A FRAMEWORK FOR DEVELOPING REMOTE SENSING APPLICATIONS

T. Ahmad, M.F. Hayat, M. Afzal, H.M.S. Asif and K.H. Asif

Department of Computer Science and Engineering, University of Engineering and Technology Lahore, Pakistan.
Corresponding Author E-mail: tauqir_ahmad@hotmail.com

ABSTRACT: Remote Sensing Application (RSA) is important as one of the critical enabler of e-systems such as e- governments, e-commerce, and e-sciences. In this study, we argued that owning to the specialized needs of RSA such as volatility and interactive nature, a customized Software Engineering (SE) approach should be adapted for their development. Based on this argument we have also identified the shortcomings of the conventional SE approaches and the classical waterfall software development life cycle model. In this study, we have proposed a modification to the classical waterfall software development life cycle model for proposing a customized software development Framework for RSAs. We have identified four (4) different types of changes that can occur to an already developed RS application. The proposed framework was capable to incorporate all four types of changes.

Keywords: Remote Sensing, software engineering, functional requirements, types of changes.

INTRODUCTION

The remote sensing e-systems/applications are data-driven and data intensive, collecting huge volume of data/information and this information/data is not only used for short term decision makings, but it is also used for long term decision making (Ahmad and Shah, 2010). The heterogeneous nature of remote sensing applications and their temporal and spatial diversity compound complexity of their development process, management, and their successful execution (Longley et al., 2005). The conventional software development frameworks/methodologies and database technologies have some drawbacks that make them unsuitable to use for the development of such type of e-systems (Ahmad and Shah, 2010). Main reasons of their unsuitability are their peculiar characteristics such as temporal instability, diverse data formats, broad context, voluminous nature etc. (Brown and Jones, 2001). These characteristics not only make a need of a new class of specialized information systems but also a customized approach towards the development of these specialized applications, e.g., remote sensing systems/applications. A major challenge in the development of such applications/ systems is to capture their high dynamicity and evolutionary nature. Such evolutionary domains necessitate a highly iterative development processes especially a highly iterative maintenance process after their development. Although the agile development approaches are employed to develop such type of applications/ systems, but they are unable to develop their all functional requirements. The requirement of some applications such as Geographical Information System (GIS) is to trace back the history of changes that occur to an application. This requirement makes the application development process more complex.

MATERIALS AND METHODS

The advancement of computing Power communication technologies, have revolutionized the world through the comprehension of e-systems such as e-governments, e-commerce, and e-sciences (Bagchi et al., 2001). The e-systems based on RSAs were data driven and data intensive, collecting countless information not only for short term decision makings but also for long term decision making (Ahmad and Shah, 2010). The information collected varies from business, medical and personal information to scientific, satellite and surveillance information (Fang, 2002). The data describing this information may itself be spatial, multimedia or hypertext documents. This diversity of information and data adds an element of complexity to the management and successful execution of e-systems (Longley et al., 2005).

Conventional database technologies have been deficient and unsuitable to handle this huge amount of data and information with such diversity in content and format (Patrick, 1993).

The literature has classically viewed the process and the product of system projects in the early days which were developed/ implemented in an un-systematic way. However, as the complexity of projects increasesed, software engineers thought some systematic ways of developing it. System development frameworks/methodology which gave a chance to subdivide a project to reduce the overall complexity (Anthony, 2007). By the use of software development methodology, the system development process seemed to be transparent and provided the best control of project management thus reducing the overall risk and uncertainty (Brian, 2009). However in the last three decades, with the advancement of satellite communication and computational technologies the challenges of developing a good information system was the main issue in computer science. Due to this reason, the development of a framework suitable for a particular domain was the subject of extensive research (Josh, 2012).

