

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ**

**ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

**Σαββάκη Κρυσταλλία  
Μεταπτυχιακή Φοιτήτρια**

**Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης  
Επόπτης Μεταπτυχιακής Εργασίας: Καθηγητής, Α. Σαββίδης**

**Παρασκευή, 23 Απριλίου 2021 , ώρα 15:00 μ.μ.**

**Join Zoom Meeting**

<https://zoom.us/j/98228314371>

**“Εξαγωγή της Αρχιτεκτονικής Λογισμικού από τον Πηγαίο Κώδικα”**

### **Περίληψη**

Η αρχιτεκτονική λογισμικού έχει βασικό ρόλο στον προσδιορισμό των προδιαγραφών και το σχεδιασμό της δομής ενός συστήματος, αποτελώντας τον συνδετικό κρίκο μεταξύ των απαιτήσεων και της υλοποίησης του. Παρόλα αυτά, κατά την διαδικασία ανάπτυξης λογισμικού, υπάρχει πιθανότητα να παραβιαστεί η αρχιτεκτονική του εκάστοτε συστήματος. Ως εκ τούτου, ενδέχεται να προκύψουν απροσδόκητες ή ανεπιθύμητες εξαρτήσεις μεταξύ των τμημάτων στον πηγαίο κώδικα. Επιπλέον, η κατανόηση της βασικής αρχιτεκτονικής σε συστήματα μεγάλης κλίμακας είναι απαραίτητη για την αποτελεσματική συντήρηση και συνεχή βελτίωση τους. Ωστόσο, είναι αδύνατο να αντικατοπτριστεί άμεσα στον πηγαίο κώδικα, καθώς δεν αποτελεί δομικό στοιχείο των γλωσσών προγραμματισμού. Επομένως, είναι αναγκαία η ανάπτυξης μεθόδων και εργαλείων που βοηθούν στην κατανόηση της δομής ενός συστήματος, εξάγοντας πληροφορίες σχετικά με την αρχιτεκτονική του απευθείας από τον πηγαίο κώδικα.

Η αντίστροφη μηχανική είναι η διαδικασία ανάλυσης ενός συστήματος με σκοπό τον εντοπισμό και την αναπαράσταση των σχέσεων μεταξύ των βασικών του στοιχείων. Η εξαγωγή της αρχιτεκτονικής αποτελεί υποσύνολο της αντίστροφης μηχανικής, η οποία στοχεύει στην ανίχνευση αφαιρέσεων που αντανακλούν τα βασικά στοιχεία του συστήματος σε υψηλό επίπεδο. Στο πλαίσιο αυτό, μέθοδοι αντίστροφης μηχανικής μπορούν να χρησιμοποιηθούν στην διαδικασία υπολογισμού της πραγματικής αρχιτεκτονική όπως αυτή αντικατοπτρίζεται, ώστε να συγκριθεί μετέπειτα με την αρχικά σχεδιασμένη αρχιτεκτονική.

Στην εργασία αυτή, προτείνουμε ένα εργαλείο το οποίο στοχεύει στην ανακατασκευή της αρχιτεκτονικής ενός συστήματος όπου ο πηγαίος κώδικας έχει αναπτυχθεί σε C++. Για να το πετύχουμε αυτό, εστιάζουμε στις σχέσεις μεταξύ των κλάσεων, καθώς αποτελούν τα πιο βασικά στοιχεία σε επίπεδο πηγαίου κώδικα. Πιο συγκεκριμένα, το εργαλείο που έχουμε αναπτύξει πραγματοποιεί στατική ανάλυση σε C++ εφαρμογές, εξάγοντας δεδομένα σχετικά με τις κλάσεις και τις μεταξύ τους σχέσεις. Έπειτα, η προσέγγισή μας δημιουργεί και οπτικοποιεί ένα γράφο εξαρτήσεων που αντιπροσωπεύει όλες τις σχέσεις μεταξύ των κλάσεων. Στην συνέχεια, στον εξαγόμενο γράφο εφαρμόζουμε αλγορίθμους ομαδοποίησης, ώστε να δημιουργήσουμε μία αφαιρετική εικόνα του συστήματος εντοπίζοντας τα αρχιτεκτονικά του στοιχεία. Συγκεκριμένα χρησιμοποιήσαμε αλγορίθμους ομαδοποίησης, όπως τους *Louvain*, *Infomap* και *Layered Label Propagation*, ενώ δίνεται η δυνατότητα στους χρήστες να επιλέξουν ομαδοποίηση βάση συγκεκριμένων χαρακτηριστικών μέσω χώρων ονομάτων και φακέλων. Τέλος, έχουμε αναπτύξει αρκετά σενάρια χρήσης ώστε να ελέγξουμε την εγκυρότητα και την αποτελεσματικότητα της προσέγγισής μας.

**University of Crete**

**Computer Science Department**

**M.Sc. Thesis presentation / examination**

**Kristallia Savaki**

**Master's Thesis Supervisor: Professor, A. Savidis**

**Friday, 23 April 2021, 15:00 p.m.**

**Join Zoom Meeting**

<https://zoom.us/j/98228314371>

## “Software Architecture Mining From Source Code”

### Abstract

Software architecture plays a primary role in system specifications and design, acting as a bridge between requirements and implementation. As software systems evolve over time, conformance to the initial architecture may be broken, while unexpected or undesirable component dependencies may arise due to the way the source code evolves. Additionally, understanding the underlying architecture of large applications is required for effective maintenance and continuous improvement. However, the problem is that the architecture is not somehow reflected in the source code since it is not a programming language construct. As a consequence, methods and tools to extract architecture-related information from source code that can aid software developers in understanding system structure are required.

Reverse engineering is the process of analyzing a system in order to identify and represent the relationships between its components. Architecture mining is a subset of reverse engineering in which meaningful high-level abstractions that represent system components are detected. In this context, reverse engineering methods may be used to compute concrete architectures and compare it to the original conceptual architecture.

In this thesis, we propose an architecture mining tool with the aim of reconstructing the software architecture only from C++ source code. In order to accomplish this, we focus on class relationships, since they represent key elements at the source level. Specifically, our system parses C++ projects and statically analyzes the source code to extract class-related information. Then, it generates and visualizes a dependency graph that represents all the relationships between classes. On top of this graph, we apply clustering methods to identify high-level architectural entities. We use clustering algorithms like *Louvain*, *Infomap* and *Layered Label Propagation*, but also allow users to choose ad-hoc clustering via namespaces and folders. Finally, we have carried out a few case studies to assess and validate the utility of our architecture mining approach.