

An XML-based Collaborative Framework for ASIC eDesign

Amir Hekmatpour
IBM Microelectronics
ASIC Development
Research Triangle Park, NC, U.S.

Azadeh Salehi
IBM Microelectronics
ASIC Development
Research Triangle Park, NC

James Coulter
IBM Microelectronics
Processor Development
Research Triangle Park, NC

Abstract

Efficient management of the large number of expertise and services required for design and verification of complex integrated circuits demands a collaborative design environment. Such an environment should be able to accommodate distributed design teams and design services, as well as heterogeneous design tools and data models. This paper describes a web-based framework for collaborative design of Application Specific Integrated Circuits (ASIC) and System-on-a-Chip (SoC). The XML-based framework provides a collaborative design environment for On-Demand conversion and secure exchange of heterogeneous design data and services. It enables sharing of data among EDA tools and applications by providing a common design schema. The framework provides integrated design data, design services and EDA tools across a unified XML standard. The central database provides access, control, collaboration, and sharing of expertise and design services across LAN, Intranet, or Internet. Secure design data is delivered across distributed design centers and out-sourced expertise and resources via IBM Customer Connect or standard HTTP/HTTPS tunneling protocols. A prototype of this framework called BlueIP is developed for on-Demand IP design specification, qualification and data exchange.

Keywords: ASIC eDesign, Collaborative ASIC Design, XML-based Design Framework, Web-based ASIC design, Distributed SoC Design, Collaborative Applications.

1 Introduction

Rapid improvements in integrated circuits (IC), particularly the advent of deep-submicron process technology, have led to the development of the complex Application Specific Integrated Circuits (ASIC) and System-on-a-Chip (SoC) - by which a complete, fully functional system is designed and integrated on a single silicon chip. The current trend towards the realization of even more complex ASICs requires the combined efforts and attention of experts in a wide range of areas including embedded hardware/software systems, IP cores, signal and image processing architectures, low power design techniques, design methods and CAD tools, test and verification, modeling, and timing issues. Additionally, the number of logic gates in an ASIC and the variety of Intellectual Property cores (IPs) available to designers are

growing at a much faster rate than the designer's ability to take full advantage of them in the current homogeneous design environments. To take full advantage of the potential the technology has to offer, today's ASIC and SoC designers require a variety of expertise, services, and tools. Figure 1 gives an overview of ASIC and SoC design processes and their major components and tasks.

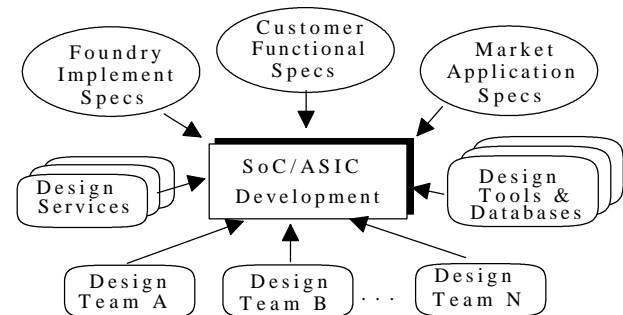


Figure 1. SoC/ASIC Design Process

In a majority of ASIC and SoC design projects, the design expertise and associated CAD tools may be in different corporate locations and, in many cases, across multiple corporations. Finding all the skilled candidates to complete specific design tasks in a single geographical location may not be possible. As a result, collaboration around the traditional white board within a single location has been replaced by remote tele-commuting among design teams, and completing design projects with such a geographically dispersed teams is quickly becoming common. As shown in Figure 1, different design teams with specific expertise may be assigned to each functional unit of the system. Several dispersed teams may provide tool support and expertise for models, HDL languages, and optimization techniques used in design and validation.

