PUC–Rio

Research Plan

Safe Dynamic Abstractions for Embedded Systems

Autor:
Francisco Sant'Anna

Advisors:
Roberto Ierusalimschy
Noemi Rodriguez

September 24, 2013

Contents

1 Abstract

This document describes the research plan for a one-year postdoctoral program in PUC–Rio. The proponent, Francisco Figueiredo Goytacaz Sant'Anna, is a PhD in computer science at PUC–Rio.

The proposed research plan is entitled "Safe Dynamic Abstractions for Embedded Systems" XXX how to extend the synchronous programming languages CÉUwith dynamic abstractions

describes how the programming language "CÉU", being developed by the proponent during his PhD, can be used in the context of Wireless Sensor Networks and secure communications.

2 Introduction

Software for embedded systems is usually developed in the C programming language, and the addition of a real-time operating system may extend it with preemptive and/or cooperative multithreading [**?**, **?**]. However, concurrency in C requires a low-level exercise related to scheduling, synchronizing, and the life cycle of activities (i.e. creating and destroying threads). Concurrency in C also lacks safety warranties, given that they are susceptible to unbounded execution, race conditions and deadlocks. Nonetheless, safety is a fundamental aspect in embedded systems, which have scarce resources and must run for long periods without human intervention.

The programming language CÉU [**?**] developed as part of the proponent's PhD research in PUC–Rio is targeted at highly constrained embedded systems (such as Wireless Sensor Networks [**?**]) and incorporates features found in dataflow and imperative reactive languages [**?**, **?**]. CÉUsupports concurrent lines of execution that run in time steps and are allowed to share variables. However, the synchronous and static nature of CÉUenables a compile time analysis that can enforce deterministic and memory-safe programs, offering a high-level and safe alternative to the predominating C based multithreaded systems. The CÉUcompiler generates code comparable to handcrafted C programs in terms of size and portability.

Currently, both data and control memory are managed manually, an approach that is known to be hard and error prone. In this work we show how to extend a synchronous concurrent language to provide safe and automatic management of dynamic memory. Our strategy reconciles data and control state into the single concept of *organisms*, which provide multiple lines of execution with an object-like interface. Unlike objects in Java, organisms are lexically scoped, and their life cycle is controlled following the structure

of the program, rather than by runtime reachability algorithms. As a result, memory reclamation is efficient, providing rich dynamic functionality as in Java, with little overhead in comparison to C-based systems.

3 Objectives

The main objective of our research during the sandwich period in Chalmers is to evaluate the applicability of CÉU in the context of Wireless Sensor Networks and secure communications. The evaluation involves quantitative measures (e.g. memory usage) and qualitative measures (e.g. ease of programming).

The Distributed Computing and Systems group in the Chalmers University has an open PhD Sandwich position in the area of "Secure Wireless Sensor Networks". The research position describes programming languages technologies as one of its focus: *"The Security group has developed the link between two areas of computer science: programming languages and computer security. The group explores security models and enforcement mechanisms based on programming-language technology"*.

We believe that Wireless Sensor Networks are an ideal scenario to employ the CÉU programming language, given it is targeted at highly constrained embedded systems, offering fine-grained concurrency, low memory overhead, and safety warranties.

Many number of software services are required when building a sensor network application. The technique of layering services on top of each other, building functionality on top of functionality and, at the same time, separating them into manageable units are a powerful approach.

As an illustrative example, WSN applications typically run tweaked versions of classical distributed algorithms, such as *topology discovery*, *leader election*, *broadcast*, etc [**?**]. These algorithms give support for more general purpose applications and must run continuously with them, given the dynamicity of WSNs.

The WSN group at PUC–Rio is currently working on developing a library of a set of such classical algorithms. With CÉU, these algorithms are easily composed with a parallel construct, making the main application run concurrently with the distributed algorithm seamlessly. The language detects any inconsistency with the composition, such as concurrent access to shared variables and unreachable code. This idea can be extended to algorithms used and/or developed by the DCS group in Chalmers (e.g. [**?**]).

We are also planning to investigate ways to describe an approach for synthesizing data representations for concurrent programs. It is part of the

process to evolve the language together with the DCS group when identifying new requirements not currently addressed.

The DCS group in Chalmers also does research in many programming languages topics, which can also be explored during the PhD Sandwich [**?**, **?**, **?**]. The group has a research library of concurrent constructs (NOBLE[1]) and a testbed of applications ranging from SmartGrid ones to connected car applications (GULLIVER[2]).

The following list of research topics, aligned with the programming model of Céu, was extracted from the DCS website:

- Investigate new techniques for achieving high parallelism and fault-tolerance in distributed or parallel software.
- Evaluate the performance of non-blocking synchronization in parallel application and system software.
- Non-blocking/fine-grain synchronization, aiming at increased parallelism, fault-tolerance, avoiding convoy effects and priority inversion.
- Enhancing performance by cooperative scheduling-and-synchronization.

4 Methodology

During the period of the PhD Sandwich, we intend to develop fully working WSN applications in Céu in order to evaluate its viability as an alternative to the current employed technologies. We will use quantitative and qualitative metrics for the relevant aspects in WSNs, and compare the implementations of applications in Céu and other languages (e.g. C).

