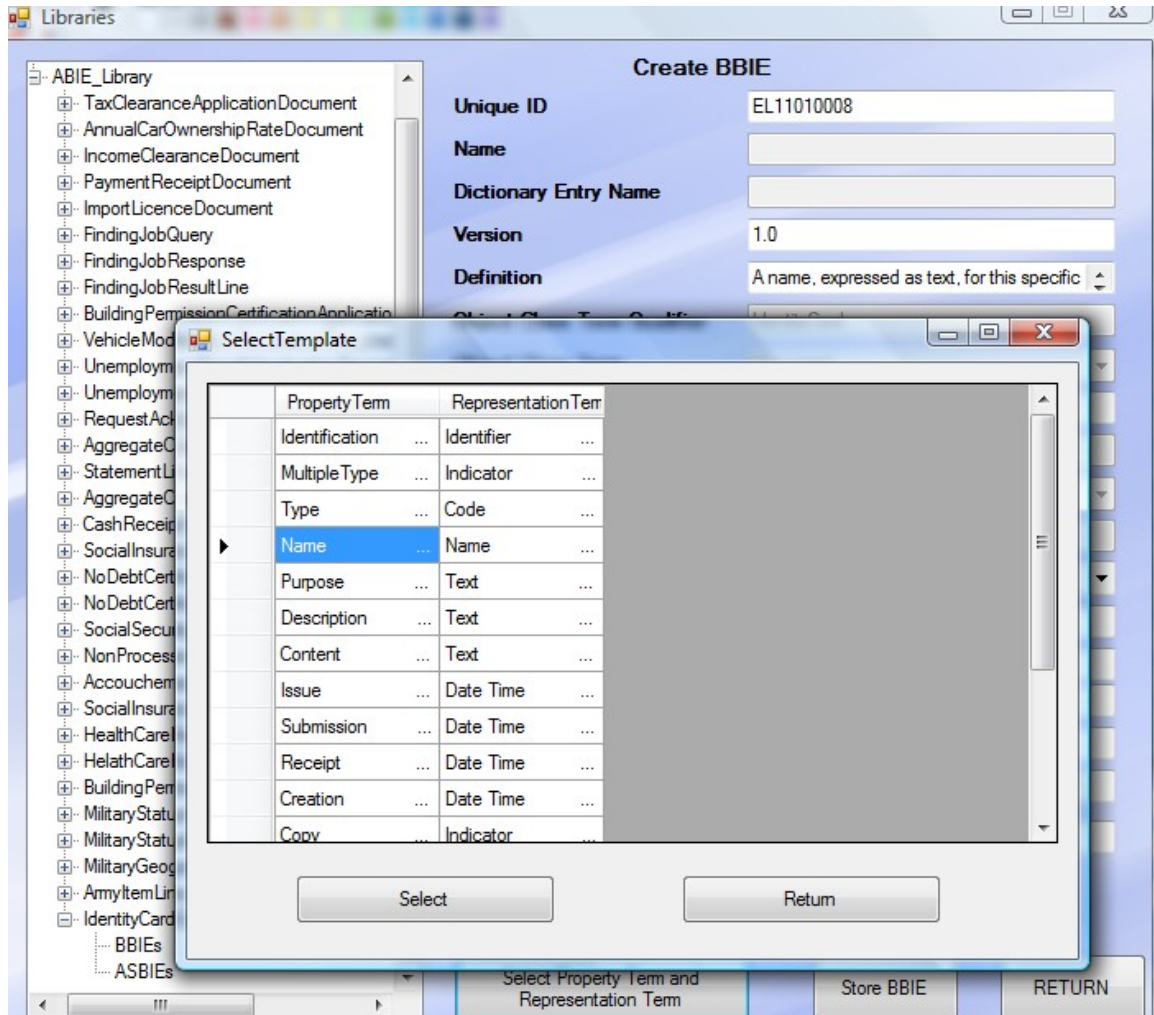


Εικόνα 4.3-56: Τεκμηρίωση (13)

Συνεχίζουμε με τη δημιουργία των δυο BBIE:

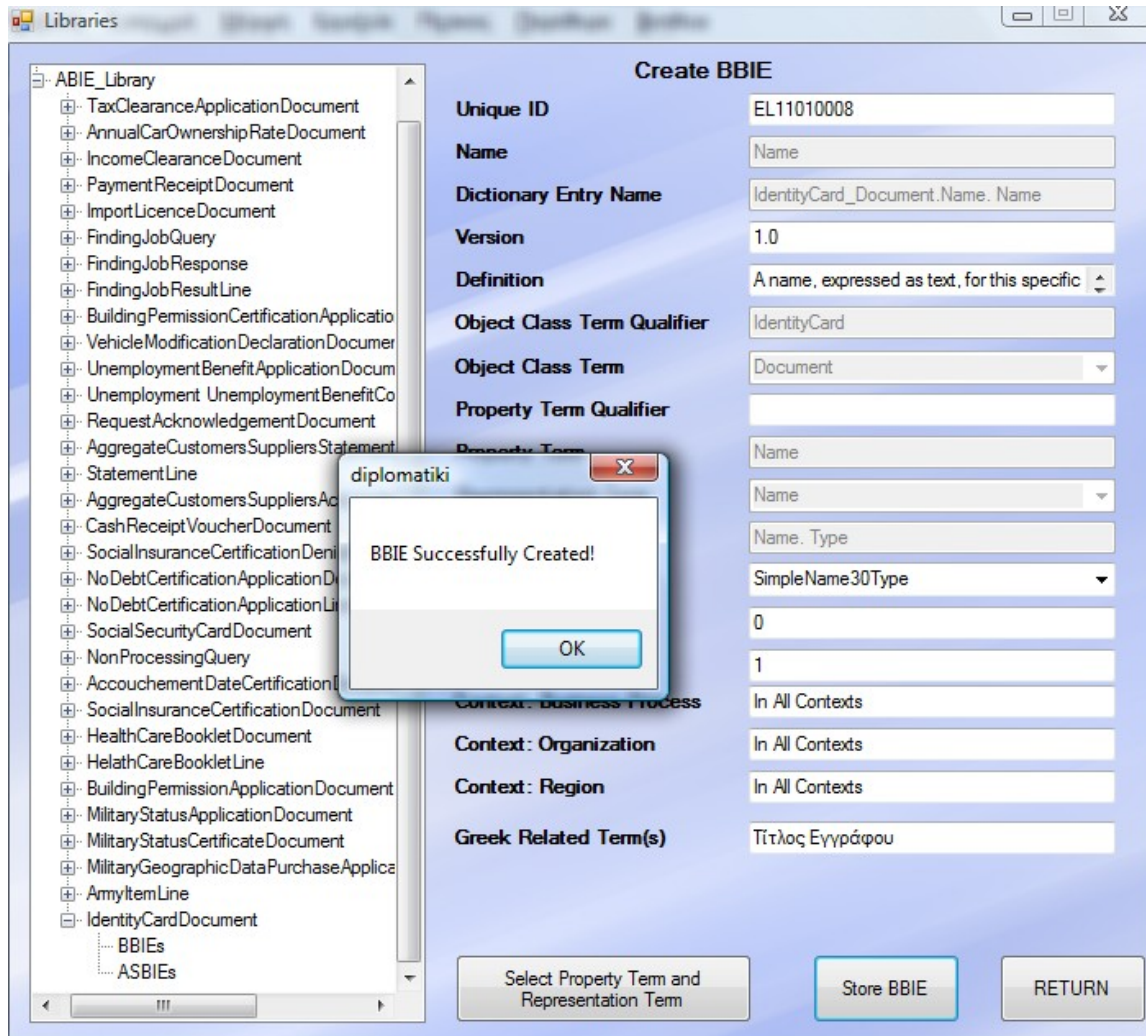


Το ενδιαφέρον με τη δημιουργία των BBIE και ASBIE είναι ότι δε μπορούν να είναι οποιουδήποτε είδους αλλά παίρνουν συγκεκριμένες επιτρεπόμενες τιμές ανάλογα με το ACC που το ABIE τους έχει ως template. Πιο συγκεκριμένα, κατά τη δημιουργία του BBIE, υπάρχει ένας επιτρεπόμενος συνδυασμός από Property Term και Representation Term που μπορεί να έχει και τα οποία ορίζονται από τα Property Term και Representation Term των BCC του ACC – template του ABIE του BBIE:



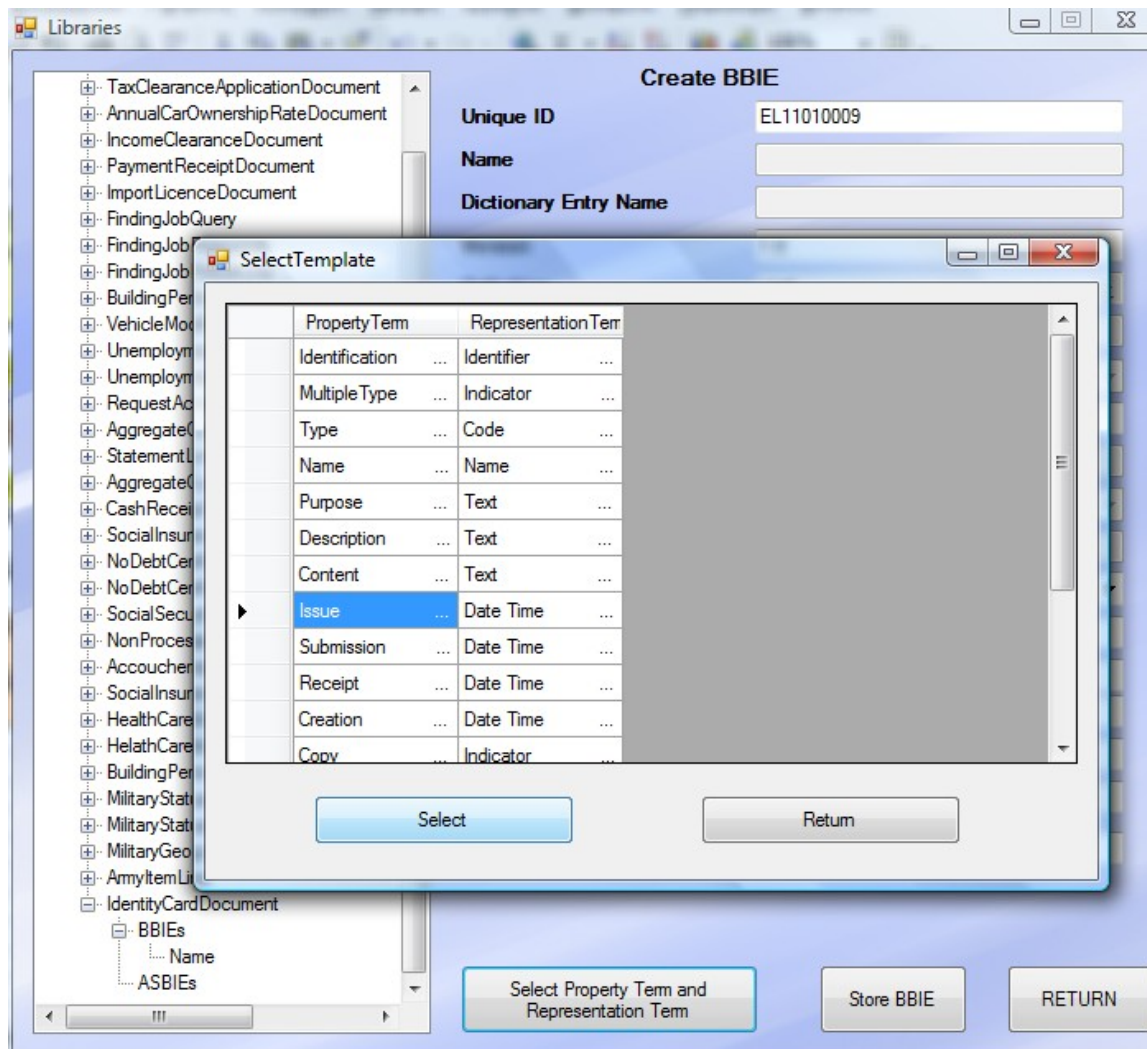
Εικόνα 4.3-57: Τεκμηρίωση (14)

Έπειτα επιλέγουμε δημιουργία και αποθήκευση:

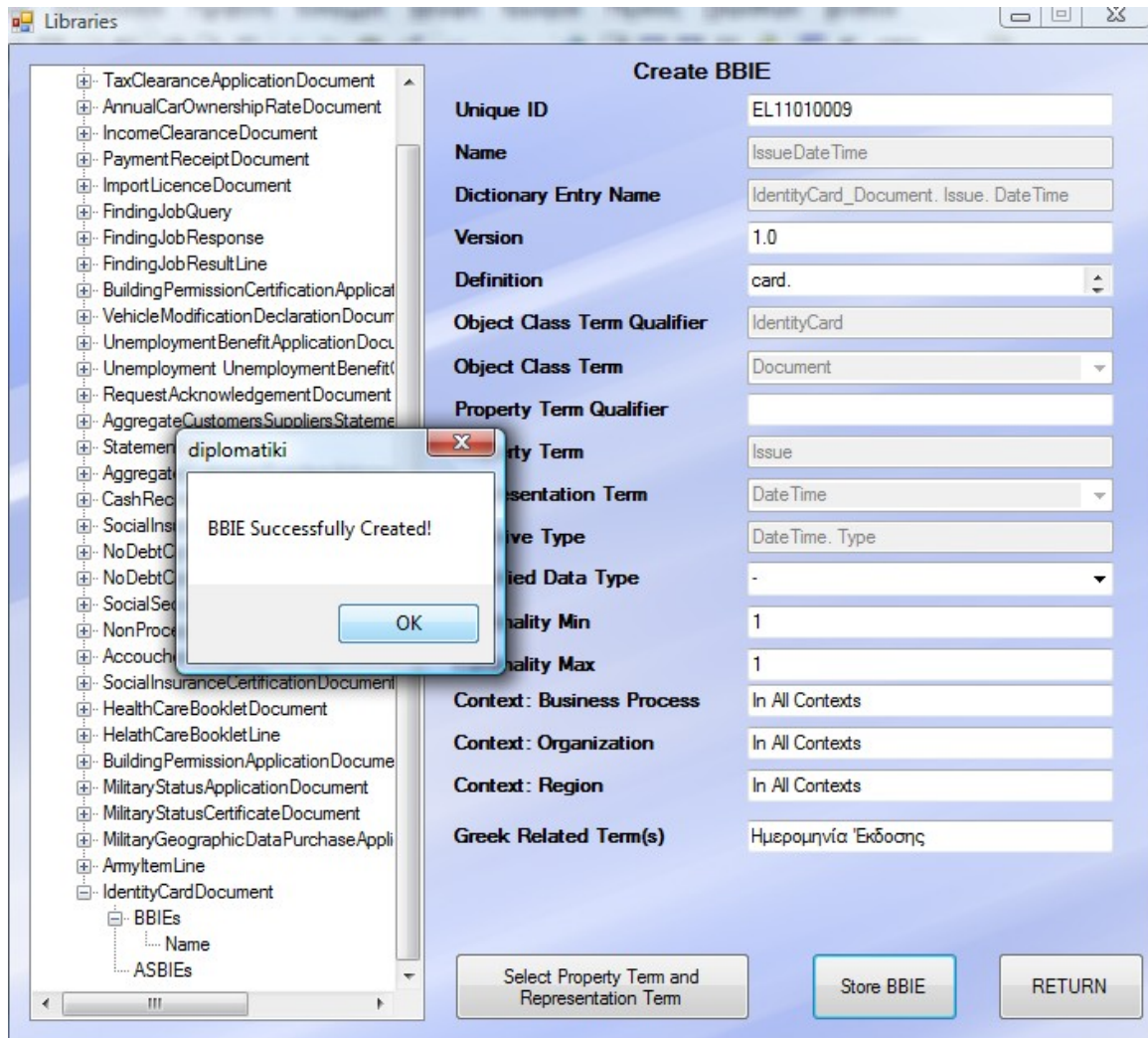


Εικόνα 4.3-58: Τεκμηρίωση (15)