In response to increasing design complexity and the demand for faster turn around and lower cost, many companies are outsourcing design tasks such as IP design and optimization, synthesis and timing, physical design, or verification. However, outsourcing means that design teams have to collaborate and share design information, libraries, and even design automation tools amongst themselves (Figure 1). Such collaboration is not feasible with the traditional communication methods (i.e., phone, fax, email, mail, or express mail) and traditional homogeneous design frameworks. Relocation of design

teams to a single location and the black-box data conversion methods among tools are also impractical and extremely costly.

There is continued pressure to decrease the design cycle, design cost, and Time-To-Market, and to improve quality. Therefore, a true on-time collaborative design infrastructure is required where the geographics and language barriers, as well as style differences are managed transparent to the design and design teams. In this paper we introduce a new and flexible design framework which can support a number of globally dispersed teams, provide seamless integration for heterogeneous design data models and tools, and provide a secure mechanism for sharing design tasks with third party design services .

2 Related Work

EDA tools have been evolving from stand-alone utilities to integrated tools to CAD frameworks to integrated network-based environments [1]. These networked design environments provide various mechanisms for tool integration, configuration, and collaborative team support [2]. Some platforms have reported using XML-based data exchange over Web-based access to tight network integration [3][4]. These design frameworks are usually for a specific task and based on in-house design tools. The framework described in this paper utilizes XML not only as a carrier of design data but also as an open standard mechanism for on-demand design data conversion and design rules generation and exchange between heterogeneous design tools and utilities.

Several reported design frameworks use web-based modeling and analysis interfaces [3][5], on-demand simulation and design utility access [6], and loosely integrated design flows [2]. These frameworks provide improved productivity and they take advantage of web-based access and interaction, but they do not address the problem of heterogeneous design data and tools. It is also not clear whether the web-based application delivery is suitable to all ASIC design tasks. The integrated methodology described in [6] is suitable for a well defined and optimized design flow utilizing predefined tools, HDLs, and design techniques. It is not suitable for distributed design tasks. Collaboration techniques described in [7] are more suitable to partitioned sequences of design stages rather than asynchronous design tasks with real-time collaboration. Other similar log-based collaborative design frameworks [6][8] provide a synchronous protocol for monitoring ongoing design activities to decide when a new task can be initiated, or whether a required resource is available. These architectures cannot accommodate dynamic and intelligent collaboration that is necessary to efficiently manage today's complex ASIC design flows.

None of the above systems and methods provide a solution for efficiently designing an ASIC when the design teams are geographically dispersed and are utilizing different tools, standards, and formats. The system and method described in this paper thus recognizes that it is necessary to provide a system for enabling collaborative design and verification of ASICs by geographically dispersed design teams. Such a system should also enable teams to utilize heterogeneous tools and design data models, and enable true collaboration among different design teams, their applications, HDL specific methods and design databases. The remainder of this paper will describe the architecture, data model and some of the major features of the proposed system that address some of the shortcomings described above.

3 An XML-based Design Framework

The framework should integrate various design automation tools utilized in SoC design and provide the features that enable these tools to be accessed and utilized effectively via the Internet. An analysis of the applications involved and the steps entailed in the design flow is first completed. IC design tasks, which are less process intensive (such as layout), can be efficiently accomplished remotely via the Internet. Simulation or routing, which are process intensive are better suited to be performed locally. The proposed design framework is architected such that it provides local (i.e., client-side), remote (i.e., design center-side or server-side), as well as peer-to-peer processing (P-to-P) processing. An online knowledge-base estimates the processing time/resources needed for each design task and makes a recommendation on the mode of design (i.e., local, remote, or P-to-P).

A very important aspect of the design environment is the timely sharing or dissemination of the new generated design data and updated design attributes across distributed teams and local design databases. The framework provides a secure mechanism by which various design data formats are encapsulated and delivered to other designers. This is handled by the various data manipulation functions which use XML to structure information extracted from the database so that the data can be distributed and published more easily.

