5-16-2011

PDC Modules for Every Level: A Comprehensive Model for Incorporating PDC Topics into the Existing Undergraduate Curriculum

Konstantin Läufer
Loyola University Chicago

Chandra N. Sekharan
Loyola University Chicago, csekhar@luc.edu

George K. Thiruvathukal
Loyola University Chicago

PDC Modules for Every Level: A Comprehensive Model for Incorporating PDC Topics into the Existing Undergraduate Curriculum

Konstantin Läufer Chandra N. Sekharan George G. Thiruvathukal

Department of Computer Science
Loyola University Chicago
{*laufer,chandra,gkt*}*@cs.luc.edu*

***Abstract*—We propose to give a half-hour presentation on incorporating PDC topics into our existing undergraduate computer science curriculum. We first describe our experience with single-course and cross-curricular undergraduate PDC education since 1997. We then provide details about our proposed model for incorporating PDC topics into our undergraduate curriculum. To this end, we have distilled a set of consistent, regularly offered and required three-week core and advanced PDC modules from our previously offered PDC content.**

***Keywords*-Computer science education, parallel and distributed computing education, concurrency, distributed computing, distributed systems, parallel computing, curriculum design, programming methodology, CSE, PDC**

I. Overview

Our proposed half-hour presentation comprises the following two themes: 1) experience with incorporating PDC topics into core computer science courses, and 2) models for incorporating PDC topics into the core computer science curriculum. We start by describing our experience with undergraduate PDC education since 1997: For almost a decade, we focused mostly on a regular single-course offering but, for the last four years, switched to a cross-curricular effort involving about seven advanced/elective courses in response to scheduling needs. Now, based on our collective experience, the growing importance and broader availability of novel architectures, and general progress on the relevant paradigms, we propose a new model for incorporating PDC topics into our undergraduate curriculum. To this end, we have distilled a set of consistent, regularly offered and required three-week core and advanced PDC modules from our previously offered PDC content.

II. The Past: Our Early Single-Course Approach

As active researchers and before-the-fact adopters, we have not only argued for covering PDC topics in a foundational, application-independent fashion early in the major, but we have offered a course in concurrent programming in a functional/object-oriented environment aimed at sophomores (preferably in their fourth semester) since fall 1997 [1]. This offering has been well-received and successful as measured informally in terms of students observed performance in subsequent related courses, research projects, and industry.

III. The Present: Our Later Cross-Curricular Approach

In response to departmental staffing and scheduling needs, our single-course offering had become irregular. Therefore, we have switched to the cross-curricular alternative discussed in our paper, that is, breaking up this body of knowledge into learning units we incorporated across our fairly broad range of existing electives in suitable ways. This approach has worked for us and now turns out to be consistent with the stated TCPP rationale. PDC topics have regularly been taught as part of the following advanced/elective courses, of which two are usually offered every semester:

- CS 322: Software Development for Wireless/Mobile Devices
- CS 338: Server-Side Software Development
- CS 339: Distributed Systems
- CS 342: Web Services Programming
- CS 364: High-Performance Computing
- CS 372: Programming Languages
- CS 373: Advanced Object-Oriented Development

IV. The Future (I): Our Proposed Set of Required Core Modules

In an effort to regularly and consistently expose all undergraduate majors to PDC core knowledge, however, we have decided to push a selection of suitable topics down into required foundation courses typically taken in the sophomore year. Specifically, we are in the process of designing the following three-week core PDC modules (20% of our 15-week semester or 30% of a 10-week quarter), consisting mostly of suitable K and C level topics:

- *Introduction:* to be offered every semester in our CS2 course (CS 271). Introductory topics from all areas, such as parallel control statements; shared memory language extensions and libraries; tasks, threads, and synchronization; and performance considerations. We

are envisioning C# as the teaching language (at least for this module) because it provides well-designed mechanisms that correspond to these topics along with foundationally sound teaching materials [2]; thanks to the Mono Project, C# and other .NET CLI languages can be used on all major platforms.

- *Programming:* to be offered every semester in our intermediate object-oriented development course (CS 313). Paradigms (mostly thread-based) and related semantics and correctness issues; parallel programming paradigms; parallel programming notations; semantics and correctness issues; and select concurrency topics. C# would be an effective teaching language for this module and the entire course.
- *Architecture:* to be offered every fall in our intro systems/architecture course (CS 264). The various dimensions of the architectural design space, memory hierarchy, data representation, and performance metrics.
- *Algorithms:* to be offered every spring in our data structures and algorithms course (CS 363). Models of computation and complexity, basic algorithmic paradigms, and specific problems and their algorithmic solutions.

V. The Future (II): Our Proposed Set of Advanced/Elective Modules

Several advanced modules in programming and distributed computing, typically offered every three semesters, are also under development:

- *Advanced Programming:* parallel programming and concurrency topics from a programming language principles and paradigms perspective—including actors, software transactional memory, and data parallelism—using F# or Scala for programming projects. This module will be incorporated in the following courses:
  - CS 372: Programming Languages
  - CS 373: Advanced Object-Oriented Development
- *Distributed Foundations:* foundational topics including architecture classes, models and complexity, and concurrency topics. This module will be incorporated in the following courses:
  - CS 339: Distributed Systems
  - CS 364: High-Performance Computing
- *Distributed Programming and Applications:* languages, frameworks, and software architectures for distributed computing—including client-server and message-based approaches—, semantics and correctness issues, performance issues, and advanced topics. This module will be incorporated in the following courses:
  - CS 322: Software Development for Wireless/ Mobile Devices
  - CS 338: Server-Side Software Development
  - CS 342: Web Services Programming

VI. Tying Everything Together

We are planning to develop the core PDC modules during spring and summer of 2011 so that we can start incorporating them in the fall. Meanwhile, we will continue teaching PDC topics in advanced/elective courses and developing the corresponding modules.

Evaluation and dissemination will be key aspects of the proposed broad-based integration of PDC topics into our existing curriculum without the creation of new courses. We plan to measure the effectiveness of the integration of PDC topics into our curriculum quantitatively and qualitatively, and longitudinally over a three- to five-year period. We intend to refine our evaluation plan further by working with the TCPP, fellow early adopters, and Loyola's Center for Science & Math Education. For dissemination, we plan to hold workshops for subsequent adopters in the Midwest.

Selected References

[1] C. Colby, R. Jagadeesan, K. Läufer, and C. Sekharan. Interaction, Concurrency, and OOP in the Curriculum: a Sophomore Course. *OOPSLA 1998 Educators' Symposium and Poster.*

[2] T. Ball et al. *Practical Parallel and Concurrent Programming.* Microsoft Research. Available at http://ppcp.codeplex.com/.