University of Alberta

An Integrated Real-Time Control System for Structural Steel Fabrication Projects

By

Reza Azimi

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Construction Engineering and Management

Department of Civil and Environmental Engineering

Fall 2011
Edmonton, Alberta

Dedication

This thesis is dedicated with great love, appreciation, admiration, and respect to my kindhearted mother, my dear father, and my beloved sisters for their love, endless support and encouragement.

Abstract

Off-site structural steel fabrication has become increasingly popular in industrial construction projects over the past decade due to advantages of higher productivity, better quality, better safety and improved time and cost effectiveness. The fast paced nature of off-site structural steel fabrication environments, constraints induced by the construction site, and the existing limitations within the shop, make the control process of off-site construction quite challenging. Successful delivery of off-site structural steel fabrication projects requires a reliable monitoring and control system. A key factor for the success and effectiveness of such a system is its ability to perform real-time monitoring and to evaluate and analyze project performance fast enough so that in the case of deviation the appropriate corrective actions can be taken in a timely manner. New achievements in automated data capturing systems and computer simulation techniques provide a reliable platform for effective project monitoring and control, as they can respectively reduce the time required to obtain as-built data and simulate different scenarios for possible corrective actions ahead of time. The functionality of control systems can be greatly increased by integrating these two techniques; however, such a feasibility has not been examined in great detail. The primary objective of this research is to address this issue by proposing a framework for new generation of project monitoring and control systems focusing

on integrating automated data acquisition systems and computer simulation techniques that could be a robust decision support tool for project managers.

The second objective of this research is enhancing control process in the dynamic environments of off-site construction shops based on the control theory concept. The proposed method makes project managers capable of prompt non-linear data analysis and of predicting the behavior of the performance of the off-site construction fabrication shops. This leads to promptly determining the best corrective action in the case of variance to achieve the desired performance of the shop with minimum cost. To illustrate the feasibility and practicality of the proposed framework and enhanced project control method, they were successfully implemented in an actual off-site structural steel fabrication shop.

Acknowledgement

First and foremost I would like to thank God. You have held my hands all along the way and have given me the power to believe in myself and pursue my dreams.

I take immense pleasure to express my sincere and deep sense of gratitude to my supervisors, Dr. Simaan AbouRizk and Dr. SangHyun Lee for their sustained enthusiasm, creative suggestions, motivation and exemplary guidance throughout the course of my doctoral research.

I am very thankful to Dr. Tavakoli, from the Department of Electrical & Computer Engineering, for his guidance, constructive comments and patience he demonstrated.

I am also grateful to my doctoral committee: Dr. Nashwan Dawood, Dr. Yasser Mohamed, and Dr. Zhi-Jun (Tony) Qiu for their thoughtful review and providing me with invaluable advice and deep insight and invaluable suggestions to improve the quality of my work.

I would like to take this opportunity to thank all the individuals, colleagues and friends who assisted me during this research project. I extend my sincere thanks to Dr. Mohamed Al-Hussein, Stephen Hague, Brenda Penner, Steve Gamble, Andrea Hasenbank and other faculty and staff in the Construction Engineering and Management Program who assisted me in various aspects of this research. My special thanks go to Holly Parkis, for editing and proofreading my manuscripts meticulously.

Last, but not least, I would like to express my deepest gratitude to my beloved parents and my kind sisters Elaheh and Nafiseh who have always been a major source of motivation and support for me in every minute of my life.

Reza Azimi

List of Tables

List of Figures

Chapter 1. Introduction [1]

1.1. Background

The nature of construction projects is becoming more complex and more dynamic, creating challenges for project managers, who are responsible for overall project success. Monitoring and control are considered to be important management functions for the successful delivery of ongoing projects (Li et al. 2005). “Monitoring” refers to tracking deviations and variations in project performance indices during the project execution. “Control” refers to the corrective actions taken to diminish undesirable deviations. Thus, the objective of project monitoring and control is to measure the actual values of the basic variables of a project, such as the amount of work done, the quality of work, and cost. These data then need to be analyzed to determine if the project is meeting the targets of the work plan, and to make necessary adjustments to meet project objectives (Oberlender 2000).

The construction industry is an enormous industry in North America and a huge amount of money is spent in this industry (Azimi et al. 2010b). As an example, in 2010, US$ 9.62 trillion was spent in construction projects in the US (US Census Bureau, 2011). Due to this sizable investment, prudent monitoring and control is

---

[1] Parts of this chapter have been published at Azimi et al. 2010a & b and Azimi et al. 2009.

required to ensure best value for money. Thus, issues regarding current monitoring and control systems have developed into a major field of interest for researchers.

1.2. Problem statements

Project monitoring and control have been two areas of concentration for researchers due to the fact that one of the major causes for project failures is deficiencies in monitoring and control. Project monitoring is important because adequate, real-time and up-to-date actual data are necessary for timely and accurate tracking of the performance and status of the ongoing projects. This enables project managers and control systems to rely more on deductive and analytical methods rather than on their judgment (Nasr 2005) in making informed decisions shortly after any deviation occurs. Traditional monitoring methods are based on manual data collection, which may be slow and inaccurate. Researchers have been trying to make use of modern technologies to improve the quality and accuracy of the captured actual data, and consequently improving monitoring performance of construction projects during the course of their execution. For example, Geographic Information System (GIS) was utilized for monitoring precast building construction (Cheng and Chen 2002).

Studies were also carried out to improve different aspects of the project control process, such as integrating cost and schedule data for making better decisions. A number of tools that utilize standard project control techniques are also available to assist project managers in achieving their goals as effectively as possible. These efforts in enhancing monitoring systems, improving project control process and developing decision making tools have been shown to be of great worth in helping project managers to deal with some of their major responsibilities. However, a detailed literature review presented in chapter 2 identified several areas that still need to be improved (Azimi et al. 2011a). Integration between monitoring systems and control systems in a unified environment is therefore a significant area for researchers in the construction industry. For example, as an RD&D consortium which includes owners, contractors, suppliers and research organizations, FIATECH tries to accelerate the deployment of integration and automation technology for capital projects. In its Capital Project Technology Roadmap (http://fiatech.org), FIATECH addresses nine critical elements for achieving a fully automated project and facility management environment integrated across all phases of a facility's life cycle. FIATECH states that there is a missing link regarding the ability to integrate all the necessary components of a project management system in a unified environment (http://fiatech.org).

Current monitoring systems typically communicate ineffectively with project control systems which deal with forecasting performance, analyzing variances, and recommending corrective actions. In other words, because monitoring systems are not integrated with control systems in a unified environment, they act as stand-alone islands regarding communication and data sharing and such a gap can result in late decision making and, consequently, cost overruns and delays (Nassar et al. 2003, Moselhi et al. 2004, Li et al. 2005, Nasr 2005).

Monitoring and control systems also suffer from lack of flexibility, interoperability and extendibility (see chapter 2) as well as slowness regarding decision making (see chapter 4).

This research addresses these challenges by proposing a new project control framework in which an automated real-time monitoring system is integrated with an enhanced control system for facilitating the decision making process in terms of taking effective and timely corrective actions.

1.3. Research objectives

The primary objectives of this research are as follows:

1) Developing an integrated real-time monitoring and control framework that facilitates decision making by enabling project managers to take corrective actions right after any deviation takes place and mitigate potential damage to ongoing projects.
2) Enhancing the project control process by introducing a practical and effective method for prompt data analysis and performance forecasting which leads to speeding up the decision making process.

1.4. Scope of research

This research project is focused on improving monitoring and control processes for off-site construction projects. System implementations and case studies were carried out in a complicated off-site construction environment namely a steel fabrication shop. In terms of monitoring, this research utilized Radio Frequency Identification (RFID) technology for automated data capture regarding the locations of the steel pieces being processed in the fabrication shop and proposed methods to translate this information into meaningful managerial data such as project progress, schedule performance index and cost performance index. In this way, high quality real-time actual data can be captured and become ready to be used for project control purposes. To be effective, the control system should be able to make use of these real-time actual data for timely data analysis and quick decision making. Thus, in other parts of this research methods for speeding up the decision making in the control process were proposed. The proposed control system is capable of being extended and used in different off-site construction projects.

1.5. Methodology

The strategy used in this research is based on developing a decision support tool for managers regarding improved performance monitoring and control. The plan of this research can be divided into four phases:

Phase 1: Literature Review

The first phase consists of two steps:

- Investigating current usage, trends and limitations of existing data capture techniques, and monitoring and control systems in the construction industry.
- Exploring available technologies for real-time monitoring and their feasibility for use in the research area.

Phase 2: Data Capturing- Monitoring

For each project, two methods were used to collect the required data:

A) As-planned automated data capture: Required as-planned data were determined and extracted from the database or derived from drawings and 3-D models of the ongoing projects. This enabled the automatic generation of a baseline (i.e., the accepted and approved plan) of each project with minimum human involvement, which is slow and error-prone.

B) Automated real-time actual data capture (monitoring): The data required to be captured included labour (man hours), location, and drawing number pertaining to each processed steel piece. RFID technology was implemented in the steel fabrication shop as a prototype for movement tracking and data capturing. These location tracking data were utilized for determining managerial information such as project performance, project progress and so on.

Phase 3: Framework Development

In this phase, based on the knowledge obtained in Phase 1, an integrated control framework and its components was conceptualized and developed.

Efforts have been focused on developing and integrating components required to monitor and control structural steel fabrication projects.

Phase 4: Validation

Case studies in a collaborating steel fabrication company were conducted to test and validate the following:

- Conceptual model validation, to determine whether the algorithms, theories and assumptions were rational and correct.
- Functionality and operational validation, to ensure that output results depicted the actual situation of ongoing projects and fit with the objectives of the research.

Depending on the type of validation, interviews with experts and time studies were carried out in conjunction with any additional activities deemed necessary.

1.6. Implementation Environment

To bring the conceptual model of the proposed integrated monitoring and control framework into reality, development was carried out in programming and simulation environments. A brief explanation of the environments, programming language, and database medium used for developing and implementing the proposed framework is presented below:

- Microsoft Visual Studio.NET was the general programming environment for developing different graphical user interfaces and windows forms. One of the capabilities of Visual Studio is incorporating dynamic-link library (DLL) files, which can significantly facilitate data sharing between interfaces which have been developed and external software.
- The Simphony.NET 3.5 simulation modeling engine (AbouRizk and Hague 2009) developed by construction engineering and management researchers at the University of Alberta was utilized for developing the discrete event simulation (DES) component of the proposed framework. The Dynamic-link Library (DLL) files provided by Simphony.NET 3.5 are used for the implementation of the DES component of the framework. Users of Simphony.NET 3.5 can not only utilize predefined modeling elements for modeling real-world systems but also have the option of adding code within certain modeling elements in the VB.NET programming language to meet their needs.
- AnyLogic® 6.4.1 (http://www.xjtek.com/) was another simulation modeling environment used in the last case study of this research due to its unique capability for modeling dynamic changes of resources for different activities and developing appropriate libraries and solutions for specific requirements.

The simulation models developed in AnyLogic® can be exported as Java applications and can be deployed on other machines. These Java applications can be linked to databases, external files and other applications.

The proposed research utilized the Construction Synthetic Environment (COSYE) (AbouRizk and Hague 2009), a distributed simulation environment developed at the University of Alberta. COSYE is an environment optimized for development of huge simulation models in different construction domains and provides a suite of generic modeling elements. In COSYE, the large simulation model is split into several sub-models that are developed separately and can be run on different machines. During run-time, COSYE provides the necessary communication, information exchange, and data sharing protocols and assures synchronization, coordination and consistency among different submodels.

1.7. Dissertation Outline

This thesis is prepared in a paper based format, consisting of five chapters and four appendices. During the course of this research, the author produced and published several manuscripts in the form of journal papers and conference articles to address the identified challenges. Except for the first and the last chapters, i.e., the introduction and conclusion chapters, each chapter is an independent paper composed of material from a corresponding publication that describes the details of major challenges involved, and algorithms and methods studied, adopted, developed, and implemented to achieve a particular research objective. The chapters are logically coherent and pertinent to the theme of the thesis. The thesis begins with an introductory chapter which is an overview of the entire research, including background, problem statement, research objectives, methodologies and research contributions. Chapter 2 describes the proposed framework for automated and integrated project monitoring and control and its application in off-site steel fabrication projects. Chapter 3 addresses an implementation of the proposed framework equipped with Artificial Intelligence for improving look-ahead scheduling in off-site steel fabrication projects. Chapter 4 describes a new vision in off-site construction project control and proposes a method based on "Control Theory" for enhancing the project control process in terms of timely data analysis and making timely decisions. Chapter 5 summarizes what has been achieved in this research, and outlines a proposal for future research. Appendix A is a paper published in the proceedings of the ASCE Construction Research Congress (CRC) regarding resource performance indicators and how they can be used for improving the performance of the fabrication shops. Appendix B deals with the details of the data acquisition system introduced in Chapter 2. Appendix C discusses the details of the monitoring and control federation which was discussed in Chapter 2. Appendix D contains further details of the intelligent scheduling system discussed in Chapter 3.

Conclusion

This thesis research addressed some fundamental shortfalls of the current monitoring and control systems by proposing an automated and integrated project monitoring and control framework as well as a simple but effective method based on control theory for enhancing the project control process in off-site construction projects. These led to achieving the ultimate goal of the research: providing project managers with efficient decision making tools resulting in taking prompt corrective actions and mitigating potential damage to ongoing off-site construction projects. This research opens the door for future endeavors regarding improving different aspects of the project control process, such as reasoning and diagnosing project performance deviations. The proposed framework and the method introduced for enhancing project control presented herein have been successfully applied to a complicated off-site construction environment, a steel fabrication shop.

References

AbouRizk, S.M. and Hague, S. (2009). “An Overview of the COSYE Environment for Construction Simulation.” Proceedings of the winter simulation conference; Austin, Texas, USA, 2624-2634.

Azimi, R., Lee, S., AbouRizk, S.M., and Alvanchi, A. (2011 a). “A Framework for an Automated and Integrated Project Monitoring and Control System for Steel Fabrication Projects.” Journal of Automation in Construction, 20(1), 88-97.

Azimi, R., Lee, S., and AbouRizk, S.M. (2010 a) “Integrated simulation-based Look-Ahead Scheduling for steel fabrication projects.” Proceedings of the Second International Conference on Advances in System Simulation (SIMUL 2010), Nice, France, 129-133.

Azimi, R., Lee, S., and AbouRizk, S. M. (2010 b) “Resource performance indicators in controlling industrial steel projects”, ASCE Construction Research Congress, Banff, Canada, 21-30.

Azimi, R., Lee, S., and AbouRizk, S.M. (2009) “Automated project control system for steel projects.” Proceedings of the Joint International Conference on Construction Engineering and Management and Construction Project Management (ICCEM-ICCPM), Jeju, Korea, 479-486.

Cheng, M. and Chen, J. (2002). “Integrating barcode and GIS for monitoring construction progress.” Journal of Automation in Construction, 11(1), 23-33.

Li, J., Moselhi, O., and Alkass, S. (2005). “A reasoning process in support of integrated project control.” Canadian Journal of Civil Engineering, 32(3), 500-16.

Moselhi, O., Li, J., and Alkass, S. (2004). “Web-based integrated project control system.” Journal of Construction Management and Economics, 22(1), 35-46.

Nasr, E.B. (2005). “An integrated project planning and control system approach for measuring project performance.” Ph.D. thesis, University of Colorado at Boulder, USA.

Nassar N., Halepota T. and AbouRizk S. M. (2003). “Integrated project controls system - IPCS.” Proceedings of the Annual Conference of the Canadian Society for Civil Engineering, 51-60.

Oberlender, G.D. (2000). “Project management for engineering and construction.” The McGraw-Hill Companies, Inc., Boston, MA.

US Census Bureau. (2011). <https://www.census.gov/const/C30/totsa2010.xls>, Accessed on June 1, 2011

Chapter 2. A Framework for an Automated and Integrated Project Monitoring and Control System for Steel Fabrication Projects [2]

2.1. Introduction

For successful project delivery, a reliable monitoring and control system is essential. Actual data is collected to measure the actual values of a project's basic variables, such as the percent complete, the quality of work, and cost. Comparing this data with planned values indicates if the project is meeting the targets of the work plan. The necessary adjustments then can be made to meet the project objectives (Oberlender 2000).

In industrialized countries, the construction industry is an important industry that contributes approximately 10% to the gross national product (GNP) (Allmon et al. 2000). Accordingly, national economies can be significantly affected by the performance of the construction industry, which highlights the importance of tracking and control in construction projects. In construction, while the average duration of activities typically falls within a range of days, the usual frequency of reporting is weekly or monthly. Therefore, in the case of deviations, traditional

---

[2] A version of this chapter has been published in the journal of Automation in Construction, 20(1), pp. 88-97.

project control systems used in the construction industry may yield corrective actions too late. This can result in cost and schedule overruns. As well, the longer it takes to recognize deviations, the greater the potential damage (Azimi et al. 2009).

Steel construction plays a significant role in the construction industry as many projects incorporate steel structures. Steel construction has a considerable impact on the total cost of such projects and it is commonly located on the critical path of the schedule. It involves designing, fabrication, shipment, and site erection of thousands of steel elements with different dimensions and complexities. To deal with such a broad area, an integrated monitoring and control system is a great tool for project managers to ensure the steel elements are fabricated and erected according to the project plan. Such a system enables real-time high-quality data to be captured, project performance indicators to be evaluated and forecasted, and timely corrective measures to be executed as needed. This helps minimize or prevent damage caused by project discrepancies.

Recent advances in automated data capturing systems and computer simulation techniques provide a reliable platform for effective project monitoring and control, as they can respectively reduce the time required to obtain as-built data and simulate different scenarios for potential corrective actions ahead of time (Azimi et al. 2009). Integrating these two techniques can significantly increase the functionality of control systems. However, the feasibility of such a concept has not yet been examined in great detail. This chapter addresses this issue by proposing a framework for a new generation of project monitoring and control systems. This framework unites automated data acquisition systems and computer simulation techniques (for control purposes) in order to create a robust decision support tool. Though the ultimate target is to develop a framework for steel construction projects, this chapter specifically deals with the fabrication phase of steel projects.

2.2. Background and state of the art

Monitoring and control are two important issues in construction management. Many efforts have been made by researchers to improve these areas. The output of an automated monitoring system is high quality real-time data that can be used by control systems and project managers to analyze variances and to make effective decisions. On the other hand, traditional monitoring methods are based on manual data collection, which is slow, imprecise, and error-prone. To address this issue, modern Automatic Data Capturing (ADC) technologies have been exploited to monitor performance of construction projects throughout their execution duration (Echeverry and Beltran 1997; Ciesielski 2000; Deng et al 2001; Cheng and Chen 2002; Navon and Goldschmidt 2003; Cheok and Stone 2004; Sacks et al. 2003; Navon 2005; Sacks et al. 2005; Ghanem and Abdelrazig 2006; Navon and Sacks 2007; Chin et al. 2008). For example, Radio Frequency Identification (RFID) combined with four-dimensional computer-aided design

(4D CAD) has been used to monitor steel works in high rise buildings (Chin et al. 2008).

In terms of integrated control systems, most of the developed systems focus on data models that integrate cost with schedule data (Abudayyeh and Rasdorf 1991; Singh 1991; Diekmann and Al-Tabtabai 1992; Alshaibani 1999; Fayek 2001; Nassar 2003; Li 2004; Moselhi et al. 2004; Nasr 2005; Kim et al. 2008; Po-Han 2008). However, current systems are plagued by one or more of the following shortfalls (Azimi et al. 2009):

- Manual data collection/entry.
- The lack of some of the important functions of an integrated project control system, such as forecasting performance, analyzing variances, and recommending corrective actions.
- System inflexibility and interoperability issues.
- Difficulties regarding data sharing.

A review of the literature reveals that an integrated and real-time project control system could have a significantly positive impact on the construction industry (Ghanem and Abdelrazig 2006; Navon and Goldschmidt 2003; Navon 2005; Navon and Sacks 2007). However, systems that simultaneously provide data acquisition, performance evaluation, performance forecasting, performance analysis, and corrective action recommendations in a unified environment are still under development. This research addresses the aforementioned shortfalls by proposing an effective project control framework in which real-time as-built information taken from a monitoring system is integrated with a control system made up of different components and simulation models. The proposed framework will enable project progress and performance to be monitored in a timely manner; managers will automatically be notified of any deviations and provided with complete information on the impacts of problems or changes. The framework will also help managers to assess risks, forecast the performance of a project in any future interim periods, and take necessary corrective actions. Thus, productivity will be improved in real time and the system will create a database of plans and outcomes based on historical experience. This information can subsequently be used to optimize all types of processes for future projects.

The framework is enhanced with High Level Architecture (HLA) for providing a reliable infrastructure for the efficient integration of all the components of a project control system. Interoperability, reusability, flexibility, and system speedup are some advantages of using HLA as the infrastructure for the proposed framework. This research not only deals with integration, but also attempts to take advantage of modern technologies for real-time data logging in order to construct an integrated real-time control system.

2.3. Tools and concepts utilized for framework development

To provide the reader with a better outlook on the research topic, a brief review of the concepts, tools, and infrastructure used in developing the framework is presented in this section.

2.3.1. Automatic Data Capturing (ADC)

ADC pertains to automatically gathering data and entering that data directly into a database with no human involvement. Decreasing costs of advanced technologies in terms of ADC, along with their improved accuracy, motivate the researchers to make use of them for real-time data acquisition. Video cameras for the monitoring and control of construction sites (Deng et al. 2001), the Global Positioning System (GPS) for the automated monitoring of man-hours spent carrying out a task (Sacks and Navon 2003), and LADAR (laser detection and ranging) to monitor construction progress are some examples of modern technologies utilized in ADC. Navon and Sacks (2007) discuss the advanced technologies available for collecting actual data and propose a map that indicates which technologies may be appropriate for each area of need in different fields of the construction industry. The suggested technologies are classified in nine major categories: GPS, Radio Frequency Identification (RFID), barcode, video, audio, load gauges, accelerometer, equipment location, and LADAR. Taking into consideration their proposed map, the RFID technology has been utilized in this research due to its capabilities for automatic actual data capture regarding the location of steel elements in a fabrication shop. This chapter discusses how location information can be employed for monitoring and controlling the cost (e.g., spent man-hrs), schedule (e.g., start time and finish time), and progress of the steel fabrication projects. Compared to its predecessor (i.e., barcode technology), RFID technology has the advantages of the durability as well as reusability of RFID tags, no need for line-of-sight between the reader and the tag, and greater reading distance (Chin et al. 2008).

2.3.2. Simulation

Simulation techniques are useful for representing real-world systems by creating computer-based and mathematical-logical models for experimentation. External factors that influence the experimentation results, as well as the evaluation criteria and thresholds used to assess the value and effectiveness of a decision/scenario should be included in the models (Pritsker 1986; Pritsker et al. 1989; Law 2006). The capabilities and robustness of simulation in terms of modeling complex systems have made it a common decision support tool for project managers in the construction industry.

2.3.3. High Level Architecture (HLA)

The High Level Architecture (HLA) standard, developed by the U.S. Department of Defense, is defined as an approach to integrate separate, autonomous simulators into a single distributed simulation system. The autonomous simulators are called "federates," while the resulting distributed simulation execution is known as a "federation" (Santoro and Fujimoto 2008). The aim of the HLA is to provide a structure that supports the reuse of different simulations, ultimately reducing the cost and time required to create a synthetic environment for new purposes (Fujimoto 2003).

Using this technology, the system is broken down into component models that are able to run independently (Gan et al. 2000). The HLA structure significantly improves the development of construction simulations because there are many common components and similarities between the simulation models of different construction applications. For example, a system dynamics (SD) federate that takes into account the influence of company policies (regarding skilled workers, overtime hours, and so on) on productivity can be developed. Such a federate may be used in different applications, such as simulating tunneling, earth moving, etc. This structure also helps when developing each federate separately with a different simulator, although the federates can ultimately be combined and work together (Kuhl et al. 2000). Furthermore, the exchange of information remotely over the HLA can bring together complementary data and information from different locations and make this data available to all of the federates in the HLA federation.

2.3.4. Construction Synthetic Environment (COSYE)

To achieve an integrated model of all processes, provide means for automation, and develop intelligence through their design, the construction engineering and management group at the University of Alberta developed an HLA-based "Construction Synthetic Environment" called COSYE (AbouRizk and Hague 2009). This platform is a software environment enhanced with the required simulation core services and specific modeling tools for creating distributed simulation models for complex applications in the construction domain.

2.4. Steel Construction

For decades, steel has commonly been an indispensable component of simple and complex structures, such as towers, buildings, refineries, and bridges. Owners, designers, and contractors tend to favor structural steel construction because it is not only economical and efficient, but also orderly. A typical steel construction project generally has five major stages: (1) detailing (designing); (2) procurement; (3) steel fabrication; (4) shipment; and (5) site erection (Figure 2-1). Some projects require modules to be assembled at a location other than the erection site. In these cases, module assembly can be considered a part of the process of steel construction.

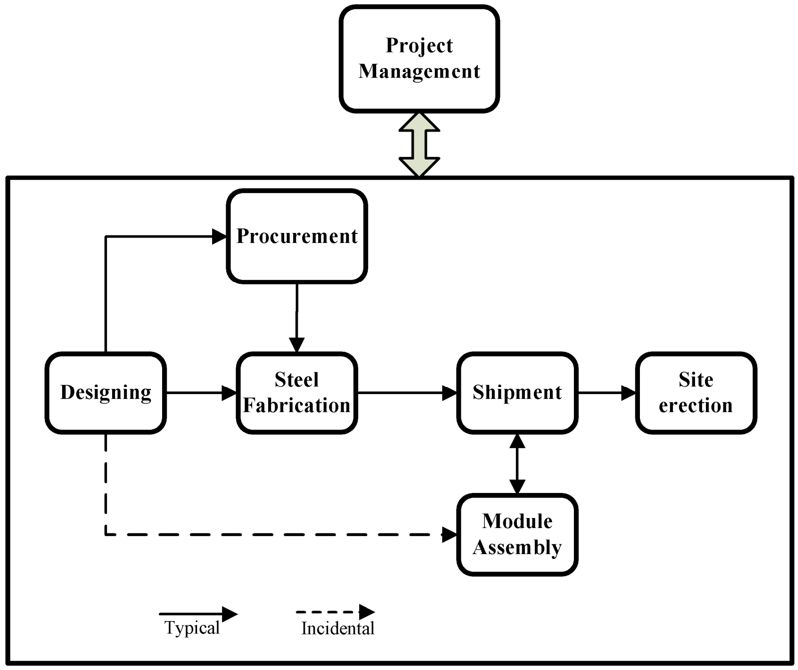


Figure 2-1. Process model of steel projects.

The diversity and complexity of steel pieces produced, their dimensions and processing requirements, and the variety of equipment and labor disciplines required from detailing to site erection, indicate the large number of potential resources, activities, and interactions required. This highlights the importance of monitoring and control in steel construction projects.

As steel fabrication is a critical stage in steel construction projects, this research predominantly focuses on the controlling of steel fabrication.

Steel fabrication refers to the production of steel or metal items, such as columns, beams, plates, tanks, and pressure vessels, in a controlled shop environment. The fabrication shop has several shops: the cutting (detailing) shop, fitting/welding shops, and the painting and sand blasting shop. In the cutting shop, there are different cutting machines used to cut different types of materials (e.g., beams, plates, and angles). The most time-consuming work is performed at the fitting/welding shops; therefore, there are usually several fitting/welding shops to distribute the work load. However, each shop may be specially equipped for specific jobs (e.g., heavy or light products). All products are inspected after proceeding through the fitting and welding shops. The steel elements that require surface protection or painting then are processed in the painting shop.

The main inputs of the steel fabrication shop are raw steel material and product fabrication drawings, while the fabricated steel element is the main output of the system. In terms of control, this research focuses on the most detailed level, the piece level (Figure 2-2).

Utilizing this "bottom-up" approach, it is easier to achieve useful information in the other levels.

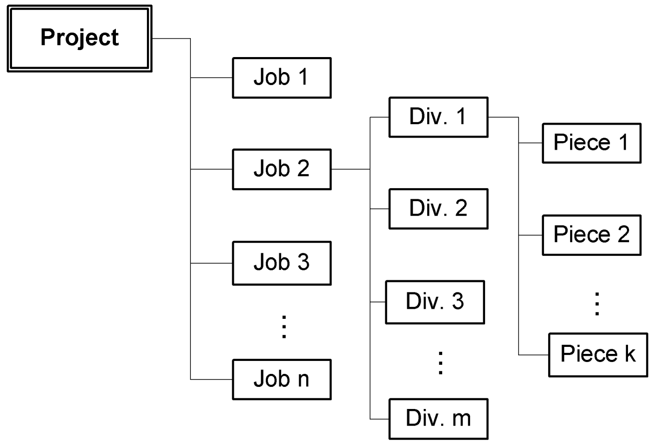


Figure 2-2. Work Breakdown Structure of a steel fabrication project.

2.5. Proposed framework for an integrated monitoring and control system

Good decision making is crucial in construction projects and can save a considerable amount of money. Therefore, a reliable decision support tool is essential for project managers to make effective and on-time decisions. This research proposes an integrated and automated monitoring and control system that will assist project managers of steel projects in executing appropriate corrective actions in a timely manner. The objectives, current needs, and characteristics of the research domain make up the proposed conceptual framework depicted in Figure 2-3.

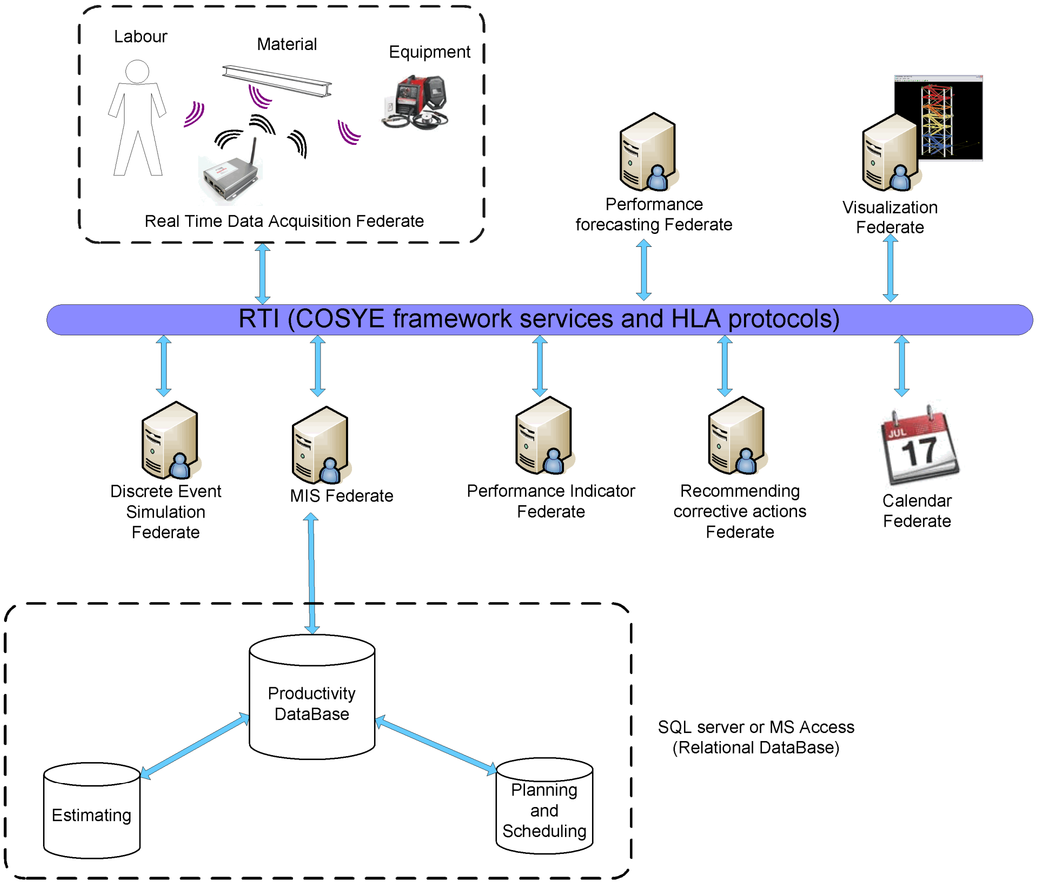


Figure 2-3. Proposed integrated Real-Time control system.

The framework covers all the major elements of a monitoring and control system. These elements include project baseline, automated actual data acquisition, performance evaluation (comparing actual data and baseline data), performance forecasting, visual reports on current and forecasted project status, and corrective actions. The framework is enhanced with a simulation component which utilizes as-planned and actual data to detect probable future deviations in project performance. This component also enables managers to experience different scenarios in virtual reality and identify the best solution for occurring or anticipated issues.

The HLA's capabilities make it an authentic foundation for integrating the specified components of this project control system. The proposed framework incorporates the HLA concepts, including several distinct components called "federates", which collaborate, cooperate, and interact with each other in real-time through the Run-Time Infrastructure (RTI) (Azimi et al. 2009). Each federate can be developed separately and run at geographically dispersed locations. Synchronization, communication, and data exchange between federates are major services provided by the RTI. The object model template (OMT) defines the structure of the shared data within the framework and allows for common federation object documentation and interoperability. These services, which are defined by the HLA standard, ensure communication and integration between the framework components. Real-time communication between the monitoring and control federates supports detection of deviations immediately after they occur, therefore assisting in the on-time delivery of appropriate corrective actions.

Furthermore, in the framework, the Management Information System (MIS) handles the baseline and as-built information of all the scheduled, ongoing, and completed projects. The data acquisition federate provides as-built data which is compared with the baseline by the Performance Indicator federate to evaluate project performance and related resources and detect current deviations. The forecasting federate determines probable deviations in projects or resources based on historical trends, current status, and expert judgment. The Recommended Corrective Actions federate recommends corrective actions based on current or probable deviations. The Discrete Event Simulation simulates different scenarios (decisions/actions) to observe their effects on the current and future performance of projects. A detailed discussion of the framework development and the developed federates can be found in the next section.

2.6. Framework implementation

In the previous section, the conceptual framework and its components were briefly described. To implement the proposed framework, the development of its federates in COSYE has been started. As the first step, the object model, including all the essential objects and their attributes, was developed and the interactions between the objects in different federates were well-defined. Then, the Object Model Template (OMT) of the framework was created. The OMT encompasses the objects, their attributes, parameters, types, and sharing methods between the federation's federates. For the sake of reusability and maintaining reliable connections between federates and other CAD software, a product model, resource, and environmental models also were developed. Visual Studio 2008 was used for programming and developing the federates and interfaces in COSYE. The developed federates include: real-time data acquisition, Management

Information System (MIS), Discrete Event Simulation (DES), Performance Indicator (PI), calendar, and visualization (Figure 2-4).

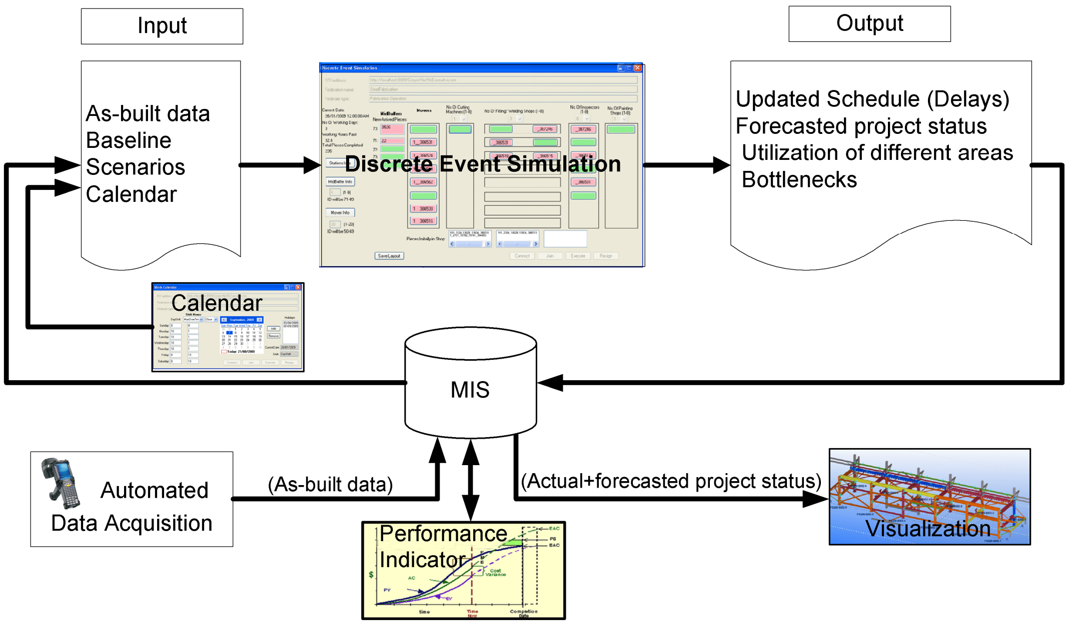


Figure 2-4. Major elements of the developed integrated control system.

- *Real-Time Data Acquisition Federate*

This federate is responsible for capturing actual data regarding major resources (i.e., labour, material, and equipment) in a timely manner. It acts as a bridge between the data capturing devices and the MIS federate. In the present research, Radio Frequency Identification (RFID) technology—which includes putting ID tags on steel pieces and reading tags at certain locations with portable and stationary readers—was deployed to capture as-built data.

Figure 2-5 depicts a typical steel fabrication shop and the locations where the tags are read by tag readers. Tag readers are located at the entry and exit of each fitting, welding, and painting area. Piece tracking did not occur in cutting areas due to the fairly high associated tracking costs in this part of the fabrication process. A piece may require several cut and/or preset components to get ready for being fitted. Tracking these components significantly increases the handling costs that come from pasting tags onto each component, updating the database, and then removing the tags in the next step when the components are put together to form a piece (i.e., fitting). Thus, for the implementation phase, the option of tracking pieces in the cutting area was turned down. Another tag reader is located at the shipment area to detect shipped steel pieces.

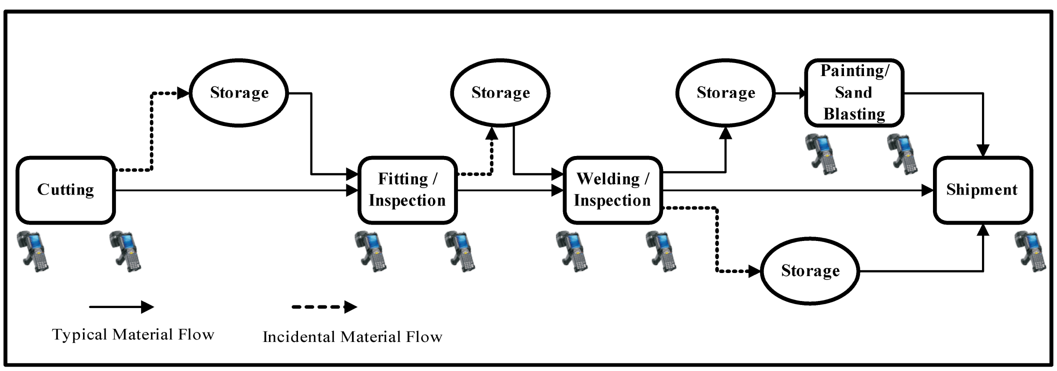


Figure 2-5. A typical steel fabrication shop and assigned tag readers' locations.

Each tag has a unique ID which is assigned to a steel piece. The first time a tag is read indicates that the related piece is located at the fitting area and the fitting process has started. The second read indicates the end of the fitting process. By subtracting the time of the first read from the second read, the man-hours and equipment hours spent on fitting that piece can be calculated. The same approach is taken to determine piece location and the resources (labour and equipment) used to weld, paint, ship, and install each piece (e.g., the third read reflects the start of the welding process and so on). If tag reading indicates that a process is done but the successor process has not yet started, the piece is interpreted to be stored in a related storage area. Thus, by utilizing RFID technology, the location of each piece and the man-hours spent to process different pieces at different stages are captured and transferred automatically to the database. In this way, by comparing baseline and actual data, essential controlling variables, such as project performance, productivity, and percent complete can be derived. Regarding the data exchange, TCP/IP is the protocol used for inter-process communication between the tag readers and the federate.

In terms of the as-planned data, once the design of a steel project is finished, a specific file format from the 3D model (created in the designing software), which contains detailed information on different pieces, is generated and updates the database.

- *Management Information System (MIS)*

This federate handles the as-planned (baseline) and as-built information of all the scheduled, ongoing, and completed projects. It is a shared database among different federates that feeds these federates with the required data, and it is automatically updated over time. It provides data such as the list of current projects, delivery times, estimated work volume, preliminary schedules, and actual project performance information. Actual performance indices, forecasted performance indices, and delays are some examples of the updates it receives from other federates.

- *Performance Indicator (PI) Federate*

As-built data is used to control project status. One of the common techniques to control project status is the Earned Value Method, which takes schedule and cost performance indices into account (Li 2004). This method discloses deviations in project performance. The Cost Performance Index (CPI) determines if a budget is over or under run, while the Schedule Performance Index (SPI) determines if a project is ahead or behind schedule. To identify what causes variance in each performance index, the concept of the control object (Li et al. 2005) was used and developed in this federate.

As shown in Figure 2-6, a steel project is represented by a set of control objects which are steel pieces. Each piece has its own resources (i.e., labour, material, equipment and/or subcontractor) as well as its as-planned and actual data. Status class refers to the actual start and finish time of a piece. Piece location and progress also are considered as classes in the model.

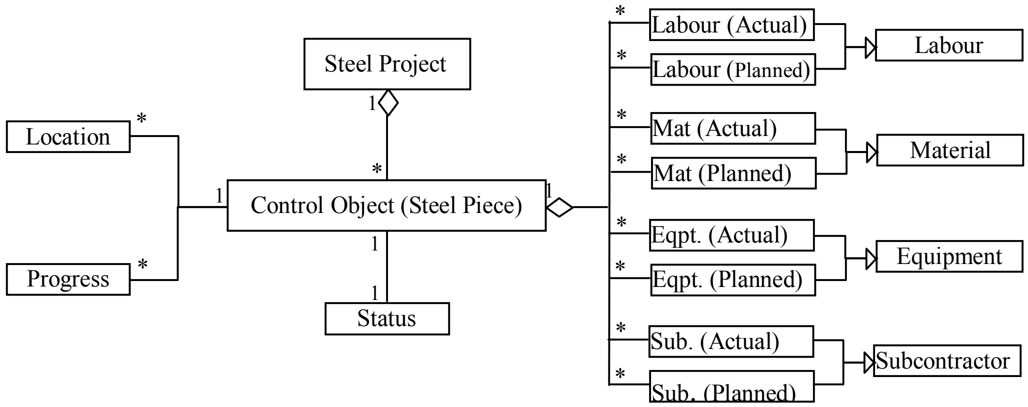


Figure 2-6. Object model of a steel project.

To evaluate cost and schedule variances, the earned-value method is used at the piece level, as discussed by Moselhi et al. (2004) and Li (2004), as well as at the resource level. To determine the performance indices at the piece level, actual cost (man-hours spent in processing the piece) and actual progress (percent complete) should be calculated. Actual cost (man-hours) can be derived from the actual data acquisition system by adding the man-hours spent on processing a specific piece at each area. The progress of each piece is then determined by translating piece location to percent complete using the rule of credit extracted from experts.

For instance, piece fitting is equivalent to 45% progress in the fabrication phase, according to the experts. This results in determining performance information not only at the project level, but also at the division and job levels. In the case of performance variance at the piece level, the cost and schedule performance indices (CPI and SPI) at the resource level are observed to determine the resource(s) that caused the deviation. For example, for a piece with a variance in the CPI, if all the resources have reasonable cost performance indices (CPI≈1) except for labour, labour is considered to be the cause of the steel piece performance variance. Li et al. (2005) propose thirteen resource performance indicators, which include crew attendance, labour unit productivity, material usage, etc. These are calculated by the PI federate and can be used by the Recommended Corrective Actions (RCA) federate to detect the reason(s) behind the performance variance of each resource.

- *Calendar Federate*

Day shifts, night shifts, overtime hours, and national holidays that affect the results of the DES federate are defined in this federate. It is a generic federate that is applicable to almost all projects. It can be used by federations developed for other fields in construction (reusability of the federate).

- *Discrete Event Simulation (DES) Federate*

The DES federate simulates the fabrication phase of steel projects based on the forecasted resource performance indicators and the available baseline information and captured actual data related to the scheduled and/or ongoing projects. The Dynamically Loadable Library (DLL) files provided by Simphony.Net 3.5 (http://irc.construction.ualberta.ca/Simphony35/), developed by construction engineering and management researchers at the University of Alberta, are used for the implementation of this federate. In the developed DES model, an entity resembles a steel piece at the different levels of the fabrication process, starting from a work order sent to the fabrication shop and ending with a completely fabricated piece. Stations (e.g., cutting, fitting, and welding), movers (e.g., cranes, rail cars, and forklifts), and mid-storage are the available resources within the fabrication shop. All of these elements were developed to be as flexible as possible in order to represent structural steel fabrication shops with various layouts and different types of equipment.

New entities are created and sent to the fabrication shop based on the planned and actual data which have been recorded in the database. Based on the defined sequence of activities, the structure of the piece, and the availability of the resources, pieces are sent from one resource to another, while the required data are collected from each step. For example, consider a piece that requires a welding activity as its next step of fabrication. If there is at least one idle welding station that can work on the piece, the piece will be moved from its current location to the idle welding station; otherwise, the piece is moved to storage. The volume of the work at each step is determined for each piece based on the read data from the database. However, there is also a degree of randomness involved in each activity; this can be defined by the federate user through different provided types of statistical distributions (e.g., triangular or uniform distributions).

As a result, the DES federate is able to provide fabrication shop managers with a variety of information, such as bottlenecks and utilization of active stations in the fabrication shop, and the present and future status of projects (earned value information). In this chapter, bottlenecks refer to the areas where the number of pieces waiting to be processed exceeds the threshold value. Storage capacity usually determines this threshold value. This federate enables managers to experience and compare different scenarios in virtual reality and find the best solution for current or anticipated issues in a fabrication shop (such as a low utilization rate, long queues of steel pieces behind a station, etc.).

- *Visualization Federate*

This federate provides end users with intuitive reports about multivariate information (e.g., schedule and cost) in a holistic manner. A colour coding concept (Lee and Peña-Mora 2006) is used to represent various performance information (Figure 2-7). Based on the performance indices (i.e., SPI and CPI), generated from the PI federates for each steel piece, a colour is assigned to a corresponding component in the 3D project model. In other words, the 5D model, which concurrently represents schedule and cost performance information, is generated for users. This allows them to visually comprehend the actual and forecasted status of each project. For example, as seen in Figure 2-7, red indicates that the control object suffers from budget overrun (CPI<1) and is behind schedule (SPI<1).

