Effective Utilization of ERDI Algorithm Based on Software Prototyping

1Dr. G. Guru Kesava Dasu, 2Dr. P. Bala Krishna Prasad
1Professor and HOD, 2Professor and Principal.
1,2Dept of Computer Science & Engineering
1,2Eluru College of Engineering & Technology, Eluru, W.G (Dt), AP, India.

Abstract:
Rapid prototyping is a broad field encompassing many domains and varied approaches. This article presents our experiences using user-centered rapid prototyping approaches for developing display-based ubiquitous systems, to be deployed over the longer term, in ‘real world’ situations. In contrast to the usual use of rapid prototyping to produce a single ‘proof of concept’ demonstrator to investigate technical feasibility, we have found that without investigation of real use, technical feasibility can be meaningless. So, to overcome this problem we have introduced ERDI algorithm as it will investigate the technical feasibility so that it would be helpful for the future researches to select the particular type of prototype techniques for future projects.

Key Words: Rapid prototyping, technical feasibility and ERDI algorithm.

I. Introduction:

Software Engineering:
Software Engineering is the study of design, development and maintenance of software. In other words it is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software and an engineering discipline that is concerned with all aspects of software production.

Communication skills, team dynamics, working with a “customer,” and creativity are also important factors in the software engineering. Software engineering can broadly be split into the following steps:

• Requirements say what the software should do.
• Software design is usually done on paper. It says what the different parts of the software are, and how they talk to each other.
• After the design phase is done, each component (part) of the software is coded. Code is what tells the computer exactly what to do at each step.
• Testing is done to see if the components meet the requirements and that the system as a whole meets the requirements.
• Part or all of this process can repeat if bugs are found or new requirements are needed.

Objectives:
• Understand the software processes.
• Able to create and understand Unified Modeling Language (UML) diagrams.
• Able to meet software requirements.
• Know how to use design patterns and frameworks.
• Familiar with unit testing and refactoring.

Software development process:
A set of activities that leads to the production of a software product is known as software process. Computer-aided software engineering (CASE) tools are being used to support the software process activities. However, due to the vast diversity of software processes for different types of products, the effectiveness of CASE tools is limited. There is no ideal approach to software process that has yet been developed. Some fundamental activities, like software specification, design, validation and maintenance are common to all the process activities.

A software development is also known as software development life cycle (SDLC). It is a term used to describe a process of analysis, planning, design, development, deployment and implementation of an application.
The above diagram says about the steps that occur in creation of software as mentioned above it have 5 stages. So this process will be done continuously for the obtaining of the best software.

- **Systems analysis, requirements definition**: Defines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- **Planning**: Planning is an objective of each and every activity, where we want to discover things that belong to the project.
- **Implementation**: It is the part of the process where software engineers actually program the code for the project.
- **Testing**: It is an integral and important phase of the software development process. This part of the process ensures that defects are recognized as soon as possible.
- **Deployment**: It starts directly after the code is appropriately tested, approved for release, and sold or otherwise distributed into a production environment. This may involve installation, customization (such as by setting parameters to the customer’s values), testing, and possibly an extended period of evaluation.

**II. Related Work:**
Cheverst, K., D. Fitton stated that within the field of ubiquitous computing, many of the issues related to the notion of ‘situated interaction’ remain very much under explored. So, here we describe the development, deployment and initial evaluation of the Hermes system, a system that comprises a collection of small interactive display units placed outside a number of offices within a University department. This will be exploring the both public and private properties. To date, the development of the system has been guided by the principles of participatory design and our intention is that the use of the Hermes system will continue to evolve over a longitudinal period of time as it is used on a day-to-day basis by university staff and students alike.

K. Clarkestated in this paper that presents some early design work of the ‘Digital Care’ project, developing technologies to assist care in the community for user groups with different support needs. Our focus is on developing a SMS Public Asynchronous Messenger (SPAM) system for SMS messaging to a situated display in hostels for ex-psychiatric patients run by a charitable Trust. Such settings pose both methodological and design challenges.

G., Sheridan stated that one of the most promising possibilities for supporting user interaction with public displays is the use of personal mobile phones. Furthermore, by utilizing Bluetooth users should have the capability to interact with displays without incurring personal financial connectivity costs. This paper describe the findings of an exploratory study involving our Hermes Photo Display which has been extended to enable users with a suitable phone to both send and receive pictures over Bluetooth. We present both the technical challenges of working with Bluetooth and, through our user study. We present initial insights into general user acceptability issues and the potential for such a display to facilitate notions of community.

