

Összefoglaló

A modellvezérelt szoftvertervezési megközelítés különböző rendszermodelleket helyez előtérbe a fejlesztési életciklus során. Ezen modellek elkészítése, majd feldolgozása kiemelkedő fontosságú, ehhez pedig széles körű nyelv- és eszköztámogatás szükséges.

A modellvezérelt fejlesztést támogató eszközökhöz hatékony támogatást nyújthat a modell-lekérdezések technológiája, pl. a modellezési nyelv jólformáltsági kényszereinek validációjára, modelltranszformációk elvégzésére, stb. A lekérdezések elvégzése történhet általánosan, programozott módszerrel is; kifejezőbb lehet azonban, ha valamilyen speciális, deklaratív nyelvvvel fogalmazunk meg egy ilyen lekérdezést, és a kiértékelés ez alapján automatikusan történik meg.

A Méréstechnika és Információs Rendszerek Tanszéken készült EMF-IncQuery eszköz lehetővé teszi az Eclipse Modeling Framework-alapú modelleken a lekérdezéseket. Segítségével egy deklaratív nyelvvvel fogalmazhatóak a lekérdezések, az ún. gráfminták formalizmusa szerint. A saját deklaratív nyelv hátránya, hogy már egyszerűbb minták megfogalmazása is szakértelmet igényel, továbbá a különböző kényszerek megfogalmazásához ismerni kell a modell absztrakt reprezentációját, ami a konkrét szintaxis szerinti használatkor nem feltétlenül elérhető minden esetben.

A modell-lekérdező rendszerhez hatékony kiegészítést nyújthat egy olyan eszköz, amely a felhasználó által a modellen példaként kijelölt elemeket veszi alapul, és a hozzájuk hasonló elrendezésű elemeket azonosító gráfmintát állít elő kimeneteként. A módszer automatikusan elvégzi a példánymodell felderítését, majd lehetőséget ad a felhasználónak különböző finomhangolási beállítások elvégzésére, végül a példára illeszkedő, futtatható lekérdezést generál. A kiterjesztés felhasználói interfészét ideális lenne úgy megoldani, hogy azt az IncQuery felületével hatékonyan együtt lehessen használni egy kompakt fejlesztőkörnyezet képében.

Az EMF-IncQuery keretrendszer számos EMF-alapú konkrét szintaxis szerkesztőt támogat, így fontos, hogy a példavezérelt fejlesztőkörnyezet is együttműködjön ezen fejlesztőkörnyezetek jeletős részével.

Abstract

The model-based software design approach forces the usage of various system models during the development lifecycle. The creation of these models, and after that, their processing is extremely important - for these tasks, extensive language and device support is necessary.

The technology of model queries could be a great assistance for the model-driven development tools, for example validating the constraints of a modeling language, performing model transformations, etc. These queries might be implemented via the programming way, but it could be more expressive, if we define a query like this with a special, declarative language, and is automatically evaluated based on it.

The EMF-IncQuery tool, which was developed in the Department of Measurement and Information Systems, allows model queries over the Eclipse Modeling Framework instance models. An own, declarative language is provided to draw up the queries, by the formalism of graph patterns. The disadvantage of an own, declarative language is that even for the simpler patterns' definition, special skills are required, furthermore the knowledge of the given model's abstract representation is also required for wording various constraints, and it is not necessarily provided in every single case.

For this model-query system, an effective extension could be a tool, that is able to produce a graph pattern by an example, which consists of the selected elements by the user on an instance model. The output graph pattern will identify all of model elements, that have a similar layout. The method automatically explore the given instance model, allows the user to perform omnifarious fine tuning settings, finally generates a runnable query that will match the example. It would be ideal to have a user interface for the extension, that could be used together with IncQuery's views within a compact development environment.

The EMF-IncQuery user interface supports several EMF-based concrete syntax editor, so it will be important to make the example-based development environment to cooperate with most of these environments.