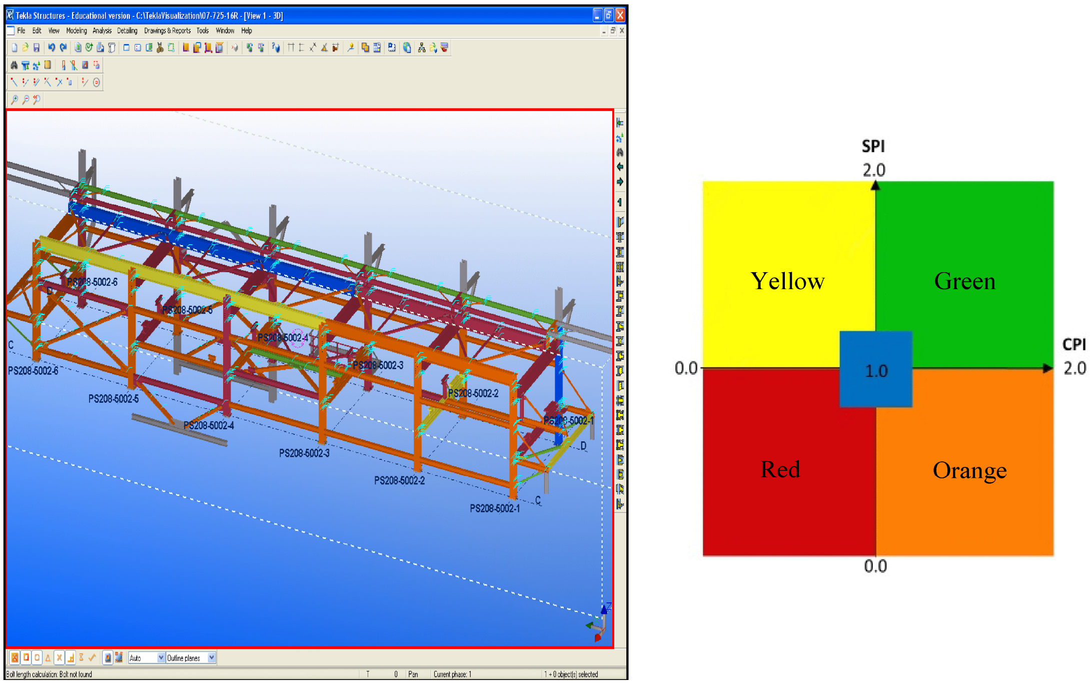


Figure 2-7. Colour coding for performance analysis.

Sample federates developed for the proposed monitoring and control system are shown in Figure 2-8. A snapshot of the developed calendar federate, in which working hours (dashed rectangle A), overtime hours (dashed rectangle B), and holidays (dashed rectangle C) can be defined, is demonstrated in Figure 2-8a. Figure 2-8b and Figure 2-8c depict the components of the implemented visualization federate. In this federate, detailed information about overall CPI, SPI and the progress of each division (dashed rectangle D), and its steel pieces (dashed rectangle E) in ongoing projects is provided for the end users. Based on the selected activity (dashed rectangle G), specific performance indices can be provided as well (dashed rectangle F). The users also have the option to intuitively monitor and control the projects within the 5D models (Figure 2-8c) generated based on the color coding concept shown in Figure 2-7.

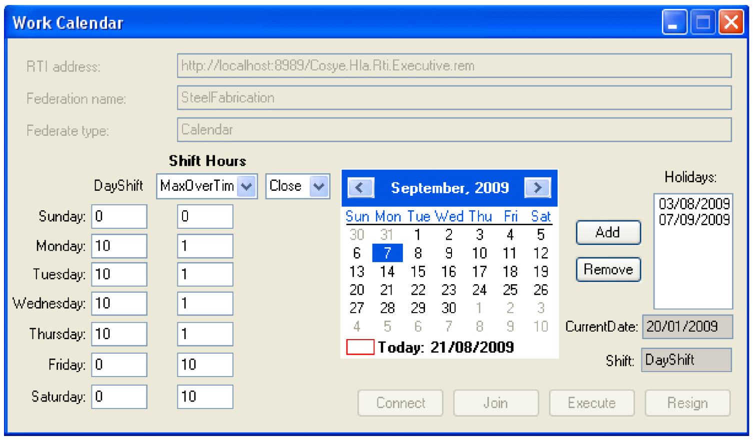


Figure 2-8a. Developed form for the calendar federate.

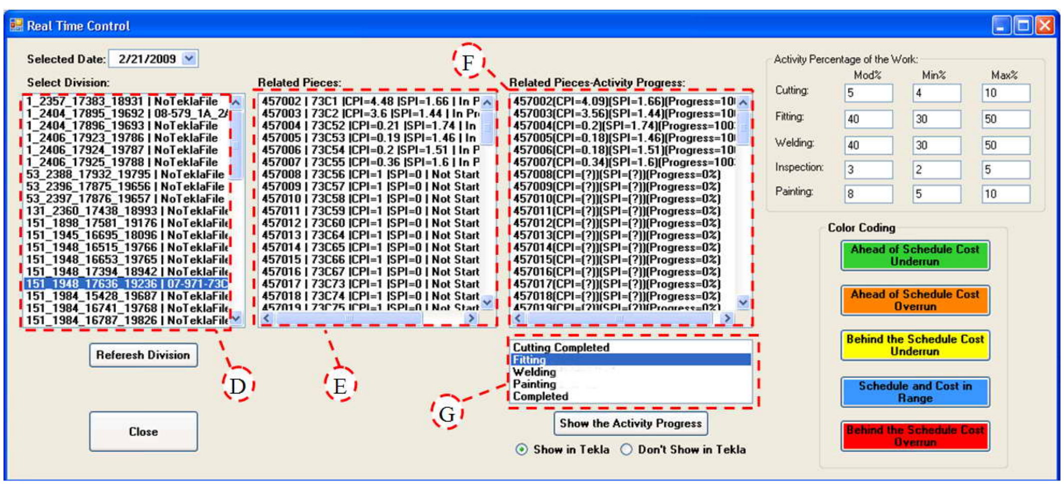


Figure 2-8b. The visualization federate.

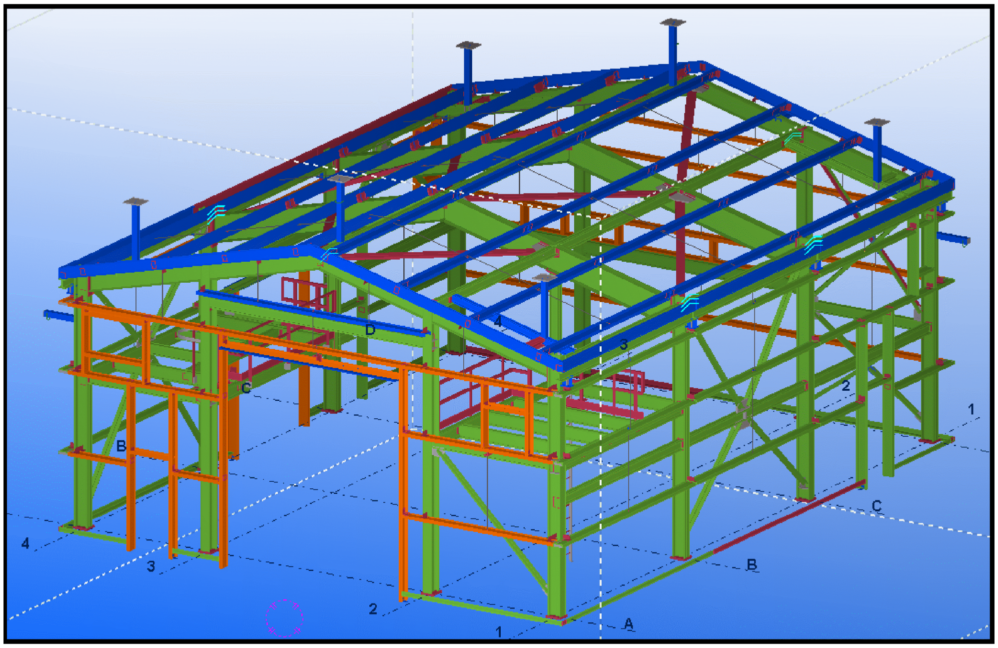


Figure 2-8c. Intuitive progress report (5D model).

Figure 2-8. Sample federates developed for the monitoring and control federation

2.7. Framework verification

To verify the functionality and effectiveness of the developed automated monitoring and control system, a pilot case study was conducted in a steel fabrication shop. Portable tag readers and RFID tags were used for real-time tracking of labour and materials at different stages (e.g., fitting, welding, painting, and shipment). The data-acquisition system consists of RFID tags, RFID tag readers, a wireless router, and a host computer.

Each steel piece comes with a detailed drawing to be processed. These drawings have unique drawing numbers and, for tracking purposes, each RFID tag is assigned to a drawing number in the database. Since steel causes serious interference and loss of performance in the RFID tags, a new setup was utilized to overcome this issue. The tags were put on a foam backing with a thickness of 5 mm, while magnet strips were used on the other side of the backing to stick the tags to the steel elements (Figure 9). This ensures that the tags are removable and that the position of the tags can be changed by the operator if required. Also, at the end of tracking, the tags can be used for another project (reusability of tags).

Figure 9. Tagged pieces ready to be shipped.

In this verification, the automated data capturing system collected data regarding piece location and piece processing duration at each area. The collected data was successfully transferred to the host computer and the MIS database was updated in real-time. Two additional computers located at different locations were involved in the verification process. One machine took care of the developed PI and Visualization federate, while the other was responsible for the DES federate. Communication and real-time data exchange between the machines took place over the internet and was coordinated by the RTI. The PI federate calculated the cost and schedule performance indices automatically, while actual performance of ongoing projects could be immediately observed in the visualization federate. Though the PI federate is designed to identify resource performance indicators, for the purpose of this verification, the defined target for this federate was to determine project performance indices using the Earned Value Method. Table 2-1 shows sample as-built data captured automatically and the outputs of the PI federate for performance evaluation.

Table 2-1. Earned value analysis for steel pieces.

Piece ID	BCWS (Manhrs)	BCWP (Manhrs)	ACWP (Manhrs)	CV (Manhrs)	SV (Manhrs)	CPI	SPI
471731	100	85	122	-37	-15	0.70	0.85
471779	110	70	65	5	-40	1.08	0.64
471712	70	60	60	0	-10	1.00	0.86

Where:

ACWP : Actual Cost of Work Performed

BCWP : Budgeted cost of work performed

BCWS : Budgeted cost of work scheduled

CV: Cost Variance = BCWP-ACWP

SV: Schedule Variance = BCWP- BCWS

CPI: Cost Performance Index= BCWP/ACWP

SPI: Schedule Performance Index= BCWP/ BCWS

CV and CPI indicate if there is a budget over-run (CV<0 or CPI<1) or budget under-run (CV>0 or CPI>1). The expenditure is according to plan provided that CV=0 (or CPI=1). On the other hand, SV and SPI determine if the progress is behind schedule (SV<0 or SPI<1) or ahead of schedule (SV>0 or SPI>1). The progress is according to plan when SV=0 (or SPI=1). As an example, in Table 2-1 for the piece with ID #471731, one can conclude that the piece is behind schedule (i.e., SPI=0.85 <1) and has a budget over-run (i.e., CPI= 0.7 <1). Budget over-run here refers to spending man-hrs greater than what has been planned for processing that piece.

The captured data was not only used to determine actual performance indices, but also to feed the DES federate. A number of variables (e.g., the number of active stations and available operators) were defined for the DES federate. By running the simulation, the DES federate connected to the MIS federate, utilized the as-planned with as-built information as its input, predicted the project performance indices for any time in the future (which were intuitively presented in the visualization federate), and generated the updated schedule.

Table 2-2 represents the as-planned data (scheduled start dates and due dates) of several divisions versus their calculated start and finish dates and delays based on the simulation results.

Table 2-2. Sample results of the Simulation federate.

DivisionID	DivCalStDate	SchStDate	DueDate	DivCalcFinDate	Delay(Days)
3479	10/15/2008	10/15/2008	9/9/2008	10/15/2008	-36
13376	11/14/2008	11/14/2008	10/10/2008	11/14/2008	-35
13375	11/14/2008	11/14/2008	10/10/2008	11/14/2008	-35
13374	11/14/2008	11/14/2008	10/17/2008	11/14/2008	-28
13436	11/14/2008	11/14/2008	10/27/2008	11/14/2008	-18
13377	11/14/2008	11/14/2008	10/27/2008	11/14/2008	-18
15406	10/15/2008	10/15/2008	10/23/2008	10/15/2008	8
13974	11/14/2008	11/14/2008	12/12/2008	11/14/2008	28
15279	11/13/2008	11/13/2008	12/17/2008	11/13/2008	34

Ultimately, the successful deployment of the developed system confirmed the functionality and practicality of the proposed automated and integrated framework.

2.8. Future work

Despite the fact that the implemented DES federate takes care of certain aspects of performance forecasting and recommended corrective actions, these two functions are comprehensively under development as two explicit and separate federates. Meanwhile, the DES federate will be adjusted to cover other phases of steel construction, such as site erection. The extendibility of the federation enables it to add more components, which will make it more reliable. For example, a weather generation federate can be developed and added to the federation. This would ensure that weather conditions—which can significantly impact construction—are taken into consideration year-round. Once fully developed and to maintain accuracy and integrity, the entire system will be validated through further case studies of real projects.

Conclusion

In traditional control systems, manual data collection, improper data sharing, and the gap between monitoring and control usually result in late identification of deviations in project performance. This subsequently leads to late corrective actions, which often results in cost and schedule overruns. To address these issues, this chapter introduced an automated and integrated real-time monitoring and control framework and its components for the fabrication phase of steel construction projects. High Level Architecture (HLA) was employed as the infrastructure of the proposed framework, in which individual monitoring and control components cooperate and interact with each other in real-time. This framework is enhanced with a simulation component that uses as-planned and captured as-built data to predict the effects of different scenarios on project performance. An integrated model consisting of automated data capturing, a database, performance indicator, and simulation and visualization components was developed. This model then was implemented in a sample steel fabrication project to verify the concept and test the feasibility of the proposed framework.

This framework can be a useful decision support tool for project managers, as it utilizes actual real-time data to provide dynamic analysis and evaluation of the project performance of ongoing and future projects. This tool enables the detection of current and potential deviations, thus enabling project managers to take appropriate corrective actions in a timely manner. Successful verification of the functionality of this framework as well as its reusability, interoperability, and the extendibility of its federates, opens new doors for effective project monitoring and control in other construction domains.

References

AbouRizk, S.M. and Hague, S. (2009). "An Overview of the COSYE Environment for Construction Simulation." Proceedings of the winter simulation conference; Austin, Texas, USA, 2624-2634.

Abudayyeh, O. Y., and Rasdorf, W. J. (1991). "Integrated cost and schedule control automation." Proceedings of Construction Congress '91, 679-686.

Allmon, E., Haas, C.T., Borcherding, J.D., and Goodrum, P.M. (2000). "U.S. construction labor productivity trends, 1970-1998." Journal of Construction Engineering and Management , 126(2), 97-104.

Alshaibani, A. (1999). "A computerized cost and schedule control system for construction projects." Master's thesis, Concordia University, Montreal, Canada.

Azimi, R., Lee, S., and AbouRizk, S.M. (2009). "Automated project control system for steel projects." Proceedings of the Joint International Conference on Construction Engineering and Management and Construction Project Management (ICCEM-ICCPM), Jeju, Korea, 479-486.

Cheng, M. and Chen, J. (2002). "Integrating barcode and GIS for monitoring construction progress." Journal of Automation in Construction, 11(1), 23-33.

Cheok, G.S., and Stone, W.C. (2004). "Performance Evaluation Facility for LADARs." Proceedings of SPIE - The International Society for Optical Engineering, 54-65.

Chin, S., Yoon, S., Choi, C., and Cho, C. (2008). "RFID+4D CAD for progress management of structural steel works in high-rise buildings." Journal of Computing in Civil Engineering, 22 (2), 74-89.

Ciesielski, C.A. (2000). "ADCIC—automatic data collection in construction." Proceeding of the A.I.M.'s Global Academic and Scientific Conference, Cairo, Egypt.

Diekmann, J., and Al-Tabtabai, H. (1992). "Knowledge-based approach to construction project control." International Journal of Project Management, 10(1), 23-30.

Echeverry, D. and Beltran, A. (1997). "Bar-code control of construction field personnel and construction materials." Journal of *Computing in Civil Engineering,* 341-347.

Fayek, A.R. (2001). "Activity-based job costing for integrating estimating, scheduling, and cost control." Cost Engineering (Morgantown, West Virginia) 43(8), 23-30.

Fujimoto, R.M. (2003). “Distributed simulation systems.” Proceedings of the winter simulation conference; New Orleans, LA, USA, 124-134.

Gan, B.P., Liu, L., Jain, S., Turner, S.J., Cai, W., and Hsu, W. (2000). “Distributed supply chain simulation across enterprise boundaries.” Proceedings of the winter simulation conference, 1245-1251.

Ghanem, A. G. and AbdelRazig, Y. A., (2006). “A Framework for Real-time Construction Project Progress Tracking.” Proceedings of the Tenth Biennial ASCE Aerospace Division International Conference on Engineering, Construction, and Operations in Challenging Environments, 1-8.

Kim, S., Park, C., Lee, S., and Son, J. (2008). “Integrated cost and schedule control in the Korean construction industry based on a modified work-packaging model.” Canadian Journal of Civil Engineering, 35(3), 225-35.

Kuhl, F., Weatherly, R., and Dahmann, J. (2000). “Creating computer simulation systems: An introduction to the high level architecture.” PTR Upper Saddle River, NJ, USA: Prentice Hall.

Law, AM. (2006). “Simulation modeling and analysis.” McGraw-Hill, Inc. New York, NY, USA.

Lee, S., and Peña-Mora, F. (2006). "Visualization of Construction Progress Monitoring." Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada.

Li, J. (2004). "Web-based integrated project control." Ph.D. thesis, Concordia University, Canada.

Li, J., Moselhi, O., and Alkass, S. (2005). "A reasoning process in support of integrated project control." Canadian Journal of Civil Engineering, 32(3), 500-16.

Moselhi, O., Li, J., and Alkass, S. (2004). "Web-based integrated project control system." Journal of Construction Management and Economics, 22(1), 35-46.

Nasr, E.B. (2005). "An integrated project planning and control system approach for measuring project performance." Ph.D. thesis, University of Colorado at Boulder, USA.

Nassar N., Halepota T. and AbouRizk S. M. (2003). "Integrated project controls system - IPCS." Proceedings of the Annual Conference of the Canadian Society for Civil Engineering, 51-60.

Navon, R. (2005). "Automated project performance control of construction projects." Journal of Automation in Construction, 14(4), 467-76.

Navon, R. and Goldschmidt, E. (2003). “Can labor inputs be measured and controlled automatically?” Journal of Construction Engineering and Management, 129(4), 437-45.

Navon, R. and Sacks, R. (2007). “Assessing research issues in automated project performance control (APPC).” Journal of Automation in Construction, 16(4), 474-84.

Oberlender, G.D. (2000). “Project management for engineering and construction.” The McGraw-Hill Companies, Inc., Boston, MA.

Po-Han, C. (2008). “Integration of cost and schedule using extensive matrix method and spreadsheets.” Journal of Automation in Construction, 18(1), 32-41.

Pritsker, A. (1986). “Introduction to simulation and SLAM II.” John Wiley & Sons, Inc., New York, NY, US.

Pritsker, A., Alan, B. and Hammesfahr, R.W. (1989). “Slam II network models for decision support.” Englewood Cliffs, Prentice-Hall, Englewood Cliffs, NJ, US.

Sacks, R., Navon, R., Brodetskaia, I., and Shapira, A. (2005). “Feasibility of automated monitoring of lifting equipment in support of project control.” Journal of Construction Engineering and Management, 131(5), 604-14.

Sacks, R., Navon, R., and Goldschmidt, E. (2003). “Building project model support for automated labor monitoring.” Journal of Computing in Civil Engineering, 17(1),19-27.

Santoro, A. and Fujimoto, R.M. (2008), “Offloading data distribution management to network processors in HLA-based distributed simulations.” IEEE Transactions on Parallel and Distributed Systems, 19(3) , 289-98.

Singh, A. (1991). “Knowledge bases for C/SCSC.” Cost Engineering (Morgantown, West Virginia), 33(6), 39-49.

Deng, Z.M., Li, H., Tam, C.M., Shen, Q.P., Love, P.E.D. (2001). “An application of the Internet-based project management system.” Journal of Automation in Construction, 10 (2), 239–246.

Chapter 3. Intelligent Look-Ahead Scheduling for Structural Steel Fabrication Projects [3]

3.1. Introduction

The complexity and variety of products, and the large number of potential resources, activities, interactions, constraints and uncertainties, make the planning and control of ongoing and forthcoming steel fabrication projects a complicated task. Project planning involves several activities, including master scheduling and short-term or look-ahead scheduling, which are two key elements in successful delivery of the projects. Master scheduling refers to the overall view of the projects and general fabrication strategies. Such scheduling may be used for several reasons: forecasting demand, long-term coordination (e.g., regarding material requirements and staffing level) and rough budgeting. However, master scheduling suffers from a lack of information about actual durations and cannot be properly detailed far into the future. Conversely, a look-ahead schedule is a detailed plan for work packages to be completed in a relatively short time frame. Look-ahead scheduling helps project managers focus on the work packages that should be done at some time in the future and the corrective actions in the present

---

[3] A version of this chapter has been published in the International Journal on Advances in Software, 2011, Vol. 4, No 1&2, pp. 207-217. Some introductory material has been removed to avoid duplication.

that will lead to finishing those work packages on time, within budget and to a specified quality. These detailed schedules should be developed and updated in a timely manner based on the actual project performance data and the conditions in the construction environment, to precisely represent the tasks that have been done and the tasks that remain (Hinze 2008). Such a detailed schedule is a solid foundation in terms of performance analysis and taking effective corrective actions.

This chapter amplifies the work originally presented by Azimi et al. (2010a) and represents a successful implementation of the project monitoring and control framework proposed by Azimi et al. (2011).

This chapter proposes a steel fabrication shop modeling approach for efficient look-ahead scheduling of steel fabrication projects in the Construction Synthetic Environment (COSYE) based on the High Level Architecture (HLA) infrastructure. In this approach, real-time, high-quality actual project data are captured and fed into a simulation model along with the uncertainties and the factors influencing the productivity of the fabrication shop. In this way, reliable updated look-ahead schedules are generated by the simulation model and "guesstimates" are rarely needed. Current practice and research carried out regarding look-ahead scheduling is discussed, and the conceptual framework of the integrated real-time simulation-based scheduling system is then described. The feasibility of the proposed framework is demonstrated by developing a prototype system that was used for a case study in a steel fabrication shop. The advantages and limitations of the proposed system are also discussed in this chapter.

3.2. Background and literature review

The basis of Look-ahead Scheduling (LAS) is similar to regular scheduling. Usually, activities or work packages are regular and predictable. When that is not the case, work packages need to be appropriately classified and work packages that are considered similar may differ in terms of process duration and required resources. Similar to regular scheduling, standard data should be generated using time studies or expert judgment. Once these standards are established, work packages can be scheduled, allowing foremen and project managers to forecast and control projects over comparatively short time intervals. LAS usually refers to a foreman's schedule, settled and tracked by foremen on a short-term basis. The foremen determine which work packages would be processed by their crew during the next few days, and the project managers monitor the accuracy of the schedule. This monitoring leads to identifying factors that affect the production rate. Once these factors are identified, the project managers can address the implicit causes and routinely enhance the accuracy of the LAS. LAS helps improve productivity by eliminating or reducing time spent that is not adding value to projects. It also helps ensure all the required resources and material are ready for ongoing projects at any time needed. Since LAS sets realistic and obtainable goals for a short time span, as a psychological effect, the workers tend to get the work done as soon as possible.

Smith (1981) argues that LAS has been successfully implemented in different domains such as material handling, quality assurance, manufacturing, maintenance, engineering, and assembly operations. LAS has been largely used for mass production systems where immediate follow-up and corrective actions are a must in the case of deviations (Ramírez-Valdivia et al. 2003). As an example, combined with pairwise comparison LAS was utilized for scheduling random operations in job-shops (Golenko and Gonik 1997).

In spite of the fact that very little information on LAS is provided in the literature, effective LAS are crucial to the successful completion of construction projects (Hinze 2008). Daneshgari and Moore (2009) state that on average up to 70% of construction job schedules experience changes. They observed four projects over a four-month period, ranked the impact of the unscheduled activities on the lost productivity and concluded that implementation of LAS is a great tool to improve the productivity. Similarly, Hadavi and Krizek (1993) concluded that short-term scheduling results in higher productivity compared to long-term scheduling. Studying decision support systems in manufacturing operations and determining the types of data required to plan and control effectively, Schmahl (1996) concluded that LAS can be used to support continuous improvement efforts in production operations. Scheduling problems could be solved for either the next hour or the next few weeks by LAS (Zweben and Fox 1994). Guidelines regarding developing work packages for effective utilization of LAS as well as the situations where LAS is more likely to succeed have been discussed by Ramireza-Valdivia et al. (2003). Finally, emphasizing the key role of LAS in enhancing production control, Ballard (1997) proposed strategies for improving LAS.

In construction, many operations are repetitive and involve uncertainty and resource constraints. This motivated researchers to deploy discrete-event simulation for LAS problems (Song et al. 2009). Simulation is a mathematical-logical model representing a real-world evolving system. Users can use the simulation model to analyze and forecast the performance of a system considering different scenarios. Actual data captured from ongoing projects along with the uncertainties of the construction environment can be fed into the simulation model to "tune it up" for generating better results (Chung et al. 2006 and Lu et al. 2007). A proper updating process for simulation input modeling based on high quality data is necessary to achieve simulation accuracy. The emergence of new technologies has enhanced data acquisition systems by providing automated high-quality real-time data. For example, Radio Frequency Identification (RFID) has been used to track real-time locations of steel pieces in a steel fabrication shop (Azimi et al. 2011) and monitor steel works in high-rise buildings (Chin et al. 2008). The same data acquisition system based on RFID technology developed by the authors (Azimi et al. 2011) is also used in the prototype system of this research to closely supervise the operations that lead to obtaining the benefits of LAS.

Current studies focus on real-time data acquisition and improving simulation input modeling, which involves finding statistical distributions of the model input parameters, such as the durations of different activities (Song et al. 2009). For instance, Song et al. (2009) used Global Positioning System (GPS) technology to capture required data, such as truck hauling time, and update a simulation input model for LAS of asphalt hauling and paving projects. These simulation systems commonly have two characteristics: first, they usually model repetitive operations, and second, the final output of these operations is one or (rarely) a few limited products. This chapter deals with simulating a steel fabrication shop, in which the operations are repetitive while each product (i.e., steel piece) is typically unique. This uniqueness means the time required for processing each steel piece varies. Estimating processing duration is dependent on productivity; the degree of precision depends on the nature of the work and is influenced by several factors. The relationship between these factors and the processing duration/productivity cannot be demonstrated in an accurate and clear manner, increasing the difficulty of estimating the steel piece processing activity durations which are used for updating the simulation input model. A well-structured Artificial Neural Network (ANN) model, which has an optimal structure regarding layers and nodes, is capable of learning from data sets and reliably approximates any complicated relationships between dependent and independent variables (Bishop 1995). ANN models can also handle moderate amounts of noise, which is common in the historical data, and can generate knowledge from defective or noisy data (Swingler 1996). ANNs have been widely used for modeling productivity in construction; for example, concrete construction productivity (Sonmez and Rowings 1998), formwork production rates (AbouRizk et al 2001) and pipe spool fabrication productivity (Lu et al. 2000). ANNs have also been exploited in this research to intelligently generate process durations for each steel piece to be manufactured, considering influencing factors. A framework is proposed to integrate a real-time actual data collection system for steel fabrication projects with an intelligent input data generation system and with simulation models, which utilizes the as-built data for updating input models and improving the simulation results for LAS.

3.3. Look-ahead Scheduling using simulation technique

In steel fabrication projects, LAS involves a number of uncertain factors and constraints. No project can be started earlier than a given date due to the limitations regarding the availability of required material, space and equipment. Each project should be delivered by a certain date depending on client demand and the conditions of the erection site. The scheduler should also take into account the limits on other resources such as skilled workers, cranes and active stations in the steel fabrication shop.

A typical steel fabrication project may contain a few hundred steel pieces, which makes using traditional techniques such as the Critical Path Method (CPM) a time-consuming and tedious exercise. Moreover, in the case of any deviation from the baseline or changes in resource availability, adjusting the schedule would be very difficult. Computer simulation is a powerful technique to efficiently react to system changes and generate updated schedules. Simulation is used here as the underlying technique to model the fabrication shop and resources and activities required to process steel pieces. To be effective, the simulation model should be updated based on the most recent system changes. Then, the impacts of these changes need to be observed by the model for modifying the LAS. To address this, this chapter presents an intelligent and integrated simulation system based on the framework previously developed by the authors for automated and integrated project monitoring and control (Azimi et al. 2011). Several components of the framework established using the HLA infrastructure have been modified and the whole system is enhanced by adding intelligence for reliable look-ahead scheduling purposes. An as-planned database, as-built data acquisition, discrete event simulation, a steel fabrication process knowledge base, a calendar, and an intelligent adjuster are the major components of the proposed system (Figure 3-1).

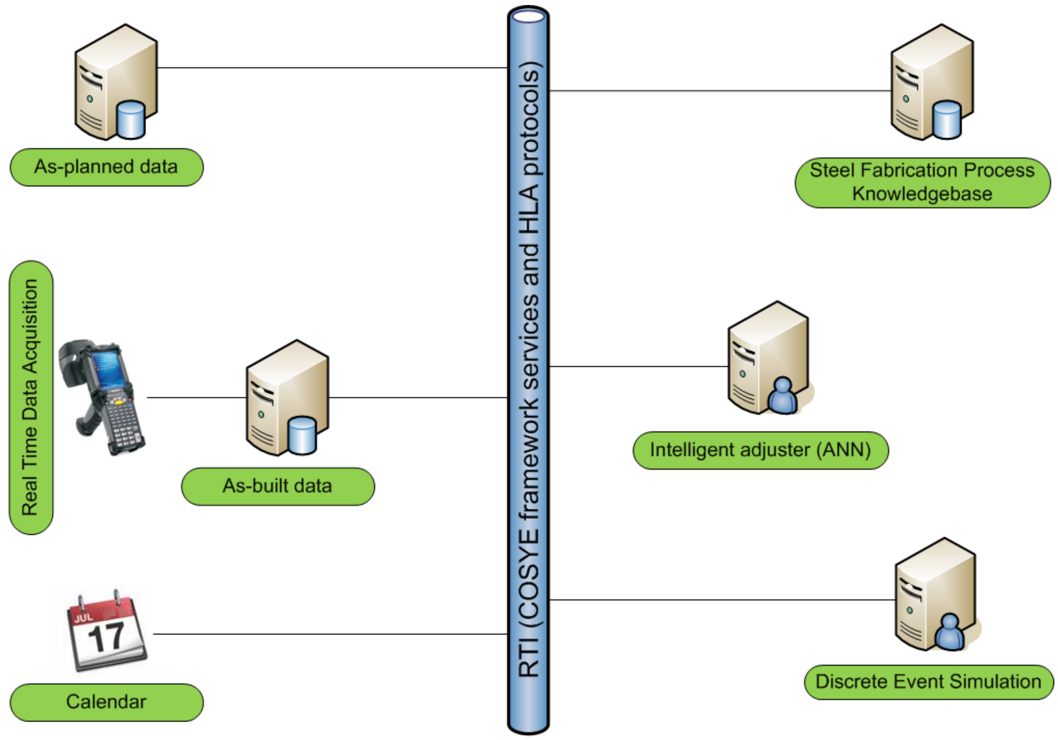


Figure 3-1. Integrated LAS system- a modified version of the model originally proposed by Azimi et al. 2011.

HLA provides a reliable infrastructure for efficient integration of all the components of the LAS system. Interoperability, reusability, flexibility, and system speedup (Azimi et al. 2011) are some advantages of utilizing HLA as the backbone for the proposed LAS system.

To have an effective LAS system, the following steps must be taken. First, the process of the steel fabrication should be well investigated and a simulation model based on that is established. For each project a baseline schedule is defined by a scheduler. During the execution course of the project, real-time data is collected from the fabrication shop and utilized to update the simulation model so that it reflects the dynamic nature of the project environment. Once the initial state of the simulation model is set, it can be used for experimenting with different scenarios and an updated LAS can be generated based on the simulation results.

A brief review of the steel fabrication process as well as the components of the proposed system is presented in the following sections to provide the reader with an overview of the research topic.

3.3.1. Steel Fabrication Process

A typical steel project consists of a number of steel pieces, such as beams, columns, or trusses, with different dimensions and specifications. Fabrication of steel pieces starts with the detailing area. In this area, the components of a steel piece are cut and/or punched according to the engineering design. Cut components are transferred to the fitting area to be fitted. Fitters bring the components together using tack welds to form the steel piece. Once inspected, the fitted piece is sent to the welding area where the welders weld the piece according to the provided specifications. Another inspection happens once the welding is done. If required, the piece is sent to the painting area, otherwise the piece is ready to be shipped to the erection site (Figure 3-2).

Each area in the fabrication shop is composed of several stations which makes it possible to have several pieces processed in each area simultaneously. Steel pieces are transferred by rail carts or cranes depending on the situation. If a piece must be processed in a working area but there are not enough resources available to process that piece, it is piled in a certain storage area and waits until the required resources are available.

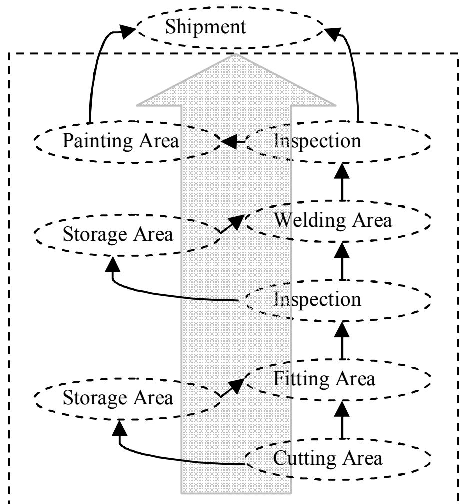


Figure 3-2. Steel fabrication process.

Rework will be necessary if a piece is rejected by an inspector.

3.3.2. Components of the Proposed System

The HLA serves as the backbone in the development of the proposed system. Some components of the proposed system such as Discrete Event Simulation, the calendar, and real-time data capturing have already been developed and detailed information about these components can be found in (Azimi et al. 2011). A brief explanation of those components is also provided below.

a. Real-Time Data Capturing

As in (Azimi et al. 2011), Radio Frequency Identification (RFID) technology has also been used in this research to collect real-time data. RFID tags are put on steel pieces and the locations of the tagged pieces are tracked using portable RFID tag readers. The location tracking also captures the time a piece enters each area and the time it leaves that area. These data are then interpreted to provide project performance data, project progress, and activity duration in the steel fabrication process knowledge base component. Inter-process communication between the tag readers and the as-built database occurs via Transmission Control Protocol over Internet Protocol (TCP/IP).

b. Steel Fabrication Process Knowledge Base

The real-time data acquisition system generates raw data. Useful performance data, such as man-hours spent on fitting or welding each piece, project percent complete, production rate and so on need to be extracted from these raw data to be used for updating LAS. Interpretation of these raw data can be automated by correlating the location of a steel piece with fabrication events and activities based on the experts' knowledge regarding the process of fabrication. A comparison between the location of a steel piece and pre-defined areas discloses meaningful information about the fabrication operation. For example, the entrance of the steel piece into the fitting area is recorded as a start-fitting event and exiting the fitting area is considered as an event called end-fitting. Once done, it is time to extract activity information from the event data. Generally speaking, each activity is begun by an event and is finished by another event. So, for the previous example, the duration of fitting activity for each steel piece can be calculated considering the start- and end-fitting events. To be effective, the gained knowledge should be managed properly in terms of capture, arrangement and retrieval of knowledge. To enforce that, knowledge bases are frequently used by the practitioners and researchers. As an example a process knowledge base for asphalt hauling is developed in (Song et al. 2009). A sample hierarchical structure of the required knowledge for structural steel fabrication data interpretation is presented in Table 3-1.

Table 3-1. Sample elements of the steel fabrication process knowledge base.

Area	Action	Event	Activity
Fitting Area	Enter	Start fitting	Fitting steel piece
Fitting Area	Leave	End fitting	Waiting (storage area)
Welding Area	Enter	Start welding	Welding steel piece
Welding Area	Leave	End welding	
…	…	….	

Different techniques can also be used to calculate other performance data. For example, the progress of each piece is determined by translating piece location to percent complete using the rule of credit based on expert knowledge (Azimi et al. 2011). For instance, the experts may consider the stage, "piece fitting", as 40% progress in the fabrication phase, i.e., if a steel piece gets fitted, 40% of the required processes is accomplished. Project percent complete can thus be calculated by summing up the weight of each piece times its progress, all divided by the weight of the project.

c. Discrete Event Simulation

This component is a modified and improved version of the discrete event simulation (DES) model of steel fabrication shops developed earlier (Azimi et al. 2011). Modeling the fabrication shop and prediction of the behavior of the fabrication shop is not based on mathematical models and analytical solutions. Discrete event simulation was utilized for this purpose due to the fact that simple closed form analytical solutions are not available for the modeling steel fabrication process. However, the DES federate has the capability to incorporate any available mathematical or analytical solutions.

To be effective, the DES component of the proposed LAS system should be updated in a timely manner to reflect changes in the fabrication shop's environment. The proposed DES is a self-adjustive component that exploits the captured actual real-time data to update its input models. The initial state of the simulation model is set based on the most recent data. The actual data can also be used to generate updated distributions that are substituted with the earlier input model. The intelligent adjuster is a component that takes care of this responsibility. It automates the input-model update procedure with no slow and error-prone human involvement.

Scheduling is performed in DES model based on the earliest-due-date (EDD) dispatching rule, such that the project with the earliest due date is selected and processed first. The scheduling engine of the DES model enables automated project schedule updates to be generated and stored in an MS Access database. The DES model is also capable of performing earned value analysis, and cost and schedule performance indices can be calculated for all the steel pieces.

d. Intelligent Adjuster

Updating input models can be carried out by external prediction models (AbouRizk and Sawhney 1993). The proposed intelligent adjuster is an autonomous Artificial Neural Network (ANN) component that is trained with the actual data available in the aforementioned steel fabrication process knowledge base to generate updated distributions, e.g., regarding duration of different activities required to process each piece of steel for the simulation input model. While it is difficult and sometimes not feasible for estimators to consider in their estimations all the influencing factors for a huge number of steel pieces with different dimensions and specifications, artificial intelligence has the capability to overcome this issue and forecast and update the distributions to be used in the steel fabrication simulation model. In this way, the accuracy of the simulation model is enhanced by explicitly modeling uncertainty variables and their impacts on the performance of the fabrication shop and ongoing projects.

e. Calendar

Schedules are highly influenced by day shift hours, night shift hours, overtime hours, and holidays. The user defines these parameters within the calendar federate which sets related initial values for the DES model (Azimi et al. 2011).

3.4. System implementation

3.4.1. Infrastructure

A prototype system has been developed for look-ahead scheduling of steel fabrication projects. The proposed system implements the Construction Synthetic Environment (COSYE) software environment discussed in detail in (Azimi et al. 2011).

In this prototype system, the primary project LAS (baseline) is prepared by a scheduler and is stored in a MS Access database. Captured real-time actual data are also stored in a MS Access database. The base model of the DES component (federate) was developed within the Simphony.NET simulation environment (AbouRizk and Hajjar 1998) for modeling steel fabrication operations. Figure 3-3 depicts the interface for the DES federate (Alvanchi et al. 2010).

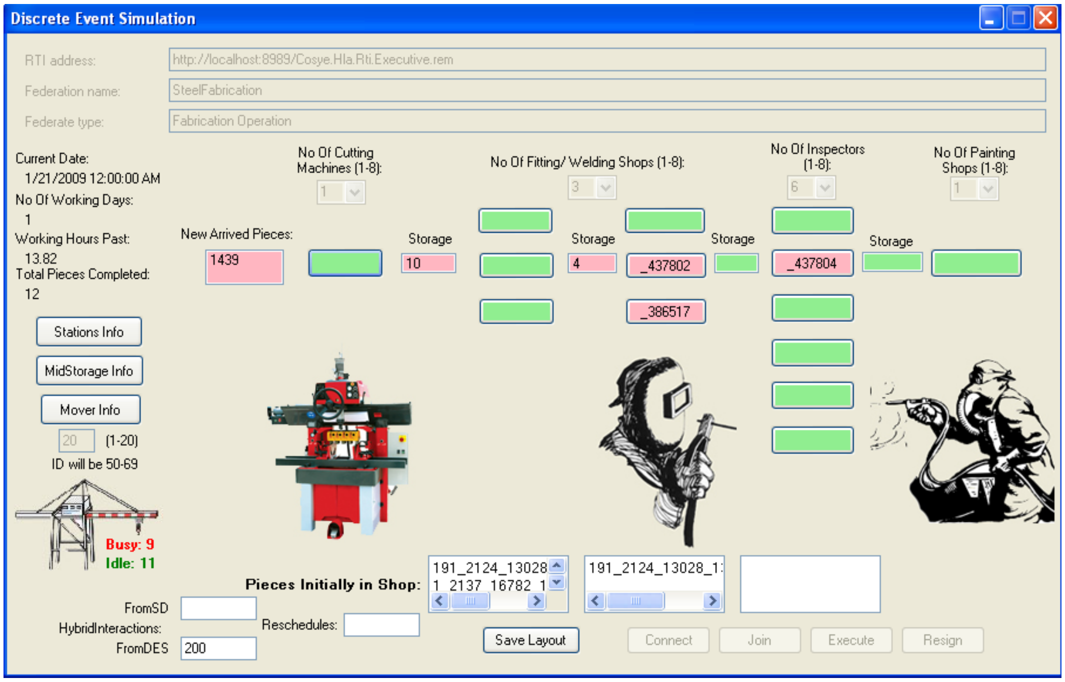


Figure 3-3. The DES federate (Alvanchi et al. 2010).

3.4.2. Factors affecting the intelligent adjuster's forecasting

The process knowledge base, demonstrated in Table 3-1, is used to extract fabrication activity information from the actual data and feed the intelligent adjuster component.

Fitting and welding operations are the critical operations in the structural steel fabrication and usually take more than 80% of the available resources on average. In the proposed system the intelligent adjuster is composed of two Artificial Neural Network (ANN) models and predicts the steel fitting and welding productivity/durations based on the complexity of each steel element and other influencing factors. Influencing factors can be divided into two major categories: the steel piece itself and the fabrication shop environment. Song et al. (2005) proposed four piece-oriented influencing factors, such as number of fittings, number of cutouts, piece length, piece weight, and two influencing factors regarding the fabrication shop environment for the fitting operation, such as worker rank and work shift. There are two concerns regarding the proposed piece oriented factors. Firstly, although it is clear for beams and columns, piece length is a vague concept for other steel pieces such as frames and stairs. As an example, for a square steel frame, one person may consider the side length as the piece length while another person may use the diagonal of the square as the piece length. Piece dimension is an important factor because piece movement and piece flipping in each operation are highly affected by this property of steel pieces. However, piece weight commonly has a close and positive correlation with piece dimension (i.e., the bigger the piece, the greater its weight). This means that by considering piece weight as an influencing factor the piece dimensions are implicitly addressed. Secondly, the number of fittings and cutouts are two influencing factors in the fitting operation but not necessarily for the welding operation. Two different approaches can be taken considering the influencing factors. One approach is to define the factors for each specific operation, while the other approach is to define general influencing factors that can be used for different operations in steel fabrication. Within this research the second approach has been taken because of its universality and usage in the whole fabrication process. Therefore, the influencing factors considered in this research that are related to the characteristics of the steel piece include piece complexity –replacing the number of fittings and cutouts – and piece weight; the ones that are related to the shop environment are the rank of the workers and the working shift in which the pieces are manufactured.

Having said that, the inputs of the intelligent adjuster are the weight of the steel element (piece/assembly), the shift in which it has been fabricated, the rank of the worker who has processed that steel element (the higher the rank the more experienced the worker) and piece complexity. Piece complexity is represented by a parameter called "complexity factor." "Complexity factor" refers to the number of the components in each steel piece. For instance, the steel beam shown in figure 4a is composed of an I-shaped beam and 5 stiffeners. Thus, there are 6 components forming that steel piece which results in a complexity factor equal to 6. Steel pieces differ, sometimes significantly, regarding their complexity factors. In other words, while some steel pieces are fairly simple (e.g. Figure 3-4a), some pieces can be considerably more complicated (e.g., Figure 3-4b).

The complexity factor can also be calculated for each division with the same concept, i.e., total number of steel components divided by the number of steel pieces forming the division. To reduce the calculation burden on the end users, a template was developed to automatically capture complexity factors and weight of steel pieces from 3D models (Figure 3-5) of steel projects. These 3D models use

Building Information Modeling (BIM) which is a common way to construct a building virtually before building it in the real world, bringing the structures from concept to reality (Brown 2009).

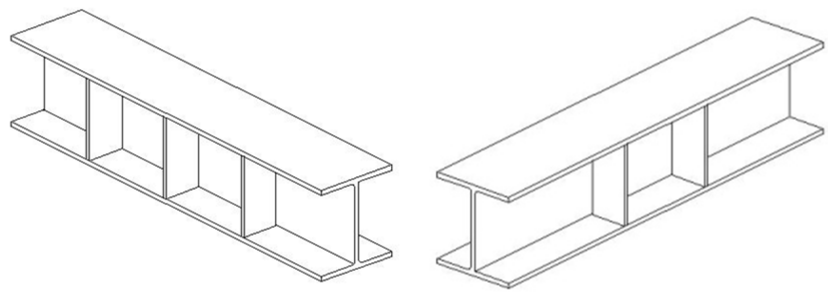


Figure 3-4a. Two side views of a steel beam with 5 stiffeners.

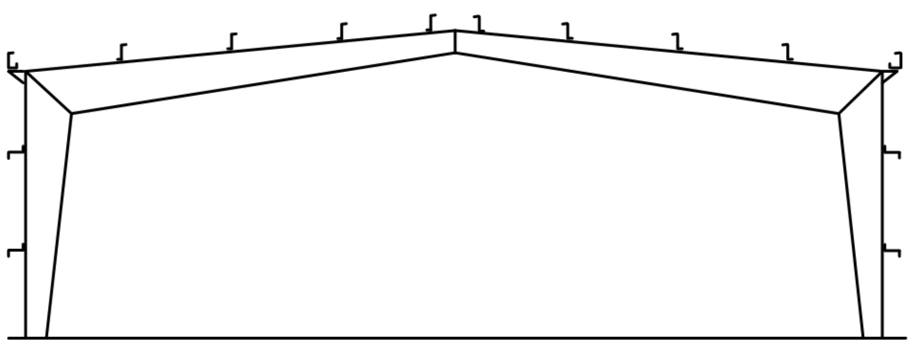


Figure 3-4b. A steel frame consisting of tapered beams and columns with a large complexity factor.

Figure 3-4. Sample of structural steel pieces with different complexities.

Automatically capturing information from 3D models with minimum human involvement guarantees high-quality data with great speed for data analysis purposes.

