Ankündigung einer Lehrveranstaltung im Sommersemester 99

Entwicklung einer viusellen Umgebung für eine visuelle Sprache (Hauptstudiumsprojekt)

1331 L 257, 6 SWS

Ausgehend von graphischen Editoren und einem graph- und regelbasieren Interpreter wollen wir eine vorhandene visuelle Umgebung um verschiedene Werkzeuge für Syntax- und Semantikchecks erweitern. Zum Entwurf verwenden wir die Softwareentwicklungsmethode OMT (Object Modeling Technique) mit UML-Notation (Unified Modeling Language). Implementiert wird in Java.

Die konkrete und abstrakte Syntax einer visuellen Sprache läßt sich durch eine Graphgrammatik beschreiben. Eine Graphgrammatik enthält, ähnlich wie eine Wortgrammatik (z.B. BNF), eine Menge von Regeln und einen Startgraph, der das kleinste Element der Sprache beschreibt. Einer visuelle Sprache ist dann die Menge aller aus diesem Startgraph abgeleiteten Graphen. Ein Graph beschreibt einen visuellen Satz (, eine Graphik oder ein Diagramm), der Symbole und Relationen enthält. Ein visuelles Alphabet besteht nicht nur aus einer Menge von Symbolen, sondern auch einer Menge von Relationen, die zwischen bestimmten Symbolen bestehen können. Symbole können einfache oder auch beliebig komplexe Graphiken sein und typische räumliche Relationen sind z.B. *über, neben, enthält,...*.

Für die meisten visuellen Sprachen gibt es syntaxgesteuerte Editoren, um Graphiken der Sprache einzugeben. Obwohl syntaxgesteuerte Editoren für textuelle Sprachen kaum noch benutzt werden, weil sie recht unflexibel in der Benutzung sind, werden sie für visuelle Sprachen fast ausschließlich gebraucht. Dies mag überwiegend an der Einfachheit der Grammatiken für visuelle Sprachen liegen. Betrachten wir Sprachen mit komplexeren Grammatiken, kann es vorteilhaft sein, freies Editieren auf einem festen Alphabet zu erlauben. Dann muß die erstellte Graphik allerdings parsiert werden. Wir werden in diesem Projekt einen Parsierungsalgorithmus für visuelle Sprachen implementieren.

Desweiteren ist es möglich, auf der formalen Basis der Graphgrammatiken verschiedene Semantikchecks durchzuführen. Wir werden im Projekt eine Teilsprache der UML als visuelle Sprache betrachten. Die UML bietet verschiedene Diagrammarten zur Modellierung von statischen und dynamischen Aspekten von Softwaresystemen. Die Semantikchecks werden z.B. eine Konsistenzanalyse zwischen verschiedene Diagrammen ermöglichen.

Kenntnisse in objektorientierter Modellierung und Programmierung sind erwünscht. Kenntnisse, wie sie in der LV "Visuelle Sprachen und Entwicklungsumgebungen" vermittelt werden, sind von Vorteil. Das Projekt wird auf einer bereits vorhandenen Graphtransformationsmaschine (AGG) aufsetzen. Jeder Student/jede Studentin sollte deshalb die Bereitschaft mitbringen, sich in zugrundeliegende theoretische Konzepte und deren vorhandene Implementierung einzuarbeiten. Die Fähigkeit, sich schnell in eine bereits bestehende Software einzuarbeiten, ist eine der wichtigsten im späteren Arbeitsalltag eines/r Softwareentwicklers/in.

Die LV ist als Projekt im Studiengebiet 'THI' oder 'SSG' anrechenbar.

Veranstalterin: Gabriele Taentzer, FR6019, Tel.: 314 - 27787
e-mail: gabi@cs.tu-berlin.de

Termine: Mo, 14-16 Uhr
Mi, 14-16 Uhr
Beginn: Montag, den 19. April 1999

Anmeldung: Per E-mail an `gabi@cs.tu-berlin.de` **oder spätestens am 19. April in der LV**

Weitere Informationen: http://tfs.cs.tu-berlin.de/lehre/teaching.html