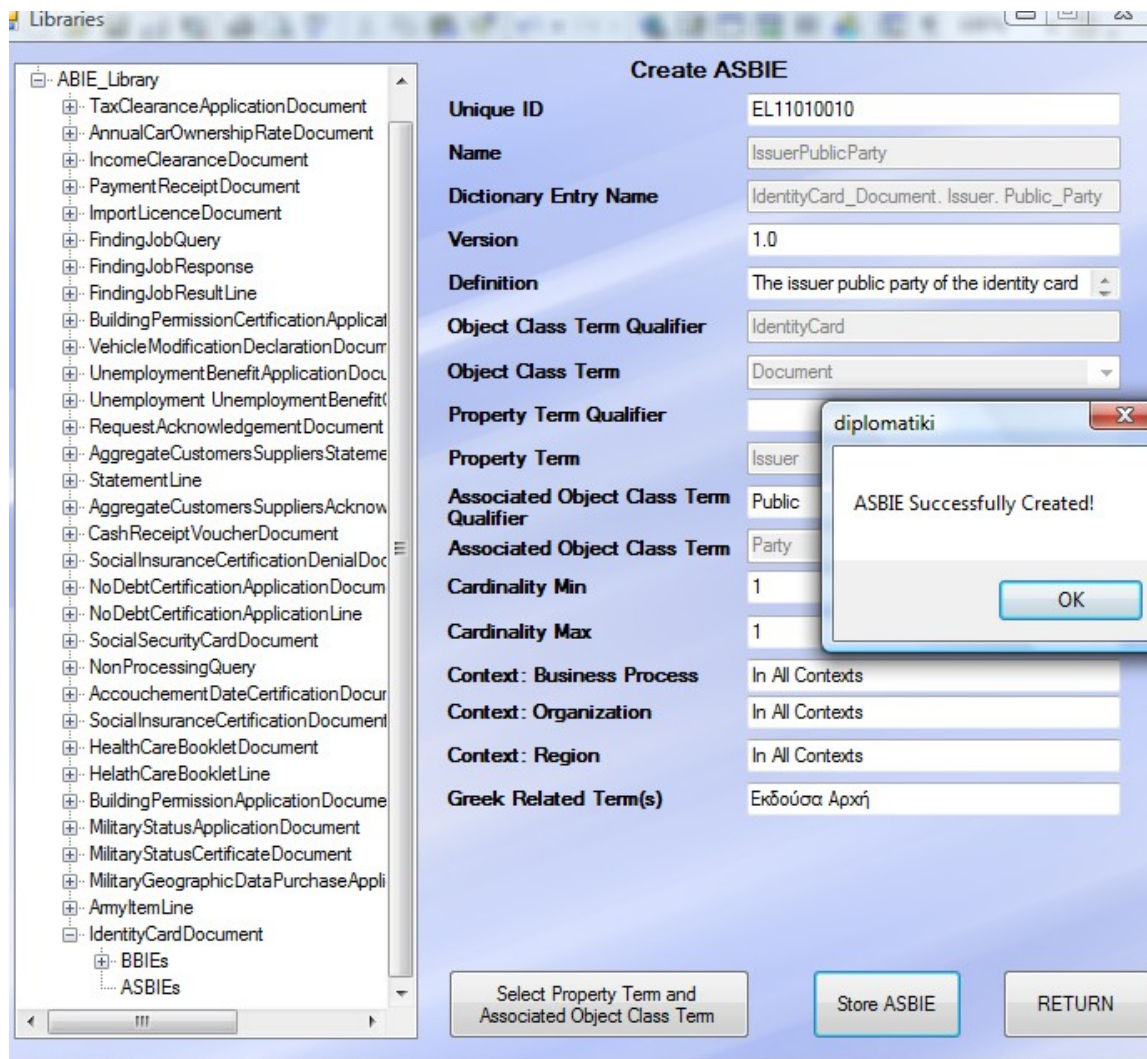
Ομοίως και για το 2^ο BBIE:



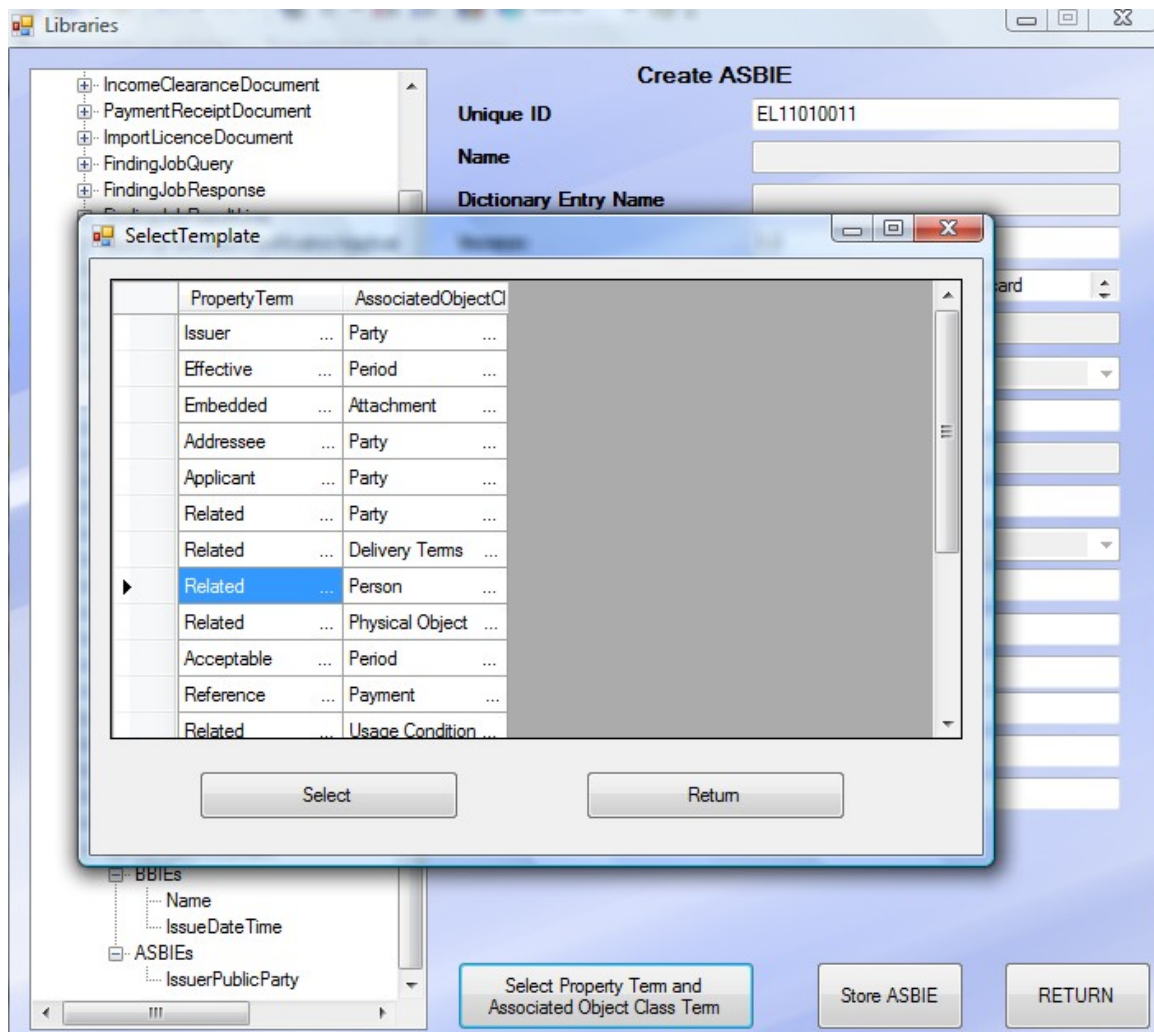
Εικόνα 4.3-59: Τεκμηρίωση (16)



Εικόνα 4.3-60: Τεκμηρίωση (17)

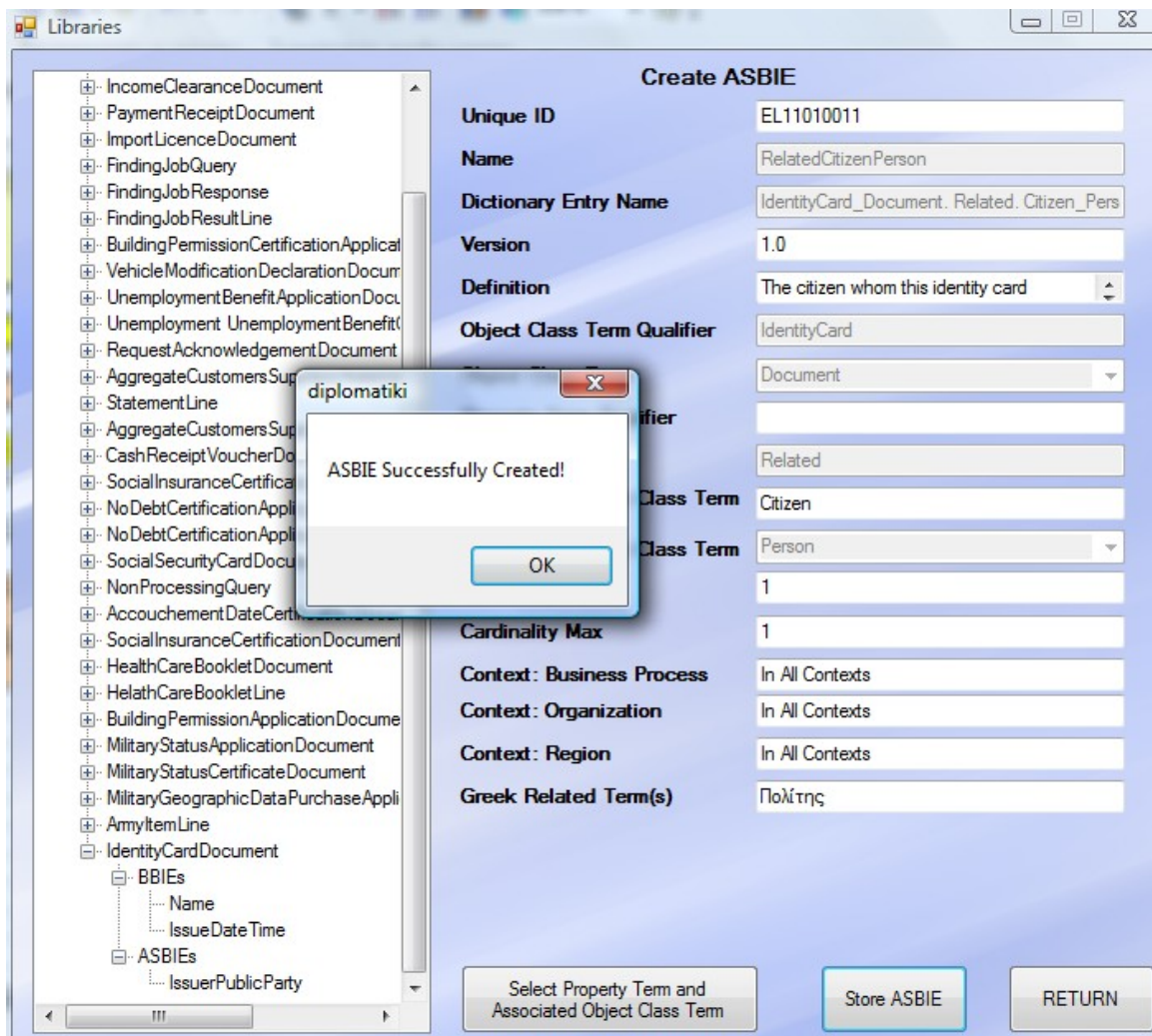


Εικόνα 4.3-61: Τεκμηρίωση (18)



