New Research Focuses on Communicating the Value of Architecting within Agile Development

Every year, the SEI funds a select handful of Independent Research and Development (IRAD) projects, which can be used to determine whether a project should become a research priority for the SEI.

In the coming year, Nanette Brown, Robert Nord, and Ipek Ozkaya—all researchers within the SEI's Research, Technology, and System Solutions program—will begin an IRAD research project to determine the amount of architecture needed to support agility in the context of large-scale projects. The team is also collaborating with Philippe Kruchten, currently a professor of software engineering at the University of British Columbia.

The impetus for this work grew out of the observation that while agile software development methods are appealing to practitioners and are getting a lot of attention in industry and DoD, organizations are facing difficulties scaling these methods up. One of the key issues the team has identified in applying agile techniques in large settings is the lack of attention to architectural design and development. Such inattention is often based on the misconception that the goal of architecture practices is to produce documents that sit on a shelf.

"Understanding how architecture-related tasks and features appear in iteration and release planning is an essential aspect of finding the right balance," explains Ozkaya, who is leading the research team. "Determining the amount of architecture needed to support agility is at the heart of this research initiative. And while the separation of architecture, code, and features might aid in project management, the reality is that they are not clearly separable."

Architecture can focus planning, backlog management, and refactoring for large-scale projects. It can help prioritize features that are architecturally significant. However, this requires eliciting architecture-related features, assigning value and cost to them, and having mechanisms that treat customer-visible features simultaneously with architectural features based on their dependencies.

One measure of success that the team has defined for itself is that it will be able to provide guidance and case studies demonstrating how to incorporate architecture more effectively into the large-scale project management in an agile context.

Ozkaya says that throughout the year-long project, the team will post frequent status updates on the SATURN Network blog at saturnnetwork.wordpress.com

"Being overly protective of ideas can lead to overthinking and overanalyzing, common pitfalls in any project, and contrary to agility," says Ozkaya. "Agility requires open communication, frequent feedback, courage, and simplicity."

SEI Members from left, Vanessa Mendoza of TEDT-WDS-PA White Sands Missile Range, Byron R. Frank of Westinghouse Electric Company, Sheela Raman of Tieto Technology Park, and Karen Lafond of PEO Integration.

Go paperless! If you would like to receive an email containing a PDF version of *The Monitor* instead of a printed version, email membership@sei.cmu.edu.

Save the Date
SEI Member Awards & Luncheon
March 23, 2010
More Details to Come!

SEI Membership
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
412-268-5800
Toll-free: 888-201-4479
Email: **membership@sei.cmu.edu**
www.sei.cmu.edu/membership

System Architecture and Virtual Integration

Aircraft manufacturers face a big problem, and they know what is causing it. The complexity of the software in commercial aircraft has increased tremendously—as has software size and cost. In the last two decades, for example, the amount of onboard software, measured in source lines of code (SLOC), has doubled every four years. And the projection is that the size will hit 27 million SLOC in the next decade, with a price tag of $10 billion.

With size and complexity, unfortunately, are likely to come more errors that will strain an already creaky traditional software development approach to the breaking point. Under this approach, 80 percent of the problems introduced are not detected until the integration-testing development phase or even in operation, where they are much more costly to correct.

SEI Technical Report: System Architecture Virtual Integration: An Industrial Case Study
www.sei.cmu.edu/library/abstracts/reports/09tr017.cfm

"There is a smarter way," said Peter Feiler, a senior member of the technical staff at the SEI who is leading its participation in the Aerospace Vehicle Systems Institute (AVSI), a global cooperative of aircraft manufacturers, government organizations, and academic institutions. This industry-wide initiative aims to change the way software is developed and acquired, focusing on architectural models that include aircraft virtual integration.

"Integrate and then build is a new kind of approach and new technology that allows you to catch things earlier in the software development life cycle," explained Feiler.

This new architecture-centric model approach, known as the System Architecture Virtual Integration (SAVI), supports virtually integrated system models with respect to multiple operational quality attributes including performance, safety, and reliability.

There are four phases to the initiative and the first, a proof of concept, was just completed in 2009. In the proof of concept, an aircraft system was modeled using the industry standard Architecture Analysis and Design Language (AADL). Feiler serves as the standard's technical lead and was instrumental in its development. An independent study commissioned by AVSI identified AADL as the best technology match for SAVI.

