

Component-Based Design and Development for Robust Medical Applications

M. Brian Blake

Department of Computer Science, Georgetown University, Washington, DC, USA

blakeb@cs.georgetown.edu

Kevin Cleary, Hee Su Kim, and Sohan Ranjan

Imaging Sciences and Information Systems Center, Georgetown University. Washington, DC, USA

{cleary, hkim, ranjan}@isis.imac.georgetown.edu

Kevin Gary

Division of Computing Studies, Arizona State University, Mesa, Arizona, USA

kgary@asu.edu

Julien Jomier and Stephen Aylward

Department of Radiology, University of North Carolina-Chapel Hill, Raleigh, NC

{jjomier, aylward}@unc.edu

Luis Ibanez

Kitware Incorporated Clifton Park, New York, USA

luis.ibanez@kitware.com

Lead author biographies:

M. Brian Blake received his PhD in Information and Software Engineering from George Mason University and currently is on the faculty of Computer Science at Georgetown University. His research interests lie in intelligent systems and workflow management, component-based software engineering and software engineering education. Currently, he is investigating the development of software process and procedures in the development of reusable medical components.

Kevin Cleary is a Research Associate Professor and Deputy Director of the Imaging Science and Information Systems Center in the Department of Radiology at Georgetown University Medical Center. He is an expert in the field of medical robotics and specializes in the development of technology for image-guided and minimally invasive medical procedures. He is the principal investigator for an NIH-funded project to develop an open source software toolkit for image-guided procedures.

Component-Based Design and Development for Robust Medical Applications

1. INTRODUCTION

Many times, software applications for medical devices are designed and developed by medical specialists as opposed to experienced software engineers. Although it not uncommon for medical specialists to be effective programmers, in some cases, they lack the software design expertise that has been developed in the software engineering discipline over the last 40 years. Typically, shortcomings in the software analysis and design phases result in rigid software that is not resilient to change, and ultimately reduce the operational life of the underlying applications. The Imaging Science and Information Systems (ISIS) Center at Georgetown University has teamed with Kitware, Inc. to develop an open source, component-based infrastructure to support the area of image-guided surgery (Image-Guided Surgery Toolkit (IGSTK)) [1][2][3]. In collaborating with software engineering faculty members from Arizona State University, Georgetown University, and the University of North Carolina, several interesting software engineering challenges and potential resolutions have been determined in designing and developing software applications in the medical domain. In particular, this alliance of medical specialists and software engineers has investigated the customization of software engineering practices for medical applications-based software design processes and implementation practices. Section 2 and 3 of this position paper discuss the challenges, research needs, and next mid-term steps with respect to the workshop topics, *Foundations for Integration of Medical Device Systems* and *High Confidence Medical Device Software Development and Assurance*. These discussions reflect upon the observations made in this unique environment.

2. Future Software Integration for Medical Device Systems and Models

2.1 Challenges when Developing Medical Software that Scales

When developing component-based medical applications in the IGSTK environment, an overarching requirement addressed is designing software specific enough to meet a particular need, but general enough to be reusable across similar applications. This concept of software generality seemed counter-intuitive to medical specialists, who felt that more specialized code would be more reliable and easier to verify. Let's consider an example. In the field of image-guided surgery, the tracking of medical instruments is used to provide image overlay on pre-operative CT or MRI images to help guide the physician in precision instrument placement. This requires tracking software to track a device or tool as it is positioned in a patient's body. The general requirements for this software extend across multiple tracking devices, but there are specific commercial embodiments of trackers, each typically requiring some customization of the code. By employing proper software analysis and design principles, the resulting robust tracking software would require minimal customization changes in order to be applied to any number of tracker devices. As shown in this example, it is extremely important to effectively scope medical software modules. Functionality that is too specific should be isolated from functions that meet the requirements for reuse.

Another challenge in designing software for medical devices is *interface design*. Although properly partitioned software is reusable, without effective interfaces, these modules cannot be plugged into other software systems. As the medical profession progresses, medical software devices will require integration. In the current environment, proprietary devices and systems are still dominant, and it is not possible to integrate different medical devices to provide a common flow of information. In particular, timing issues, error-handling protocols, and atomicity all represent issues that must be resolved before reliability can be placed on systems of systems in the medical domain.

Software applications must essentially speak the same language before they can be properly integrated. In making medical decisions the context of information is extremely important. The semantic meaning of information passed between independent software-controlled medical devices can control the *state* of the entire application.

2.2 Research Addressing the Integration of Medical Applications

There is large gap between specific medical application needs and the ability to define reusable component-based functions. In the IGSTK project, we have determined a need for *human-driven and/or semi-automated processes to assist in the definition of modules or components*. Although the authors have made incremental progress in this area [1] with regards to requirements definition, there still remains the need for more standard approaches across the full software design and development lifecycle.

To standardize interfaces between medical devices, new research should investigate the use of universal data representation languages for information exchange. General, well-understood medical software should also consist of *standardized information exchange formats*, perhaps leveraging languages such as the Extensible Markup Language (XML) [5].