Proposed Framework: Software development (SD) process was a messy activity, commonly known by a phrase “code and fix”. For small systems, SD can be carried out having less emphasis on an underlying plan and design of the system. But for large systems, it was not easy to add-up/ update features to a system. Also, bugs/ errors became increasingly rampant and increasingly hard to fix. Hence, there was need to have a software development framework to structure, plan and control the process of developing an information system. This framework compels a regimented process upon SD with the objective of making SD more probable and more creative. This can be accomplished by developing a thorough process with a sturdy stress on planning like many other engineering disciplines. In this paper, we have proposed software development framework for remote sensing applications and argued that there was a need to have a specific framework for these types of applications. The advantage of having such framework was that it defines step to be taken, forcing the developers to follow the defined framework in a customary way. It specified the way to develop the models in a sequence, and how to derive a model from another one at the abstraction level immediately above it. The framework guaranteed that developers will be familiar at every instant during the development life cycle what was to be done next for achieving their goals.

The major contribution of our research work was the introduction of Pre-Analysis Phase in software development life cycle. In our opinion, as conventional methodologies for open-ended system modeling/ development were generic therefore Universe of Discourse (UoD) needed to be well defined and delimited. However, in application areas such as Remote Sensing, most of types of UoD were already known and it made little sense to input entire Default Knowledge (DK) in Analysis Phase. Due to this reason, we have introduced a Pre-Analysis Phase (PAP) which reuses recognized, firm and concrete artifacts of the domain.

The situations when software was developed and maintained were influenced by the types of changes that would occur to software during or after the software development. These possible changes that could occur during or after software development as listed below:

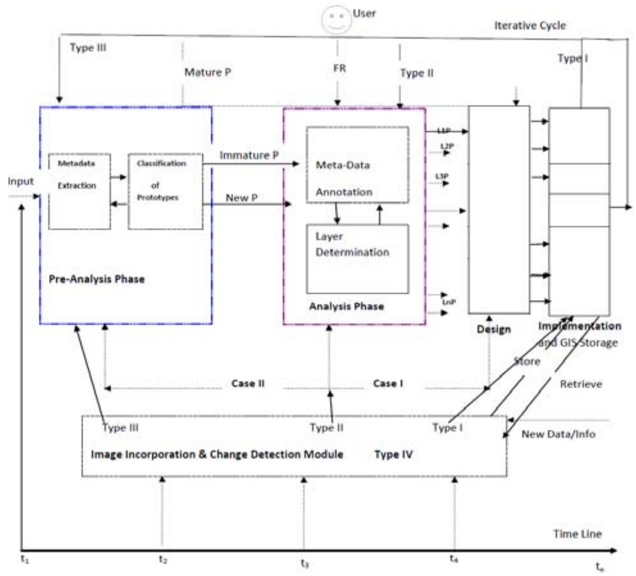

Figure 3.1: Proposed Framework for Remote Sensing Applications development

Type I: Change occurs to data of software (data change).
Type II: Change occured in functional requirements/metadata of software.
Type III: Change occured to both data and metadata (structural change).
Type IV: Change occured due to both (data change and structural change).

As we have said earlier that these four (4) types of changes could affect the development and maintenance processes, the fact was also pointed out by Shah in (Shah, 2001). The waterfall software development life-cycle was modified by Shah and based on this modified waterfall software development life-cycle, a new framework was proposed for the prototype-based software development methodologies. Here for remote sensing software/applications, we extend the list of changes that was suggested by Shah. We noted that this extended list influences the development and maintenance process as it has already been pointed out by in (Shah, 2001).The detail of the above types of changes are given below:

To elaborate the four (4) types of changes and modifications in an existing/developed application, we take an example of a canal passing through a mass of trees. Changes may occurred to the geographical features of the application, which may affect the development and maintenance processes of this remote sensing application.

i) **Type I:** Change occurs to data of the application

This change occurs when there was change to data parameter of an already developed remote sensing application. This change type can simply be handled/ incorporated to the application without affecting FR or structure of the application and to incorporate it, we needed less development effort. In our example of canal passing through a mass of trees, if there was a change in the number of trees or flow rate of water in the canal, this change can be easily detected and incorporated directly by the application.

ii) **Type II**: Change occurred to functional requirements (FR) of the application.

