

---

# Software Evolution Monitor

---

Software Evolution Monitor  
Σχεδίαση  
(Design)

Έκδοση 1.0

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

## Ιστορικό Προηγούμενων Εκδόσεων

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
1/12/2014	1.0	1 <sup>η</sup> Έκδοση Σχεδίου	Φωτιάδης Ευστάθιος

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

## Περιεχόμενα

1.	Εισαγωγή	4
1.1	Σκοπός	4
1.2	Εύρος	4
1.3	Αναφορές	4
1.4	Σύνοψη	4
2.	Αρχιτεκτονική Αναπαράσταση	4
2.1	Διάγραμμα πακέτων	5
2.2	Διάγραμμα κλάσεων	5
2.3	Διαγράμματα ακολουθίας	5
3.	Αρχιτεκτονικοί Στόχοι και Περιορισμοί	5
4.	Αρχιτεκτονική Άποψη	5
4.1	Σύνοψη	6
4.2	Υποσυστήματα	6
5.	Τεχνική Άποψη	7
5.1	Σχέδιο Υποσυστημάτων	7
5.1.1	Υποσύστημα SEMonitorViews	7
5.1.2	Υποσύστημα SEMonitorModels	8
5.1.3	Υποσύστημα SEMonitorPresenter	9
5.1.4	Υποσύστημα SEMonitorIO	10
5.2	Περιπτώσεις Χρήσης	10
6.	Ποιότητα	13

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

# Σχεδίαση (Design)

## 1. Εισαγωγή

Το παρόν κείμενο περιλαμβάνει την αρχιτεκτονική και τεχνική σχεδίαση για την κατασκευή ενός εργαλείου παρακολούθησης της εξέλιξης συστημάτων λογισμικού. Συνδέεται άμεσα με το κείμενο Καθορισμού Απαιτήσεων, οι περιπτώσεις χρήσεις αναλύονται εδώ τεχνικά λαμβάνοντας υπ' όψη τις λειτουργικές και μη απαιτήσεις.

### 1.1 Σκοπός

Ο σκοπός του κειμένου είναι να παρουσιάσει με δομημένο τρόπο την σχεδίαση των υποσυστημάτων του εργαλείου υπό ανάπτυξη καθώς και τις μεταξύ τους σχέσεις από στατική και δυναμική άποψη.

Η περιγραφή γίνεται με χρήση εργαλείων αναπαράστασης όπως η UML, έτσι ώστε ο προγραμματιστής να μπορεί να αντιληφθεί τις εξαρτήσεις και το πως συσχετίζονται τα επιμέρους μέρη.

### 1.2 Εύρος

Η οργάνωση της κατασκευής, η υλοποίηση και ο τελικός έλεγχος του εργαλείου παρακολούθησης της εξέλιξης συστημάτων λογισμικού θα γίνει με βάση το παρόν κείμενο.

### 1.3 Αναφορές

Για την σχεδίαση αντλήσαμε υλικό από τις παρακάτω πηγές:

[1] Software Evolution Monitor Requirements Specification v1.0

[2] A.Zarras “Τεχνολογία Λογισμικού” <http://cs.uoi.gr/~zarras/se.htm>

[3] I. Sommeriville. “Εισαγωγή στην Τεχνολογία Λογισμικού”, Κλειδάριθμος

### 1.4 Σύνοψη

Το υπόλοιπο κείμενο χωρίζεται σε πέντε ενότητες. Αρχικά παρουσιάζεται η αρχιτεκτονική αναπαράσταση του συστήματος από αρχιτεκτονική και τεχνική άποψη. Μετέπειτα αναλύονται οι αρχιτεκτονικοί στόχοι και περιορισμοί που λαμβάνονται υπόψη στον σχεδιασμό. Στις επόμενες δύο ενότητες αναλύονται διεξοδικά η αρχιτεκτονική και η τεχνική άποψη αντίστοιχα. Τέλος ακολουθεί μια περιγραφή της ποιότητας του συστήματος με βάση μετρικές ποιότητας.

## 2. Αρχιτεκτονική Αναπαράσταση

Η σχεδίαση περιλαμβάνει το αρχιτεκτονικό και το τεχνικό σχέδιο. Για την παρουσίαση του αρχιτεκτονικού σχεδίου χρησιμοποιείται διάγραμμα πακέτων. Το τεχνικό σχέδιο παρουσιάζεται από στατική άποψη με

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

διάγραμματα κλάσεων και από δυναμική άποψη από διαγράμματα ακολουθίας.

## 2.1 Διάγραμμα πακέτων

Στο διάγραμμα πακέτων αναπαριστά τα υποσυστήματα του εργαλείου και τις μεταξύ τους σχέσεις με οπτική υψηλού επιπέδου (high-level). Τα υποσυστήματα είναι μια μονάδα του συστήματος που ομαδοποιεί συνεκτικά επιμέρους στοιχεία του συστήματος που υλοποιούν ένα συγκεκριμένο στόχο.

## 2.2 Διάγραμμα κλάσεων

Στο διάγραμμα κλάσεων οποίο απεικονίζεται η δομή των επιμέρους κλάσεων κάθε υποσυστήματος. Επίσης απεικονίζονται και μεταξύ τους σχέσεις όπως κληρονομικότητα, εξάρτηση, συσχέτιση συνάθροιση, σύνθεση κλπ.

## 2.3 Διαγράμματα ακολουθίας

Από δυναμικής άποψης χρησιμοποιούνται διαγράμματα ακολουθίας που παρουσιάζουν την επικοινωνία μεταξύ των κλάσεων του κώδικα για κάθε περίπτωση χρήσης. Στο οριζόντιο άξονα εμφανίζονται οι κλήσεις και τα μηνύματα μεταξύ των κλάσεων ενώ ο κατακόρυφος άξονας απεικονίζει τη χρονική εξέλιξη της εκτέλεσης του κώδικα.

## 3. Αρχιτεκτονικοί Στόχοι και Περιορισμοί

Με βάση το κείμενο Καθορισμού Απαιτήσεων απαιτείται υλοποίηση με βάση το αντικειμενοστραφές στυλ προγραμματισμού βασισμένο στη γλώσσα προγραμματισμού Java.