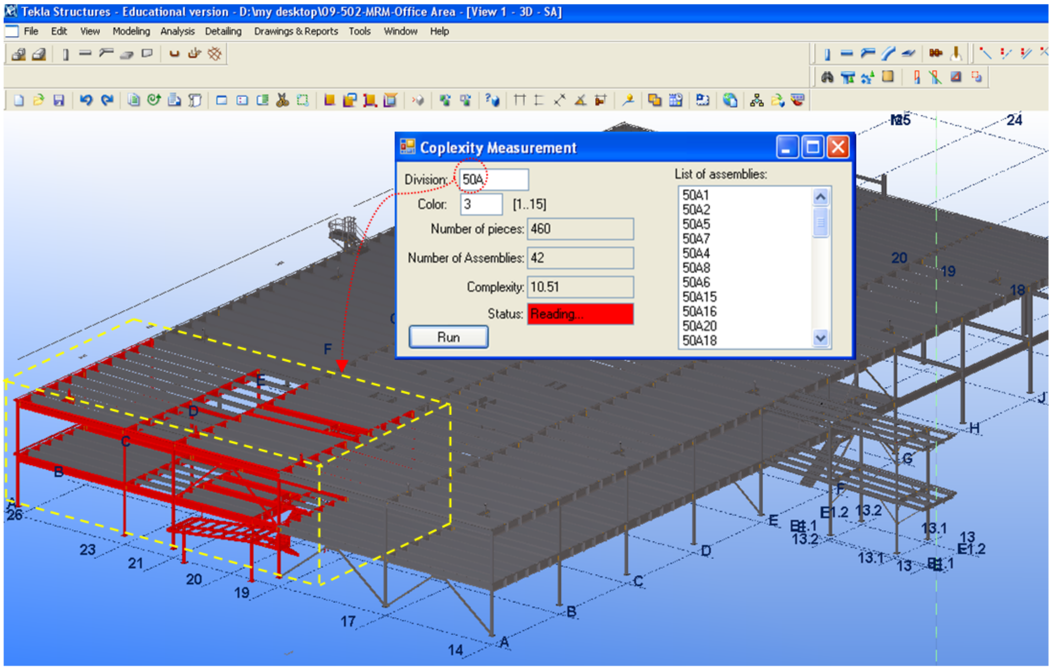


Figure 3-5. Developed template for capturing complexity factor from 3D models

A sample of data captured automatically from a 3D model by the developed template is represented in Table 3-2.

The intensity of complexity is an indicator that determines how complicated a steel element is (i.e., piece complexity) and has a direct relationship with the complexity factor. It is one of the input variables utilized for training the intelligent adjuster and is defined as in Table 3-3.

Table 3-2. Sample of data captured by the developed template from 3D models.

Piece ID	C.F.*	I.C.**	Weight(kg)
50A1	12	4	582
50A2	17	5	555
50A5	10	3	529
50A7	17	5	539
50A4	10	3	529
50A8	17	5	542
50A6	11	3	519
50A15	19	5	293
50A16	9	3	280
50A20	9	3	132
50A18	7	2	137

* Complexity Factor

**Intensity of Complexity

Table 3-3. Definition of the intensity of complexity.

I.C.	Definition	Explanation
1	Not complicated	$1 \leq$ C.F. < 4
2	moderately complicated	$4 \leq$ C.F. < 8
3	complicated	$8 \leq$ C.F. < 12
4	very complicated	$12 \leq$ C.F. < 16
5	extremely complicated	$16 \leq$ C.F.

3.4.3. Training the intelligent adjuster

The intelligent adjuster uses two back-propagation networks, each with 4 input nodes, one hidden layer, and one output node at the output layer.

The number of hidden neurons for the networks is calculated with the following equation (Neuroshell2 2010):

$$H_n=0.5\times(I+O) + \sqrt{P} \qquad (1)$$

Where:

H_n: Number of hidden neurons

I: Number of inputs

O: Number of outputs

P: Number of patterns

"Neuroshell 2" (Neuroshell2 2010) was used to train the networks. 114 fitting data points and 61 welding data points were captured by the real-time RFID data capturing system during a two-week time/case study used for training and validating the ANN models. For the fitting operation, 92 data points were randomly selected for training and 22 data points were used for testing. The learning rate and momentum of the fitting ANN model were set to 0.1. The initial weight of the links within the ANN model was set to 0.3 and numeric range of the linear scaling function used for the input layer was [-1,1]. For the welding operation, the number of training and testing data points were 49 and 12 respectively and the initial settings for learning rate, momentum etc. were similar to the settings of the fitting ANN model. The labor-driven nature of the steel fabrication process may lead to variance in productivity or activity duration even for similar steel pieces processed with laborers with the same rank and in the same shift. In this research one assumption is that such a variance is trivial; if that is not the case some techniques such as data filtration or using averages of the data can help in compensating for variances in the data. Sample actual data regarding welding operation that were used for training the intelligent adjuster are presented in Table 3-4.

Table 3-4. Historical data used for training the intelligent adjuster.

Weight (kg)	Shift (Day:1-Night:2)	Rank	I.C.	Duration (Minutes)
519	1	2	3	109
87	1	1	4	163
122	1	2	2	55
919	1	1	5	197
549	2	2	3	160
111	1	1	3	90
973	2	2	4	218
1250	1	3	4	121
…	…	…	…	…

The training results regarding duration of fitting and welding operations (in minutes) are summarized in Table 3-5. There are several factors that may affect the maximum absolute error in terms of duration prediction presented in Table 3-5. For instance, once a fitted or welded piece is rejected by an inspector, it requires rework, and the activity duration is extended and is greater than a situation in which rework is not required. Missing components of a steel piece or unclear or impractical drawings are other examples that extend the normal fitting/welding durations. With that said, and considering the wide duration ranges in fitting and welding data sets (i.e., from 10 to 290 minutes for the fitting data set and from 20 to 352 minutes for the welding data set), the trained networks are considered proportionately accurate in forecasting the fitting and welding durations with an acceptable margin of error.

Table 3-5. Intelligent adjuster training results.

Item	**Fitting**	**Welding**
Patterns processed:	114	61
R^2:	0.89	0.87
Mean absolute error:	13.46	26.14
Min. absolute error:	0.02	0.11
Max. absolute error:	73.20	53.00
Correlation coefficient r:	0.95	0.94

Two indicators, including R Squared (R^2) and the correlation coefficient (r), usually used for interpreting the neural network models, are also presented in Table 3-5. R^2, the coefficient of multiple determination, is a statistical indicator that refers to a multiple regression fit. This indicator compares the accuracy of the model to the accuracy of a benchmark model in which just the mean of all of the samples is used for prediction (Neuroshell2 2010). The closer the R^2 value is to one, the more qualified the model. As indicated in Table 3-5, the R^2 values for the fitting and welding operations are 0.89 and 0.87 respectively. The strength of the relationship between the actual and predicted outputs is statistically measured by the correlation coefficient (Neuroshell2 2010). This indicator's value can be any number between -1 and +1. The stronger the positive linear relationship, the closer r is to 1. According to the network training results presented in Table 3-5, the correlation coefficients for the fitting and welding operations are 0.95 and

0.94 respectively, which implies a strong positive relationship between the model outputs and the actual outputs for these operations. Figure 3-6 shows the trained network predictions against the actual welding duration values for the welding data points.

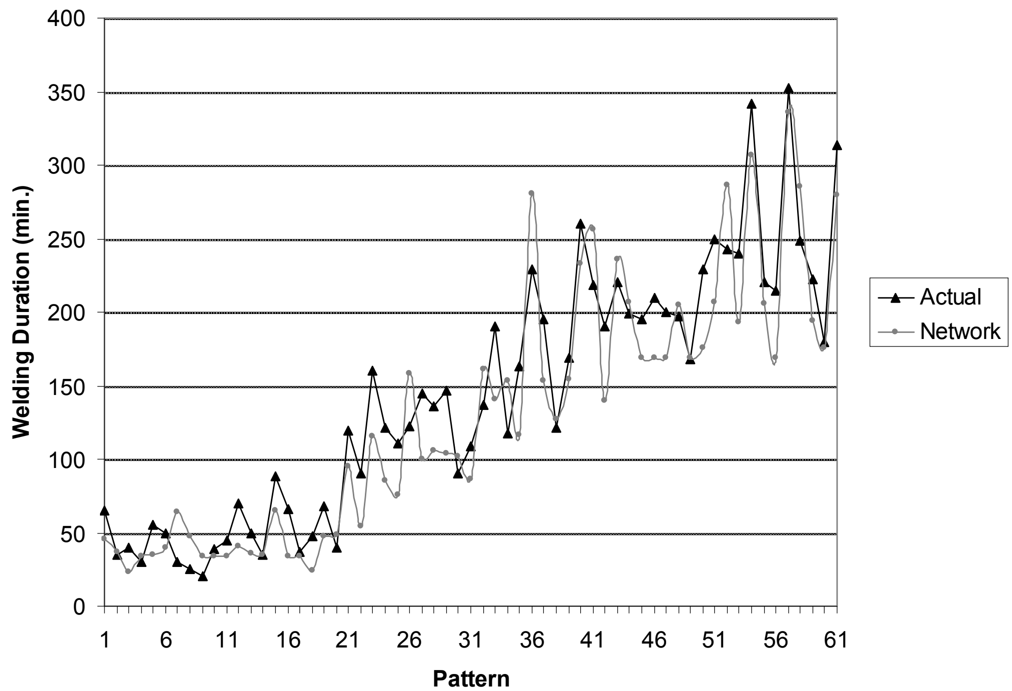


Figure 3-6. Actual vs. forecasted welding duration for data points.

The trained ANN models were compiled as VB.NET code and used for developing the intelligent adjuster federate. This federate is capable of generating the distributions and other scheduling variables, such as productivity rate, based on the attributes (e.g. weight, complexity intensity, etc.) of each steel piece. An example of the generated scheduling variables is shown in Figure 3-7. This figure is a snapshot of an interface developed to provide managers with an intuitive report on the forecasted trend of productivity rates for fitting and welding activities within a 3-day time interval ending at the date selected by the user. Such information can be a heads-up for impending deviations from desired values.

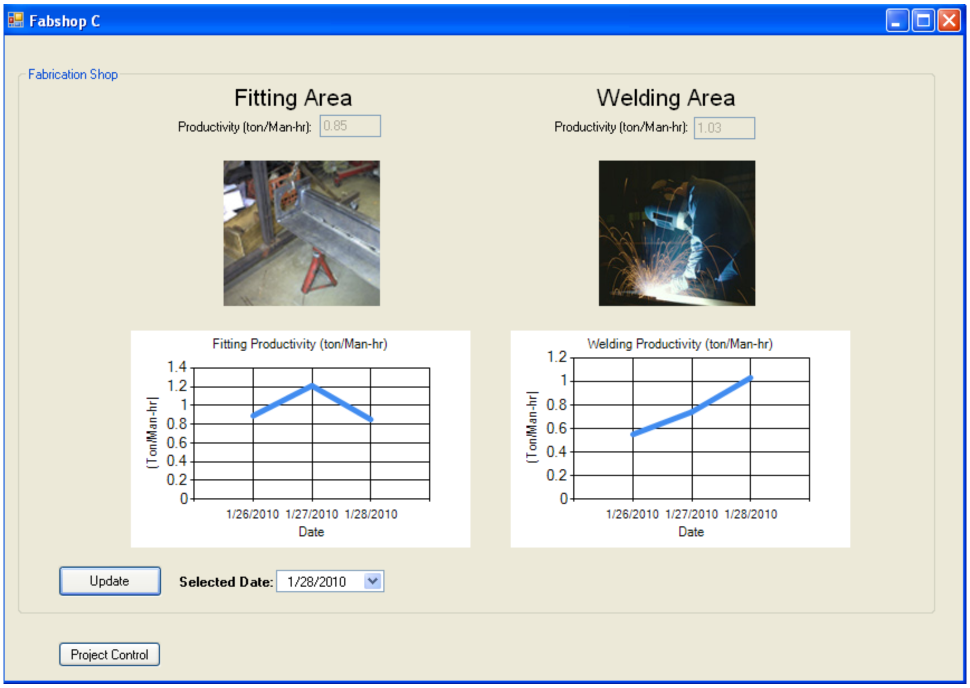


Figure 3-7. Fitting and Welding production forecasted by the intelligent adjuster component for the fabrication shop.

The generated distributions then will be fed into the DES federate to improve its results. If required, the user can modify the influencing factors and experiment with different scenarios with the simulation model to find the best corrective actions to converge on the project goals.

3.5. Case study

A case study, the construction of an administrative office, was carried out to verify the feasibility and accuracy of the implemented simulation-based LAS system for steel fabrication projects. Because of the size of the building, the whole project is divided into 57 “divisions.” Each division itself includes several hundred steel pieces, including beams, columns, stairs and frames. The baseline schedules for all the divisions were prepared by the scheduler and stored in the baseline database. The data acquisition system was set up in the fabrication shop. The major elements of the data acquisition system are RFID tags, a tag reader, and a router connected to a computer. Data capturing began with the launching of the project in the shop (Figure 3-8). The DES federate models four activities – cutting, fitting, welding, and painting. The resources required for this model include active stations (which correlate with the number of operators) in different areas, cranes and rail carts, inspectors, and storage areas.

Usually, several divisions are processed at the same time in a fabrication shop and certain stations in each area are assigned to each division. The data capturing was carried out for all the steel elements that were being processed in the fabrication shop. In this way, scheduling information for the project level as well as the performance information (such as production rate at different areas) at the shop level was determined.

Figure 3-8. Collecting actual data using RFID technology.

The actual data collected during the case study were used to examine the accuracy of the simulation model. However, during the case study several outstanding discrepancies were captured. First, some fitting and welding stations shut down due to the fact that a number of fitters and welders were laid off during the case study. Second, the actual process time in each station was longer than expected in many cases. Third, the production rate predicted by the simulation model was significantly greater than what happened in reality. These discrepancies detract from the utility of the LAS and mean that the simulation model needs to be properly modified and updated in some aspects to address these deviations.

Updating the simulation model is easy with regard to changing available resources (e.g., the number of active stations), but with variables such as process time and production rate, it is less straightforward. Variances in the process time and production rate can be caused by faults in estimation, external factors that were not considered in the simulation model, or both. While the faults in process time estimation are usually covered by the actual data, considering the effects of the external factors in the simulation model is a difficult task. For example, during the course of the case study, a number of workers received termination notices due to the recession conditions. After being informed that their employment will cease by the next month, the productivity of these workers was far lower than what was estimated. The arrival of a new project with a higher priority that had not been considered in the master plan was another issue that meaningfully affected the available resources (e.g., raw material), shop performance and LAS for ongoing projects during this case study. A thorough tracking of the variances regarding schedules and budget (i.e., assigned man-hours) resulted in determining seven major factors causing variances in the studied fabrication shop (Figure 3-9). Once these factors are tracked, they need to be addressed to improve the performance of the ongoing projects and the fabrication shop itself. Project managers and foremen try to make improvements in different areas that cause deviations and variances over the course of the fabrication, but it is still a major concern for them to have reliable LAS for different projects which take into account current situation in the fabrication shop. This is the benefit of the LAS system which has been developed, which enables users to have updated LAS automatically generated based on the most recent actual data.

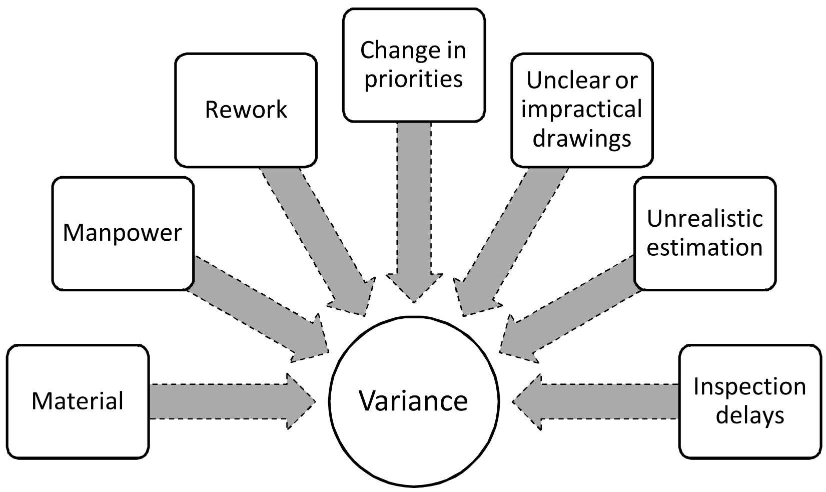


Figure 3-9. Major factors causing cost/schedule variance in the case study.

An updated LAS is a sound foundation for decision making and system analysis due to the fact that it reflects the most recent situation of the ongoing projects. Sample results of the simulation model, based on 30 simulation runs, regarding LAS based on the actual data are shown in Table 3-6. The simulation results in Table 3-6 represent four divisions including 321 steel pieces that originally were planned to be fabricated in the specified time interval (i.e., from January 18 – 27, 2010). Scheduled start and finish dates come from the project baseline defined by a shop manager and two foremen, each with more than 15 years of experience. The calculated start and finish dates are the simulation output, generated based on the current shop conditions and the activity duration distributions forecasted by the intelligent adjuster for the pieces planned to be manufactured within the stipulated period of time. A comparison between the actual finish dates and the estimated finish dates generated by the people involved in scheduling, and what the developed intelligent system generated, results in determining estimation errors. Estimation error analysis reveals that the intelligent system could generate more reliable managerial information and LAS. As an example, for division 52A, human error regarding the finish date was 11 days, while the intelligent system had an error of 4 days. The average absolute estimation errors for the 4 divisions shown in Table 3-6 was 9 days in the case of human judgment, while the intelligent system's estimation had an absolute error of 1.75 days on average. This may be attributable to the fact that the intelligent system generates schedules based on the recent conditions of the dynamic environment of the fabrication shop, perceived influencing factors and their combined interactions, while human beings are seldom able to consider all of these parameters in scheduling. It should be noted that a shortage of actual historical data that could be used for training the intelligent adjuster limited the accuracy of the developed system. Even though the performance of the developed LAS system is quite promising, having more actual historical data can enhance the intelligent adjuster forecasts and subsequently improve the accuracy of the simulation results.

Table 3-6. Sample results of the Simulation federate.

Division ID	No. of pieces	Scheduled Start Date	Scheduled Finish date	Calculated Start date	Calculated Finish Date	Actual Finish Date	Error in Estimating Finish Date (days)	
							Human Estimator	Intelligent LAS system
52A	64	18/01/2010	22/01/2010	18/01/2010	29/01/2010	1/2/2010	11	4
51A	34	19/01/2010	25/01/2010	19/01/2010	29/01/2010	29/01/2010	4	0
5A	103	19/01/2010	27/01/2010	20/01/2010	4/2/2010	5/2/2010	9	1
50A	120	21/01/2010	27/01/2010	25/01/2010	10/2/2010	8/2/2010	12	-2

Conclusion

Look-ahead scheduling of steel fabrication projects that considers projects' constraints as well as the fabrication shop's constraints is very complicated. This chapter implements an intelligent and integrated simulation-based LAS framework for an actual case study in a steel fabrication shop. The system that was developed utilizes RFID technology to capture as-built data. As-built data along with the as-planned data are fed into the system; raw actual data are translated to meaningful data, and an intelligent component generates/forecasts essential scheduling variables based on the actual and historical data for each steel piece, considering several piece-oriented and environmental influencing factors, and updates the simulation model to allow it to produce reliable look-ahead schedules. The proposed system is expanded by employing High Level Architecture (HLA). In this way, the model is split into several components that are linked together in a well-structured format.

Unlike traditional scheduling methods that are static and time-consuming to update, the capability of the proposed system to dynamically and intelligently incorporate the most recent project data and changes in the fabrication shop environment can improve the accuracy of LAS and reduce input modeling burdens on end users.

References

AbouRizk, S., Knowles, P. and Hermann, U. R. (2001). “Estimating labor production rates for industrial construction activities.” Journal of Construction Engineering and Management, 127(6), 502-11.

AbouRizk, S.M. and Hajjar, D. (1998). “A framework for applying simulation in construction.” Canadian Journal of Civil Engineering, 25 (3), 604-617.

AbouRizk, S.M., and Sawhney, A. (1993) “Subjective and interactive duration estimation.” Canadian Journal of Civil Engineering, 20(3), 457–470.

Alvanchi A., Azimi R., Lee S. and AbouRizk S. (2010). “Virtual model of structural steel construction using COSYE Framework.” Proceedings of the 10th International Conference on Construction Applications of Virtual Reality, Sendai, Miyagi, Japan, 283-290.

Azimi, R., Lee, S., and AbouRizk, S.M. (2010 a). “Integrated simulation-based Look-Ahead Scheduling for steel fabrication projects.” Proceedings of the Second International Conference on Advances in System Simulation (SIMUL 2010), Nice, France, 129-133.

Azimi, R., Lee, S., and AbouRizk, S. M. (2010 b). “Resource performance indicators in controlling industrial steel projects.” Proceedings of the ASCE Construction Research Congress, Banff, Canada, 21-30.

Azimi, R., Lee, S., AbouRizk, S. and Alvanchi, A. (2011). “A framework for automated and integrated project monitoring and control system for steel fabrication projects,” Automation in Construction, 20(1), pp. 88-97.

Azimi, R., Lee, S., and AbouRizk, S.M. (2009). “Automated project control system for steel projects.” Proceedings of the Joint International Conference on Construction Engineering and Management and Construction Project Management (ICCEM-ICCPM), Jeju, Korea, 479-486.

Ballard, H.G. (1997). “Look-ahead planning: the missing link in production control,” Technical Report No. 97-3, in Proceedings of the 5th Annual Conference of the International Group for Lean Construction, Griffith University, Gold Coast, Australia, 13–25.

Bishop, C. M. (1995). “Neural networks for pattern recognition.” Oxford University Press, New York, NY.

Brown, J.L. (2009). “Wisconsin bets on BIM.” Civil Engineering, 79(9), 40-42.

Chin, S., Yoon, S., Choi, C., and Cho, C. (2008). “RFID+4D CAD for progress management of structural steel works in high-rise buildings.” Journal of Computing in Civil Engineering, 22 (2), 74-89.

Chung, T.H., Mohamed, Y., and AbouRizk, S.M. (2006). “Bayesian updating application into simulation in the North Edmonton sanitary trunk tunnel project.” Journal of Construction Engineering and Management, 132(8), 882–894.

Daneshgari P. and Moore H. (2009). “The secret to short-interval scheduling,” Electrical Construction and Maintenance, 108(2).

Golenko-Ginzburg, D. and Gonik, A. (1997). “Using look ahead techniques in job-shop scheduling with random operations.” International Journal of Production Economics, 50 (1), 13-22.

Hadavi, A., and Krizek, R.J. (1993). “Short-Term Goal Setting for Construction.” Journal of Construction Engineering and Management, 119 (3), 622-630.

Hinze, J. W. (2008). “Construction planning and scheduling.” 3rd ed., Pearson, NJ.

Lu, M., Abourizk, S. and Hermann, U. H. (2000). “Estimating labor productivity using probability inference neural network.” Journal of Computing in Civil Engineering, 14(4), 241-8.

Lu, M., Dai, F., and Chen, W. (2007). “Real-time decision support for planning concrete plant operations enabled by integrating vehicle tracking technology, simulation, and optimization.” Canadian J. of Civil Eng., 34(8), 912–922.

Ramírez-Valdivia M.T., Christian P.H., Frederick P., Aksoy B., and Zimmers Jr. E.W. (2003). “Development of a software tool to improve performance of packaging operations through short interval scheduling.” Packaging Technology and Science, 16(5), 179-89.

Schmahl, K.E. (1996). “Variation in success of implementation of a decision support/finite scheduling system.” Production and Inventory Management Journal, 37(1), 28-35.

Smith M. (1981). “Manufacturing Controls: How the Manager Can Improve Profitability.” Wiley, Chichester.

Song L., Cooper C., and Lee S. (2009). “Real-time simulation for look-ahead scheduling of heavy construction projects.” Proceedings of the ASCE Construction Research Congress, 1318-1327.

Song, L., AL-Battaineh, H., and AbouRizk, S. M. (2005). “Modeling uncertainty with an integrated construction simulation system.” Canadian Journal of Civil Engineering, Vol. 32(3), 533-542.

Sonmez R., and Rowings J.E. (1998). "Construction labor productivity modeling with neural networks." Journal of Construction Engineering and Management, 124(6), 498-504.

Swingler, K. (1996). "Applying neural networks: a practical guide." Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Neuroshell2. (2010). <www.wardsystems.com/manuals/neuroshell2> (Accessed on 12/20/2010).

Zweben M. and Fox M. (1994). "Intelligent Scheduling." Morgan Kaufmann: Bethlehem, PA.

Chapter 4. Revisiting project control using basic control theory principles [4]

4.1. Introduction

The increasing use of manufacturing technologies has improved productivity in the construction industry (Eastman and Sacks 2008). In many projects, certain elements can be fabricated either on site or off site in manufacturing shops. For example, concrete walls can be cast in place utilizing formwork or they can be precast in a shop, cured, and shipped to the construction site. Off-site construction has some advantages, because activities occur in a controlled environment. Factors such as weather conditions do not have a significant impact on projects being executed within an enclosed fabrication shop, the production rate increases, and a greater degree of supervision and control of safety and quality of the manufactured elements is in place. Despite the proven advantages of off-site construction over on-site construction projects, the extent of its usage is still restricted (Lu 2009). Thomas and Sanvido (2000) noted several issues regarding off-site construction fabrication, including late and out-of-sequence deliveries, double-handling of components, fabrication errors, and incompatibility between fabrication rates and site requirements. Taking into account these issues, along

---

[4] A version of this chapter has been submitted to the journal of Computing in Civil Engineering, ASCE.

with other characteristics of off-site construction environments such as the fast paced nature, the existing resource limitations within the shop, and the constraints established by clients and construction sites, it is clear that controlling off-site construction projects can be a difficult task for project managers.

To be competitive in the construction industry, monitoring and control systems in off-site construction projects should lead to timely decision making in case of variances from desired plan. This results in taking effective corrective measures, which can bring about remarkable cost and time savings (Dissanayake and Fayek 2008). The emergence of new technologies such as the Global Positioning System (GPS) and Radio Frequency Identification (RFID) has influenced and improved the monitoring of construction projects by providing high-quality actual data in real-time (Azimi et al. 2011a). Once actual data are captured, project performance must be analyzed by comparing actual and as-planned data, which leads to detecting variances and deviations. If any deviation is detected and it is not within the predefined threshold levels, corrective action should be taken (Figure 4-1). For this purpose, different scenarios are examined, construction performance is forecasted for each scenario, and the best corrective action is selected, based on the forecasted results. Efforts have been carried out to improve different aspects of project monitoring and control, such as real-time data capture (Chin et al. 2008, Navon and Sacks 2007), data analysis (Kim et al. 2008) and productivity modeling (Song and AbouRizk 2008). However, Dissanayake and Fayek (2008) argue that current construction project control practices primarily deal with performance monitoring.

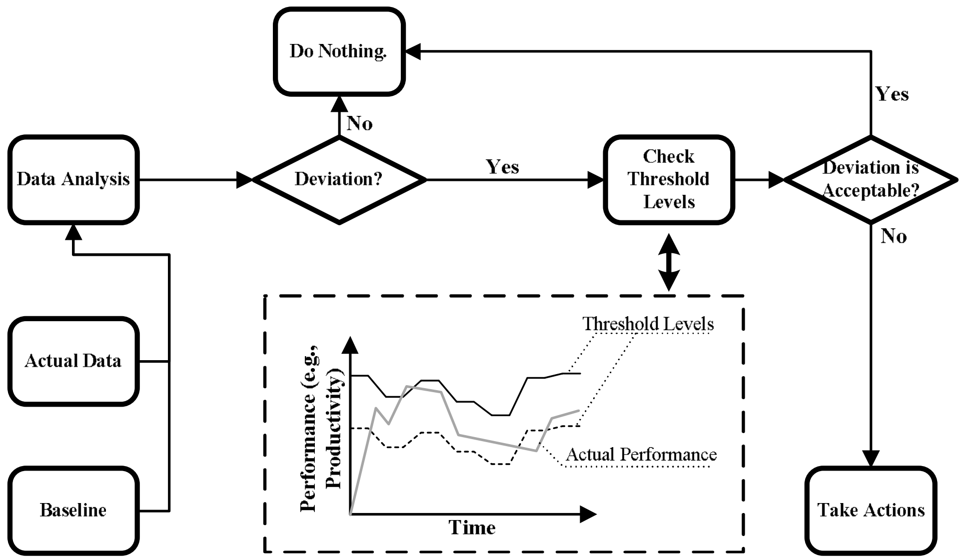


Figure 4-1. Overview of the project control process.

Taking timely corrective actions is a vital element of a robust project control system. A variety of methods, including factor models (Thomas et al. 1990, Thomas and Sakarcan 1994), linear regression (Sanders and Thomas 1993; Smith 1999), fuzzy logic (Fayek and Oduba 2005, Dissanayake and Fayek 2008), artificial neural networks (Karshenas and Feng 1992, Chao and Skibniewski 1994, Maier and Dandy 2000, Lu et al. 2001) and simulation (Alvanchi et al. 2011, Azimi et al. 2011a) have been used for performance forecasting purposes that can be used for decision making. Two drawbacks of these methods can be categorized as: 1) requiring availability of historical data and/or expert judgment, and 2) slowness. In many cases, quantitative and qualitative historical data such as field measurements, progress reports, and expert judgment are not available or are limited and cannot be used for accurate decision making. Additionally, slowness in the project control process results in late decisions based on outdated data, which is not very effective, especially in fast paced environments such as off-site construction shops.

To address these challenges, this chapter presents an approach for enhancing project control by enabling timely decision making in the unique and dynamic environment of off-site fabrication shops. The foundation of the proposed approach has been established based on “Control Theory” concepts which have been proven to be effective in real-time handling of deviations in dynamic systems by using real-time actual system performance data. The chapter consists of the following major sections: (1) an overview of off-site construction and related project control challenges, (2) an introduction to control theory and its capabilities, (3) the introduction of the proposed approach for enhancing project control and (4) a case study to investigate the feasibility and practicality of the proposed approach, which was implemented in an actual case of one of the complicated off-site construction manufacturing systems i.e., structural steel fabrication.

4.2. Control theory

The ultimate goal of “Control Theory” is controlling systems, making those systems accomplish particular objectives or making them act in a desired manner

(Leigh 2004). Off-site construction shops usually involve repetitive operations and project managers endeavor to maintain shop performance at a desired level, which makes control theory as a potential option for achieving such a target. In control theory the system or environment under consideration is called a plant (Ogata 2002), and a controller is a mechanism that settles actions to implement in the plant to accomplish a certain objective (McKay 2010). Control theory involves determining input variables of a plant that will cause the plant outputs to be as close as possible to a desired value, called a reference value. Figure 4-2 depicts a general overview of a plant to be controlled using the control theory concept. System outputs (y_i) are dependent variables which must be close enough to the reference values (r_i). The inputs of the system (u_i) are usually independent variables that govern the behavior of the system. Disturbances (d_i) refer to unknown or unpredictable inputs that influence the system. The controller is responsible for assessing the current state of the plant (i.e., comparing y_i with r_i) before determining its next action – referred to as feedback in control theory. This assessment leads to executing actions that will best meet the desired objectives.

The simplicity and application-independency of its concepts make this theory a favorable candidate for dealing with controlling dynamic systems.

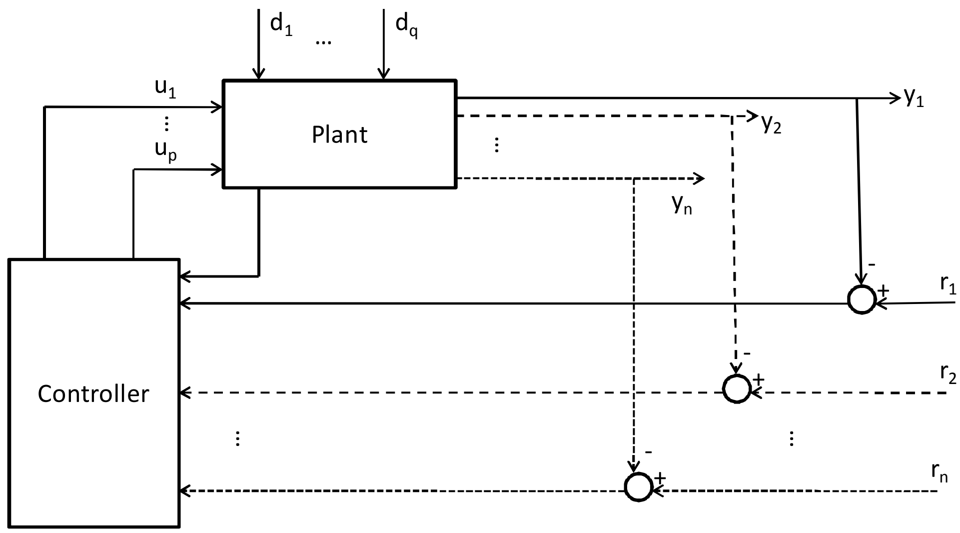


Figure 4-2. A general block diagram of a system to be controlled.

However, successful application of control theory in a system has several major requirements. Firstly, an objective regarding the future state of the system should be defined. This objective is related to the system performance over a certain period of time. Generally speaking, the objective can be the desired performance of the system. Secondly, a set of possible scenarios regarding corrective actions needs to be existed. Thirdly, some methods for selecting the corrective actions that will bring about the desired system performance need to be developed. This requires a model capable of forecasting the effect of possible control actions on the system performance not only adequately and reliably but also comparatively quickly.

In spite of broad utilization in different domains such as manufacturing (Deif 2007, Ma 2005,Wiendahl and Breithaupt 2000), control theory concepts have not been widely used or implemented for controlling construction projects due to complexity of these projects. In manufacturing systems, block diagrams and dynamic equations in continuous or discrete time are used for creating the Production Planning and Control system models based on control theory to meet desired production goals (Ma 2005). In this approach, the models usually do not contain processes with stochastic uncertainties and these uncertainties are treated as disturbances in control theory (Ma 2005). A dynamic equation of the observed system, commonly in mathematical format, is used for control purposes.

While advances in data acquisition technologies have made it quite possible to have prompt performance information in construction projects, methods still need to be proposed regarding making use of these data for timely project control. This chapter investigates the feasibility of using control theory concepts in enhancing project control in off-site construction, which suffers from a high level of nonlinearity and a lack of analytical models regarding input/output relationships.

4.3. Proposed approach for enhancing project control

In the course of execution of a construction project, project control involves several activities such as comparing actual data with the plan and determining deviations. Threshold levels need to be already available and used for determining if deviations are acceptable or not. Corrective actions are initiated by deviations and if any deviation is not within the margins defined with threshold levels a corrective action needs to be taken.

Taking corrective actions can be based on reasoning, i.e., determining the reasons behind the poor project performance. Another approach can be similar to controlling mechanical or electrical systems using control theory. In this approach the ultimate goal is maintaining a desired performance and the cause of deviation is not dealt with. For example, a cruise control system in a car is responsible for maintaining a speed close to the desired speed set by the driver, regardless of whether the car goes up and down hills, or whether other disturbances are occurring, such as wind that influences the car's speed.

In construction projects, it is crucial to maintain a certain project performance value. For instance, in an earth moving project for constructing a road, the desired project performance may be moving 40 cubic meters of soil per hour. If the current performance is not close to the desired value, corrective actions need to be taken. In many cases, a high priority is given to maintaining the desired performance, rather than finding the cause of deviation. The reasons behind such a priority can be avoiding an incoming bad weather season, reducing overhead costs, maintaining company's reputation in the market, avoiding high rate penalties, gaining incentives and so on. This chapter focuses on the same approach and suggests a method in terms of finding the best corrective action that leads to the desired project or system performance. The idea behind the proposed method was inspired by two topics in control theory, Nonlinear Model Predictive Control (NMPC) (Findeisen et al. 2007) and System identification (Verhaegen and Verdult 2007). In the method presented here, the relationship between the input variables and output variables of the off-site construction shop under control are studied and a mathematical equation is constructed based on the available input/output (I/O) data sets representing the relationship between the performance (i.e., output) of the fabrication shop and the influencing factors (i.e., input variables). Once the equation is generated, a cost function for the influencing factors, along with other constraints, is established. The best corrective action is then determined by minimizing the cost function given all the constraints including the generated performance equation greater (or less) than a desired value. An overall view of the proposed method is depicted in Figure 4-3 and major steps, indicated in grey, are briefly discussed below.

4.3.1. System Identification

The first step of the proposed method is called system identification. In this phase, the off-site construction environment is carefully observed and the system output change with regards to the changes in influencing factors (i.e., inputs) is measured. If an appropriate amount of historical data regarding the effects of different scenarios (i.e., input change) on output change is available, no more observation is required and this step can be skipped.

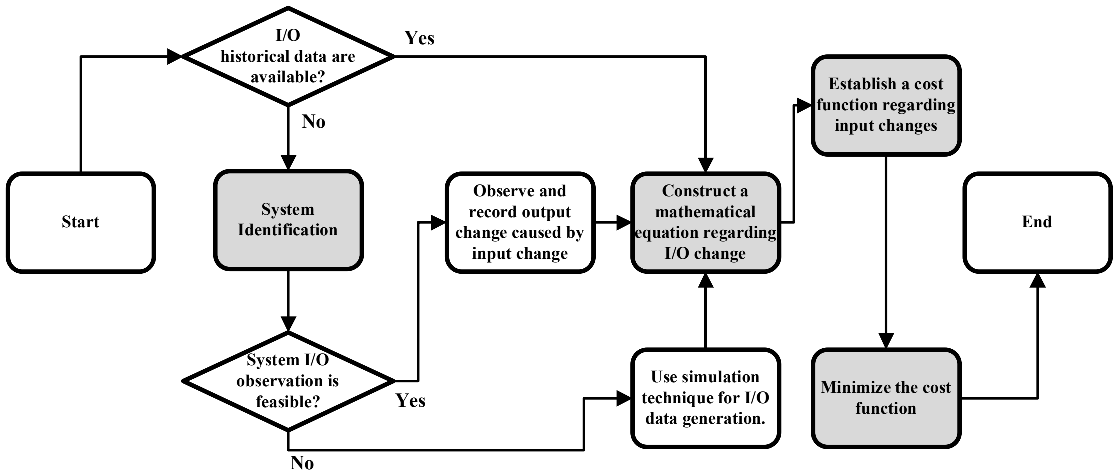


Figure 4-3. Major steps need to be taken for determining the best action in the proposed method.

However, in most cases, available historical data are limited and data sets that represent the change in project performance in relation to input change are not broadly available. Additionally, changing input variables and measuring the corresponding change in output variable(s) of the off-site construction shop is a very time consuming process and sometimes is not feasible. For instance, the number of skilled workers is one influencing factor in many off-site construction shops, but hiring and firing workers to observe the change in shop output is not very feasible. To overcome these challenges and considering the fact that off-site construction commonly involves repetitive operations that occur during the course of construction, we proposed the computer simulation technique for system identification purposes. A simulation model can be created and available historical data or other techniques can be used for model validation. Once validated, the simulation model is assumed to acceptably resemble the real-life off-site construction system. It can be used for capturing how the system output (e.g., production rate or productivity) will change regarding changes in its input variables, and for generating I/O data sets.

4.3.2. Constructing a mathematical equation

Once the data sets regarding output change with regards to input change are available (i.e., from historical data or simulation model), they can be used for generating a mathematical equation that represents the system under control. This mathematical equation leads to speeding up the decision making process.

In construction projects such a mathematical equation is usually complex and nonlinear, and generating this equation can be a remarkable challenge. Group Method of Data Handling (GMDH) algorithm has been used in this research to generate such an equation due to its capacity for identifying I/O relations in large complex systems (Onwubolu 2009) such as off-site fabrication shops. Two major advantages of using the GMDH algorithm in this research are: first, identifying the important input variables (influencing factors) and second, generating the mathematical equation representing the input/output relationship for the off-site fabrication shop.

The GMDH algorithm, also known as polynomial neural networks (Ivakhnenko 1971), generates a mathematical model of optimal intricacy based on Input/Output relationships within the system after filtering out insignificant variables. The backbone of GMDH is established upon two basic concepts in neural network modeling (Chang and Hung 2007):

- The system analysis approach is based on the black-box method and system's input-output data sets;
- Complex functions are represented by connectionism through networks of elementary functions (Muller 1998).

The power of GMDH compared to most other Artificial Neural Network (ANN) approaches is its capacity to select an optimal structure of the network regarding layers and nodes (Maier and Dandy 2000) and generate a symbolic description of the model with a polynomial function of the relevant input variables.

This algorithm uses a heuristic self-organizing learning process for constructing the mathematical model (Onwubolu 2009). The GMDH algorithm constructs a high-order polynomial which relates M input variables (i.e., x_1, x_2,…, x_M) to a single output variable (i.e., y) of the form (Ivakhnenko 1971):

$$y = a_0 + \sum_{i=1}^{M} a_i x_i + \sum_{i=1}^{M}\sum_{j=1}^{M} a_{ij} x_i x_j + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} a_{ijk} x_i x_j x_k + \cdots \qquad Eq.(1)$$

where a_0, a_i, a_{ij}, a_{ijk}, … are called summand coefficients. All combinations of input variables are tried with this method in order to construct an optimum polynomial description of a system. With the GMDH algorithm, each neuron in the network implements a nonlinear function of its input from two other neurons.

These two input variables, which we will call x_i and x_j, are used for generating a polynomial term (descriptor) in the form of a simple quadratic transfer function similar to Eq. (2) (Ivakhnenko 1971):

$$y = A_2(x_i, x_j) = b_0 + b_1 x_i + b_2 x_j + b_3 x_i^2 + b_4 x_j^2 + b_5 x_i x_j \qquad Eq.(2)$$

where the subscript in A_2 denotes a second order transformation of the inputs, and $\{b_0, b_1, \ldots, b_5\}$ are statistically determined for each transfer function. In the case of n input variables, there are $\frac{n(n-1)}{2}$ neurons in the first layer of the network, because this layer covers all the transfer functions of each pair of input variables. These transfer functions become input candidates for the second layer, and so on. In other words, each transfer function in a layer becomes an input candidate for the upper layer to be used for building new transfer functions, similar to the aforementioned procedure for generating transfer functions in the first layer. Only the candidates (i.e., transfer functions) which give the best approximation of the output variable are allowed to pass to the next layer. Thus, while the first layer is made up of a set of second order polynomials of the input variables, the second layer consists of the fourth order polynomials, the third layer involves eighth degree polynomials, and so on. The output (i.e., transfer function) of the last layer (consisting of only one node) is the output of the whole net. With GMDH, the network structure is self-organizing and neurons at each layer are selected based on performance criteria to limit the size of the network. The parameters of each transfer function are determined during the learning phase, which leads to building up the whole structure automatically [for more details see Ivakhnenko (1971)].

GMDH has many advantages compared to the regression method. While choosing the correct set of polynomial terms and determining the level of complexity of the terms (i.e., the largest degree of the polynomial) are problematic issues when using the regression method, GMDH addresses these issues by trying all possible combinations.

4.3.3. Establishing a cost function for the under controlled system

Once the mathematical equation is generated, a cost function for the important influencing factors, along with other constraints, is established. In the case of N important influencing factors, if the change in the i^{th} influencing factor is represented by U_i , the cost function will be in the form of:

$$f(U_1, U_2, \ldots, U_N) = \sum_{i=1}^{N} c_i \times U_i \qquad \textit{Eq. (3)}$$

where c_i is the unit cost associated with changing the i^{th} influencing factor. Constraints enforced by the environment, policies, clients, etc., are also defined in this step.

4.3.4. Minimizing the cost function

Within this research, the best corrective action has been defined as the action which causes the system performance to meet or exceed a desired value with minimum cost. Thus, determining the best corrective action becomes minimizing the cost function subject to all the constraints including the performance equation greater (or less) than a desired value. This step leads to determining the optimum corrective action (i.e., change in influencing factors), for maintaining desired performance (system output). It should be noted that the desired performance can be related to cost, time, quality or a combination of them. To provide the readers with a better overview of the proposed method, a case study where this control method has been utilized is described in detail within the next section.

4.4. Case Study

To verify the practicality of the proposed method, a case study of an off-site structural steel fabrication shop was performed. Structural steel fabrication shops play an important role in off-site construction and hundreds of thousands of structural steel elements are manufactured in steel fabrication shops every year to be used in steel construction projects. Monitoring and controlling structural steel fabrication projects is a difficult task for project managers due to the unique characteristics of these projects. Manufacturing a huge number of steel elements with different dimensions, complexity and processing requirements requires a variety of equipment, skilled work, resources, activities and interactions which accent the intricacy of project monitoring and control in steel fabrication shops (Azimi et al. 2011a).

Unlike other mass production industries where the final products are a single type or a very limited number of types, structural steel fabrication deals with producing a variety of different steel pieces. Steel fabrication shop performance (e.g., shop production rate) fluctuates substantially with several influencing factors, such as complexity of steel pieces planned to be fabricated, their weights, workers' level of skill and so on. Taking timely corrective actions in the fast paced environment of the steel fabrication shop in the case of deviations is one of the major difficulties project managers encounter, and this case study intends to address that issue by utilizing the proposed approach.

4.4.1. System Identification using simulation modeling

The output (or performance) of the structural steel fabrication shop that should be observed and maintained to the desired value is production rate, represented by tonnage of steel elements fabricated per day. In the structural steel fabrication shop, hereafter referred to as the fabrication shop, the production rate can significantly change with the complexity of the steel pieces which are manufactured, provided that other influencing factors do not change. In other words, the more complex the steel pieces, the lower the production rate. This can be a challenge for project managers regarding taking corrective measures, because low production rate can be justified due to high complexity of the ongoing projects. To address this issue, we defined the production rate as:

$$\frac{\sum c_i \times w_i}{Day} \qquad \textit{Eq. (4)}$$

where c_i is the piece complexity (Azimi et al. 2011b), and w_i is the weight of the fabricated steel piece. Here, based on the number of steel components forming the steel piece, a number between 1 and 5 is assigned to the steel piece and referred to as piece complexity. For example, if the steel piece is composed of up to 4 components it is considered a "not complicated" steel piece and its complexity is 1. Project complexity can be defined as:

$$\frac{\sum c_i \times w_i}{\sum w_i} \qquad \textit{Eq. (5)}$$

where c_i is the piece complexity, and w_i is the weight of each steel piece in the project (for details see Azimi et al. 2011b). This definition of production rate can be a reliable index for initiating corrective action. For instance, let us assume that currently the average shop production rate for the ongoing steel project with a complexity of 3 is 70 tons/day, while last month the average production rate for the fabricated steel project with a complexity of 2 was 107 tons/day. Although the traditional definition of production rate shows a large fluctuation between 70 tons/day and 107 tons/day, our modified definition shows that the production fluctuation is fairly minor (i.e., 3×70=210 tons/day vs. 2×107=214 tons/day). Thus, from this point forward, production rate or system output refers to our modified definition of production rate, which takes into account the concept of complexity.

Influencing factors that affect the performance of the steel fabrication shop were determined through literature review and interviews with shop managers, foremen and estimators. In this case study, even though large amounts of historical data concerning daily production were provided, limited data regarding shop output changes with regard to changes in its influencing factors were available. On the other hand, changing all the important input variables, such as hiring and firing fitters in the shop in order to monitor output change, was not an option in the fast paced environment of the steel fabrication shop, due to reluctance of the management and other restrictions enforced by employee unions, company policies, and so on. This motivated us to make use of simulation techniques for system identification.

One of the great benefits of using simulation in the fabrication shop environment is that it lets project managers get a system-wide view of the effects of input changes on the shop throughput. Predicting the change in performance of a certain station with its input change may be doable; however, it may be difficult to determine the impact of this change on the performance of the whole shop without the help of simulation.

Model development began with several meetings with experts regarding defining objectives, assumptions, and data requirements, and reviewing initial simulation results and modifying model assumptions. Software called *AnyLogic®* was utilized for developing the fabrication shop simulation model. Figure 4-4 depicts snapshots of the simulation model which was developed. The native Java environment of this software provides favorable extensibility regarding custom Java code, external libraries, and external data sources. The collaborative company supplied a large amount of data that already existed in its database server; however, because most of the data had been entered manually over the years, effort was required for filtration and ensuring data validity regarding weight of the pieces, assigned man-hours and so on. This was necessary, because data play an important role in developing the simulation model and building the conceptual model and model validation.