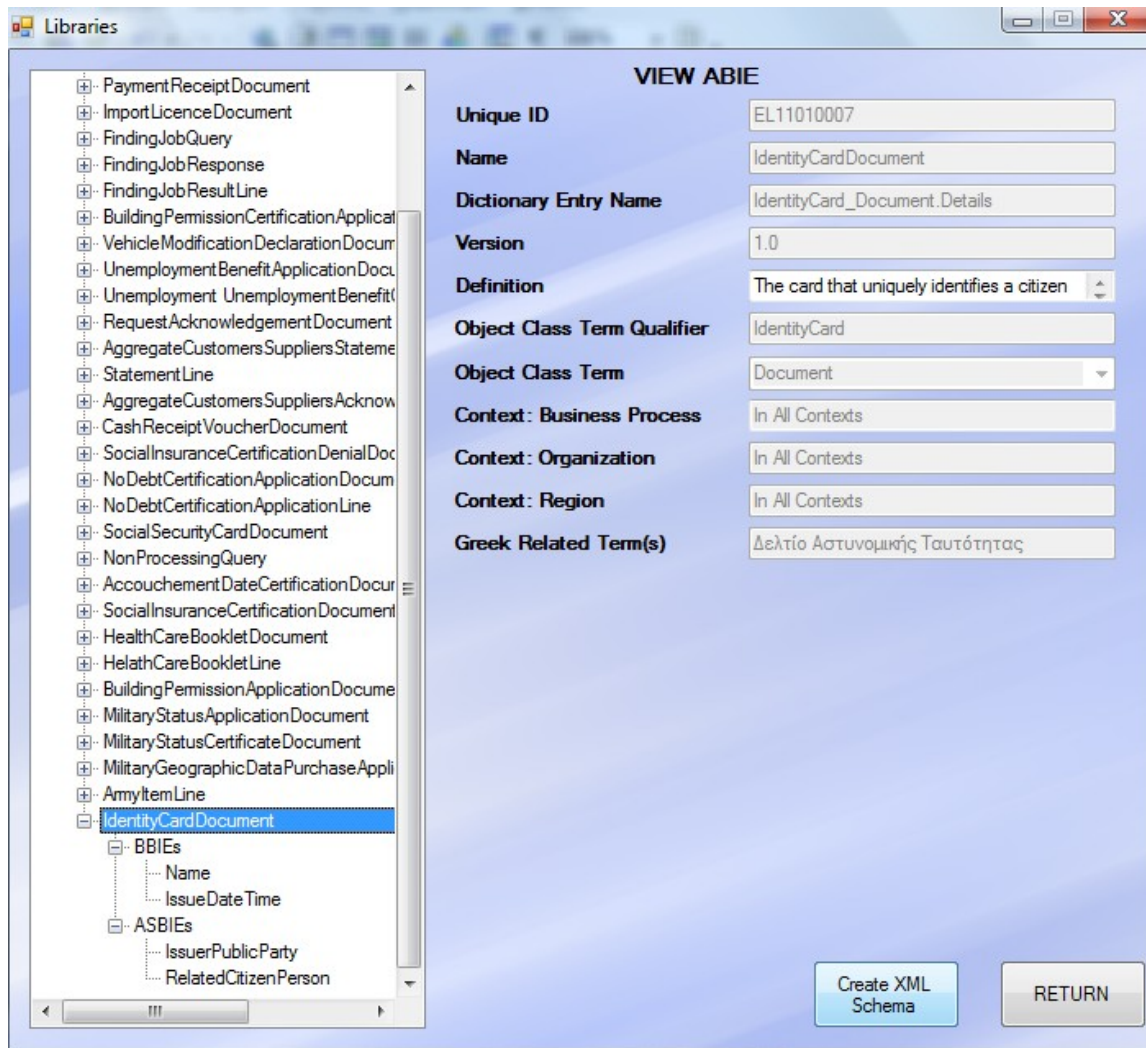
Εικόνα 4.3-62: Τεκμηρίωση (19)

Εδώ φαίνεται πάλι αυτό που αναφέρθηκε και πιο πριν για επιτρεπόμενο συνδυασμό Property Term και Associated Object Class Term σε αυτή την περίπτωση της δημιουργίας του ASBIE.

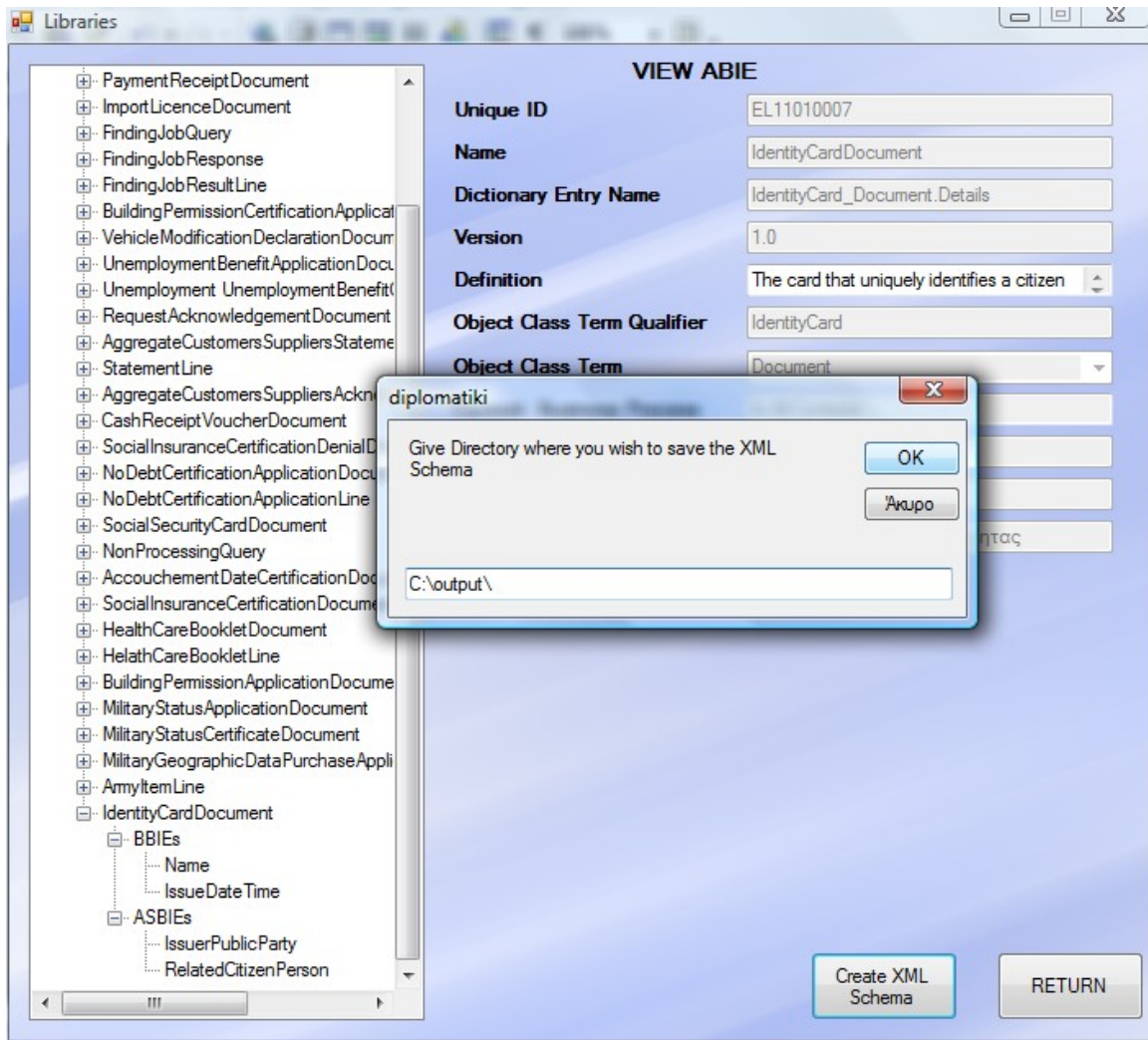


Εικόνα 4.3-63: Τεκμηρίωση (20)

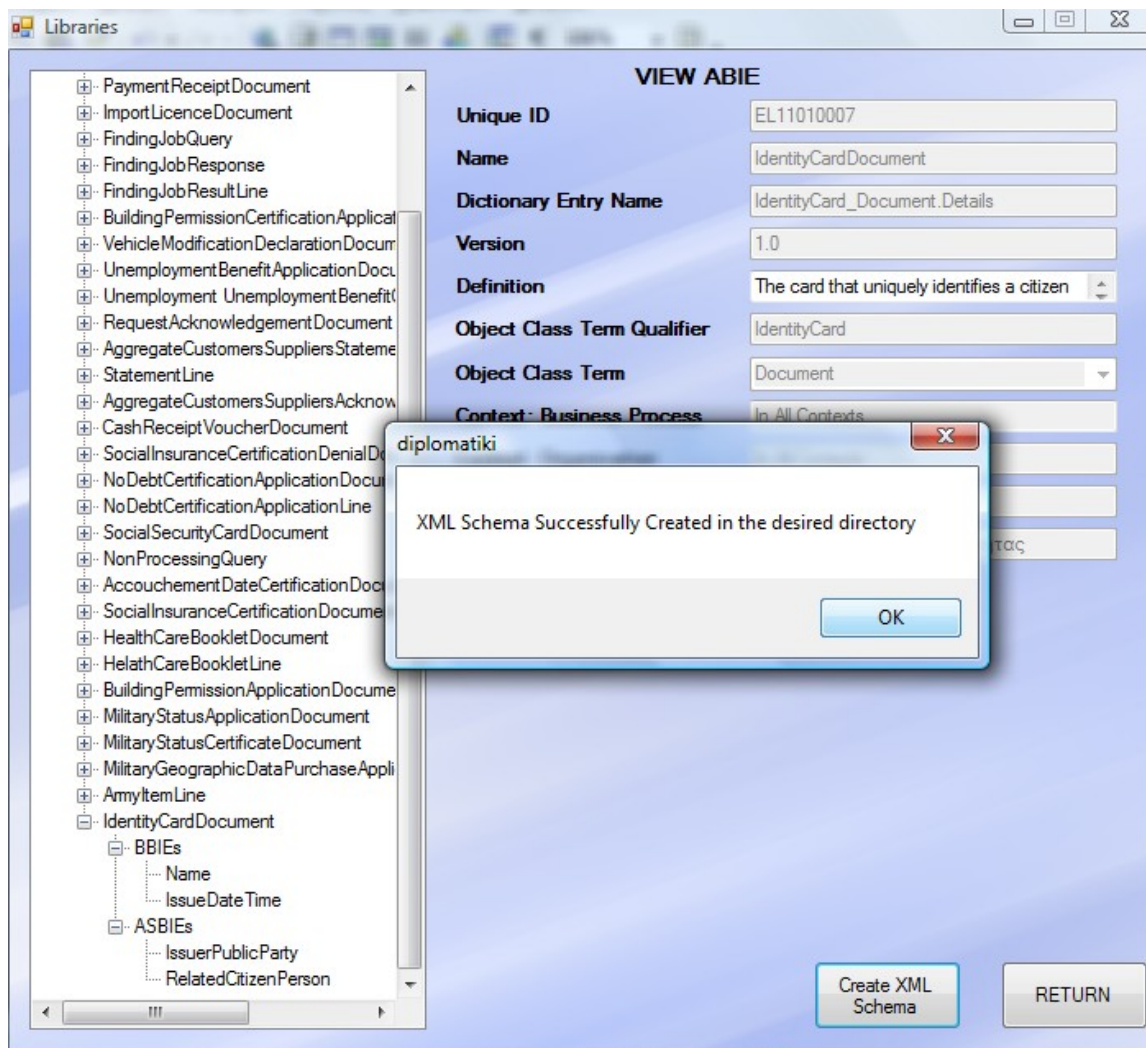
Και τώρα είναι το σημείο όπου επιλέγεται το ABIE που δημιουργήθηκε και θα ακολουθήσει η δημιουργία του XML Σχήματος αφού πρώτα δοθεί ο προορισμός στον οποίο επιθυμούμε να αποθηκευθεί το νέο XML Σχήμα. Έστω ότι θέλουμε να το αποθηκεύσουμε σε ένα φάκελο με το όνομα output που βρίσκεται στο φάκελο του σκληρού δίσκου C. Η εκτέλεση θα έχει ως εξής:



Εικόνα 4.3-64: Δημιουργία XML Σχήματος (1)



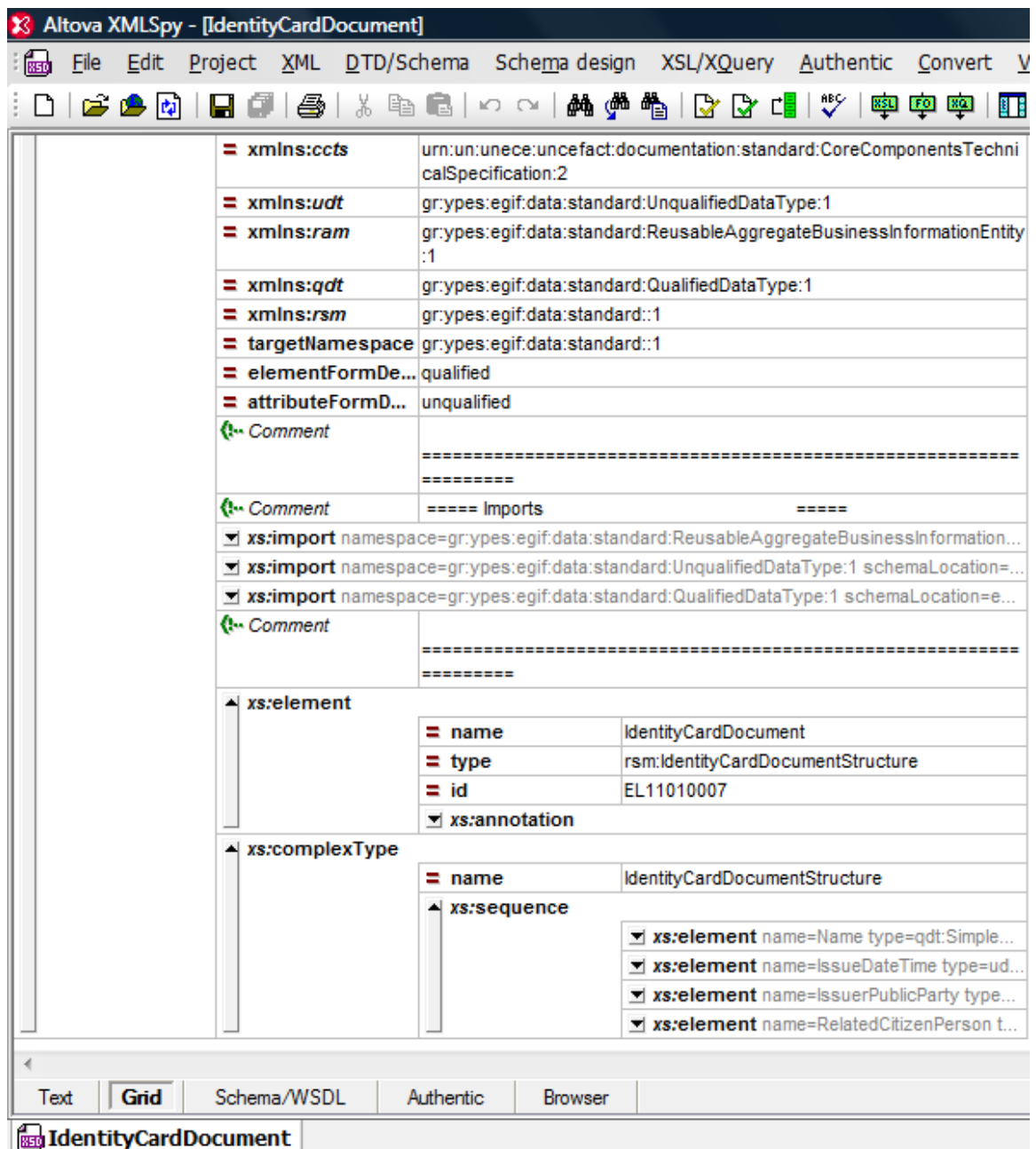
Εικόνα 4.3-65: Δημιουργία XML Σχήματος (2)



Εικόνα 4.3-66: Δημιουργία XML Σχήματος (3)

Και με αυτό τον τρόπο, δημιουργήθηκε το αρχείο IdentityCardDocument.xsd το οποίο είναι ένα well formed και valid XML Σχήμα το οποίο μοντελοποιεί το Δελτίο Αστυνομικής Ταυτότητας ενός πολίτη, καθορίζει δηλαδή το πώς θα πρέπει να είναι η δομή, ενός αρχείου *.xml, το οποίο θα περιέχει όλα εκείνα τα στοιχεία που απαρτίζουν το Δελτίο Αστυνομικής Ταυτότητας ενός πολίτη δομημένα όμως με συγκεκριμένο τρόπο και κανόνες.