Κατά το σχεδιασμό του συστήματος λήφθηκαν υπόψιν και άλλες απαιτήσεις όπως η ύπαρξη γραφικής διεπαφής, η επεκτασιμότητα και η συντήρηση του συστήματος.

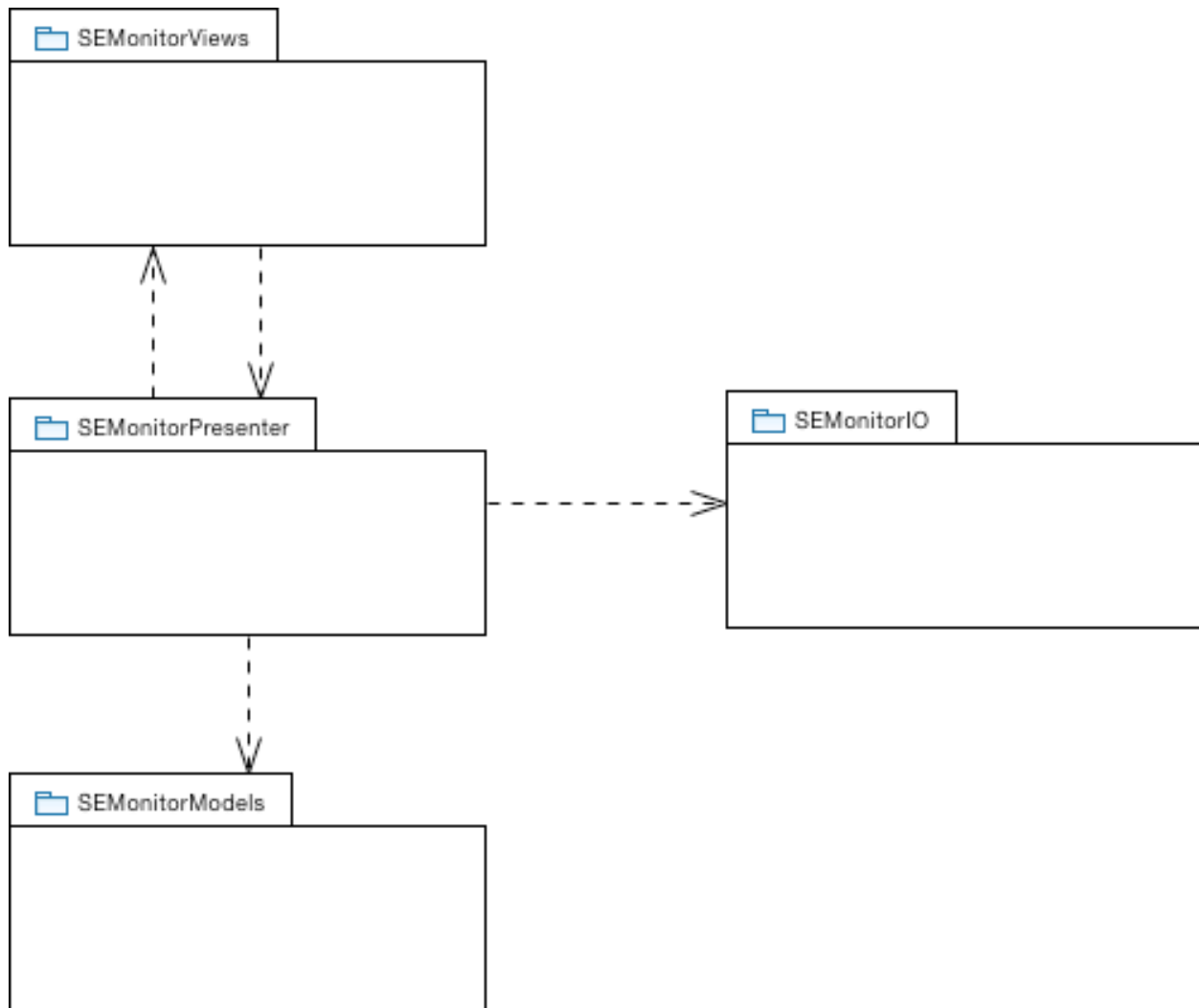
Ο σχεδιασμός έγινε μέσα στα πλαίσια του αντικειμενοστραφούς στυλ έτσι ώστε να ικανοποιούνται οι παραπάνω απαιτήσεις. Γι' αυτό έγινε χρήση τεχνικών όπως σύνθεση και η αφαίρεση, που διευκολύνουν την επέκταση και επαναχρησιμοποίηση του κώδικα, μειώνοντας ως εκ τούτου τον χρόνο υλοποίησης και συντήρησης.

## 4. Αρχιτεκτονική Άποψη

Στην ενότητα αυτή μελετάται το σύστημα από αρχιτεκτονικής άποψη. Παρουσιάζονται το διάγραμμα πακέτων καθώς και η περιγραφή του κάθε πακέτου.

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

#### 4.1 Σύνοψη



#### 4.2 Υποσυστήματα

Το εργαλείο αποτελείται από τέσσερα υποσυστήματα. Η αρχιτεκτονική σχεδίαση είναι βασισμένη στο πρότυπο Model-Presenter-View. Η απόφαση αυτή πάρθηκε καθότι θέλαμε ένα ευέλικτο εργαλείο με γραφική διεπαφή χρήστη. Το συγκεκριμένο πρότυπο επιτρέπει τον διαχωρισμό μεταξύ μοντέλων και διεπαφής μειώνοντας έτσι την σύζευξη μεταξύ των υποσυστημάτων. Τον ρόλο του ενδιάμεσου αναλαμβάνει ο Presenter.

Το πρώτο υποσύστημα είναι το SEMonitorModels που περιλαμβάνει τα μοντέλα των οντοτήτων του συστήματος. Οι κλάσεις του υποσυστήματος χρησιμοποιούνται για την διατήρηση στην μνήμη και την επεξεργασία των δεδομένων που προέρχονται από το αρχείο εισόδου. Μοντελοποιούμε πέντε οντότητες:

- Ιστορικό εξέλιξης
- Εγγραφή απότιμησης των νόμων
- Πληροφορίες έκδοσης
- Εγγραφή Μετρικών
- Εγγραφή Αλλαγών

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

Το δεύτερο υποσύστημα είναι το SEMonitorViews που περιλαμβάνει τις γραφικές διεπαφές χρήστη όπως αυτές ορίστηκαν στο έγγραφο Καθορισμού Απαιτήσεων. Συγκεκριμένα περιλαμβάνει μια κύρια διεπαφή και μία διεπαφή ανάλυσης και αποτίμησης των νόμων του Lehm.

