Studies in Health Technology and Informatics, vol. 85
(*Proceedings of Medicine Meets Virtual Reality 02/10.* J. D. Westwood, *et al., eds.*),
Amsterdam: IOS Press, 2002, pp. 586-592.

Engineering and Algorithm Design for an Image Processing API: A Technical Report on ITK - the Insight Toolkit

Terry S. Yoo[1], Michael J. Ackerman[1], William E. Lorensen[2], Will Schroeder[3], Vikram Chalana[4], Stephen Aylward[5], Dimitris Metaxas[6], Ross Whitaker[7]

[1]*National Library of Medicine, National Institutes of Health, Bethesda, MD 20894*
[2] *General Electric Corporate Research and Development, Niskayuna, NY 12309*
[3] *Kitware, Inc. Clifton Park, NY 12065*
[4] *Insightful, Inc, Seattle WA 98109*
[5] *Dept. of Radiology, Univ. of North Carolina at Chapel Hill, Chapel Hill, NC 27599*
[6] *Computer and Information Science Dept, Univ. of Pennsylvania, Philadelphia, PA 19104*
[7] *School of Computing, Univ. of Utah, Salt Lake City, UT 84112*

Abstract. We present the detailed planning and execution of the Insight Toolkit (ITK), an application programmers interface (API) for the segmentation and registration of medical image data. This public resource has been developed through the NLM Visible Human Project, and is in beta test as an open-source software offering under cost-free licensing. The toolkit concentrates on 3D medical data segmentation and registration algorithms, multimodal and multiresolution capabilities, and portable platform independent support for Windows, Linux/Unix systems. This toolkit was built using current practices in software engineering. Specifically, we embraced the concept of generic programming during the development of these tools, working extensively with C++ templates and the freedom and flexibility they allow. Additional software development tools for distributed consortium-based code development have been created and are available from the project. In this paper, we discuss our assumptions, design decisions, and some of the lessons we have learned.

1. Introduction

The National Library of Medicine (NLM), in partnership with the National Institute for Dental and Craniofacial Research (NIDCR), the National Eye Institute (NEI), the National Science Foundation (NSF), the National Institute for Neurological Disorders and Stroke (NINDS), the National Institute on Deafness and other Communication Disorders (NIDCD), and the National Cancer Institute (NCI), has founded the Insight Software Consortium to support the creation of a public resource in high-dimension data processing tools. The initial emphasis of this effort is to provide public software tools in 3D segmentation and deformable and rigid registration, capable of analyzing the head-and-neck anatomy of the Visible Human Project data. The eventual goal is for the consortium to provide the cornerstone of a self-sustaining software community in 3D, 4D and higher dimensional data analysis. The consortium is committed to open-source code, public software including open interfaces supporting connections to a broad range of visualization and graphic user interface platforms.

The Insight Software Research Consortium including partners in academia and in industry has been formed to carry this work forward. The prime contractors are: General Electric Corporate R&D, Kitware, Inc., Insightful, Inc., the University of North Carolina at Chapel Hill, the University of Pennsylvania (the VAST Lab and the Department of Radiology), and the University of Tennessee. Subcontracts from the prime contractors have been extended to: Harvard Brigham and Women's Hospital, U. Penn's GRASP Lab, the University of Pittsburgh, the University of Utah, and Columbia University. The prime contractors and their subcontractors comprise the software research consortium with the principal investigators of the prime contractors serving as the governing board. Together with the NLM Office of High Performance Computing and Communications as the executive member, the Insight Software Consortium is working to deliver a software toolkit to improve and enable research in volume imaging for all areas of health care.

NLM NATIONAL LIBRARY OF MEDICINE
THE VISIBLE HUMAN PROJECT®
GE Corporate Research & Development
COLUMBIA
Insightful intelligence from data
UT
VE RI TAS
Kitware
PENN VASTLAB
UNIVERSITY OF PITTSBURGH
THE UNIVERSITY OF UTAH
PENN GRASP LAB

Figure 1. The Insight Software Consotium Members: The Office of High Performance Computing and Communications – NLM, General Electric Corporate R&D, Kitware, Inc., Insightful, Inc., the University of North Carolina at Chapel Hill, the University of Pennsylvania (the VAST Lab and the Department of Radiology), and the University of Tennessee, Harvard Brigham and Women's Hospital, U. Penn's GRASP Lab, the University of Pittsburgh, the University of Utah, and Columbia University.