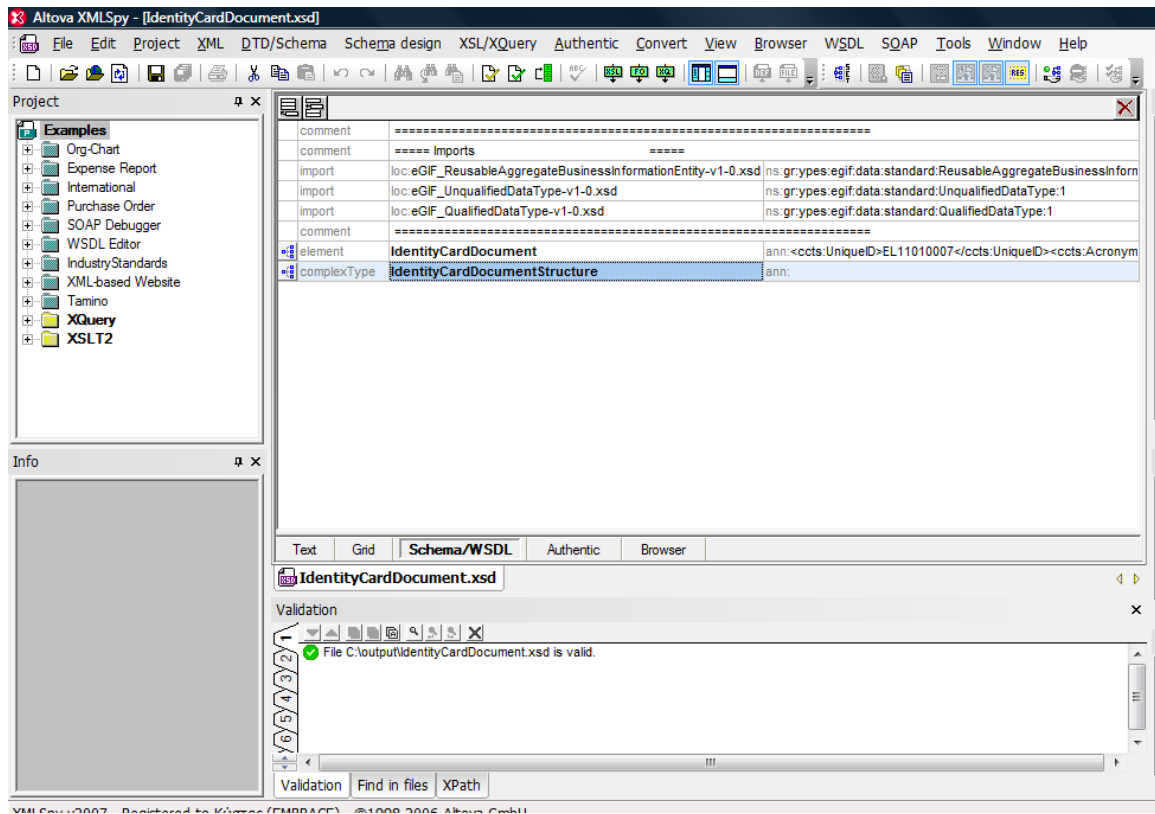
Απόδειξη του ότι το XML Σχήμα είναι valid και well formed αποτελεί η εκτέλεσή του με ένα ειδικό πρόγραμμα, το Altova XML SPY, ένα πρόγραμμα για σχεδιασμό και ανάπτυξη πολλών μορφών αρχείων σε XML και όχι μόνο, το οποίο δοθέντος των κανόνων που το CCTS πρότυπο έχει θέσει, κάνει ένα «πέρασμα» (parsing) όλου του XML σχήματος και ελέγχει την εγκυρότητά του ενώ δίνει και τη δομική του όψη. Ακολουθούν στιγμιότυπα εκτέλεσης του προγράμματος για το παραχθέν XML σχήμα της παρούσας εργασίας:



Εικόνα 4.3-67: Τεκμηρίωση validity & well form

Το πρόγραμμα κάνει import – δηλαδή εισάγει μια αναφορά σε τρία αρχεία τα οποία περιέχουν τους κανόνες που έχει θέσει το πρότυπο CCTS και το Υπουργείο Εσωτερικών στα πλαίσια του Ελληνικού Πλαισίου Ηλεκτρονικής Διακυβέρνησης προκειμένου να ελέγξει την εγκυρότητα του κειμένου του αρχείου που ακολουθεί. Στο παράθυρο εξόδου του XML Spy (παράθυρο validity) δεν υπάρχουν errors γεγονός που εξασφαλίζει την validity του XML Σχήματος.

Ακολουθεί και η τεκμηρίωση της validity:



Εικόνα 4.3-68: Validity of XML Schema


```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ccts="
urn:un:unece:uncefact:documentation:standard:CoreComponentsTechnicalSpecification:2" xmlns:udt="
gr:ypes:egif.data.standard.UnqualifiedDataType:1" xmlns:ram="
gr:ypes:egif.data.standard.ReusableAggregateBusinessInformationEntity:1" xmlns:qdt="gr:ypes:egif.data.standard.QualifiedDataType:1"
xmlns:rsm="gr:ypes:egif.data.standard:1" targetNamespace="gr:ypes:egif.data.standard:1" elementFormDefault="qualified"
attributeFormDefault="unqualified">
3 <!-- Imports -->
4 <!-- Imports -->
5 <xs:import namespace="gr:ypes:egif.data.standard.ReusableAggregateBusinessInformationEntity:1" schemaLocation="
eGIF_ReusableAggregateBusinessInformationEntity-v1-0.xsd"/>
6 <xs:import namespace="gr:ypes:egif.data.standard.UnqualifiedDataType:1" schemaLocation="eGIF_UnqualifiedDataType-v1-0.xsd"/>
7 <xs:import namespace="gr:ypes:egif.data.standard.QualifiedDataType:1" schemaLocation="eGIF_QualifiedDataType-v1-0.xsd"/>
8 <!-- Imports -->
9 <xs:element name="IdentityCardDocument" type="rsm:IdentityCardDocumentStructure" id="EL11010007">
10 <xs:annotation>
11 <xs:documentation xml:lang="en">
12 <ccts:UniqueID>EL11010007</ccts:UniqueID>
13 <ccts:Acronym>ACC</ccts:Acronym>
14 <ccts:DictionaryEntryName>IdentityCard_Document.Details</ccts:DictionaryEntryName>
15 <ccts:Version>1.0</ccts:Version>
16 <ccts:Definition>The card that uniquely identifies a citizen.</ccts:Definition>
17 <ccts:ObjectClassTermQualifier>IdentityCard</ccts:ObjectClassTermQualifier>
18 <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
19 <ccts:BusinessProcessContextValue>In all contexts</ccts:BusinessProcessContextValue>
20 <ccts:OrganizationContextValue>In all contexts</ccts:OrganizationContextValue>
21 <ccts:RegionContextValue>In all contexts</ccts:RegionContextValue>
22 </xs:documentation>
23 </xs:annotation>
24 </xs:element>
25 <xs:complexType name="IdentityCardDocumentStructure">
26 <xs:sequence>
27 <xs:element name="Name" type="qdt:SimpleName30Type" id="EL11010008" minOccurs="0" maxOccurs="0">
28 <xs:annotation>
29 <xs:documentation xml:lang="en">
30 <ccts:UniqueID>EL11010008</ccts:UniqueID>
31 <ccts:Acronym>BBIE</ccts:Acronym>
32 <ccts:DictionaryEntryName>IdentityCard_Document.Name.Name</ccts:DictionaryEntryName>
33 <ccts:Version>1.0</ccts:Version>
34 <ccts:Definition>A name, expressed as text, for this specific identity card.</ccts:Definition>
35 <ccts:ObjectClassTermQualifier>IdentityCard</ccts:ObjectClassTermQualifier>
36 <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
37 <ccts:PropertyTerm>Name</ccts:PropertyTerm>
38 <ccts:RepresentationTerm>Name</ccts:RepresentationTerm>
39 <ccts:Cardinality>0..1</ccts:Cardinality>
40 <ccts:BusinessProcessContextValue>In All Contexts</ccts:BusinessProcessContextValue>
41 <ccts:OrganizationContextValue>In All Contexts</ccts:OrganizationContextValue>
42 <ccts:RegionContextValue>In All Contexts</ccts:RegionContextValue>
43 </xs:documentation>
44 </xs:annotation>
45 </xs:element>
46 <xs:element name="IssueDateTime" type="udt:DateTimeType" id="EL11010009" minOccurs="0" maxOccurs="0">
47 <xs:annotation>
48 <xs:documentation xml:lang="en">
49 <ccts:UniqueID>EL11010009</ccts:UniqueID>
50 <ccts:Acronym>BBIE</ccts:Acronym>
51 <ccts:DictionaryEntryName>IdentityCard_Document.Issue.DateTime</ccts:DictionaryEntryName>
52 <ccts:Version>1.0</ccts:Version>
53 <ccts:Definition>The date, time, date time or other date time value for the issuance of this identity card.</ccts:Definition>
54 <ccts:ObjectClassTermQualifier>IdentityCard</ccts:ObjectClassTermQualifier>
55 <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
56 <ccts:PropertyTerm>Issue</ccts:PropertyTerm>
57 <ccts:RepresentationTerm>DateTime</ccts:RepresentationTerm>
58 <ccts:Cardinality>1..1</ccts:Cardinality>
59 <ccts:BusinessProcessContextValue>In All Contexts</ccts:BusinessProcessContextValue>
60 <ccts:OrganizationContextValue>In All Contexts</ccts:OrganizationContextValue>
61 <ccts:RegionContextValue>In All Contexts</ccts:RegionContextValue>
62 </xs:documentation>
63 </xs:annotation>
64 </xs:element>
65 <xs:element name="IssuerPublicParty" type="ram:PublicPartyStructure" id="EL11010010">
66 <xs:annotation>
67 <xs:documentation xml:lang="en">
68 <ccts:UniqueID>EL11010010</ccts:UniqueID>
69 <ccts:Acronym>ASBIE</ccts:Acronym>
70 <ccts:DictionaryEntryName>IdentityCard_Document.Issuer.Public_Pa</ccts:DictionaryEntryName>
71 <ccts:Version>1.0</ccts:Version>
72 <ccts:Definition>The issuer public party of the identity card.</ccts:Definition>
73 <ccts:ObjectClassTermQualifier>IdentityCard</ccts:ObjectClassTermQualifier>
74 <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
75 <ccts:PropertyTerm>Issuer</ccts:PropertyTerm>
76 <ccts:AssociatedObjectClassTermQualifier>Public</ccts:AssociatedObjectClassTermQualifier>
77 <ccts:AssociatedObjectClassTerm>Party</ccts:AssociatedObjectClassTerm>
78 <ccts:Cardinality>1..1</ccts:Cardinality>

```

```

79      <ccts:BusinessProcessContextValue>In All Contexts</ccts:BusinessProcessContextValue>
80      <ccts:OrganizationContextValue>In All Contexts</ccts:OrganizationContextValue>
81      <ccts:RegionContextValue>In All Contexts</ccts:RegionContextValue>
82    </xs:documentation>
83  </xs:annotation>
84 </xs:element>
85 <xs:element name="RelatedCitizenPerson" type="ram:CitizenPersonStructure" id="EL11010011">
86   <xs:annotation>
87     <xs:documentation xml:lang="en">
88       <ccts:UniqueID>EL11010011</ccts:UniqueID>
89       <ccts:Acronym>ASBIE</ccts:Acronym>
90       <ccts:DictionaryEntryName>IdentityCard_Document. Related. Citizen_</ccts:DictionaryEntryName>
91       <ccts:Version>1.0</ccts:Version>
92       <ccts:Definition>The citizen whom this identity card identifies</ccts:Definition>
93       <ccts:ObjectClassTermQualifier>IdentityCard</ccts:ObjectClassTermQualifier>
94       <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
95       <ccts:PropertyTerm>Related</ccts:PropertyTerm>
96       <ccts:AssociatedObjectClassTermQualifier>Citizen</ccts:AssociatedObjectClassTermQualifier>
97       <ccts:AssociatedObjectClassTerm>Person</ccts:AssociatedObjectClassTerm>
98       <ccts:Cardinality>1..1</ccts:Cardinality>
99       <ccts:BusinessProcessContextValue>In All Contexts</ccts:BusinessProcessContextValue>
100      <ccts:OrganizationContextValue>In All Contexts</ccts:OrganizationContextValue>
101      <ccts:RegionContextValue>In All Contexts</ccts:RegionContextValue>
102    </xs:documentation>
103  </xs:annotation>
104 </xs:element>
105 </xs:sequence>
106 </xs:complexType>
107 </xs:schema>
108

```

Εικόνα 4.3-69: Τελικό XML Schema

5

Επίλογος

Το παρόν κεφάλαιο αποτελεί τον επίλογο του τόμου της διπλωματικής εργασίας. Παρατίθενται τα συμπεράσματα από την εκπόνηση και χρήση του λογισμικού συστήματος καθώς επίσης και οι πιθανές μελλοντικές επεκτάσεις που θα μπορούσαν να εφαρμοσθούν σε αυτό.