Το τρίτο υποσύστημα είναι το SEMonitorPresenter και είναι υπεύθυνο για δύο πράγματα:

- Τον έλεγχο της ροής της πληροφορίας μεταξύ των μοντέλων
- Την εκτέλεση των εντολών του χρήστη που προέρχονται από την διάδραση με την διεπαφή

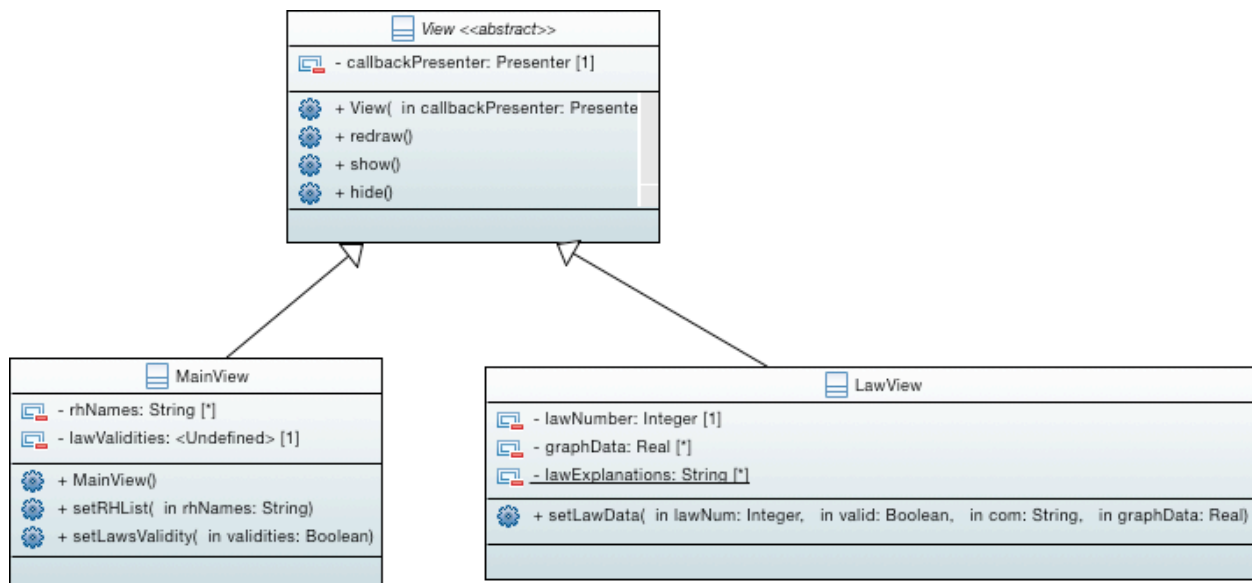
Το τέταρτο υποσύστημα είναι βοηθητικό και περιέχει στατικές κλάσεις για το άνοιγμα και την επεξεργασία του αρχείου δεδομένων καθώς και για την αποθήκευση της αναφοράς. Είναι ένα πακέτο με λογική συνεκτικότητα.

## 5. Τεχνική Αποψη

Στην ενότητα αυτή μελετάται ο τεχνικός σχεδιασμός του συστήματος. Παρουσιάζονται τα διαγράμματα κλάσεων για κάθε υποσύστημα καθώς και διαγράμματα ακολουθίας για κάθε περίπτωση χρήσης.

### 5.1 Σχέδιο Υποσυστημάτων

#### 5.1.1 Υποσύστημα SEMonitorViews



Στο υποσύστημα SEMonitorViews βρίσκονται οι κλάσεις που είναι υπεύθυνες για την γραφική διεπαφή χρήστη.

