

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Αποστολίδης Ιωάννης

Μεταπτυχιακός Φοιτητής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Επόπτης Μεταπτ. Εργασίας: Καθηγητής, Α. Σαββίδης

Τρίτη, 13/10/2015, 12:00

Αίθουσα Β108, Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

" META JAVASCRIPT "

ΠΕΡΙΛΗΨΗ

Ο μεταπρογραμματισμός είναι η ικανότητα διαχείρισης του πηγαίου κώδικα ως πρώτης κλάσης δεδομένα στο μετα-πρόγραμμα και εισαγωγής τους στο πρόγραμμα. Στον μεταπρογραμματισμό πολλαπλών επιπέδων η παραπάνω διαδικασία μπορεί να επαναληφθεί σε άπειρο βάθος, δημιουργώντας δυνητικά άπειρη αναδρομή. Συνήθως η γλώσσα μεταπρογραμματισμού είναι διαφορετική από την κανονική γλώσσα προγράμματος.

Στην δουλειά μας, εστιάζουμε στην JavaScript και την επεκτείνουμε, υποστηρίζοντας μεταπρογραμματισμό. Προς αυτήν την κατεύθυνση δημιουργήσαμε ένα σύστημα μεταπρογραμματισμού όπου η απλή γλώσσα είναι υποσύνολο της μετα-γλώσσας και μοιράζονται την ίδια υλοποίηση. Υλοποιήσαμε τον μεταπρογραμματισμό επεκτείνοντας την ανοιχτού λογισμικού μηχανή JavaScript SpiderMonkey του Mozilla. Υποδείξαμε τις μινιμαλιστικές αλλαγές που χρειάζεται το SpiderMonkey κάνοντας επεκτάσεις στο

λεξικογραφικό αναλυτή, στο συντακτικό αναλυτή, στις εσωτερικές δομές το αφαιρετικού συντακτικού δέντρου (ΑΣΔ), τις συναρτήσεις βιβλιοθήκης, τους μηχανισμούς ανάλυσης της γλώσσας, καθώς και την προσθήκη ενός συστήματος για αποσφαλμάτωση του ΑΣΔ. Στην δική μας προσέγγιση κάθε στάδιο του μεταπρογραμματισμού είναι μια ανεξάρτητη οντότητα όπου συλλέγει τα μετα-σύμβολα για το συγκεκριμένο βάθος, σεβόμενο την σειρά, δημιουργώντας ένα ανεξάρτητο πρόγραμμα.

Επαληθεύουμε την υλοποίηση μας έχοντας δημιουργήσει μία εφαρμογή ηλεκτρονικού ταχυδρομείου χρησιμοποιώντας μεταπρογραμματισμό για να παράξουμε αυτόματα τον κώδικα για τις σελίδες της εφαρμογής. Επίσης, μελετήθηκαν επαναληπτικά μοτίβα πυγαίου κώδικα σε JavaScript, από βιβλιοθήκες που χρησιμοποιούνται στην βιομηχανία, και τις βελτιστοποιήσαμε στα πλαίσια του μεταπρογραμματισμού με σκοπό να πετύχουμε καλύτερη απόδοση και βέλτιστη επαναχρησιμοποίηση τους. Τέλος, υποστηρίζουμε την αποσφαλμάτωση του μετα-κώδικα απευθείας στο περιβάλλον του προγράμματος περιήγησης. Για αυτό το σκοπό υλοποιήσαμε μία εφαρμογή σε μορφή ιστοσελίδας που επικοινωνεί με το κομμάτι αποσφαλμάτωσης του SpiderMonkey. Η εφαρμογή προσφέρει χρήσιμα εργαλεία για την αποσφαλμάτωση του μετα-προγράμματος όπως την απεικόνιση των ΑΣΔ, την εξαγωγή του κώδικα που αντιστοιχεί σε κάθε κόμβο του ΑΣΔ, καθώς και χρήσιμες πληροφορίες για το περιβάλλον εκτέλεσης.

Apostolidis Ioannis

M.Sc. Thesis

Computer Science Department

University of Crete

Master's Thesis Supervisor: Professor A. Savidis

Tuesday, 13/10/2015, 12:00

Room B108, Computer Science dept., University of Crete

“META JAVASCRIPT”

ABSTRACT

Generative metaprogramming is the ability to treat source code as first-class data and emit it to the program itself, thus modifying it. In multi-stage metaprogramming the above procedure can be repeated in infinite nesting level. Usually, the metaprogramming language is quite different

from the normal language, overall resulting in two distinct languages with separate respective implementations.

In our work, we focused on JavaScript and extended the language with a metaprogramming layer programmable directly in JavaScript. This way, we managed to deliver a full-scale metaprogramming framework where: (i) the normal language is a pure subset of the meta-language; and (ii) they share the same underlying implementation. Our JavaScript multi-staging support has been implemented by extending the open source Mozilla SpiderMonkey JavaScript engine. We indicate the minimal modifications required in SpiderMonkey by performing minor extensions in the syntax, parser and internal AST structures, and the addition of an unparser, a staging loop, some library functions and a debugger backend component for AST inspection. Our approach handles each stage as an isolated entity by collecting all the meta-annotations for the specific depth, respecting the order, and creating a coherent program.

We validate our implementation through the implementation of an email application using metaprogramming to fully generate the user-interface code. Additionally, we studied JavaScript source code patterns from the real world libraries and reengineered various parts to achieve improved reusability and performance through metaprogramming. Finally, we supported metaprogram debugging directly in the browser environment using the available tools. For this purpose, we implemented a debugger client as a web page application that communicates with the extended SpiderMonkey debugger backend. The client provides extra features particular to metaprogramming debugging process, including visualizations of ASTs and respective unparsed source code fragments.