Automated Status Report Management System

Peeyush Tugnawat\textsuperscript{1} and Mohamed E. Fayad, Ph.D.\textsuperscript{2}

\textsuperscript{1}J.D. Edwards and Company
One, Technology Way, Denver, CO 80237
peeyush_tugnawat@jdedwards.com

\textsuperscript{2}Computer Engineering Dept., College of Engineering, San José State University
One Washington Square, San José, CA 95192-0180
m.fayad@sjsu.edu

Abstract
A status report is a way to convey the details of the work done during a particular period of time. In a business context, it helps to keep track of employee time and progress on the tasks assigned to them. Conventional status reports in a corporate environment are text documents with a defined format that are sent to the manager and/or team leader. It takes additional effort to create, edit, store and retrieve information using these text documents. ASRMS is intended to automate this process by eliminating the use of text documents. ASRMS provides an interface to fill in the status data for employees. The data is stored in a database for convenient retrieval. ASRMS provides powerful mechanisms for generating custom status and project reports easily.

Keywords
ASRMS, Process Improvement, and Risk Analysis

1. INTRODUCTION

Almost all corporate organizations utilize status reports as a means to provide information updates about tasks and projects that employees have been working on for a specified period of time. This has become a standard business practice used for keeping track of employee time and progress on the tasks assigned to them. Employees send status reports to managers through email or hard copy on a weekly basis. Information in the status reports usually include the time spent, job status, dates, and plans for the next week for the projects and tasks. Usually the status reports are text documents—for example, MS Word, PDF, etc., depending on the company and the team. Managers examine the status reports sent by the employees and thus keep track of the combined project status and time. Streamlining and automating this process can reduce the effort expended by employees and especially managers.

2. MOTIVATION

Conventionally, it requires a lot of effort to keep track of relevant status information, store status reports, and retrieve past information. The main motivation in building this application is the production and storage of structured data. Its advantage is the improvement of the process by automating the input and storage mechanisms.

3. DESIRED ASRMS
The desired Automated Status Report Management System (ASRMS) should provide a complete automated infrastructure for entering, storing, managing, and presenting status reports. It will feature a composite status entry interface (CSRI) through which users can enter their status information conveniently without worrying about the formatting and style. Information gathered is stored in a central database. Dynamic status reports can be generated easily with the report-generating interface (RGI), which allows intense customization in terms of output format, time periods, and desired information. Other features will include the ability to graphically analyze the status data, compare past status reports, and classify and log the major activities of ASRMS. The user interfaces of ASRMS, i.e. CSRI and RGI, should be interoperable. They should be able to run on all software and hardware platforms. ASRMS should provide remote access features. It should implement a request-response model to accommodate for external requests apart from RGI. Remote requests from small and resource-constrained internet devices should be processed using this request response model. Responses should be customized based on the capabilities of the requesting device. ASRMS should have risk analysis capabilities based on the status data stored in the database. It should be able to forecast and categorize risks based on the severity associated with them. ASRMS provides interfaces for data mining and risk measurement operations performed by external preexisting tools.

4. SYSTEM COMPONENTS

Based on the requirements mentioned in the detailed requirement section, ASRMS should have the following components:

![ASRMS System components](image)

**4.1 Composite Status Reporting Interface (CSRI)**
The CSRI provides users with an input interface for submitting their status report information. The input forms are generated dynamically and are customized based on the user ID. The CSRI looks for information pertaining to the corresponding user ID and presents the user with an input form containing pertinent information to choose from and fill in.

**4.2 Data Structuring Component (DSC)**
The DSC provides a means to structure the data retrieved from the database. It generates a standard XML document containing status information. Different standard structures are predefined in a DTD based on the report generation options. The DSC also validates the generated XML document with the corresponding DTD.

**4.3 Intelligent Presentation Component (IPC)**
The IPC generates dynamic status reports based on user-specified style and criteria. Structured XML documents generated by the DSC are transformed to customized status reports. Standard templates and transformation rules are defined for transformation of XML documents to produce the desired output report.
4.3 Request-Response Handler (RRI)
The RRI handles and manages the requests by the RGI. Requests are classified as local and remote. Local requests are processed with the default XML and stylesheet preferences set for the IPC. All requests from the computer systems within the connected network of ASRMS are considered local. Remote requests are received over the Internet from computers or resource constrained devices such as cell phones, or handhelds. Remote requests are processed by first determining the capability of the requesting device and then formatting the return response data accordingly. RRI uses IPC and DSC to provide customized transformation of data for the remote requests.