5.1 Σύνοψη

Στα πλαίσια της παρούσας διπλωματικής εργασίας πραγματοποιήθηκε ο σχεδιασμός, η ανάπτυξη και η υλοποίηση ενός λογισμικού συστήματος το οποίο συνοπτικά:

- Υλοποιεί μια εφαρμογή που αυτοματοποιεί τη διαδικασία μοντελοποίησης πληροφορίας Ηλεκτρονικής Διακυβέρνησης και παράγει με επιτυχία ένα τελικό XML Σχήμα, το οποίο είναι well formed σύμφωνα με τους κανόνες της XML και valid σύμφωνα με το πρότυπο του Πλαισίου Ηλεκτρονικής Διακυβέρνησης και το οποίο αποτελεί τη βάση των επιχειρησιακών εγγράφων που ανταλλάσσονται μεταξύ των φορέων που προσφέρουν υπηρεσίες Ηλεκτρονικής Διακυβέρνησης στους πολίτες.
- Περιλαμβάνει την υλοποίηση μιας ενδεικτικής βάσης δεδομένων η οποία επικοινωνεί με την εφαρμογή που υλοποιήσαμε και έχει καταχωρημένα σε ξεχωριστούς πίνακες τα διάφορα Core Components και business information entities με συγκεκριμένο τρόπο και αρχιτεκτονική παρέχοντας τη δυνατότητα επεξεργασίας και διαχείρισής

τους μέσω του προγράμματος που υλοποιήθηκε χωρίς η ίδια η βάση να τα τροποποιεί.

- Περιλαμβάνει ειδικό μηχανισμό διαχείρισης σφαλμάτων με κατάλληλα μηνύματα ενημέρωσης του χρήστη για αποτυχία εκτέλεσης κάποιας λειτουργίας και το λόγο που την προκάλεσε ώστε να μην υπάρξει περίπτωση να «κολλήσει» το σύστημα και να μην αποκρίνεται. Συνεπώς, ο χρήστης θα έχει το δικαίωμα να αναιρέσει την επιλογή που οδήγησε σε σφάλμα και να κάνει μια νέα αίτηση για την εκάστοτε επιθυμητή λειτουργία που θα θέλει να πραγματοποιήσει.

Με την υλοποίηση του παραπάνω λογισμικού συστήματος υλοποιούνται, όπως άλλωστε ήταν επιθυμητό από την αρχική περιγραφή του αντικειμένου της διπλωματικής και της περιγραφής του φυσικού προβλήματος, τα ακόλουθα:

- Δημιουργία, επεξεργασία και αποθήκευση όλων των δομών που είναι απαραίτητες για την μοντελοποίηση δεδομένων Ηλεκτρονικής Διακυβέρνησης. Τώρα πια, τα δεδομένα αυτά έχουν συγκεκριμένο εννοιολογικό περιεχόμενο και διέπονται από ένα σύνολο σχέσεων μεταξύ τους όπως αυτό επιβάλλεται από τα πρότυπα για την μοντελοποίηση πληροφορίας Ηλεκτρονικής Διακυβέρνησης που αναλύθηκαν σε προηγούμενα κεφάλαια. Με λίγα λόγια λοιπόν, έχει δημιουργηθεί μια κοινή γλώσσα για να επικοινωνούν μεταξύ τους τα ετερογενή συστήματα που διαχειρίζονται και ανταλλάσσουν πληροφορία Ηλεκτρονικής Διακυβέρνησης. Δε χρειάζεται να μεταφράζονται πια τα δεδομένα από το ένα σύστημα στο άλλο αφού η λήψη τους από το σύστημα – αποστολέας συνεπάγεται αυτόματη αναγνώριση των δεδομένων και πλήρη αντιστοίχιση των διαφόρων πεδίων χωρίς την επίπονη και χρονοβόρα ενδιάμεση εργασία του ανθρώπου και χωρίς τον κίνδυνο απώλειας δεδομένων κατά τη μεταφορά.
- Διασφαλίζεται, λόγω της εξαντλητικής εφαρμογής των κανόνων μοντελοποίησης και των κανόνων της XML φυσικά, ότι τα τελικά παραγόμενα XML Σχήματα θα είναι έγκυρα και well formed ώστε να είναι δυνατή η παραπάνω επικοινωνία.

5.2 Συμπεράσματα

Το παρόν λογισμικό σύστημα δε θα είχε κανένα νόημα αν εστίαζε μόνο στην μοντελοποίηση των δεδομένων. Το σημαντικότερο κομμάτι του, είναι το τελικό XML Σχήμα που παράγεται, και το οποίο αποτελεί ένα πρότυπο που καθορίζει με πλήρη και αυστηρό τρόπο τη μορφή και τη δομή των XML αρχείων τα οποία περιέχουν την πληροφορία και αποτελούν το μέσον για την μετάδοσή της ανάμεσα στα διάφορα συστήματα. Και αφού η ίδια η XML δε μπορεί να αποδώσει σημασιολογικό περιεχόμενο στα δεδομένα που περιέχει κάθε αρχείο της, το σημαντικό αυτό έργο το αναλαμβάνουν τα παραγόμενα από το παρόν λογισμικό σύστημα XML Σχήματα, υλοποιώντας κατ' αυτό τον τρόπο την έννοια της Διαλειτουργικότητας στην Ηλεκτρονική Διακυβέρνηση.

Επιπλέον, το ευρύτερο μοντέλο συστήματος που προτείνεται στην παρούσα διπλωματική εργασία δεν βρίσκει εφαρμογή μόνο στο συγκεκριμένο λογισμικό σύστημα που παρουσιάστηκε, αλλά μπορεί να αποτελέσει υπόδειγμα για αντίστοιχες περιπτώσεις εκσυγχρονισμού διαδικασιών για την εξυπηρέτηση της διαλειτουργικότητας μεταξύ κρατικών οργανισμών ή μεταξύ οργανισμών και επιχειρήσεων.

Η μορφή που έχει, δηλαδή η μοντελοποίηση ενός συνόλου δεδομένων οποιουδήποτε τύπου με βάση συγκεκριμένους κανόνες, μπορεί να εφαρμοσθεί και σε άλλες περιπτώσεις για τραπεζικά, παραδείγματος χάριν, δεδομένα τα οποία θα μπορούσαν να μοντελοποιηθούν με βάση έναν ενιαίο κανονισμό για την αναπαράστασή τους και την επίτευξη διαλειτουργικότητας και σε τραπεζικά συστήματα. Είναι μεγάλο το πλήθος των διαφορετικών συστημάτων και των τύπων δεδομένων όπου θα μπορούσε να εφαρμοσθεί η λογική του παρόντος λογισμικού συστήματος και η παραγωγή ενός τελικού XML Σχήματος με διαφορετική δομή κάθε φορά.

Σημαντικό ρόλο στην ολοκλήρωση του παρόντος έργου έπαιξαν και οι διάφορες τεχνολογίες που χρησιμοποιήθηκαν. Αρχικά, η χρήση του Microsoft Visual Studio και της γλώσσας Visual Basic .NET προσέφερε τη δυνατότητα γρήγορης, φιλικής και αποδοτικής υλοποίησης των επιμέρους φορμών οι οποίες ενσωματώνουν τους διάφορους κανόνες μοντελοποίησης και συνιστούν στο σύνολό τους ένα περιβάλλον σαν αυτό που έχουν οι σύγχρονες εφαρμογές και τα σύγχρονα λογισμικά πακέτα. Αυτό οφείλεται στα χαρακτηριστικά και τις δυνατότητες που προσφέρουν η πλατφόρμα και η γλώσσα προγραμματισμού όπως αυτά αναλύθηκαν προηγουμένως στο αντίστοιχο κεφάλαιο. Έπειτα, η χρήση της βάσης δεδομένων και η εκμετάλλευση της τεχνολογίας ADO.NET προσφέρει τη σύνδεση και την αποδοτική διαχείριση μιας βάσης δεδομένων μέσω ενός

προγράμματος το οποίο είναι γραμμένο σε γλώσσα υψηλού επιπέδου. Τέλος, η χρήση της XML, η οποία έχει την δυνατότητα να ενσωματώνει σε ένα αρχείο δεδομένα με συγκεκριμένη μοντελοποίηση, όχι μόνο συμβάλλει στη μοντελοποίηση των δεδομένων αλλά αποδεσμεύει τα δεδομένα από το εκάστοτε σύστημα και καθιστά τη μεταφορά τους ανεξάρτητη συστήματος με βάση το ενιαίο πρότυπο και κανόνες που διέπουν την XML.

5.3 Μελλοντικές Επεκτάσεις

Το παρόν σύστημα, έτσι όπως είναι υλοποιημένο, απαιτεί από τον χρήστη να έχει εγκατεστημένο στο σύστημά του έναν SQL Server προκειμένου να διατηρεί τη βάση δεδομένων στην οποία είναι αποθηκευμένοι οι πίνακες με τα διάφορα Core Components και business information entities. Συνεπώς, από άποψη φορητότητας και εκτέλεσης της παρούσας εφαρμογής σε διαφορετικά υπολογιστικά συστήματα, το παρόν λογισμικό σύστημα έχει, σε αυτή του την έκδοση, ένα σημείο που επιδέχεται περαιτέρω βελτίωση.

Μια πρώτη ιδέα θα ήταν η χρησιμοποίηση XML αρχείων που θα φέρουν αποθηκευμένες τις προαναφερθείσες δομές. Δηλαδή, το σύστημά μας θα μοντελοποιούσε με τον ίδιο τρόπο την πληροφορία αλλά θα την αποθήκευε σε ένα αρχείο XML το οποίο σε μελλοντική κλήση του προγράμματος θα μπορούσε να χρησιμοποιήσει ώστε να ανακτήσει όλες τις βασικές δομές προκειμένου να δημιουργήσει το τελικό XML Σχήμα. Βέβαια, αυτή η λύση συνεπάγεται μεγάλο όγκο αρχείων και ξεφεύγει από την αρχική φιλοσοφία του παρόντος συστήματος λογισμικού.

Μια αρκετά ενδιαφέρουσα ιδέα είναι η χρήση του SQL Server Mobile Edition η οποία πολύ πρόσφατα αντικαταστάθηκε από τη νέα έκδοση SQL Server Everywhere. Το καινούργιο προϊόν υπόσχεται να είναι ένα «συμπαγές αλλά πλούσιο υποσύνολο» των δυνατοτήτων του SQL Server για χρήση σε desktop εφαρμογές ή συσκευές. Πρόκειται για μια ελαφριά, ενσωματωμένη βάση δεδομένων η οποία συνοδεύει το κυρίως πρόγραμμα και αποδεσμεύει τους τελικούς χρήστες από την εγκατάσταση της πλήρους έκδοσης του SQL Server. Βέβαια, αξίζει να σημειωθεί ότι οι ενσωματωμένες βάσεις δεδομένων είναι σχετικά καινούριο προϊόν και η κλασική και δοκιμασμένη λύση της πλήρους βάσης είναι ακόμα σημαντική και ασφαλής. Παρόλα αυτά, δεδομένης της γρήγορης εξέλιξης όλων των τεχνολογικών προϊόντων, η έκδοση του παρόντος λογισμικού συστήματος με ενσωματωμένη βάση δεδομένων είναι ικανή να το καταστήσει μελλοντικά ένα ασφαλές και ανταγωνιστικό πακέτο λογισμικού για την μοντελοποίηση πληροφορίας Ηλεκτρονικής Διακυβέρνησης.

Μια ακόμη δυνατότητα που θα μπορούσε να προστεθεί στο πρόγραμμα αυτό, θα ήταν η σύνδεση του χρήστη στον παγκόσμιο ιστό ώστε να λαμβάνει ενημερώσεις σχετικά με τα πρότυπα και τους κανόνες μοντελοποίησης, με τα βασικά δομικά στοιχεία τα οποία μπορεί να μεταβάλλονται ή να εμπλουτίζονται ή με άλλες παραμέτρους του συστήματος. Θα μπορούσε λοιπόν να προστεθεί κατάλληλος κώδικας για να ενημερώνεται το σύστημα και οι κανόνες που το διέπουν στη μοντελοποίηση και τελική δημιουργία XML Σχημάτων.

Τέλος, μια άλλη επέκταση που θα μπορούσε να προβλεφθεί, είναι η από κοινού χρήση της βάσης δεδομένων από πολλούς χρήστες μέσα στον εκάστοτε φορέα δημόσιας διοίκησης στην περίπτωση που δεν είναι μόνο ένα άτομο που έχει αναλάβει την μοντελοποίηση των δεδομένων αλλά περισσότερα τα οποία ενημερώνουν τη βάση από διαφορετικά σημεία και σε διαφορετικό χρόνο για τα δομικά στοιχεία και οντότητες που έχουν δημιουργήσει και καταχωρήσει. Θα μπορούσαν δηλαδή, να αποδοθούν στο πρόγραμμα χαρακτηριστικά εφαρμογής που εκτελείται σε επίπεδο δικτύου μέσα σε μια υπηρεσία ή φορέα δημόσιας διοίκησης.