3.1 XML-based Query and Data Presentation

Data integration solutions use XML to automate the exchange of data. In general, with an integration solution, XML serves as an interface layer or wrapper for data being passed between data sources, making it possible for a wide variety of applications, legacy systems, and databases to exchange information. Virtual Database Management Systems (VDBMS) take data from many different sources and create a single relational database

interface to all the data sources. Afterwards, XML is utilized to map the various structures used by the data sources to a common syntax for easy access. Finally, a set of publishing XSLT rules (a set of transformation rules based on eXtensible Style Sheet Language (XSL) is provided to extract information on demand and publish the information. The resulting environment is an XML-based information repository that is able to update itself whenever the related data sources change. The associated SOCML data manager, data exchange, and data publishing are shown in Figure 2. Design attribute query and presentation on the client side are shown in Figure 3.

An important advantage of XML is its search enhanced features and data architecture. Unlike HTML search tools, which use keywords and text, XML-based search tools use data structure and meta data. The XML search tools check a query against a Design's DTD that contains tags matching what is being searched (e.g., content, intent, and use). The XML-based search tool provides a more efficient and accurate search capability and returns information containing exactly what the user is looking for. The returned information is formatted in its own industry standard format based on its XSLT style sheet transformation rules. The same design data can be mapped to different XSLT for different users.

Refining a traditional database query requires a complete round-trip session from the browser to the web server to the database and back again. Using XML, the query results are extracted from the database and may be further refined on the client device as shown in Figure 3. Some operating systems may require a special application or a browser plug-in, however, majority of operating systems' browsers provide full XML support.

3.2 XML-based Design Data Exchange

The XML-based platform allows exchange of data across the Internet or Intranet and facilitates development of web-based design applications. XML also allows design teams to create smart design documents for exchange across the framework and local or remote processing of the design data by a set of standard applications. Common design data and design resource management based on XML and a common design data and design resource presentation based on XSL are provided (Figure 4). A set of rules define the XML tags that are utilized for the structure, format, and content of design data components that are exchanged. The resulting extended XML definition is called ASICML (ASIC Markup Language), SoCML (System-on-a-Chip Markup Language), or IPXML (IP XML Markup Language).