Changes/modifications that occurred to functional requirements of the application and they were categorized as Type II changes.

FR captures the planned behavior/ functionality of an application. This behavior may be termed as service/tasks or functions that the application was needed to perform, whereas, the structure of an application provided both the structural and behavioral capabilities of an application. Structural requirements are defined by a set of instance variables and methods (or operations). In other words, the main difference between FR and structure (structural requirements) of an application was that, the set of FR may be a subset of the structural requirements.

Type II changes occured to a remote sensing application due to lack of communication between clients and developers, insufficient domain knowledge, change in UoD or changes in the requirements of clients. Due to the change (Type II change) in the FR to a remote sensing application, the following two cases may arise.

Case I: Change to FR may not cause a structural change:

For example, in the running example, change to the FR was: Finding buildings of area greater than 100 square feet along the canal bank. This new requirement may be needed to incorporate in the cases such as urban planning, telecommunication, or environment monitoring, etc.

Case II: A functional requirement may cause a structural change:

In this example, a new FR was: Capture all features in a 20 m corridor along the canal. The addition of new requirements (or the change) to the existing (or already developed) application, resulted in a structural change.

iii) **Type III:** Change occured to structure of the application:

Every fresh data item was to be input for similarity check process; either known or a new artifact. The similarity check compared its similarity with the core elements of the existing artifacts. A structural change, depending upon its degree, may significantly alter the structure of an existing object to warrant its similarity check. This means that a Type III change will be handled in a way similar to when a new artifact is input to the system.

Type IV: Change was a combination of the above three (3) types of changes:

In Type IV, it can be a change in data, functional requirements and structure of a remote sensing application. It was to be noted that if we go from the changes Type I to Type IV, the complexity of incorporating these changers was in increasing order. The increased complexity from Type I to Type IV resulted in corresponding increase in development effort. Type II and Type III changes in which functional requirements changes and structural change needed to be incorporated, result in more development effort as compared to Type I change in which only new data needs to be incorporated. Also, a change in already existing system needed more development effort as compared to a corresponding change in a system under development.

Functionality of Pre-Analysis Phase

The pre-analysis phase consisted of two sub phases:

i) Classification of Prototypes

ii) Metadata Extraction

The pre-analysis phase was triggered with the new facts which arrive during system development after scratch and the changes occurred during system development.

i) Classification of prototypes

This phase classified artifacts into:

- New Prototypes
- Immature Prototypes
- Mature Prototypes

New Prototypes are previously known artifacts. These artifacts are passed onto Meta Data Annotation Sub-phase for subsequent input into Analysis Phase.

Immature Prototypes are partially recognized artifacts and needs further clarification and recognition, and were passed onto Analysis Phase to be known artifact with the help of stake holders and domain experts.

Mature Prototypes are known to the system and were directly input to the design phase. We have categorized the changes that may occur during RS Application development into [Type I, Type II, Type III and Type IV], already discussed in Section 3. Type I can be handled by implementation phase, while pre-analysis phase handles Type II (when caused structural change) and Type III changes.

Analysis Phase

The analysis phase consisted of two sub phases:

i) Metadata Extraction/Annotation

ii) Layer Determination

The analysis phase generated an analysis report containing different types of identified layers, for example, vegetation layer, water body layer, buildings layer etc. These layers were then passed onto the design phase for subsequent designing of layers so that implementation can be done in a proper way. The analysis report also depends upon the functional requirements of the user. User was an important part throughout the process of analysis.