"We just finished the first proof of concept showing that the technology is mature enough that it can make a difference," Feiler explained, adding that work on the second phase began November 1. "In the next phase, SAVI will ask tool vendors to work with them to contribute to a tool infrastructure to the SAVI approach. I call it architecture-centric virtual integration."

The first phase of the SAVI initiative was carried out by representatives of Boeing, Airbus, Lockheed Martin, Rockwell Collins, BAE Systems, GE Aviation, U.S. Army, and the Federal Aviation Administration.

The initiative will run through 2014 and by that time, aircraft manufacturers will have spent $80 million to upgrade to the new practice of modeling using aircraft virtual integration earlier in the development process.

That investment will be spent on internal development processes including incorporating the new virtual integration technology, changing certification processes, and developing workshops that focus on interacting with standards groups. The money will also fund additional pilots.

For the initial phase, the SEI served as part of the planning team. For the second phase, they will take a more involved role including writing the business case.

"The SEI's role is to turn that technology into a practice," Feiler explained.

Multicore and Mixed Criticality

Imagine two members of a family, both approaching an ATM in different locations at exactly the same time; one to withdraw money from an account, one to make a deposit into that same account.

The person making the withdrawal may be delayed because the deposit-acceptance process had not yet completed when the withdrawal-request process needed to access the balance information is started. This delay would also be suffered by the other clients waiting to use the ATM where the withdrawal is being done.

"If we know two programs will need shared data, we can deploy them in the same computer with multicore processing and avoid delays on unrelated programs," explained Dionisio de Niz, a senior member of the technical staff at the SEI. "For instance, in the case of the ATM example, making both the deposit and withdrawal to the same account in the same ATM avoids imposing delays on unrelated clients in more than one ATM."

A multicore processor combines two or more independent cores (normally CPUs) into a single, integrated circuit. The increasing availability of processors with many computing cores requires better approaches to developing and deploying concurrent software.

"Real-time programs are programs that need to finish their execution by a specific time," explains de Niz. "A common example is the program that inflates the airbag of a car in a crash. In this case, the program needs to detect the crash and inflate the airbag in 20 milliseconds (ms) or less to prevent the driver from hitting the steering wheel. If I put a collection of real-time programs in a single processor, I need to make sure that all of them finish their execution as required by their situation (e.g., 20 ms for the airbag). This requirement is known as its deadline. When I run a collection of real-time programs in the same processor, the part of the operating system known as the scheduler decides the order in which these programs run. Such an order has the

Continued on Page 4

SATURN 2010

May 17-21, 2010 • Minneapolis, MN

Since 2005, the annual SATURN Conference has been the gathering place for the SEI Architecture Technology User Network (SATURN)—a community of software and systems engineers, enterprise architects, architecture practitioners, and researchers who share ideas about new and current architecture technologies and practices.

The theme of the SATURN 2010 Conference, Architecting for Change, emphasizes the need for organizations to meet their business goals in rapidly and continuously changing environments. The conference will present experience reports and discussions of architecture methods, techniques, and practices that enable organizations to adapt to persistent change.

For SATURN 2010, which will be held May 17-21 in Minneapolis, Minn., the SEI is partnering with *IEEE Software* magazine for the first time to bring SATURN attendees expanded conference offerings including

- presentations and talks by *IEEE Software* board members
- opportunities for networking with *IEEE Software* board members
- opportunities for publication in IEEE periodicals including *Computing Now* and *IEEE Software*

Through IEEE Software, the IEEE Computer Society will participate on the SATURN 2010 advisory board. Also, Hakan Erdogmus, editor-in-chief of the magazine and Frances Paulish, chair of the advisory board, joined the technical committee for SATURN 2010.

During the conference, two presenters will receive the IEEE Software Presenter awards. In addition, select papers will be featured in a future issue of *IEEE Software*. Magazine staff will establish a mentoring process to help deliver the experiences of the selected presentations to the broader practitioner community.

