PADS – A Model Driven Engineering Framework for Learning Distributed Systems Algorithms

Yogesh Barve, Anirban Bhattacharjee and Aniruddha Gokhale
Department of Electrical and Computer Science Engineering
Vanderbilt University
Nashville, TN, USA
{ yogesh.d.barve, anirban.bhattacharjee, a.gokhale }@vanderbilt.edu

***Abstract*— The design and development of reliable and fault tolerant distributed systems requires a deep understanding of fundamental concepts and algorithms that make them operational. With an ever increasing presence of internet of things in our day to day lives, and emergence of new technologies like fog and edge computing gaining traction among the research and industrial community, understanding of distributed systems algorithms becomes even more crucial. Practical implementation of these algorithms and a system for prototyping and testing plays a key role in the holistic understanding of distributed systems. Today, a lack of learning tools which provide an end-to-end framework for learning, designing and testing of distributed systems algorithms on real and/or emulated environments, poses significant impediments to the learning of these algorithms. To tackle these concerns, we have developed a model-driven learning environment called Playground of Algorithms for Distributed Systems (PADS). The PADS design is based on feature modeling and model driven engineering. Using PADS' domain-specific modelling language, a learner can specify distributed system network topology, different actors in the distributed system algorithm, and algorithmic constraints using our model driven web based tool. The web based tool can then be used to generate the associated implementation glue code, and deploy and run the experiments on real/emulated platforms to enable a study of the algorithm and its behavior. In this tutorial, we shall cover the fundamentals of domain specific modeling using the WebGME web-based MDE tool, writing model interpreters and code generation. Using an example publish/subscribe distributed system, we shall introduce how to use PADS in the construction of distributed systems algorithms and its deployment in a Mininet emulation framework which showcases ease of use and rapid deployment benefits of using PADS.**

Keywords— Learning systems, model-driven engineering (MDE), distributed systems

I. Tutorial Organization

This half day tutorial (3.0 hours including break) will be organized as follows:

Part I: Introduction & Overview:

1. Welcome and Introductions: Welcome message and introduction of speakers and audience members (Time: 15 mins)
2. Complexities in learning Distributed Systems: Concepts and issues. (Time: 20 mins)
3. Role of MDE & Feature Models: Describe how model driven engineering and feature modelling can help resolve the problem (Time: 15 mins)
4. MDE Fundamentals using WebGME: Fundamental concepts of MDE used in PADS' including domain-specific modeling languages (DSMLs) and generative programming. (Time: 20 min)

Part II: Hands-on Activity:

5. PADS Modeling-I: Introduction to DSMLs and their meta-models in PADS for distributed systems algorithms. (Time: 20 min)
6. Coffee break (Time: 30 mins)
7. PADS Modeling-II: Introduction to construction of example models in the context of PADS for distributed systems algorithms. (Time: 40 min)
8. Code Generation: Model interpretation technique used in the PADS for targeting different deployment simulation and emulators.(Time: 10 mins)
9. Execution & Deployment of Algorithms: Deployment of the constructed distributed system algorithm on target deployment platform. For the tutorial we will use Mininet emulator to do deployment and execution of the system. (Time: 20 min)
10. Discussions: Questions and answers throughout the tutorial. Feedback from audience. Next steps discussions. (Time: 20 min)

II. Level of Expertise & Knowledge Required

We expect **intermediate** level of expertise. Attendees are expected to have a basic understanding of model driven engineering (MDE) technologies like meta models, models and code generation.

III. Intended Audience and Tutorial Outcomes

Educators, researchers and graduate students who are seeking to understand, implement and or teach distributed systems algorithm for research and study purpose. Using PADS

framework one can construct distributed systems topologies like star, mesh or any custom topologies for varied scales and quickly generate the necessary topology specific codes for the targeted deployment platform such as emulator or simulator.

By using a concrete example for a publisher/subscriber scheme of communication in distributed systems, the audience will find the ease of use, visual modelling and code generation benefits in addition to algorithm learning from using PADS MDE in a practical teaching problem through this tutorial very interesting. Upon completing this tutorial, attendees will be able to:

1. Recognize the inherent and accidental complexities involved in designing and developing distributed systems algorithms.
2. Gain knowledge as to how MDE techniques and tools can help alleviate such complexities in the design of a learning tool for teaching distributed systems algorithms.
3. Activity-based learning of using MDE design techniques presented in the PADS framework for creating DSML, meta models, models and writing model interpreters for constructing distributed systems algorithms.
4. Hands on knowledge of how to use web based WebGME modelling framework through the use of PADS learning toolkit.

