

Van Cam PHAM

SOFTWARE ENGINEER/PHD IN MODEL-BASED SOFTWARE ENGINEERING (11/2014-11/2017[ANTICIPATED])

2, Jacques Margottin, Bourg-la-reine, 92340, France

☎ (+33) 07 50 89 72 11 | ✉ phamvancam2104@gmail.com, vancam.pham@cea.fr



Education

Software Engineer/PhD at LISE (Laboratory of Model-Driven Engineering for Embedded Systems)

Saclay, France

THESIS TITLE: METHODOLOGIES FOR MODEL-CODE SYNCHRONIZATION FOR REACTIVE SYSTEM DEVELOPMENT

Nov. 2014 - Nov. 2017 [anticipated]

- **Working place:** Laboratory of Model-Driven Engineering for Embedded Systems (LISE), CEA-List, Saclay, France.
- **Brief Synopsis:** I have been working as a software engineering as well as a PhD student in the Laboratoire d'Ingénierie dirigée par les modèles pour les systèmes embarqués in the domain of MDE for reactive embedded systems. The domains that I've been working on are Model-Based Software Engineering (MBSE), component-based architecture design and software programming. MBSE focuses on using abstract diagram-based modeling languages such as UML to design complex system architecture. By using models, MBSE enables early model-based system assessment, communication between stakeholders, complexity management, and automatic code generation, which improves software productivity and quality. On the other hand, software programming has been one of the most important tasks during software development with many strong and widely used programming languages such as Java and C++. My thesis is to propose an approach for combining both of the modeling and programming practice to benefit their advantages. Specifically, I propose a synchronization approach, which keeps UML-based software architecture model and object-oriented programming code such as C++ and Java consistent in case there are modifications of the model and the code. The software structure is designed based on component-based engineering, which enables decoupling between software components and re-usability of components. The software behavior is described by using state machines, which are appropriate to manage the discrete event-driven reactive system behaviors.

U-PSUD(University of Paris-Sud)

Orsay, France

MASTER IN EMBEDDED SYSTEMS AND SIGNAL PROCESSING

Sept. 2013 - Sept. 2014

- Program: Embedded Systems and Signal Processing (SETI) cooperated by University of Paris Sud, ENS Cachan, INSTN and ENSTA ParisTech, France
- Scholarship Student of International Relationships of University of Paris-Sud
- Modules included: Real-time Control Numeric Systems, Complex Embedded System Design, Algorithm-Architecture Ad-equation, Network and Quality of Services, Multimedia Data Compression, Data Fusion, Statistic Learning and Neural Network, and Initiation to Research.

HUST(Hanoi University of Science and Technology)

Hanoi, Vietnam

ENGINEER IN INFORMATION SYSTEM AND COMMUNICATION

Sept. 2007 - Jul. 2012

- Program: Programme de Formation d'Ingénieur d'Excellence au Vietnam (PFIEV)
- Final project: Understanding pros and cons of the client-server and peer-to-peer models, and combining these two models for developing distributed applications.

Employment

Laboratory of Model-Driven Engineering for Embedded Systems (LISE), CEA-LIST

CEA, Saclay, France

SOFTWARE ENGINEER/PHD STUDENT AT LABORATORY OF MODEL-DRIVEN ENGINEERING FOR EMBEDDED SYSTEMS (LISE)

Nov. 2014 - Nov. 2017 [anticipated]

- I work as a software engineering and PhD Student, under supervision of Dr. Ansgar RADERMACHER, Dr. Sébastien GÉRARD, and Dr. Shuai LI, at the LISE laboratory, which develops the Papyrus industrial modeling tool, in the field of MBSE, especially the use the UML modeling language and the synchronization between model-model and between model-generated code. My work involves: the use of UML for modeling component-based architecture and event-driven behaviors for reactive and distributed systems in the context of the Papyrus Designer tool; the corresponding implementation of the UML-based design; the process of automatically translating the UML-based software design model into efficient executable code; and the process of automatically propagating modifications of the code back to the model. Technically, I have been working a lot with several mainstream programming languages such as Java, C++, and C and have a deep understanding of the programming languages in order to generate code from models. I also extend the standard C++ programming language to support component-based development for better managing complexity in code-centric development approaches, especially for reactive systems.

Laboratory of Model-Driven Engineering for Embedded Systems (LISE), CEA-LIST

CEA, Saclay, France

INTERNSHIP AT LABORATORY OF MODEL-DRIVEN ENGINEERING FOR EMBEDDED SYSTEMS (LISE)

Apr. 2014 - Sept. 2014

- I worked as an intern undersupervision of Dr. Ansgar RADERMACHER in the context of applying component-based modeling and design to distributed system development. I used UML and a UML profile for designing interactions between distributed components in distributed systems. The created design is then automatically translated into code, which uses the ZeroMQ middleware to exchange data between distributed components.