Also new this year is the SATURN blog, www.saturnnetwork.wordpress.com, which was launched in May 2009 as a channel for posting information about the 2009 SATURN Conference. It now provides a forum for practitioners to share ideas and network online. Topics covered in the blog include

- architecture and Agile Development
- architecture competence
- architecture-centric engineering and practices
- quality attribute analysis
- service-oriented architecture
- ultra-large-scale systems
- For more information

SATURN 2010: www.sei.cmu.edu/saturn/2010/

SATURN: www.sei.cmu.edu/saturn/

IEEE Software: www.computer.org/portal/web/software/home

SEI Special Report : Proceedings of the Workshop on Software Engineering Foundations for End-User Programming (SEEUP 2009)
www.sei.cmu.edu/library/abstracts/reports/09sr015.cfm

Educational Opportunities in Architecture-Centric Engineering

● Software Architecture Professional Certificate Program*
● ATAM Evaluator Certificate Program*
● SEI-Certified ATAM Leader Certification Program* (also requires an ATAM observation)
* Requires a passing score on the Software Architecture: Principles and Practices exam

Software Architecture: Principles and Practices	Documenting Software Architectures	Software Architecture Design and Analysis	Software Product Lines	ATAM Evaluator Training	ATAM Leader Training	Modeling System Architectures using the Architecture Analysis & Design Language (AADL)
● ● ●	● ●	● ●	● ●	●	●	
Gain a better understanding of the relationships between system qualities and software architectures; learn about software architecture evaluation, attribute-driven design, software architecture documentation, and architectural reuse. Find out more at www.sei.cmu.edu/training/p35.cfm (classroom format) or www.sei.cmu.edu/training/V07.cfm (eLearning format)	Learn how to choose the set of views that will be most valuable to the architecture's community of stakeholders and to use formal and informal notations (including Unified Modeling Language [UML]) Find our more at www.sei.cmu.edu/training/p33.cfm	Get an in-depth understanding of the concepts needed to effectively design and analyze a software architecture. It is based on the books *Software Architecture in Practice, Second Edition* and *Evaluating Software Architectures: Methods and Case Studies.* Find out more at www.sei.cmu.edu/training/p34.cfm	Learn about product line practice patterns, a roadmap for software product line adoption, and a product line diagnostic method; determine which product line practice patterns best apply to their organizations; and form a path to software product line adoption. Find our more at www.sei.cmu.edu/training/p36.cfm	Learn the SEI Architecture Tradeoff Analysis Method (ATAM) and how to apply it to evaluate software architectures. A significant portion of the course is dedicated to participants performing an ATAM evaluation exercise with guidance from instructors. Find out more at www.sei.cmu.edu/training/p31.cfm	Gain a better understanding of how to apply proven meeting management and facilitation techniques during an ATAM evaluation, how to manage the roles of ATAM participants, and how to listen for architectural risks and capture them faithfully. Find out more at www.sei.cmu.edu/training/p32.cfm	Learn about the value of model-based engineering for system development in their application domains, core elements of the AADL modeling language, and the quantitative validation of quality attributes through analysis of system architecture. Find out more at www.sei.cmu.edu/training/p72.cfm

Continued from Page 2

double purpose of ensuring that all the programs finish by their deadlines while maximizing the work performed by the programs on the same processor. If the collection of programs we need to run does not fit a single processor, then we need to decide how many additional processors we would need to use to run all of them. This is a problem known as bin-packing and typically has the objective of minimizing the number of processors needed to run the collection of programs."

The use of this approach for systems with real-time requirements (the ATM is a simple, common example) is at the center of de Niz's research. Part of this research also looks at the use of multicore processors in real-time systems with mixed criticality. As an example, de Niz points to the increasing amounts of software being found in cars. "Even though the number of new features (e.g., GPS, DVD, games, voice recognition) increases," explains de Niz, "the cost of the new cars need to keep decreasing or stay the same. This forces some features to be run on a common processor. If these features have different levels of criticality, then we would have a mixed-criticality system. For instance, the anti-lock brake system of a car is more critical for safety than the DVD system. If for some reason the computer cannot execute both programs at the same time, it should give all the attention to the brakes. However, having this kind of mixture is also an advantage because we can completely stop the DVD to give full attention to the brakes without getting into a life-threatening situation. In contrast, if you put both the airbags and the brakes in the same processor, they are both critical and you shouldn't have a delay in either one of them. This means that if for some reason you cannot run both of them, you cannot stop either of them without getting into a life-threatening situation."

Multicore work and mixed criticality are also important for use in radar systems. As radar systems track approaching aircraft, they classify them as friendly or non-friendly. As more and more aircraft enter the radar system, the system may have difficulty keeping up.