6

Βιβλιογραφία

Κατά τη συγγραφή του παρόντος τόμου και του συστήματος της εφαρμογής χρησιμοποιήθηκε η βιβλιογραφία και οι αναφορές που ακολουθούν:

Βιβλιογραφία

- [1] Planet Ανώνυμη Εταιρία Παροχής Συμβουλευτικών Υπηρεσιών, Ερευνητικό Πανεπιστημιακό Ινστιτούτο Συστημάτων Επικοινωνιών και Υπολογιστών, ATHENS TECHNOLOGY CENTER A.B.E.T.E.. Μοντέλο Τεκμηρίωσης , 2007
- [2] United Nations Center for Trade Facilitaion and Electronic Business. Core Component Technical Specification – Part 8 of the ebXML Framework, 2003

Αναφορές:

- [3] Ελληνικό Πλαίσιο Διαλειτουργικότητας
Ηλεκτρονικής Διακυβέρνησης <http://www.e-gif.gov.gr>
- [4] Ευρωπαϊκή Ένωση: IDABC – Interoperable
Delivery of European eGovernment Services to
public Administrations, Businesses and Citizens <http://europa.eu.int/idabc/>

- [5] Ηνωμένο Βασίλειο: Government Interoperability Framework (e-GIF) <http://www.govtalk.gov.uk/>
- [6] Ηνωμένες Πολιτείες Αμερικής – NIEM (National Information Exchange Model) www.niem.gov
- [7] XML Schema Part 0: Primer Second Edition <http://www.w3.org/TR/xmlschema-0/>
- [8] XML Schema Part 1: Structures Second Edition <http://www.w3.org/TR/xmlschema-1/>
- [9] ebXML: Enabling a Global Electronic Market www.ebxml.org
- [10] Πλαίσιο Ηλεκτρονικής Διακυβέρνησης (ΠΗΔ) www.infosoc.gr
- [11] Κέντρο πληροφοριών προϊόντων της Microsoft <http://www.microsoft.com/products/info/>

7

Παράρτημα

Στο παράρτημα αυτό παρατίθενται ενδεικτικά μερικά από τα έγγραφα που μοντελοποιήθηκαν και των οποίων το τελικό XML Σχήμα παράχθηκε με επιτυχία, στο πλαίσιο της παρούσας διπλωματικής εργασίας. Τα έγγραφα αυτά είναι τα ακόλουθα:

- Πιστοποιητικό Οικογενειακής Κατάστασης (*FamilyStatusDocument.xsd*)
- Δελτίο Αστυνομικής Ταυτότητας (*IdentityCardDocument.xsd*)
- Αίτηση Φορολογικής Ενημερότητας (*TaxClearanceApplicationDocument.xsd*)
- Φορολογική Ενημερότητα (*TaxClearanceDocument.xsd*)

Και παρατίθενται στις επόμενες σελίδες:

7.1 FamilyStatusDocument.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ccts="urn:un:unece:uncefact:documentation:standard:CoreComponentsTechnicalSpecification:2"
  xmlns:udt="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
  xmlns:ram="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
  xmlns:qdt="gr:ypes:egif:data:standard:QualifiedDataType:1" xmlns:rsm="gr:ypes:egif:data:standard:1"
  targetNamespace="gr:ypes:egif:data:standard:1" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- ===== Imports ===== -->
  <!-- ===== Imports ===== -->
  <xs:import namespace="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
    schemaLocation="eGIF_ReusableAggregateBusinessInformationEntity-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
    schemaLocation="eGIF_UnqualifiedDataType-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:QualifiedDataType:1" schemaLocation="eGIF_QualifiedDataType-
    v1-0.xsd"/>
  <!-- ===== -->
  <xs:element name="FamilyStatusDocument" type="rsm:FamilyStatusDocumentStructure" id="EL11010000">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        <ccts:UniqueID>EL11010000</ccts:UniqueID>
        <ccts:Acronym>ACC</ccts:Acronym>
        <ccts:DictionaryEntryName>Family Status_Document.
Details</ccts:DictionaryEntryName>
        <ccts:Version>1</ccts:Version>
        <ccts:Definition>The family status document represents the certificate that municipalities
handle in citizens in order for them to prove their family status.</ccts:Definition>
        <ccts:ObjectClassTermQualifier>Family Status</ccts:ObjectClassTermQualifier>
        <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
        <ccts:BusinessProcessContextValue>In All
Contexts</ccts:BusinessProcessContextValue>
        <ccts:OrganizationContextValue>Municipality</ccts:OrganizationContextValue>
        <ccts:RegionContextValue>In All Contexts</ccts:RegionContextValue>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="FamilyStatusDocumentStructure">
    <xs:sequence>
      <xs:element name="ProtocolID" type="udt:IDType" id="EL11010001" minOccurs="0"
maxOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            <ccts:UniqueID>EL11010001</ccts:UniqueID>
            <ccts:Acronym>BBIE</ccts:Acronym>
            <ccts:DictionaryEntryName>Family Status_Document. Protocol_
Identification. Identifier</ccts:DictionaryEntryName>
            <ccts:Version>1</ccts:Version>
            <ccts:Definition>The protocol identifier for the
document</ccts:Definition>
            <ccts:ObjectClassTermQualifier>Family
Status</ccts:ObjectClassTermQualifier>
            <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
            <ccts:PropertyTermQualifier>Protocol
</ccts:PropertyTermQualifier>
            <ccts:PropertyTerm>Identification</ccts:PropertyTerm>
            <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
            <ccts:Cardinality>1..1</ccts:Cardinality>
            <ccts:BusinessProcessContextValue>In All
Contexts</ccts:BusinessProcessContextValue>
            <ccts:OrganizationContextValue>Municipality</ccts:OrganizationContextValue>
            <ccts:RegionContextValue>In All
Contexts</ccts:RegionContextValue>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Name" type="qdt:SimpleName30Type" id="EL11010002" minOccurs="0"
maxOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            <ccts:UniqueID>EL11010002</ccts:UniqueID>
            <ccts:Acronym>BBIE</ccts:Acronym>

```


Name</cts:DictionaryEntryName> <cts:DictionaryEntryName>Family Status_Document.
 certificate.</cts:Definition> <cts:Version>1</cts:Version>
 Status</cts:ObjectClassTermQualifier> <cts:Definition>A name, expressed as text, for this specific
 <cts:ObjectClassTermQualifier>Family
 <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
 <cts:PropertyTerm>Name</cts:PropertyTerm>
 <cts:RepresentationTerm>Name</cts:RepresentationTerm>
 <cts:Cardinality>0..1</cts:Cardinality>
 Contexts</cts:BusinessProcessContextValue> <cts:BusinessProcessContextValue>In All
 <cts:OrganizationContextValue>Municipality</cts:OrganizationContextValue>
 <cts:RegionContextValue>In All
 Contexts</cts:RegionContextValue>
 </xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="IssueDateTime" type="udt:DateTimeType" id="EL11010003" minOccurs="0"
 maxOccurs="0">
 </xs:annotation>
 <xs:documentation xml:lang="en">
 <cts:UniqueID>EL11010003</cts:UniqueID>
 <cts:Acronym>BBIE</cts:Acronym>
 <cts:DictionaryEntryName>Family Status_Document. Issue. Date
 T</cts:DictionaryEntryName>
 <cts:Version>1</cts:Version>
 the issuance of this document.</cts:Definition> <cts:Definition>The date, time, date time or other date value for
 Status</cts:ObjectClassTermQualifier> <cts:ObjectClassTermQualifier>Family
 <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
 <cts:PropertyTerm>Issue</cts:PropertyTerm>
 <cts:RepresentationTerm>DateTime</cts:RepresentationTerm>
 <cts:Cardinality>1..1</cts:Cardinality>
 Contexts</cts:BusinessProcessContextValue> <cts:BusinessProcessContextValue>In All
 <cts:OrganizationContextValue>Municipality</cts:OrganizationContextValue>
 <cts:RegionContextValue>In All
 Contexts</cts:RegionContextValue>
 </xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element name="IssuerPublicParty" type="ram:PublicPartyStructure" id="EL11010004">
 </xs:annotation>
 <xs:documentation xml:lang="en">
 <cts:UniqueID>EL11010004</cts:UniqueID>
 <cts:Acronym>ASBIE</cts:Acronym>
 <cts:DictionaryEntryName>Family Status_Document. Issuer.
 Public_Party</cts:DictionaryEntryName>
 <cts:Version>1.0</cts:Version>
 document</cts:Definition> <cts:Definition>The issuer public party of the
 Status</cts:ObjectClassTermQualifier> <cts:ObjectClassTermQualifier>Family
 <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
 <cts:PropertyTerm>Issuer</cts:PropertyTerm>
 <cts:AssociatedObjectClassTermQualifier>Public</cts:AssociatedObjectClassTermQualifier>
 <cts:AssociatedObjectClassTerm>Party</cts:AssociatedObjectClassTerm>
 <cts:Cardinality>1..1</cts:Cardinality>
 Contexts</cts:BusinessProcessContextValue> <cts:BusinessProcessContextValue>In All
 <cts:OrganizationContextValue>Municipality</cts:OrganizationContextValue>
 <cts:RegionContextValue>In All
 Contexts</cts:RegionContextValue>
 </xs:documentation>
 </xs:annotation>
 </xs:element>

```

id="EL11010005">
    <xs:element name="RelatedResidentPerson" type="ram:ResidentPersonStructure"
        <xs:annotation>
            <xs:documentation xml:lang="en">
                <ccts:UniqueID>EL11010005</ccts:UniqueID>
                <ccts:Acronym>ASBIE</ccts:Acronym>
                <ccts:DictionaryEntryName>Family Status_Document. Related.
Resident_Person</ccts:DictionaryEntryName>
                <ccts:Version>1.0</ccts:Version>
                <ccts:Definition>The person which is resident in a municipality
and whom this certificate concerns</ccts:Definition>
                <ccts:ObjectClassTermQualifier>Family
Status</ccts:ObjectClassTermQualifier>
                <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
                <ccts:PropertyTerm>Related</ccts:PropertyTerm>

                <ccts:AssociatedObjectClassTermQualifier>Resident</ccts:AssociatedObjectClassTermQualifier>

                <ccts:AssociatedObjectClassTerm>Person</ccts:AssociatedObjectClassTerm>
                <ccts:Cardinality>1..1</ccts:Cardinality>
                <ccts:BusinessProcessContextValue>In All
Contexts</ccts:BusinessProcessContextValue>

                <ccts:OrganizationContextValue>Municipality</ccts:OrganizationContextValue>
                <ccts:RegionContextValue>In All
Contexts</ccts:RegionContextValue>
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="RelatedSpousePerson" type="ram:SpousePersonStructure" id="EL11010006"
minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                <ccts:UniqueID>EL11010006</ccts:UniqueID>
                <ccts:Acronym>ASBIE</ccts:Acronym>
                <ccts:DictionaryEntryName>Family Status_Document. Related.
Spouse_Person</ccts:DictionaryEntryName>
                <ccts:Version>1.0</ccts:Version>
                <ccts:Definition>The spouse of the person this certificate concerns
if there is one</ccts:Definition>
                <ccts:ObjectClassTermQualifier>Family
Status</ccts:ObjectClassTermQualifier>
                <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
                <ccts:PropertyTerm>Related</ccts:PropertyTerm>

                <ccts:AssociatedObjectClassTermQualifier>Spouse</ccts:AssociatedObjectClassTermQualifier>

                <ccts:AssociatedObjectClassTerm>Person</ccts:AssociatedObjectClassTerm>
                <ccts:Cardinality>0..unbounded</ccts:Cardinality>
                <ccts:BusinessProcessContextValue>In All
Contexts</ccts:BusinessProcessContextValue>

                <ccts:OrganizationContextValue>Municipality</ccts:OrganizationContextValue>
                <ccts:RegionContextValue>In All
Contexts</ccts:RegionContextValue>
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="RelatedChildPerson" type="ram:ChildPersonStructure" id="EL11010007"
minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                <ccts:UniqueID>EL11010007</ccts:UniqueID>
                <ccts:Acronym>ASBIE</ccts:Acronym>
                <ccts:DictionaryEntryName>Family Status_Document. Related.
Child_Person</ccts:DictionaryEntryName>
                <ccts:Version>1.0</ccts:Version>
                <ccts:Definition>The children of the person whom this certificate
concerns of there are any</ccts:Definition>
                <ccts:ObjectClassTermQualifier>Family
Status</ccts:ObjectClassTermQualifier>
                <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
                <ccts:PropertyTerm>Related</ccts:PropertyTerm>

```

```

    <cts:AssociatedObjectClassTermQualifier>Child</cts:AssociatedObjectClassTermQualifier>

    <cts:AssociatedObjectClassTerm>Person</cts:AssociatedObjectClassTerm>
        <cts:Cardinality>0..unbounded</cts:Cardinality>
        <cts:BusinessProcessContextValue>In All

Contexts</cts:BusinessProcessContextValue>

    <cts:OrganizationContextValue>Municipality</cts:OrganizationContextValue>
        <cts:RegionContextValue>In All