4.4.2. Model Verification and Validation:

To perform system identification using a simulation model that represents a complex real-world fabrication shop with acceptable accuracy, one should ensure that the simulation model and its results are realistic. This is usually addressed through model verification and validation. The purpose of model verification is to ensure that the model has been programmed properly, the algorithms used are carried out correctly and there is no error or bug in the simulation model.

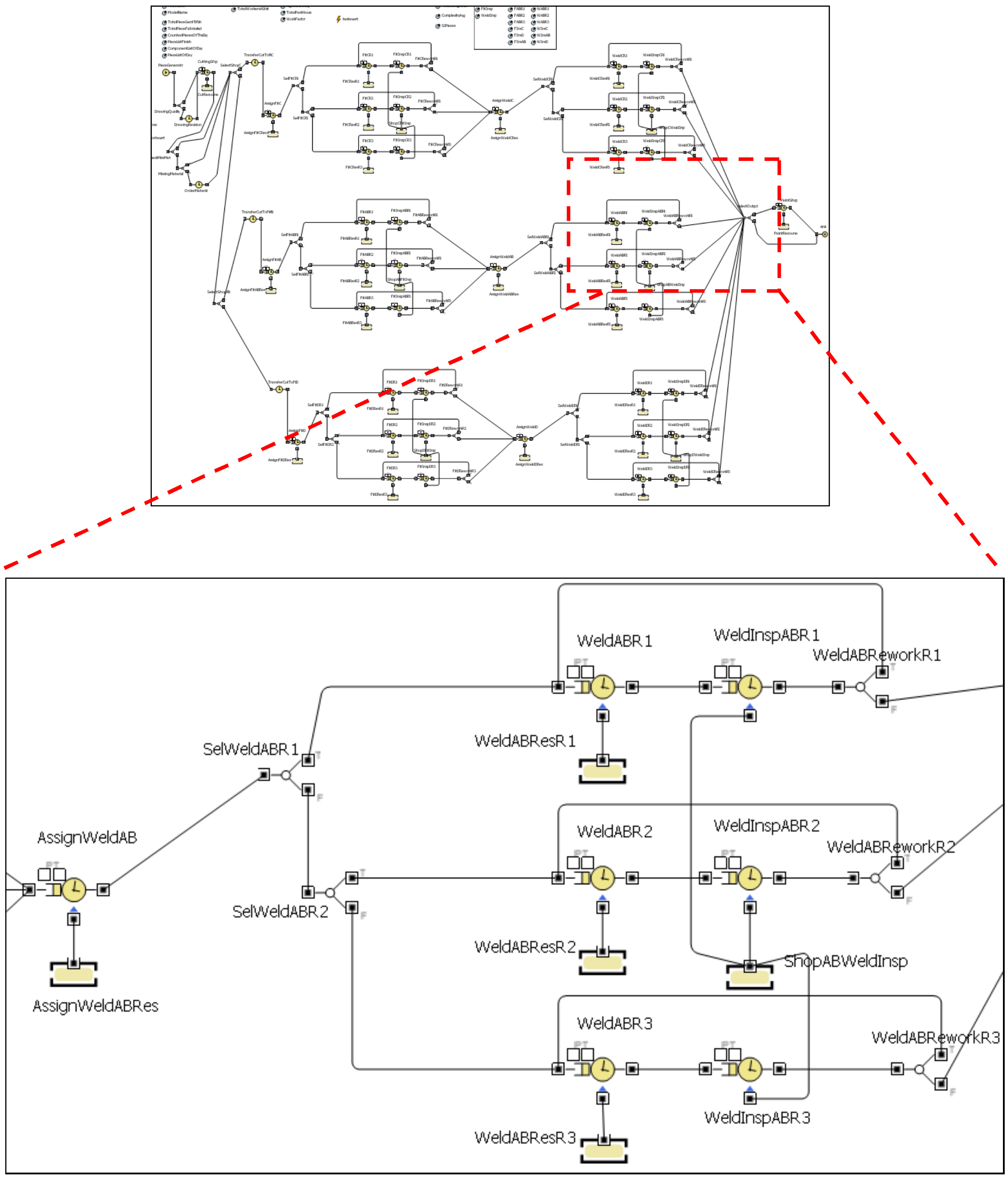


Figure 4-4. Simulation model developed in Anylogic®.

Model validation guarantees that the methods employed and the results obtained meet the intended requirements and accuracy. The desired accuracy needs to be specified by the end users, who are usually project managers and decision makers.

Different techniques were used for model verification. In the course of developing the model, the whole model was divided into several submodels. The submodels were coded and debugged one by one to form a moderately detailed model. This model was then gradually made as complex as needed to adequately represent the fabrication shop under study. Tracing was also conducted to check the state of the simulated system with regards to different items such as the number of fitted and welded pieces, number of pieces sent to the shop, number of pieces fabricated etc., after each certain event occurred, and compare them with available data.

Model validation began with face validation in which the model and its results were presented to the managers and decision makers; their feedback showed the credibility of the model and that the results were accepted as correct. In another effort to determine model validity, actual data from the fabrication shop system and corresponding simulation results were compared. Because the output of the steel fabrication shop and the simulation model are nonstationary (i.e., the distributions of the successive observations change over time), classical statistical tests such as the Mann-Whitney, two-sample Kolmogorov-Smirnov, etc., are not directly applicable (Law and Kelton 2000).

Since a large amount of data from the fabrication shop system and the simulation model was available, the confidence-interval approach (Law and Kelton 2000) was used for comparing the model and the actual system. The simulation model considered the steel pieces planned to be processed in the fabrication shop between January 2008 and March 2008. Let

X=average production rate (tons/day) in steel fabrication shop,

and

Y=average production rate (tons/day) for simulation model.

Let also Y_j be the average of the simulation results in the jth experiment with mean $\mu_y=E(Y_j)$. Since actual historical data were available, X_j, which represents the average of the production rate in the steel fabrication shop for the aforementioned interval of time (i.e., Jan-Mar 2008) in each experiment, is a constant number with mean $\mu_X= X_j=150$ tons/day. Due to confidentiality of data provided by the collaborative company, data presented in this chapter (i.e., production rates and weights) are normalized data. Table 4-1 gives sample results from the 30 simulation runs used for checking model validation. To determine whether the model is an accurate representation of the fabrication shop, a 90 percent confidence interval for $\zeta= \mu_X - \mu_y$ using the paired-t approach was constructed.

Table 4-1. Sample Results of the fabrication shop simulation model.

Experiment *j*	1	2	3	4	5	6	…	27	28	29	30
$\mathbf{Y}_j$	149.90	147.62	152.43	149.65	148.27	152.79	…	152.95	149.66	149.18	151.92
$W_j = X_j - Y_j$	0.10	2.38	-2.43	0.35	1.73	-2.79	…	-2.95	0.34	0.82	-1.92
$[W_j - \overline{W}(30)]^2$	0.33	8.14	3.86	0.67	4.83	5.40	…	6.17	0.66	1.65	2.10

Using Table 4-1 we have:

$$\overline{W}(30) = \overline{X}(30) - \overline{Y}(30) = -0.47$$

$$V\hat{a}r[\overline{W}(30)] = \frac{\sum_{j=1}^{30}[W_j - \overline{W}(30)]^2}{30 \times 29} = 0.137$$

And for the 90 percent confidence interval for ζ we have:

$$\overline{W}(30) \pm t_{29,0.95} \times \sqrt{V\hat{a}r[\overline{W}(30)]} = -0.47 \pm 0.485 = [-0.955, 0.015]$$

Since the interval contains 0, the observed difference is not statistically significant and the simulation model is valid for representing the average production rate of the steel fabrication shop over a three month work interval. In this case study, the control cycle duration is three months. Sample results from the simulation model vs. actual data are shown in Figure 4-5.

As mentioned, the simulation model is a powerful tool to overcome the infeasibility of changing certain input parameters in the shop; it can be used for trying different scenarios and generating reliable input/output data sets to be used for project control purposes. 65 different scenarios were tried using the simulation model and I/O data sets were generated accordingly. Each data set is the result of 10 simulation runs.

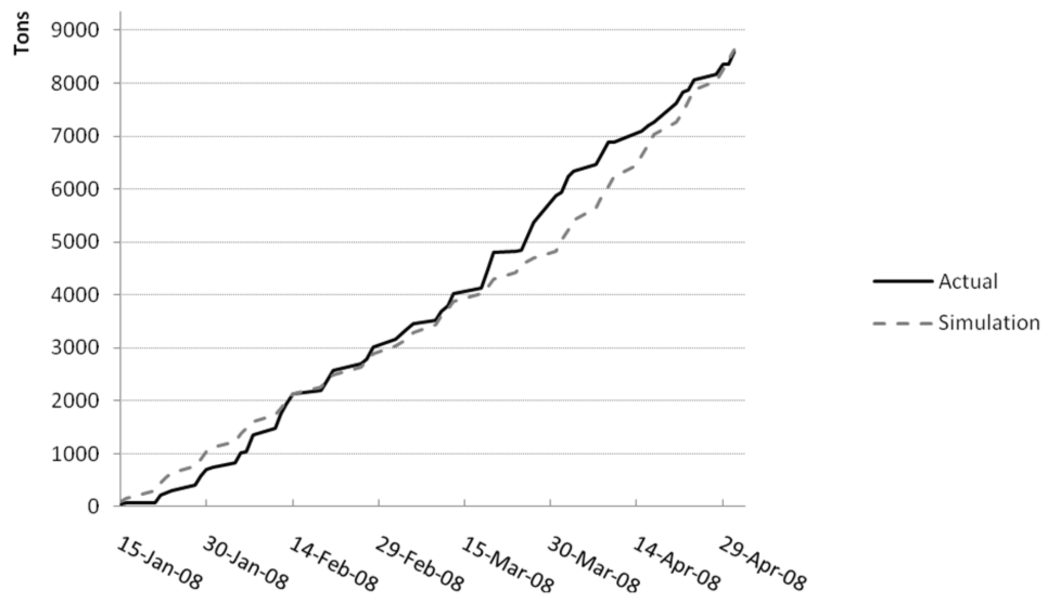


Figure 4-5. Accumulated weight of steel pieces fabricated in the shop vs. simulation results.

The simulation model has 11 input variables (e.g., fitters, welders, painters, material availability, etc.) and the simulation output is the average of production rate. To generate data sets, first the simulation model was run for the base scenario. Input variables and their initial values for the base scenario are shown in Table 4-2. In this table, for the laborers, (i.e., columns 1 to 9) the numbers represent the number of workers in each shift extracted from the collaborating company's database. The rank of workers is an indicator regarding their skill and experience. Regarding material availability and drawing quality, three states, Poor (-1), Fair (0), and Good (+1), were defined based on historical data and expert judgment. For example, a Fair quality of drawing means that X % of the drawings need to be revised, where X=Uniform distribution (0.5, 1).

Table 4-2. Values of inputs variables for the base scenario.

Fitter Rank1	Fitter Rank2	Fitter Rank3	Welder Rank1	Welder Rank2	Welder Rank3	Fit Insp.	Weld Insp.	Painter	Material Availability	Drawing Quality	Shift
18	13	4	11	8	5	3	3	2	0	0	Day
1	24	2	6	10	5	3	3	1	0	0	Night

Then, the base scenario becomes the foundation for trying other scenarios. In other words, using the simulation model, output change is observed with regards to input change compared to the base scenario. A sample of these scenarios is shown in Table 4-3. In this table, the first row represents the base scenario. Other rows show how changes in input variables can affect system output (i.e., production rate). For instance, the second row shows that by adding 1 more fitter rank1 and 1 more welder rank1 to the number of workers defined in base scenario, the average production rate (i.e., column 12) will increase by about 2.5% (i.e., column 13).

4.4.3. Generating Mathematical equation

Scenarios regarding input/output relationships in a fabrication shop, generated by the developed simulation model (shown in Table 4-3) are used as input and output in GMDH. In other words, each row in Table 4-3, represents input variables (i.e., columns 1-11) and an output variable (i.e., column 13), used in GMDH for developing the mathematical model.

Table 4-3. Output change with regards to Input change compared to base scenario.

Fitter Rank1 (1)	Fitter Rank2 (2)	Fitter Rank3 (3)	Welder Rank1 (4)	Welder Rank2 (5)	Welder Rank3 (6)	Fit Insp (7)	Weld Insp (8)	Painter (9)	Material Availability (10)	Drawing Quality (11)	Avg. Output* Ton /day (12)	ΔY (output change) (13)
0	0	0	0	0	0	0	0	0	0	0	150.00	0
1	0	0	1	0	0	0	0	0	0	0	153.77	0.025115
1	0	0	1	0	0	1	1	0	0	0	162.93	0.086174
1	0	0	0	1	0	1	1	0	0	0	162.22	0.08149
1	1	1	1	1	0	1	1	2	0	0	167.60	0.117353
1	1	2	2	1	0	1	1	0	-1	1	175.24	0.168299
1	1	2	2	1	0	1	0	0	-1	1	164.91	0.099376
1	1	2	2	1	1	3	3	0	-1	1	185.27	0.235163
1	1	2	2	1	1	3	3	0	-1	0	186.29	0.241903
1	1	2	0	1	1	3	3	0	-1	0	174.15	0.160975
1	1	2	2	1	0	4	4	0	0	0	183.67	0.224443
1	1	2	2	1	0	4	4	0	1	0	184.54	0.230242
4	4	1	4	4	1	3	3	0	1	0	201.39	0.3426
5	5	1	5	5	1	3	3	0	1	0	207.42	0.38282
6	6	6	6	6	6	3	3	0	1	0	226.88	0.512534
7	7	6	7	7	6	4	4	0	1	0	230.91	0.539395

*Data have been normalized due to confidentiality.

Software called "*Neuroshell 2*" equipped with the GMDH algorithm was utilized for generating the polynomial mathematical equation, depicting how the output (i.e., production rate) changes with regard to input changes in the observed fabrication shop. The generated mathematical equation is below:

$$0.00342 + 0.05418x - 0.00882x^2 + 0.036y + 0.03375z + 0.0315q = \Delta Output$$

Eq. (6)

Where:

x: changes to number of ranked1welders.

y: changes to number of ranked 2welders.

z: changes to number of ranked 3welders.

q: changes to number of weld inspectors.

According to the training set statistics, Mean Squared Error (MSE), R squared and the Correlation Coefficient for the generated equation are 0.001, 0.988 and 0.993 respectively, which show that the performance of the mathematical model in terms of reflecting the causal relationships between input-output is quite acceptable compared with the simulation results. Figure 4-6 depicts a comparison between the simulation model and the generated mathematical model in terms of calculating output change for 50 different scenarios.

The equation constructed (i.e., Eq 6) is a quadratic equation which indicates the nonlinearity of the observed shop. Another feature of this equation generated based on GMDH algorithm is determining the important input variables, such as the number of welders ranked 1, 2, and 3 and the number of weld inspectors out of the 11 variables considered (shown in Table 4-3), are the most influential variables.

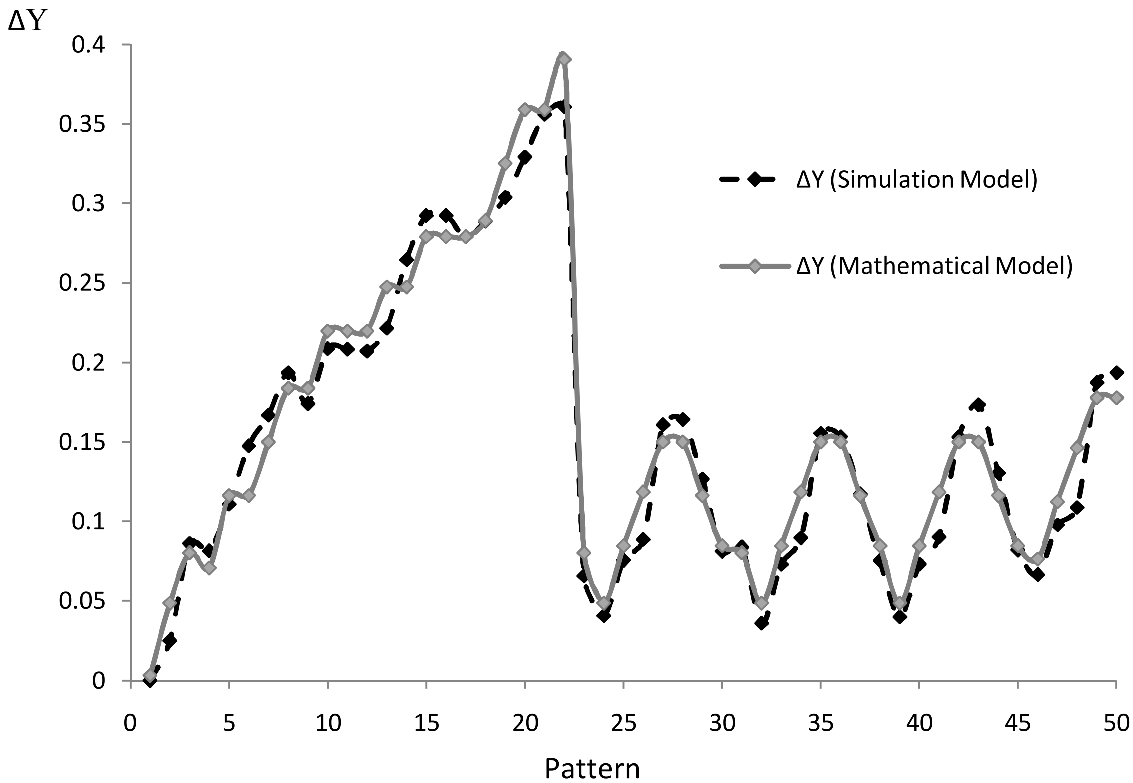


Figure 4-6. Output change (ΔY) calculated by the mathematical model vs. simulation results.

This can be useful for interpreting the current situation of the system. For example, the equation is not sensitive to the number of fitters, which means that there is an excessive number of fitters in the shop. This can be useful for project managers in terms of taking secondary actions such as reducing the number of fitters.

4.4.4. Establishing and minimizing the cost function

The next step for determining the best action is establishing a cost function for the selected important influencing factors and minimizing this function in order to realize the corrective action with the minimum cost. It should be noted that in this research, the "best" corrective action has been defined as the action which causes the system performance to meet or exceed a desired value with minimum cost. The cost function for the observed system is in the form of:

$$f(x,y,z,q)=c_x \times x+c_y \times y+c_z \times z+c_q \times q \qquad \textit{Eq. (7)}$$

where:

C_x: cost of hiring one welder (rank1),

C_y: cost of hiring one welder (rank2),

C_z: cost of hiring one welder (rank3),

C_q: cost of hiring one weld inspector .

The cost function should be minimized considering all the constraints for determining the best action. To clarify how this works, let us assume that the company is running late and the ongoing projects are behind schedule. The analysis shows that to meet the deadlines for in-progress projects, and because of upcoming new projects, the company needs to increase its production rate by at least 15%. To find the best action they can take, first the cost function is established as following (costs of hiring have been normalized here, for confidentiality):

$$f(x,y,z,q) = 100 \times x + 80 \times y + 70 \times z + 110 \times q$$

Then the constraints are defined. For example, the company policy indicates that at least one welder with rank 1 should be available among the new employees. Thus, the constraints are:

$$0.00342 + 0.05418x - 0.00882x^2 + 0.036y + 0.03375z + 0.0315q \geq 0.15$$

$$x \geq 1$$

$$y \geq 0$$

$$z \geq 0$$

$$q \geq 0$$

Since the cost function is a linear equation, and the variables are restricted to be integers, minimizing the cost function can be considered as an integer programming problem. By minimizing the cost function subject to the aforementioned constraints, we have: x=1, y=0, z=3 and q=0. Thus, the best action would be adding 1 more welder ranked 1 and 3 more welders with rank 3 to the fabrication shop. In this case, using the mathematical equation which was generated, the production rate will increase by 15% compared to the average production rate calculated for the base scenario (i.e., 150 tons/day). To double-check the accuracy of the mathematical model, the best action determined by the mathematical model was considered as a scenario in the developed simulation model. According to the third row in Table 4-4, the average system output calculated by the simulation model is 174.6 tons/day with a standard deviation of 0.59 tons. Therefore, the results of ten simulation runs for this scenario show a 16.4% increase in average production rate compared to the base scenario (i.e., 174.6 tons/day vs. 150 tons/day). This indicates that the results of the mathematical model are acceptably close to the results of the validated simulation model, yet a great deal of time can be saved using the generated mathematical equation for decision making, because it is fairly quick compared to using the simulation model, enabling decision makers to make prompt control decisions.

Table 4-4. Simulation results for the corrective action determined by the mathematical model.

Item	Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8	Run9	Run10
Number of pieces simulated	29698	29895	29650	29705	29563	29230	29853	29535	29599	29609
Avg. piece Complexity	1.88	1.89	1.88	1.88	1.89	1.89	1.89	1.88	1.89	1.89
Avg. Output (Ton /day)	175.00	174.67	173.61	173.99	175.48	174.40	175.32	173.95	175.06	174.53

Conclusion

This chapter presents a method aimed at enhancing the project control process by allowing timely corrective actions in the dynamic environment of off-site construction projects, inspired by the concepts of control theory. The beauty and novelty of the proposed method lies in the new project control vision behind it, that is to say, in the case of performance deviation in the fast-paced environment of off-site construction, instead of reasoning, which can be a time consuming process, the cause of deviation is not dealt with and the ultimate goal is maintaining a desired performance in a timely manner with minimum cost. The Group Method of Data Handling (GMDH) was utilized for simple but effective and practical generation of a mathematical equation reliably representing the system under control. A method was proposed for the use of this equation in quick determination of the best action in the case of project deviation. This not only helps off-site fabricators deliver projects on time, reduce over-head costs, maintain their reputation in the market, avoid high rate penalties, and so on, but also gives them time to discover the reasons behind the deviation which occurred. Challenges that may be faced regarding implementation of the proposed method in controlling off-site construction projects are also addressed and solutions are provided.

The implementation and application of the proposed method in practice is illustrated using an actual case study, which demonstrates that the proposed method can noticeably improve the decision making process regarding taking timely corrective measures. Although the method is basically proposed for controlling off-site construction projects, it also can be utilized in other construction projects that involve repetitive operations; e.g., tunneling and earth moving projects.

References

Alvanchi, A., Lee, S., and AbouRizk, S.M. (2011). “Modeling Framework and Architecture of Hybrid System Dynamics and Discrete Event Simulation for Construction.” Computer-Aided Civil and Infrastructure Engineering, 26(2), 77-91.

Azimi, R., Lee, S., AbouRizk, S.M., and Alvanchi, A. (2011 a). “A Framework for an Automated and Integrated Project Monitoring and Control System for Steel Fabrication Projects.” Automation in Construction 20(1), 88-97.

Azimi, R., Lee, S., and AbouRizk, S. M. (2011 b). “Intelligent Look-Ahead Scheduling for Structural Steel Fabrication Projects.” International Journal on Advances in Software, In Press.

Chang, J.R., and Hung, C.T. (2007). “Development of Pavement Maintenance and Rehabilitation Decision Model by Group Method of Data Handling (GMDH).” Proc., ASCE International Workshop on Computing in Civil Engineering, 51-58.

Chao, L.C., and Skibniewski, M.J. (1994). “Estimating construction productivity: Neural-network-based approach.” Journal of Computing in Civil Engineering, ASCE, 8 (2), 234-251

Chin, S., Yoon, S., Choi, C., and Cho, C. (2008). “RFID+4D CAD for progress management of structural steel works in high-rise buildings.” Journal of Computing in Civil Engineering, 22 (2), 74-89.

Deif, A. (2007). “Dynamic Analysis of Agile Manufacturing Planning and Control (MPC) Systems using Control Theory.” Ph.D. thesis, University of Windsor, Canada.

Dissanayake, M., and Fayek, A. R. (2008). “Soft Computing Approach to Construction Performance Prediction and Diagnosis.” Canadian Journal of Civil Engineering, 35(8),764-776.

Eastman, C. M., and Sacks, R. (2008). “Relative Productivity in the AEC Industries in the United States for on-Site and Off-Site Activities.” Journal of Construction Engineering and Management, 134(7), 517-526.

Fayek, A. R., and Oduba, A. (2005). “Predicting Industrial Construction Labor Productivity using Fuzzy Expert Systems.” Journal of Construction Engineering and Management, 131(8), 938-941.

Findeisen, R., Allgöwer, F. and Biegler, L. (2007). “Assessment and Future Directions of Nonlinear Model Predictive Control.” Springer, Berlin, New York.

Ivakhnenko, A.G. (1971). “Polynomial Theory of Complex Systems.” IEEE Transactions on Systems, Man and Cybernetics SMC-1, no. 4, 364-378.

Karshenas, S., and Feng, X. (1992). "Application of neural networks in earthmoving equipment production estimating." Proc., 8th Conf. Computing in Civil Engineering, New York, NY., 841-847.

Kim, S., Park, C., Lee, S., and Son, J. (2008). "Integrated cost and schedule control in the Korean construction industry based on a modified work-packaging model." Canadian Journal of Civil Engineering, 35(3), 225-35.

Law, A. M., and Kelton, W. D. (2000). "Simulation Modeling and Analysis." 3rd ed., McGraw-Hill, Boston.

Leigh, J.R. (2004). "Control Theory." 2nd Ed., The Institution of Electrical Engineers, London, UK.

Lu, M., AbouRizk, S.M., and Hermann, U.H. (2001). "Sensitivity analysis of neural networks in spool fabrication productivity studies." Journal of Computing in Civil Engineering, ASCE, 15(4), 299-308.

Lu, N. (2009). "The Current Use of Offsite Construction Techniques in the United States Construction Industry." Proc., Construction Research Congress, ASCE, Seattle, USA, 946-955.

Ma, Y. (2005). "Work-in-Process and Production Control of Manufacturing Systems using Optimal Control Theory." Ph.D. thesis, University of Michigan, MI.

Maier, H. R., and Dandy, G. C. (2000). "Neural networks for the production and forecasting of water resource variables: a review and modeling issues and application." Environmental modeling and software, 15, 101–124.

McKay, S. (2010). "A Control Theory Based Hybrid Architecture to Anticipate and Shape Adversarial Behavior." Ph.D. thesis, Purdue University, IN.

Müller, J. A. (1998). "Automatic model generation." System Analysis Modeling Simulation (SAMS), 31(1), 1–32.

Navon, R. and Sacks, R. (2007). "Assessing research issues in automated project performance control (APPC)." Automation in Construction, 16(4), 474-84.

Ogata, K. (2002). "Modern Control Engineering." 4th Ed., Prentice Hall, New Jersey.

Onwubolu, G. C. (2009). "Prediction of Burr Formation during Face Milling using a Hybrid GMDH Network Model with Optimized Cutting Conditions." International Journal of Advanced Manufacturing Technology, 44(11-12), 1083-1093.

Sanders, S.R., and Thomas, H.R. (1993). "Masonry productivity forecasting model." Journal of Construction Engineering and Management, 119(1), 163-179.

Smith, S.D. (1999). “Earthmoving productivity estimation using linear regression techniques.” Journal of Construction Engineering and Management, 125(3), 133-141

Song, L., and AbouRizk, S.M. (2008). “Measuring and modeling labor productivity using historical data.” Journal of Construction Engineering and Management, 134 (10), 786-794.

Thomas, H. R., and Sakarcan A. S. (1994). “Forecasting Labor Productivity using Factor Model.” Journal of Construction Engineering and Management 120(1), 228-239.

Thomas, H.R. and Sanvido, V.E. (2000). “Role of the Fabricator in Labor Productivity.” ASCE Journal of Construction Engineering and Management, 126(5), 358-365.

Verhaegen, M., and Verdult, V. (2007). “Filtering and System Identification: A Least Squares Approach.” Cambridge University Press, Cambridge, New York.

Wiendahl, H. P., and Breithaupt, J. W. (2000). “Automatic Production Control Applying Control Theory.” International Journal of Production Economics, 63(1), 33-46.

Chapter 5. Conclusion and recommendations

5.1. Summary of thesis work

Successful delivery of construction projects requires disciplined project management equipped with updated knowledge and skills and efficient decision-making tools and techniques. The main objective of this research was advancing the project management body of knowledge through introducing enhanced decision making tools and techniques that lead to taking a timely and effective course of corrective actions in the fast-paced, dynamic environments of off-site structural steel projects. The provided tools and techniques were targeted toward project monitoring and control as two important and challenging functions in project management. In conjunction with a major structural steel fabrication company in Canada, the thesis research conducted successful case studies on the theoretical basis and practical considerations for improving project monitoring and control in off-site construction. The thesis research has addressed: (1) how to efficiently bridge the gap between monitoring project performance and decision making regarding taking corrective actions; (2) how to capture high-quality data in real-time and translate raw data into meaningful managerial information and use them intelligently for enhancing the decision making process; and (3) how to utilize control theory concepts for effective and prompt reaction to anticipated or occurred performance deviations. To do so, three major steps were taken: First, a framework for integrated and real-time project monitoring and control was proposed and its practicality and feasibility was verified (Chapter 2). Second, the proposed framework was utilized for improving look-ahead scheduling, which is one of the more difficult and challenging tasks for project managers in structural steel fabrication projects (Chapter 3). Finally, an approach based on control theory concepts was introduced to enhance the control process, focusing on timely data analysis and prompt decision making for off-site construction projects (Chapter 4).

5.2. Academic contributions

This research project contributes to the following areas:

- Proposing an automated and integrated project monitoring and control framework which possesses flexibility and interoperability for facilitating the decision making process for off-site construction project managers. This has been documented in a paper titled, "A Framework for Automated and Integrated Project Monitoring and Control System for Steel Fabrication Projects", and has been published in the *Journal of Automation in Construction*. The functionality and capabilities of the proposed framework were also illustrated by using the framework for improving project scheduling. This has been documented in a paper titled, "Intelligent Look-Ahead Scheduling for Structural Steel Fabrication Projects", and has been published in the *International Journal on Advances in Software*;

- Enhancing the project control process regarding taking timely corrective actions by proposing a new vision in controlling off-site construction projects based on Control Theory concepts. This has been documented in a paper titled, "Revisiting project control using basic control theory principles: an application to off-site construction shops", which has been submitted for publication by the *Journal of Computing in Civil Engineering*, ASCE.

5.3. Industrial contributions

- The automated real-time data capturing system developed and implemented at the collaborating company considerably improved the quality of data captured for monitoring and control purposes. It also helped in terms of reducing cost and paperwork associated with the data collection process. Within the implemented system, real-time actual data are captured and compiled to measure progress against targets and key performance indicators.
- Intuitive 5D reports generated promptly based on the actual data captured by the automated data acquisition system and the data sharing features of the developed framework enable project managers to monitor ongoing projects in real-time, be aware of variances and take action while sitting in their offices, which may be far from the fabrication shop.
- The simulation model developed establishes a powerful tool for identifying bottlenecks and forecasting the performance of structural steel fabrication

projects at different intervals and at completion which leads to improvements regarding the accuracy of planning, scheduling and control practices.

- Improvements in project monitoring and control discussed in this research have great potential for cost and time savings in ongoing and forthcoming off-site construction projects (e.g., by reducing paper work).

5.4. Lessons learned

Over the past few years as a PhD student, I have learned many valuable lessons both regarding completing a PhD and the technical issues encountered in performing the research. Some of these lessons are shared as follows:

1- As a student I had many ideas regarding different aspects of the research topic. Over time, I found it very important to talk to experts about the value and feasibility of the ideas. Many people may find it a bit intimidating going to a supervisor or an expert with new ideas and discuss it but I believe it is worth it. In my case, talking about my ideas also resulted in providing some perspective and helping clarify what areas I was interested in.

2- I found time management and planning in advance are very important skills I needed to be equipped with in graduate studies. For instance, in many cases I needed to work on several items such as reports, programming, and academic papers simultaneously. For me, it seemed that everything took longer than I thought it would. This means that to be able to complete all the items I was working on at the scheduled time; I needed to create good plans in advance and improve my time management skills.

3- Having my supervisors' feedback on my work helped me a lot. Sending a draft to my supervisors and having it come back full of Track Changes and/or red pen was disappointing for me at the beginning. But after a while it came to me though, that this was one of the ways I was going to learn. In most cases changes were on wording and writing style rather than the ideas. I learned that it is always a good idea to be willing to learn from people who have been doing it for a while.

4- Another important lesson I learnt was that I'm not supposed to reinvent the wheel. I realized that sometimes there was a simple but effective solution right there for what I wanted to do. In spite of the fact that the construction industry is unique, it has many similarities with other industries. I found that the concepts, methods and tools that have been successfully used in other industries can be adopted and with some modifications can be utilized in the construction industry to resolve some of its problematic areas. However, such modifications are not always easy or feasible.

5- Speaking technically and specifically about my research, I found some areas for improvement with Simphony.NET 3.5. For instance, dynamic changes regarding the number of resources are not feasible in the current version of Simphony which forced me to use another simulation software package (i.e., AnyLogic®).

6- Despite the fact that COSYE is a powerful tool enhanced with the required simulation core services and specific modeling tools for creating distributed simulation models for complex applications in the construction domain, it still needs improvements. At the time of developing the structural steel federation, COSYE did not provide a user friendly environment and developments required a great deal of coding. Efforts need to be made to make COSYE more user friendly which can significantly minimize the end user burden for developing distributed simulation systems.

5.5. Recommendations for future research

The framework presented in this research along with the proposed new vision for taking timely corrective actions have created a solid foundation for future research. Future work can be conducted based on the findings from this study; some potential research areas can be summarized as follows:

1) The proposed framework and monitoring and control tools can be used for continuous off-site construction productivity improvement. Future research can be performed by utilizing the developed framework combined with process improvement guidelines to improve the overall system performance. Lean production principles can be used for identifying potential areas for productivity improvement in the off-site construction supply chain. A generic method needs to be introduced for facilitating the process of evaluating and deploying the productivity improvement efforts. The proposed framework can

be used to check the effectiveness of the productivity improvement efforts before they are actually carried out. By verifying the productivity improvement efforts with the tools proposed in this research, risks associated with changes in the off-site construction environment can be minimized.

2) Herein, as discussed in detail in Chapter 4, the proposed vision behind decision making is taking corrective actions to maintain desired project and system performance without dealing with the reasons that caused the deviations. Methods need to be introduced regarding reliable and accurate detection of the root cause of performance variance. Concepts of "fault diagnosis," which encompasses detection, analysis and recovery of performance deviations, could be a sound foundation for this purpose.

3) One of the areas that still needs improvement is the visualization area. There is a lack of a simple and intuitive way for sharing the necessary project performance information. For instance, in this research cost and schedule information were added to the 3D models of the ongoing projects and resulted in constructing 5D models. To be able to have access to the generated 5D models, project managers needed to have specific visualization software installed on their computers. Attempts should be made to make use of the Industry Foundation Classes (IFC) data models, which are a commonly used format for Building Information Modeling (BIM) to facilitate interoperability and data sharing. This can allow parties involved in the projects access to

visualized project performance data without requiring installation of any special visualization software.

4) The structural steel federation was developed particularly from fabricators' point of view. However, different stakeholders and parties are commonly involved in steel projects. Research needs to be performed for identifying the areas where these parties can contribute to the federation. Federation extension and developments can be carried out accordingly.

Appendix A: Resource Performance Indicators in Controlling Industrial Steel Projects[5]

Reza Azimi[a], SangHyun Lee[b] , and Simaan M. AbouRizk[a]

[a]Department of Civil & Environmental Engineering, Hole School of Construction University of Alberta, Edmonton, Alberta, Canada

[b]Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, 48109, USA

Abstract:

Industrial steel construction deals with the design, fabrication, and site erection of steel elements and encompasses many activities, interactions, and potential resources. An effective project control system for such an environment, as well as issues concerning traditional control systems, are important. This has motivated the authors to propose an integrated and real-time monitoring and control system for steel projects. This paper discusses the application of the concept of resource performance indicators in major components of the proposed system that deal with measuring project performance and the simulation of different scenarios. An object model is utilized and a control object is defined for steel projects to ease the control process. The focus of this research is on labour performance indicators in the fabrication phase of industrial steel projects. To improve the accuracy of the

---

[5] This paper has been published at proceedings of the ASCE Construction Research Congress, May 2010, pp. 21-30.

outputs of the system, the Markov Chain method has been used to forecast the values of the resource performance indicators that are fed into the simulation model.

Introduction

Industrial steel construction deals with a variety of buildings, from heavy industrial buildings and petrochemical refineries to sheds, shelters, and roofs. Reduced construction times, efficiency, and cost-effectiveness can be considered as the major benefits of using industrial steel elements. Detailing (designing), procurement, steel fabrication, shipment, and site erection are five major phases in a typical industrial steel construction project.

The complex nature of steel construction (i.e., the large number of potential resources, activities, and interactions required, the industry's growth, and the adoption of new fabrication technologies and materials) requires advanced and effective tools to estimate, schedule, monitor, and control industrial steel projects. The authors (Azimi et al. 2009) proposed an automated and real-time monitoring and control system for steel projects. With the help of a collaborative steel company, the components of the framework are being implemented in the fabrication phase of steel construction. Steel fabrication refers to the manufacturing of steel pieces through a series of operations, including cutting, fitting, welding, sand blasting, and painting, in a fabrication shop according to the engineering design. Steel fabrication is one of the most complicated phases in industrial steel construction projects because of the unique of steel elements, the huge number of fabricated pieces, material handling and inspection activities, the large variety of steel pieces produced in terms of dimension and processing requirements, and the variety of equipment and labour disciplines involved.

The focus of the paper will be on the resource performance indicators used in the system for automated analysis purposes. Generally speaking, resource performance indicators are used for reasoning (Li et al. 2005) and this paper explains how these indicators can be used and forecasted to improve the reliability of the outputs of the automated control system.

Background

Monitoring and control refers to measuring the actual values of the basic variables of a project (such as the percent completed, the quality of work, and cost), analyzing these data, determining if the project is meeting the targets of the work plan, and taking necessary actions to meet project objectives if there are any deviations in project performance (Oberlender 2000). The construction industry is an enormous industry in North America and a huge amount of money is spent in this industry. As an example, in the first eleven months of 2007, US$12.82 trillion was spent in construction projects in the US (www.researchandmarkets.com/reports/656143). This highlights the importance of the role of monitoring and control systems in construction due to the fact that a small variance in the performance of the ongoing projects leads to the loss of billions of dollars. Thus, issues regarding current monitoring and control systems have developed into a major field of interest for researchers. In terms of monitoring, project managers and control systems require a high quality and real-time data to take timely and proper corrective actions after any deviation occurs. The emergence of new technologies, such as Radio Frequency Identification (RFID), has helped researchers to capture performance data for in-progress projects in real-time (Chin et al. 2008; Navon and Sacks 2007; Ghanem and Abdelrazig 2006; Cheng and Chen 2002; Ciesielski 2000). To be effective, there needs to be integration between monitoring and controlling. Gaps between monitoring and controlling components (e.g., evaluation, analysis, and taking corrective actions) may lead to delays and cost overruns. To deal with this issue, integrated monitoring and control systems are under development (Diekmann and Al-Tabtabai 1992; Kim et al. 2008; Moselhi et al. 2004; Nasr 2005; Nassar et al. 2003; Po-Han Chen 2008). However, current systems are suffering from at least one of the following shortfalls (Azimi et al. 2009):

- Lack of some important components of an integrated project control system, such as forecasting performance, analyzing variances, and recommending corrective actions.
- Manual data entry, which is slow and error-prone.
- Inflexibility and lack of interoperability.
- Difficulties in terms of sharing data (centralized and office-based systems).

To address these issues, Azimi et al. (2009) proposed a project control framework in which real-time, as-built information from a monitoring system is integrated with a control system that includes different components and simulation models (Figure A-1). The framework, which uses High Level Architecture (HLA) concepts (IEEE 1516), enables project managers to monitor the progress and performance of each project in a timely manner and to be notified automatically of any deviations; this notification includes complete information regarding the impacts of these problems.

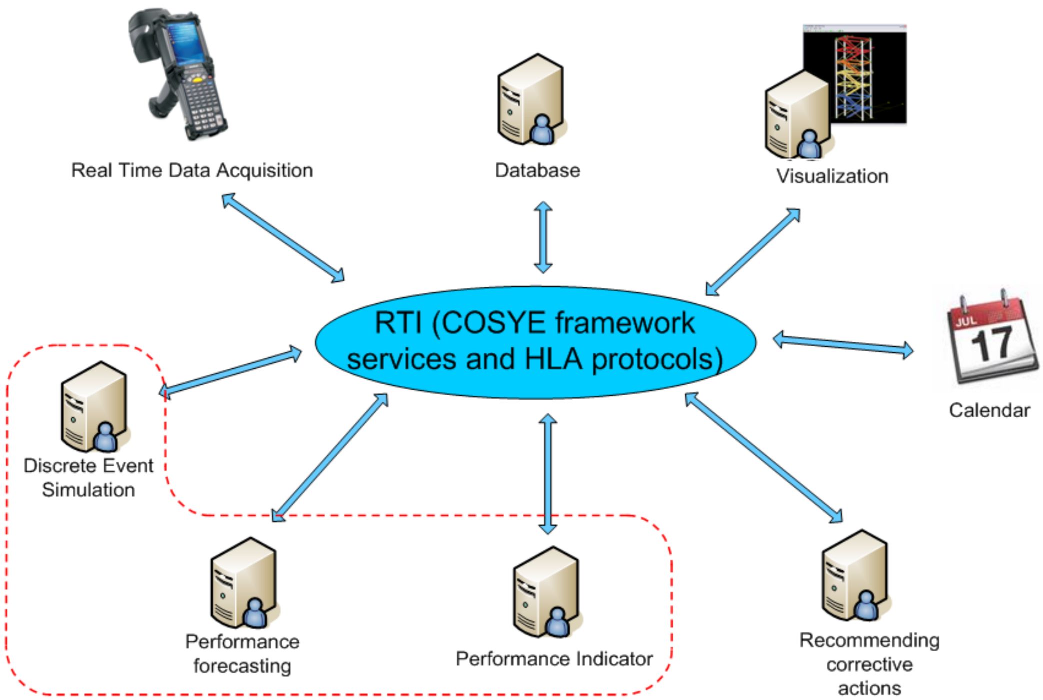


Figure A-1. Proposed monitoring and control framework (Azimi et al 2009).

The infrastructure, the components, and the interactions between the components in the proposed framework have been discussed previously (see Azimi et al. 2009). This paper therefore deals specifically with the development and implementation aspects of the framework components marked with the red dashed line in Figure A-1. A prototype of the framework has been implemented in the collaborative company. An automated data capturing system using RFID technology provides real-time information for other components of the system;

evaluation and analysis regarding project performance is carried out by other federates based on captured data. The following sections explain how current and predicted project performance are determined for controlling in the implementation phase of the framework.

Earned Value Analysis on Control-Objects

For control purposes, the level of detail for monitoring and analyzing project performance information needs to be determined. Figure A-2 represents the Work Breakdown Structure (WBS) of a typical steel fabrication project.

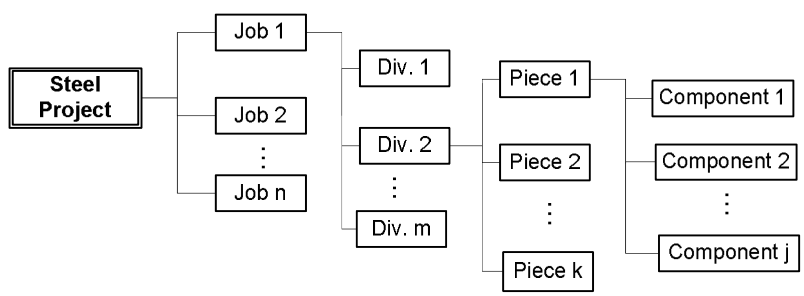


Figure A-2. WBS of a typical steel fabrication project.

In this research, division and project levels are monitored and controlled. Once progress and performance information at the division level is calculated with the bottom-up approach, project performance at the project level can be determined. A steel project is represented by a set of divisions, which are the control objects. Each division (control object) has its own resources (i.e., labour, material, equipment and, in rare cases, subcontractors) as well as as-planned and actual data, as shown in Figure A-3.

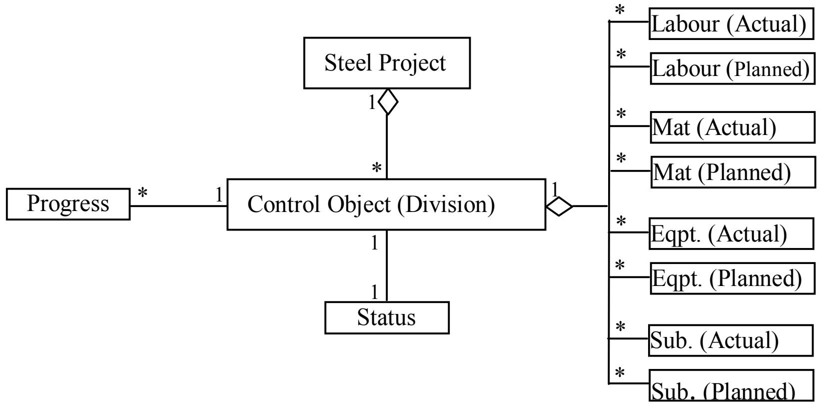


Figure A-3. Proposed object model of a steel fabrication project.

Status class refers to the actual start and finish time of a division. Additionally, progress of the division is considered as a class in the model.

To control project performance at the division level, the Earned Value Method (Li 2004) is used to disclose deviations in cost and time in project performance. This is done by calculating two indices: the Cost Performance Index (CPI) and the Schedule Performance Index (SPI). The CPI determines if a budget is over or under run, while the SPI determines if a project is ahead of or behind schedule. To determine the performance indices at the division level, actual cost (man-hours spent in processing the division) and actual progress (percent completed) need to be calculated. Actual cost (man-hours) can be derived from the implemented actual data acquisition system by adding up the man-hours spent on processing pieces for the division. The implemented data acquisition system makes use of Radio Frequency Identification (RFID) technology to track labour and material in the fabrication shop (Azimi et al. 2009). The progress of each division is determined by dividing the weight of the fabricated pieces of that division (captured by the data acquisition system) by the weight of the division (recorded in the database). For performance variance at the division level, the cost and schedule performance indices (CPI and SPI) at the resource level are examined to determine the resource(s) that caused the deviation. A division with a variance in the CPI, for example, has reasonable cost performance indices (CPI≈1) for all resources except labour; labour then is considered to be the cause of the performance variance.

Resource Performance Indicators

Li (2004) introduced the concept of the resource performance indicators to identify the cause of variance in each performance index. Resource performance indicators therefore are used to detect problematic areas (Li et al. 2005). The monitoring and control system of this study is implemented in a collaborative company where labour is the major concern; this research thus takes into account three performance indicators suggested by Li et al. (2005) regarding labour performance in a control object (Table A-1).

Table A-1. Labour performance indicators used in the proposed control system.