This work is a continuation of the successful Visible Human Project™. The original project was a far-reaching enterprise in human anatomy [3]. However, it fell short of the goal of creating a self-supported community of imaging research. The absence of inexpensive adequate imaging software tools eroded the momentum of the community. The current effort is one part of a multi-prong effort in anatomical and imaging research, the Visible Human Project™: From Data to Knowledge [1]. Two other efforts, one on advanced data acquisition and the other on distributed delivery of Visible Human Project™ content in the form of an online head and neck atlas are also in progress. Together, through these three works-in-progress as well as through three sponsored projects in Next Generation Internet distributed anatomical education and collaboration, we are attempting to stimulate the research community by providing new vehicles for the distribution of content, new data and data acquisition techniques, and new software image analysis tools.

2. Background

In the Fall of 1999, the National Library of Medicine awarded six contracts as part of an announced consortium effort called the Visible Human Project™ Image Processing Tools. The collective image processing contracts represent a 3-year, $7.5 million project in software image analysis tools. These contracts were organized through a flexible administrative mechanism of annual work assignments, permitting the constant redirection of the development of a software tool set as its design principles evolved. A first meeting of the software consortium was held in November 1999, and the initial vision for the software consortium was published at MMVR2000 [4]. A software architecture meeting was held in January 2000, and a meeting on algorithm validation was held in March 2000. Initial C++ classes were released at the second organizational meeting of the software consortium in June 2000. The name *Insight* was formally adopted at that time.

In our previous paper [4], we described our commitment to:

- Open Source Software – cost-free software with source code
- Toolkits as a Software Engineering Philosophy - APIs
- Compactness – not encumbering the software with multiple licenses
- Versatility – compatibility with multiple computing platforms
- Long Range Support – including distribution, education, and maintenance
- Validation – shared validation methods to promote algorithm development
- 3D – a strong preference for volume techniques and higher dimensions

The Insight Software Consortium has completed design and development of an initial version of a public medical image segmentation and registration toolkit known as the Insight ToolKit (ITK).

3. Architecture – Requirements

Beyond governing principles, the first task in formulating any software engineering task is to evaluate the user requirements as well as assess the available resources and current practices. We briefly cover the design decisions, the targeted user community, and the user requirements selected early in the evolution of ITK.

Primary users - We have selected application developers as the primary audience for ITK. Specifically, a clinician or other practitioner is not the first target user of the products of this work. The goal is to create an application programmer's interface (API) which can be used by developers in medical programs and software products wherever the problems of image or volume segmentation and registration exist. Unlike previous NIH imaging software development efforts, the goal is specifically NOT to create a single monolithic program. Rather, we are pursuing a software foundation from which a broad range of programs can be supported. The Insight mission is to provide: a software foundation for future research, an archival repository of image processing algorithms, a catalog of validation techniques, as well as a platform for advanced product development.

ITK supports native and generic data types (native: long, unsigned long, float, double, Char, Unsigned char, Int, unsigned int, shorts, unsigned shorts. Generic: u9, u16, etc.) as well as multi-component (intensity, intensity/□, RGB, RGB/□, vector or diffusion tensor data, etc.) and multidimensional data (x-y-z, x-y-z-time, x-y-z-scale, etc.). Insight has supported this notion of high dimensionality, both in independent and dependent variables, from its inception. N-dimensional datasets are explicitly support for specific operations, and where possible all pixel-wise operations support n-dimensions. These requirements lead to the widespread use of templates and generic types.

Processing requirements – ITK supports the processing of large datasets on modest processing platforms including multiprocessor systems. At the time of the design meeting (January 2000), the consortium selected mid-range desktop machines as the target (450 MHz CPU with a 0.5 GB of core memory). It was anticipated that mid-range desktops would track growing dataset sizes with a ratio of 10: 1 for dataset size to main-memory capacities. Current mid-range desktop computers (January 2002) can be found in the range of 1.25 GHz CPUs and main memory sizes of 2 GB are not uncommon. However, we continue to believe that our target of 10:1 dataset size to main memory is still valid as dataset sizes continue to increase. The increase in availability of multiprocessing shared-memory computers was also anticipated, and accommodations for pipelined and streaming software architectures were made early in the design.