IV. Required equipment for tutorial classroom and participants

Overhead projector for laptop required; paper board desirable, internet connectivity required. We shall provide a virtual machine on a USB stick with preconfigured software packages. Participants will need a laptop for the hand-on activities. They should be able to install our supplied virtual machine, which will be a VirtualBox VM. For those unable to install a virtual machine, we will provide access to a cloud-based deployment of the PADS framework. However, these participants may not be able to execute their modeled system on a Mininet emulator to test their hypotheses.

V. Novelty and contributions

The PADS framework provides a distributed systems learning toolkit which is based on MDE technologies and tools. We surveyed previous tutorials that have been covered in the MODELS conference since 2006. None of the works presented in these tutorials focused on the fundamental learning domain of distributed systems algorithms. Another missing aspect that we observed was the lack of tutorials that covered benefits of MDE for teachers and students in educational settings. With this tutorial we hope to address these gaps, and motivate the teaching and research community the benefits of using MDE in learning frameworks, such as PADS, in the context of todays' highly connected and ubiquitous distributed systems presence.

VI. Prior Tutorial Experience

We have not presented a tutorial on PADS before apart from using it in our Distributed Systems course we teach at Vanderbilt University. One of the authors, Aniruddha Gokhale, has been heavily involved in the MDE community for over ten years. He has previously given a total of six tutorials on his work on the CoSMIC MDE tool [1] for deployment and configuration of component-based systems and their continuous integration at a variety of venues including MODELS as listed below.

1. "Resource-aware Deployment, Configuration and Adaptation for Fault-tolerance in Distributed Real-time Embedded Systems," Tutorial at OMG Real-time Workshop, Arlington, VA, USA, May 2010.
2. "Model-Driven Engineering for Distributed Real-time and Embedded Systems," Joint tutorial with Dr. James Hill, IEEE/ACM MODELS 2009 Conference, Denver, CO, USA, Sep 30-Oct 5, 2009.
3. "Model-driven Engineering for Continuous System Integration of Large-scale Component-based Systems," Joint tutorial with James Hill, IEEE/ACM MODELS 2008 Conference, Toulouse, France, Sep 28–Oct 3, 2008.
4. "Model-Driven Engineering for Distributed Real-time and Embedded Systems," IEEE/ACM MODELS 2007 Conference, Nashville, TN, USA, Sep 30-Oct 5, 2007.
5. "Model-Driven Engineering for Distributed Real-time and Embedded Systems," OMG Real-time and Embedded Systems Workshop, Arlington, VA, USA, July 9–12, 2007.
6. "MDE4DRE: Model-Driven Engineering for Distributed Real-time and Embedded Systems," Joint tutorial with Dr. Doug Schmidt, 13th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2007), Bellevue, WA, United States, April 3–6, 2007.

VII. Relevant Publication

The PADS framework [2] was published and presented at the IEEE COMPSAC 2016 conference. A journal version with user studies is currently under review. A pre-print version of the conference paper is available for download at: http://www.dre.vanderbilt.edu/~gokhale/WWW/papers/COMPSAC16_CELT_PADS

References

[1] Aniruddha Gokhale, Krishnakumar Balasubramanian, Jaiganesh Balasubramanian, Arvind S. Krishna, George Edwards, Gan Deng, Jeff Parsons and Douglas C. Schmidt, Model Driven Middleware: A New Paradigm for Developing and Provisioning Distributed Real-time and Embedded Applications, *Elsevier Journal of Science of Computer Programming, Special Issue on Foundations and Applications of Model Driven Architectures*, Editors Mehmut Aksit and Ivan Kurtev, vol. 73, No. 1, Sept 2008, pp. 39–58.

[2] Barve, Yogesh D., Prithviraj Patil, and Aniruddha Gokhale. "A Cloud-Based Immersive Learning Environment for Distributed Systems Algorithms." Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual. Vol. 1. IEEE, 2016