Integration of Bug Tracking System in Distributed Environments

Kalin Krasimirov Rashev, Ivan Haralampiev Furnadjiev, Dimiter Tenev Tenev, Vasil Janov Vassilev
Department of Computer Systems, Department of Electronics, Technical University of Sofia, 1756 Sofia, Bulgaria, kalin@lark.vmei.acad.bg

Abstract. The paper describes the need and advantages of integration bug tracking system in distributed design environments for improving design team coordination and speedup their work on common projects. It also discusses the developed web-based bug-tracking prototype tool incorporated as a module in the web-based system for collaborative FPGA design. Written in Java Server Pages and database technologies bug-tracking tool can be accessed from anywhere, anytime, and any platform through a Web browser.

1. Introduction

The development of the Internet, networking and data processing technologies, together with increasing complexity of the data and electronics design process has provided many needs and opportunities for wide-area collaboration in complex systems design. And because development teams are not necessary centralized support of distributed teams is needed. The web-based system for collaborative FPGA design is created to syncronize geographically separated team members in their work on common projects, by giving them an integrated design environment [1]. The system provides to the geographically distributed developers all tools for design and communication.

Nowadays design complexity increases among with shortening development time. The contemporary requirements for fast research and development of wide area projects force team managers to find a way to improve design team productivity. This time to market pressure requires higher quality control in order to ensure fault free products on the market.

Bugs are part of every product development process. Bugs that are found but not properly tracked might slip away and be discovered by the customers on the market. How to track the bugs found during product development cycle is very important task for quality insurance. Integrating bug tracking system as a part of project development and testing process will increase quality of the final product and reduces to some extent the research and development time.

To answer these requirements a Web-based bug-tracking tool is developed and integrated as a module in the distributed design environment for FPGA design. Integrating bug tracking to the distributed environment helps in coordinating workflow in any development project, be it software or hardware. The bug tracking tool synchronizes developers and testers by keeping track of bugs through stages in development process so helping in management of complex project. It also will improve productivity and the final product quality.

2. Architecture of the FPGA Distributed Environment

The architecture of implemented FPGA collaborative design environment is shown in Figure 1. At a macro level it follows a three-tier model consisting of client-tier, server applications and a database.

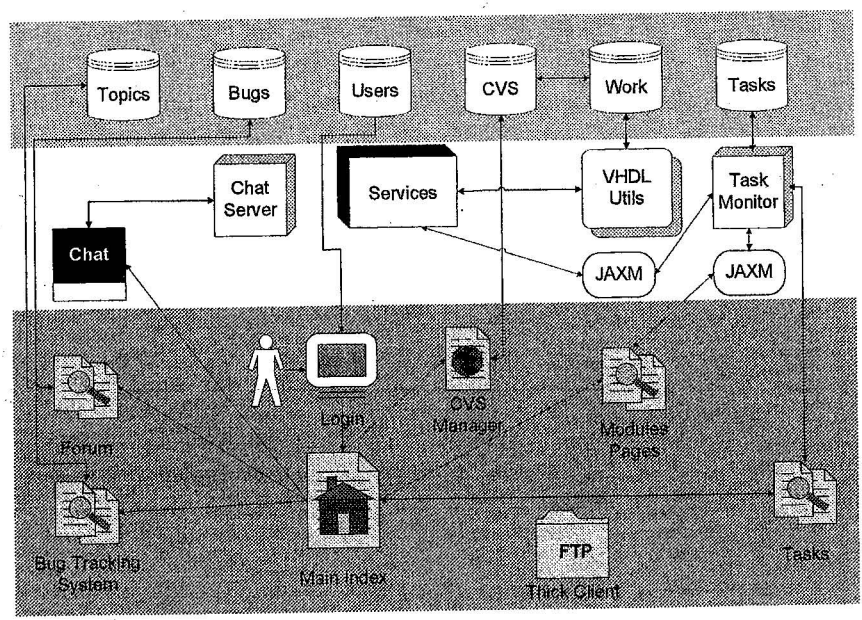

Figure 1. System architecture

The client side of the system is developed as Web site, which gives users access to the modules via Internet. On a client machine a browser provides the user interface. Application server provides the application logic for bug tracking, and handles all requests for data by communicating with the data services tier. Project data is stored in a database.