Toshiba Software Development Vietnam (TSDV)

Hanoi, Vietnam

SOFTWARE ENGINEER FOR COMPUTERS AND EMBEDDED SYSTEM

Jul. 2012 - Aug. 2013

- I worked as a software developer in three projects: (1) deal with requirement analysis, detailed design, implementation, unit test, integration test, and optimization at the programming level for software of an embedded system: Toshiba G2R protection relay. The development using C as the development language and Visual Studio as an integrated development environment (IDE) includes using IEC 60870-5-103 to control and communicate between smart electric device, and testing on real hardware and using debug tools to debug on hardware; (2) develop a desktop application to control electric devices by using a private protocol of Toshiba and the C# language. The application polls the devices to receive data of recorded errors, and creates a chart for analysis of errors causes; and (3) use the Qt IDE and the C++ ffmpeg library for implementing a multimedia player application, which plays both audio and video files.

Tamtay.vn Company

Hanoi, Vietnam

INTERNSHIP FOR DEVELOPING GAMES AND APPLICATIONS FOR ANDROID

Feb. 2012 - Apr. 2012

- I worked as an intern at the tamtay.vn company, that develops applications and games for the tamtay.vn social network. My responsibility is to investigate the development framework Android SDK for Android applications. I then collaborated with another intern to develop a simple multimedia player application, that downloads and plays music files and associated information transferred in JSON files. I was then responsible for investigating the AndEngine game engine to learn skills in game development.

School of Communication and Information Technology - Hanoi University of Science and Technology

Hanoi, Vietnam

INTERNSHIP FOR INVESTIGATING GOOGLE CLOUD PLATFORM

Jul. 2011 - Aug. 2011

- I worked as an intern at the School of Communication and Information Technology. I investigated the *Google App Engine* Google Cloud Platform for building scalable webs and mobile backends using Java, Eclipse IDE, TomCat, plugin of Google App Engine for Java.

Skills and Experiences

Programming	J AVA, C/C++ (especially for embedded systems), C#, Androind, Xtend (Modernized Java), JavaScript, ASM, Matlab, Python, SQL
Modeling	U ML (class, state machine, composite structure, deployment, ALF, profile, PSCS, PSSM), MARTE, Papyrus, component-based software modeling, event-driven behavior modeling for reactive systems, round-trip engineering and synchronizaton for model-model and model-code, Model-Based Software Engineering (model transformation such as QVT and ATL, code generation such as Xtend and Acceleo, IncQuery-based model query)
Framework and Tools	V isual Studio, Eclipse (plug-in development), Java Spring, Spark, CodeBlocks, QT, TomCat, Android Studio, Oracle, NetBeans, VirtualBox, IBM Rhapsody, LaTeX, Papyrus
Operating Systems	U buntu, Windows 7, 10, XP
Network experiences	P eer-to-peer vs client-server architecture, TCP/IP, IPv4, SNMP, application layer protocols, routing protocols, wireless network
Other technical skills	N eural network for statistic learning, machine learning (clustering, classification), VHDL for simulation, real-time tasks scheduling, multimedia compression, background VHDL and signal processing
Teaching	S upervise practicals for undergraduate students in Android Programming, SQL for database query, C++ and JAVA Algorithm and Programming
Languages	E nglish, French, Vietnamese
Time-project management	T his skill is important to complete my PhD during 3 years and to manage my own research and development project (that I will finish it in November of this year). Furthermore, I also met the many deadlines for submitting conference papers and in my teaching. I have learned to manage these different tasks.
Problem solving	A s a PhD student, I have learned two very important skills, which are problem identification, by rigourously analyzing the state of the art and the practice, and problem solving, by deliberately thinking and discussing various solutions to a particular problem and finding the best one, while taking in consideration other constraints such as deadlines and teaching duties.

Topics of Interests

Software programming language and application

- As a PhD student and a practitioner in software engineering, I am passionate in programming, especially the use of mainstream programming languages such as C/C++, Java, and Python for development of software applications (e.g. reactive embedded system applications) for solving everyday life problems. I'm also interested in knowing practices how to be productive and qualitative in programming.
- A major part of my thesis work is about extending, engineering, and transforming software programming language. Especially, I'm interested in extending current programming languages to support a state machine-based event-driven model for developing reactive systems.
- During PhD time, I have learned about application of event-driven architecture to asynchronous systems such as distributed systems. I'm interested in using programming model, such as state machine-based model, and network protocols to develop distributed and/or embedded applications, which will involve network technologies such as communication protocols or routing algorithms as well as the combination of peer-to-peer and client-server model in my undergraduate at HUST.

Application to Modeling and Development of Embedded and Distributed Systems

- In my thesis, I work in the context of the Papyrus Designer - an extension of the Papyrus modeling tool. Papyrus Designer provides component-based modeling and code generation for distributed embedded systems, using the concepts of interaction components to model remote interactions between distributed components. I'm interested in studying methodologies for applying Papyrus Designer with its UML profile to distributed systems, especially component-based distributed reactive systems.

Machine learning

- The field of machine learning and its application have been attracting many practitioners including myself. During my master education, I was taught about machine learning (modules data fusion and statistic learning and neural network). Since there, I have been keeping my passion and learning practically, especially using Python, the use of different learning models for understanding data and prediction.

