OPEN DRAGON PROGRAMMER'S TOOLKIT:
A FRAMEWORK FOR LEARNING GEOINFORMATICS SOFTWARE DEVELOPMENT

Sally E. Goldin, Faculty
Kurt T. Rudahl, Faculty
Department of Computer Engineering
King Mongkut's University of Technology Thonburi
91 Pracha Uthit Road, Bangmod
Bangkok 10140, Thailand
seg@goldin-rudahl.com
rudahl@rsgis.net

ABSTRACT

The OpenDragon Project develops and distributes free remote sensing and GIS software to educational organizations in Southeast Asia. In addition to providing a turn-key package that can be used for application studies in geography, forestry, geology, agriculture,and other natural resource related disciplines, OpenDragon can also serve as a testbed for students working on research projects in geoinformatics, developing new algorithms or systems, or exploring the data structures and processing techniques common in remote sensing and GIS. This paper briefly describes the OpenDragon architecture and the Programmer's Toolkit, which provides mechanisms that allow students to add new functionality to the system. We describe a number of student research projects that used the OpenDragon framework. We also discuss two university courses, for very different audiences, that focused on software development for geoinformatics and utilized OpenDragon source code and libraries for examples and assignments.

INTRODUCTION

Continued progress in geoinformatics research and applications depends on educating new generations of theorists and practitioners (Konecny & Nyapola, 2006; UNECA, 2000). Effective education and training in turn depend on the availability of appropriate tools, including software. The OpenDragon Project was established in 2004 to help satisfy this requirement. The project, originally funded by King Mongkut's University of Technology Thonburi (KMUTT) in Thailand, develops and distributes free remote sensing and GIS software to educational organizations in Southeast Asia. Details of the project can be found at *http://www.open-dragon.org*.

The primary software offered by the project is OpenDragon. OpenDragon is based on the commercial Dragon/ips® package originally developed by Goldin-Rudahl Systems, Inc. (Goldin & Rudahl, 1986). Dragon/ips, initially released in 1987, was the first remote sensing software to run on off-the-shelf personal computers and was designed from its inception to meet the needs of developing countries. OpenDragon is simple, portable, easy to use, and modest in its hardware requirements. Furthermore, it is fully internationalized, so that producing versions in languages other than English is a straightforward process of translating a few dozen pages of text. Currently, OpenDragon includes a variety of Asian languages, including Thai and Bahasa Indonesia. Vietnamese, Chinese and Mongolian translations are under development.

OpenDragon, like Dragon itself, can be used as a turnkey platform to teach undergraduates and even secondary school students (Goldin & Rudahl, 2005) about geography, agriculture, geology, meteorology, forestry and other natural resource disciplines. In this role, it supports the education of practitioners and users of geoinformation. Due to its modular design and the computer engineering background of its developers, however, OpenDragon can fulfill another role. It can provide a testbed for students and researchers to explore new geoinformatics algorithms and data structures, and to learn about the special challenges of geoinformatics software development. Thus, unlike most educational software packages, it facilitates the education of theorists and the development of improved software tools, providing a firm foundation for application-oriented practitioners.

This paper describes our experiences using OpenDragon and its Programmer's Toolkit to support student research and to teach about geoinformatics software development.

OPENDRAGON FUNCTIONAL CAPABILITIES

The OpenDragon turnkey application provides core remote sensing image processing functionality, as well as some raster GIS operations, in a simple, easy-to-use desktop package. Its functional capabilities include:

- Multi-image gray scale, pseudocolor and composite display, with Unicode image annotation
- Histogram and scatterplot display, metadata display and editing, interactive image data value exploration
- Image arithmetic, convolution and median filtering, vegetation indices, masking, recoding, principal components
- Interactive training, signature display and editing, supervised and unsupervised classification, classification error analysis
- Geometric correction and image-to-image registration
- Vector feature import, display, capture, and export
- Image feature measurement; area histograms and line profiles; irregular subset extraction
- Slope and aspect calculations; raster buffers; rule-based multi-band combinations
- Context sensitive help and on-line manual.