A major advantage of the distributed system is its module structure, which allows high scalability and easy maintenance. The interface between modules in the system is accomplished by the Internet protocols TCP, UDP and HTTP. Every module sends information to its corresponding server application(s) – in most cases implemented as Java servlet(s). The servlet uses a database to get or post the requested information, process the data, and send the response back to user.

Benefits of the three-tier model are:

- Application logic is centrally maintained on a Web server, and can be easily updated
- A browser handles all presentation/user interface services, so the system requirements for the client machine are low.

• Clients do not communicate directly with the database, but through the application tier. This reduces the number of database connections, and therefore the load on the database server.

3. Bug Tracking System Integration

The bug tracking system is implemented as a separate module in distributed environment. Figure 2 shows the integration of developed prototype bug tracking system in the whole environment. On the client side the bug tracking system is implemented as bug tracking pages. It is accessed from distributed environment's main page by hyperlink. The bug tracking pages provide access to all functions, implemented in the developed system. The connection to the bug database is established by using JDBC protocol.

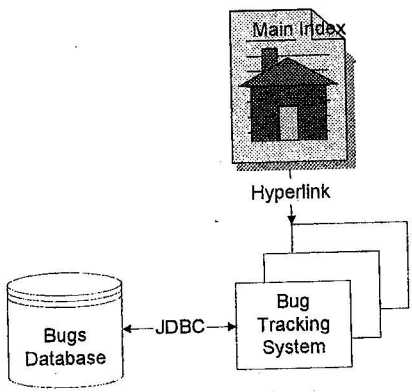

Figure 2. Integrating the Bug Tracking System in the entire environment.

The main functions of the developed bug tracking system are to:

• **Submit a new bug.** When a bug is discovered, the team member registers it in the Bugs Database by filling a specific form.

• **Assign the bug to developers.** The system automatically assigns bugs to the head project developer. Any further assignments can be made by team members, responsible to do that.

• **View reports about bugs.** On request, Bug Tracking Database provides up-to-date information any time to any staff member.

• **Keep track of bug state.** This process is the core of bug tracking. After a bug is opened it "lives" in the database, going through several modifications, until it is closed.

4. Principles of Bug Tracking

When the bug is detected it should be registered in the database. Bug reporting is a single operation, which is performed for bug registration. With this step a bug tracking process is activated. It involves all steps from bug reporting till bug debugging and closing. The bug lifecycle model (figure 3) includes all the different states a bug passes through, during its lifetime. It includes bug registration/ activation (the first step of bug tracking process), bug assignment, debugging, testing and finally bug closing. The process is iterative and passes through several reassignment, reactivation and solution steps.

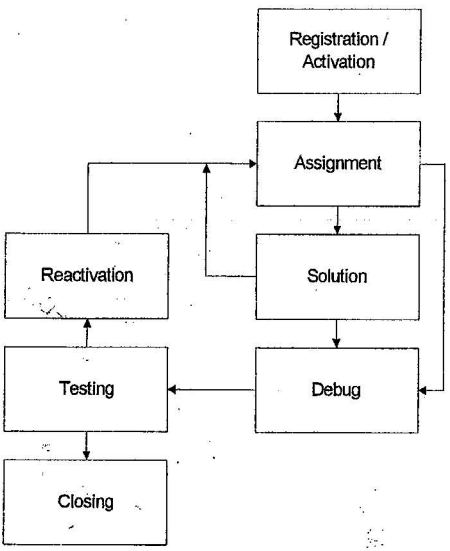

Figure 3. Bug life cycle

Detected bug should be properly described during bug **Registration/Activation** phase in order to facilitate further operations of bug debugging and fixing. At this stage a bug registration form [2] is filled in. This form includes three fields, which should contain detail information of:

1. How to reproduce the observed bug. This is a step-by-step explanation of all actions performed when the bug appears. Test engineer have to type exactly what he did to demonstrate the problem and complete text of any error messages (if any).