Contexts</cts:RegionContextValue>
    </xs:documentation>
    </xs:annotation>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
</xs:schema>

```

7.2 IdentityCardDocument.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:cts="urn:un:unece:uncefact:documentation:standard:CoreComponentsTechnicalSpecification:2"
  xmlns:udt="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
  xmlns:ram="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
  xmlns:qdt="gr:ypes:egif:data:standard:QualifiedDataType:1" xmlns:rsm="gr:ypes:egif:data:standard:1"
  targetNamespace="gr:ypes:egif:data:standard:1" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- ===== Imports ===== -->
  <!-- ===== Imports ===== -->
  <xs:import namespace="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
    schemaLocation="eGIF_ReusableAggregateBusinessInformationEntity-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
    schemaLocation="eGIF_UnqualifiedDataType-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:QualifiedDataType:1" schemaLocation="eGIF_QualifiedDataType-
v1-0.xsd"/>
  <!-- ===== -->
  <xs:element name="IdentityCardDocument" type="rsm:IdentityCardDocumentStructure" id="EL11010007">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        <cts:UniqueID>EL11010007</cts:UniqueID>
        <cts:Acronym>ACC</cts:Acronym>

        <cts:DictionaryEntryName>IdentityCard_Document.Details</cts:DictionaryEntryName>
        <cts:Version>1.0</cts:Version>
        <cts:Definition>The card that uniquely identifies a citizen</cts:Definition>
        <cts:ObjectClassTermQualifier>IdentityCard</cts:ObjectClassTermQualifier>
        <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
        <cts:BusinessProcessContextValue>In All

Contexts</cts:BusinessProcessContextValue>
        <cts:OrganizationContextValue>In All Contexts</cts:OrganizationContextValue>
        <cts:RegionContextValue>In All Contexts</cts:RegionContextValue>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="IdentityCardDocumentStructure">
    <xs:sequence>
      <xs:element name="Name" type="qdt:SimpleName30Type" id="EL11010008" minOccurs="0"
maxOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            <cts:UniqueID>EL11010008</cts:UniqueID>
            <cts:Acronym>BBIE</cts:Acronym>
            <cts:DictionaryEntryName>IdentityCard_Document.Name.

Name</cts:DictionaryEntryName>
            <cts:Version>1.0</cts:Version>
            <cts:Definition>A name, expressed as text, for this specific
identity card.</cts:Definition>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

    <cts:ObjectClassTermQualifier>IdentityCard</cts:ObjectClassTermQualifier>
    <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
    <cts:PropertyTerm>Name</cts:PropertyTerm>
    <cts:RepresentationTerm>Name</cts:RepresentationTerm>
    <cts:Cardinality>0..1</cts:Cardinality>
    <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
    <cts:OrganizationContextValue>In All
Contexts</cts:OrganizationContextValue>
    <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
    </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="IssueDateTime" type="udt:DateTimeType" id="EL11010009" minOccurs="0"
maxOccurs="0">
    </xs:annotation>
    <xs:documentation xml:lang="en">
      <cts:UniqueID>EL11010009</cts:UniqueID>
      <cts:Acronym>BBIE</cts:Acronym>
      <cts:DictionaryEntryName>IdentityCard_Document. Issue.
DateTime</cts:DictionaryEntryName>
      <cts:Version>1.0</cts:Version>
      <cts:Definition>The date, time, date time or other date time value
for the issuance of this identity card.</cts:Definition>
    </xs:documentation>
  </xs:element>
  <xs:element name="IssuerPublicParty" type="ram:PublicPartyStructure" id="EL11010010">
    </xs:annotation>
    <xs:documentation xml:lang="en">
      <cts:UniqueID>EL11010010</cts:UniqueID>
      <cts:Acronym>ASBIE</cts:Acronym>
      <cts:DictionaryEntryName>IdentityCard_Document. Issuer.
Public_Pa</cts:DictionaryEntryName>
      <cts:Version>1.0</cts:Version>
      <cts:Definition>The issuer public party of the identity
card</cts:Definition>
    </xs:documentation>
  </xs:element>
  <xs:element name="RelatedCitizenPerson" type="ram:CitizenPersonStructure" id="EL11010011">
    </xs:annotation>
    <xs:documentation xml:lang="en">
      <cts:UniqueID>EL11010011</cts:UniqueID>
      <cts:Acronym>ASBIE</cts:Acronym>
    </xs:documentation>
  </xs:element>
  <cts:AssociatedObjectClassTermQualifier>Public</cts:AssociatedObjectClassTermQualifier>
  <cts:AssociatedObjectClassTerm>Party</cts:AssociatedObjectClassTerm>
  <cts:Cardinality>1..1</cts:Cardinality>
  <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
  <cts:OrganizationContextValue>In All
Contexts</cts:OrganizationContextValue>
  <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
  </xs:documentation>
  </xs:annotation>
  </xs:element>
  <xs:element name="RelatedCitizenPerson" type="ram:CitizenPersonStructure" id="EL11010011">
    </xs:annotation>
    <xs:documentation xml:lang="en">
      <cts:UniqueID>EL11010011</cts:UniqueID>
      <cts:Acronym>ASBIE</cts:Acronym>
    </xs:documentation>
  </xs:element>
  <cts:AssociatedObjectClassTermQualifier>IdentityCard</cts:AssociatedObjectClassTermQualifier>
  <cts:AssociatedObjectClassTerm>Document</cts:AssociatedObjectClassTerm>
  <cts:PropertyTerm>Issuer</cts:PropertyTerm>
  <cts:Cardinality>1..1</cts:Cardinality>
  <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
  <cts:OrganizationContextValue>In All
Contexts</cts:OrganizationContextValue>
  <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
  </xs:documentation>
  </xs:annotation>
  </xs:element>
  <xs:element name="RelatedCitizenPerson" type="ram:CitizenPersonStructure" id="EL11010011">
    </xs:annotation>
    <xs:documentation xml:lang="en">
      <cts:UniqueID>EL11010011</cts:UniqueID>
      <cts:Acronym>ASBIE</cts:Acronym>
    </xs:documentation>
  </xs:element>

```

```

Citizen_ </cts:DictionaryEntryName>
<cts:DictionaryEntryName>IdentityCard_Document. Related.
<cts:Version>1.0</cts:Version>
identifies.</cts:Definition>
<cts:Definition>The citizen whom this identity card

<cts:ObjectClassTermQualifier>IdentityCard</cts:ObjectClassTermQualifier>
<cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
<cts:PropertyTerm>Related</cts:PropertyTerm>

<cts:AssociatedObjectClassTermQualifier>Citizen</cts:AssociatedObjectClassTermQualifier>

<cts:AssociatedObjectClassTerm>Person</cts:AssociatedObjectClassTerm>
<cts:Cardinality>1..1</cts:Cardinality>
<cts:BusinessProcessContextValue>In All

Contexts</cts:BusinessProcessContextValue>
<cts:OrganizationContextValue>In All

Contexts</cts:OrganizationContextValue>
<cts:RegionContextValue>In All

Contexts</cts:RegionContextValue>
</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>

```

7.3 TaxClearanceApplicationDocument.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:cts="urn:un:unece:uncefact:documentation:standard:CoreComponentsTechnicalSpecification:2"
  xmlns:udt="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
  xmlns:ram="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
  xmlns:qdt="gr:ypes:egif:data:standard:QualifiedDataType:1" xmlns:rsm="gr:ypes:egif:data:standard:1"
  targetNamespace="gr:ypes:egif:data:standard:1" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- ===== Imports ===== -->
  <xs:import namespace="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
    schemaLocation="eGIF_ReusableAggregateBusinessInformationEntity-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
    schemaLocation="eGIF_UnqualifiedDataType-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:QualifiedDataType:1" schemaLocation="eGIF_QualifiedDataTypes-
    v1-0.xsd"/>
  <!-- ===== -->
  <xs:element name="TaxClearanceApplicationDocument" type="rsm:TaxClearanceApplicationDocumentStructure"
    id="EL11010020">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        <cts:UniqueID>EL11010020</cts:UniqueID>
        <cts:Acronym>ACC</cts:Acronym>
        <cts:DictionaryEntryName>Tax Clearance
Application_Document.Details</cts:DictionaryEntryName>
        <cts:Version>1</cts:Version>
        <cts:Definition>The tax clearance application represents the document with which a tax
payer declares its intention to obtain a tax clearance certificate by the competent tax authority</cts:Definition>
        <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
        <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
        <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
        <cts:OrganizationContextValue>Taxation Authority</cts:OrganizationContextValue>
        <cts:RegionContextValue>In All Contexts</cts:RegionContextValue>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="TaxClearanceApplicationDocumentStructure">
    <xs:sequence>
      <xs:element name="MultipleTypeIndicator" type="udt:IndicatorType" id="EL11010021"
minOccurs="0" maxOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">

```

```

        <cts:UniqueID>EL11010021</cts:UniqueID>
        <cts:Acronym>BBIE</cts:Acronym>
        <cts:DictionaryEntryName>Tax Clearance
Application_Document.Multiple Type.Indicator</cts:DictionaryEntryName>
        <cts:Version>1</cts:Version>
        <cts:Definition>The indication of whether or not a document is
an aggregation of different types of governmental documents</cts:Definition>
        <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
        <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
        <cts:PropertyTerm>Multiple Type</cts:PropertyTerm>
        <cts:RepresentationTerm>Indicator</cts:RepresentationTerm>
        <cts:Cardinality>1..1</cts:Cardinality>
        <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
        <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
        <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element name="Name" type="qdt:SimpleName30Type" id="EL11010022" minOccurs="0"
maxOccurs="0">
        </xs:annotation>
        <xs:documentation xml:lang="en">
        <cts:UniqueID>EL11010022</cts:UniqueID>
        <cts:Acronym>BBIE</cts:Acronym>
        <cts:DictionaryEntryName>Tax Clearance
Application_Document.Name.Name</cts:DictionaryEntryName>
        <cts:Version>1</cts:Version>
        <cts:Definition>A name, expressed as text, for this specific
certificate</cts:Definition>
        <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
        <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
        <cts:PropertyTerm>Name</cts:PropertyTerm>
        <cts:RepresentationTerm>Name</cts:RepresentationTerm>
        <cts:Cardinality>0..1</cts:Cardinality>
        <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
        <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
        <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element name="SubmissionDateTime" type="udt:DateTimeType" id="EL11010023"
minOccurs="0" maxOccurs="0">
        </xs:annotation>
        <xs:documentation xml:lang="en">
        <cts:UniqueID>EL11010023</cts:UniqueID>
        <cts:Acronym>BBIE</cts:Acronym>
        <cts:DictionaryEntryName>Tax Clearance
Application_Document.Submission.Date Time</cts:DictionaryEntryName>
        <cts:Version>1</cts:Version>
        <cts:Definition>The date, time, date time or other date time value
for the submission of this document</cts:Definition>
        <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
        <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
        <cts:PropertyTerm>Submission</cts:PropertyTerm>
        <cts:RepresentationTerm>DateTime</cts:RepresentationTerm>
        <cts:Cardinality>1..1</cts:Cardinality>
        <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
        <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
        <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element name="CopyNumeric" type="qdt:PositiveIntegerNumericType" id="EL11010024"
minOccurs="0" maxOccurs="0">
        </xs:annotation>

```

```

        <xs:documentation xml:lang="en">
            <cts:UniqueID>EL11010024</cts:UniqueID>
            <cts:Acronym>BBIE</cts:Acronym>
            <cts:DictionaryEntryName>Tax Clearance
Application_Document.Copy.Numeric</cts:DictionaryEntryName>
            <cts:Version>1</cts:Version>
            <cts:Definition>The number of requested copies for the tax
clearance document</cts:Definition>
            <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
            <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
            <cts:PropertyTerm>Copy</cts:PropertyTerm>
            <cts:RepresentationTerm>Numeric</cts:RepresentationTerm>
            <cts:Cardinality>1..1</cts:Cardinality>
            <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
            <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
            <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="DeclarationContent" type="udt:TextType" id="EL11010025" minOccurs="0"
maxOccurs="0">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            <cts:UniqueID>EL11010025</cts:UniqueID>
            <cts:Acronym>BBIE</cts:Acronym>
            <cts:DictionaryEntryName>Tax Clearance
Application_Document.Declaration_Content.Text</cts:DictionaryEntryName>
            <cts:Version>1</cts:Version>
            <cts:Definition>A textual description of the content of this
document in case it cannot be decomposed</cts:Definition>
            <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
            <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
            <cts:PropertyTermQualifier>Declaration
</cts:PropertyTermQualifier>
            <cts:PropertyTerm>Content</cts:PropertyTerm>
            <cts:RepresentationTerm>Text</cts:RepresentationTerm>
            <cts:Cardinality>0..1</cts:Cardinality>
            <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
            <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
            <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="AddresseePublicParty" type="ram:PublicPartyStructure" id="EL11010026">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            <cts:UniqueID>EL11010026</cts:UniqueID>
            <cts:Acronym>ASBIE</cts:Acronym>
            <cts:DictionaryEntryName>Tax Clearance
Application_Document.Addressee.Public_Party</cts:DictionaryEntryName>
            <cts:Version>1.0</cts:Version>
            <cts:Definition>The addressee party of the
document</cts:Definition>
            <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
            <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
            <cts:PropertyTerm>Addressee</cts:PropertyTerm>
            <cts:AssociatedObjectClassTermQualifier>Public</cts:AssociatedObjectClassTermQualifier>
            <cts:AssociatedObjectClassTerm>Party</cts:AssociatedObjectClassTerm>
            <cts:Cardinality>1..1</cts:Cardinality>
            <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
            <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
            <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
    </xs:annotation>

```



```

        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ApplicantTaxPayerParty" type="ram:TaxPayerPartyStructure"
id="EL11010027">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          <cts:UniqueID>EL11010027</cts:UniqueID>
          <cts:Acronym>ASBIE</cts:Acronym>
          <cts:DictionaryEntryName>Tax Clearance
Application_Document.Applicant.Tax Payer_Party</cts:DictionaryEntryName>
          <cts:Version>1.0</cts:Version>
          <cts:Definition>The tax payer (person or business) to which this
document addresses</cts:Definition>
          <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
          <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
          <cts:PropertyTerm>Applicant</cts:PropertyTerm>
          <cts:AssociatedObjectClassTermQualifier>Tax
Payer</cts:AssociatedObjectClassTermQualifier>
          <cts:AssociatedObjectClassTerm>Party</cts:AssociatedObjectClassTerm>
          <cts:Cardinality>1..1</cts:Cardinality>
          <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
          <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
          <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SpouseApplicantTaxPayerPerson" type="ram:TaxPayerPersonStructure"
id="EL11010028" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          <cts:UniqueID>EL11010028</cts:UniqueID>
          <cts:Acronym>ASBIE</cts:Acronym>
          <cts:DictionaryEntryName>Tax Clearance
Application_Document.Spouse_Applicant.Tax Payer_Person</cts:DictionaryEntryName>
          <cts:Version>1.0</cts:Version>
          <cts:Definition>The tax payer's spouse in case the applicant is a
married woman</cts:Definition>
          <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
          <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
          <cts:PropertyTermQualifier>Spouse</cts:PropertyTermQualifier>
          <cts:PropertyTerm>Applicant</cts:PropertyTerm>
          <cts:AssociatedObjectClassTermQualifier>Tax
Payer</cts:AssociatedObjectClassTermQualifier>
          <cts:AssociatedObjectClassTerm>Person</cts:AssociatedObjectClassTerm>
          <cts:Cardinality>0..1</cts:Cardinality>
          <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
          <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
          <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ApplicantRelatedTaxPayerParty" type="ram:TaxPayerPartyStructure"
id="EL11010029" minOccurs="0" maxOccurs="3">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          <cts:UniqueID>EL11010029</cts:UniqueID>
          <cts:Acronym>ASBIE</cts:Acronym>
          <cts:DictionaryEntryName>Tax Clearance
Application_Document.Applicant_Related.Tax Payer_Party</cts:DictionaryEntryName>
          <cts:Version>1.0</cts:Version>
          <cts:Definition>The businesses whose member or representative
is the applicant issuer tax payer party.</cts:Definition>
          <cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>

```