"The radar system will not be able to keep up, so you tell it 'the critical aircraft to track are the hostile aircraft; track them,'" de Niz explained.

Through his research, de Niz has found that if critical and non-critical tasks are bundled in the same core, it is possible to completely stop the non-critical task to give attention to the critical one.

His team is also beginning the second year of a two-year independent research and development (IRAD) project. In the first year, they looked at the best options for scheduling for multicore processors. In the upcoming year, their research will examine how to modify multicore processors to get better results for real-time systems and coordinating within the operating-system scheduler.

SEI Webinar Series:
SoS Architecture Evaluation and Quality Attribute Specifications
by Mike Gagliardi
www.sei.cmu.edu/events/webinars/index.cfm

Evaluating System of System Architectures through Mission Thread Workshops

To be successful, any system of systems (SoS) has to deliver its capabilities, and it must also satisfy its quality attributes—behaviors such as performance, reliability, security, safety, and modifiability. A premise of SEI work is that capabilities and quality attributes drive the architecture of the SoS, according to Mike Gagliardi, a senior member of the technical staff.

It's not hard to see why. Severe integration and operational problems can arise from inconsistencies, ambiguities, and omissions in addressing the quality attributes of SoS architectures, Gagliardi explained.

Gagliardi and his team have devised a method called the Mission Thread Workshop (MTW) to address inconsistent, ambiguous, or incomplete information about quality attributes for an SoS. The purpose of MTWs is to augment SoS end-to-end mission threads with quality attributes considerations that shape the SoS architecture and to identify SoS architectural challenges early in the life cycle. The mission thread augmentation is performed with inputs from key stakeholders. An end-to-end mission thread (in a DoD context) is a sequence of activities and events beginning with an opportunity to detect a threat or element that ought to be attacked and ending with a commander's assessment of damage after an attack.

In the 18 months since the method was developed, the MTW has been applied about 20 times with a variety of Department of Defense programs. Gagliardi's team currently works with the U.S. Navy, employing the MTW to augment the SoS end-to-end mission threads to be used for architecture development and downstream architecture evaluation activities.

"A system of systems has to satisfy multiple mission threads," Gagliardi says. "You have to augment the threads with quality attribute considerations—because that is what will drive the architecture."

An MTW brings together key stakeholders representing a variety of organizations, roles, and points of view to discuss the quality attributes needed for each step in the mission thread. "In an MTW, we find more than just quality attribute issues. We find capability mismatches and engineering issues as well, and it serves as a great communication vehicle for the architects and stakeholders to get on the same page," explained Gagliardi.

Information developed from an MTW is then made available to the SoS architecture development, prototyping, modeling and simulation, integration, and test activities. The information is also reduced to a set of architectural challenges to inform SoS, system and software architecture development, and acquisition activities. Constituent legacy system/software architectures can also be evaluated using the augmented SoS end-to-end mission threads to identify any architectural mismatches between the SoS and the candidate legacy systems, early in the SoS life cycle.

The Mission Thread Workshop can be also applied to commercial end-to-end capability threads, business process threads, and development threads.

For more information about the Mission Thread Workshop, visit **www.sei.cmu.edu/architecture/tools/evaluatingsos/mission.cfm**

Member Profile

James "J.D." Baker

Member Since April 2006
jdbaker@computer.org

James "J.D." Baker's work in the field of systems engineering touches upon many areas.

Baker, a principal systems engineer at Armstrong Process Group, has worked or conducted research in the fields of enterprise architecture, requirements development, software and systems architecture, and systems design processes and methodologies that support model-based systems engineering.

But the story does not end there.

Baker also serves as an elected Architecture Board member at the Object Management Group (OMG), an international computer industry consortium that works to develop enterprise integration standards for a wide range of technologies, and an even wider range of industries. According to Baker, OMG is best known for its work developing the standard known as Common Object Request Broker Architecture (CORBA)—which supports multiple platforms by enabling software components written in multiple computer languages and running on multiple computers to work together—and the Unified Modeling Language (UML), for the analysis and design of software-intensive systems.

> *"I like the fact that SATURN focuses not necessarily on SEI presentations but on industry presentations from people who have experience in the world of software architecture or system architecture."*

"The UML tool is a language of choice that can be used to model software-intensive systems," Baker explained, adding that a second released version of UML provides a mechanism for formal and informal extensions. Through his involvement with OMG, Baker served as one of the authors for the Systems Modeling Language (SysML), an extension of UML for systems engineering and software architecture. He describes it as one of many highlights in his career.