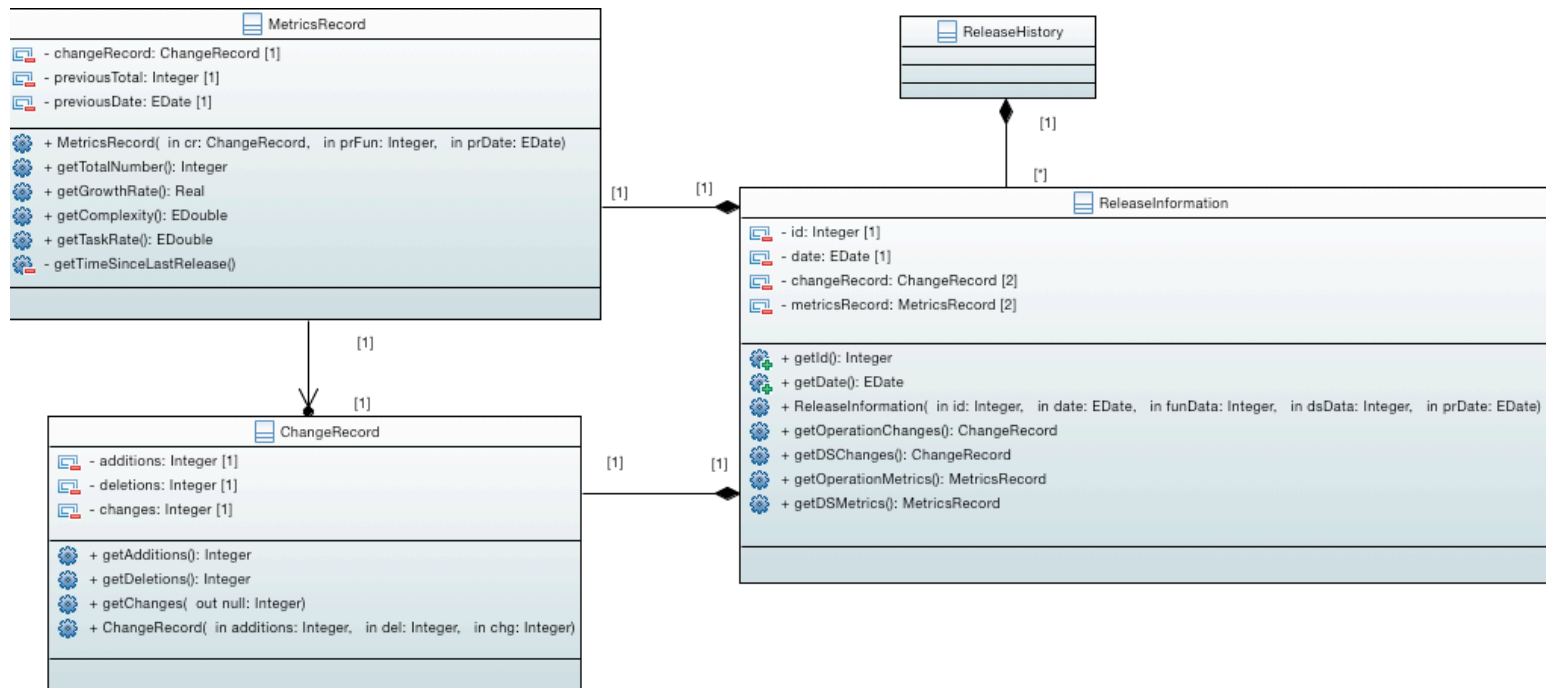
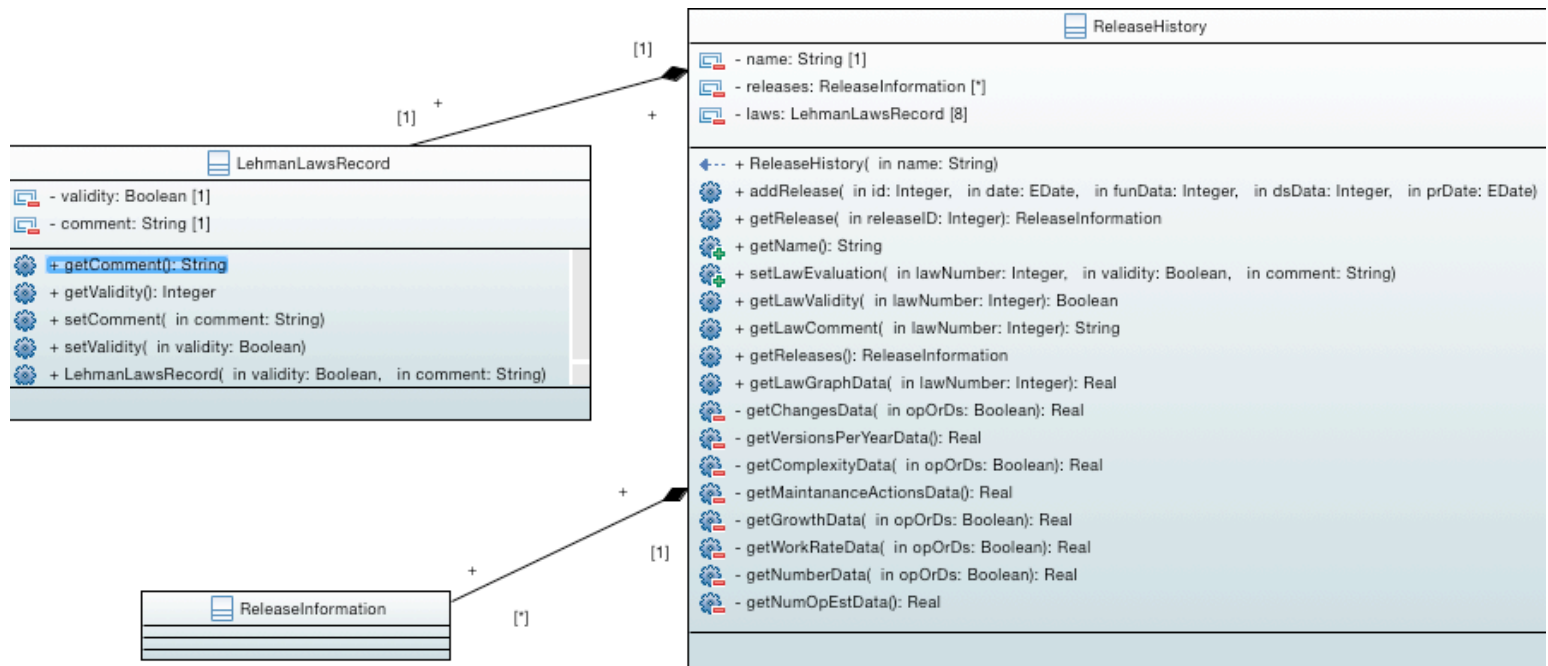
Η υλοποίηση γίνεται με τη βοήθεια abstract κλάσης στην οποία περιγράφονται οι γενικές μέθοδοι εμφάνισης, απόκρυψης και επανασχεδιασμού (redraw). Επίσης κατά την δημιουργία ενός αντικειμένου γίνεται inject ο Presenter, που παραμένει ως μεταβλητή μέλος, έτσι ώστε να καλείται μετά από κάθε ενέργεια του χρήστη.

Δημιουργούνται παράγωγες κλάσεις για κάθε όψη που προβλέπεται στο έγγραφο καθορισμού απαιτήσεων. Για την κύρια όψη μας ενδιαφέρει να μπορούμε να της μεταφέρουμε τη λίστα με τα ονόματα των συστημάτων λογισμικού που είναι φορτωμένα στη μνήμη καθώς και την validity των νομων του Lehm για το επιλεγμένο σύστημα λογισμικού. Η όψη αποτίμησης των νόμων χρειάζεται να τροφοδοτείται με τον επιλεγμένο νόμο, την

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

υπάρχουσα αποτίμηση και σχόλιο για αυτόν καθώς και τα δεδομένων για τα γραφήματα (που κανονικά είναι List of Real[] με cardinality 2-3). Τα παραπάνω βρίσκονται σε αντιστοιχία με το πρότυπο διεπαφής χρήστη που παρουσιάστηκε στο έγγραφο καθορισμού απαιτήσεων.