Language and hardware support requirements – C++ was selected as the project programming language with the caveat that the library support multiple language bindings, permitting compilation and linking with a range of contemporary computer languages. Java provided better cross-hardware-platform portability, but it was deemed too young a language. Since part of the mission of Insight is to provide an archival mechanism for algorithms, C++ is adequate for programming a library. It has been observed that many of the software libraries still in use today are written in C or even in Fortran; however they serve the purposes of usable APIs and algorithm archives so long as they support multiple language bindings.

ITK is supported on multiple hardware systems. We specifically targeted Sun, SGI, Linux, and Windows/9x/NT/2000 as primary goals. Macintosh systems are not excluded, but were not specifically included in the original specifications. The advent of Mac OSX makes support for Apple systems more likely. By far, the greatest constraint on the platform support has been the adequate availability of compilers that correctly handle C++ templates. As mentioned before, the requirement for handling multi-dimensional images drives the use of generic types and a broad incorporation of templates in the ITK software.

Available resources and current practices – Engineering as an enterprise is always subject to the available resources and the current methods in use. Software engineering is no exception; the resources in question are measured in the expertise of your personnel, access to machines, and other intangible factors. The Insight Software Consortium is geographically widely distributed, so collaboration and communication have been limiting resources. Much of our software engineering has been in support of the distributed software development enterprise, including automated test and build procedures, online reporting of the state of the software, automated generation of documentation, and bug tracking. As a result, all software elements have been accompanied by inline documentation and built-in software regression tests.

As a rule, we adopted the most current software development practices in object oriented programming, while adhering to the principles of compactness and versatility. We selected the *vxl* numerical programming library from another open source initiative; vxl has no encumbering language preventing free development or distribution of derivative work. The consortium also incorporated the Standard Template Library (STL) as well as programmed through the creation of new templated classes. The compactness principle dictates that the toolkit shall not have any dependencies on specific visualization software; however it must have hooks and interfaces for interactive operations, callback, and event handlers for dynamic visualization.

4. Software Design and Project Execution

The user requirements and design goals of the Insight project directed the project along certain predictable lines. The memory and parallel processing requirements dictated a pipelined streaming software architecture. The distributed and sophisticated nature of the programming effort placed other, more novel concerns before the consortium, setting specific directions for the group. Nevertheless, knowing the direction of our code development and foreseeing the impact and breadth of the work being attempted are two different things. The Insight Consortium has broken new ground in collaborative software development and the use of generic programming in API development.

Extreme programming – The brevity of our development cycle and limited communications among the programming teams suggested a departure from the traditional government sponsored software development process of design specification followed by implementation, quality assurance, and testing. A continuously re-iterative design, implementation, and coding process, called *extreme programming* [2], has been adopted by the consortium. The ITK API has undergone continuous review and modification by the expert developers on the programming team. GE Corporate R&D along with Kitware, Inc. serve as the lead design groups, with GE CRD handling issues of testing and quality control during development and Kitware serving as system integrators for the project. Other teams from across the consortium are contributing new implementations of complex medical image processing algorithms, helping to refine the software architecture as the toolkit grows in complexity and mathematical sophistication. There are substantial costs to the constant redesign, but the reward is early implementation and rapid improvement through frequent revision.

Generic Programming – The Insight Software Consortium has undertaken an aggressive approach in the design and execution of ITK. Advanced programming concepts and techniques in object-oriented programming have been incorporated into the toolkit. C++ is the chosen language for the software development; however, it is the comprehensive use of programming templates and the adoption of generic programming practices which have driven this project to the edge-of-the-art in programming. These choices permit algorithm development without strong typing of the atomic image elements, providing for great flexibility in the toolkit. Color data, greyscale, floating point, and other image values are handled with equal facility. More importantly, the generic programming style permits rapid development of the toolkit by many team members simultaneously, allowing large changes to the software across all aspects of the toolkit with minimal disruption. The pervasive use of these methods makes ITK unique among software APIs.

CMake and CABLE - Early in the development process, we recognized the need for a comprehensive build and test environment. We were also aware of the need to support multiple languages, especially interpretive development languages for rapid application prototyping, including the Tool Control Language (TCL), Python, and Java. As part of our design process, we therefore included the creation of CMake, a cross platform build environment supporting Windows (Microsoft Visual C++), GNU C++, and SGI Irix C++. CMake prevents skew of our code by eliminating the need for simultaneous development f the software on multiple platforms. In addition to CMake, we included the development of CABLE, a generic code wrapping tool to provide multiple language bindings for languages such as TCL. These advanced software engineering concepts are applicable to a wide cross section of programming initiatives and are available in source code form as separate tools offered from the Insight Software Development Consortium Indeed, CMake has already been adopted internationally among software developers faced with writing code for Windows and Unix platforms.