4.4 Report Generation Interface (RGI)
The RGI provides the ability to customize the status reports based on various factors and criteria. Users can specify the style and format of the status report. Users may choose from existing templates or specify their own template rules for the output. The language for specifying rules and templates should be the eXtensible Stylesheet Language (XSL).

5. DETAILED REQUIREMENTS

R01: The application should provide an interface for the collection of status information in a convenient way.
R02: Input forms for status information should be customized based on user preset information and the user ID.
R03: Status information should be stored in a central database.
R04: Updating and retrieving of information in the database should be made possible.
R05: The application should provide mechanisms to generate structured information from a database. XML documents should be generated for the status information retrieved from the database.
R06: Customization and transformation of structured documents should be possible.
R07: Users should be able to specify custom transformations and preferences through scripts.
R08: Users should be able to view the same information based on different criteria and formats through the RGI.
R09: Users should be able to attach related files while entering their status information. CSRI should have proper fields for allowing users to attach multiple files with their status information. The database should also have the proper fields to allow for various file types i.e. documents, presentations, multimedia etc.
R10: For remote access, requests should be sent to the server in an agreed upon XML format. Responses for any external remote request must be customized based on the requesting device’s capabilities. A typical scenario may include a manager requesting information about the status of an employee though an Internet capable device (handheld/mobile).
R11: Presentation style: Presentation style for the status reports and custom data retrieval can be specified by the user. These presentation styles are stored as stylesheets for the IPC. Users can also specify and create their own custom stylesheets. Stylesheets are written in the eXtensible Stylesheet language (XSL).
R12: Users can identify risks and request a risk analysis and assessment per status report or for multiple status reports. These reports generate a formatted risk log that includes the following fields per entry: Risk Id, Risk Name, Risk Description, Number of SR where the risk is identified, Risk Criticality, Recommend Action, and Risk Status.
R13: The users can generate several trends based on comparing the current data from status reports with historical data. Status reports for an employee, project, or team could be compared with their past status reports. Results from these comparisons can be customized into a user specified view to keep track of the progress of the employees and their efforts.

R15: Logs should be generated for the major tasks performed by ASRMS. These logs should be classified and recorded for status comparison, remote request processing, and risk analysis tasks. Users should be able to specify the output data file and/or data storage for storing and maintaining the logs. Users also should have the ability to turn the log generation feature on/off as desired.

R16: Users should be able to perform data mining on the existing status data. Such data mining operations are performed by an external data mining tool. ASRMS provides the interface for the data mining tools to connect to ASRMS and use the database for data mining operations. Data mining may result in the discovery of various patterns that might be useful for the company to predict and measure their strength and direction.

6. USE CASES

6.1 Login
Role in Context: To enable user login
Description: A user enters their username and password. Based on the roles defined for the user, CSRI customizes the status input form. Input form fields are based on the tasks and projects that the user is working on.

6.2 CSRI Status Entry
Role in Context: Users enter their status through CSRI
Description: An input form is displayed dynamically to the user based on their login ID. The user fills out the pertinent status information in the form fields, submits the data, and confirms the validity of the entered data. CSRI displays a confirmation screen after the data has been successfully entered into the database.

6.3 Storing the Form Data
Role in Context: Stores status data in a corresponding database
Description: After the user has filled in the status information in the CSRI, the input data is validated for syntax and data type. The data gathered from the input form(s) is stored into the corresponding database tables.

6.4 Generate XML Data from the Database
Role in Context: Generation of structured data by DSC
Description: DSC performs the role of structuring data. As per RGI internal request, corresponding data is retrieved from the database. DSC creates a corresponding standard XML document and fills it with the data retrieved. The XML document that is formed is validated for structure and contained data.

6.5 Transform XML Data
Role in Context: Transforming the XML data for customization
Description: Based on the output template for the request, IPC reads the corresponding XSL script and the XML document to be transformed. The corresponding input XML document is processed and transformed by the XSLT engine based on the rules specified in the style sheet.
6.6 Get Device Capability

Role in Context: Retriever of information about remote device capability and preferences

Description: The remote device sends the request in the form of an agreed upon standard XML format. The tag containing the device name and its id is read to obtain the id of the device that sent the request. The properties file is referenced for that device, where the information about the device’s capabilities, such as screen size, preferred data font and size, etc., is defined. Once the information is retrieved, the response is customized for that device as per Use Case 6.6.

6.7 Process Remote Requests

Role in Context: Process ASRMS remote server requests

Description: The request document is parsed and checked for validity. The device’s capabilities are determined from the already defined property file for compatible devices. Based on the defined device capabilities and preferences, a customized response is prepared. The response is send to the requesting device in an XML wrapper.