The Programmer's Toolkit exposes an Application Programming Interface (API) that allows a user with fairly basic C language programming skills to create new modules compatible with OpenDragon's data and file structures. The Toolkit can be used for a variety of purposes:

- To create new data import and output capabilities;
- To implement processing options not currently offered by the turnkey package;
- To implement novel or advanced image processing and vector processing algorithms that can be used in conjunction with the turnkey package.

THE ARCHITECTURE OF OPENDRAGON

OpenDragon is a modular system that consists of four basic components, as shown in Figure 1.

The ***User Interface*** component, written in Java, provides flexible and convenient access to OpenDragon functionality using familiar graphical conventions. The contents of the User Interface screens are generated at runtime, based on a set of XML description files plus language-specific message files.

The ***Image Processing Engine*** handles all manipulation of image and vector data. It receives its instructions via command files, and thus can be run in a batch mode without the user interface.

The ***Viewport*** (of which there are multiple instances) is responsible for image display and real time interaction (e.g. interactive measurement or heads-up digitizing). It provides scrolling, zooming, capturing and printing of images as well supporting image annotation (legends and text).

The ***Dragon Executive*** maintains application state, coordinates communication between the other components, and supports scripting.

Presently all components except for the Viewport are platform-independent and can run on Windows, Linux and Mac OS/X. Re-implementing the Viewport using platform-independent technologies is a high-priority task if we can obtain new funding for the project.