2.3 Next Steps for Integrative Software

In addressing research needs to integrate medical applications, new special interest groups should be formed to intersect proven software engineering processes, such as the Rational Unified Process and Agile Software Development [4][6] with applications specific to the medical domain. Such specialized forums and working groups will help to discover customization of these well-established approaches that integrate well with the idiosyncrasies specific to the medical domain.

To assist in integrating similar medical applications across organizations, new infrastructures should be established to promote the sharing of development experience, results, and software implementations. The authors have adopted various open-source development practices that have proven to be beneficial in promoting software interoperability with other research groups. Our practices build on the approaches used to develop the Insight Toolkit (ITK) and Visualization Toolkit (VTK) [7][7]. Larger-scaled endeavors towards this end would promote software interoperability not just in one domain but across multiple medical disciplines. The importance of testing, validation, and robustness would be magnified when put in the context of specific technology-assisted applications.

Finally, there should be support for research that explores the discovery and categorization of information passed between medical applications. Such a data management effort would be a first step towards the standardization of medical information and the establishment of standard formats and representations. There are some medical information standards such as DICOM for medical image file formats and HL7 for patient data, but there are no standards for medical device integration.

3. High-Confidence Medical Device Software: Design-Time and Operations-Time

3.1 Challenges to Reliable Medical Software and Procedures

It is important to ensure that, for every step in a medical intervention, the supporting medical software reflects a valid state. At design time, standard medical procedures should be modeled in such a way that the use of medical software is understood and documented. This may require development of workflows and use cases for medical procedures, and some efforts have been started in this direction. Later, pre-operative models should be used in real-time to confirm the necessary steps during the medical procedure. Operative monitoring approaches of this sort should also incorporate support for error-handling and fault tolerance. With solutions to these challenges, the effectiveness of medical processes can be evaluated and generally enhanced for common procedures.

3.2 IT Research Needs Addressing Reliability

The overarching research need is toward software assurance. There is a need for general *state-based control standards, mechanisms, and diagnostic tools* to support medical software. These approaches will help connect the actions in a medical operation with the corresponding actions of the supporting medical software. In addition, there should be *general validation approaches and supporting real-time diagnostic tools* to support pre-operative and post-operative procedures.

3.3 Next Steps

As a next step, *general validation models should be investigated* to validate medical processes during medical procedures. The validation models should be vetted across multiple medical disciplines. Building on the earlier open source development suggestions, state-based control components should be investigated and shared. By creating an open forum to discuss concepts and implementation for software validation, standard approaches will naturally evolve.

4. Conclusions

Within the IGSTK project, we have identified a number of important issues surrounding the areas of software design and implementations for software reliability. Based on our observations, we believe that new research should help to unite research organizations while standardizing approaches of information exchange and data management. In addition, new principled approaches should be established to help medical specialists validate processes before, during, and after medical procedures. Designing high confidence software for medical devices is a challenge that will require both software engineers and domain experts to work together to improve patient care in the 21st century.

5. Acknowledgements

This research is supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) at the National Institute of Health (NIH) under grant R41 EB000374-01A1 and by U.S. Army grants DAMD17-99-1-9022 and W81XWH-04-1-0078. The content of this manuscript does not necessarily reflect the position or policy of the U.S. Government. We also acknowledge the fruitful conversations with Will Schroeder of Kitware Incorporated.

References

- [1] Blake, M.B., Cleary, K., Ibanez, L., Ranjan, S. and Gary, K. " Use Case-Driven Component Specification: A Medical Applications Perspective to Product Line Development", *Proceedings of the ACM Symposium on Applied Computing (SAC'05)*, Software Engineering Track, Sante Fe, New Mexico, March 2005
- [2] Cleary, K., Clifford, M., Stoianovici, D., Freedman, M.T., Mun, S.K., Watson, V. "Technology improvements for image-guided and minimally invasive spine procedures" *IEEE Transactions on Information Technology in Biomedicine* Vol. 6, No. 4, pp 249-261, 2002
- [3] Cleary, K., Ibanez, L., Ranjan, S.R., and Blake, M.B. "IGSTK: A Software Toolkit for Image-Guided Surgery Applications", *Proceedings of the 18th International Conference on Computer-Assisted Radiology (CARS2004)*, Chicago, IL, June 2004, Elsevier, 473-479.
- [4] Cockburn, A. *Agile Software Development*. Addison-Wesley 2002
- [5] Extensible Markup Language (XML) (2005): <http://www.w3.org/XML/>
- [6] Kruchten, P. (2000). *The Rational Unified Process—An Introduction*, Second Edition, Addison-Wesley
- [7] Ibanez, L., Schroder, W., Lydia, N., and Cates, J. *The ITK Software Guide*, Kitware, Inc. Publishers, ISBN 1-930934-10-6
- [8] Schroeder, W., Martin, K., and Lorensen, B. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, Kitware, Inc. Publishers