H.E. Byun stated that it is important that systems that exhibit proactive behavior do so in a way that does not surprise or frustrate the user. Consequently, it is desirable for such systems to be...
both personalized and designed in such a way as to enable the user to scrutinize her user model. This article describes on-going work to investigate the design of a prototype system that can learn a given user’s behavior in an office environment in order to use the inferred rules to populate a user model and support appropriate proactive. We explore the tension between user control and proactive services and consider issues related to the design of appropriate transparency with a view to supporting user comprehensibility of system behavior. The evolution of the system has been guided by feedback from a number of real-life users in a university department. A questionnaire study has yielded supplementary results concerning the extent to which the approach taken meets users’ expectations and requirements.

C Saslis-Lagoudakis stated that we describe a new method for use in the process of co-designing technologies with users called technology probes. Technology probes are simple, flexible, adaptable technologies with three interdisciplinary goals: the social science goal of understanding the needs and desires of users in a real-world setting, the engineering goal of field-testing the technology, and the design goal of inspiring users and researchers to think about new technologies. We present the results of designing and deploying two technology probes, the messageProbe and the videoProbe, with diverse families in France, Sweden, and the U.S. We conclude with our plans for creating new technologies for and with families based on our experiences.

Bederson stated in this paper we investigate the use of scene graphs as a general approach for implementing two-dimensional (2D) graphical applications, and in particular Zoomable UserInterfaces (ZUIs). Scene graphs are typically found in three-dimensional (3D) graphics packages such as Sun’s Java3D and SGI’s OpenInventor. They have not been widely adopted by 2D graphical user interface toolkits.

To explore the effectiveness of scene graph techniques, we have developed Jazz, a general-purpose 2D scene graph toolkit. Jazz is implemented in Java using Java2D, and runs on all platforms that support Java 2. This paper describes Jazz and the lessons we learned using Jazz for ZUIs. It also discusses how 2D scene graphs can be applied to other application areas.

III. Existing System:

Rapid prototyping is an approach used in many domains and one that appears highly suited to research in the area of ubiquitous computing [8]. This approach primarily comes from manufacturing/engineering industries, where the process is the relatively fast physical fabrication of a design or concept for purposes such as demonstration, evaluation, or testing. Rapid prototyping is also used in software engineering, where requirements are uncovered through feedback from analysis of prototypes provided early in the development process. When developing ubiquitous computing systems, which often include both hardware, software and (of course) human factors, these two processes form an ideal design and development methodology. The goal of much work on display-based ubiquitous systems (and rapid prototyping in general) appears to be more motivated by the production of proof of concept demonstrators, usually in order to gauge technical feasibility and limited initial user feedback, for example[9]. However, in our work we have found that often it is equally important to investigate factors such as use and appropriation and that in some cases, without user studies, technical feasibility can be meaningless [10].

Rapid Prototyping has also been referred to as solid free-form manufacturing, computer automated manufacturing, and layered manufacturing. RP has obvious use as a vehicle for visualization. In addition, RP models can be used for testing, such as when an airfoil shape is put into a wind tunnel. RP models can be used to create male models for tooling, such as silicone rubber molds and investment casts. In some cases, the RP part can be the final part, but typically the RP material is not strong or accurate enough. When the RP material is suitable, highly convoluted shapes (including parts nested within parts) can be produced because of the nature of RP.

The basic methodology for all current rapid prototyping techniques can be summarized as follows:

1. A CAD model is constructed, then converted to STL format. The resolution can be set to minimize stair stepping.
2. The RP machine processes the .STL file by creating sliced layers of the model.
3. The first layer of the physical model is created. The model is then lowered by the thickness of the next layer, and the process is repeated until completion of the model.
4. The model and any supports are removed. The surface of the model is then finished and cleaned.

IV. Proposed System:

ERDI algorithm:

We have developed a software prototype of the effective research direction instructor System. It is a web-position software running on Microsoft Windows 2000/XP that is interfaced with a simulated web ability gateway. The core route way planning methods is written in C#. The ASP .NET framework is used to develop the web interface so that the prototype is portable to real e-ability gateways that comply with the framework. The simulated web-ability gateway contains a Microsoft Access database that stores test data sets, i.e., individual alumni’s background and e-ability history. The overall architecture of the prototype is depicted.

