Detecting deadlocks using static analysis in .NET

Filip Navara

filip.navara@gmail.com

What did I do last week?

- Located old version of eM Client before the tool was first used on the project...
  - Revision 2039 (current revision is > 9900) from October 18, 2007.
  - ... and used the original L.O.V.E. prototype based on updated CSLint to determine what problems were found.
  - Purpose: Regression testing, Determining improvements

New options

- Added options to mimic the old CSLint behavior and allow other experiments:
  - „--noaliasing“ ... Treats all fields as unaliased
  - „--noaliasingaftermerge“ ... Make the symbolic heap objects that callee locks unaliased when merging into caller
  - „--ignoresystemnamespace“ ... Don‘t analyze methods in the „System“ namespace (ie. the .NET Framework itself)

Delegate resolution

- Fixed bug in delegate resolution that caused no delegates to be resolved to the called function during interprocedural analysis!
  - Now the analysis takes a lot longer on complex programs and consumes roughly twice as more memory.
  - The delegate resolution suffers greatly from the lack of alias analysis, but at least now there is a clear room for improvements.

Old CSLint prototype

- Some of the lock order violations were found due to bugs in the original prototype!
- Plenty of the false positives, at least when run against more recent eM Client versions, were introduced due to the lack of interprocedural analysis. Main cause were reentrant acquisitions of locks.
- Only small part of the program was analyzed, because virtual methods and delegates were not resolved. Thus only few false positives were reported.

Current L.O.V.E. prototype

- Analysis of eM Client 2039 takes roughly 15 minutes and consumes about 1.8 Gb memory with the –noalias –ignoresystemnamespace options.
- It finds all the deadlocks that the old CSLint prototype did...
- ...and at least one that wasn‘t found by the old prototype...
- ...and plenty of false positives.

What do I plan to do next week?

- **Paper! Paper! Paper!**
- Incorporate static constructors into the analysis
  - Williams et al. also missed this in the original thesis, but later mentioned it in the ECOOP 2005 paper