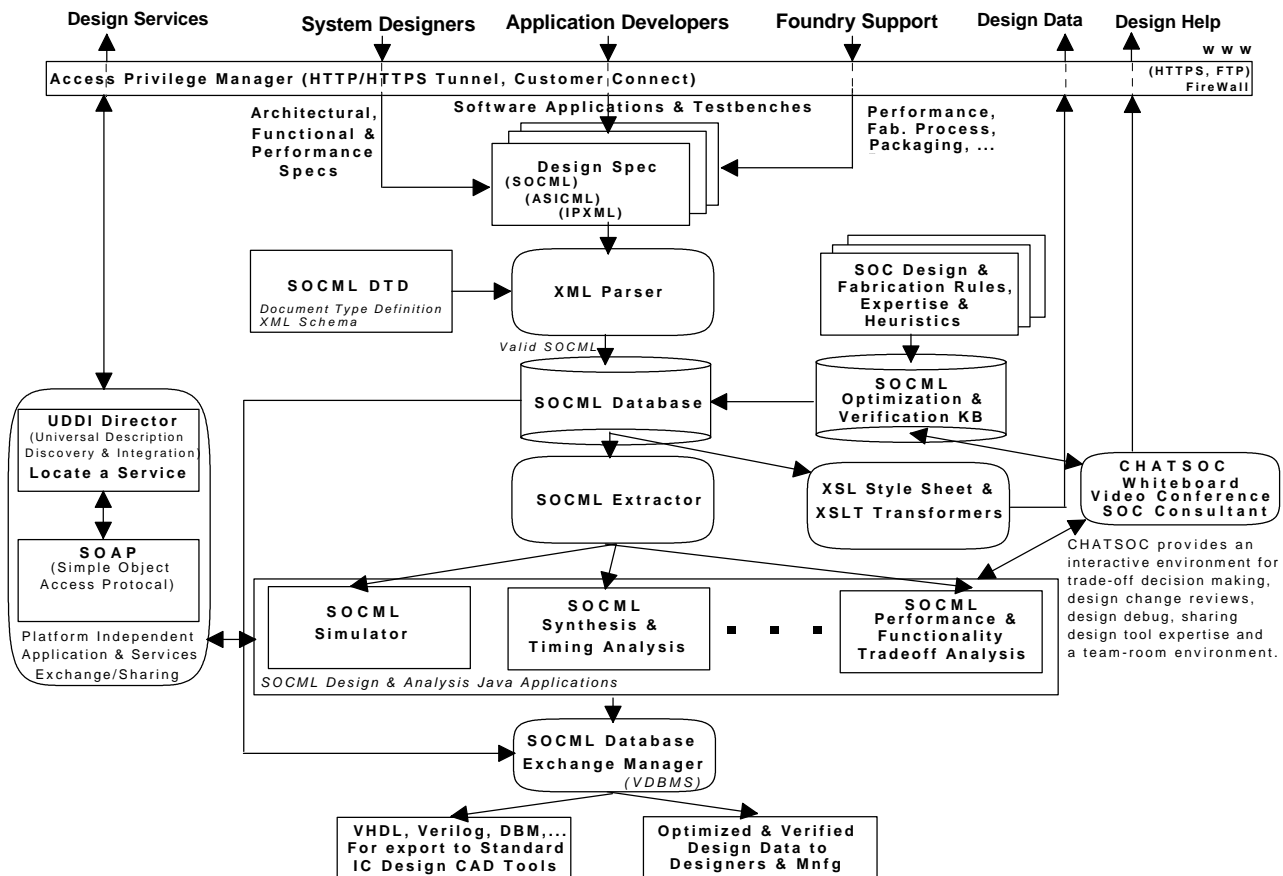


Figure 2: System Architecture and Functional Diagram of the Proposed Design Framework

SOCML structure and content constructs can be defined using Document Type Definitions (DTD) or XML Schema. The DTD provides a basic grammar for defining an XML Document in terms of the metadata that comprise the shape of the document. An XML Schema provides this plus a detailed way to define what the data can and cannot contain. It provides far more control for the developer over what is legal and it provides an Object Oriented approach, with all the benefits this entails. We should mention that XML Schema is a rather new technology and is not as widely accepted and supported as DTD. A majority of legacy systems and data warehouses are developed based on DTD. The current implementation of SOCML uses a DTD to define: (1) what elements may exist in a SOCML document; (2) what attributes the elements may have; (3) what elements can be nested (found inside other elements); (4) what order the elements should be defined in an SOCML; and (5) the XML design files adhering to SOC DTD that constitute SOCML.

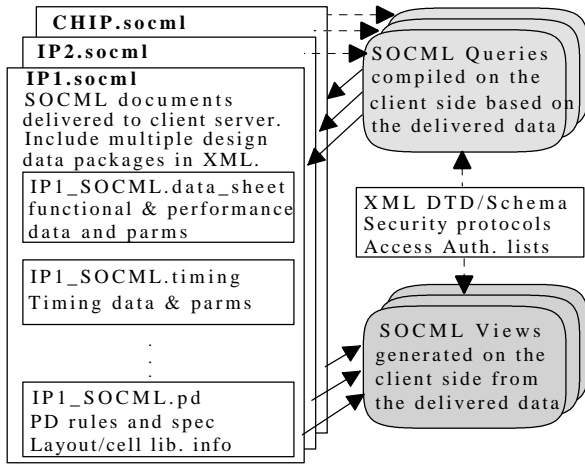


Figure 3. SOCML XML Schema, attributes and client-side query & view generation

A well defined schema such as SOCML provides several benefits: (1) it can enforce certain standards without restricting the exact content; (2) system designers and technology foundries can easily exchange design information based on SOCML and its DTD; (3) each side can develop applications based on this schema and verify its input/output against the SOCML DTD independent of the exact content, and (4) the content of each element of an SOCML can be generated by a CAD tool in a specific format as long as the document structure adheres to SOCML. Presentation and formatting of content of SOCML elements in each environment and for a specific design tool is governed by XSL attributes.

