Van Cam PHAM

PhD in Software Engineering, Lead Blockchain Researcher & Plasma Enthusiast

Hanoi, Vietnam

☎ (+84) 1693 651 831 | ✉ phamvancam2104@gmail.com

"The only place where SUCCESS comes before WORK is in the dictionary."

Employment

Tomochain.com Pte. Ltd.   *219 Trung Kinh, Hanoi, Vietnam*

Lead Blockchain Researcher at Tomochain.com   *May 2018 - Present*

- I joint Tomochain in the middle of May as a Blockchain Researcher. My work is to collaborate with blockchain engineers to design the Tomochain public blockchain to achieve the goals of Tomochain including scalability and security while maintaining the decentralization of the system at an acceptable level. Specifically, we design our blockchain consensus algorithm, Proof-of-Stake Voting and are in way of implementing it. Also, we released our technical paper on the consensus and a series of in-depth technical articles focusing on the similarities and differences between Tomochain and other major blockchain infrastructures, including EOS, Cardano, Ethereum Casper, and Tendermint. I'm in charge of writing our technical documention, including papers and blogs, and analyzing technical strengths and limitations of our solution and the solutions of other blockchains such as Ethereum and EOS.
- I'm currently developing the solution to the blockchain challenging problem: blockchain sharding, in order to improve further the performance of Tomochain. The sharding design paper is currently released and I am integrating it into TomoChain's current architecture.
- Plasma integration: My vision is that, if many DApps will be used by people, a single public blockchain will need both on-chain scaling solutions and off-chain solutions. Shading for on-chain scaling and Plasma for off-chain scaling are an idealized solution I'm developing. Each DApp can run on a Plasma side chain off the main chain and only sends settlement transactions to the main chain.

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 have been working as a software engineering and PhD Student, under supervision of Dr. Ansgar RADERMACHER, Dr. Sébastien GÉRARD, and Dr. Shuai LI, at LISE, which develops the Papyrus industrial modeling tool, especially the use of UML to design complex system software. My work involves: the use of UML for designing 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 with several mainstream programming languages such as Java, C++, and C and have a deep understanding of these programming languages in order to generate code from models. I also extended the standard C++ programming language to support component-based developemnt for better managing complexity in code-centric development approaches, especially for reactive systems. An Arm-based Lego Car factory application is developed for demonstration.

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 undersupervison 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.

Skills and Experiences

Programming	**JA**VA, C/C++ (especially for embedded systems), C#, Androind, Xtend (Modernized Java), Golang, Python, SQL
Blockchain	**Bi**tcoin, Ethereum, EOS, Zilliqa, consensus algorithms (Proof-of-Work, Proof-of-Stake, Delegated Proof-of-Stake, decentralized randomization), blockchain scalability solutions (blockchain sharding, Plasma, State Chanel), Cryptocurrency, Cryptography, Ethereum Virtual Machine (EVM), Smart Contract: Solidity, Solidity security best practices, debugging, smart contract formal security verification tools (Oyente, Securify, Mythril), EVM bytecode, smart contract hacking
Framework and Tools	**Vi**sual Studio, Eclipse (plug-in development), Android Studio, Oracle, NetBeans, VirtualBox, LaTeX, Papyrus
Networking	**Pe**er-to-peer vs client-server distributed architecture, TCP/IP vs OSI architecture, IPv4, routing protocols, MPLS (LDP & RSVP-TE)
Software architecture	**As** a PhD student, working with design and implementation for embedded software and software in general, I have learned object-oriented design patterns, component-based software architecture design and microservice architecture.
Modeling	**UM**L (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 syncrhonization 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)
Technical writing	**In**-depth technical writing, academy articles writing, software architecture writing, blockchain and cryptocurrency writing, technical literature review
Operating Systems	**Ub**untu, Windows 7, 10, XP
Time-project management	**Th**is 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 January 2018). 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	**As** 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.
Teaching	**Su**pervise practicals for undergraduate students in Android Programming, SQL for database query, C++ and JAVA in Algorithm and Programming
Languages	**En**glish, French, Vietnamese