6.8 View Status Reports

Role in Context: Generate and view status reports

Description: The user selects the options for viewing the report from the main menu of RGI. RGI triggers DSC to retrieve the corresponding data and generates the standard XML document. Based on the customization required, IPC transforms the XML document to the desired output by applying the corresponding rules.

6.9 Analyze Graphical Reports

Role in Context: Ability to graphically analyze status information

Description: Status information can be viewed graphically in different chart formats and styles. Based on the selected data about the status of an individual, team, or project, various customized graphs are generated by the RGI. Users can change the style of the graph for the same data to view it from a different perspective.

6.10 Track Project Status

Role in Context: View project details for an existing project

Description: The project status can be tracked from RPI. The user specifies the project and timeline for which they wish to see the status of. All the employees associated with the project are identified and their status information is retrieved from the database. DSC forms the standard project status XML document and fills in the retrieved data. IPC applies the rules for viewing the data in the desired output format. Output from IPC is displayed to the user by RGI.

6.11 Backtrack Employee

Role in Context: Generate employee status details for a period of time

Description: A User specifies the employee for which the backtracking is desired for. Based on the time period specified for backtracking, all the status information for that employee pertaining to that time period is retrieved. DSC generates a standard backtracking XML document and sends it to IPC. IPC transforms the XML document based on the transformation rules specified in the script. The output is generated by IPC in the desired customized format. RGI displays the output to the user.
6.12 Compare Status Reports
*Role in Context:* Compare the current status report with past reports
*Description:* The User specifies the employee and corresponding time period for performing a status comparison. Based on the time period specified for comparison, status information is retrieved. Both the specified status reports are compared based on the selected criteria by the user. The comparison output is generated as per user specified preferences.

6.13 Identify Risks
*Role in Context:* To identify and forecast major risks for a project
*Description:* The User specifies the project for which a risk evaluation is desired. The status reports of all employees working on that project are retrieved. Based on the project start and finish date, the aggregate of individual effort and status is evaluated to check for any foreseeable risks. If the evaluation does not fit with the initial project schedule, a risk forecast is presented to the user. The forecasted risk is classified based on the severity of the delay or any unexpected work determined by the evaluation.

6.14 Risk Analysis Comparison
*Role in Context:* To compare the risk analysis performed with the preset company metrics
*Description:* The risk analysis results are stored as risk analysis logs. Based on the project specified for risk analysis, the logs are compiled together and the risk associated is compared with the metrics defined in the company for the project. Results based on the comparison are provided to the users in forms of suggestions that might exist to match the project status with the deadlines and other dates defined in the company metrics.

6.15 Manage attachments
*Role in Context:* Manage the attachments associated with employee statuses
*Description:* The User can classify, organize, and manage the attachments associated with the status reports. The classification can be based on the project, team or individual basis. Based on the classification performed, retrieval options should be automatically updated to include the specified classification.

6.16 Retrieve Custom Data
*Role in Context:* To retrieve and present user specified data
*Description:* Instead of generating entire status reports, users can submit queries for individual pieces of data from the existing status report data. Based on the data requested, DSC retrieves the corresponding data from the database. An XML document is formed for the request and is sent to the IPC to format and present as defined by the user.

6.17 Generate Logs
*Role in Context:* Generate classified logs
*Description:* Logs are generated and classified based on the user actions. Separate logs are generated for actions relating to status report comparison, risk evaluation, backtracking, and the processing of remote requests.

7. ARCHITECTURE
The high level architecture is represented in figure 2.
8. INTERFACES

ASRMS should be able to produce an output XML document for a particular status report generated. The generated XML document should have all the pertinent information for the corresponding status report. This XML document could be used for any other application that desires to utilize the data in its structured format. Users should have the ability to specify the desired output location and other preferences for the XML file.

An Interface should be provided for the risk analysis tool. Log files should be generated for risk analysis and status comparison features. These logs should be generated in both XML and text format. The format and structure of the logs should comply with the company’s existing metrics analysis tools.
An Interface should be provided for the major data mining tools to perform data mining operations on the status information database.

8. GLOSSARY

XML: eXtensible Markup Language
ASRMS: Automated Status Report Management System
CSRI: Composite Status Reporting Interface
XSL: eXtensible Stylesheet Language
DSC: Data Structuring Component
IPC: Intelligent Presentation Component
RGI: Report Generation Interface
MS: Microsoft
PDF: Portable Document Format
HTML: Hyper Text Mark Up Language

9. REFERENCES