4 A Collaborative Design Framework

Collaborative systems include both general infrastructures and specific applications for supporting collaboration. Our approach to collaboration is similar to

the integrated infrastructure and application collaboration advocated by [10]. Our framework enhances the SoC and ASIC design methodology by replacing the after-the-fact manual collaboration of separate design teams with an integrated collaboration and exchange environment. It allows local and global collaboration and data exchange between the various design teams throughout the entire design process. As illustrated in Figure 2, the entire design process can be tracked over a LAN, Intranet, or Internet. The design process utilizes the XML Schema and its associated rules and utilities (e.g. ASICML, SOCML, IPXML) to allow cross-interaction and data exchange among design teams utilizing different tools and operating systems. The data format requirements for each design step, the associated tools and utilities, as well as data access and update authority protocols are disseminated and enforced by the system's access manager as shown in Figures 2 and 3. The framework is architected such that it can provide local (i.e., client-side) processing, remote (i.e., design center-side or server-side) processing as well as peer-to-peer (P-to-P) processing (Figure 4). The online knowledge-base estimates the processing time/resources needed and makes a recommendation on the mode of design (i.e., local, remote, or P-to-P). Mobile access to the framework can also be accommodated [12].

Depending on a project's specific requirements, design tasks, application development tasks, and all the necessary support and design services can be provided from different locations as shown in Figure 4. This scenario is a rather typical for fabless design and service companies. To accommodate this project three instances of the framework are instantiated at three different locations. The design and services group defines its own local design flow and access control protocols as shown in the upper left corner framework instance in Figure 4. The ASIC or SoC customer (or a hired third party vendor) will be responsible for application development and testing utilizing the simulation models generated by the system designers and performance models provided by the foundry (middle right framework instance in Figure 4). A central repository (MySQL database) and the associated data/services exchange manager (based on the ASICML, SoCML, or IPXML schema, and the XML, SOAP, UDDI protocol) provide the interaction between the three framework instances. The actual exchange mechanism across local firewalls and corporate networks is managed by the Access Manager implemented as an IBM Customer Connect or a custom HTTP/HTTPS tunnel.

5 Web-based Interface

The proposed system taps into the potential of the Internet to allow access from any geographical location and also taps into the functional features of XML to aid in the collaborative SoC design process. Design iterations

are reduced through enhanced communication and collaboration, and redundant effort is eliminated through design reuse and better library-management practices. From the design team perspective, any standard browser may be utilized to access the design framework and the Internet essentially acts as a common user interface by providing a common ground for the distribution and management of design information. This allows design data and requirements to be shared with a multitude of design teams and expertise in a variety of disciplines. Thus, package designers, system software designers,

marketing and project managers, etc. are able to make better decisions and planning based on actual design data, progress and status. The system architecture described in this paper provides a design and development environment that is consistent with extension of the Internet as an ideal medium for the design, marketing, sale, distribution and support of ICs, ASICs, and SoCs. Both wired and wireless interaction/access can be deployed [12].

The design framework provides collaboration and services/expertise delivery through the "Custom Help At

