Vrije Universiteit Brussel

Proposal bachelor thesis

Title: A Tool for Visualizing the Runtime Behavior of a Distributed Rule-based Pattern Recognition Engine

Promotor: Prof. Dr. Wolfgang De Meuter

Advisor: Thierry Renaux

Includes preparation course: No

Context

Technologies in application domains ranging from multitouch gesture recognition to automated detection of fraudulent credit card transactions, make use of pattern recognition software. The two main branches of pattern recognition can be roughly described as machine learning (i.e., having software build a model from a set of examples and counterexamples) and declarative specification (i.e., having domain experts write down "rule of thumb"-style descriptions of *what* the problem is, not *how* to solve it, and providing software which can interpret those descriptions). The thesis would be about the latter approach.

One of the research artifacts created at the Software Languages Lab is PARTE. PARTE is a runtime environment and an accompanying declarative language specifically tailored towards recognizing patterns in streams of events. The term events is traditionally used in computer science to refer to data which has a temporal component. Examples include mouse-clicks, the (dis-)appearance of a network connection, etc., but also less 'dry' examples such as finger positions on a touchscreen, changes in device orientation, body poses tracked by a Kinect system, as well as for instance the checking out of a shopping cart in a webshop.

PARTE differs from most other pattern recognition systems in that it consists out of many independently running 'actors' [1] forming a Rete graph [2]. Loosely speaking, this means a PARTE instance takes the form of multiple (easily multiple hundreds) processes which cooperate by exchanging messages. This offers a bunch of interesting properties, a.o. for scaling up over multiple processor cores, for distribution over multiple machines, or for dealing with catastrophic failure in part of the system. Yet, since the actors work in parallel, the dynamics of the system are complex. A linear textual log is not able to clearly convey what happens in a running PARTE instance.

Proposal bachelor thesis

The proposed thesis consists out of designing, implementing and validating a (graphical) visualization tool that can depict the graph of actors, displays memory- and processor-load, as well as (the rate of) the stream of messages sent between actors. The tool should be able to run together with a PARTE instance, and continuously present an overview of the current state of the PARTE instance's actors.

The student has some liberty in picking the representation, but has to take into account the orders of magnitude that can be expected: message rates upwards of a thousand messages per second per edge are likely to occur. The visualization tool should not delay the PARTE instance significantly; the PARTE instance may not have to wait for the visualization tool. Hence, the visualization tool needs to display an eventually consistent [3] approximation on the state of the PARTE instance.

Concretely, the milestones for the thesis are:

1. get acquainted with the idea of asynchronously communicating actors [1];
2. make a (static) representation of the graph of entities forming a PARTE instance. PARTE's existing GraphViz Dot [4][5] output can serve as a starting point;
3. design and implement a way to asynchronously (periodically or depending on runtime factors) publish statistics from a PARTE instance to an external tool;
4. design and implement methods for visualizing those runtime statistics dynamically on the graph from milestone 2;
5. analyze the impact of the tool on PARTE, and demonstrating that the interference is acceptable for a monitoring and debugging tool.

References

[1] http://en.wikipedia.org/wiki/Actor_model
[2] http://en.wikipedia.org/wiki/Rete_algorithm
[3] http://en.wikipedia.org/wiki/Eventual_consistency
[4] http://en.wikipedia.org/wiki/Graphviz
[5] http://www.graphviz.org