Engineering a Collaboration: DART – an automated build and test environment – Finally, we have created a comprehensive test environment that supports build and regression testing. Continual and nightly builds of ITK Test results are posted on a web-accessible dashboard, providing immediate feedback for software developers as to the integrity of their modifications and their compatibility with all supported computing platforms. Coordination of a large software project, especially one with a geographically distributed team would be nearly impossible without this facility. Like CMake and CABLE, DART is available as a separate software engineering suite in source code form.

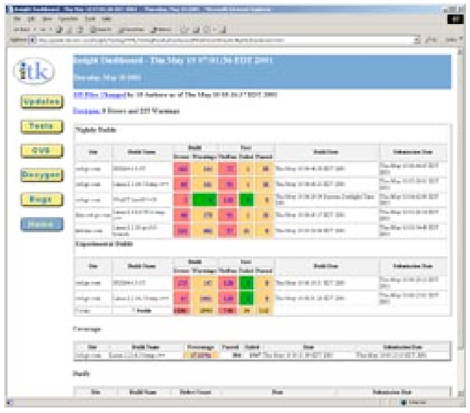

Figure 2. A DART Dashboard for ITK. This is a postage stamp view representing a full web page that reflects the state of the toolkit at any time. Nightly and continuous builds constantly update the online report showing failed tests and warnings on multiple platforms with multiple compilers. These builds are run at different locations in the consortium with the results displayed on a single portal. Recent changes to the source code can be tracked through the dashboard including the programmers responsible for changes that lead to build failures. Distributed development teams can be coordinated in multiple time zones using this facility. DART is a product of the Insight Software Consortium and is available as an open source product.

5. Algorithms, Examples and the ITK Beta Release: an invitation

The consortium members represent a broad cross-section of skills and backgrounds, each with considerable expertise in medical image analysis. The algorithm developers in the group have proposed and are contributing methods representing a comprehensive range of approaches for the filtering, processing, segmenting, and registering high-dimensional medical data. Watersheds, level set mathematics, geometric partial differential equations, finite element models, finite differencing engines, statistical pattern recognition techniques, and Voronoi spatial decomposition hybridized with deformable models for segmentation are among the many methods represented in ITK. Many of these methods are multithreaded and pipelined for maximum performance. Insight is dedicated to open-source code, so all of these methods are publicly available through the API source.

The Insight Software Consortium is beginning a public beta release of ITK. Interested programmers should search the following URL – http://www.itk.org. Additional documentation, examples, and a thorough review of the materials in this paper can be found at that site. The consortium is especially interested in feedback from application developers on the integration of ITK with existing visualization and interactive medical image analysis packages. Beta test results will be solicited in June 2002 with the goal of a full release of ITK by October 2002.

6. Acknowledgements

The authors would like to thank the National Library of Medicine (NLM) and its Visible Human Project™ for its support for this project, along with its partners: the National Institute for Dental and Craniofacial Research (NIDCR), the National Eye Institute (NEI), the National Science Foundation (NSF), the National Institute for Neurological Disorders and Stroke (NINDS), the National Institute on Deafness and other Communication Disorders (NIDCD), and the National Cancer Institute (NCI).

References

[1] Ackerman, M.J., T.S. Yoo, D. Jenkins. 2000. The visible human project: from data to knowledge. Computer Assisted Radiology and Surgery (Proceedings of CARS2000), H. U. Lemke, et al. eds., Elsevier Press, Amsterdam: 11-16.

[2] Beck, Kent. 1999. Embracing change with extreme programming. IEEE Computer, October 1999. 32(10). 70-77.

[3] V. Spitzer, M. J. Ackerman, A. L. Scherzinger, and D. Whitlock. 1996. The Visible Human Male: A Technical Report. J. of the Am. Medical Informatics Assoc. 3(2) 118-130.

[4] Yoo, T.S. and M.J. Ackerman, 2000, A new program in medical image data processing, Medicine Meets Virtual Reality 2000 (Proceedings of the 8th Annual Medicine Meets Virtual Reality Conference), James Westwood, et al. eds., IOS Press, Amsterdam. 385-391.