```

<cts:ObjectClassTerm>Document</cts:ObjectClassTerm>

<cts:PropertyTermQualifier>Applicant</cts:PropertyTermQualifier>
<cts:PropertyTerm>Related</cts:PropertyTerm>
Payer</cts:AssociatedObjectClassTermQualifier>
<cts:AssociatedObjectClassTerm>Party</cts:AssociatedObjectClassTerm>
<cts:Cardinality>0..3</cts:Cardinality>
Contexts</cts:BusinessProcessContextValue>
<cts:BusinessProcessContextValue>In All
Authority</cts:OrganizationContextValue>
<cts:OrganizationContextValue>Taxation
Contexts</cts:RegionContextValue>
<cts:RegionContextValue>In All
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="RelatedSimplifiedUsageCondition"
type="ram:SimplifiedUsageConditionStructure" id="EL11010030">
<xs:annotation>
<xs:documentation xml:lang="en">
<cts:UniqueID>EL11010030</cts:UniqueID>
<cts:Acronym>ASBIE</cts:Acronym>
<cts:DictionaryEntryName>Tax Clearance
Application_Document.Related.Simplified_Usage_Condition</cts:DictionaryEntryName>
<cts:Version>1.0</cts:Version>
<cts:Definition>The usage condition for the requested tax
clearance document</cts:Definition>
<cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
<cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
<cts:PropertyTerm>Related</cts:PropertyTerm>

<cts:AssociatedObjectClassTermQualifier>Simplified</cts:AssociatedObjectClassTermQualifier>
<cts:AssociatedObjectClassTerm>Usage
Condition</cts:AssociatedObjectClassTerm>
<cts:Cardinality>1..1</cts:Cardinality>
Contexts</cts:BusinessProcessContextValue>
<cts:BusinessProcessContextValue>In All
Authority</cts:OrganizationContextValue>
<cts:OrganizationContextValue>Taxation
Contexts</cts:RegionContextValue>
<cts:RegionContextValue>In All
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="DeclarationContentSimplifiedHierarchicalStructure"
type="ram:SimplifiedHierarchicalStructure" id="EL11010031" minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation xml:lang="en">
<cts:UniqueID>EL11010031</cts:UniqueID>
<cts:Acronym>ASBIE</cts:Acronym>
<cts:DictionaryEntryName>Tax Clearance
Application_Document.Declaration_Content.Simplified_Hierarchical_Structure</cts:DictionaryEntryName>
<cts:Version>1.0</cts:Version>
<cts:Definition>An hierarchical structure for the content of the
declaration in case it can be decomposed</cts:Definition>
<cts:ObjectClassTermQualifier>Tax Clearance
Application</cts:ObjectClassTermQualifier>
<cts:ObjectClassTerm>Document</cts:ObjectClassTerm>

<cts:PropertyTermQualifier>Declaration</cts:PropertyTermQualifier>
<cts:PropertyTerm>Content</cts:PropertyTerm>

<cts:AssociatedObjectClassTermQualifier>Simplified</cts:AssociatedObjectClassTermQualifier>
<cts:AssociatedObjectClassTerm>Hierarchical
Structure</cts:AssociatedObjectClassTerm>
<cts:Cardinality>0..unbounded</cts:Cardinality>
Contexts</cts:BusinessProcessContextValue>
<cts:BusinessProcessContextValue>In All
Authority</cts:OrganizationContextValue>
<cts:OrganizationContextValue>Taxation
Contexts</cts:RegionContextValue>
<cts:RegionContextValue>In All

```

```
</xs:documentation>  
</xs:annotation>  
</xs:element>  
</xs:sequence>  
</xs:complexType>  
</xs:schema>
```

7.4 TaxClearanceDocument.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ccts="urn:un:unece:uncefact:documentation:standard:CoreComponentsTechnicalSpecification:2"
  xmlns:udt="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
  xmlns:ram="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
  xmlns:qdt="gr:ypes:egif:data:standard:QualifiedDataType:1" xmlns:rsm="gr:ypes:egif:data:standard:1"
  targetNamespace="gr:ypes:egif:data:standard:1" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- ===== Imports ===== -->
  <xs:import namespace="gr:ypes:egif:data:standard:ReusableAggregateBusinessInformationEntity:1"
    schemaLocation="eGIF_ReusableAggregateBusinessInformationEntity-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:UnqualifiedDataType:1"
    schemaLocation="eGIF_UnqualifiedDataType-v1-0.xsd"/>
  <xs:import namespace="gr:ypes:egif:data:standard:QualifiedDataType:1" schemaLocation="eGIF_QualifiedDataType-
v1-0.xsd"/>
  <!-- ===== -->
  <xs:element name="TaxClearanceDocument" type="rsm:TaxClearanceDocumentStructure" id="EL11010011">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        <ccts:UniqueID>EL11010011</ccts:UniqueID>
        <ccts:Acronym>ACC</ccts:Acronym>
        <ccts:DictionaryEntryName>Tax
Clearance_Document.Details</ccts:DictionaryEntryName>
        <ccts:Version>1</ccts:Version>
        <ccts:Definition>The tax clearance document represents the certificate that tax
authorities handle in tax payers in order for them to prove that they do not owe money to the public state</ccts:Definition>
        <ccts:ObjectClassTermQualifier>Tax Clearance</ccts:ObjectClassTermQualifier>
        <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
        <ccts:BusinessProcessContextValue>In All
Contexts</ccts:BusinessProcessContextValue>
        <ccts:OrganizationContextValue>Taxation Authority</ccts:OrganizationContextValue>
        <ccts:RegionContextValue>In All Contexts</ccts:RegionContextValue>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="TaxClearanceDocumentStructure">
    <xs:sequence>
      <xs:element name="ProtocolID" type="udt:IDType" id="EL11010012" minOccurs="0"
maxOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            <ccts:UniqueID>EL11010012</ccts:UniqueID>
            <ccts:Acronym>BBIE</ccts:Acronym>
            <ccts:DictionaryEntryName>Tax Clearance_Document.
Identification. Identifier</ccts:DictionaryEntryName>
            <ccts:Version>1</ccts:Version>
            <ccts:Definition>The protocol identifier for the
document</ccts:Definition>
            <ccts:ObjectClassTermQualifier>Tax
Clearance</ccts:ObjectClassTermQualifier>
            <ccts:ObjectClassTerm>Document</ccts:ObjectClassTerm>
            <ccts:PropertyTermQualifier>Protocol
</ccts:PropertyTermQualifier>
            <ccts:PropertyTerm>Identification</ccts:PropertyTerm>
            <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
            <ccts:Cardinality>1..1</ccts:Cardinality>
            <ccts:BusinessProcessContextValue>In All
Contexts</ccts:BusinessProcessContextValue>
            <ccts:OrganizationContextValue>Taxation
Authority</ccts:OrganizationContextValue>
            <ccts:RegionContextValue>In All
Contexts</ccts:RegionContextValue>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Name" type="qdt:SimpleName30Type" id="EL11010013" minOccurs="0"
maxOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            <ccts:UniqueID>EL11010013</ccts:UniqueID>
            <ccts:Acronym>BBIE</ccts:Acronym>

```

Name</cts:DictionaryEntryName>	<cts:DictionaryEntryName>Tax Clearance_Document. Name.
certificate</cts:Definition>	<cts:Version>1</cts:Version> <cts:Definition>A name, expressed as text, for this specific
Clearance</cts:ObjectClassTermQualifier>	<cts:ObjectClassTermQualifier>Tax <cts:ObjectClassTerm>Document</cts:ObjectClassTerm> <cts:PropertyTerm>Name</cts:PropertyTerm> <cts:RepresentationTerm>Name</cts:RepresentationTerm> <cts:Cardinality>0..1</cts:Cardinality> <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>	<cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>	<cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>	
	</xs:documentation>
	</xs:annotation>
	</xs:element>
maxOccurs="0">	<xs:element name="IssueDateTime" type="udt:DateTimeType" id="EL11010014" minOccurs="0"
	</xs:annotation>
	<xs:documentation xml:lang="en">
	<cts:UniqueID>EL11010014</cts:UniqueID>
	<cts:Acronym>BBIE</cts:Acronym>
	<cts:DictionaryEntryName>Tax Clearance_Document. Issue.
Date Time</cts:DictionaryEntryName>	<cts:Version>1</cts:Version>
for the issuance of this document</cts:Definition>	<cts:Definition>The date, time, date time or other date time value
Clearance</cts:ObjectClassTermQualifier>	<cts:ObjectClassTermQualifier>Tax <cts:ObjectClassTerm>Document</cts:ObjectClassTerm> <cts:PropertyTerm>Issue</cts:PropertyTerm> <cts:RepresentationTerm>DateTime</cts:RepresentationTerm> <cts:Cardinality>1..1</cts:Cardinality> <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>	<cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>	<cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>	
	</xs:documentation>
	</xs:annotation>
	</xs:element>
maxOccurs="0">	<xs:element name="Purpose" type="qdt:SimpleText50Type" id="EL11010015" minOccurs="0"
	</xs:annotation>
	<xs:documentation xml:lang="en">
	<cts:UniqueID>EL11010015</cts:UniqueID>
	<cts:Acronym>BBIE</cts:Acronym>
	<cts:DictionaryEntryName>Tax Clearance_Document. Issue.
Purpose. Text</cts:DictionaryEntryName>	<cts:Version>1</cts:Version>
document</cts:Definition>	<cts:Definition>The purpose, expressed in text, of this
Clearance</cts:ObjectClassTermQualifier>	<cts:ObjectClassTermQualifier>Tax <cts:ObjectClassTerm>Document</cts:ObjectClassTerm> <cts:PropertyTerm>Purpose</cts:PropertyTerm> <cts:RepresentationTerm>Text</cts:RepresentationTerm> <cts:Cardinality>0..unbounded</cts:Cardinality> <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>	<cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>	<cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>	
	</xs:documentation>
	</xs:annotation>
	</xs:element>
	<xs:element name="IssuerPublicParty" type="ram:PublicPartyStructure" id="EL11010016">
	</xs:annotation>
	<xs:documentation xml:lang="en">
	<cts:UniqueID>EL11010016</cts:UniqueID>
	<cts:Acronym>ASBIE</cts:Acronym>

```

Clearance_Document.Issuer.Public_Party</cts:DictionaryEntryName>
    <cts:DictionaryEntryName>Tax
    <cts:Version>1.0</cts:Version>
    <cts:Definition>The issuer party of the
document</cts:Definition>
    <cts:ObjectClassTermQualifier>Tax
Clearance</cts:ObjectClassTermQualifier>
    <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
    <cts:PropertyTerm>Issuer</cts:PropertyTerm>
    <cts:AssociatedObjectClassTermQualifier>Public</cts:AssociatedObjectClassTermQualifier>
    <cts:AssociatedObjectClassTerm>Party</cts:AssociatedObjectClassTerm>
    <cts:Cardinality>1..1</cts:Cardinality>
    <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
    <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
    <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
    </xs:documentation>
    </xs:annotation>
    </xs:element>
    <xs:element name="AddresseeTaxPayerParty" type="ram:TaxPayerPartyStructure"
id="EL11010017">
    </xs:annotation>
    <xs:documentation xml:lang="en">
    <cts:UniqueID>EL11010017</cts:UniqueID>
    <cts:Acronym>ASBIE</cts:Acronym>
    <cts:DictionaryEntryName>Tax
Clearance_Document.Adressee.Tax Payer_Party</cts:DictionaryEntryName>
    <cts:Version>1.0</cts:Version>
    <cts:Definition>The tax payer (person or business) to which this
document addresses</cts:Definition>
    <cts:ObjectClassTermQualifier>Tax
Clearance</cts:ObjectClassTermQualifier>
    <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
    <cts:PropertyTerm>Addressee</cts:PropertyTerm>
    <cts:AssociatedObjectClassTermQualifier>Tax
Payer</cts:AssociatedObjectClassTermQualifier>
    <cts:AssociatedObjectClassTerm>Party</cts:AssociatedObjectClassTerm>
    <cts:Cardinality>1..1</cts:Cardinality>
    <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
    <cts:OrganizationContextValue>Taxation
Authority</cts:OrganizationContextValue>
    <cts:RegionContextValue>In All
Contexts</cts:RegionContextValue>
    </xs:documentation>
    </xs:annotation>
    </xs:element>
    <xs:element name="EffectiveSimplifiedPeriod" type="ram:SimplifiedPeriodStructure"
id="EL11010018">
    </xs:annotation>
    <xs:documentation xml:lang="en">
    <cts:UniqueID>EL11010018</cts:UniqueID>
    <cts:Acronym>ASBIE</cts:Acronym>
    <cts:DictionaryEntryName>Tax
Clearance_Document.Effective.Simplified_Party</cts:DictionaryEntryName>
    <cts:Version>1.0</cts:Version>
    <cts:Definition>The period during which the tax clearance
document is effective</cts:Definition>
    <cts:ObjectClassTermQualifier>Tax
Clearance</cts:ObjectClassTermQualifier>
    <cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
    <cts:PropertyTerm>Effective</cts:PropertyTerm>
    <cts:AssociatedObjectClassTermQualifier>Simplified</cts:AssociatedObjectClassTermQualifier>
    <cts:AssociatedObjectClassTerm>Period</cts:AssociatedObjectClassTerm>
    <cts:Cardinality>1..1</cts:Cardinality>
    <cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>

```

```

Authority</cts:OrganizationContextValue>
Contexts</cts:RegionContextValue>
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="RelatedSimplifiedUsageCondition"
type="ram:SimplifiedUsageConditionStructure" id="EL11010019">
</xs:annotation>
<xs:documentation xml:lang="en">
<cts:UniqueID>EL11010019</cts:UniqueID>
<cts:Acronym>ASBIE</cts:Acronym>
<cts:DictionaryEntryName>Tax
Clearance_Document.Related.Simplified_Party</cts:DictionaryEntryName>
<cts:Version>1.0</cts:Version>
<cts:Definition>The usage condition for the tax clearance
document</cts:Definition>
Clearance</cts:ObjectClassTermQualifier>
<cts:ObjectClassTerm>Document</cts:ObjectClassTerm>
<cts:PropertyTerm>Related</cts:PropertyTerm>
<cts:AssociatedObjectClassTermQualifier>Simplified</cts:AssociatedObjectClassTermQualifier>
<cts:AssociatedObjectClassTerm>Usage
Condition</cts:AssociatedObjectClassTerm>
<cts:Cardinality>1..1</cts:Cardinality>
<cts:BusinessProcessContextValue>In All
Contexts</cts:BusinessProcessContextValue>
Authority</cts:OrganizationContextValue>
Contexts</cts:RegionContextValue>
</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>

```