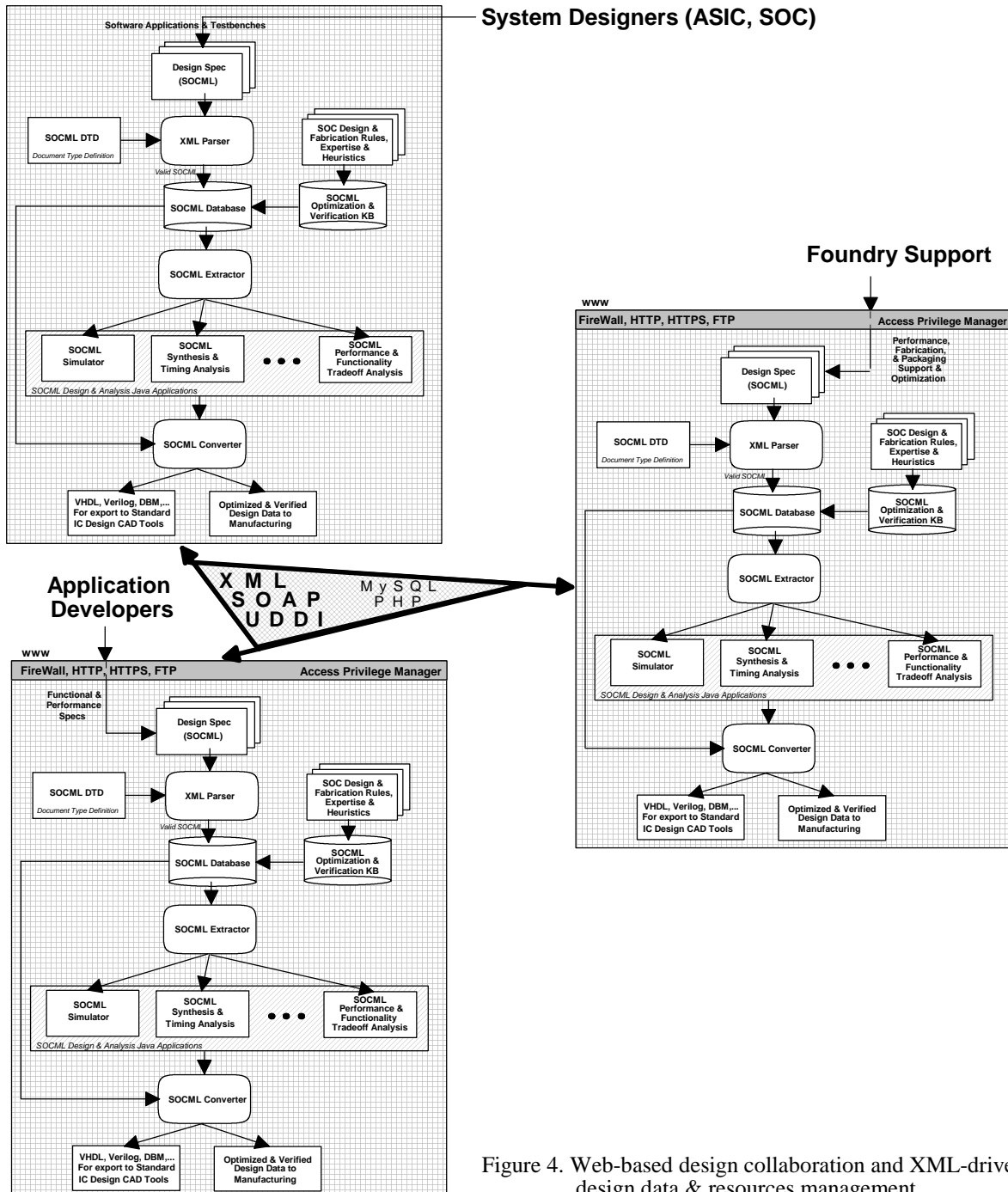


Figure 4. Web-based design collaboration and XML-driven design data & resources management

Terminal” (CHAT). CHAT includes dedicated online video conferencing between design teams, design team members and other personnel, Instant Messaging, Web Conferencing, etc. It provides each design team with local ownership of a particular task within the design, and the level of corporation with other design teams and the level of information sharing allowed. A design task may be divided into a number of sub-tasks. The design framework can then assign each sub-task to tools and services available within the design environment and assign process-intensive tasks such as simulation to fast computing resources available in the framework. Some of the features of CHAT are similar to virtual engineering features of Ford Collaborative environment [9].