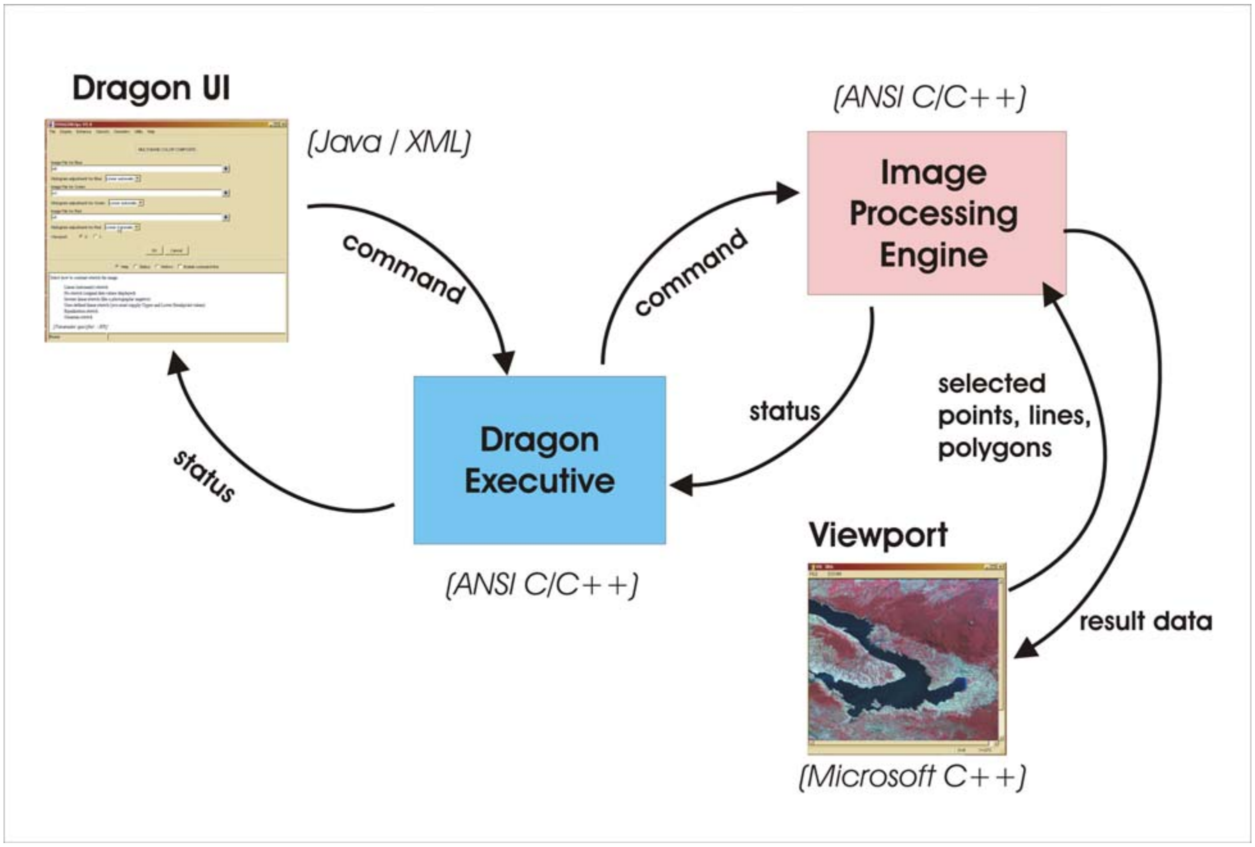

Figure 1. Primary architectural components of OpenDragon **.**

For research and programming studies, the most important component is the Image Processing Engine. As shown in Figure 2, this component consists of operation-specific application code, functions to handle communication with other components of the system, and a set of core functions for reading, writing, and querying geographic data. The core functions provide access to raster and vector data sets and their metadata, as well as some ability to manipulate other categories of information such as training signatures and image statistics.

Students and researchers need access primarily to the core functions in order to start writing their own geoinformatics modules. Wrapper functions make it easier to use these functions from a C language program (since the lowest level functions are implemented in C++). The core libraries, the wrapper functions, a simplified set of header files, HTML documentation, sample programs, and build scripts are packaged together into the OpenDragon Programmer’s Toolkit. With the toolkit, students and researchers can write, compile and link code that will read and write OpenDragon images and vector data, and take advantage of other data management facilities.

Note that at present OpenDragon is not truly open source. Educational organizations can request source code and the project will evaluate each request individually. However, the project is still in the process of reorganizing, simplifying and rationalizing the source code to make it comprehensible to a non-expert programmer. If the project receives long-term funding to support this activity, we will move OpenDragon into a true open source model. Currently, however, other goals such as adding functionality, completing the transition to platform independence, and supporting additional national languages have a higher priority.

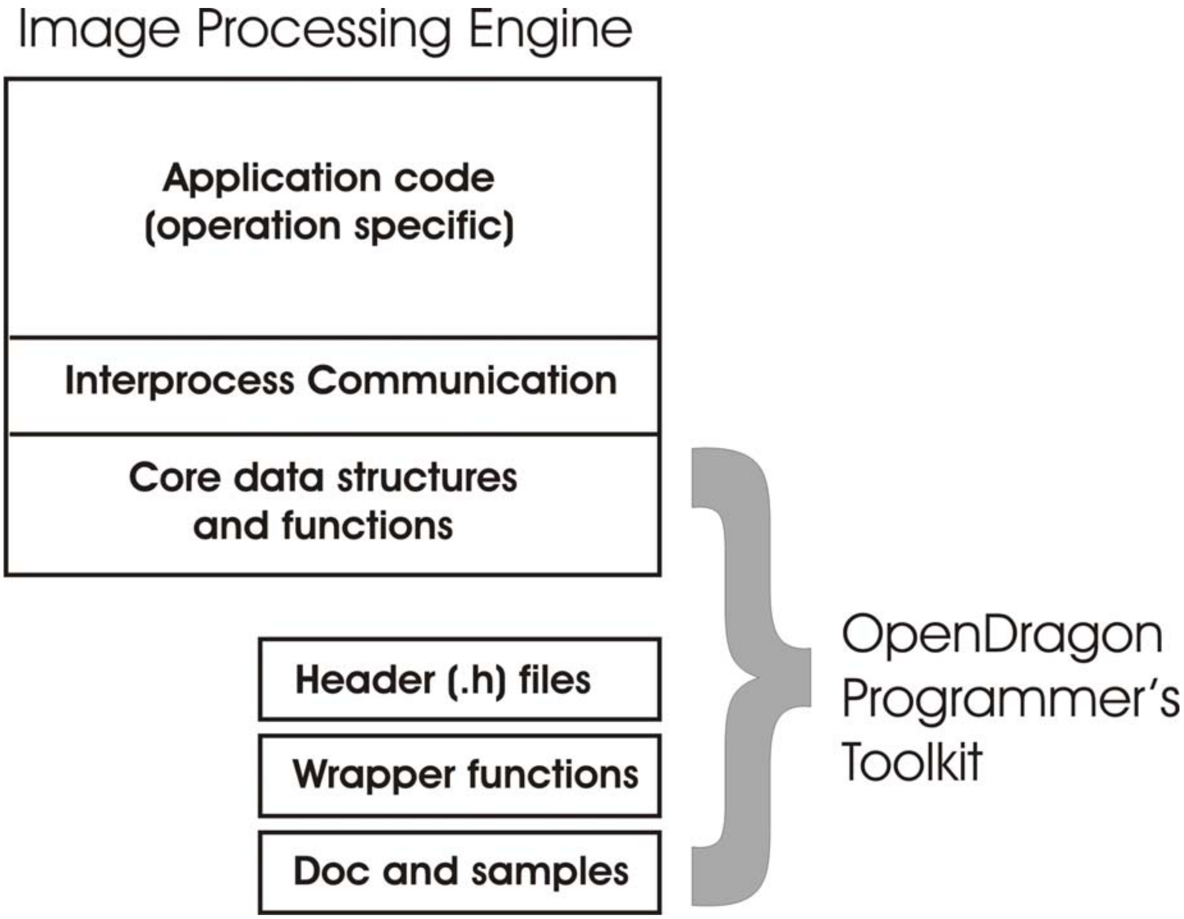
Image Processing Engine
Application code (operation specific)
Interprocess Communication
Core data structures and functions
Header (.h) files
Wrapper functions
Doc and samples
OpenDragon Programmer's Toolkit

