Project summary

Project name	Data Lineage Analysis of C# Programs for Manta
Acronym	MANTACS
Supervisor	Pavel Parízek (parizek@d3s.mff.cuni.cz)
Consultants	Lukáš Hermann (lukas.hermann@getmanta.com)
Abstract	The main goal of this project is to design and implement the support for data lineage analysis of C# programs in the context of the Manta Flow platform. Team members will have to develop four main components – interpreter of the CIL intermediate code, symbolic analysis of data lineage for C# programs that access databases and perform I/O, transformation of analysis results into flow graphs, and integration with the Manta Flow platform.

Motivation

Data lineage is an important property of software systems that involve databases. It enables tracking of data flow and transformations that make a part of audit information required, for example, by government regulators. Manta Tools is a software company whose main product is Manta Flow, a platform for data lineage analysis. Currently it supports many database systems (e.g., including Oracle and MS SQL) and Java programs together with popular frameworks, such as MyBatis, Apache Kafka and Spring.

Multiple customers of the Manta Tools company use also programs written in C#, and therefore expressed their desire for data lineage analysis of C# programs.

Project description

The main goal of this project is to design and implement data lineage analysis for C# programs within the context of the Manta Flow platform. We call the system to be created as C# scanner.

Actually, the scanner will analyze programs in the CIL intermediate code for the .NET platform that is produced by a compiler from the C# source code. This could make the scanner applicable also for programs compiled from other programming languages, such as Visual Basic, but only programs written in C# will be used for testing within the scope of this project. We plan to support a large subset of the features of C# and the core libraries for database access and I/O, neglecting features that are difficult from the perspective of static program analysis (e.g., reflection and threads). A precise list of supported features of C# and relevant libraries will be provided in the detailed specification. In addition, design of the scanner will support future extensions in the form of plugins for popular data management frameworks on the .NET platform, such as the Entity Framework.

Members of the project team will reuse many concepts and algorithms from the existing analysis of Java programs, which has been developed by the project supervisor. They will adapt the concepts and algorithms for C# where needed, but the implementation will be done completely from scratch. The general approach is to use a symbolic modular analysis that computes data flow summaries for all methods in a given program, and then create a nice data flow graph from the method summaries.

We plan to decompose the system into 4 main components – interpreter of the CIL intermediate

code, symbolic data lineage analysis, transformation of the symbolic analysis results into flow graphs, and integration with the Manta Flow platform.

Other important parts of the project will include (1) a large test suite, involving both unit and integration tests, and (2) a very extensive documentation (to be written in English).

Technology

Implementation of the C# scanner will build upon popular frameworks, libraries and tools for processing of CIL intermediate code and C# source code, including especially Mono.Cecil and the Roslyn compiler framework.

Team members will actively follow best practices according to guidelines provided by consultants from the Manta Tools company, and they will also use the development environment and setup provided by the company (including Git repositories, Jenkins for continuous integration, etc).

Expected effort

Number of participants: 4

Completion date: 9 months from the start

Main project schedule and overall plan:

- The first step will be to create a layer over the Mono.Cecil library that will provide the class hierarchy information and call graph.
- The second step will be to develop the interpreter of the CIL intermediate code and prepare test subject programs for all the important features of C#.
- After that, project team members will design and implement the remaining components of the C# scanner. Each member should be primarily responsible for one component.

We already have a team of four students who plan to join this project. They have already become familiar with the Manta Flow platform, and currently they are learning technical details of the existing implementation of the data lineage analysis for Java programs.

Consultants from the Manta Tools company will actively participate in supervision of the project, and they will also provide technical support. Regular weekly meetings are planned for the whole duration of the project.

Project characterization

The project targets the following areas (mark suitable areas):

Discrete models and algorithms	
	discrete mathematics and algorithms
	geometry and mathematical structures in informatics

	optimization
Theoretical informatics	
	theoretical informatics
Software and data engineering	
x	software engineering
x	software development
	web engineering
	databases
	big data analysis and processing
Software systems	
	system programming
	dependable systems
	high-performance systems
Mathematical linguistics	
	computer and formal linguistics
	statistical methods and machine learning in computer linguistics
Artificial intelligence	
	smart agents
	machine learning
	robotics
Computer graphics and game development	
	computer graphics
	game development