Model-Based Software Engineering

- The main research interest of my PhD thesis are methodologies for synchronization of code and model specified in UML-based component-based modeling and UML state machines, and studying different techniques for model transformation such as QVT, ATL, Triple-Graph Grammar (TGG), and Change-driven transformation, for model synchronization such as QVT-Relation and TGG. I'm also involved in researching approaches for a mapping of a UML model containing the ALF code, and programming language code. This mapping will enable a synchronization of UML model with ALF and programming code, and eventually provide synchronization of a platform-independent model with code. Furthermore, I'm also interested in approaches for simulating a system from its models and generating fully operational code from models, especially UML models, which might contain ALF code, approaches for optimizing software system at the model level (e.g. optimization ALF code or UML state machines), and if possibly approaches for model interpretation and compilation. The final goal is harmonization of software programming and MDE practices.

References

Dr. Ansgar RADERMACHER
Ansgar.RADERMACHER@cea.fr
Laboratory: LISE
CEA-LIST, CEA, Saclay, France

Dr. Shuai LI
Shuai.LI@cea.fr
Laboratory: LISE
CEA-LIST, CEA, Saclay, France

Dr. Sebastien GERARD
Sebastien.GERARD@cea.fr
Laboratory: LISE- Papyrus development
CEA-LIST, CEA, Saclay, France

Publications

Conference Proceedings and Journal Articles

- V. C. Pham, S. Li, A. Radermacher, S. Gerard, and C. Mraidha. Fostering Software Architect and Programmer Collaboration. In *21st International Conference on Engineering of Complex Computer Systems, ICECCS 2016, Dubai, United Arab Emirates, November 6-8, 2016*, pages 3–12, a.
- V. C. Pham, A. Radermacher, and S. Gérard. From UML State Machine to Code and Back Again! In *Position Papers of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016.*, pages 283–290, b.
- V. C. Pham, A. Radermacher, S. Gérard, and S. Li. Bidirectional Mapping Between Architecture and Code for Synchronization. In *ICSA 2017 - Proceedings of the 14th International Conference on Software Architecture, Gothenburg, Sweden, 3-7 April, 2017.*, c.
- V. C. Pham, A. Radermacher, S. Gérard, and S. Li. Complete Code Generation from UML State Machine. In *MODELSWARD 2017 - Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development, Porto, Portugal, 19-21 February, 2017.*, d.
- V. C. Pham, A. Radermacher, S. Gérard, and F. Noyrit. Change Rule Execution Scheduling in Incremental Roundtrip Engineering Chain: From Model-to-Code and Back. In *MODELSWARD 2016 - Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development, Rome, Italy, 19-21 February, 2016.*, pages 225–232, e.
- V. C. Pham, Ö. Gürçan, and A. Radermacher. Interaction Components Between Components based on a Middleware. In *1st International Workshop on Model-Driven Engineering for Component-based Software Systems (ModComp'14)*, 2014.

Presentation/Courses Attended

14th ICSA (International Conference on Software Architecture)

Gothenburg, Sweeden

PRESENTER FOR CONFERENCE PAPER: BIDIRECTIONAL MAPPING BETWEEN ARCHITECTURE AND CODE FOR

Apr. 2017

SYNCHRONIZATION

- Presented an approach, which enables a bidirectional mapping between UML-based architecture model and code. This proposed mapping is part of a synchronization approach, which synchronizes code and UML-based model, using UML component-based modeling and UML state machines.

5th MODELSWARD (International Conference on Model-Driven Engineering and Software Development)

Porto, Portugal

PRESENTER FOR CONFERENCE PAPER: COMPLETE CODE GENERATION FROM UML STATE MACHINES

Feb. 2017

- Presented an approach, which enables generating efficient code from UML state machines with full UML state machine features, including pseudo states and UML events. The generated code is evaluated for its runtime execution semantics conforming to the UML specification, its performance in event processing speed, and its static and dynamic memory consumption.

21th ICECCS (International Conference on Engineering of Complex Computer Systems)

Dubai, UAE

PRESENTER FOR CONFERENCE PAPER: FOSTER SOFTWARE ARCHITECT AND PROGRAMMER COLLABORATION

Nov. 2016

- Presented a model-code synchronization methodological pattern, which allows model and code to be concurrently modified by software architects and programmers, respectively, and synchronizes the modifications made in the model and the code.

36th SEW (International Software Engineering Workshop)

Gothenburg, Sweed

PRESENTER FOR CONFERENCE PAPER: FROM UML STATE MACHINE TO CODE AND BACK

Sept. 2016

- Presented a mapping between UML state machine elements and object-oriented code. The purpose is to provide a round-trip engineering between UML state machine and code, which means that modifications in code can be reflected back to model.

6th International Summer School on Domain Specific Modelling Theory and Practice (DSM-TP)

Antwerp, Belgium

ATTENDEE AT SUMMER SCHOOL

Aug. 2015

- Attended as a doctoral student. I learned different concepts and practices of MDE in this summer school such as model transformation, domain-specific modeling language design, mapping abstract and concrete syntax as well as the theory of temporal logic.