Figure 2. The structure of the OpenDragon Programmer's Toolkit.

STUDENT RESEARCH USING THE OPENDRAGON TOOLKIT

During the past four years, a number of undergraduate and graduate student projects have used OpenDragon and the Toolkit. The fact that students do not need to develop new data formats and can use the OpenDragon application saves them time and allows them to focus on the novel aspects of their research. This section summarizes the work our students have done or are doing with OpenDragon.

Undergraduate Research Projects

Vector GIS using PostgresSQL (2005). Songpon Yookeaw and Nopporn Jinbunluphol created a prototype vector GIS system in order to evaluate the spatial database capabilities of the open source PostgresSQL database system. They used OpenDragon to process raster images to serve as backgrounds for vector display, and the OpenDragon vector file format for vector import and export.

Remote Sensing Image Classification using Neural Network (2005). Pawarat Kitmanomai and Prachya Lalitnorasate created a neural network classifier and compared its performance to the OpenDragon maximum likelihood classification. They used the Toolkit to read and write training and result images and to process signatures.

Generation and Animation of Perspective Views (2006). Korapin Atichaichotikul and Piyana Sukanghong developed an application to create a perspective view of a landscape based on a digital elevation model, draping the model with satellite image information in order to enhance the realism and information content. They used the Toolkit to read and process the DEM and image files.

Automatic Extraction of Road Networks (2006). Sasiwimon Ratchatasomboon **a**nd Wittaya Unjai developed a system to identify roads in medium resolution imagery, using a combination of low-level (e.g. filtering) and high-level (e.g. context-based rules) approaches. They used OpenDragon and the Toolkit for all input, output and results display.

Image Processing in Hardware (2007). Kittituch Manakul and Surachai Chatchalermpul developed a co-processor to execute time-consuming components of image processing operations, such as gray-level co-occurence matrix calculations, in hardware. They used OpenDragon and the Toolkit to create a baseline, software-only implementation of their algorithms for comparison purposes.

Interactive Map and Directory for KMUTT Campus (2007). Aunyarath Prasittiarpa and Kanokporn Sudsangswang created a special-purpose GIS application to serve as an English/Thai interactive guide to the KMUTT campus. They used OpenDragon and the Toolkit to create a vector representation of campus roads and building footprints.