2. What is expected to happen – Here information of fault-free behavior of the product is written according specification.

3. What happened instead - A description of the incorrect behaviour is given and all observed faults are reported.

After the bug is properly registered its status is set to "Activated" and it should be **assigned** to somebody (usually head developer) to eliminate it. When the assignee finds a **solution**, he can **debug** himself, or reassign the bug to somebody else. The **solution-reassign** steps can be done many times before the actual debugging. After the bug is fixed, its status is changed and the product should be **tested again**. If the bug is really fixed and does not appear any more in the repro steps the person, who opens the bug **closes** it. If the bug is still present the test engineer **reactivates** it and the bug is returned to the head developer.

5. Example

The prototype of the bug tracking system is designed as a set of web pages, providing access to all bug tracking functions. There are input data pages for bug reporting and changing bug state. These pages are designed as simple as possible, because if user needs to fill very long input data forms, the bug reporting process will become too complex. View all bugs and bug details are pages for getting information from bug tracking database. The database is organized in a manner to facilitate customisation and incorporation of different report templates.

Figure 4 shows an example of bug tracking database organisation. The main table ("Bugs") contains all data for bug tracking, while "Statuses" and "Priorites" are supporting tables. They contain lookup values for bug statuses and priorities.

The database is designed with Microsoft® Access™. The JSP pages and servlets, which are used to communicate with the database, are stored in Apache Tomcat Server.

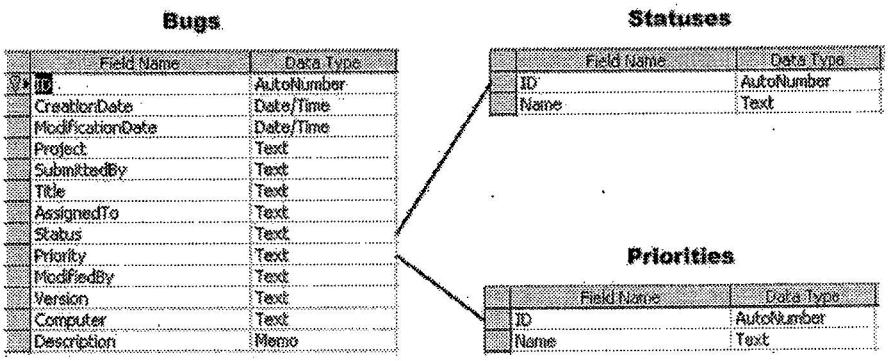

Figure 4. Example of Bug Tracking Database Organisation

6. Benefits of Web-based bug tracking

Integrating bug-tracking tool to the distributed design environments helps in managing complex products and improves productivity. It also improves communication and teamwork. All team members can access the bug-tracking system from anywhere, anytime, and any platform. All they need is a Web

browser. Everybody on the team, developers, tech writers, support, quality assurance, and project managers can participate, collaborate, and keep track of the defects assigned to them. So everybody can contribute to the development, testing, and release of the final products. Developed Web-based bug-tracking system

- Is easy to install, use and administrate. It is fast, reliable, and lightweight
- Is platform and database system independent (Java and J2EE)
- Supports multiple projects and access control
- Has intuitive user interface (no learning curve)
- Has completely customisable workflow (bug lifecycle)
- Provides any time access from any computer with an Internet connection

7. Conclusion

The developed web-based bug tracking tool allows development teams in different locations to share bug and defect data 24 hours a day from any computer regardless of platform or operating system. It provides a vital communications link between team members. All bug data is stored in a centralized database, creating a searchable, sortable, historical archive from which development teams can identify urgent and fix related bugs. The bug tracking provides a significant improvement to the quality assurance process. It allows higher-quality products to be delivered faster, reduces project costs, and improves customer satisfaction.

The bug tracking system is used in the pilot project-based learning course in ASIC Design within a framework of the Netpro II EC Leonardo De Vinci Project.

References

1. Furnadjiev, V. Vassilev, K. Rashev, D. Tenev, Web-based system for collaborative FPGA design, TU-Sofia, 2002
2. Joel Spolsky, Painless Bug Tracking, www.joelsoftware.com, 2000