An existing work [**?**] measures the performance of different programming languages regarding memory usage, battery consumption, and responsiveness specifically for WSNs. These aspects are of extreme importance, given the severe hardware constraints in this context.

We also consider safety an important aspect, as motes must run for long periods without human intervention. Finally, high expressiveness, is desired for any programming language in any context.

Follows the list of aspects we will evaluate during our research:

- Memory usage: how much applications use in terms of volatile (RAM) and non-volatile (ROM) memory.

[1] http://www.cse.chalmers.se/research/group/noble/
[2] http://www.chalmers.se/hosted/gulliver-en/

- Battery consumption: how much energy applications consume.
- Responsiveness: how fast applications acknowledge high-priority requests (e.g. radio messages).
- Safety: which warranties the language offers, releasing the programmer from such concerns.
- Expressiveness: how easy applications can be developed and maintained.

The first three aspects can be easily evaluated with quantitative metrics, while the last two require a more in-depth analysis.

Initially, we will port to Céu a set of existing applications developed in other languages and compare the quantitative aspects of the implementations. The results immediately provide feedback regarding the implementations in Céu, which can be used to improve the language.

Then, we will focus on developing new applications with higher complexity in order to evaluate the qualitative aspects. We can still use quantitative measures such as number of lines of code and period of development to help on the evaluation.

5 Schedule

Our proposal includes a six-month schedule that we intend to follow during the sandwich period:

	M1	M2	M3	M4	M5	M6
Presentation of Céu	X		X		X	
Development of existing apps	X	X				
Improvements on Céu		x	X	x	X	
Development of new apps			X	X	X	
Paper					X	X
Other activities	x	x	x	x	x	x

(**X** cells indicate high activity, while **x** cells indicate low activity)

Months 1–2: During the first month, we will present the language Céu to the research group, focusing on its applicability to WSNs and on how it differs from existing systems.

Then, with the support of the group, we will choose existing WSN applications to be ported to Céu in order to perform the quantitative analysis of the implementations until the end of the second month.

Months 3–4: With the work on existing applications, we will collect a comprehensive feedback to be used for improvements on the language.

We will also present the achievements of the first two months, and discuss the development of new applications with higher complexity for the next two months.

Months 5–6: By the beginning of the fifth month, we expect to have developed some complex applications that can be used in a in-depth qualitative analysis of the language.

Both the quantitative and qualitative analysis will be used in a paper to be written to a conference on WSNs.

Months 1–6: We also have interest in exploring other research areas of the group related to programming languages and distributed systems.

6 Expected Results

By the end of the PhD Sandwich, we expect that Céu becomes a real alternative for the development of fully working WSN applications. Our methodology involves a quantitative analysis that provides immediate feedback regarding key aspects in WSNs, such as memory usage and battery consumption. Hence, we can evolve the language in a small development cycle.

We have already presented a short paper about Céu in the doctoral colloquium of SenSys last year [**?**], with some initial experiments that show competitive results in terms of memory usage and responsiveness. With a broader usage, we will have more confidence on the measurements and achieve even better results.

We also expect that the ongoing research in our WSN group at PUC–Rio will take advantage of the interchange with the group in Chalmers and vice-versa. Both groups share common research interests and a continuous relationship may arise from this first experience. For instance, the library of distributed algorithms for WSNs being developed by the group at PUC–Rio is open-source and can be more easily adopted by the group at Chalmers through this interchange program.

We intend to write a full paper to a top conference together with the group in Chalmers to describe the advances of Céu in the context of WSNs:

- ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN). Submission deadline expected to October 2012.
- ACM Conference on Embedded Networked Sensor Systems (SenSys). Submission deadline expected to April 2013.

Another outcome of our research could be the adoption of Céu in new projects in the DCS group in Chalmers, which need not be directly related to research in programming languages.

7 Researchers

PUC–Rio (proponent):

Francisco Sant'Anna is a third year Ph.D. student in the Computer Science department at PUC-Rio. He earned his BSc (2003) and MSc (2007) degrees also in the Computer Science department at PUC-Rio.

The current title of his PhD thesis is "Céu: Embedded, Safe, and Reactive Programming", which is expected to be concluded in September 2013. His advisors are Prof. Roberto Ierusalimschy and Prof. Noemi Rodriguez, which actuate, respectively, in the field of programming languages and distributed systems.

Prof. Roberto Ierusalimschy is an associate professor of informatics at PUC-Rio (Pontifical University in Rio de Janeiro). He is the leading architect of the Lua programming language and the author of Programming in Lua.[**?**]

His research activities currently focus on programming languages, mainly alternative languages such as scripting languages and domain-specific languages.

Prof. Noemi Rodriguez is an associate professor of informatics at PUC–Rio. Her research interests include the area of concurrent and distributed programming, with an emphasis on the role of programming languages in this context. She also leads the Wireless Sensor Networks research group at PUC–Rio.

Chalmers University (host):

Prof. Philippas Tsigas is a professor in the Department of Computing Science and Engineering at Chalmers University. He is the co-leader of the Distributed Computing and Systems Research Group. His research interests center on distributed/parallel computing and systems and information visualization in general.

Homepage: `http://www.cse.chalmers.se/~tsigas/index.html`