Labor Performance Indicator	Indicator ratio
Division production	Qa / Qb
Unit productivity (fitting, welding & painting)	$\frac{Qa \times Mhrsb}{Qb \times Mhrsa}$
Crew attendance	$\frac{\sum_{t=1}^{n} N_t}{N_b \times n}$

Where:

Mhrsa : cumulative actual man-hours spent to process each steel division

Mhrsb : budgeted man-hours (cumulative)

Nb: planned number of crew members

Nt: actual number of crew members on the t^{th} day

Qa: actual number of processed divisions

Qb: number of divisions planned to be processed

n: the number of days that have been passed

The labour performance indicators are used for two purposes: reasoning and simulation. In terms of reasoning, a list of problem source factors is linked to a list of possible corrective actions and the best required action can be determined. For example, in the case of low unit productivity ratio (less than 1), inadequate skilled labor force or low work moral can be considered as two sources of problem. To deal with that, one can consider adding skilled workers and/or reallocating labour forces as possible corrective actions. Li et al. (2005) discusses in detail how the reasoning can be conducted using the resource performance indicators which is out of the scope of this paper.

Resource Performance Indicators in Simulating Steel Fab Shop

Simulation has been proved to be a powerful tool to analyze and evaluate complicated systems and processes such as steel fabrication process. The simulation in the proposed control system enables managers to experience and compare different scenarios in virtual reality and find the best solution for current or anticipated variances in a project(e.g. delays) as well as the fabrication shop (e.g. low utilization rate and long queues of steel pieces behind a station). To be reliable, the simulation model not only must be a good emulation of the operations and processes of a system in the real world, but the influencing factors as well. Given that in the collaborative company, the labour has the most contribution on the performance of projects, labour performance indicators have been involved in the simulation model.

While different scenarios (e.g. increasing active stations, overtime hours and etc.) can be tried, three scenarios are available for the simulation model regarding the labour performance indicators:

1. Fixed
2. User defined
3. Automatically forecasted

In the first scenario, the most recent value of each indicator captured by the data acquisition system is fed in to the simulation model and as the advances in the model, these variables don't change. Project managers can play with numbers regarding all the influencing factors and resource performance indicators in the second scenario to see the effects of changing each variable on the final results of the simulation. In the third scenario, the value of labour performance indicators at certain times in the future (e.g. on monthly basis) are automatically forecasted and are fed into the simulation model. In this case, the final simulation results are a better representation of the project performance and the fabrication shop in future.

Forecasting Labour Performance Indicators

A) Crew attendance indicator: Historical data has been used to predict the crew attendance indicator in this research. The database of a labour tracking system keeps records of timekeeping information on daily basis which forms the historical data. A preliminary analysis on the available data shows that on mondays and fridays or during certain months such as June, July and August crew attendance indicator is more likely deviating from its desired value which is one. Although crew attendance indicator can be predicted on daily or weekly basis based on the past information, for the implemented system, this indicator is forecasted on monthly basis.

B&C) Unit productivity & Div. Production: Industrial steel projects typically have different stages and processes which are dynamic and complex. This makes it difficult to predict the labour Unit Productivity Indicator (UPI) and Division Production Indicator (DPI) and forecasting these indicators just based on historical data is not always accurate. There are many internal and external factors that can affect these indicators, and thus a system should be designed to model the dynamic nature of these indicators. A good forecasting embraces both historical data and reliable judgments based on construction experience and knowledge (Diekmann and Al-Tabtabai 1992). To address the stochastic nature of these two indicators, the concept of the Dynamic Markov Chain method introduced by Nassar (2005) is used in this research for predicting purposes. In this model, forecasting is based on current indicators as well as the influencing factors that affect the indicators. These influencing factors get involved in the model using the judgment of experts. Five states for UPI and DPI at time t, (i.e. A(t), B(t), C(t), D(t), & F(t)) are defined as in Table A-2.

Table A-2. List of states for the UPI & DPI.

State	Indicator Rating	Ratio Range	Average Index (PI)
A	Outstanding	R>1.15	1.2
B	Exceeds Target	1.05<R<=1.15	1.1
C	Within Target	0.95<R<=1.05	1
D	Below Target	0.85<R<=0.95	0.9
F	Poor	R<=0.85	0.8

Once the states of the indicators are defined, probability transition matrices for each of the indicators can be established (Figure A-4).

$$[P]_{UPI}= \begin{array}{c} State \\ A \\ B \\ C \\ D \\ F \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & F \end{array} \\ \begin{bmatrix} P_{AA} & P_{AB} & P_{AC} & P_{AD} & P_{AF} \\ P_{BA} & P_{BB} & P_{BC} & P_{BD} & P_{BF} \\ P_{CA} & P_{CB} & P_{CC} & P_{CD} & P_{CF} \\ P_{DA} & P_{DB} & P_{DC} & P_{DD} & P_{DF} \\ P_{FA} & P_{FB} & P_{FC} & P_{FD} & P_{FF} \end{bmatrix} \end{array} \qquad [P]_{DPI}= \begin{array}{c} State \\ A \\ B \\ C \\ D \\ F \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & F \end{array} \\ \begin{bmatrix} P'_{AA} & P'_{AB} & P'_{AC} & P'_{AD} & P'_{AF} \\ P'_{BA} & P'_{BB} & P'_{BC} & P'_{BD} & P'_{BF} \\ P'_{CA} & P'_{CB} & P'_{CC} & P'_{CD} & P'_{CF} \\ P'_{DA} & P'_{DB} & P'_{DC} & P'_{DD} & P'_{DF} \\ P'_{FA} & P'_{FB} & P'_{FC} & P'_{FD} & P'_{FF} \end{bmatrix} \end{array}$$

Figure A-4. Probability Transition Matrices for UPI and DPI.

Where,

$\boldsymbol{P_{ij}}$= Transition probability of the indicators changing from state **i** at time **t** to state **j** at time **t+1** for i,j= A, B, C, D, and F.

For example, P_{CB} is the probability of unit productivity indicator changing from state **C**, (i.e. within target) at time ***t*** to state **B** (i.e. above target), at time ***t+1***.

Each P_{ij} and/or P'_{ij} can be defined by combining historical data and experts' judgments. The Analytic Hierarchy Process (AHP) is used to take into account judgment of the experts in terms of defining Probability Transition Matrices for UPI and DPI.

Since resource indicators with respect to UPI and /or DPI at any time t has to be one of the states (A, B, C, D, and F), so:

$$\sum_{j=A}^{F} P_{ij} = 1 \qquad \text{For i=A, B, C, D, and F} \qquad \text{Eq. (1)}$$

A state probability vector for the proposed states for any indicator at any time t can be defined as follows:

$P(t) = [P_A(t), P_B(t), P_C(t), P_D(t), P_F(t)]$

Initial probability at the initial time t_0=0 can be defined by using the most recent data captured by the data acquisition system. For example if the actual UPI is equal to 0.97 we have:

$P_C(t_0)=1$ and $P_A(t_0)= P_B(t_0)=P_D(t_0)= P_F(t_0)=0$

In this method: $P(t+1)= P(t)*[P]^{t+1}$ Eq. (2)

Where, $[P]^{t+1}$ represents the probability transition matrix, same as what is shown in Figure A-4 but at time (t+1). In current implementation, it's been assumed that the probability transition matrix does not change by time, in other words: $[P]^{t+1}=[P]$

Thus:

$P_A(t+1)$=Probability of outstanding performance in UPI at time $t+1$

$=[\, P_A(t) * P_{AA}(t+1) + P_B(t) * P_{BA}(t+1) + P_C(t) * P_{CA}(t+1) + P_D(t) * P_{DA}(t+1) + P_F(t) * P_{FA}(t+1)]$

$$= [\, P_A(t), P_B(t), P_C(t), P_D(t)\; P_F(t)] \times \begin{bmatrix} P_{AA}(t+1) \\ P_{BA}(t+1) \\ P_{CA}(t+1) \\ P_{DA}(t+1) \\ P_{FA}(t+1) \end{bmatrix} \qquad \text{Eq. (3)}$$

The same calculation is carried out for determining other probabilities at time $t+1$. Based on the above, having the present value of the indicators (initial state), the forecasted indicator as at time t can be calculated as:

UPI(t)= $\sum_{S=A}^{F} Ps(t) * PIs = P_A(t) * PI_A + P_B(t) * PI_B + P_C(t) * PI_C + P_D(t) * PI_D + P_F(t) * PI_F$

$$= P_A(t) * 1.2 + P_B(t) * 1.1 + P_C(t) * 1.0 + P_D(t) * 0.9 + P_F(t) * 0.8 \qquad \text{Eq. (4)}$$

Where PI_A, PI_B, PI_C, PI_D, and PI_F are the average PI values for different states as proposed in Table A-2. As an example if current actual UPI(t_0)=0.92 (state D) then we have:

$P_D(t_0)=1$ *and* $P_A(t_0)= P_B(t_0)=P_C(t_0)= P_F(t_0)=0$

Assuming $[P]_{UPI}= \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.4 & 0.3 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 & 0.4 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.3 & 0.4 \end{bmatrix}$ & Eq. 3→ $P_A(t=1)=0.1$, $P_B(t=1)=0.2$, $P_C(t=1)=0.1$, $P_D(t=1)=0.4$ and $P_F(t=1)=0.1$.

Eq. 4→ UPI(t=1)= 0.88 (or state D)

According to the Probability Transition Matrix in this example, P_{DD} has the highest probability (i.e. 40%) and remaining in state D at time (t=1) is reasonable. The same approach can be followed to forecast DPI(t), the only difference is the transition matrix that should be involved (i.e. P'_{ij} instead of P_{ij}). It should be noted that different PI (average index) values can be used for each of the two labour performance indicators here. Project control philosophy of the company and the specific conditions of a project can also influence the PI values for the Unit productivity and Division Production indicators. Forecasting these indicators is on a daily basis due to the importance of them on the final results of the simulation model.

Conclusion

A monitoring and control system plays an important role in industrial steel construction owing to its complexity. The ability of the control system to accurately and reliably analyze the current and future situation regarding the project performance is a key to successful completion of the project. Project performance is so sensitive to the performance of the major resources involved. This paper proposes an approach in terms of utilizing the resource performance indicators, which are originally defined as sensors for detecting troublesome areas, evaluating the current status as well as predicting the future status of the ongoing projects. Three labour performance indicators namely Crew attendance, Division production and Unit productivity that significantly influence the fabrication phase of the industrial steel projects were addressed and exploited in the paper. The Markov Chain method was deployed for forecasting the values of the performance indicators used in determining the future performance of projects. The project performance analysis discussed in this paper is a part of a comprehensive automated and integrated monitoring and control system which is under development and being implemented in the real world. Given the intricacy of the industrial steel projects, it is believed that this system can considerably help project managers to mitigate damages and loss in many projects.

References

Azimi, R., Lee, S. and AbouRizk, S. (2009). “Integrated Project Monitoring and Control for Steel Construction." Joint International Conference on Construction Engineering and Management and Construction Project Management (ICCEM-ICCPM), Jeju, Korea., 479-486.

Cheng, M. and Chen, J. (2002). "Integrating barcode and GIS for monitoring construction progress." Automation in Construction, 11(1), 23-33

Chin S, Yoon S, Choi C, Cho C. (2008). "RFID+4D CAD for progress management of structural steel works in high-rise buildings." Journal of Computing in Civil Engineering 22(2),74-89.

Ciesielski, CA. (2000) "ADCIC-automatic data collection in construction." Proceeding of the A.I.M.'s Global Academic and Scientific Conference, Cairo, Egypt.

Diekmann J and Al-Tabtabai H. (1992). "Knowledge-based approach to construction project control." International Journal of Project Management, 10(1), 23-30.

Ghanem A. G. and Abdelrazig Y. A. (2006). "A framework for real-time construction project progress tracking." Earth and space 2006 - 10th biennial international conference on engineering, construction, and operations in challenging environments, League City/Houston, TX, United States: ASCE, Reston, VA, United States.

IEEE (2000) Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) -Federate Interface Specification 1516.1 -2000

Kim S, Park C, Lee S, Son J. (2008). "Integrated cost and schedule control in the korean construction industry based on a modified work-packaging model." Canadian Journal of Civil Engineering, 35(3), 225-35.

Li, J. (2004). "Web-based integrated project control." Ph.D. thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Canada.

Li J, Moselhi O, Alkass S. (2005). "A reasoning process in support of integrated project control." Canadian Journal of Civil Engineering, 32(3), 500-16.

Moselhi O, Li J, Alkass S. (2004). Web-based integrated project control system. Construction Management and Economics 22(1):35-46.

Nasr, E.B. (2005). "An integrated project planning and control system approach for measuring project performance." Ph.D. thesis, University of Colorado at Boulder, USA.

Nassar N., Halepota T. and AbouRizk S. M. (2003). "Integrated project controls system – IPCS." Proceedings of the Annual Conference of the Canadian Society for Civil Engineering, 51-60

Nassar NK. (2005). "An integrated framework for evaluation, forecasting and optimization of performance of construction projects." Ph.D. thesis, Department of Civil and Environmental Engineering, University of Alberta, Alberta, Canada.

Navon R and Sacks R. (2007). "Assessing research issues in automated project performance control (APPC)." Automation in Construction, 16(4), 474-84.

Oberlender, G.D. (2000). "Project management for engineering and construction", The McGraw-Hill Companies, Inc., Boston, MA.

Po-Han, C. (2008). "Integration of cost and schedule using extensive matrix method and spreadsheets.:" Automation in Construction, 18(1), 32-41.

Appendix B: Automated data capture using RFID technology

Introduction

RFID technology was used for real-time data acquisition within this research as discussed in Chapter 1Chapter 2, and Chapter 3. As shown in Figure B-1, the data capturing system is composed of (1) passive RFID tags, (2) a tag reader and (3) a router.

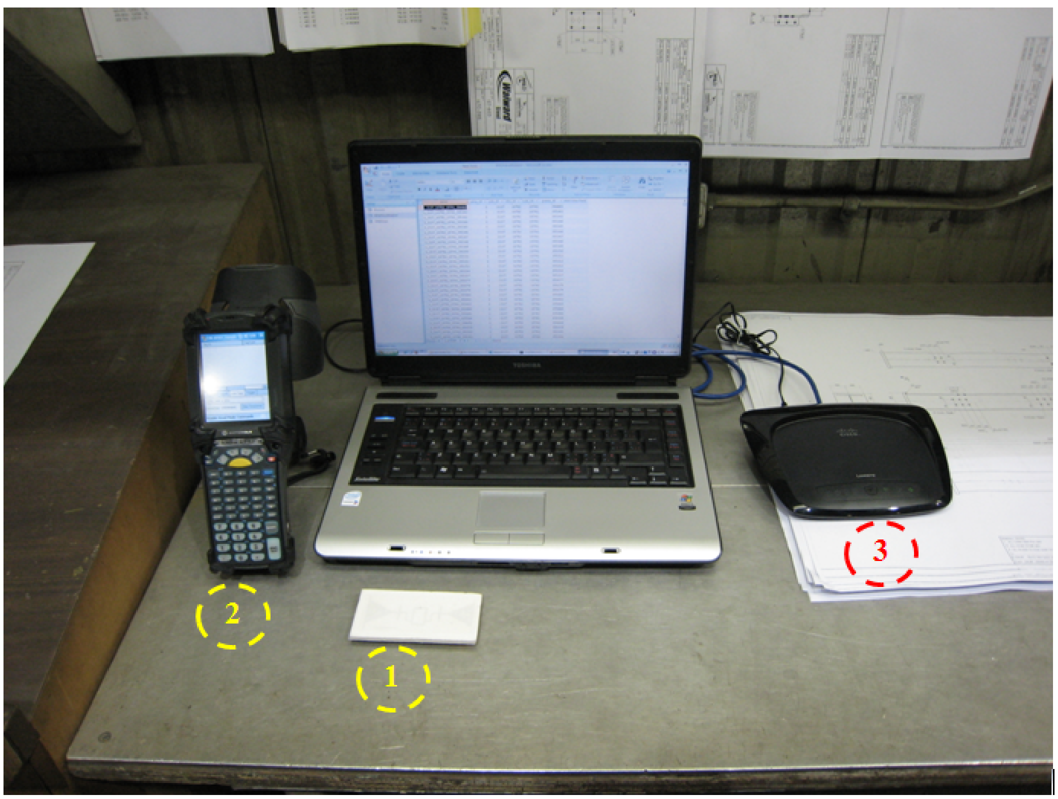
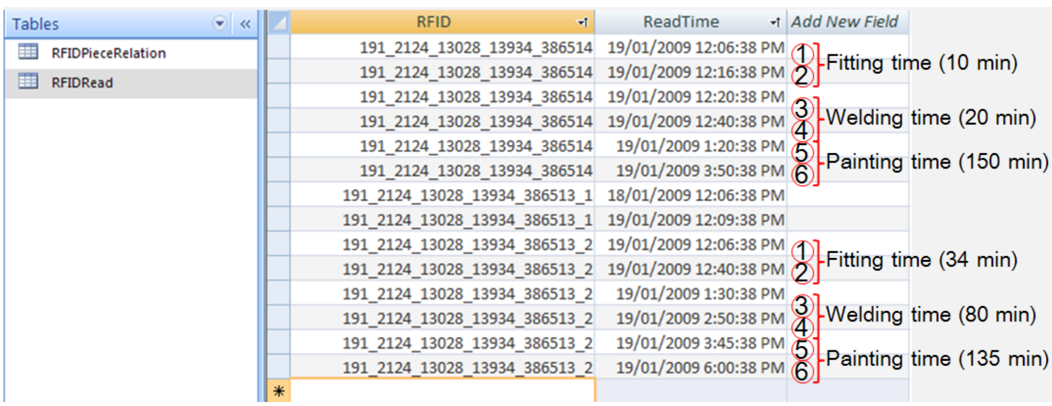


Figure B-1. Data acquisition system based on RFID technology.

a) RFID tags: the tags were writable and steel piece IDs were stored on the tags for piece tracking purposes.

b) Tag reader: at the beginning and at the end of each operation (e.g., fitting operation) the tag reader is used for retrieving data stored on the RFID tags and data collection regarding the location of the steel pieces. The location data are then translated into meaningful data such as activity duration as shown in Figure B-2.



RFID	ReadTime	Add New Field
191_2124_13028_13934_386514	19/01/2009 12:06:38 PM	
191_2124_13028_13934_386514	19/01/2009 12:16:38 PM	
191_2124_13028_13934_386514	19/01/2009 12:20:38 PM	
191_2124_13028_13934_386514	19/01/2009 12:40:38 PM	
191_2124_13028_13934_386514	19/01/2009 1:20:38 PM	
191_2124_13028_13934_386514	19/01/2009 3:50:38 PM	
191_2124_13028_13934_386513_1	18/01/2009 12:06:38 PM	
191_2124_13028_13934_386513_1	19/01/2009 12:09:38 PM	
191_2124_13028_13934_386513_2	19/01/2009 12:06:38 PM	
191_2124_13028_13934_386513_2	19/01/2009 12:40:38 PM	
191_2124_13028_13934_386513_2	19/01/2009 1:30:38 PM	
191_2124_13028_13934_386513_2	19/01/2009 2:50:38 PM	
191_2124_13028_13934_386513_2	19/01/2009 3:45:38 PM	
191_2124_13028_13934_386513_2	19/01/2009 6:00:38 PM	

Figure B-2. Activity durations calculated based on captured data.

c) Router: the router is the medium that enables the tag reader to pass the collected data in digital form to a computer system.

Data collection included developing an interface for the tag reader device (Figure B-3) and coding the device as well as developing an interface in Visual Basic.NET which runs on the computer, receives data from the tag reader, and updates the database. As shown in Figure B-3 the IP address of the machine that receives data should be defined for the tag reader. If inspectors find defective pieces they touch the related box on the screen of the reader (i.e., Fit. Rew. or

Weld. Rew. as shown in Figure B-3) before pressing the trigger button. In the case of fitting rework number 1 is stored with the piece ID in the database and if welding rework is required number 2 is stored with the piece ID in the database.

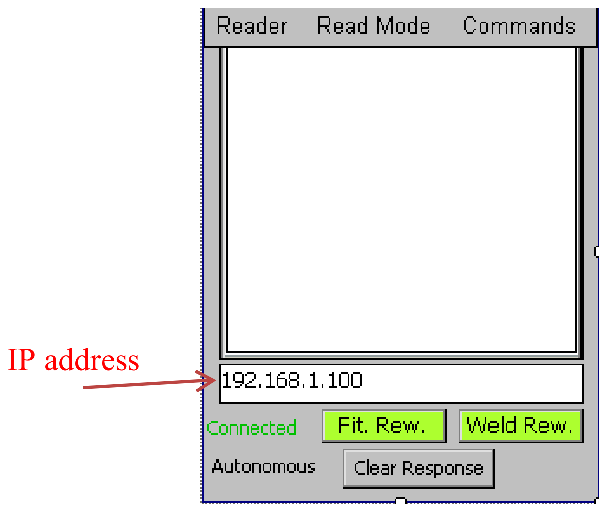


Figure B-3. Developed interface for the tag reader.

Once captured, the actual data are used for setting up the initial condition of the fabrication shop simulation model and the simulation model is fed with the most recent data which results in more accurate simulation results. A schematic view regarding the interaction between the data acquisition system and the structural steel federation and some sample areas in the DES federate that are influenced by the data capture system is depicted in Figure B-4.

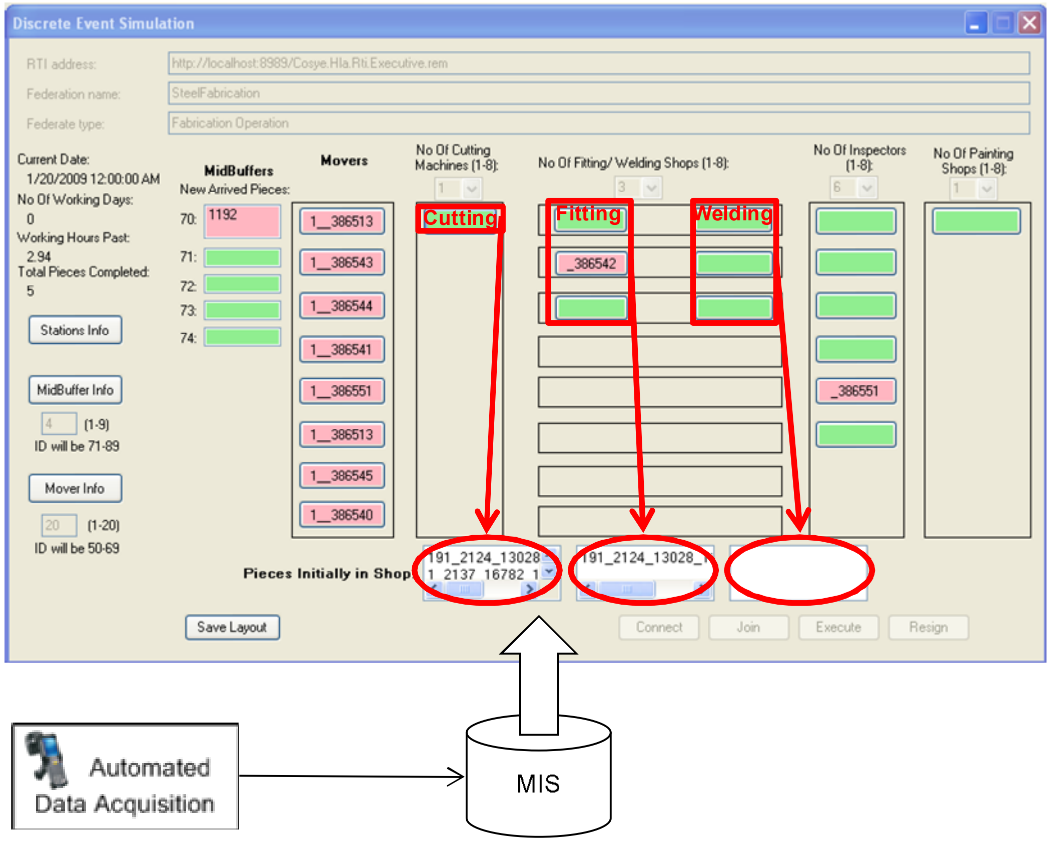


Figure B-4. A schematic view regarding the integration between the developed federation and RFID technology.

Coding the tag reader device

Visual Basic.NET 2008 has been used as the programming environment for coding the RFID tag reader device and developing a user friendly interface for the device, easing the process of data collecting for the people using the device in the steel fabrication shop. In addition to the standard classes embedded in the Visual Basic, the set of class libraries and sample applications provided by Enterprise Mobility Developer Kit (https://docs.symbol.com/KanisaPlatform/Publishing/652/12874_f.html) have also been imported and used for coding the tag reader device. The main body of code is presented below:

```vb
Imports System.Collections.Generic
Imports System.Collections
Imports System.ComponentModel
Imports System.Data
Imports System.Drawing
Imports System.Diagnostics
Imports System.Net
Imports System.Net.Sockets
Imports System.Text
Imports System.Xml
Imports System.IO
Imports System.Reflection
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Symbol.RFID2
Imports Symbol.ResourceCoordination
Imports System.Threading
Partial Public Class MainFormMobil
    Inherits Form
    Private Class ErrorCodes
        Public RFID_WRITE_NOT_VERIFIED As ULong = &H94
        Public RFID_WRITE_ERASE_ERROR As ULong = &H95
        Public RFID_WRITE_LOCK_ERROR As ULong = &H96
        Public RFID_TAG_ERR_MEM_OVERRUN As ULong = &H98
        Public RFID_TAG_ERR_MEM_LOCKED As ULong = &H99
        Public RFID_TAG_ERR_LACK_POWER As ULong = &H9A
        Public RFID_TAG_ERR_NON_SPECIFIC As ULong = &H9B
    End Class
    Private displayTextLock As New Object()
    Public Delegate Sub DisplayOutputHandler(ByVal text As String, ByVal args As ReaderEventArgs)
    Public Delegate Sub DisplayAutoOutputHandler(ByVal args As TagEventArgs)
    Private deviceReader As IRFIDReader = Nothing
    Private Const m_ReaderName As String = ""
    Private m_ReaderModel As ReaderModel = Program.m_ReaderModel
    Private htInParams As Hashtable
    Private displayHandler As DisplayOutputHandler = Nothing
    Private displayAutoHandler As DisplayAutoOutputHandler = Nothing
    Private MyTrigger As Symbol.ResourceCoordination.Trigger = Nothing
    Private MyTriggerHandler As Symbol.ResourceCoordination.Trigger.TriggerEventHandler = Nothing
    Private TriggerDev As Symbol.ResourceCoordination.TriggerDevice = Nothing
    Private sr As New Gen2Parameters(Gen2ParameterSel.Ignore_SL, Gen2ParameterSession.S0,
Gen2ParameterTarget.Bit_A, 4)
    '##########################
    Private SelRecPage As FrmSelectRecord = Nothing
    Private Gen2ReadPage As FrmGen2Read = Nothing
    Private Gen2WritePage As FrmGen2Write = Nothing
    Private LockTagPage As FrmGen2Lock = Nothing
    Private KillTagPage As FrmGen2Kill = Nothing
    '##########################
    Public Sub New()
        InitializeComponent()
        Try
            displayHandler = New DisplayOutputHandler(AddressOf DisplayText)
            displayAutoHandler = New DisplayAutoOutputHandler(AddressOf DisplayAutoTags)
            Dim strPath As String = Assembly.GetExecutingAssembly().ManifestModule.FullyQualifiedName
            strPath = strPath.Replace(Assembly.GetExecutingAssembly().ManifestModule.Name, "DeviceReader.Config")
            Dim configStreamStr As String = String.Empty
            Try
                If Not File.Exists(strPath) Then
                    CreateFile(strPath)
                End If
                Dim readerStream As New StreamReader(strPath)
                Dim configStream As Stream = readerStream.BaseStream
                Dim configBytes As Byte() = New Byte(Convert.ToInt32(configStream.Length) - 1) {}
                configStream.Read(configBytes, 0, configBytes.Length)
                configStreamStr = System.Text.Encoding.UTF8.GetString(configBytes, 0, configBytes.Length)
```

```
            deviceReader = ReaderFactory.CreateReader(m_ReaderModel, configStreamStr)
            SetupTriggerResource()
            SetupEvents()
          Catch
            If m_ReaderModel = ReaderModel.MC9090 Then
              configStreamStr = "<?xml version='1.0' ?>" & vbCr & vbLf & "                              <ReaderConfig
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>" & vbCr & vbLf & "
<ComPortSettings>" & vbCr & vbLf & "                              <COMPort>COM7</COMPort>" & vbCr & vbLf & "
<BaudRate>57600</BaudRate>" & vbCr & vbLf & "                              </ComPortSettings>" & vbCr & vbLf & "
<ReaderInfo>" & vbCr & vbLf & "                              <Model>MC9090</Model>" & vbCr & vbLf & "
<StartingQ>6</StartingQ>" & vbCr & vbLf & "                              <StartingQWrite>6</StartingQWrite>" & vbCr &
vbLf & "                              </ReaderInfo>" & vbCr & vbLf & "                              </ReaderConfig>"
            Else
              configStreamStr = "<?xml version='1.0' ?>" & vbCr & vbLf & "                              <ReaderConfig
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>" & vbCr & vbLf & "
<ComPortSettings>" & vbCr & vbLf & "                              <COMPort>COM3</COMPort>" & vbCr & vbLf & "
<BaudRate>57600</BaudRate>" & vbCr & vbLf & "                              </ComPortSettings>" & vbCr & vbLf & "
<ReaderInfo>" & vbCr & vbLf & "                              <Model>RD5000</Model>" & vbCr & vbLf & "
<StartingQ>6</StartingQ>" & vbCr & vbLf & "                              <StartingQWrite>6</StartingQWrite>" & vbCr &
vbLf & "                              </ReaderInfo>" & vbCr & vbLf & "                              </ReaderConfig>"
            End If
            deviceReader = ReaderFactory.CreateReader(m_ReaderModel, configStreamStr)
            SetupTriggerResource()
            SetupEvents()
          End Try
        Catch ex As Exception
          MessageBox.Show(ex.Message, "CS_RFID2_Sample")
          Application.[Exit]()
        End Try
    End Sub
    Private Sub deviceReader_TagEvent(ByVal sender As Object, ByVal args As ReaderEventArgs)
        Try
          If args IsNot Nothing Then
            listView1.Invoke(displayAutoHandler, DirectCast(args, TagEventArgs))

            'Gen2ReadPage.Invoke(
            If Gen2ReadPage IsNot Nothing Then
            End If
          End If
        Catch ex As Exception
          MessageBox.Show(ex.Message)
        End Try
    End Sub
    Private Sub SetupTriggerResource()
        Try
          'create a trigger object
          TriggerDev = New
Symbol.ResourceCoordination.TriggerDevice(Symbol.ResourceCoordination.TriggerID.ALL_TRIGGERS,
DirectCast(Nothing, Symbol.ResourceCoordination.TriggerState()))
          MyTrigger = New Symbol.ResourceCoordination.Trigger(TriggerDev)
          'create an event handler and attach a handler method for trigger
          MyTriggerHandler = New Symbol.ResourceCoordination.Trigger.TriggerEventHandler(AddressOf MyTriggerH)
          AddHandler MyTrigger.Stage2Notify, MyTriggerHandler
        Catch ex As Exception
          MessageBox.Show("Failed to create Trigger: " & ex.Message, "Error")
          Shutdown()
          Me.Close()
          Return
        End Try
    End Sub
    Private Sub SetupEvents()
        AddHandler deviceReader.TagEvent, New ReaderEventHandler(AddressOf deviceReader_TagEvent)
    End Sub
    Private Sub Shutdown()
        If deviceReader IsNot Nothing Then
          deviceReader.ReadMode = ReadMode.ONDEMAND
        End If
        Thread.Sleep(50)
        If MyTrigger IsNot Nothing Then
```

```
            MyTrigger.Dispose()
        End If
        If deviceReader IsNot Nothing Then
            deviceReader.Disconnect()
        End If
    End Sub
    ''' DisplayAutoTags processes and displays tag data received from deviceReaderTagEvent handler
    Private Sub DisplayAutoTags(ByVal args As TagEventArgs)
        Dim i As Integer, len As Integer
        Dim j As Integer, ln As Integer
        Dim TagArgs As TagEventArgs = Nothing
        Dim Tags As IRFIDTag() = Nothing
        Dim tag_ID As Byte() = Nothing

        TagArgs = args
        Tags = TagArgs.Tags
        Try
            len = Tags.Length
            For i = 0 To len - 1
                tag_ID = Nothing
                tag_ID = Tags(i).TagID
                Try
                    If tag_ID Is Nothing OrElse tag_ID.Length = 0 Then
                        Return
                    End If
                    Dim strTagID As String = Nothing
                    ln = tag_ID.Length
                    For j = 0 To ln - 1
                        strTagID += tag_ID(j).ToString("X2") & " "
                    Next
                    Dim Message = strTagID '+ ", " + Date.Now.ToString()
                    If FitRewButton.BackColor = Color.Red Then
                        Message = Message + "1"
                    ElseIf WeldRewButton.BackColor = Color.Red Then
                        Message = Message + "2"
                    Else
                        Message = Message + "0"
                    End If
                    Send(IpTextBox.Text, 11000, Message)
                    Beep()
                    FitRewButton.BackColor = Color.GreenYellow
                    WeldRewButton.BackColor = Color.GreenYellow
                    Dim lvItem As New ListViewItem(New String() {strTagID, Tags(i).TagType.ToString(),
Tags(i).LastSeen.ToString("HH:mm:ss"), Tags(i).AntennaName})
                    listView1.Items.Insert(0, lvItem).Selected = True
                    If listView1.Items.Count > 1024 Then
                        listView1.Items.RemoveAt(1024)
                    End If
                Catch ex As Exception
                    MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
                End Try
            Next
        Catch ex As Exception
            MessageBox.Show("DisplayText:" & ex.Message)
        End Try
    End Sub
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
        Try
            If deviceReader.ReaderStatus <> ReaderStatus.ONLINE Then
                deviceReader.Connect()
            End If
            If deviceReader.ReaderStatus = ReaderStatus.ONLINE Then
                ReaderConnected()
            Else
                ReaderDisconnected()
            End If
            InitializeParamHashtable()
        Catch
            MessageBox.Show("Error: Reader not initialized", "CS_RFID2_Sample")
```

```
        ReaderDisconnected()
        Application.[Exit]()
    End Try
End Sub
Private Sub ReaderConnected()
    lblConnect.Text = "Connected"
    lblConnect.ForeColor = Color.Green
    menuCommands.Enabled = True
    menuReaderMode.Enabled = True
    menuSetAntenna.Enabled = True
    menuReaderInfo.Enabled = True
    menuConnect.Enabled = False
    menuDisconnect.Enabled = True
    lblReaderMode.Text = "OnDemand"
    deviceReader.ReadMode = ReadMode.ONDEMAND
    menuConnect.Checked = True
    menuDisconnect.Checked = False
    menuItem1.Enabled = True
    menuCapability.Enabled = True
    menuTriggerMode.Enabled = True
    menuTriggerMode.Checked = False
    menuItem2.Enabled = True
End Sub
Private Sub ReaderDisconnected()
    lblConnect.Text = "Disconnected"
    lblReaderMode.Text = ""
    lblConnect.ForeColor = Color.Red
    menuCommands.Enabled = False
    menuReaderMode.Enabled = False
    menuSetAntenna.Enabled = False
    menuReaderInfo.Enabled = False
    menuConnect.Enabled = True
    menuDisconnect.Enabled = False
    menuConnect.Checked = False
    menuDisconnect.Checked = True
    menuOnDemand.Checked = True
    menuOnDemand.Enabled = False
    menuAutoMode.Checked = False
    menuAutoMode.Enabled = True
    menuItem1.Enabled = False
    menuCapability.Enabled = False
    menuTriggerMode.Enabled = False
    menuTriggerMode.Checked = False
    menuItem2.Enabled = False
    '           menuMotionSensor.Text = "Enable MotionSensor";
    '           menuProxSensor.Text = "Enable ProximitySensor";
End Sub
Private Sub InitializeParamHashtable()
    htInParams = New Hashtable()
    htInParams.Add("TagID", Nothing)
    htInParams.Add("TagType", Nothing)
    htInParams.Add("WriteData", Nothing)
End Sub
Private Sub CreateFile(ByVal path As String)
    Try
        Dim wrXmlConfig As New XmlTextWriter(path, Encoding.UTF8)
        wrXmlConfig.WriteStartDocument()
        wrXmlConfig.WriteStartElement("ReaderConfig")
        wrXmlConfig.WriteAttributeString("xmlns", "xsi", Nothing, "http://www.w3.org/2001/XMLSchema-instance")
        wrXmlConfig.WriteStartElement("ComPortSettings")
        wrXmlConfig.WriteStartElement("COMPort")
        If m_ReaderModel = ReaderModel.MC9090 Then
            wrXmlConfig.WriteString("COM7")
        End If
        wrXmlConfig.WriteEndElement()
        wrXmlConfig.WriteStartElement("BaudRate")
        wrXmlConfig.WriteString("57600")
        wrXmlConfig.WriteEndElement()
        wrXmlConfig.WriteEndElement()
```

```
            wrXmlConfig.WriteStartElement("ReaderInfo")
            wrXmlConfig.WriteStartElement("Model")
            If m_ReaderModel = ReaderModel.MC9090 Then
                wrXmlConfig.WriteString("MC9090")
            End If
            wrXmlConfig.WriteEndElement()
            wrXmlConfig.WriteEndElement()
            wrXmlConfig.WriteEndElement()
            wrXmlConfig.WriteEndDocument()
            wrXmlConfig.Close()
        Catch ex As Exception
            MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
        End Try
    End Sub
    Private Sub UpdateListView(ByVal Tag As IRFIDTag)
        Dim i As Integer, len As Integer
        Try
            Dim strTagID As String = String.Empty
            Dim tag_ID As Byte() = Tag.TagID
            If tag_ID Is Nothing OrElse tag_ID.Length = 0 Then
                Return
            End If
            len = tag_ID.Length
            For i = 0 To len - 1
                strTagID += tag_ID(i).ToString("X2") & " "
            Next
            Dim lvItem As New ListViewItem(New String() {strTagID, Tag.TagType.ToString(),
Tag.LastSeen.ToString("HH:mm:ss"), Tag.AntennaName})
            listView1.Items.Insert(0, lvItem)
            lvItem.Focused = True
            lvItem.Selected = True
        Catch ex As Exception
            MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
        End Try
    End Sub
    Private Function GetTagID() As Byte()
        Try
            Dim chTagID As Char() = htInParams("TagID").ToString().ToCharArray()
            Dim tagIdByteArr As New ArrayList()
            For i As Integer = 0 To chTagID.Length - 2 Step 2
                Dim strTemp As New String(New Char() {chTagID(i), chTagID(i + 1)})
                Dim idByte As Byte = Convert.ToByte(strTemp, 16)
                tagIdByteArr.Add(idByte)
            Next
            Return DirectCast(tagIdByteArr.ToArray(GetType(Byte)), Byte())
        Catch ex As Exception
            MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
            Return Nothing
        End Try
    End Function
    Private Sub UpdateResponse(ByVal tags As IRFIDTag())
        Try
            Dim tagSN As Byte() = Nothing
            Dim tagIDStr As String = [String].Empty
            Dim tagDataStr As String = [String].Empty
            If tags Is Nothing OrElse tags.Length = 0 Then
                listView2.Items.Insert(0, New ListViewItem())
                listView2.Items.Insert(0, New ListViewItem("No Tags Read"))
                listView2.Items(0).ForeColor = Color.Red
                Return
            End If
            For Each newTag As IRFIDTag In tags
                tagDataStr = [String].Empty
                tagIDStr = [String].Empty
                tagSN = newTag.TagID
                For Each b As Byte In tagSN
                    tagIDStr += b.ToString("X2")
                Next
                listView2.Items.Insert(0, New ListViewItem())
```

```
            listView2.Items.Insert(0, New ListViewItem([String].Format("Antenna name:{0}", newTag.AntennaName)))
            listView2.Items(0).ForeColor = Color.Green
            listView2.Items.Insert(0, New ListViewItem([String].Format("Reader Name: {0}", newTag.ReaderName)))
            listView2.Items(0).ForeColor = Color.Green
            listView2.Items.Insert(0, New ListViewItem("Tag Type:" & newTag.TagType.ToString()))
            listView2.Items(0).ForeColor = Color.Green
            listView2.Items.Insert(0, New ListViewItem("Tag ID:" & tagIDStr))
            listView2.Items(0).ForeColor = Color.Green
            If newTag.TagData IsNot Nothing Then
                tagSN = newTag.TagData
                For Each b As Byte In tagSN
                    tagDataStr += b.ToString("X2")
                Next
                listView2.Items.Insert(0, New ListViewItem("Tag Data:" & tagDataStr))
                listView2.Items(0).ForeColor = Color.Green
            End If
        Next
        listView2.Items.Insert(0, New ListViewItem())
        If tags.Length = 1 Then
            listView2.Items.Insert(0, New ListViewItem(String.Format("{0} Tag Found", tags.Length)))
        Else
            listView2.Items.Insert(0, New ListViewItem(String.Format("{0} Tags Found", tags.Length)))
        End If
        listView2.Items(0).ForeColor = Color.Green
    Catch ex As Exception
        MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
    End Try
End Sub
Private Sub menuGetTags_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuGetTags.Click
    If (Not VerifyOnDemand()) Then
        Return
    End If
    SendCommand("GET TAGS")
End Sub
Private Sub SendCommand(ByVal cmdName As String)
    Dim statusColor As Color = Color.Green
    '        byte[] tagID = new byte[] { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    Dim tagID As Byte() = New Byte(127) {}
    Dim tagsRead As IRFIDTag() = Nothing
    If Not VerifyOnDemand() Then
        Return
    End If
    Dim executionStatus As String = "Success"
    Dim errMsg As String = ""
    Try
        Select Case cmdName
            Case "GET TAGS"

                If deviceReader.ReadMode = ReadMode.ONDEMAND Then
                    tagsRead = deviceReader.GetTags()
                    If tagsRead IsNot Nothing Then
                        executionStatus = "Success"
                    Else
                        executionStatus = "Failed: No Tags Read!"
                    End If
                Else
                    executionStatus = "Enter OnDemand First"
                End If
                Exit Select
            Case "GET ANTENNA NAMES"
                deviceReader.GetAntennaNames()
                Exit Select
            Case "ERASE TAGS (CLASS1GEN2)"
                deviceReader.EraseTag(TagType.EPCClass1_GEN2)
                executionStatus = "Success"
                Exit Select
            Case "ERASE TAGS (CLASS1)"
                deviceReader.EraseTag(TagType.EPCClass1)
                executionStatus = "Success"
```

```
                Exit Select
            Case "ERASE TAGS (CLASS0)"
                deviceReader.EraseTag(TagType.EPCClass0)
                executionStatus = "Success"
                Exit Select
            Case "ERASE TAGS (CLASS0+)"
                deviceReader.EraseTag(TagType.EPClass0_PLUS)
                executionStatus = "Success"
                Exit Select
            Case "KILL TAGS (CLASS0)"
                tagID = GetTagID()
                deviceReader.KillTag(TagType.EPCClass0, tagID)
                executionStatus = "Success"
                Exit Select
            Case "KILL TAGS (CLASS1)"
                tagID = GetTagID()
                deviceReader.KillTag(TagType.EPCClass1, tagID)
                executionStatus = "Success"
                Exit Select
            Case "KILL TAGS (CLASS0+)"
                tagID = GetTagID()
                deviceReader.KillTag(TagType.EPClass0_PLUS, tagID)
                executionStatus = "Success"
                Exit Select
            Case "KILL TAGS (CLASS1GEN2)"
                deviceReader.KillTag(TagType.EPCClass1_GEN2, tagID)
                executionStatus = "Success"
                Exit Select
            Case "LOCK TAGS (CLASS1GEN2)"
deviceReader.LockTag(htInParams("TagID").ToString(), New AntennaConfig())
                executionStatus = "Success"
                Exit Select
            Case "WRITE TAG ID (CLASS1)"
                tagID = GetTagID()
                deviceReader.WriteTagID(TagType.EPCClass1, tagID)
                executionStatus = "Success"
                Exit Select
            Case "WRITE TAG ID (CLASS0)"
                tagID = GetTagID()
                deviceReader.WriteTagID(TagType.EPCClass0, tagID)
                executionStatus = "Success"
                Exit Select
            Case "WRITE TAG ID (CLASS1GEN2)"
'###############################################
                tagID = GetTagID()
                deviceReader.WriteTagID(TagType.EPCClass1_GEN2, tagID)
                executionStatus = "Success"
                Exit Select
            Case "WRITE TAG ID (CLASS0+)"
                tagID = GetTagID()
deviceReader.WriteTagID(TagType.EPClass0_PLUS, tagID)
                executionStatus = "Success"
                Exit Select
            Case Else
                executionStatus = "Failed"
                statusColor = Color.Red
                Exit Select
        End Select
    Catch ex As Exception
        executionStatus = "Failed"
        statusColor = Color.Red
        errMsg = ex.Message
    Finally
        listView2.Items.Insert(0, New ListViewItem())
        If cmdName = "GET TAGS" AndAlso executionStatus = "Success" Then
            UpdateResponse(tagsRead)
        End If
        UpdateCommandListView(errMsg, executionStatus, cmdName, statusColor)
    End Try
```

```
    tabPage2.BringToFront()
    tabPage2.Focus()
    tabControl1.SelectedIndex = 0
End Sub
Private Sub menuAutoMode_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuAutoMode.Click
    Dim executionStatus As String = "Success"
    Dim errMsg As String = ""
    Dim statusColor As Color = Color.Green
    Try
        If deviceReader.ReaderStatus = ReaderStatus.ONLINE Then
            deviceReader.ReadMode = ReadMode.AUTONOMOUS
            menuAutoMode.Enabled = False
            menuAutoMode.Checked = True
            menuOnDemand.Checked = False
            menuOnDemand.Enabled = True
            menuTriggerMode.Checked = False
            menuTriggerMode.Enabled = True
            lblReaderMode.Text = "Autonomous"
        End If
        tabPage2.BringToFront()
        tabPage2.Focus()
        tabControl1.SelectedIndex = 1
    Catch ex As Exception
        executionStatus = "Failed"
        errMsg = ex.Message
        tabPage1.BringToFront()
        tabPage1.Focus()
        tabControl1.SelectedIndex = 0
    Finally
        UpdateCommandListView(errMsg, executionStatus, "SET AUTONOMOUS MODE", statusColor)
    End Try
End Sub
Private Sub MyTriggerH(ByVal sender As Object, ByVal evt As Symbol.ResourceCoordination.TriggerEventArgs)
    If deviceReader.ReaderStatus = ReaderStatus.ONLINE Then
        Dim executionStatus As String = "Success"
        Dim errMsg As String = ""
        Dim Mode As String = "AUTONOMOUS MODE (via Trigger)"
        Dim statusColor As Color = Color.Green
        Try
            If evt.NewState = Symbol.ResourceCoordination.TriggerState.STAGE2 Then
                deviceReader.ReadMode = Symbol.RFID2.ReadMode.AUTONOMOUS
                menuAutoMode.Enabled = True
                menuAutoMode.Checked = False
                menuOnDemand.Checked = False
                menuOnDemand.Enabled = True
                menuTriggerMode.Checked = True
                menuTriggerMode.Enabled = False
                lblReaderMode.Text = "Triggering"
                tabPage2.BringToFront()
                tabPage2.Focus()
                tabControl1.SelectedIndex = 1
            ElseIf evt.NewState = Symbol.ResourceCoordination.TriggerState.RELEASED Then
                Mode = "ONDEMAND"
                deviceReader.ReadMode = Symbol.RFID2.ReadMode.ONDEMAND
                menuAutoMode.Enabled = True
                menuAutoMode.Checked = False
                menuOnDemand.Checked = True
                menuOnDemand.Enabled = False
                menuTriggerMode.Checked = False
                menuTriggerMode.Enabled = True
                lblReaderMode.Text = "OnDemand"
            End If
        Catch ex As Exception
            executionStatus = "Failed"
            errMsg = ex.Message
            tabPage1.BringToFront()
            tabPage1.Focus()
            tabControl1.SelectedIndex = 0
        Finally
```

```
            UpdateCommandListView(errMsg, executionStatus, Mode.ToString(), statusColor)
        End Try
    End If
    'if
End Sub
Private Sub menuOnDemand_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuOnDemand.Click
    Dim executionStatus As String = "Success"
    Dim errMsg As String = ""
    Dim statusColor As Color = Color.Green
    Try
        If deviceReader.ReaderStatus = ReaderStatus.ONLINE Then
            deviceReader.ReadMode = ReadMode.ONDEMAND
            menuAutoMode.Enabled = True
            menuAutoMode.Checked = False
            menuOnDemand.Checked = True
            menuOnDemand.Enabled = False
            menuTriggerMode.Checked = False
            menuTriggerMode.Enabled = True
            lblReaderMode.Text = "OnDemand"
        End If
        tabPage1.BringToFront()
        tabPage1.Focus()
        tabControl1.SelectedIndex = 0
    Catch ex As Exception
        executionStatus = "Failed"
        errMsg = ex.Message
    Finally
        UpdateCommandListView(errMsg, executionStatus, "ON DEMAND MODE", statusColor)
    End Try
End Sub
Private Sub btnClear_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnClear.Click
    If tabControl1.SelectedIndex = 0 Then
        listView2.Items.Clear()
    ElseIf tabControl1.SelectedIndex = 1 Then
        listView1.Items.Clear()
    End If
End Sub
Private Sub menuConnect_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuConnect.Click
    Try
        deviceReader.Connect()
        If deviceReader.ReaderStatus = ReaderStatus.ONLINE Then
            ReaderConnected()
        End If
    Catch ex As Exception
        MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
    End Try
End Sub
Private Sub menuExit_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuExit.Click
    Try
        deviceReader.Disconnect()
        Application.[Exit]()
    Catch ex As Exception
        MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
    End Try
End Sub
Private Sub menuDisconnect_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuDisconnect.Click
    Try
        deviceReader.Disconnect()
        If deviceReader.ReaderStatus <> ReaderStatus.ONLINE Then
            ReaderDisconnected()
        End If
    Catch ex As Exception
        MessageBox.Show("Error:" & ex.Message, "CS_RFID2_Sample")
    End Try
End Sub
Private Sub menuCapability_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuCapability.Click
    Dim readerCapabilies As New FrmCapabilties()
    readerCapabilies.Capabilty = deviceReader.ReaderCapability
    If readerCapabilies.Capabilty IsNot Nothing Then
```

```
            readerCapabilies.ShowDialog()
        Else
            MessageBox.Show("Reader Capabilities not found")
        End If
    End Sub
    Private Sub menuReaderInfo_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuReaderInfo.Click
        Try
            Dim antInfo As New FrmReadAntennaInfo()
            Dim antcfg As AntennaConfig() = deviceReader.Antennas
            antInfo.GetAntenna = antcfg
            antInfo.ShowDialog()
        Catch ex As Exception
            MessageBox.Show(ex.Message, "CS_RFID2_Sample")
        End Try
    End Sub
    Private Sub menuProg_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuProg.Click
        If (Not VerifyOnDemand()) Then
            Return
        End If
        Dim setTAGID As New FrmSettings()
        setTAGID.currentCommand = FrmSettings.Commands.ProgramTag
        setTAGID.ShowDialog()
        If setTAGID.htInParamSet IsNot Nothing Then
            htInParams("TagID") = setTAGID.htInParamSet("TagID").ToString()
            htInParams("TagType") = setTAGID.htInParamSet("TagType").ToString()
            SendCommand("WRITE TAG ID (" & htInParams("TagType").ToString() & ")")
        End If
    End Sub
    Private Sub menuWrite_Click(ByVal sender As Object, ByVal e As EventArgs)
        If (Not VerifyOnDemand()) Then
            Return
        End If
        SendCommand("WRITE TAGS (CLASS1GEN2)")
    End Sub
    Private Sub MainForm_Closing(ByVal sender As Object, ByVal e As CancelEventArgs) Handles MyBase.Closing
        deviceReader.Disconnect()
        Application.[Exit]()
    End Sub
    Private Sub menuSetAntenna_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuSetAntenna.Click
        If (Not VerifyOnDemand()) Then
            Return
        End If
        Try
            Dim antConfig As AntennaConfig
            Dim arrAntennaConfig As AntennaConfig() = deviceReader.Antennas
            Dim f_setAntenna As New FrmSetAntenna(deviceReader.GetAntennaNames())
            f_setAntenna.SetAllAntenna = arrAntennaConfig
            f_setAntenna.m_txMax = deviceReader.MaxTxPower
            f_setAntenna.m_txMin = deviceReader.MinTxPower
            f_setAntenna.ShowDialog()
            If f_setAntenna.setAntennaConfig Then
                antConfig = f_setAntenna.SetAntenna
                deviceReader.SetAntennaConfiguration(antConfig)
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message, "CS_RFID2_Sample")
        End Try
    End Sub
    Private Sub menuLock_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuLocking.Click
        Try
            If (Not VerifyOnDemand()) Then
                Return
            End If
            Dim setTAGID As New FrmSettings()
            setTAGID.currentCommand = FrmSettings.Commands.LockTag
            setTAGID.ShowDialog()
            If setTAGID.htInParamSet IsNot Nothing Then
                htInParams("TagID") = setTAGID.htInParamSet("TagID").ToString()
                SendCommand("LOCK TAGS (CLASS1GEN2)")
```

```
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message, "CS_RFID2_Sample")
    End Try
End Sub
Private Sub menuKillTags_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuKillTags.Click
    If (Not VerifyOnDemand()) Then
        Return
    End If
    Dim setTAGID As New FrmSettings()
    setTAGID.currentCommand = FrmSettings.Commands.KillTag
    setTAGID.ShowDialog()
    If setTAGID.htInParamSet IsNot Nothing Then
        htInParams("TagID") = setTAGID.htInParamSet("TagID").ToString()
        htInParams("TagType") = setTAGID.htInParamSet("TagType").ToString()

        SendCommand("KILL TAGS (" & htInParams("TagType").ToString() & ")")
    End If
End Sub
Private Sub menuEraseTag_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuEraseTag.Click
    If (Not VerifyOnDemand()) Then
        Return
    End If
    Dim setTAGID As New FrmSettings()
    setTAGID.currentCommand = FrmSettings.Commands.EraseTag
    setTAGID.ShowDialog()
    If setTAGID.htInParamSet IsNot Nothing Then
        htInParams("TagType") = setTAGID.htInParamSet("TagType").ToString()
        SendCommand("ERASE TAGS (" & htInParams("TagType").ToString() & ")")
    End If
End Sub
Private Sub menuItem1_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuItem1.Click
    Dim htReaderInfo As New Hashtable()
    htReaderInfo.Add("DeviceSerialNumber", deviceReader.ReaderInfo.DeviceSerialNumber)
    htReaderInfo.Add("DeviceModelNumber", deviceReader.ReaderInfo.DeviceModelNumber)
    htReaderInfo.Add("ManufacturerName", deviceReader.ReaderInfo.ManufacturerName)
    htReaderInfo.Add("ManufactureDate", deviceReader.ReaderInfo.ManufactureDate)
    htReaderInfo.Add("HardwareVersion", deviceReader.ReaderInfo.HardwareVersion)
    htReaderInfo.Add("BootLoaderVersion", deviceReader.ReaderInfo.BootLoaderVersion)
    htReaderInfo.Add("FirmwareVersion", deviceReader.ReaderInfo.FirmwareVersion)
    htReaderInfo.Add("SymbolSDKVersion", deviceReader.SDKVersionNumber)
    htReaderInfo.Add("Model", deviceReader.Model.ToString())
    Dim reader As New FrmReaderInfo(htReaderInfo)
    reader.ShowDialog()
End Sub
Private Sub deviceReader_TriggerPressedEvent(ByVal sender As Object, ByVal args As ReaderEventArgs)
    Try
        lblTrig.Invoke(displayHandler, New Object() {"TriggerState", args})
    Catch ex As Exception
        MessageBox.Show("Error in Trigger event: " & ex.Message, "CS_RFID2_Sample")
    End Try
End Sub
Private Sub menuItem2_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuTriggerMode.Click
    Dim executionStatus As String = "Success"
    Dim errMsg As String = ""
    Dim statusColor As Color = Color.Green
    Try
        If deviceReader.ReaderStatus = ReaderStatus.ONLINE Then
            menuAutoMode.Enabled = True
            menuAutoMode.Checked = False
            menuOnDemand.Checked = False
            menuOnDemand.Enabled = True
            menuTriggerMode.Checked = True
            menuTriggerMode.Enabled = False
            lblReaderMode.Text = "Use Trigger"
            If deviceReader.ReadMode = ReadMode.AUTONOMOUS Then
                deviceReader.ReadMode = ReadMode.ONDEMAND
            End If
        End If
```

```
            tabPage2.BringToFront()
            tabPage2.Focus()
            tabControl1.SelectedIndex = 1
        Catch ex As Exception
            executionStatus = "Failed"
            errMsg = ex.Message
            tabPage1.BringToFront()
            tabPage1.Focus()
            tabControl1.SelectedIndex = 0
        Finally
            UpdateCommandListView(errMsg, executionStatus, "ONDEMAND MODE", statusColor)
        End Try
    End Sub
    Private Sub UpdateCommandListView(ByVal errMsg As String, ByVal executionStatus As String, ByVal command As
String, ByVal statusColor As Color)
        listView2.Items.Insert(0, New ListViewItem())
        listView2.Items.Insert(0, New ListViewItem(errMsg))
        listView2.Items(0).ForeColor = statusColor
        listView2.Items.Insert(0, New ListViewItem("Execution Status:" & executionStatus))
        listView2.Items(0).ForeColor = statusColor
        listView2.Items.Insert(0, New ListViewItem("Response:"))
        listView2.Items(0).ForeColor = statusColor
        listView2.Items.Insert(0, New ListViewItem("Command Sent:" & command))
        listView2.Items(0).ForeColor = Color.Blue
    End Sub
    Private Sub DisplayText(ByVal text As String, ByVal args As ReaderEventArgs)
        Try
            SyncLock displayTextLock
                If text = "Tags Read" Then
                    If args Is Nothing Then
                        Return
                    End If
                    Dim tagArgs As TagEventArgs = DirectCast(args, TagEventArgs)
                    Dim tags As IRFIDTag() = tagArgs.Tags
                    If tags Is Nothing OrElse tags.Length = 0 Then
                        Return
                    End If
                    For Each tag As IRFIDTag In tags
                        UpdateListView(tag)
                    Next
                End If
                If text = "TriggerState" Then
                    Dim trigArgs As Symbol.RFID2.TriggerEventArgs = DirectCast(args, Symbol.RFID2.TriggerEventArgs)
                    lblTrig.Text = trigArgs.TriggerState.ToString()
                End If
            End SyncLock
        Catch ex As Exception
            MessageBox.Show("DisplayText:" & ex.Message)
        End Try
    End Sub
    Private Sub menuItem2_Click_1(ByVal sender As Object, ByVal e As EventArgs) Handles menuItem2.Click
        Dim readerSettings As New ReaderSettings()
        'pzhu readerSettings.StartingQ = deviceReader.StartingQ;
        'pzhu deviceReader.StartingQ = readerSettings.StartingQ;
        If readerSettings.ShowDialog() = Windows.Forms.DialogResult.OK Then
        End If
    End Sub
    Private Sub menuReadData_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuReadData.Click
        Dim read As New FrmReadData(False)
        Dim errMsg As String = ""
        Dim executionStatus As String = "Success"
        Dim statusColor As Color = Color.Green
        Dim tagsRead As IRFIDTag() = Nothing
        If read.ShowDialog() = Windows.Forms.DialogResult.OK Then
            Try
                Dim dataLoc As New TagDataLoc(read.wordPointer, read.wordCount, CType(read.memBank, MemoryBank))
                tagsRead = deviceReader.GetTags(dataLoc, read.gen2Params, read.accessPassword)
            Catch ex As Exception
                errMsg = ex.Message
```

```
                statusColor = Color.Red
                executionStatus = "Failed"
            Finally
                UpdateResponse(tagsRead)
                UpdateCommandListView(errMsg, executionStatus, "Read Data", statusColor)
            End Try
        End If
    End Sub
    Private Sub menuWriteTag_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuWriteTag.Click
        Dim read As New FrmReadData(True)
        Dim errMsg As String = ""
        Dim executionStatus As String = "Success"
        Dim statusColor As Color = Color.Green
        If (Not VerifyOnDemand()) Then
            Return
        End If
        If read.ShowDialog() = Windows.Forms.DialogResult.OK Then
            Try
                Dim chTagID As Char() = read.writeData.ToCharArray()
                Dim tagIdByteArr As New ArrayList()
                For i As Integer = 0 To chTagID.Length - 2 Step 2
                    Dim strTemp As New String(New Char() {chTagID(i), chTagID(i + 1)})
                    Dim idByte As Byte = Convert.ToByte(strTemp, 16)
                    tagIdByteArr.Add(idByte)
                Next
                Dim wrData As Byte() = DirectCast(tagIdByteArr.ToArray(GetType(Byte)), Byte())
                Dim dataLoc As New TagDataLoc(read.wordPointer, CByte((wrData.Length \ 2)), CType(read.memBank,
MemoryBank))
                Dim tagsWritten As Boolean = deviceReader.WriteTag(read.gen2Params, dataLoc, read.accessPassword,
wrData)
                If Not tagsWritten Then
                    errMsg = "Could not write tags "
                    statusColor = Color.Red
                    executionStatus = "Failed"
                End If
            Catch ex As Exception
                errMsg = ex.Message
                statusColor = Color.Red
                executionStatus = "Failed"
            Finally
                UpdateCommandListView(errMsg, executionStatus, "Write Data", statusColor)
            End Try
        End If
    End Sub
    '######################## GEN2 READ ########################
    Private Sub menuGen2Read_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuGen2Read.Click
        If (Not VerifyOnDemand()) Then
            Return
        End If
        RemoveHandler MyTrigger.Stage2Notify, MyTriggerHandler
        Gen2ReadPage = New FrmGen2Read(deviceReader)
        If Gen2ReadPage.ShowDialog() = Windows.Forms.DialogResult.OK Then
        End If
        SetupTriggerResource()
    End Sub
    '######################## GEN2 WRITE ########################
    Private Sub menuGen2Write_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuGen2Write.Click
        If (Not VerifyOnDemand()) Then
            Return
        End If

        RemoveHandler MyTrigger.Stage2Notify, MyTriggerHandler

        Gen2WritePage = New FrmGen2Write(deviceReader)

        If Gen2WritePage.ShowDialog() = Windows.Forms.DialogResult.OK Then
        End If
        SetupTriggerResource()
    End Sub
```

```
    '######################## SELECT RECORD
    Private Sub menuGen2SelRec_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuGen2SelRec.Click
        If (Not VerifyOnDemand()) Then
            Return
        End If
        RemoveHandler MyTrigger.Stage2Notify, MyTriggerHandler
        SelRecPage = New FrmSelectRecord(deviceReader)
        If SelRecPage.ShowDialog() = Windows.Forms.DialogResult.OK Then
        End If
        SetupTriggerResource()
    End Sub
    '######################## GEN2 LOCK ########################
    Private Sub menuGen2Lock_Click(ByVal sender As Object, ByVal e As EventArgs) Handles menuGen2Lock.Click
        If (Not VerifyOnDemand()) Then
            Return
        End If

        RemoveHandler MyTrigger.Stage2Notify, MyTriggerHandler

        LockTagPage = New FrmGen2Lock(deviceReader)

        If LockTagPage.ShowDialog() = Windows.Forms.DialogResult.OK Then
        End If
        SetupTriggerResource()
    End Sub
        ' Establish the remote endpoint for the socket.
        Dim Bytes() As String = IpAddress.Split(".")
        Dim Address As New IPAddress(New Byte() {Bytes(0), Bytes(1), Bytes(2), Bytes(3)})
        Dim EndPoint As New IPEndPoint(Address, Port)
        ' Create a TCP/IP socket.
        Using Socket As New Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp)
            ' Encode the data string into a byte array.
            Dim Message = Encoding.ASCII.GetBytes(Text & EOF)
            ' Connect the socket to the remote endpoint and send the data.
            Socket.Connect(EndPoint)
            Socket.Send(Message)
            'Beep()
            Socket.Shutdown(SocketShutdown.Both)
        End Using
    End Sub
    Private Sub FitRewButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
FitRewButton.Click
        If FitRewButton.BackColor = Color.Red Then
            FitRewButton.BackColor = Color.GreenYellow
        Else
            FitRewButton.BackColor = Color.Red
        End If
    End Sub
    Private Sub WeldRewButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
WeldRewButton.Click
        If WeldRewButton.BackColor = Color.Red Then
            WeldRewButton.BackColor = Color.GreenYellow
        Else
            WeldRewButton.BackColor = Color.Red
        End If
    End Sub
End Class
```