Education ————————————————————————————————

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

THESIS TITLE: METHOTHODOLOGIES 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 worked as a software engineer as well as a PhD candidate in the Laboratoire d'Ingénierie dirigée par les mod-èles pour les systèmes embarqués in the domain design and implementation of reactive application in the context of Model-Based Software Engineering (MBSE) and component-based architecture design. MBSE focuses on using abstract diagram-based modeling languages such as UML to design complex and distributed system architecture. I develop a framework, which allows to (1) use UML to design complex system software and automatically generate C/C++ code of the high-level application behavior; (2) develop algorith-mic/computational code for applications; and (3) propagate the modifications in code back to the UML design to update the design and to keep it consistent with the code. Code productivity is significantly gained by the ability to automatically produce code from design. Furthermore, C/C++ code is efficient and many bugs in code are eliminated because of the use of automatic code generation. That is to say that, the approach benefits advantages of both of MBSE such as complexity management, productivity and communi-cation by UML, and traditional software programming, which is one of the most important tasks during software development with many strong and widely used programming languages such as Java and C++.

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'Ingenieur d'Excellence au Vietnam (PFIEV) - Programming of Training for Excellent Engineers in Vietnam
- Final project: Understanding pros and cons of the client-server and peer-to-peer models, and combining these two models for developing distributed applications.

Topics of Interests

Blockchain infrastructure, scalability and applications

- Blockchain infrastructure: I have had my deep knowledge of different blockchain infrastructures and understand the current problems of the mainstream infrastructure including Bitcoin, Ethereum, EOS, just to mention a few. My interest is devoted to blockchain scalability and security problems. Specifically, on on-chain scaling solutions, I have been developing my sharding solution to Tomochain and Ethereum as well and clearly understanding the sharding of some emerging platforms such as Zilliqa and Quarkchain. I have released a sharding architecture for TomoChain and currently is integrating it into TomoChain. The sharding design has some uniqueness such as an *incentive-driven verification game* or several cross-shard transaction schemes. On off-chain scaling solutions, I have developed deep knowledge on Plasma and State Channel, as well as Lightning Network of Bitcoin. My vision is to have a main scalable blockchain on which each DApp can run on a Plasma side chain, which can significantly increase scalability while still maintaining security properties. As a Plasma enthusiast, I look forward to collaborating with researchers and developers who have special interest in Plasma.
- Ethereum smart contract: I have also developed my deep knowledge in Ethereum Virtual Machine, EVM opcodes and currently been involving smart contract security best practices and security formal verification tools such as Mythril and Securify. Hacking smart contracts is also my favorite game

Software programming language, software architecture design and implementation

- 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 at 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 assist developers to be productive and qualitative in software development.
- 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 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 as in my undergraduate at HUST.

Research and Development of Embedded and Distributed Systems

- In my thesis, I work in the context of development of embedded software systems, in which I use UML and design patterns for designing such systems (or software in general). I'm therefore interested in diving into this domain as deep as possible. Furthermore, I want to combine my knowledge about the network technologies that I have collected to produce distributed embedded software systems. The latter, in my thought, are very suitable to the development (design and coding) of IoT applications, in which devices collect, process, and exchange data with each other, and which might be combined with machine learning field. Using UML, from my point of view, would be very helpful for the complexity management task of such systems.

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.

Leisure

Football watching, tourism, book reading (especially historical + technical books, or sometimes literature), chatting

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

Presentations/Courses Attended

Dezentral Conference at Berlin Blockchain Week
Berlin, Germany

PRESENTER FOR TOMOCHAIN'S CONSENSUS PROTOCOL FOR SCALABILITY AND ON-CHAIN GOVERNANCE *Sept. 2018*

- Presented the Proof-of-Stake Voting consensus protocol proposed by TomoChain, which solves the blockchain scalability and on-chain governance problem.

GroundX Conference at Jeju island
Berlin, Germany

PRESENTER FOR TOMOCHAIN'S CONSENSUS PROTOCOL FOR SCALABILITY AND ON-CHAIN GOVERNANCE *Jul. 2018*

- Presented the Proof-of-Stake Voting consensus protocol proposed by TomoChain and joined the panel discussion about blockchain consensus and security.

14th ICSA (International Conference on Software Architecture)
Gothenburg, Sweeden

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

SYNCHRONIZATION
Apr. 2017

- 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.