### 5.1.2 Υποσύστημα SEMonitorModels





Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

Στο υποσύστημα SEMonitorModels περιλαμβάνονται οι κλάσσεις που μοντελοποιούν τις οντότητες που αποτελούν τα δεδομένα που παρέχονται από το αρχείο εισόδου. Η συγκεκριμένη μοντελοποίηση εφαρμόστηκε όπως ακριβώς περιγράφεται στο έγγραφο Περιγραφής Απαιτήσεων (βλέπε [2]). Συγκεκριμένα οι κλάσσεις είναι:

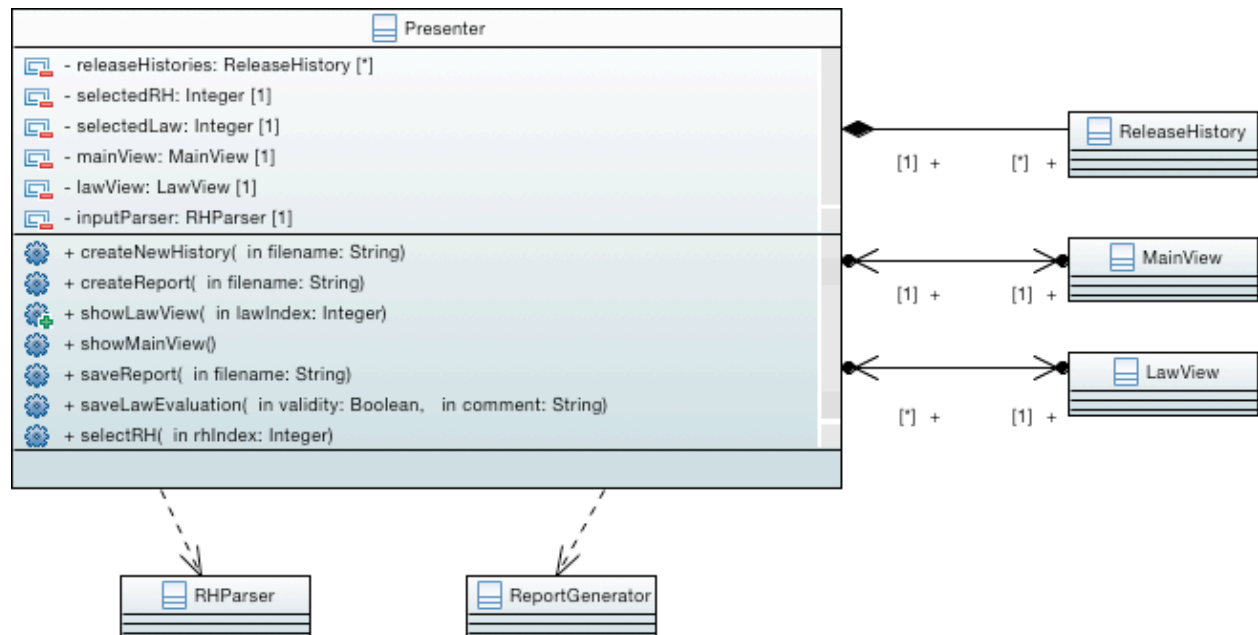
- ReleaseHistory (ιστορικό εξέλιξης ενός συστήματος λογισμικού)
- LehmanLawsRecord (εγγραφή αποτίμησης νόμων του Lehm)
- ReleaseInformation (πληροφορίες μιας έκδοσης του συστήματος λογισμικού)
- ChangeRecord (εγγραφή αλλαγών)
- MetricsRecord (εγγραφή μετρικών)

Οι κλάσσεις απεικονίζονται σε δύο γραφήματα **Models Part 1** και **Models Part 2**.

Από τις παραπάνω κλάσσεις το η ReleaseHistory είναι η μόνη που έχει σχέσεις με άλλα υποσυστήματα.

Συνοπτικά, η ChangeRecord αποτελεί μια δομή δεδομένων για την διατήρηση στη μνήμη μιας εγγραφής αλλαγών που προέρχεται από το αρχείο εισόδου. Η MetricsRecord αποτελεί μια κλάση που χρησιμοποιώντας τα δεδομένα από αντικείμενο της ChangeRecord ώστε παράγει κάποιες μετρικές για την έκδοση του λογισμικού. Το ReleaseInformation χρησιμοποιεί τα αντικείμενα ChangeRecord και MetricsRecord και παρέχει ένα interface για παροχή δεδομένων σε άλλες κλάσεις. Αυτό γίνεται με την ReleaseHistory η οποία διατηρεί τα με δεδομένα των εκδόσεων σε μια λίστα ReleaseInformation. Η ReleaseHistory διατηρεί επίσης μια εγγραφή με την validity και τα σχόλια νόμους του Lehm. Η ReleaseInformation παρέχει όλες τις πληροφορίες προς τα έξω μέσω μεθόδων οπότε ο καταναλωτής δεν χρειάζεται να γνωρίζει το πως λειτουργεί εσωτερικά (encapsulation – information hiding). Τέλος η ReleaseInformation παρέχει τα δεδομένα για την κατασκευή των γραφημάτων των όψεων.