Design Phase: The design phase of our proposed framework got its input from the analysis phase as layer prototypes (LP) for example; vegetation layer, water body layer, land cover layer etc. These layers were designated as L_1P, L_2P. L_nP and sent to the image incorporation and GIS storage phase. Vegetation layer can be further classified into sub-layers as: tree stands, individual land mark trees and large hedgerows, etc. Structure layer can be classified into sub-layers as: labs, offices, school and libraries, housing, warehouses, public buildings (fires/police stations, government offices), etc. Water body layer can be classified into sub-layers as: lakes, ponds, rivers, streams, drainage, etc. Similarly, transportation layer can be further classified as: paved roads, unpaved roads and trails, bridges, railroads, runway/helipads, parking lots paved and unpaved, roads centerlines, etc.

Image Incorporation and GIS Storage Phase: Designed layers were to be input to the image incorporation and GIS database phase. In our proposed framework, the design and implementation phases worked together for a high-productivity programming environment, as the designed layers can be manipulate by the programmer directly in the image incorporation and GIS database phase.

RESULTS AND DISCUSSION

In this study, we have proposed a software development framework for remote sensing application domain. This application domain has different and peculiar characteristics that made the development of this application domain different front than other application domains, and they also made the existing development framework inappropriate to be used for the development of applications of the RS domain. Among these characteristics, different types of changes (such as data change, structural change and combination of these changes) to the applications that occured after their development made them difficult to handle and incorporate. Since maintenance of already developed software/applications especially remote sensing applications was generally a time-consuming and costly task, therefore, the main contribution in this paper was handling and incorporation of all types of changes to already developed RS applications. The second step of the proposed framework identified different types of changes that occured to already developed remote sensing applications, their incorporation in the developed applications, and later on their tracing. Note that we store d all types of changes with time dimension, and later on they could be retrieved, traced and manipulated the stored changes temporally.

Future Directions: We actively worked for proposing a software development methodology for remote sensing and its automation. A set of algorithms and testing of methodology has to be done for different case studies. A framework is a first step for proposing a methodology to be used as a formal way for the devolvement of remote sensing applications. Replication and provenance techniques have been used successfully to handle the problems of data dissemination in Web-based data intensive domains such as e-commerce and bioinformatics. However, these techniques have not been adapted by mainstream remote sensing community. With the increasing reliance of E-systems on remote sensing data and remote sensing applications, we felt that replication and provenance techniques must be adapted by remote sensing community if it was to handle the data dissemination issues and problems.

REFERENCES

Ahmad, T. and A. Shah. Remote sensing data dissemination and management: potential of replication and provenance techniques. AJS, 6(7): 188-198 (2010).

Anthony, I. Information system design methodology. JASIS, 31(1): 1-64 (2007).

Bagchi, S., G. Kar and J. Hellerstein. Dependency analysis in distributed systems using fault injection: Application to problem determination in an e-commerce environment. 12th International workshop on distributed systems: operations and management DSO'2001 Nancy, France, (2001).

Brian, H. Human-computer interaction-whence and whether? Interacting with Comp., 43(11): 970-986 (2009).

Brown, P.J. and G.J.F. Jones. Context-aware retrieval: exploring a new environment for information retrieval and information filtering, Personal and Ubiquitous Computing, 5(4): 253-263 (2001).

Fang, Z. E-Government in digital era: concept, practice, and development school of public administration. National institute of development administration (NIDA), Thailand, Int. J of Comp., The internet and management, 10(2): 1-22 (2002).

Josh, U. Epidemiological GIS: Understanding emerging critical issues, American J of Geo. Info. Sys., 1(2): 29-32 (2012).

Longley, P., F. Michael, J. David and W. David. Geographical information systems and science. 2nd edition, Published by John Wiley and Sons, USA: 3-7 (2005).

Patrick, V. Parallel database systems: Open problems and new issues. Distributed and parallel databases, Springer. Netherland, 1(2): 137-165 (1993).

Shah, A. A framework for the prototype-based software development methodologies, Computer &

information sciences, J. King Saud University, 13: 105-125 (2001).