Algorithm:

Step1. start  
Step2. get Trainees current node  
Step3. for each node in local routing table  
Step4. if(CP>cut off)  
4.1 add to roulette wheel table  
Step5. Else goto step no.3  
Step6. if(next node is available)  
6.1 move to next node  
6.2 goto step no.3  
Step7. else if(random selection (roulette wheel))  
7.1 sort AO of nodes in roulette wheel table  
7.2 compute cumulative probability of nodes  
7.3 generate random number and select corresponding node  
7.4 recommend nodes  
Step8. Else  
8.1 list all the next node in the order of CP  
8.2 goto step 7.4  
Step9. Exit

The simulator consists of an inference engine, a controller, an inference engine, a database and an e-ability Gateway simulator. Driven by the effective research direction instructor algorithm, the inference engine computes pheromones and recommends ability routes.

The event-driven controller provides the functionality of data feeding, alumni selection and parameters control (training size, threshold values, etc.). The course data and the Trainee performance data are fed into the database which will be retrieved by the inference engine for further computations. The controller also acts as a communication hub among the other modules.

V. Experimental results:

The above diagram says about the e-ability gateway simulator was designed with the purpose of simulating the integration between the e-ability system and the effective research direction instructor. The simulator only feeds the test data into the inference engine, collects new Trainees’ preferences and displays the recommended ability route way or the next ability node. The test data consists of the course structure and alumni’s performance records. It accepts the user input and would be redirected to effective research direction instructor for recommending the best route way. Currently, we have validated and verified the prototype with simulated data. There are two different categories of test cases: (a) where number of “similar” (to the target Trainee) alumni << preset effective research direction instructor training size; (b) where number of “similar” alumni >> effective research direction instructor training size. As an illustration, consider the following test cases that are subjected to the authoritative and inductive route ways.

- Authoritative Route way – ability object IDs (nodes):
  - 1 → 2 → 3 → 4
  - 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12

- Inductive Route way: 1 → 2 → 5 → 6 → 12
• Test Case X (where number of “similar” alumni << training size)
  • Training Size = 500
  • Training threshold = 0.4
  • Test Set = 300 “similar” alumni + 200 “random” alumni
  • Eventual recommended route way: 1 → 2 → 3 → 4

  • When the effective research direction instructor unit is activated route way becomes: 1 → 2 → 3 → 5 → 6 → 7 → 9 → 10 → 11 → 12

  • Test Case Y (where number of “similar” alumni > training size)
  • Training Size = 00
  • Training threshold = 0.02
  • Test Set = 300 “similar” alumni + 200 “random” alumni

  • Genuine advised route way: 1 → 2 → 5 → 6 → 12 → 14

VI. Conclusion:

The use of prototyping has been critical to our user centered and multidisciplinary exploration of interactive display-based systems. In addition to using traditional approaches such as questionnaires, semi-structured interviews etc. we also received valuable information by incorporating logging functionality into the prototypes themselves (as well as the final deployed system). In this respect, the deployed prototypes have acted as technology probes. Typically the requirements of a given prototype vary throughout the development process and prototyping strategies should reflect this.

So, we use ERDI algorithm so as to overcome the problem of exploring the as it will investigate the technical feasibility so that it would be helpful for the future researches to select the particular type of prototype techniques for future projects.

VII. References


[7]. Rapid Prototyping and User-centred Design of Interactive Display-based Systems by Dan Fitton, Keith Cheverst, Chris Kray, Alan Dix, Mark Rouncefield and George Saslis-Lagoudakis.


Prof. G.Guru Kesava Dasu received M. Tech (I.T) Degree in 2003 and received M.E (C.S.E) Degree from ANNA University in 2008. He is having 12 years of Teaching Experience in various Engineering Colleges. At present he is working as Professor & HOD, Department of CSE, Eluru College of Engineering & Technology, Eluru. He is a certified professional of IBM Rational Rose from Rational University.

Dr. P. Bala Krishna Prasad is the Principal of Our Eluru College of Engineering & Technology, Eluru. He obtained Doctoral Degree in Engineering, Ph.D(CSE) from Acharya Nagarjuna University, in the year 2007. He received his Master’s Degree in Engineering, M.Tech (CSE) from Andhra University in the year 2000. He completed his Bachelor’s Degree in Engineering, B.Tech (CSE) from IETE, New Delhi. He has a rich experience of 18 years which includes Teaching, Research and Administration. He served various reputed engineering colleges as a Principal for several years. He is a multi-tasking personality and a go getter. His relentless efforts and commitment in Institutional Administration lead few colleges to reach greater heights. He published several technical papers in National and International Journals of repute. He is a most coveted author for Computer Science and Information Technology subjects and has 7 reputed text books in his authorship. Under his able guidance several research scholars are pursuing their Doctoral and Pre Doctoral Programs in various Universities.