### 5.1.3 Υποσύστημα SEMonitorPresenter

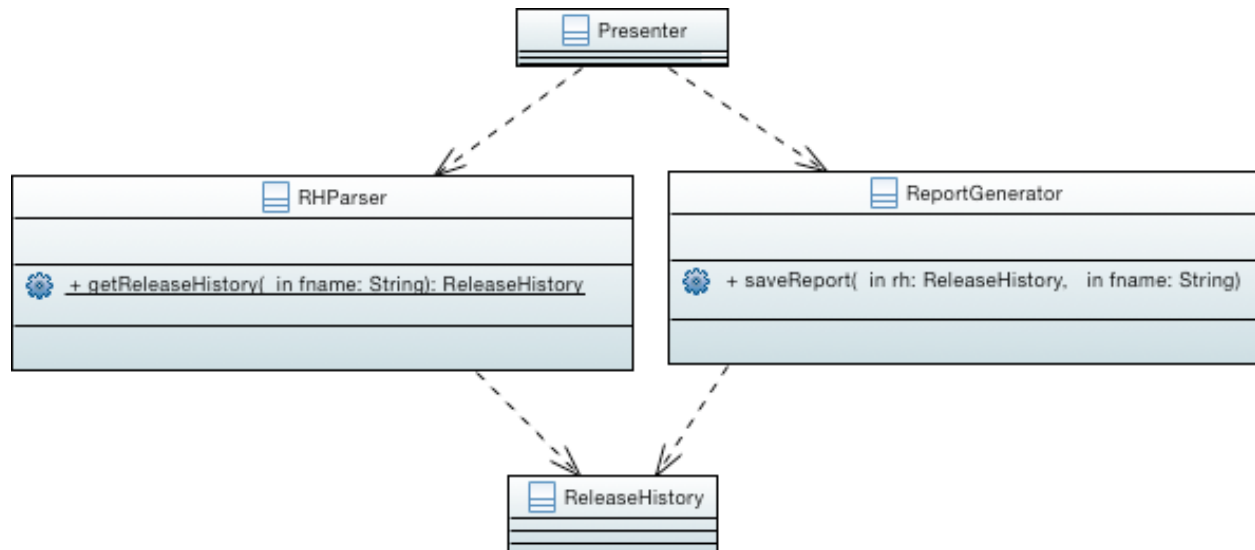


Το υποσύστημα SEMonitorPresenter αποτελείται από μία μόνο κλάση. Η κλάση αυτή αναλαμβάνει το ρόλο του μεσάζοντα (middle-man) μεταξύ της διεπαφής χρήστη και του μοντέλου των δεδομένων. Όπως βλέπουμε από το γράφημα η κλάση αυτή συνδέεται με σύνδεση με τις δύο όψεις και με το μοντέλο του ιστορικού εκδόσεων.

Όταν ο χρήστης προβαίνει σε μια ενέργεια διαμέσου της διεπαφής, το αντικείμενο της όψης καλεί τον Presenter έτσι ώστε να προβεί στις κατάλληλες ενέργειες. Για παράδειγμα το κουμπί «Δημιουργία αναφοράς» εκτελεί την μέθοδο **saveReport()**. Ο Presenter χρησιμοποιεί τις κλάσεις από το πακέτο SEMonitorIO για να διαβάσει από και να γράψει στο σύστημα αρχείων.

Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

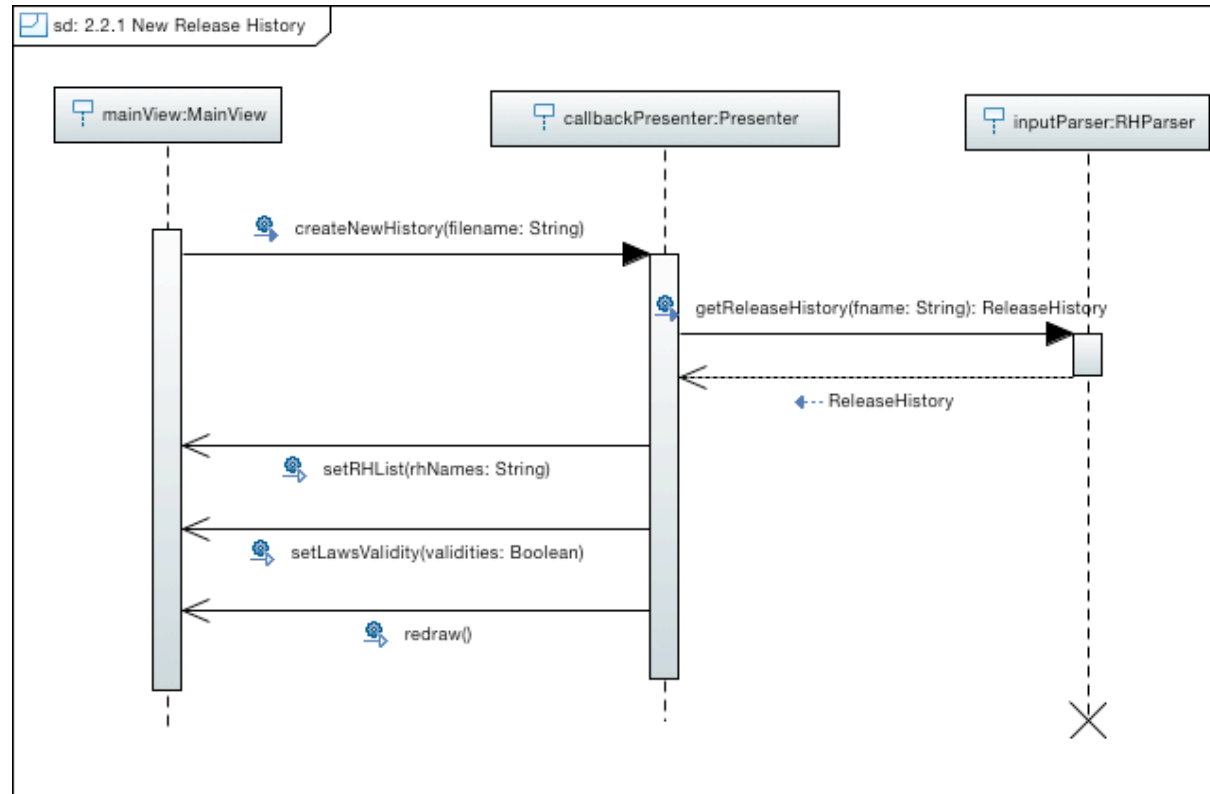
#### 5.1.4 Υποσύστημα SEMonitorIO



Το υποσύστημα SEMonitorIO αποτελείται από δύο στατικές κλάσεις (με την έννοια ότι δεν έχουν constructor μόνο στατικές μεθόδους). Η κλάση RHParseR χρησιμοποιείται για να διαβάσει ένα αρχείο εισόδου από το σύστημα αρχείων και επιστρέφει ένα αντικείμενο τύπου ReleaseHistory. Αντίστοιχα η ReportGenerator χρησιμοποιείται για την αποθήκευση της αναφοράς.