Code for database real-time updating

```
Public Class ReceiverFrame
   Declare Function Beep Lib "kernel32.dll" (ByVal dwFreq As Integer, ByVal dwDuration As Integer) As Boolean
   Private Sub SocketListener_Receive(ByVal sender As Object, ByVal e As ReceiveEventArgs) Handles
SocketListener.Receive
      Dim MyRFIDtag As String = e.Content
      If RFID_Data_Class.UpdateRFID(e.Content) Then
         TextBox.AppendText(MyRFIDtag & "    Data Base Got Updated!")
         TextBox.AppendText(Environment.NewLine)
         Me.TextBox.BackColor = Color.GreenYellow
         TextBox.Refresh()
         Beep(3000, 1000)
         Me.Timer1.Start()
      Else
         TextBox.AppendText(MyRFIDtag & "    No Updating Done!")
         TextBox.AppendText(Environment.NewLine)
      End If
   End Sub
   Private Sub ReceiverFrame_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
      SocketListener.Listen()
   End Sub
   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
      Dim MyRFIDtag As String = TextBox1.Text
      If RFID_Data_Class.UpdateRFID(MyRFIDtag) Then
         TextBox.AppendText(MyRFIDtag & "    Data Base Got Updated!")
         TextBox.AppendText(Environment.NewLine)
         Me.TextBox.BackColor = Color.GreenYellow
         TextBox.Refresh()
         Beep(3000, 1000)
         Me.Timer1.Start()
      Else
         TextBox.AppendText(MyRFIDtag & "    No Updating Done!")
         TextBox.AppendText(Environment.NewLine)
      End If
   End Sub
   Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
      Me.TextBox.BackColor = Color.White
      Me.Timer1.Stop()
   End Sub
End Class


Imports System.Data.OleDb
Public Class RFID_Data_Class
   Public Shared Function UpdateRFID(ByVal RFID As String) As Boolean
      'Create a connection
      Dim MyAccessConn As OleDbConnection = New OleDbConnection("provider = Microsoft.jet.OLEDB.4.0;" &
"Data Source= ..\..\..\..\TimeSheet.mdb")
      'Create the Select Command
      Dim SelCommandText As String = "SELECT RFIDRead.RFIDTag, Max(RFIDRead.ReadTime) AS MaxTime,
DateDiff(" & Chr(34) & "s" & Chr(34) & " ,[MaxTime],Date() " & " & " & Chr(34) & " " & Chr(34) & _
                           " & " & " Time()) AS TimePassed FROM(RFIDRead) GROUP BY RFIDRead.RFIDTag HAVING
(((RFIDRead.RFIDTag)=" & Chr(34) & RFID & Chr(34) & ")); "
      'Create a data adapter
      Dim DA As OleDbDataAdapter = New OleDbDataAdapter(SelCommandText, MyAccessConn)
      ''Create a data adapter delete command
      DA.DeleteCommand = New OleDbCommand(SelCommandText, MyAccessConn)
      'Create the Insert Command
      Dim InsertCommandText As String = "INSERT INTO RFIDRead  (RFIDTag, ReadTime )SELECT " & Chr(34) &
RFID & Chr(34) & " AS RFID, Date() " & " & " & Chr(34) & " " & Chr(34) & " & " & " Time();"
      DA.InsertCommand = New OleDbCommand(InsertCommandText, MyAccessConn)
      'Create a data Set
      Dim DS As DataSet = New DataSet
      'Use it when want to apply update command
      Dim builder As OleDbCommandBuilder = New OleDbCommandBuilder(DA)
      Dim InsertRequired As Boolean = False
      Try
```

```
            MyAccessConn.Open()
            DA.Fill(DS, "RFIDRead")
            If DS.Tables(0).Rows.Count = 0 Then
                InsertRequired = True
            Else
                For Each MyRow As DataRow In DS.Tables(0).Rows
                    If MyRow("TimePassed") > 120 Then
                        InsertRequired = True
                    End If
                Next
            End If
            If InsertRequired Then
                DA.InsertCommand.ExecuteNonQuery()
            End If
            MyAccessConn.Close()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
        Return InsertRequired
    End Function
End Class
```

Data collected during case study in the fabrication shop

The interface developments and aforementioned coding enabled us to capture high quality data from the steel fabrication shop. Real-time high quality data provided with this data acquisition system established a sound foundation for improving decision making. Table B-1 represents actual data captured in the steel fabrication shop of the collaborating company.

Table B-1. Raw data captured using RFID technology.

RFIDRead					
RFIDTag	**ReadTime**	**Rework**	**JobNo**	**DivNo**	**PieceNo**
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/19/2010 2:10:52 PM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 36 F5 00 00	1/19/2010 3:02:26 PM	0	09502	50A	36
33 B2 F1 F0 95 02 F5 0A 37 F5 00 00	1/19/2010 3:05:25 PM	0	09502	50A	37
33 B2 F1 F0 95 02 F5 0A 24 F5 00 00	1/19/2010 3:42:02 PM	0	09502	50A	24
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/19/2010 4:00:32 PM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/19/2010 4:41:30 PM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 52 F5 00 00	1/19/2010 5:40:11 PM	0	09502	50A	52
33 B2 F1 F0 95 02 F5 0A 39 F5 00 00	1/19/2010 6:45:37 PM	0	09502	50A	39
33 B2 F1 F0 95 02 F5 0A 52 F5 00 00	1/19/2010 6:55:05 PM	0	09502	50A	52
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/19/2010 7:33:31 PM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 24 F5 00 00	1/19/2010 7:50:02 PM	0	09502	50A	24
33 B2 F1 F0 95 02 F5 0A 36 F5 00 00	1/19/2010 8:01:46 PM	0	09502	50A	36

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 37 F5 00 00	1/19/2010 9:00:05 PM	0	09502	50A	37
33 B2 F1 F0 95 02 F5 0A 31 F0 00 00	1/19/2010 9:01:44 PM	0	09502	50A	31
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/19/2010 9:12:26 PM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 39 F5 00 00	1/19/2010 9:15:07 PM	0	09502	50A	39
33 B2 F1 F0 95 02 F5 0A 46 F0 00 00	1/19/2010 10:00:23 PM	0	09502	50A	46
33 B2 F1 F0 95 02 F5 0A 25 F0 00 00	1/19/2010 10:03:16 PM	0	09502	50A	25
33 B2 F1 F0 95 02 F5 0A 25 F0 00 00	1/19/2010 10:05:16 PM	0	09502	50A	25
33 B2 F1 F0 95 02 F5 0A 10 2F 00 00	1/19/2010 11:34:12 PM	0	09502	50A	102
33 B2 F1 F0 95 02 F5 0A 25 F0 00 00	1/19/2010 11:53:55 PM	0	09502	50A	25
33 B2 F1 F0 95 02 F5 0A 26 F0 00 00	1/19/2010 11:58:26 PM	0	09502	50A	26
33 B2 F1 F0 95 02 F5 0A 32 F0 00 00	1/19/2010 11:58:42 PM	0	09502	50A	32
33 B2 F1 F0 95 02 F5 0A 43 F0 00 00	1/20/2010 12:01:15 AM	0	09502	50A	43
33 B2 F1 F0 95 02 F5 0A 32 F0 00 00	1/20/2010 12:02:48 AM	0	09502	50A	32
33 B2 F1 F0 95 02 F5 0A 26 F0 00 00	1/20/2010 12:02:53 AM	0	09502	50A	26
33 B2 F1 F0 95 02 F5 0A 26 F0 00 00	1/20/2010 12:05:02 AM	0	09502	50A	26
33 B2 F1 F0 95 02 F5 0A 26 F0 00 00	1/20/2010 12:05:04 AM	0	09502	50A	26
33 B2 F1 F0 95 02 F5 0A 32 F0 00 00	1/20/2010 12:05:09 AM	0	09502	50A	32
33 B2 F1 F0 95 02 F5 0A 32 F0 00 00	1/20/2010 12:06:09 AM	0	09502	50A	32
33 B2 F1 F0 95 02 F5 0A 29 F0 00 00	1/20/2010 12:07:35 AM	0	09502	50A	29
33 B2 F1 F0 95 02 F5 0A 29 F0 00 00	1/20/2010 12:07:36 AM	0	09502	50A	29
33 B2 F1 F0 95 02 F5 0A 29 F0 00 00	1/20/2010 12:07:37 AM	0	09502	50A	29
33 B2 F1 F0 95 02 F5 0A 29 F0 00 00	1/20/2010 12:07:38 AM	0	09502	50A	29
33 B2 F1 F0 95 02 F5 0A 31 F0 00 00	1/20/2010 12:23:41 AM	0	09502	50A	31
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/20/2010 12:30:06 AM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 46 F0 00 00	1/20/2010 12:58:30 AM	0	09502	50A	46
33 B2 F1 F0 95 02 F5 0A 34 F0 00 00	1/20/2010 1:02:16 AM	0	09502	50A	34
33 B2 F1 F0 95 02 F5 0A 43 F0 00 00	1/20/2010 1:06:15 AM	0	09502	50A	43
33 B2 F1 F0 95 02 F5 0A 44 F0 00 00	1/20/2010 1:08:25 AM	0	09502	50A	44
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/20/2010 1:54:45 AM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 23 F0 00 00	1/20/2010 1:56:10 AM	0	09502	50A	23
33 B2 F1 F0 95 02 F5 0A 10 1F 00 00	1/20/2010 2:10:52 AM	0	09502	50A	101
33 B2 F1 F0 95 02 F5 0A 47 F0 00 00	1/20/2010 2:12:16 AM	0	09502	50A	47
33 B2 F1 F0 95 02 F5 0A 48 F0 00 00	1/20/2010 2:22:27 AM	0	09502	50A	48
33 B2 F1 F0 95 02 F5 0A 44 F0 00 00	1/20/2010 2:50:09 AM	0	09502	50A	44
33 B2 F1 F0 95 02 F5 0A 51 F0 00 00	1/20/2010 6:05:31 AM	0	09502	50A	51
33 B2 F1 F0 95 02 F5 0A 59 F0 00 00	1/20/2010 6:16:16 AM	0	09502	50A	59
33 B2 F1 F0 95 02 F5 0A 58 F0 00 00	1/20/2010 6:16:17 AM	0	09502	50A	58
33 B2 F1 F0 95 02 F5 0A 47 F0 00 00	1/20/2010 6:22:16 AM	0	09502	50A	47
33 B2 F1 F0 95 02 F5 0A 33 F0 00 00	1/20/2010 6:29:20 AM	0	09502	50A	33
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/20/2010 6:32:51 AM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 39 F5 00 00	1/20/2010 6:34:20 AM	0	09502	50A	39
33 B2 F1 F0 95 02 F5 0A 52 F5 00 00	1/20/2010 6:34:43 AM	0	09502	50A	52
33 B2 F1 F0 95 02 F5 0A 38 F0 00 00	1/20/2010 6:43:54 AM	0	09502	50A	38
33 B2 F1 F0 95 02 F5 0A 48 F0 00 00	1/20/2010 6:45:19 AM	0	09502	50A	48
33 B2 F1 F0 95 02 F5 0A 74 F0 00 00	1/20/2010 6:59:11 AM	0	09502	50A	74

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 41 F0 00 00	1/20/2010 7:22:17 AM	0	09502	50A	41
33 B2 F1 F0 95 02 F5 0A 51 F0 00 00	1/20/2010 7:35:52 AM	0	09502	50A	51
33 B2 F1 F0 95 02 F5 0A 25 F0 00 00	1/20/2010 7:45:52 AM	0	09502	50A	25
33 B2 F1 F0 95 02 F5 0A 59 F0 00 00	1/20/2010 7:52:11 AM	0	09502	50A	59
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/20/2010 7:58:25 AM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 35 F0 00 00	1/20/2010 8:08:05 AM	0	09502	50A	35
33 B2 F1 F0 95 02 F5 0A 61 F0 00 00	1/20/2010 8:27:15 AM	0	09502	50A	61
33 B2 F1 F0 95 02 F5 0A 24 F5 00 00	1/20/2010 8:34:08 AM	0	09502	50A	24
33 B2 F1 F0 95 02 F5 0A 10 1F 00 00	1/20/2010 8:41:02 AM	0	09502	50A	101
33 B2 F1 F0 95 02 F5 0A 42 F0 00 00	1/20/2010 8:45:12 AM	0	09502	50A	42
33 B2 F1 F0 95 02 F5 0A 10 1F 00 00	1/20/2010 8:46:36 AM	0	09502	50A	101
33 B2 F1 F0 95 02 F5 0A 51 F0 00 00	1/20/2010 8:55:31 AM	0	09502	50A	51
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/20/2010 9:09:57 AM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 72 F0 00 00	1/20/2010 9:09:57 AM	0	09502	50A	72
33 B2 F1 F0 95 02 F5 0A 35 F0 00 00	1/20/2010 9:22:39 AM	0	09502	50A	35
33 B2 F1 F0 95 02 F5 0A 37 F5 00 00	1/20/2010 9:28:35 AM	0	09502	50A	37
33 B2 F1 F0 95 02 F5 0A 41 F0 00 00	1/20/2010 9:30:37 AM	0	09502	50A	41
33 B2 F1 F0 95 02 F5 0A 49 F0 00 00	1/20/2010 9:33:03 AM	0	09502	50A	49
33 B2 F1 F0 95 02 F5 0A 42 F0 00 00	1/20/2010 9:34:33 AM	0	09502	50A	42
33 B2 F1 F0 95 02 F5 0A 74 F0 00 00	1/20/2010 9:35:47 AM	0	09502	50A	74
33 B2 F1 F0 95 02 F5 0A 63 F0 00 00	1/20/2010 9:39:26 AM	0	09502	50A	63
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/20/2010 9:42:43 AM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 31 F0 00 00	1/20/2010 9:43:27 AM	0	09502	50A	31
33 B2 F1 F0 95 02 F5 0A 23 F0 00 00	1/20/2010 9:45:35 AM	0	09502	50A	23
33 B2 F1 F0 95 02 F5 0A 37 F5 00 00	1/20/2010 9:55:46 AM	0	09502	50A	37
33 B2 F1 F0 95 02 F5 0A 36 F5 00 00	1/20/2010 9:56:18 AM	0	09502	50A	36
33 B2 F1 F0 95 02 F5 0A 10 1F 00 00	1/20/2010 9:56:26 AM	0	09502	50A	101
33 B2 F1 F0 95 02 F5 0A 51 F0 00 00	1/20/2010 9:59:35 AM	0	09502	50A	51
33 B2 F1 F0 95 02 F5 0A 63 F0 00 00	1/20/2010 10:02:28 AM	0	09502	50A	63
33 B2 F1 F0 95 02 F5 0A 27 F0 00 00	1/20/2010 10:04:45 AM	0	09502	50A	27
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/20/2010 10:25:41 AM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 41 F0 00 00	1/20/2010 10:32:37 AM	0	09502	50A	41
33 B2 F1 F0 95 02 F5 0A 36 F5 00 00	1/20/2010 10:38:04 AM	0	09502	50A	36
33 B2 F1 F0 95 02 F5 0A 23 F0 00 00	1/20/2010 11:09:04 AM	0	09502	50A	23
33 B2 F1 F0 95 02 F5 0A 31 F0 00 00	1/20/2010 11:15:22 AM	0	09502	50A	31
33 B2 F1 F0 95 02 F5 0A 23 F0 00 00	1/20/2010 11:22:54 AM	0	09502	50A	23
33 B2 F1 F0 95 02 F5 0A 33 F0 00 00	1/20/2010 11:23:46 AM	0	09502	50A	33
33 B2 F1 F0 95 02 F5 0A 60 F0 00 00	1/20/2010 11:33:59 AM	0	09502	50A	60
33 B2 F1 F0 95 02 F5 0A 41 F0 00 00	1/20/2010 11:37:25 AM	0	09502	50A	41
33 B2 F1 F0 95 02 F5 0A 60 F0 00 00	1/20/2010 11:51:35 AM	0	09502	50A	60
33 B2 F1 F0 95 02 F5 0A 72 F0 00 00	1/20/2010 11:52:39 AM	0	09502	50A	72
33 B2 F1 F0 95 02 F5 0A 75 F0 00 00	1/20/2010 11:58:59 AM	0	09502	50A	75
33 B2 F1 F0 95 02 F5 0A 27 F0 00 00	1/20/2010 11:59:41 AM	1	09502	50A	27
33 B2 F1 F0 95 02 F5 0A 55 F0 00 00	1/20/2010 12:14:18 PM	0	09502	50A	55
33 B2 F1 F0 95 02 F5 0A 39 F5 00 00	1/20/2010 12:18:42 PM	0	09502	50A	39

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 34 F0 00 00	1/20/2010 12:33:23 PM	0	09502	50A	34
33 B2 F1 F0 95 02 F5 0A 10 2F 00 00	1/20/2010 12:40:54 PM	0	09502	50A	102
33 B2 F1 F0 95 02 F5 0A 49 F0 00 00	1/20/2010 12:54:00 PM	0	09502	50A	49
33 B2 F1 F0 95 02 F5 0A 24 F5 00 00	1/20/2010 12:54:54 PM	0	09502	50A	24
33 B2 F1 F0 95 02 F5 0A 48 F0 00 00	1/20/2010 1:13:59 PM	0	09502	50A	48
33 B2 F1 F0 95 02 F5 0A 49 F0 00 00	1/20/2010 1:14:16 PM	0	09502	50A	49
33 B2 F1 F0 95 02 F5 0A 59 F0 00 00	1/20/2010 1:17:07 PM	0	09502	50A	59
33 B2 F1 F0 95 02 F5 0A 47 F0 00 00	1/20/2010 1:30:05 PM	0	09502	50A	47
33 B2 F1 F0 95 02 F5 0A 34 F0 00 00	1/20/2010 1:55:28 PM	0	09502	50A	34
33 B2 F1 F0 95 02 F5 0A 75 F0 00 00	1/20/2010 1:56:57 PM	0	09502	50A	75
33 B2 F1 F0 95 02 F5 0A 10 2F 00 00	1/20/2010 2:01:37 PM	0	09502	50A	102
33 B2 F1 F0 95 02 F5 0A 38 F0 00 00	1/20/2010 2:19:31 PM	0	09502	50A	38
33 B2 F1 F0 95 02 F5 0A 73 F0 00 00	1/20/2010 2:35:40 PM	0	09502	50A	73
33 B2 F1 F0 95 02 F5 0A 38 F0 00 00	1/20/2010 3:17:49 PM	0	09502	50A	38
33 B2 F1 F0 95 02 F5 0A 47 F0 00 00	1/20/2010 3:17:49 PM	0	09502	50A	47
33 B2 F1 F0 95 02 F5 0A 28 F0 00 00	1/20/2010 3:18:49 PM	0	09502	50A	28
33 B2 F1 F0 95 02 F5 0A 38 F0 00 00	1/20/2010 3:24:02 PM	0	09502	50A	38
33 B2 F1 F0 95 02 F5 0A 48 F0 00 00	1/20/2010 3:24:02 PM	0	09502	50A	48
33 B2 F1 F0 95 02 F5 0A 49 F0 00 00	1/20/2010 3:26:30 PM	0	09502	50A	49
33 B2 F1 F0 95 02 F5 0A 10 3F 00 00	1/20/2010 3:28:40 PM	0	09502	50A	103
33 B2 F1 F0 95 02 F5 0A 73 F0 00 00	1/20/2010 3:29:23 PM	0	09502	50A	73
33 B2 F1 F0 95 02 F5 0A 42 F0 00 00	1/20/2010 3:42:34 PM	0	09502	50A	42
33 B2 F1 F0 95 02 F5 0A 44 F0 00 00	1/20/2010 3:43:02 PM	0	09502	50A	44
33 B2 F1 F0 95 02 F5 0A 46 F0 00 00	1/20/2010 3:43:50 PM	0	09502	50A	46
33 B2 F1 F0 95 02 F5 0A 54 F0 00 00	1/20/2010 5:18:39 PM	0	09502	50A	54
33 B2 F1 F0 95 02 F5 0A 10 3F 00 00	1/20/2010 5:19:26 PM	0	09502	50A	103
33 B2 F1 F0 95 02 F5 0A 40 01 F0 00	1/20/2010 5:22:14 PM	0	09502	50A	4001
33 B2 F1 F0 95 02 F5 0A 40 02 F0 00	1/20/2010 5:22:34 PM	0	09502	50A	4002
33 B2 F1 F0 95 02 F5 0A 55 F0 00 00	1/20/2010 5:29:48 PM	0	09502	50A	55
33 B2 F1 F0 95 02 F5 0A 61 F0 00 00	1/20/2010 5:31:43 PM	0	09502	50A	61
33 B2 F1 F0 95 02 F5 0A 57 F0 00 00	1/20/2010 5:33:24 PM	0	09502	50A	57
33 B2 F1 F0 95 02 F5 0A 58 F0 00 00	1/20/2010 5:38:01 PM	0	09502	50A	58
33 B2 F1 F0 95 02 F5 0A 55 F0 00 00	1/20/2010 5:39:04 PM	0	09502	50A	55
33 B2 F1 F0 95 02 F5 0A 50 F0 00 00	1/20/2010 5:40:57 PM	0	09502	50A	50
33 B2 F1 F0 95 02 F5 0A 54 F0 00 00	1/20/2010 5:42:10 PM	0	09502	50A	54
33 B2 F1 F0 95 02 F5 0A 53 F0 00 00	1/20/2010 5:59:10 PM	0	09502	50A	53
33 B2 F1 F0 95 02 F5 0A 50 F0 00 00	1/20/2010 6:00:03 PM	0	09502	50A	50
33 B2 F1 F0 95 02 F5 0A 55 F0 00 00	1/20/2010 6:24:21 PM	0	09502	50A	55
33 B2 F1 F0 95 02 F5 0A 60 F0 00 00	1/20/2010 6:24:42 PM	0	09502	50A	60
33 B2 F1 F0 95 02 F5 0A 60 F0 00 00	1/20/2010 6:52:07 PM	0	09502	50A	60
33 B2 F1 F0 95 02 F5 0A 58 F0 00 00	1/20/2010 7:26:28 PM	0	09502	50A	58
33 B2 F1 F0 95 02 F5 0A 42 F0 00 00	1/20/2010 8:10:36 PM	0	09502	50A	42
33 B2 F1 F0 95 02 F5 0A 43 F0 00 00	1/20/2010 8:11:06 PM	0	09502	50A	43
33 B2 F1 F0 95 02 F5 0A 40 03 F0 00	1/20/2010 8:13:42 PM	0	09502	50A	4003
33 B2 F1 F0 95 02 F5 0A 64 F0 00 00	1/20/2010 10:16:17 PM	0	09502	50A	64

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 43 F0 00 00	1/20/2010 10:21:28 PM	0	09502	50A	43
33 B2 F1 F0 95 02 F5 0A 44 F0 00 00	1/20/2010 10:22:37 PM	0	09502	50A	44
33 B2 F1 F0 95 02 F5 0A 46 F0 00 00	1/20/2010 10:44:11 PM	0	09502	50A	46
33 B2 F1 F0 95 02 F5 0A 28 F0 00 00	1/20/2010 11:19:51 PM	0	09502	50A	28
33 B2 F1 F0 95 02 F5 0A 57 F0 00 00	1/20/2010 11:21:27 PM	0	09502	50A	57
33 B2 F1 F0 95 02 F5 0A 64 F0 00 00	1/20/2010 11:23:36 PM	0	09502	50A	64
33 B2 F1 F0 95 02 F5 0A 46 F0 00 00	1/20/2010 11:49:07 PM	0	09502	50A	46
33 B2 F1 F0 95 02 F5 0A 10 3F 00 00	1/20/2010 11:53:31 PM	0	09502	50A	103
33 B2 F1 F0 95 02 F5 0A 10 3F 00 00	1/21/2010 12:02:12 AM	0	09502	50A	103
33 B2 F1 F0 95 02 F5 0A 56 F0 00 00	1/21/2010 12:31:59 AM	0	09502	50A	56
33 B2 F1 F0 95 02 F5 0A 56 F0 00 00	1/21/2010 12:43:59 AM	0	09502	50A	56
33 B2 F1 F0 95 02 F5 0A 40 02 F0 00	1/21/2010 12:55:09 AM	0	09502	50A	4002
33 B2 F1 F0 95 02 F5 0A 56 F0 00 00	1/21/2010 12:55:59 AM	0	09502	50A	56
33 B2 F1 F0 95 02 F5 0A 40 01 F0 00	1/21/2010 12:56:03 AM	0	09502	50A	4001
33 B2 F1 F0 95 02 F5 0A 56 F0 00 00	1/21/2010 12:56:09 AM	0	09502	50A	56
33 B2 F1 F0 95 02 F5 0A 27 F0 00 00	1/21/2010 1:50:50 AM	0	09502	50A	27
33 B2 F1 F0 95 02 F5 0A 27 F0 00 00	1/21/2010 2:50:50 AM	0	09502	50A	27
33 B2 F1 F0 95 02 F5 0A 62 F0 00 00	1/21/2010 5:40:43 AM	0	09502	50A	62
33 B2 F1 F0 95 02 F5 0A 16 F0 00 00	1/21/2010 6:06:42 AM	0	09502	50A	16
33 B2 F1 F0 95 02 F5 0A 18 F0 00 00	1/21/2010 6:06:45 AM	0	09502	50A	18
33 B2 F1 F0 95 02 F5 0A 17 F0 00 00	1/21/2010 6:06:52 AM	0	09502	50A	17
33 B2 F1 F0 95 02 F5 0A 15 F0 00 00	1/21/2010 6:06:56 AM	0	09502	50A	15
33 B2 F1 F0 95 02 F5 0A 19 F0 00 00	1/21/2010 6:06:58 AM	0	09502	50A	19
33 B2 F1 F0 95 02 F5 0A 20 F0 00 00	1/21/2010 6:07:14 AM	0	09502	50A	20
33 B2 F1 F0 95 02 F5 0A 22 F0 00 00	1/21/2010 6:07:26 AM	0	09502	50A	22
33 B2 F1 F0 95 02 F5 0A 74 F0 00 00	1/21/2010 6:13:03 AM	0	09502	50A	74
33 B2 F1 F0 95 02 F5 0A 40 02 F0 00	1/21/2010 6:14:43 AM	0	09502	50A	4002
33 B2 F1 F0 95 02 F5 0A 40 01 F0 00	1/21/2010 6:15:21 AM	0	09502	50A	4001
33 B2 F1 F0 95 02 F5 0A 62 F0 00 00	1/21/2010 6:20:43 AM	0	09502	50A	62
33 B2 F1 F0 95 02 F5 0A 74 F0 00 00	1/21/2010 7:39:55 AM	0	09502	50A	74
33 B2 F1 F0 95 02 F5 0A 53 F0 00 00	1/21/2010 7:40:18 AM	0	09502	50A	53
33 B2 F1 F0 95 02 F5 0A 19 F0 00 00	1/21/2010 8:06:58 AM	0	09502	50A	19
33 B2 F1 F0 95 02 F5 0A 50 F0 00 00	1/21/2010 8:23:43 AM	0	09502	50A	50
33 B2 F1 F0 95 02 F5 0A 62 F0 00 00	1/21/2010 8:26:04 AM	0	09502	50A	62
33 B2 F1 F0 95 02 F5 0A 28 F0 00 00	1/21/2010 8:38:56 AM	0	09502	50A	28
33 B2 F1 F0 95 02 F5 0A 54 F0 00 00	1/21/2010 8:54:27 AM	0	09502	50A	54
33 B2 F1 F0 95 02 F5 0A 50 F0 00 00	1/21/2010 8:55:00 AM	0	09502	50A	50
33 B2 F1 F0 95 02 F5 0A 53 F0 00 00	1/21/2010 8:55:23 AM	0	09502	50A	53
33 B2 F1 F0 95 02 F5 0A 59 F0 00 00	1/21/2010 8:56:46 AM	0	09502	50A	59
33 B2 F1 F0 95 02 F5 0A 62 F0 00 00	1/21/2010 8:58:40 AM	0	09502	50A	62
33 B2 F1 F0 95 02 F5 0A 54 F0 00 00	1/21/2010 9:13:26 AM	0	09502	50A	54
33 B2 F1 F0 95 02 F5 0A 73 F0 00 00	1/21/2010 9:24:12 AM	0	09502	50A	73
33 B2 F1 F0 95 02 F5 0A 61 F0 00 00	1/21/2010 9:26:17 AM	0	09502	50A	61
33 B2 F1 F0 95 02 F5 0A 59 F0 00 00	1/21/2010 10:17:27 AM	0	09502	50A	59
33 B2 F1 F0 95 02 F5 0A 61 F0 00 00	1/21/2010 10:19:16 AM	0	09502	50A	61

