Bachelor / Master Thesis Topic

Extraction of mode diagrams from Blech

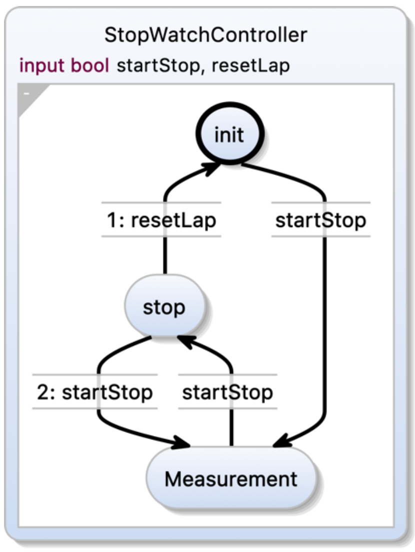
Robert Bosch GmbH
Renningen
70465 Stuttgart
Besucher:
Robert-Bosch-Campus 1
71272 Renningen
Telefon +49 711 811-0
www.bosch.com

The new programming language Blech (blech-lang.org) offers high-level abstractions and safety guarantees for reactive, real-time embedded programming. Blech leverages imperative, synchronous control flow to specify behaviour in terms of sequential workflows and concurrent compositions thereof. In particular, modes of operation can be encoded concisely. There is no need to specify all program locations and all possible transitions between them explicitly. Blech compiles to C code, which may be integrated into existing projects or simulation frameworks.
In practice, visualisations of programs may greatly improve documentation and provide an intuitive understanding of the structure or behaviour of the code. For example, the control flow graph gives a quick overview of the structure of a program. The behaviour is often visualised by graph consisting of “modes” and transitions between them. This represents the stateful nature of the program. Extracting this diagram from code is the topic of this thesis project.
The challenge is to find the right level of abstraction or to provide the user with an intuitive interface to choose the granularity of the visual representation. One extreme would be to translate Blech programs into state charts. They represent every aspect of a synchronous program including every computation step but hence become too large and complex to serve as an intuitive overview over the intended modes of operation. The opposite extreme is to interpret an activity call in Blech as a start of a new mode and produce an activity call graph ignoring all other statements. This may be too coarse and does not necessarily reveal the intended mode transition system.

Example, with original code and – here very abstract – diagram to be extracted automatically:

```
activity StopWatchController (startStop: bool, resetLap: bool)
                             (display: Display)
    var totalTime: int32
    var lastLap: int32
    repeat
        totalTime = 0     // State init
        lastLap = 0
        writeTicksToDisplay(totalTime)(display)
        await startStop   // Transition init -> run
        repeat
            cobegin weak
                await startStop
            with weak
                run Measurement(resetLap)
                               (totalTime, lastLap, display)
            end
            // State stop, show total time and wait
            writeTicksToDisplay(totalTime)(display)
            await startStop or resetLap
            // Run again if only startStop was pressed
        until resetLap end // Back to init if
    end                    // resetLap was pressed
end
```

$\Rightarrow$



Note that the example shows just one activity with one transition diagram. One could also expand a mode which is represented by a subactivity and inline that activity's graph. This gives rise to hierarchical behaviour models.

Project goals:

- Propose an algorithm that, given the abtract syntax tree of the code, extracts mode diagrams as above
- Discuss different levels of detail or abstraction of the visualisation
- Implement it as a Visual Studio Code plugin to allow viewing code and diagram side by side, using KIELER/ELK diagramming/auto-layout technology
- Implement expansion or stepping into modes

Prerequisites:

- Proactive working attitude, self-motivation
- Fast, direct communication
- Interest in visualisation, UI, web-technology

What we offer:

- Experience Bosch's research campus in Renningen, alternatively conduct work at Kiel University
- A friendly working environment
- Supervisors you can quickly reach any day
- Compensation for living costs, if work is carried out in Renningen

Contact:

Dr. Friedrich Gretz, CR/AEE1
Friedrich.Gretz@de.bosch.com Phone +49 711 811-43645

Prof. Dr. Reinhard von Hanxleden
Department of Computer Science
Kiel University
rvh@informatik.uni-kiel.de Phone +49 431 880-7281