6 Access Control and Management

ASIC and SoC design processes are made up of a hierarchy of specifications, design rules generation, design task applications (CAD tools), design resources and expertise. The specific flow and sequences may change from a project to the next but the overall design process components are common among most ASIC and SoC designs in a company. Secure access to tools and data, as well as secure transport of data is provided by a set of access protocol as shown in Figures 2 and 4.

The Access Privilege Manager monitors and controls access to the design framework by design teams, design team members, and other authorized personnel and groups. It maintains a control list of authorization parameters, user identification and passwords, access authorization level for each user and group, list of all groups to which a user belongs, specific group access authority and list of project administrators. Each user profile includes an associated role, wherein access to processes and designs is only granted to a user whose profile supports said access. The Access Privilege Manager provides a two tier security abstraction; role-based access and design object type enforcement. Each user profile includes an associated role. Access to processes and design objects are granted if the process object’s scope includes (or supports) the requester’s role. Biometrics security checks such as “facial scans”, “eye scan”, “fingerprints”, and “voice authentication” can also be integrated into the framework (future work).

7 Conclusions

We described the architecture and functional features of a design framework that provides a collaborative design environment where the proprietary design information can be shared and exchanged across multiple locations. Each team can define the format and content for the required deliverables. The deliveries are automatically checked and verified for correctness, completeness, construction and content. The design requirements are verified on the

sender’s side based on rules defined and agreed upon without the need for any inspection or intervention. Each team involved in the design process will perform certain tasks on specific design components and therefore will have specific access and usage authorization. Design components maintain a complete self-contained access and utilization log (i.e., who did what, utilizing which tool, when, and why, etc.). Each design team defines design tasks that their members are responsible for and decide which tasks will be performed on the client-side and which ones on the server-side. All security, data structures, and service protocols are managed by the framework.

8 References

- [1] E. Schwartz, S. Lawson, “Online collaboration revolutionizes design process,” InfoWorld Electric, Nov 8, 1999, www.infoworld.com
- [2] S. S. Leung and M.A. Shanblatt, “A conceptual framework for designing ASIC hardware,” 24th ACM/IEEE Design Automation Conference proc., 1987.
- [3] P. A. Wilsey, “Web-Based Analysis and Distributed IP,” Proceedings of the 1999 Winter Simulation Conference, P. A. Farrington, H. B. Nembhard, et. al. (eds), 1445-1453, December 1999.
- [4] V. R. Roussev, P. Dewan, et. al., “Integrating XML and Object-Based Programming for Distributed Collaboration,” Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, June 2000.
- [5] D. M. Rao, H. W. Carter, and P. A. Wilsey, “Optimizing Costs of Web-based Modeling and Simulation”, International Workshop on Internet Computing and E-Commerce (ICEC’01), April 2001.
- [6] A. Hekmatpour, R. Devins, D. Roberts, M. Hale, “An Integrated Methodology for SoC Design, Verification, and Application Development,” Proc. Of The International Embedded Solution GSPx Conf., September 2004.
- [7] J. R. Borck, “Closing the collaboration gap, SourceForge Enterprise Fusion and Merant Professional bring distributed development teams together,” InfoWorld, March 05, 2004, www.infoworld.com
- [8] G . Chung, “Log-based collaborative infrastructure,” Doctoral Thesis, UNC Chapel Hill, January 2002.
- [9] R. Riff, “Virtual Engineering, Real-Time Collaborative Design Tools at Ford,” URL=www.e-designcenter.info
- [10] P. Dewan, “An Integrated Approach to Designing and Evaluating Collaborative Applications and Infrastructures,” Computer Supported Cooperative Work, Kluwer Publishers, 2001.
- [11] IBM Technology Center, Bangalore, India, “on-demand e-design services for ASIC and IP Design,” URL=meadev.nic.in/ind-ter/for-med/IBM-launches.htm.
- [12] O. Omojokun, P. Dewan, “Experiments with Mobile Computing Middleware for Deploying Appliance UIs,” Proc. Of the 23rd International Conference on Distributed Computing Systems, May 2003.