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 72 F0 00 00	1/21/2010 10:27:11 AM	0	09502	50A	72
33 B2 F1 F0 95 02 F5 0A 73 F0 00 00	1/21/2010 10:27:32 AM	0	09502	50A	73
33 B2 F1 F0 95 02 F5 0A 40 03 F0 00	1/21/2010 11:12:26 AM	0	09502	50A	4003
33 B2 F1 F0 95 02 F5 0A 63 F0 00 00	1/21/2010 11:26:29 AM	0	09502	50A	63
33 B2 F1 F0 95 02 F5 0A 28 F0 00 00	1/21/2010 11:35:56 AM	0	09502	50A	28
33 B2 F1 F0 95 02 F5 0A 63 F0 00 00	1/21/2010 11:36:09 AM	0	09502	50A	63
33 B2 F1 F0 95 02 F5 0A 64 F0 00 00	1/21/2010 11:37:44 AM	0	09502	50A	64
33 B2 F1 F0 95 02 F5 0A 72 F0 00 00	1/21/2010 11:46:45 AM	0	09502	50A	72
33 B2 F1 F0 95 02 F5 0A 16 F0 00 00	1/21/2010 12:08:56 PM	1	09502	50A	16
33 B2 F1 F0 95 02 F5 0A 57 F0 00 00	1/21/2010 2:21:39 PM	0	09502	50A	57
33 B2 F1 F0 95 02 F5 0A 64 F0 00 00	1/21/2010 2:21:52 PM	0	09502	50A	64
33 B2 F1 F0 95 02 F5 0A 40 03 F0 00	1/21/2010 2:48:20 PM	0	09502	50A	4003
33 B2 F1 F0 95 02 F5 0A 58 F0 00 00	1/21/2010 3:07:52 PM	0	09502	50A	58
33 B2 F1 F0 95 02 F5 0A 57 F0 00 00	1/21/2010 3:08:00 PM	0	09502	50A	57
33 B2 F1 F0 95 02 F5 0A 75 F0 00 00	1/21/2010 6:28:11 PM	0	09502	50A	75
33 B2 F1 F0 95 02 F5 0A 21 F0 00 00	1/21/2010 6:55:10 PM	0	09502	50A	21
33 B2 F1 F0 95 02 F5 0A 17 F0 00 00	1/21/2010 8:37:25 PM	0	09502	50A	17
33 B2 F1 F0 95 02 F5 0A 18 F0 00 00	1/21/2010 8:37:31 PM	0	09502	50A	18
33 B2 F1 F0 95 02 F5 0A 21 F0 00 00	1/21/2010 9:05:22 PM	0	09502	50A	21
F1 F0 95 02 F1 A4 00 0F 00 00 00 00	1/25/2010 7:34:57 AM	0	09502	1A	4000
33 B2 F1 F0 95 02 F5 0A 15 F0 00 00	1/25/2010 7:52:25 AM	0	09502	50A	15
F1 F0 95 02 F1 A4 00 8F 00 00 00 00	1/25/2010 8:25:24 AM	0	09502	1A	4008
33 B2 F1 F0 95 02 F5 0A 22 F0 00 00	1/25/2010 8:29:16 AM	0	09502	50A	22
33 B2 F1 F0 95 02 F5 0A 11 F0 00 00	1/25/2010 9:34:39 AM	0	09502	50A	11
33 B2 F1 F0 95 02 F5 0A 20 F0 00 00	1/25/2010 9:40:03 AM	0	09502	50A	20
33 B2 F1 F0 95 02 F5 0A 66 F0 00 00	1/25/2010 9:59:33 AM	0	09502	50A	66
33 B2 F1 F0 95 02 F5 0A 69 F0 00 00	1/25/2010 9:59:34 AM	0	09502	50A	69
F1 F0 95 02 F1 A4 00 1F 00 00 00 00	1/25/2010 10:02:18 AM	0	09502	1A	4001
F1 F0 95 02 F1 A4 00 0F 00 00 00 00	1/25/2010 10:02:19 AM	0	09502	1A	4000
33 B2 F1 F0 95 02 F5 0A 10 4F 00 00	1/25/2010 10:05:28 AM	0	09502	50A	104
F1 F1 03 21 F1 A4 00 2F 00 00 00 00	1/25/2010 11:31:37 AM	0	10321	1A	4002
33 B2 F1 F0 95 02 F5 0A 10 4F 00 00	1/25/2010 11:57:14 AM	0	09502	50A	104
33 B2 F1 F0 95 02 F5 0A 69 F0 00 00	1/25/2010 11:57:18 AM	0	09502	50A	69
33 B2 F1 F0 95 02 F5 0A 11 F0 00 00	1/25/2010 12:00:06 PM	0	09502	50A	11
33 B2 F1 F0 95 02 F5 0A 10 8F 00 00	1/25/2010 12:08:25 PM	0	09502	50A	108
33 B2 F1 F0 95 02 F5 0A 10 8F 00 00	1/25/2010 12:25:31 PM	0	09502	50A	108
33 B2 F1 F0 95 02 F5 0A 11 4F 00 00	1/25/2010 12:28:31 PM	0	09502	50A	114
33 B2 F1 F0 95 02 F5 0A 19 F0 00 00	1/25/2010 12:43:45 PM	0	09502	50A	19
F1 F0 95 02 F1 A4 00 1F 00 00 00 00	1/25/2010 1:02:18 PM	0	09502	1A	4001
33 B2 F1 F0 95 02 F5 0A 65 F0 00 00	1/25/2010 1:13:30 PM	0	09502	50A	65
33 B2 F1 F0 95 02 F5 0A 66 F0 00 00	1/25/2010 1:20:08 PM	0	09502	50A	66
33 B2 F1 F0 95 02 F5 0A 65 F0 00 00	1/25/2010 2:08:40 PM	0	09502	50A	65
33 B2 F1 F0 95 02 F5 0A 19 F0 00 00	1/25/2010 2:19:03 PM	0	09502	50A	19
33 B2 F1 F0 95 02 F5 0A 17 F0 00 00	1/25/2010 2:19:09 PM	0	09502	50A	17
F1 F1 03 21 F1 A4 00 3F 00 00 00 00	1/25/2010 2:31:12 PM	0	10321	1A	4003

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 66 F0 00 00	1/25/2010 2:33:58 PM	0	09502	50A	66
F1 F1 03 21 F1 A4 00 2F 00 00 00 00	1/25/2010 2:35:30 PM	0	10321	1A	4002
33 B2 F1 F0 95 02 F5 0A 21 F0 00 00	1/25/2010 2:57:03 PM	0	09502	50A	21
33 B2 F1 F0 95 02 F5 0A 17 F0 00 00	1/25/2010 2:58:32 PM	0	09502	50A	17
F1 F0 95 02 F1 A4 00 3F 00 00 00 00	1/25/2010 3:02:24 PM	0	09502	1A	4003
33 B2 F1 F0 95 02 F5 0A 16 F0 00 00	1/25/2010 3:26:49 PM	0	09502	50A	16
33 B2 F1 F0 95 02 F5 0A 18 F0 00 00	1/25/2010 3:40:30 PM	0	09502	50A	18
33 B2 F1 F0 95 02 F5 0A 21 F0 00 00	1/25/2010 3:40:30 PM	0	09502	50A	21
33 B2 F1 F0 95 02 F5 0A 15 F0 00 00	1/25/2010 3:49:17 PM	0	09502	50A	15
33 B2 F1 F0 95 02 F5 0A 18 F0 00 00	1/25/2010 6:36:55 PM	0	09502	50A	18
33 B2 F1 F0 95 02 F5 0A 40 01 F0 00	1/25/2010 8:37:38 PM	0	09502	50A	4001
F1 F1 03 21 F1 A4 00 9F 00 00 00 00	1/26/2010 12:26:00 AM	0	10321	1A	4009
F1 F1 03 21 F1 A4 00 9F 00 00 00 00	1/26/2010 12:41:54 AM	0	10321	1A	4009
F1 F0 95 02 F1 A4 3F 00 00 00 00 00	1/26/2010 12:54:45 AM	0	09502	1A	43
F1 F1 03 21 F1 A4 01 0F 00 00 00 00	1/26/2010 1:02:11 AM	0	10321	1A	4010
F1 F1 03 21 F1 A4 00 4F 00 00 00 00	1/26/2010 1:37:50 AM	0	10321	1A	4004
F1 F0 95 02 F1 A4 00 3F 00 00 00 00	1/26/2010 1:38:17 AM	0	09502	1A	4003
F1 F1 03 21 F1 A4 01 0F 00 00 00 00	1/26/2010 7:33:40 AM	0	10321	1A	4010
F1 F1 03 21 F1 A4 00 1F 00 00 00 00	1/26/2010 7:34:57 AM	0	10321	1A	4001
F1 F0 95 02 F1 A4 00 4F 00 00 00 00	1/26/2010 7:43:59 AM	0	09502	1A	4004
F1 F1 03 21 F1 A4 00 1F 00 00 00 00	1/26/2010 8:02:16 AM	0	10321	1A	4001
F1 F1 03 21 F1 A4 00 4F 00 00 00 00	1/26/2010 8:02:17 AM	0	10321	1A	4004
F1 F0 95 02 F1 A4 00 4F 00 00 00 00	1/26/2010 8:02:24 AM	0	09502	1A	4004
F1 F1 03 21 F1 A4 00 7F 00 00 00 00	1/26/2010 8:31:11 AM	0	10321	1A	4007
F1 F1 03 21 F1 A4 00 3F 00 00 00 00	1/26/2010 8:31:22 AM	0	10321	1A	4003
F1 F0 95 02 F1 A9 6F 00 00 00 00 00	1/26/2010 11:34:29 AM	0	09502	1A	96
F1 F0 95 02 F1 A4 3F 00 00 00 00 00	1/26/2010 11:36:36 AM	0	09502	1A	43
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/26/2010 9:25:14 PM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 39 F5 00 00	1/26/2010 10:20:50 PM	0	09502	50A	39
33 B2 F1 F0 95 02 F5 0A 35 F0 00 00	1/26/2010 10:57:26 PM	0	09502	50A	35
33 B2 F1 F0 95 02 F5 0A 31 F0 00 00	1/26/2010 10:58:18 PM	0	09502	50A	31
33 B2 F1 F0 95 02 F5 0A 38 F0 00 00	1/26/2010 11:02:12 PM	0	09502	50A	38
33 B2 F1 F0 95 02 F5 0A 40 03 F0 00	1/26/2010 11:12:01 PM	0	09502	50A	4003
F1 F0 95 02 F1 A4 00 4F 00 00 00 00	1/26/2010 11:22:43 PM	1	09502	1A	4004
33 B2 F1 F0 95 02 F5 0A 25 F0 00 00	1/26/2010 11:28:46 PM	0	09502	50A	25
F1 F0 95 02 F1 A9 4F 00 00 00 00 00	1/26/2010 11:46:25 PM	0	09502	1A	94
F2 F0 95 02 F1 A1 48 F0 00 00 00 00	1/27/2010 5:42:01 AM	0	09502	1A	148
F1 F0 95 02 F1 A3 2F 00 00 00 00 00	1/27/2010 5:53:54 AM	0	09502	1A	32
F2 F0 95 02 F1 A1 48 F0 00 00 00 00	1/27/2010 6:15:44 AM	0	09502	1A	148
F1 F0 95 02 F1 A1 48 F0 00 00 00 00	1/27/2010 6:42:05 AM	0	09502	1A	148
F1 F0 95 02 F1 A1 48 F0 00 00 00 00	1/27/2010 7:42:02 AM	0	09502	1A	148
F2 F1 03 21 F1 A4 00 1F 00 00 00 00	1/27/2010 8:00:12 AM	0	10321	1A	4001
F2 F1 03 21 F1 A4 00 4F 00 00 00 00	1/27/2010 8:00:12 AM	0	10321	1A	4004
F1 F0 95 02 F1 A0 1F 00 00 00 00 00	1/27/2010 8:00:13 AM	0	09502	1A	01
F1 F0 95 02 F1 A4 4F 00 00 00 00 00	1/27/2010 8:00:13 AM	0	09502	1A	44

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
F2 F1 03 21 F1 A4 00 2F 00 00 00 00	1/27/2010 8:00:13 AM	0	10321	1A	4002
F1 F0 95 02 F1 A9 0F 00 00 00 00 00	1/27/2010 8:29:13 AM	0	09502	1A	90
F1 F0 95 02 F1 A8 6F 00 00 00 00 00	1/27/2010 8:42:30 AM	0	09502	1A	86
F1 F0 95 02 F1 A9 4F 00 00 00 00 00	1/27/2010 8:42:30 AM	0	09502	1A	94
F1 F0 95 02 F1 A9 7F 00 00 00 00 00	1/27/2010 8:42:30 AM	0	09502	1A	97
F1 F0 95 02 F1 A9 0F 00 00 00 00 00	1/27/2010 8:42:41 AM	0	09502	1A	90
F1 F0 95 02 F5 2A 56 F0 00 00 00 00	1/27/2010 9:08:01 AM	0	09502	52A	56
F1 F0 95 02 F1 A1 49 F0 00 00 00 00	1/27/2010 9:17:42 AM	0	09502	1A	149
F1 F0 95 02 F1 A1 49 F0 00 00 00 00	1/27/2010 10:11:31 AM	0	09502	1A	149
F1 F0 95 02 F1 A5 6F 00 00 00 00 00	1/27/2010 10:11:44 AM	0	09502	1A	56
F1 F0 95 02 F1 A4 3F 00 00 00 00 00	1/27/2010 11:15:47 AM	0	09502	1A	43
33 B2 F1 F0 95 02 F5 0A 66 F0 00 00	1/27/2010 11:16:39 AM	0	09502	50A	66
33 B2 F1 F0 95 02 F5 0A 10 8F 00 00	1/27/2010 11:30:15 AM	0	09502	50A	108
33 B2 F1 F0 95 02 F5 0A 10 8F 00 00	1/27/2010 11:56:27 AM	0	09502	50A	108
F1 F0 95 02 F1 A5 6F 00 00 00 00 00	1/27/2010 1:20:18 PM	0	09502	1A	56
F1 F0 95 02 F5 2A 40 01 F0 00 00 00	1/27/2010 1:23:03 PM	0	09502	52A	4001
F1 F0 95 02 F5 2A 40 02 F0 00 00 00	1/27/2010 1:29:38 PM	0	09502	52A	4002
F1 F0 95 02 F1 A0 15 0F 00 00 00 00	1/27/2010 1:49:32 PM	0	09502	1A	0150
F1 F0 95 02 F1 A9 6F 00 00 00 00 00	1/27/2010 2:06:22 PM	0	09502	1A	96
F1 F0 95 02 F1 A4 00 5F 00 00 00 00	1/27/2010 2:45:03 PM	0	09502	1A	4005
F1 F0 95 02 F5 2A 40 02 F0 00 00 00	1/27/2010 2:46:45 PM	0	09502	52A	4002
F1 F0 95 02 F1 A0 15 0F 00 00 00 00	1/27/2010 3:09:07 PM	0	09502	1A	0150
F1 F0 95 02 F1 A4 00 7F 00 00 00 00	1/27/2010 3:20:40 PM	0	09502	1A	4007
F1 F0 95 02 F5 2A 51 F0 00 00 00 00	1/27/2010 3:38:07 PM	0	09502	52A	51
F1 F0 95 02 F1 A8 1F 00 00 00 00 00	1/27/2010 3:46:02 PM	0	09502	1A	81
33 B2 F1 F0 95 02 F5 0A 37 F5 00 00	1/27/2010 5:50:46 PM	0	09502	50A	37
33 B2 F1 F0 95 02 F5 0A 24 F5 00 00	1/27/2010 5:50:58 PM	0	09502	50A	24
33 B2 F1 F0 95 02 F5 0A 29 F0 00 00	1/27/2010 5:51:04 PM	0	09502	50A	29
33 B2 F1 F0 95 02 F5 0A 23 F0 00 00	1/27/2010 5:51:09 PM	0	09502	50A	23
33 B2 F1 F0 95 02 F5 0A 36 F5 00 00	1/27/2010 5:51:14 PM	0	09502	50A	36
F1 F0 95 02 F5 2A 52 F0 00 00 00 00	1/27/2010 5:56:23 PM	0	09502	52A	52
F1 F0 95 02 F5 2A 51 F0 00 00 00 00	1/27/2010 5:56:50 PM	0	09502	52A	51
F1 F0 95 02 F1 A4 00 7F 00 00 00 00	1/27/2010 6:30:54 PM	0	09502	1A	4007
33 B2 F1 F0 95 02 F5 0A 43 F0 00 00	1/27/2010 6:58:55 PM	0	09502	50A	43
33 B2 F1 F0 95 02 F5 0A 44 F0 00 00	1/27/2010 6:58:55 PM	0	09502	50A	44
33 B2 F1 F0 95 02 F5 0A 41 F0 00 00	1/27/2010 6:59:00 PM	0	09502	50A	41
F1 F0 95 02 F1 A4 00 5F 00 00 00 00	1/27/2010 7:41:03 PM	0	09502	1A	4005
F1 F0 95 02 F5 2A 52 F0 00 00 00 00	1/27/2010 7:48:40 PM	0	09502	52A	52
33 B2 F1 F0 95 02 F5 0A 48 F0 00 00	1/27/2010 8:03:50 PM	0	09502	50A	48
33 B2 F1 F0 95 02 F5 0A 10 2F 00 00	1/27/2010 8:04:00 PM	0	09502	50A	102
33 B2 F1 F0 95 02 F5 0A 42 F0 00 00	1/27/2010 8:04:00 PM	0	09502	50A	42
33 B2 F1 F0 95 02 F5 0A 47 F0 00 00	1/27/2010 8:04:00 PM	0	09502	50A	47
33 B2 F1 F0 95 02 F5 0A 49 F0 00 00	1/27/2010 8:09:36 PM	0	09502	50A	49
F1 F0 95 02 F1 A1 07 F0 00 00 00 00	1/28/2010 12:28:54 AM	0	09502	1A	107
F1 F0 95 02 F1 A1 07 F0 00 00 00 00	1/28/2010 12:58:06 AM	0	09502	1A	107

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
F1 F0 95 02 F5 0A 40 F0 00 00 00 00	1/28/2010 5:41:35 AM	0	09502	50A	40
33 B2 F1 F0 95 02 F5 0A 52 F5 00 00	1/28/2010 5:50:42 AM	0	09502	50A	52
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/28/2010 5:56:07 AM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 10 1F 00 00	1/28/2010 5:56:34 AM	0	09502	50A	101
33 B2 F1 F0 95 02 F5 0A 26 F0 00 00	1/28/2010 5:57:06 AM	0	09502	50A	26
33 B2 F1 F0 95 02 F5 0A 51 F0 00 00	1/28/2010 5:57:42 AM	0	09502	50A	51
33 B2 F1 F0 95 02 F5 0A 40 03 F0 00	1/28/2010 6:03:24 AM	0	09502	50A	4003
F1 F0 95 02 F5 2A 18 F0 00 00 00 00	1/28/2010 6:25:37 AM	0	09502	52A	18
F1 F0 95 02 F1 A8 1F 00 00 00 00 00	1/28/2010 6:27:44 AM	0	09502	1A	81
F1 F0 95 02 F5 1A 40 02 F0 00 00 00	1/28/2010 6:29:12 AM	0	09502	51A	4002
F1 F0 95 02 F5 1A 40 02 F0 00 00 00	1/28/2010 7:52:39 AM	0	09502	51A	4002
F1 F0 95 02 F5 1A 40 01 F0 00 00 00	1/28/2010 7:54:59 AM	0	09502	51A	4001
F1 F0 95 02 F5 2A 19 F0 00 00 00 00	1/28/2010 7:58:50 AM	0	09502	52A	19
F1 F0 95 02 F5 1A 14 F0 00 00 00 00	1/28/2010 8:32:02 AM	0	09502	51A	14
F1 F0 95 02 F5 2A 16 F0 00 00 00 00	1/28/2010 9:29:20 AM	0	09502	52A	16
F1 F0 95 02 F5 1A 15 F0 00 00 00 00	1/28/2010 9:43:00 AM	0	09502	51A	15
F1 F0 95 02 F1 A4 00 1F 00 00 00 00	1/28/2010 9:45:48 AM	0	09502	1A	4001
33 B2 F1 F0 95 02 F5 0A 30 F5 00 00	1/28/2010 9:49:46 AM	0	09502	50A	30
33 B2 F1 F0 95 02 F5 0A 31 F0 00 00	1/28/2010 9:49:46 AM	0	09502	50A	31
33 B2 F1 F0 95 02 F5 0A 38 F0 00 00	1/28/2010 9:49:46 AM	0	09502	50A	38
33 B2 F1 F0 95 02 F5 0A 39 F5 00 00	1/28/2010 9:51:18 AM	0	09502	50A	39
33 B2 F1 F0 95 02 F5 0A 33 F0 00 00	1/28/2010 9:52:42 AM	0	09502	50A	33
33 B2 F1 F0 95 02 F5 0A 35 F0 00 00	1/28/2010 9:52:47 AM	0	09502	50A	35
33 B2 F1 F0 95 02 F5 0A 25 F0 00 00	1/28/2010 9:52:50 AM	0	09502	50A	25
33 B2 F1 F0 95 02 F5 0A 36 F5 00 00	1/28/2010 9:52:51 AM	0	09502	50A	36
33 B2 F1 F0 95 02 F5 0A 24 F5 00 00	1/28/2010 9:53:29 AM	0	09502	50A	24
33 B2 F1 F0 95 02 F5 0A 29 F0 00 00	1/28/2010 9:53:40 AM	0	09502	50A	29
33 B2 F1 F0 95 02 F5 0A 31 F0 00 00	1/28/2010 9:53:44 AM	0	09502	50A	31
33 B2 F1 F0 95 02 F5 0A 37 F5 00 00	1/28/2010 9:54:01 AM	0	09502	50A	37
33 B2 F1 F0 95 02 F5 0A 23 F0 00 00	1/28/2010 9:54:14 AM	0	09502	50A	23
F1 F0 95 02 F1 A4 00 3F 00 00 00 00	1/28/2010 10:01:42 AM	0	09502	1A	4003
F1 F0 95 02 F5 1A 40 01 F0 00 00 00	1/28/2010 10:09:20 AM	0	09502	51A	4001
F1 F0 95 02 F5 1A 23 F0 00 00 00 00	1/28/2010 10:15:44 AM	0	09502	51A	23
F1 F0 95 02 F1 A1 07 F0 00 00 00 00	1/28/2010 10:25:18 AM	0	09502	1A	107
F1 F0 95 02 F1 A9 7F 00 00 00 00 00	1/28/2010 10:26:39 AM	0	09502	1A	97
F1 F0 95 02 F1 A9 4F 00 00 00 00 00	1/28/2010 10:28:11 AM	0	09502	1A	94
F1 F0 95 02 F1 A8 6F 00 00 00 00 00	1/28/2010 10:29:13 AM	0	09502	1A	86
F1 F0 95 02 F5 1A 23 F0 00 00 00 00	1/28/2010 11:24:58 AM	0	09502	51A	23
F1 F0 95 02 F5 1A 24 F0 00 00 00 00	1/28/2010 11:28:26 AM	0	09502	51A	24
F1 F0 95 02 F1 A1 07 F0 00 00 00 00	1/28/2010 11:32:21 AM	0	09502	1A	107
F1 F0 95 02 F1 A9 7F 00 00 00 00 00	1/28/2010 11:32:26 AM	0	09502	1A	97
F1 F0 95 02 F1 A9 4F 00 00 00 00 00	1/28/2010 11:32:30 AM	0	09502	1A	94
F1 F0 95 02 F1 A8 6F 00 00 00 00 00	1/28/2010 11:32:34 AM	0	09502	1A	86
F1 F0 95 02 F5 2A 51 F0 00 00 00 00	1/28/2010 11:34:20 AM	0	09502	52A	51
F1 F0 95 02 F5 1A 24 F0 00 00 00 00	1/28/2010 11:53:41 AM	0	09502	51A	24

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
F1 F0 95 02 F5 2A 38 F0 00 00 00 00	1/28/2010 11:53:53 AM	0	09502	52A	38
F1 F0 95 02 F1 A9 6F 00 00 00 00 00	1/28/2010 12:27:07 PM	0	09502	1A	96
33 B2 F1 F0 95 02 F5 0A 20 F0 00 00	1/28/2010 1:11:56 PM	0	09502	50A	20
33 B2 F1 F0 95 02 F5 0A 56 F0 00 00	1/28/2010 1:12:00 PM	0	09502	50A	56
33 B2 F1 F0 95 02 F5 0A 54 F0 00 00	1/28/2010 1:12:03 PM	0	09502	50A	54
33 B2 F1 F0 95 02 F5 0A 72 F0 00 00	1/28/2010 1:12:09 PM	0	09502	50A	72
33 B2 F1 F0 95 02 F5 0A 73 F0 00 00	1/28/2010 1:12:13 PM	0	09502	50A	73
F1 F0 95 02 F1 A4 00 2F 00 00 00 00	1/28/2010 1:19:54 PM	0	09502	1A	4002
F1 F0 95 02 F1 A4 00 3F 00 00 00 00	1/28/2010 1:20:35 PM	0	09502	1A	4003
33 B2 F1 F0 95 02 F5 0A 10 3F 00 00	1/28/2010 1:24:38 PM	0	09502	50A	103
33 B2 F1 F0 95 02 F5 0A 53 F0 00 00	1/28/2010 1:24:43 PM	0	09502	50A	53
33 B2 F1 F0 95 02 F5 0A 55 F0 00 00	1/28/2010 1:24:43 PM	0	09502	50A	55
33 B2 F1 F0 95 02 F5 0A 22 F0 00 00	1/28/2010 1:24:49 PM	0	09502	50A	22
33 B2 F1 F0 95 02 F5 0A 10 8F 00 00	1/28/2010 1:57:05 PM	0	09502	50A	108
33 B2 F1 F0 95 02 F5 0A 11 F0 00 00	1/28/2010 1:57:05 PM	0	09502	50A	11
33 B2 F1 F0 95 02 F5 0A 69 F0 00 00	1/28/2010 1:57:05 PM	0	09502	50A	69
F1 F0 95 02 F1 A4 3F 00 00 00 00 00	1/28/2010 2:29:58 PM	0	09502	1A	43
F1 F0 95 02 F5 2A 38 F0 00 00 00 00	1/28/2010 2:44:44 PM	0	09502	52A	38
F1 F0 95 02 F5 1A 14 F0 00 00 00 00	1/28/2010 2:57:25 PM	0	09502	51A	14
F1 F0 95 02 F5 1A 15 F0 00 00 00 00	1/28/2010 2:57:56 PM	0	09502	51A	15
33 B2 F1 F0 95 02 F5 0A 57 F0 00 00	1/28/2010 3:10:18 PM	0	09502	50A	57
33 B2 F1 F0 95 02 F5 0A 21 F0 00 00	1/28/2010 3:13:57 PM	0	09502	50A	21
33 B2 F1 F0 95 02 F5 0A 74 F0 00 00	1/28/2010 3:14:24 PM	0	09502	50A	74
33 B2 F1 F0 95 02 F5 0A 58 F0 00 00	1/28/2010 3:14:37 PM	0	09502	50A	58
33 B2 F1 F0 95 02 F5 0A 64 F0 00 00	1/28/2010 3:15:05 PM	0	09502	50A	64
33 B2 F1 F0 95 02 F5 0A 18 F0 00 00	1/28/2010 3:15:32 PM	0	09502	50A	18
F1 F0 95 02 F5 0A 40 F0 00 00 00 00	1/28/2010 3:26:38 PM	0	09502	50A	40
F1 F0 95 02 F5 2A 19 F0 00 00 00 00	1/28/2010 3:26:50 PM	0	09502	52A	19
33 B2 F1 F0 95 02 F5 0A 61 F0 00 00	1/28/2010 3:44:18 PM	0	09502	50A	61
33 B2 F1 F0 95 02 F5 0A 63 F0 00 00	1/28/2010 3:44:18 PM	0	09502	50A	63
33 B2 F1 F0 95 02 F5 0A 17 F0 00 00	1/28/2010 3:57:27 PM	0	09502	50A	17
33 B2 F1 F0 95 02 F5 0A 62 F0 00 00	1/28/2010 3:57:27 PM	0	09502	50A	62
33 B2 F1 F0 95 02 F5 0A 59 F0 00 00	1/28/2010 5:13:33 PM	0	09502	50A	59
33 B2 F1 F0 95 02 F5 0A 45 F0 00 00	1/28/2010 5:43:02 PM	0	09502	50A	45
33 B2 F1 F0 95 02 F5 0A 47 F0 00 00	1/28/2010 5:43:43 PM	0	09502	50A	47
33 B2 F1 F0 95 02 F5 0A 51 F0 00 00	1/28/2010 5:43:43 PM	0	09502	50A	51
33 B2 F1 F0 95 02 F5 0A 49 F0 00 00	1/28/2010 5:44:25 PM	0	09502	50A	49
33 B2 F1 F0 95 02 F5 0A 52 F5 00 00	1/28/2010 5:44:36 PM	0	09502	50A	52
33 B2 F1 F0 95 02 F5 0A 40 03 F0 00	1/28/2010 5:45:12 PM	0	09502	50A	4003
33 B2 F1 F0 95 02 F5 0A 10 1F 00 00	1/28/2010 5:45:28 PM	0	09502	50A	101
33 B2 F1 F0 95 02 F5 0A 10 2F 00 00	1/28/2010 5:45:43 PM	0	09502	50A	102
33 B2 F1 F0 95 02 F5 0A 26 F0 00 00	1/28/2010 5:46:06 PM	0	09502	50A	26
33 B2 F1 F0 95 02 F5 0A 44 F0 00 00	1/28/2010 5:46:33 PM	0	09502	50A	44
33 B2 F1 F0 95 02 F5 0A 48 F0 00 00	1/28/2010 5:46:44 PM	0	09502	50A	48
33 B2 F1 F0 95 02 F5 0A 43 F0 00 00	1/28/2010 5:46:51 PM	0	09502	50A	43

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 41 F0 00 00	1/28/2010 5:47:01 PM	0	09502	50A	41
33 B2 F1 F0 95 02 F5 0A 34 F0 00 00	1/28/2010 5:55:23 PM	0	09502	50A	34
F1 F0 95 02 F1 A4 00 2F 00 00 00 00	1/28/2010 6:03:05 PM	0	09502	1A	4002
F1 F0 95 02 F1 A3 7F 00 00 00 00 00	1/28/2010 7:16:15 PM	0	09502	1A	37
F2 F0 95 02 F5 1A 20 F0 00 00 00 00	1/28/2010 7:55:31 PM	0	09502	51A	20
F1 F0 95 02 F5 1A 20 F0 00 00 00 00	1/28/2010 8:15:22 PM	0	09502	51A	20
F2 F0 95 02 F5 1A 9F 00 00 00 00 00	1/28/2010 9:02:11 PM	0	09502	51A	9
F1 F0 95 02 F5 1A 20 F0 00 00 00 00	1/28/2010 9:15:02 PM	0	09502	51A	20
33 B2 F1 F0 95 02 F5 0A 75 F0 00 00	1/28/2010 9:28:11 PM	0	09502	50A	75
0F 1F 09 50 2F 51 A9 F0 00 00 00 00	1/28/2010 10:12:11 PM	0	09502	51A	9
F2 F0 95 02 F5 1A 20 F0 00 00 00 00	1/28/2010 10:18:20 PM	0	09502	51A	20
F1 F0 95 02 F1 A3 7F 00 00 00 00 00	1/28/2010 10:56:45 PM	0	09502	1A	37
F2 F0 95 02 F5 1A 9F 00 00 00 00 00	1/28/2010 11:02:11 PM	0	09502	51A	9
0F 1F 09 50 2F 51 A9 F0 00 00 00 00	1/28/2010 11:02:18 PM	0	09502	51A	9
F2 F0 95 02 F5 1A 23 F0 00 00 00 00	1/28/2010 11:05:50 PM	0	09502	51A	23
F2 F0 95 02 F5 1A 24 F0 00 00 00 00	1/28/2010 11:05:50 PM	0	09502	51A	24
F2 F0 95 02 F5 2A 18 F0 00 00 00 00	1/28/2010 11:05:50 PM	0	09502	52A	18
F2 F0 95 02 F5 2A 16 F0 00 00 00 00	1/28/2010 11:06:04 PM	0	09502	52A	16
F1 F0 95 02 F5 2A 57 F0 00 00 00 00	1/29/2010 1:13:21 AM	0	09502	52A	57
33 B2 F1 F0 95 02 F5 0A 19 F0 00 00	1/29/2010 1:54:52 AM	0	09502	50A	19
33 B2 F1 F0 95 02 F5 0A 54 F0 00 00	1/29/2010 1:55:00 AM	0	09502	50A	54
33 B2 F1 F0 95 02 F5 0A 17 F0 00 00	1/29/2010 1:55:04 AM	0	09502	50A	17
33 B2 F1 F0 95 02 F5 0A 18 F0 00 00	1/29/2010 1:55:11 AM	0	09502	50A	18
33 B2 F1 F0 95 02 F5 0A 10 3F 00 00	1/29/2010 1:55:15 AM	0	09502	50A	103
33 B2 F1 F0 95 02 F5 0A 62 F0 00 00	1/29/2010 1:55:20 AM	0	09502	50A	62
33 B2 F1 F0 95 02 F5 0A 63 F0 00 00	1/29/2010 1:55:24 AM	0	09502	50A	63
33 B2 F1 F0 95 02 F5 0A 64 F0 00 00	1/29/2010 1:55:27 AM	0	09502	50A	64
33 B2 F1 F0 95 02 F5 0A 73 F0 00 00	1/29/2010 1:55:33 AM	0	09502	50A	73
33 B2 F1 F0 95 02 F5 0A 27 F0 00 00	1/29/2010 1:55:37 AM	0	09502	50A	27
33 B2 F1 F0 95 02 F5 0A 57 F0 00 00	1/29/2010 1:55:37 AM	0	09502	50A	57
33 B2 F1 F0 95 02 F5 0A 72 F0 00 00	1/29/2010 1:55:37 AM	0	09502	50A	72
33 B2 F1 F0 95 02 F5 0A 10 8F 00 00	1/29/2010 1:55:46 AM	0	09502	50A	108
33 B2 F1 F0 95 02 F5 0A 20 F0 00 00	1/29/2010 1:55:46 AM	0	09502	50A	20
33 B2 F1 F0 95 02 F5 0A 34 F0 00 00	1/29/2010 1:55:46 AM	0	09502	50A	34
33 B2 F1 F0 95 02 F5 0A 42 F0 00 00	1/29/2010 1:55:46 AM	0	09502	50A	42
33 B2 F1 F0 95 02 F5 0A 50 F0 00 00	1/29/2010 1:55:46 AM	0	09502	50A	50
33 B2 F1 F0 95 02 F5 0A 59 F0 00 00	1/29/2010 1:55:46 AM	0	09502	50A	59
33 B2 F1 F0 95 02 F5 0A 11 F0 00 00	1/29/2010 1:55:47 AM	0	09502	50A	11
33 B2 F1 F0 95 02 F5 0A 22 F0 00 00	1/29/2010 1:55:47 AM	0	09502	50A	22
33 B2 F1 F0 95 02 F5 0A 61 F0 00 00	1/29/2010 1:55:47 AM	0	09502	50A	61
33 B2 F1 F0 95 02 F5 0A 69 F0 00 00	1/29/2010 1:55:47 AM	0	09502	50A	69
33 B2 F1 F0 95 02 F5 0A 21 F0 00 00	1/29/2010 1:55:48 AM	0	09502	50A	21
33 B2 F1 F0 95 02 F5 0A 53 F0 00 00	1/29/2010 1:55:48 AM	0	09502	50A	53
33 B2 F1 F0 95 02 F5 0A 56 F0 00 00	1/29/2010 1:55:48 AM	0	09502	50A	56
33 B2 F1 F0 95 02 F5 0A 58 F0 00 00	1/29/2010 1:55:48 AM	0	09502	50A	58

RFIDRead					
RFIDTag	ReadTime	Rework	JobNo	DivNo	PieceNo
33 B2 F1 F0 95 02 F5 0A 74 F0 00 00	1/29/2010 1:55:49 AM	0	09502	50A	74
33 B2 F1 F0 95 02 F5 0A 55 F0 00 00	1/29/2010 1:55:51 AM	0	09502	50A	55
F1 F0 95 02 F5 2A 40 01 F0 00 00 00	1/29/2010 2:14:26 AM	0	09502	52A	4001

where:

RFIDTag: The id assigned to the RFID tag.

ReadTime: The date and the time that RFID has been read in the shop.

Rework: determines if the steel piece had required rework.

div_id: The id of the division which piece is belonged to.

piece_id: The piece id.

Appendix C: Details regarding the structural steel federation

The proposed integrated and real-time monitoring and control framework discussed in Chapter 2 was verified through an implementation of that framework in the form of a federation specifically designed for structural steel fabrication projects. Two PhD students at the University of Alberta worked together on developing the structural steel federation (Azimi and Alvanchi 2009), however, based on their purposes federates were added to or removed from the federation. Details regarding the federation developed are presented below.

Conceptual design

Federation development started with conceptual design. Through literature review, several site visits and meetings with a variety of experts involved in the collaborating company, a "big picture" of the structural steel construction was developed and the major objects that play a role in structural steel construction, and their interactions and relationships, were identified (Figure C-1).

For the sake of reusability, product model, resource, and environmental models were also defined (Figure C-2 , Figure C-3, Figure C-4)

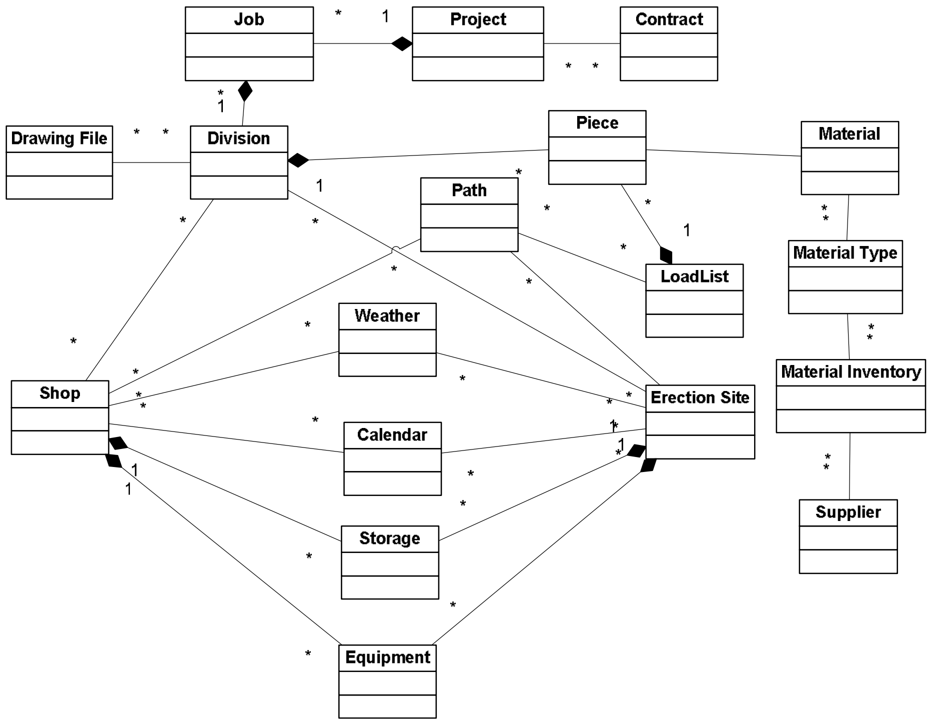


Figure C-1. Structural steel object model (object relationships).

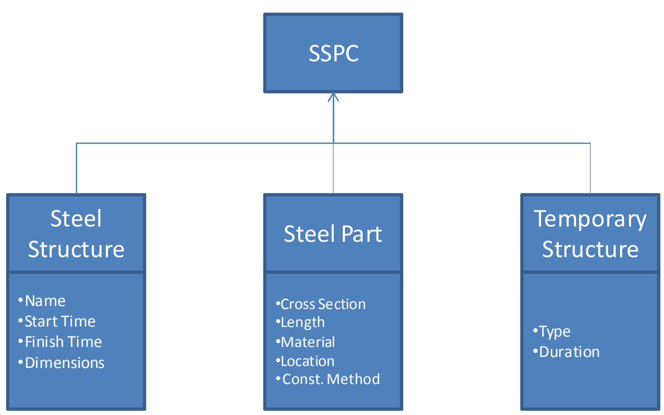


Figure C-2. Product model for steel projects.

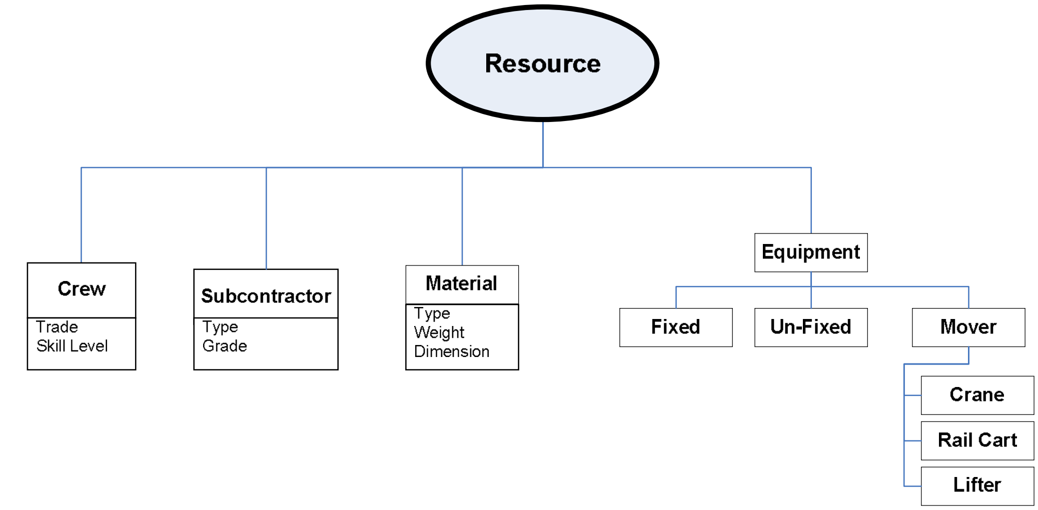


Figure C-3. Resource model for steel projects.

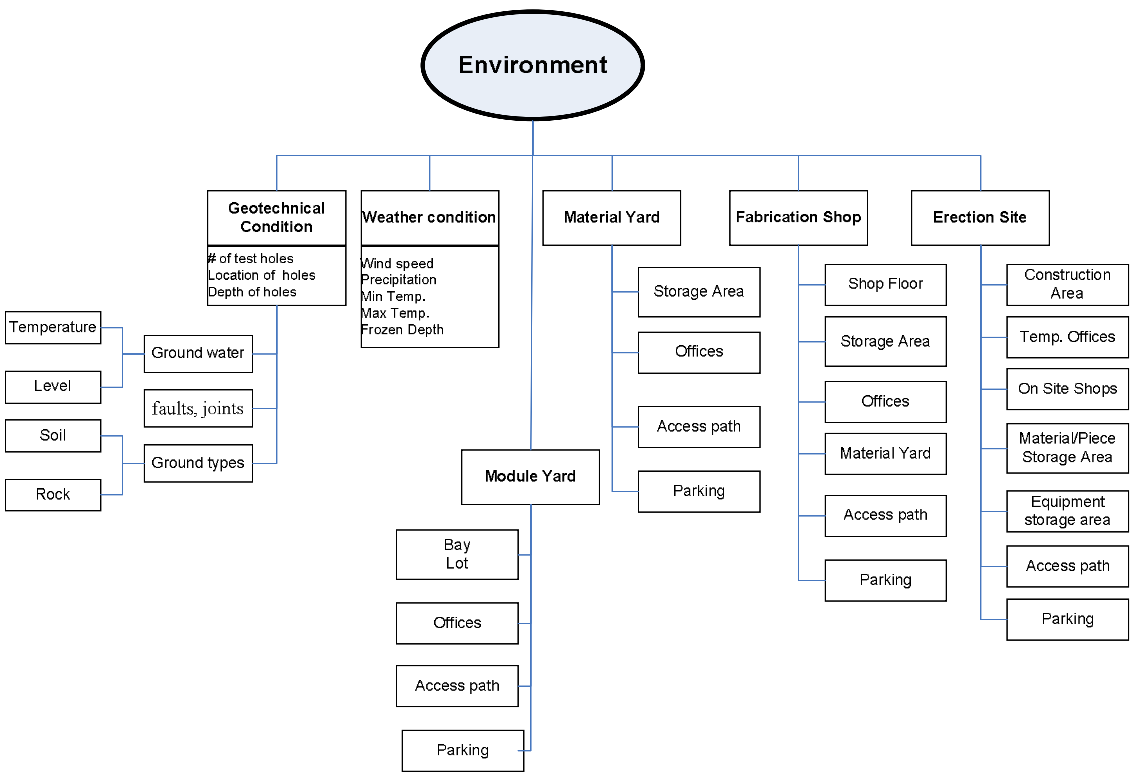


Figure C-4. Environment model for steel projects.

Focusing on the fabrication phase, different operations required for fabricating structural steel pieces were identified and summarized in the form of a flowchart depicted in Figure C-5. The next step for model development for structural steel fabrication was identifying a detailed object model relationship for fabrication shop (Figure C-6). This provides a sound base for coding the required object classes in the programming phase.

Initial federation development also requires a good understanding about the interactions between different federates shaping the federation. A data flow diagram was prepared for this purpose as shown in Figure C-7.

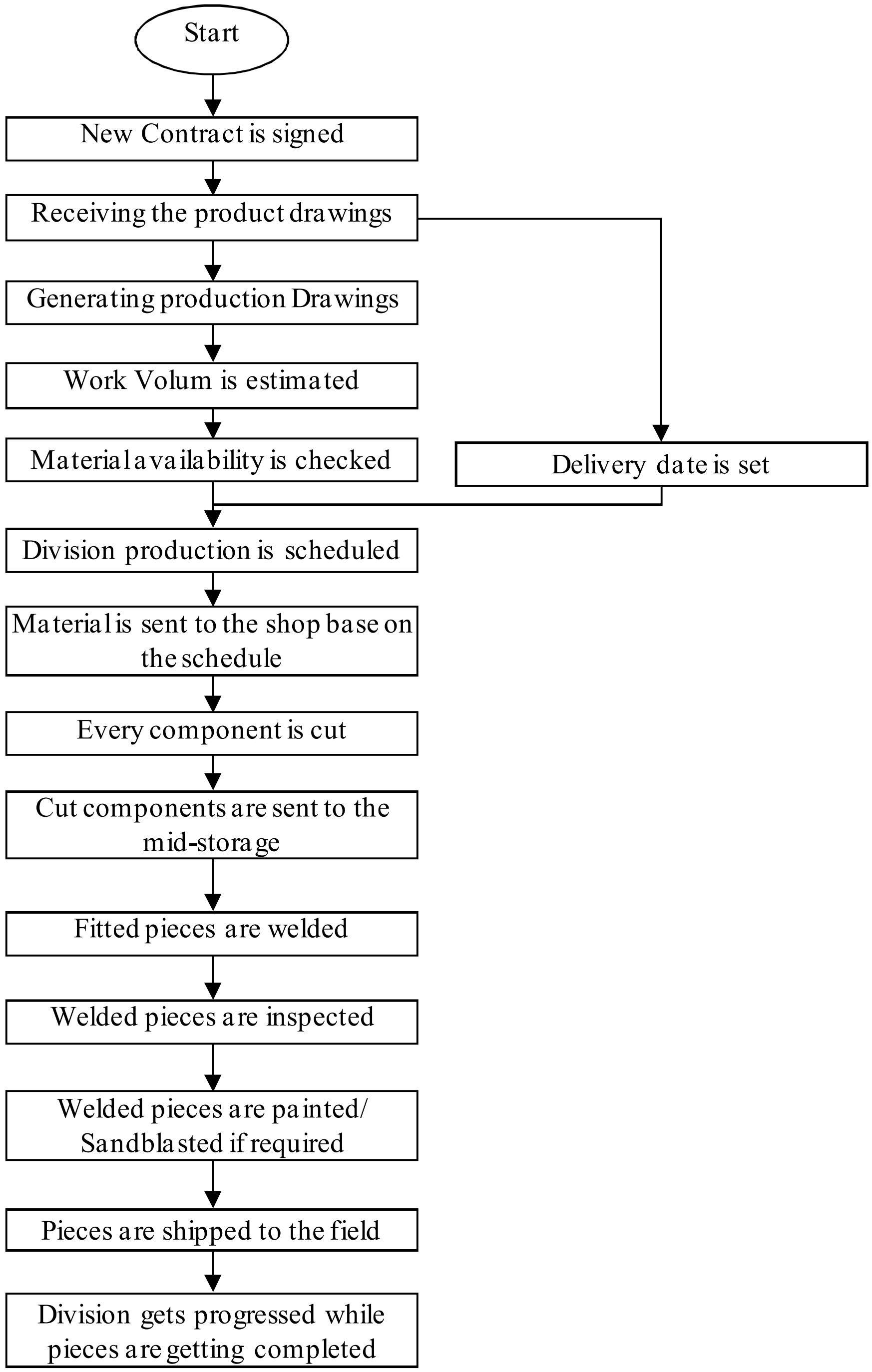


Figure C-5. Structural steel construction operations.

