ΠΑΡΟΥΣΙΑΣΗ
ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

ΗΜΕΡΟΜΗΝΙΑ: Τρίτη, 19 Μαρτίου 2019
ΩΡΑ: 12:00
ΑΙΘΟΥΣΑ: Αίθουσα Σεμιναρίων
Κτήριο Τμήματος Μηχανικών Η/Υ & Πληροφορικής
ΟΜΙΛΗΤΗΣ: Πέτρος Μανούσης

Θέμα

«Automated Representation, Quality Assessment, Evolution, Visualization, and Adaptation to Change for Data Intensive Ecosystems»

Επταμελής Εξεταστική Επιτροπή:

1. Panos Vassiliadis, Assoc. Professor, Department of Computer Science And Engineering, University of Ioannina (advisor)
2. Evaggelia Pitoura, Professor, Department of Computer Science And Engineering, University of Ioannina
3. Apostolos Zarras, Assoc. Professor, Department of Computer Science And Engineering, University of Ioannina
4. Nikos Mamoulis, Assoc. Professor, Department of Computer Science And Engineering, University of Ioannina
5. Diomidis Spinellis, Professor, Department of Management Science and Technology, Athens University of Economics and Business
6. Spiros Skiadopoulos, Professor, Department of Informatics and Telecommunications, University of Peloponnese
7. Alkis Simitsis, Research Director, Research Center "Athena" -- Research & Innovation Information Technologies
Software evolution is the most demanding part of software development, since the 60% of the resources (time, money, etc.) of a company is consumed in order to evolve its software so as to meet the new requirements of its users. Database-related software also needs to evolve, but since a database is a software part that many other parts rely on it, the difficulty of evolving a database-related project increases because changing a small part of the database schema could result in failures in many different parts of the application’s code. Additionally, the problem becomes even more difficult due to the scarcity of tools that help the developers evolve the database in sync with the software. In this research, we introduce principles, constructs, algorithms and metrics that help the database administrators and the software developers write easier to comprehend, adapt and maintain database related code. To achieve all that, we introduce a mapping between the software code of the projects that are related to a database, with the database schema. Initially, we locate the database queries in the software. Since the queries are embedded in a hosting language (PHP, C++, Java, etc.), we produce every possible variant of a query, based on the branches and loops of the hosting language providing abstract representations for each of the queries. Then, we come up with a language-independent representation of the queries and based on it, we translate the queries to concrete ones in more than one query languages (e.g. SQL, MongoDB). Second, to describe how well constructed a data-intensive ecosystem is, we propose a metric that describes the quality of the written query in the code of the project regarding the database schema of the project. Our experiments reveal that our metric is in sync with the Lehman’s low of evolution. Whenever it is possible to propose a better way to use the database schema, we propose the introduction of views along with the rewriting of queries over them, so as to achieve better metric values and less developer effort for future schema evolution maintenance. Third, we employ a rigorous formal modelling of host code queries, tables and views in the form of a graph, called Architecture Graph, to facilitate diverse tasks related to the management of a Data Intensive Ecosystems. The first task facilitated by Architecture Graph is the identification of the impact that a schema change can have. We introduce algorithms to (a) identify and (b) adapt (via the appropriate rewritings) the impacted queries. We employ a regulation of the propagation of changes in the Architecture Graph via appropriate policies. Finally, the Architecture Graph can be exploited for visualizing the Data Intensive Ecosystems. We explore alternative ways of visualization, and we extend them via proposing a “what-if” visualization method, and we conduct a user study for all the visualizing algorithms.