Travel-time GIS "Buffer" Operation (ongoing). Pirakorn Iamcharernying and Pawat Duantsatit are developing algorithms to compute buffers that take travel time along road networks into account. They are using OpenDragon for image input and display.

Crime Statistics GIS (ongoing). Manatsanam Aksornkul and Kanok Chausakul are developing a special-purpose GIS to help police visualize spatial patterns in crime statistics. They are using OpenDragon and the Toolkit for data conversion and display purposes.

Master's Theses

Improving the Estimated Area of Rice Cultivation by Classification Techniques (2008). .Ithiphol Ekahitanonda explored the efficacy of multi-temporal analysis, genetic algorithms, and fuzzy decision making approaches as ways to resolve ambiguities in identifying rice from ETM imagery. He used the OpenDragon Toolkit extensively, developing a wide range of new modules and utilities.

Change Detection in Natural Disasters: the Tsunami Case (2008). Supannee Tanathong implemented an object-based strategy for characterizing landscape change after natural disasters such as the December 2004 tsunami. She used the OpenDragon Toolkit to experiment with segmentation using texture features.

Summary

A number of Computer Engineering students have undertaken geoinformatics-related research projects in the past four years. All of these projects would have been far more difficult if students had been required to develop all software capabilities "from scratch". Although our students are studying Computer Engineering, they are not necessarily expert programmers. Nevertheless, they had no significant problems applying the Toolkit to their projects.

Prior to the availability of OpenDragon, projects and theses related to remote sensing and GIS were rare in our department. The OpenDragon Project has provided not only capabilities to assist students in their research, but also new visibility for the field of geoinformatics within our primarily engineering-oriented university.

TEACHING GEOINFORMATICS SOFTWARE DEVELOPMENT

In addition to encouraging students to use the Toolkit for research, we have also used OpenDragon to teach two different courses related to geoinformatics software development.

The first was a course for seniors and graduate students within our own Department of Computer Engineering, entitled "Algorithms and Architectures for Geoinformatics". This course introduced students to concepts of representing spatial information in computers, including rasters, non-topological and topological vectors, quadtrees, sparse data arrays, TINs, etc. We used actual source code from OpenDragon to illustrate concepts of pixel-, neighborhood- and region-based algorithms. Students completed two group projects during the semester, mostly using the OpenDragon Toolkit as a foundation. Projects included slope and aspect calculation, region-growing, buffer generation, and edge extraction. Since all students had substantial backgrounds in programming, we focused on issues such as efficiency in algorithm and data structure design, and software engineering principles such as modularity, separation of concerns, and consistent style.

The second course, "Computer Programming for Geoinformatics", was created to introduce masters students from the Chiang Mai University Department of Geography to the basics of writing software in the geoinformatics domain. The goal of this course was provide the students with sufficient background to understand and evaluate commercial or third-party software, write small programs to help integrate existing packages, and create scripts to automate work flows. The students had little or no experience with programming. Thus the course began with the elements of programming: variables, conditional statements, loops, functions and so on.

About half of the course time was spent in laboratory exercises. The labs relied heavily on the OpenDragon Toolkit for raster work, most of which was done using the C language. For labs illustrating vector concepts, we used the OpenDragon vector format in some simple Java applications. Since this course may be of greater interest to ASPRS members than the more engineering-oriented syllabus developed for our own department, we have provided a generalized schedule in Table 1.

Table 1. Outline of topics for "Computer Programming for Geographers" Course

Week	Lecture Topic	Laboratory Topic
1	Course introduction; fundamental programming concepts (1)	Introduction to C; compiling and linking
2	Raster data models and data structures	Coordinate conversions in C
3	Programming concepts (2): files, pointers	Introduction to Java; coordinate conversions in Java
4	Raster operations	Pixel-wise raster operations in C
5	Raster operations continued	Neighborhood raster operations in C
6	Vector data models and structures	Non-topological vectors in Java
7	Vector operations	Non-topological vectors in Java (continued)
8	Alternative data models	Topological vectors in Java
9	Web architectures and Web GIS	Topological vectors in Java (continued)
10	SQL and relational databases	Simple database exercise with Java/JDBC
11	Location-based services	Database exercise (continued)
12	Scripting	UI building in Tcl/Tk
13	Scripting (continued)	Automation and interfacing using Tcl/Tk
14	Program analysis and code reuse	Using C libraries to create a program
15	Evaluating commercial software	Open for practice/questions before examination