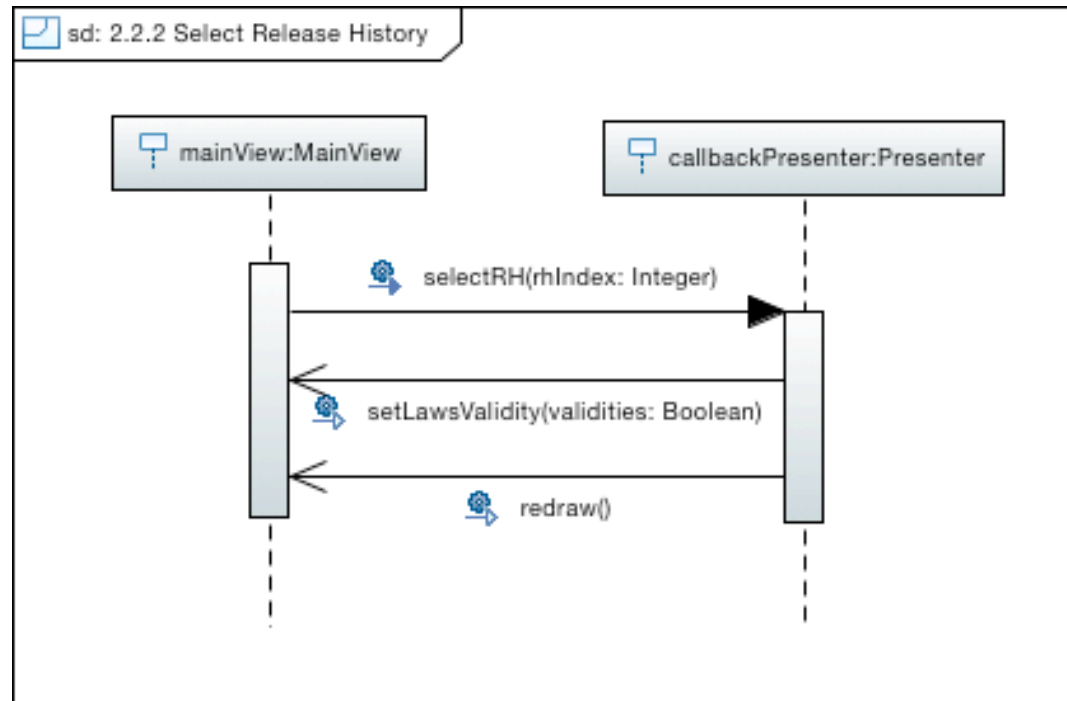
## 5.2 Περιπτώσεις Χρήσης

### 5.2.1 Δημιουργία ιστορικού εξέλιξης ενός συστήματος λογισμικού

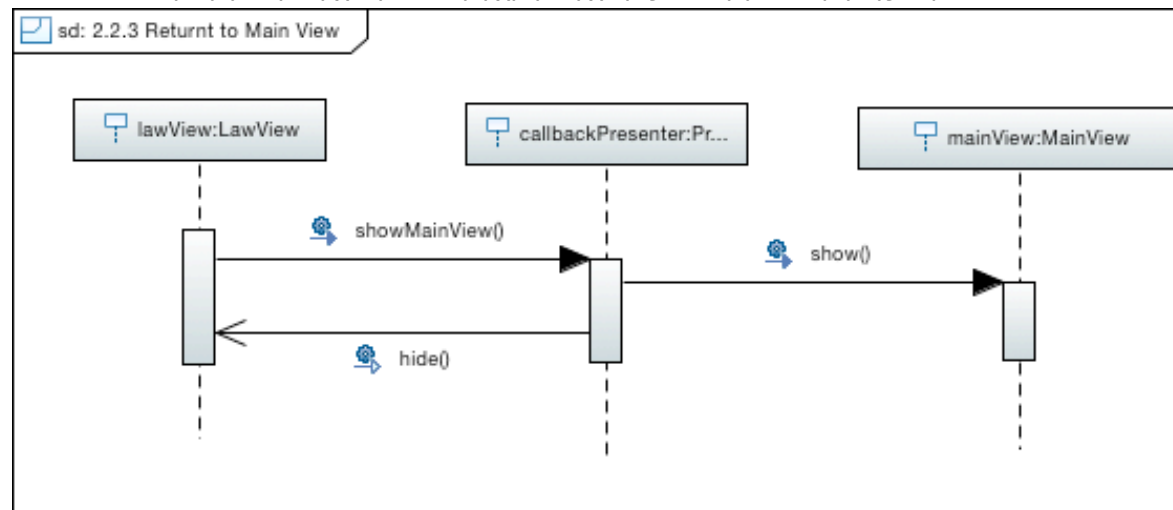


Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

### 5.2.2 Επιλογή ιστορικού εξέλιξης συστήματος λογισμικού

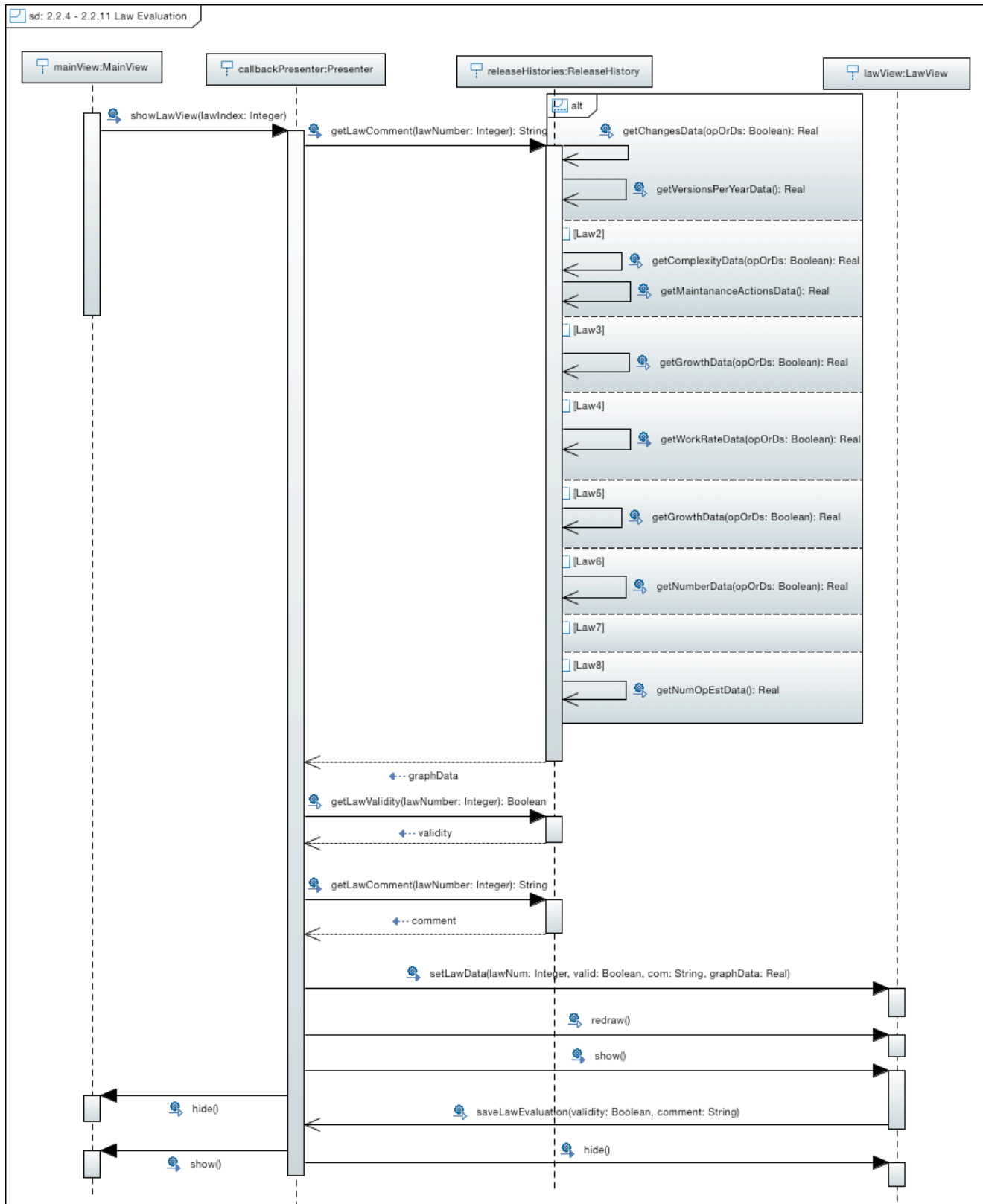


### 5.2.3 Επιστροφή στην αρχική διεπαφή χρήστη χωρίς αλλαγή αποτίμησης νόμου



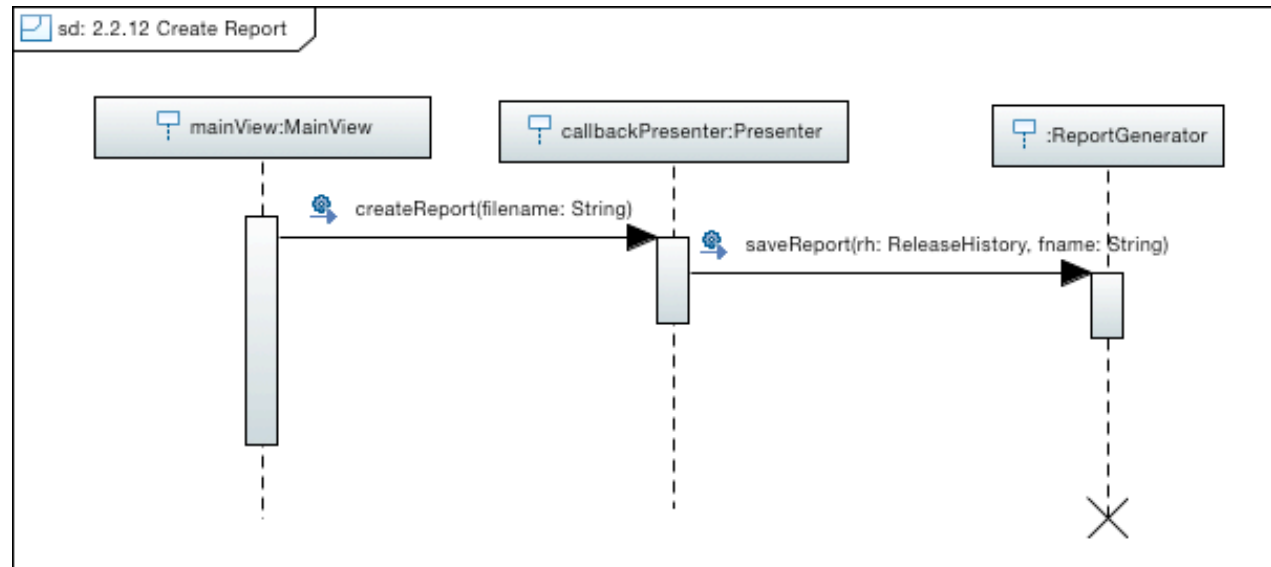
Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

#### 5.2.4 Αποτίμηση – προβολή 1<sup>ου</sup> - 8<sup>ου</sup> Νόμου



Software Evolution Monitor	Version: 1.0
Σχεδίαση	Date: 1/12/2014
Software Evolution Monitor Design	

### 5.2.5 Δημιουργία αναφοράς



## 6. Ποιότητα

Η ποιότητα του σχεδίου δίνεται με τη βοήθεια μετρικών όπως η Coupling Between Object Classes (CBO) και η Response for a Class (RFC). Είναι επιθυμητές οι μικρότερες δυνατές τιμές και για τις δύο μετρικές γιατί υποδεικνύουν χαμηλή σύζευξη μεταξύ των μερών του συστήματος, ευκολία στον έλεγχο και την κατανόηση.

Όνομα Κλάσης	CBO	RFC
Presenter	5	$25 = 6 + 5(\text{MainView}) + 4(\text{LawView}) + 8(\text{ReleaseHistory}) + 1(\text{RHParser}) + 1(\text{ReportGenerator})$
View	1	3
MainView	2	$10 = 5 + 5(\text{Presenter})$
LawView	2	$6 = 4 + 2(\text{Presenter})$
ReleaseHistory	2	$34 = (8 \text{ public} + 8 \text{ private}) + 4(\text{LehmanLawsRecord}) + 6(\text{ReleaseInformation}) + 5(\text{MetricsRecord}) + 3(\text{ChangesRecord})$
LehmanLawsRecord	0	4
ReleaseInformation	2	6
MetricsRecord	1	$9 = (5 \text{ public} + 1 \text{ private}) + 3(\text{ChangesRecord})$
ChangesRecord	0	3
RHParser	1	1
ReportGenerator	1	1