"The thing that is most exciting to me is the notion of taking systems and software and even enterprise architecting and creating both a methodology and a notation that allows us to better interact, to create things that are understandable across multiple disciplines," explained Baker, who lives in San Diego, California. Currently, he is working on creating a description for the Object-Oriented Systems Engineering (OOSEM) methodology and trying to connect it to one or more software development methodologies and then examining it from an enterprise architecture approach.

Baker has been involved with OMG for nine years. For the last two, he has also participated in the Model-Based Systems Engineering working group through the International Council on Systems Engineering (INCOSE).

"Those two standards and industry-based organizations are working together to produce specifications for modeling, designing, and architectecting software-intensive systems as well as OMG-developed standards and specifications," Baker said. "There are a number of us who are working in both groups. We use the expertise in INCOSE to inform the specifications being developed within OMG." Baker has been invited to present his work on OOSEM at OMG meetings and to the Architecture Working Group at the INCOSE International Workshop in February 2010.

This is all in addition to his day job. Since October 2009, Baker has worked as a principal systems engineer at Armstrong Process Group (APG).

"We provide training in enterprise architecture and system and software architecture. My basic role is that I'm typically on-site with customers helping them solve problems," Baker explained.

Before he worked for APG, Baker held a similar role at BAE Systems Inc., where he was a senior principal systems engineer responsible for architecture and requirements development and system design processes and methodologies. During his 14-year tenure, he was recognized for innovations in presenting and studying the application of software design patterns.

Since March 1997, he has taught and served as a faculty advisor for the University of California San Diego's Extension Information Technology and Engineering and Defense Technology departments. Baker said the courses he teaches span the entire system and software development lifecycle.

Baker first became involved with the SEI three years ago, when he attended classes to earn his SEI Software Architecture Professional and ATAM Evaluator certificates. He then developed a series of software architecture courses specifically tailored for BAE.

"The courses were intended to take software developers and lead them through the process of becoming software architects," Baker explained.

Through his SEI coursework, Baker also learned about the SEI Architecture Technology User Network (SATURN) Conference. In one of Baker's classes, Software Architecture: Principles and Practices, the instructor encouraged students to attend. Baker attended and presented at his first SATURN conference in 2008.

"I like the fact that SATURN focuses not necessarily on SEI presentations but on industry presentations from people who have experience in the world of software architecture or system architecture, who present things that work and who are willing to talk about the things that don't work," said Baker, adding that he also enjoyed being able to sit in on presentations and network with attendees, presenters, and sponsors. "One of the things that attracted me to SATURN 2009 and will continue to attract me is the intentional expansion to multiple genres of architecture."

SEI Membership | Carnegie Mellon

Customer Relations
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

SATURN 2010 | ARCHITECTING FOR CHANGE

SEI MEMBERS: SAVE 15 % | WWW.SEI.CMU.EDU/SATURN/2010/

Think about it: Over the lifetime of a system, requirements for it change; organizations that own and use it undergo changes; technologies and technology platforms used with it change; infrastructure supporting it changes; and people and personnel who maintain and manage it change.

Within just the past few years, we have seen a growing emphasis on new approaches, technologies, and concerns that were not considered important before such as

- cloud computing platforms: Google and Amazon, among other companies, have moved to new business models.
- the emergence of multicore processors: More and more organizations are focusing on adapting to this powerful platform.
- the need for cost-effective data management: Especially in the context of health care, this need is driving organizations to undertake data-modernization projects.
- the increasing cost of energy: Software engineers are seeking ways to reduce consumption and production costs in the applications they develop.

Organizations need their systems to be able to adapt to change while continuing to meet business and mission goals. Yet, organizations must be able to respond to change with agility without compromising quality. Maintaining a balance between architecting for change proactively and responding to change opportunistically continues to be a challenge.

SEI Members Save on Registration: Take advantage of a Members-only discount, and save an additional 15% off registration fees.

Registration Opens January 18, 2010.

For more information, www.sei.cmu.edu/saturn/2010/

Software Engineering Institute | Carnegie Mellon

May 17-21, 2010 • Minneapolis, MN

Sixth Annual SEI Architecture Technology User Network Conference