In both these courses, OpenDragon served as a core resource. A student studying software development will learn a great deal more from viewing real-world code than from reading a text book or listening to lectures. Writing code provides an even deeper understanding of the issues involved in software development. Our experience in Chiang Mai indicates that even non-technical people can benefit from well-written, well-organized code examples and exercises.

Most students in Geography, Planning, Forestry, Oceanography, and other natural resource related disciplines have very little exposure to programming concepts. Seeing the step-by-step operation of remote sensing or GIS processes, expressed succinctly in the form of a program, can help make the processing logic more concrete and easier to understand. In addition, of course, we need professionals with "spatial intelligence" to work on future geoinformatics tools. The remote sensing and GIS developers of the future are at least as likely to come from the "softer" disciplines of geographic science as they are from the engineering side. Tools like OpenDragon and its libraries facilitate professional development by letting students "get their hands dirty" working with real code.

FUTURE DEVELOPMENTS

KMUTT's funding commitment for the OpenDragon Project ended in November of 2006. Since then, we have been seeking new sponsors who will support the continued availability and expansion of OpenDragon's capabilities. Ideally, as noted in the Introduction, we would like to convert the package to a true open source distribution model in order to provide the greatest benefit for the widest audience. Though open source software itself is free, open source projects are not sustainable without some external funds. The most successful open source efforts (for example, Linux, Apache, Perl) all have corporate support (IBM, Novell, Sun, O'Reilly Publishing).

We have a long wish list for features that we would like to incorporate into OpenDragon, topped by a platform-independent viewer component. In the area of the Programmer's Toolkit, we want to provide a mechanism that will allow users to integrate their own geoprocessing modules into the OpenDragon user interface. A prototype of this capability already exists but needs to be made accessible to users. We would also like to expose more of the system's functionality in the Toolkit API.

As of the time this paper was written, the OpenDragon Project is more or less static due to lack of support, although the commercial version of Dragon/ips has been significantly enhanced. We continue to utilize the capabilities of both the turnkey application and the Toolkit to engage and instruct students at our own and other institutions.

CONCLUSIONS

A variety of software packages are available for teaching remote sensing and GIS at an introductory level. However, most if not all focus on application-oriented users. OpenDragon is unusual in that it also addresses the needs of students and researchers who need to program their own algorithms or who want to understand the internal workings of remote sensing and GIS software.

The OpenDragon Programmer's Toolkit provides a level of extensibility that is rare in the world of geoinformatics software. It offers a testbed for research as well as a framework for hands-on class projects. By revealing the inner workings of geoinformatics software to the students of today, OpenDragon helps build the skills of the people who will develop the spatial analysis software of tomorrow.

REFERENCES

Goldin, S.E. and K.T. Rudahl, 1986. Dragon and Phoenix: low-cost image processing for developing countries, In the *Proceedings of the South East Asian Regional Computer Conference.* Computer Association of Thailand, pp 41-42.

Goldin, S.E. and K.T. Rudahl, 2005. "Report on the High School Workshop on Remote Sensing for Drought Assessment", King Mongkut's University of Technology Thonburi, December 2005.

Konecny, M., and H. Nyapola, 2006. Capacity building for geo-information development, E/CONF.97/6/IP.2. United Nations Economic and Social Council, *Proceedings of the Seventeenth United Nations Regional Cartographic Conference for Asia and the Pacific*, 18-22 September, 2006.

UNECA (United Nations Economic Commission for Africa), 2000. Ad-hoc Expert Group Meeting on the "Study of the Future Orientation of Geoinformation Activities in Africa", Addis Ababa, 6-10 Nov, 2000, Report of Working Group .http://www.uneca.org/eca_resources/Conference_Reports_and_Other_Documents/disd/geoinformation/Report%20of%20workinggroups.pdf.