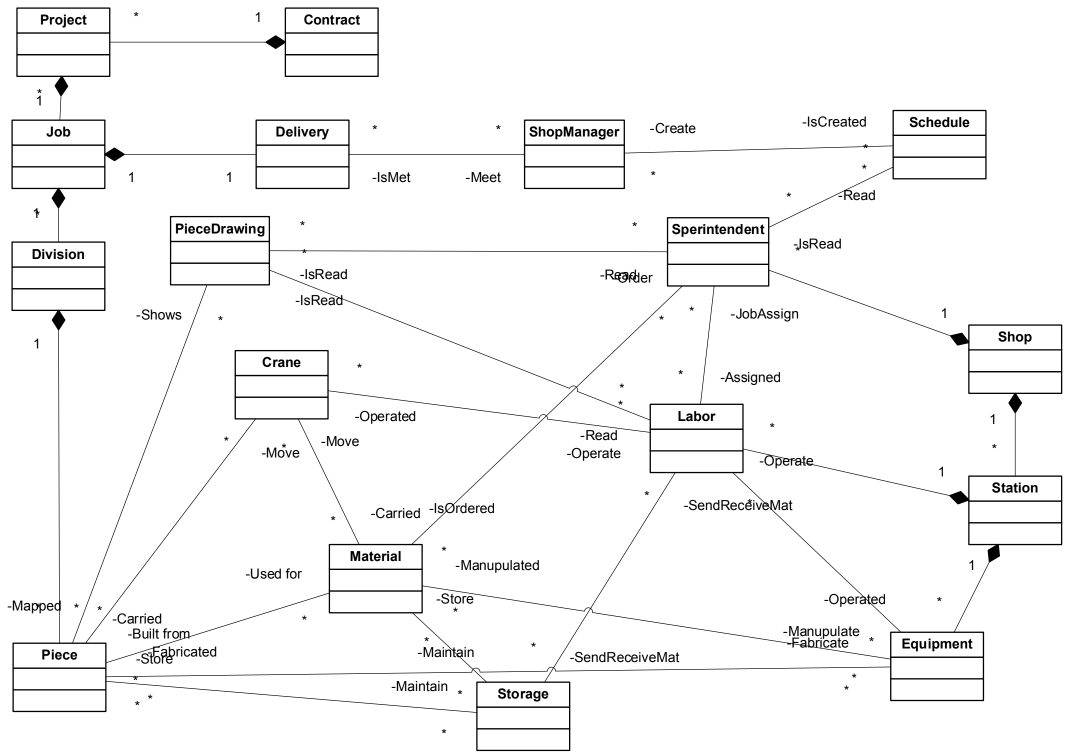


Figure C-6. Object model relationship for the structural steel fabrication shop.

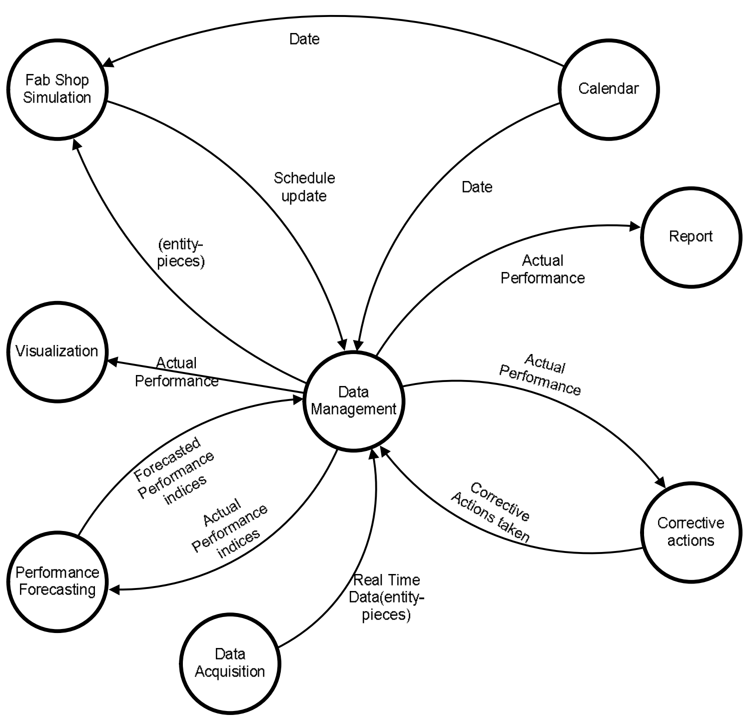


Figure C-7. Data flow diagram for the proposed federation.

The last step of the conceptual design includes developing the programming logic and drawing programming flowcharts to be used for developing each federate. Samples of the developed programming flowcharts are shown in Figure C-8, Figure C-9 and Figure C-10.

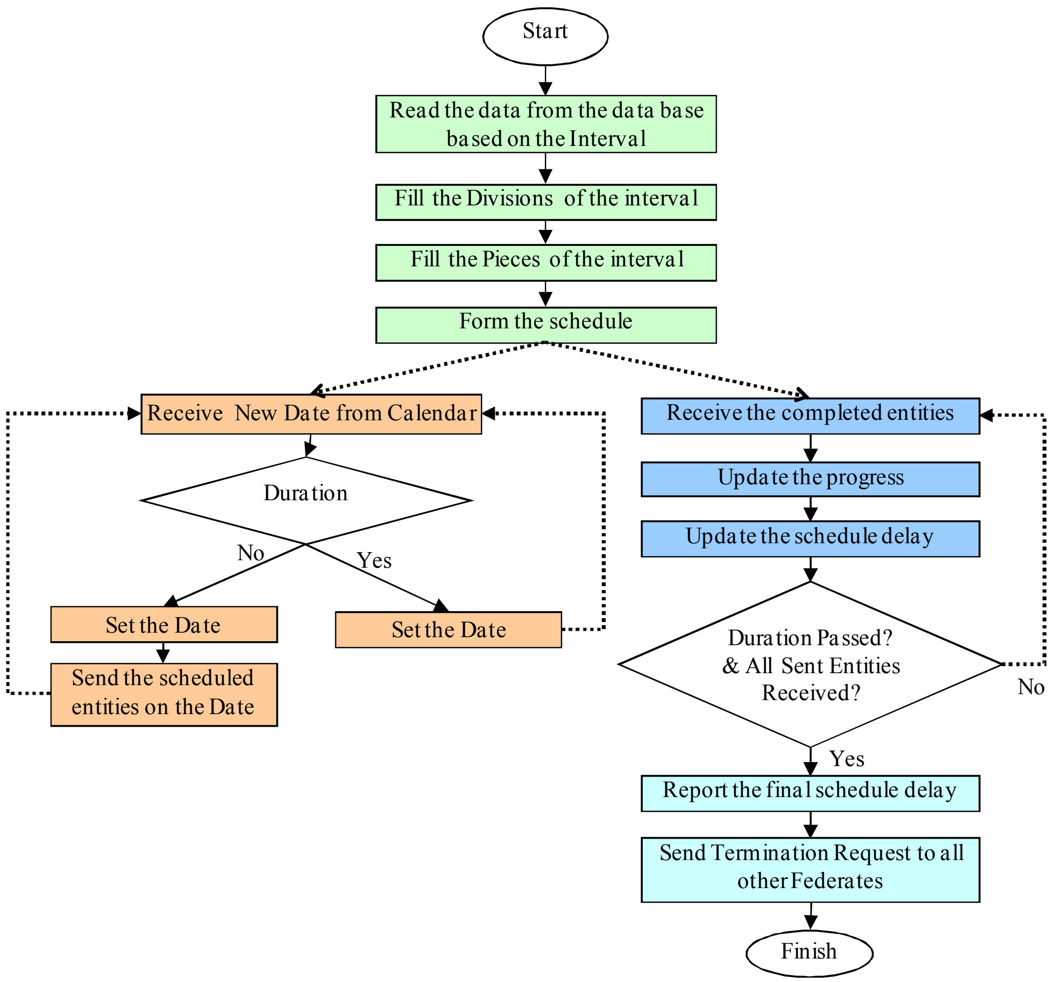


Figure C-8. Programming flowchart for the data management federate.

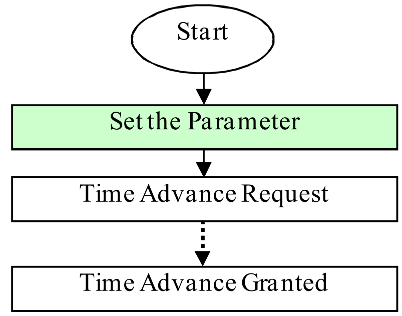


Figure C-9. Programming flowchart for the calendar federate.

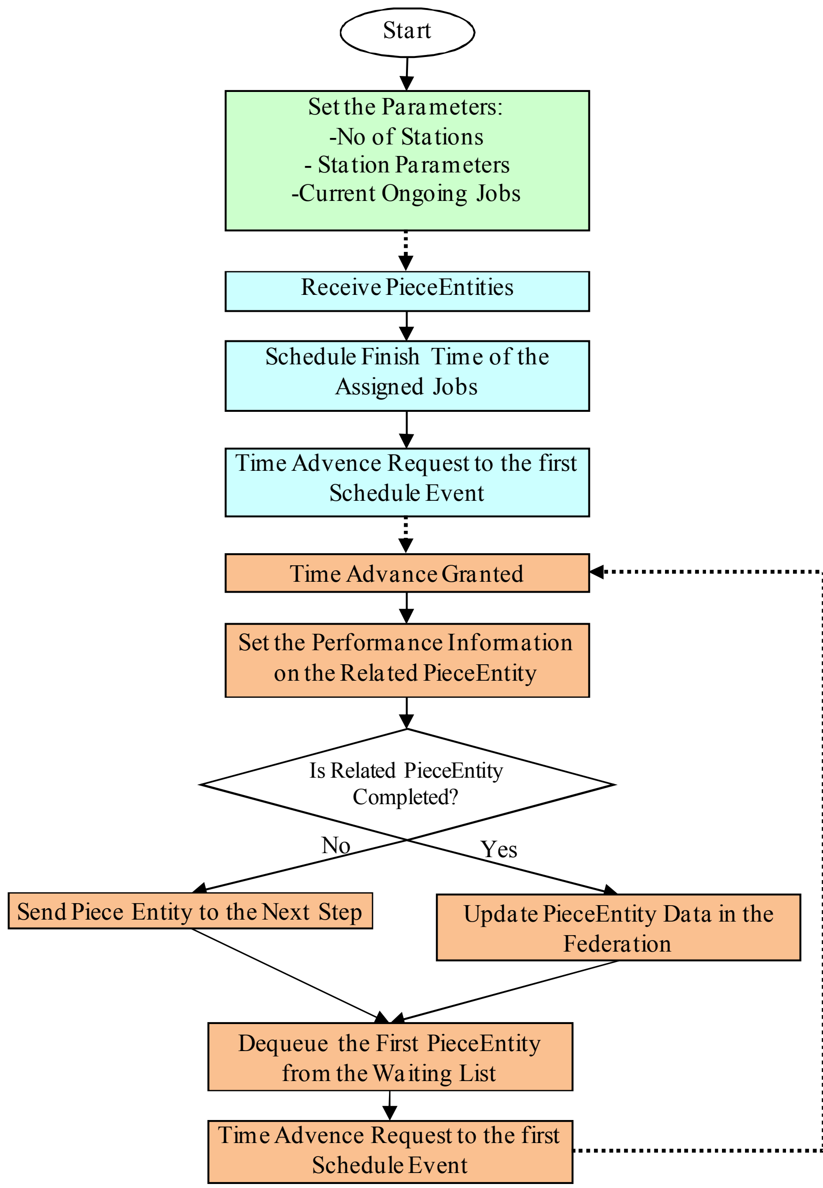


Figure C-10. Programming flowchart for the DES federate.

Interfaces for different federates

To facilitate working with the structural steel federation, several user friendly interfaces were developed which are discussed in below.

Data Management federate

This federate has been developed so that it can connect to the collaborating company's database. It generates information required by users or other federates such as: man-hours, schedule, material, number of pieces to be processed, etc. In other words, this federate provides information for different federates, and each federate is able to read/write data into it. The interface developed for the Data Management federate is shown in Figure C-11. The interface enables end users to set the simulation start time and settle the time interval the shop model requires to simulate. The weight of different fabrication operations can also be defined within this interface. During the simulation run, some managerial information such as delays, completed divisions, and finish time are shown on the interface. This federate also takes care of updating the database with project performance reports.

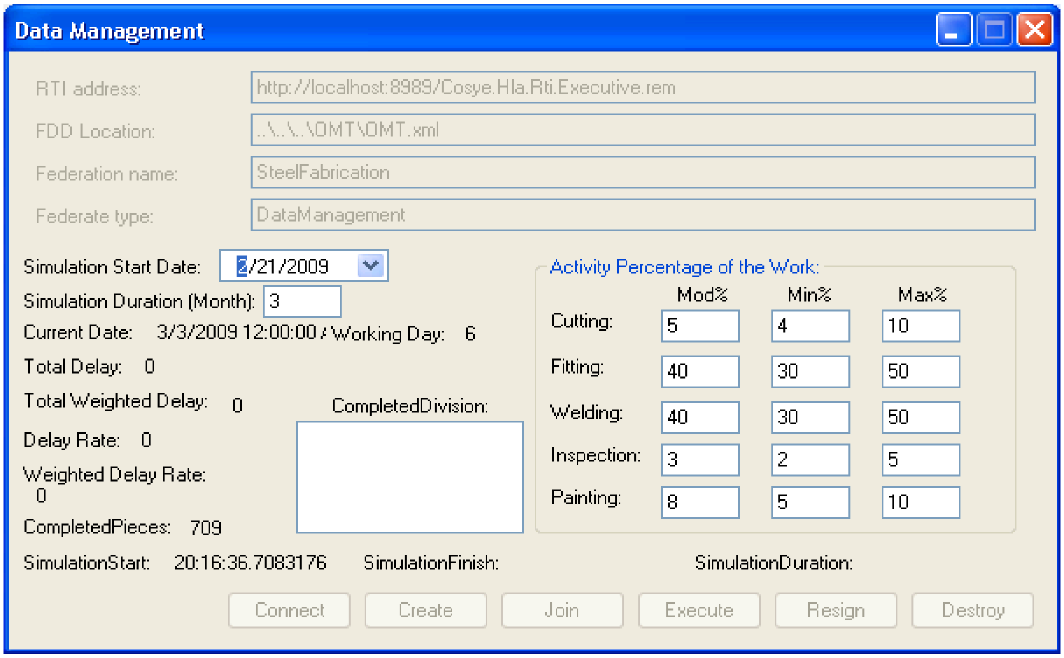


Figure C-11. The interface developed for the Data Management federate.

Calendar federate

As mentioned in Chapter 2 and Chapter 3 the calendar federate is responsible for keeping track of dates, shifts, and working and overtime hours. Holidays can be defined by users within this federate. The date information is published to the federation by the calendar federate which can be used by other federates within the federation. Such information can influence the simulation results particularly regarding forecasted project finish times. Figure C-12 depicts the interface form developed for the calendar federate.

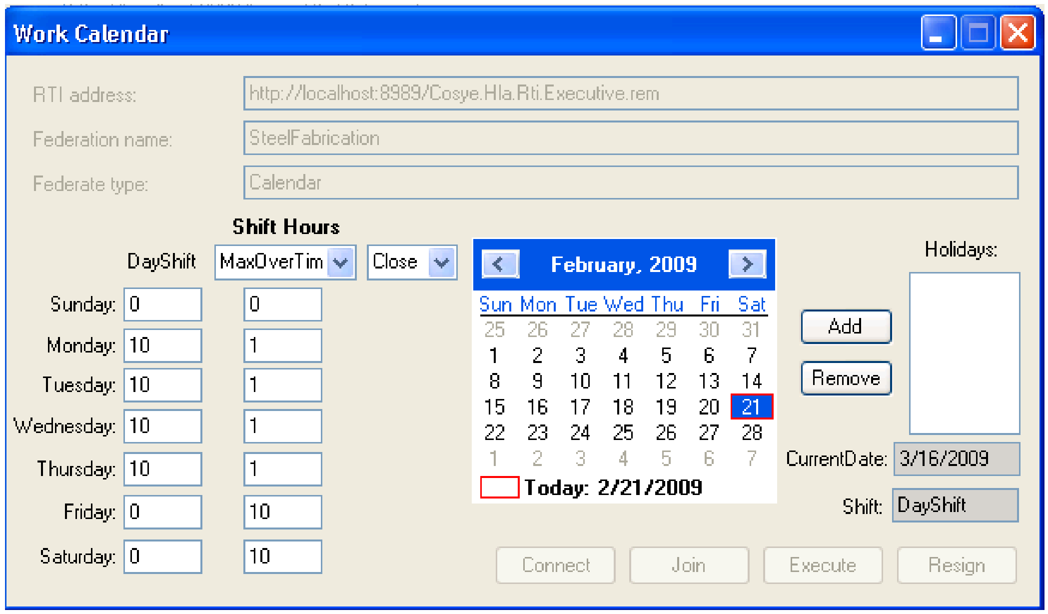


Figure C-12. Interface of the calendar federate.

Discrete Event Simulation (DES) federate

Details regarding what the Discrete Event Simulation (DES) federate does have been given in Chapter 2 and Chapter 3. The main interface of this federate has been shown in Figure 3-3. In the middle of the form different stations including cutting, fitting, welding, inspection and painting are depicted. Busy stations are shown in pink, while idle stations are shown in green. By clicking on one of the three buttons on the left side of the main DES form, another form pops up and users can set specifications regarding the stations (Figure C-13), mid buffers or storages (Figure C-14) and movers (Figure C-15).

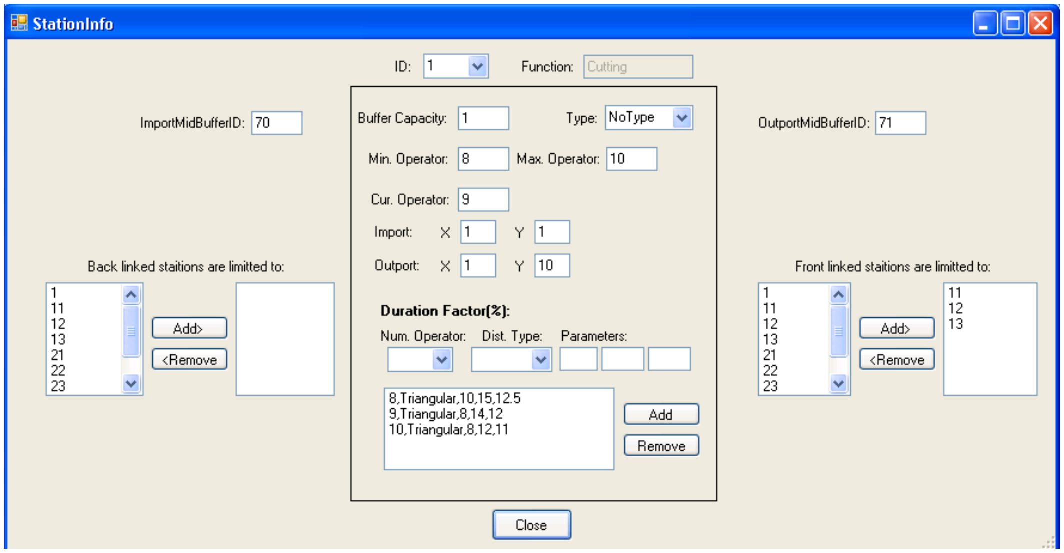


Figure C-13. Station information form for the DES federate.

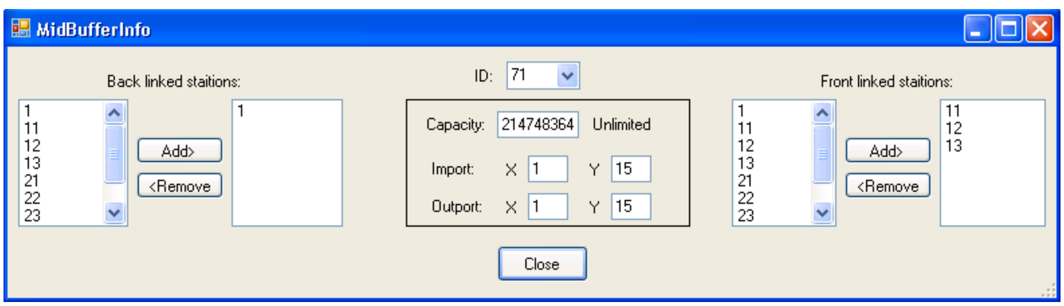


Figure C-14. Mid-buffer specifications form for the DES federate.

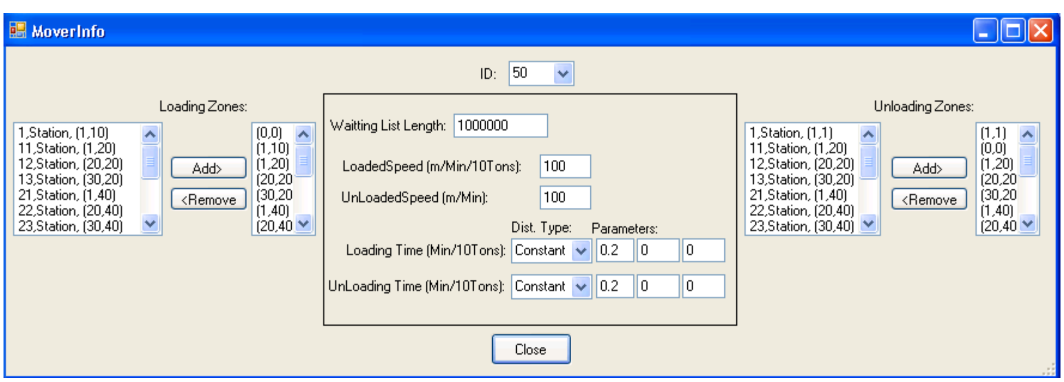


Figure C-15. Movers specifications form for the DES federate.

Visualization federate

As discussed in Chapter 2, this federate provides end users with intuitive 5D reports about multivariate information (i.e., schedule and cost) using a color coding concept (Figure 2-7). The fabricated and ongoing divisions along with the CPIs , SPIs and percent complete of the steel pieces forming the divisions are shown on separate lists on the interface (Figure C-16). From this list, the user can select any of the available divisions and load the corresponding 3D model to observe the status of the division. By selecting each division and then pushing the Change the Division button, the progress and performance indices of related pieces are shown in the list box on the right. By selecting the "Show in Tekla" radio button on the interface form the visualization federate runs software called Tekla, which is a structural steel 3D detailing package (http://www.tekla.com), and the most recent project performance information is presented to end users in a 5D format. Due to the fact that running Tekla significantly influences the federation by slowing down the simulation, the default selected radio button for the visualization federate is "Do not show in Tekla" which means that project performance information (i.e., CPI, SPI and progress) is presented in a tabular format within the right list box.

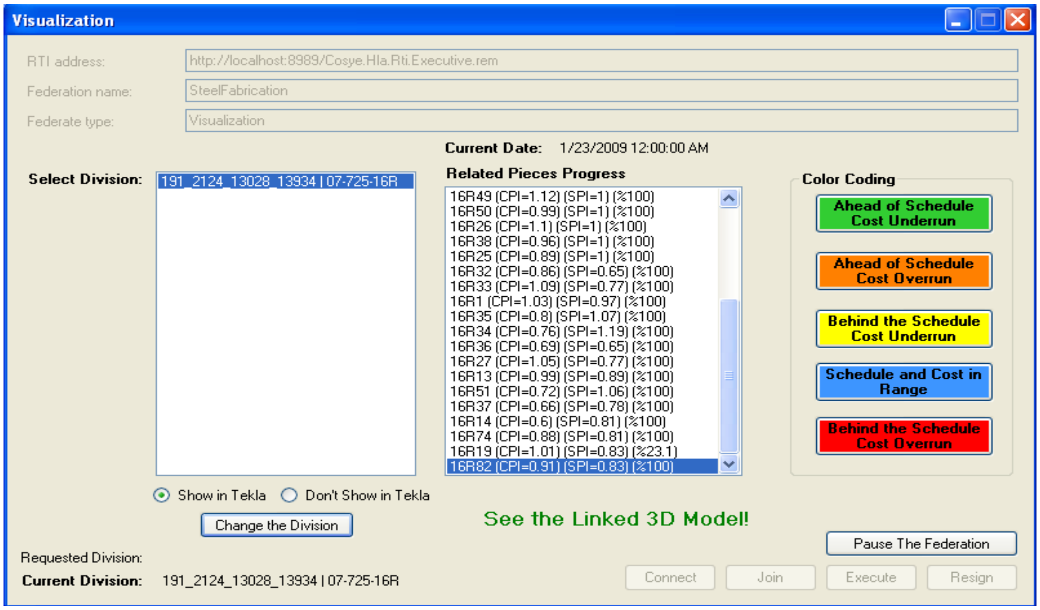


Figure C-16. Interface for the visualization federate.

Visual Basic Code

Although the end users work with the user friendly interfaces developed for the structural steel federation, many things are happening behind the scene and thousands of lines of code are required to make the federation work. The Visual Basic package of Visual Studio.NET 2008 has been used as the programming tool for developing different federates. In addition to the standard classes embedded in Visual Basic, the set of HLA related classes provided by the COSYE framework also have been imported and used in the program. The set of general purpose discrete event simulation classes, provided by Simphony.NET 3.5, also have been imported for development of the DES federate. In the following, the main body of the Visual Basic code used for developing different federates of the structural steel federation are presented. Details regarding the data acquisition federate have already been dealt with in Appendix B:

Calendar Federate

```
Imports Cosye.Hla.Rti
Public Class CalendarFederate
    Dim Holidays As New Collection
    Dim MyCalendar As Cosye.Steel.Steel_Calendar
    Dim WeekDayHours(6, 3) As Single
    Dim ShifInterval(3) As ShiftType
    Dim CurrentShiftID As Integer = 1
    Dim CurrentDayLeftHours As Double = 24

    Private Sub ReadWeekDayHours()
        Try
            WeekDayHours(0, 1) = CSng(TxtSunWork.Text)
            WeekDayHours(0, 2) = CSng(TxtSunOver.Text)
            WeekDayHours(0, 3) = Math.Max(0, 24 - WeekDayHours(0, 1) - WeekDayHours(0, 2))
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Sunday Hours!")
        End Try
        Try
            WeekDayHours(1, 1) = CSng(TxtMonWork.Text)
            WeekDayHours(1, 2) = CSng(TxtMonOver.Text)
            WeekDayHours(1, 3) = Math.Max(0, 24 - WeekDayHours(1, 1) - WeekDayHours(1, 2))
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Monday Hours!")
        End Try
        Try
            WeekDayHours(2, 1) = CSng(TxtTueWork.Text)
            WeekDayHours(2, 2) = CSng(TxtTueOver.Text)
            WeekDayHours(2, 3) = Math.Max(0, 24 - WeekDayHours(2, 1) - WeekDayHours(2, 2))
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Tuesday Hours!")
        End Try
        Try
            WeekDayHours(3, 1) = CSng(TxtWedWork.Text)
            WeekDayHours(3, 2) = CSng(TxtWedOver.Text)
            WeekDayHours(3, 3) = Math.Max(0, 24 - WeekDayHours(3, 1) - WeekDayHours(3, 2))
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Wednesday Hours!")
        End Try
        Try
            WeekDayHours(4, 1) = CSng(TxtThuWork.Text)
            WeekDayHours(4, 2) = CSng(TxtThuOver.Text)
            WeekDayHours(4, 3) = Math.Max(0, 24 - WeekDayHours(4, 1) - WeekDayHours(4, 2))
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Thursday Hours!")
        End Try
        Try
            WeekDayHours(5, 1) = CSng(TxtFriWork.Text)
            WeekDayHours(5, 2) = CSng(TxtFriOver.Text)
            WeekDayHours(5, 3) = Math.Max(0, 24 - WeekDayHours(5, 1) - WeekDayHours(5, 2))
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Friday Hours!")
        End Try
        Try
            WeekDayHours(6, 1) = CSng(TxtSatWork.Text)
            WeekDayHours(6, 2) = CSng(TxtSatOver.Text)
            WeekDayHours(6, 3) = Math.Max(0, 24 - WeekDayHours(6, 1) - WeekDayHours(6, 2))
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Saturday Hours!")
```

```
    End Try
End Sub

Private Sub ReadShiftInterval()
    ShifInterval(1) = ShiftType.DayShift
    Select Case CmbShift2.Text
        Case "Shift2"
            ShifInterval(2) = ShiftType.Shift2
        Case "MaxOverTime"
            ShifInterval(2) = ShiftType.OverTime
        Case "Close"
            ShifInterval(2) = ShiftType.Close
    End Select
    Select Case CmbShift3.Text
        Case "Shift3"
            ShifInterval(3) = ShiftType.Shift3
        Case "Close"
            ShifInterval(3) = ShiftType.Close
    End Select
End Sub

Private Sub BtnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnDelete.Click
    If LstHoliday.SelectedItems.Count > 0 Then
        Dim ToBeDeletedDate As New Date
        For i As Integer = LstHoliday.SelectedItems.Count - 1 To 0
            Holidays.Remove(CInt(LstHoliday.SelectedItems.Item(i)))
            LstHoliday.Items.Remove(LstHoliday.SelectedItems.Item(i))
        Next
    End If
End Sub

Private Sub BtnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnAdd.Click
    'Check If the picked hodliday has not been added yet
    If Not Holidays.Contains(LstPickHoliday.SelectionStart.ToShortDateString) Then
        Holidays.Add(LstPickHoliday.SelectionStart.Date, LstPickHoliday.SelectionStart.ToShortDateString)
        If LstHoliday.Items.Count = 0 Then 'No need for sorting
            LstHoliday.Items.Add(LstPickHoliday.SelectionStart.ToShortDateString)
        Else 'Item should be sorted
            Dim AddedToList As Boolean = False
            For i As Integer = 0 To LstHoliday.Items.Count - 1
                If Convert.ToDateTime(LstHoliday.Items(i)).Date > LstPickHoliday.SelectionStart.Date Then
                    LstHoliday.Items.Insert(i, LstPickHoliday.SelectionStart.ToShortDateString)
                    AddedToList = True
                    Exit For
                End If
            Next
            'Add to the list as the last item
            If Not AddedToList Then
                LstHoliday.Items.Add(LstPickHoliday.SelectionStart.ToShortDateString)
            End If
        End If
    End If
End Sub

Dim LastWorkingDate As Date = Nothing
Private Sub ChangeTheShift()
    MyCalendar.CurrentShiftHours = 0
    While MyCalendar.CurrentShiftHours = 0
        If MyCalendar.CurrentShiftType = ShiftType.Close Or _
        MyCalendar.CurrentShiftType = ShiftType.Shift3 Then 'Next shift is in tomorrow
            MyCalendar.CurrentDate = DateAdd(DateInterval.Day, 1, MyCalendar.CurrentDate)
            'Check if tomorrow is holiday
            If Holidays.Contains(MyCalendar.CurrentDate.ToShortDateString) Then 'Set all day as Close
                MyCalendar.CurrentShiftType = ShiftType.Close
                MyCalendar.CurrentShiftHours = 24
                CurrentDayLeftHours = 0
            Else 'Start from first shift of the day
                ReadWeekDayHours()
                Dim DayInWeek As Integer = MyCalendar.CurrentDate.DayOfWeek
```

```
                CurrentShiftID = 1
                MyCalendar.CurrentShiftHours = WeekDayHours(DayInWeek, CurrentShiftID)
                MyCalendar.CurrentShiftType = ShifInterval(CurrentShiftID)
                CurrentDayLeftHours = 24 - MyCalendar.CurrentShiftHours
            End If
        Else 'Next shift is in today
            Dim DayInWeek As Integer = MyCalendar.CurrentDate.DayOfWeek
            CurrentShiftID = CurrentShiftID + 1
            MyCalendar.CurrentShiftType = ShifInterval(CurrentShiftID)
            'Set the over time base on the desired overtime and max overtime
            If MyCalendar.CurrentShiftType = ShiftType.OverTime Then
                Try
                    MyCalendar.CurrentShiftHours = Math.Min(MyCalendar.DesireOverTime, WeekDayHours(DayInWeek,
CurrentShiftID))
                Catch ex As Exception
                    MyCalendar.CurrentShiftHours = 0
                End Try
            ElseIf MyCalendar.CurrentShiftType = ShiftType.Close Then
                MyCalendar.CurrentShiftHours = Math.Max(CurrentDayLeftHours, WeekDayHours(DayInWeek,
CurrentShiftID))
            Else
                MyCalendar.CurrentShiftHours = WeekDayHours(DayInWeek, CurrentShiftID)
            End If
            CurrentDayLeftHours = CurrentDayLeftHours - MyCalendar.CurrentShiftHours
        End If
    End While
    'Check if the working days have been counted
    If Not (MyCalendar.CurrentShiftType = ShiftType.Close Or LastWorkingDate = MyCalendar.CurrentDate) Then
        MyCalendar.DayNo = MyCalendar.DayNo + 1
    End If

End Sub

Private Sub MyCalendarFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyCalendarFactory.ReflectAttributeValues
    'Sets the Parameters
    ReadWeekDayHours()
    ReadShiftInterval()
    'Determine the Max possible OverTime
    Dim MaxOvertime As Double = 0
    For i As Integer = 1 To 7
        If WeekDayHours(i - 1, 2) > MaxOvertime Then
            MaxOvertime = WeekDayHours(i, 2)
        End If
    Next
    'Handle the update attributes
    MyCalendar = MyCalendarFactory(e.theObject)
    'Get the CurrentDate attribute handle
    Dim CurDateHandle As AttributeHandle = MyCalendarFactory.GetAttributeHandle("CurrentDate")
    'Just do it for the first time when Current time is defined
    If e.theValues.Contains(CurDateHandle) Then
        MyCalendar.AttributeOwnershipAcquisition("CurrentDate", "CurrentShiftHours", "CurrentShiftType", "DayNo",
"MaxOverTime", "SetOverTimeForDay")
        MyCalendar.DayNo = 0
        CurrentShiftID = 0
        MyCalendar.MaxOverTime = MaxOvertime
        If Holidays.Contains(MyCalendar.CurrentDate.ToShortDateString) Then
            MyCalendar.CurrentShiftType = ShiftType.Close
            MyCalendar.CurrentShiftHours = 24
            CurrentDayLeftHours = 0
        Else
            Dim DayInWeek As Integer = MyCalendar.CurrentDate.DayOfWeek
            MyCalendar.CurrentShiftHours = 0
            'Change the shift until ShiftHours becomes > 0
            CurrentDayLeftHours = 24
            While MyCalendar.CurrentShiftHours = 0
                CurrentShiftID = CurrentShiftID + 1
                MyCalendar.CurrentShiftType = ShifInterval(CurrentShiftID)
                If MyCalendar.CurrentShiftType = ShiftType.OverTime Then
```

```
                    Try
                        MyCalendar.CurrentShiftHours = Math.Min(MyCalendar.DesireOverTime, WeekDayHours(DayInWeek,
CurrentShiftID))
                    Catch ex As Exception
                        MyCalendar.CurrentShiftHours = 0
                    End Try
                ElseIf MyCalendar.CurrentShiftType = ShiftType.Close Then
                    MyCalendar.CurrentShiftHours = Math.Max(CurrentDayLeftHours, WeekDayHours(DayInWeek,
CurrentShiftID))
                Else
                    MyCalendar.CurrentShiftHours = WeekDayHours(DayInWeek, CurrentShiftID)
                End If
            End While
            CurrentDayLeftHours = CurrentDayLeftHours - MyCalendar.CurrentShiftHours
        End If
        MyCalendar.UpdateAttributeValues()
    End If
    'Handle the update attributes
    Dim DesireOvertimeHandle As AttributeHandle = MyCalendarFactory.GetAttributeHandle("DesireOverTime")
    'Get the DesireOvertime attribute handle
    If e.theValues.Contains(DesireOvertimeHandle) Then
        If MyCalendar.DesireOverTime > 0 Then
            MyCalendar.SetOverTimeForDay = Math.Min(MyCalendar.DesireOverTime,
WeekDayHours(MyCalendar.CurrentDate.DayOfWeek, 2))
        Else
            MyCalendar.SetOverTimeForDay = 0
        End If
        'Just for resolving possible the ownership
        Dim GainedOwnership As Boolean = False
        'Loop until ownership is granted
        While Not GainedOwnership
            Try
                MyCalendar.UpdateAttributeValues()
                GainedOwnership = True
            Catch ex As Exception

            End Try
        End While
    End If
    'Update the Interface
    Try
        TxtCurDate.Text = MyCalendar.CurrentDate.ToShortDateString
        TxtShift.Text = [Enum].GetName(GetType(Cosye.Steel.ShiftType), MyCalendar.CurrentShiftType)
    Catch ex As Exception
        'No need for interface update at this time
    End Try
End Sub

Private Sub fedAmb_TimeAdvanceGrant(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.TimeAdvanceGrantEventArgs) Handles fedAmb.TimeAdvanceGrant
    'Update the Interface
    TxtCurDate.Text = MyCalendar.CurrentDate.ToShortDateString
    TxtShift.Text = [Enum].GetName(GetType(Cosye.Steel.ShiftType), MyCalendar.CurrentShiftType)
    'Store the current shift's Hours
    Dim CurrentShiftHours As Double = MyCalendar.CurrentShiftHours
    'Prepare the parameters of the next shift
    ChangeTheShift()
    'Update the next shift when current shift gets over
    MyCalendar.UpdateAttributeValues(e.theTime + CurrentShiftHours * 3600)
    'Time Advance Request when current shift gets over
    rtiAmb.TimeAdvanceRequest(e.theTime + CurrentShiftHours * 3600)
End Sub
End Class
```

Data Management Federate

```
Imports Simphony.Mathematics
```

```vb
Imports Cosye.Hla.Rti

Public Class DataManagement
    'Dim PieceEntityItem As PieceEntity
    Dim Cutting, Fitting, FitInspection, Welding, WeldInspection, Painting As Activity
    Dim NoPieceReceived As Integer = 0
    Dim MyShopProductivity As Steel_ShopProductivity
    Dim SimulationStartTime As New Date
    Dim MyDataEnvironment As New DataEnvironment
    Dim DivisionsProgressList As New Dictionary(Of String, String)
    Dim InShopPieceSent As Boolean = False

    Public Sub SetParameters()
        ''''Date Related Parameters
        Try
            MyDataEnvironment.DataImportDuration = Val(TxtDuration.Text)
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number for Simulation Duration!")
        End Try
        ''''Activity Related Parameters
        Try
            Cutting.ActivityMod = Convert.ToDouble(TxtCutMod.Text)
            Cutting.ActivityMin = Convert.ToDouble(TxtCutMin.Text)
            Cutting.ActivityMax = Convert.ToDouble(TxtCutMax.Text)
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number For Cutting!")
        End Try
        Try
            Fitting.ActivityMod = Convert.ToDouble(TxtFitMod.Text)
            Fitting.ActivityMin = Convert.ToDouble(TxtFitMin.Text)
            Fitting.ActivityMax = Convert.ToDouble(TxtFitMax.Text)
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number For Fitting!")
        End Try
        Try
            Welding.ActivityMod = Convert.ToDouble(TxtWeldMod.Text)
            Welding.ActivityMin = Convert.ToDouble(TxtWeldMin.Text)
            Welding.ActivityMax = Convert.ToDouble(TxtWeldMax.Text)
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number For Welding!")
        End Try
        Try
            FitInspection.ActivityMod = Val(TxtInspectMod.Text) / 2
            FitInspection.ActivityMin = Val(TxtInspectMin.Text) / 2
            FitInspection.ActivityMax = Val(TxtInspectMax.Text) / 2
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number For Inspection!")
        End Try
        Try
            WeldInspection.ActivityMod = Val(TxtInspectMod.Text) / 2
            WeldInspection.ActivityMin = Val(TxtInspectMin.Text) / 2
            WeldInspection.ActivityMax = Val(TxtInspectMax.Text) / 2
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number For Inspection!")
        End Try
        Try
            Painting.ActivityMod = Convert.ToDouble(TxtPaintMod.Text)
            Painting.ActivityMin = Convert.ToDouble(TxtPaintMin.Text)
            Painting.ActivityMax = Convert.ToDouble(TxtPaintMax.Text)
        Catch ex As Exception
            MessageBox.Show("Enter a valid Number For Painting!")
        End Try
        'Set the Simulation Start Time
        LblSimStart.Text = Date.Now.TimeOfDay.ToString
        SimulationStartTime = Date.Now
    End Sub
    '' '''''''''''''''''''Sends Entities
    Private Sub SendEntities(ByVal SendDate As Date, ByVal SendingTime As Double)
        'Import the Data for the Date
```

```
        MyDataEnvironment.ImportData(SendDate)
        Dim DivisionItem As Division
        'Send the imported pieces for today
        For Each PieceItem As Piece In MyDataEnvironment.TodayPieceList.Values
            DivisionItem = New Division
            DivisionItem = MyDataEnvironment.DivisionList(PieceItem.DivisionID)
            Dim PaintingRequirePortion As Double = 1
            Dim PiecePaintingActivityDurationPortion As Double = 1 'Is set to calculate Painting Portion
            If Not DivisionItem.PaintRequire Then
                PaintingRequirePortion = 100 / (100 - Painting.ActivityMod)
                PiecePaintingActivityDurationPortion = 0
            End If
            Dim PieceActivityDuration As Double = Math.Min(Math.Max(DivisionItem.FabManHour * PieceItem.Weight,
200), 5 * 3600)
            PiecePaintingActivityDurationPortion = PiecePaintingActivityDurationPortion * PieceActivityDuration
            Dim RFIDTagCount As Integer = 0
            If MyDataEnvironment.InShopPieceList.Keys.Contains(PieceItem.PieceID) Then
                RFIDTagCount = PieceItem.RFIDList.Count
            End If
            For j As Integer = 1 To PieceItem.Quantity - RFIDTagCount
                Dim NewPieceEntity As Steel_PieceEntity = MyPieceEntityFactory.RegisterObjectInstance()
                NewPieceEntity.PieceID = PieceItem.PieceID
                NewPieceEntity.StartDate = SendDate.Date
                NewPieceEntity.Weight = PieceItem.Weight
                NewPieceEntity.DimentionLevel = Convert.ToInt32(PieceItem.Weight)
                NewPieceEntity.CuttingFinish = Nothing
                NewPieceEntity.CuttingManHour = PieceActivityDuration * Cutting.DurationPortion * PaintingRequirePortion
                NewPieceEntity.WeldingManHour = PieceActivityDuration * Welding.DurationPortion *
PaintingRequirePortion
                NewPieceEntity.FittingManHour = PieceActivityDuration * Fitting.DurationPortion * PaintingRequirePortion
                NewPieceEntity.FitInspectionManHour = PieceActivityDuration * FitInspection.DurationPortion *
PaintingRequirePortion
                NewPieceEntity.WeldInspectionManHour = PieceActivityDuration * WeldInspection.DurationPortion *
PaintingRequirePortion
                NewPieceEntity.PaintingManHour = PiecePaintingActivityDurationPortion * Painting.DurationPortion *
PaintingRequirePortion
                If SendingTime > 0 Then
                    NewPieceEntity.UpdateAttributeValues(SendingTime)
                    NewPieceEntity.UnconditionalAttributeOwnershipDivestiture("CuttingManHour", "FittingManHour",
"WeldingManHour", "FitInspectionManHour", "WeldInspectionManHour", "PaintingManHour", _
                    "CuttingFinish", "FittingFinish", "WeldingFinish", "FitInspectionFinish", "WeldInspectionFinish",
"PaintingFinish", _
                    "CuttingStart", "FittingStart", "WeldingStart", "FitInspectionStart", "WeldInspectionStart", "PaintingStart")
                Else
                    NewPieceEntity.UpdateAttributeValues()
                    NewPieceEntity.UnconditionalAttributeOwnershipDivestiture("CuttingManHour", "FittingManHour",
"WeldingManHour", "FitInspectionManHour", "WeldInspectionManHour", "PaintingManHour", _
                    "CuttingFinish", "FittingFinish", "WeldingFinish", "FitInspectionFinish", "WeldInspectionFinish",
"PaintingFinish", _
                    "CuttingStart", "FittingStart", "WeldingStart", "FitInspectionStart", "WeldInspectionStart", "PaintingStart")
                End If
            Next
        Next
    End Sub

    ' '''''''''''''''Sends In-Shop Entities
    Private Sub SendInShopEntities(ByVal SendDate As Date)
        'Import the Data for the Date
        MyDataEnvironment.ImportInShopPieces(SendDate)
        Dim DivisionItem As Division
        Dim MaxNumberOfRFIDUpdate As Integer = 3
        'Send the imported pieces for today
        For Each PieceItem As Piece In MyDataEnvironment.InShopPieceList.Values
            DivisionItem = New Division
            DivisionItem = MyDataEnvironment.DivisionList(PieceItem.DivisionID)
            Dim PaintingRequirePortion As Double = 1
            Dim PiecePaintingActivityDurationPortion As Double = 1 'Is set to calculate Painting Portion
            If Not DivisionItem.PaintRequire Then
                PaintingRequirePortion = 100 / (100 - Painting.ActivityMod)
```

```
            PiecePaintingActivityDurationPortion = 0
        Else
            MaxNumberOfRFIDUpdate = 1 + MaxNumberOfRFIDUpdate
        End If
        Dim PieceActivityDuration As Double = DivisionItem.FabManHour * PieceItem.Portion
        PiecePaintingActivityDurationPortion = PiecePaintingActivityDurationPortion * PieceActivityDuration
        For j As Integer = 0 To PieceItem.RFIDList.Count - 1
            Dim RFIDItem As New RFID
            RFIDItem = PieceItem.RFIDList(PieceItem.RFIDList.Keys(j))
            If MaxNumberOfRFIDUpdate > RFIDItem.RFIDCount Then
                Dim NewPieceEntity As Steel_PieceEntity = MyPieceEntityFactory.RegisterObjectInstance()
                NewPieceEntity.PieceID = PieceItem.PieceID
                NewPieceEntity.StartDate = SendDate.Date
                NewPieceEntity.Weight = PieceItem.Weight
                NewPieceEntity.DimentionLevel = Convert.ToInt32(PieceItem.Weight)
                NewPieceEntity.CuttingManHour = PieceActivityDuration * Cutting.DurationPortion *
PaintingRequirePortion
                NewPieceEntity.WeldingManHour = PieceActivityDuration * Welding.DurationPortion *
PaintingRequirePortion
                NewPieceEntity.FittingManHour = PieceActivityDuration * Fitting.DurationPortion * PaintingRequirePortion
                NewPieceEntity.FitInspectionManHour = PieceActivityDuration * FitInspection.DurationPortion *
PaintingRequirePortion
                NewPieceEntity.WeldInspectionManHour = PieceActivityDuration * WeldInspection.DurationPortion *
PaintingRequirePortion
                NewPieceEntity.PaintingManHour = PiecePaintingActivityDurationPortion * Painting.DurationPortion *
PaintingRequirePortion
                'Reset the values of the activity start and finish
                NewPieceEntity.CuttingFinish = Nothing
                NewPieceEntity.FittingFinish = Nothing
                NewPieceEntity.FitInspectionFinish = Nothing
                NewPieceEntity.WeldingFinish = Nothing
                NewPieceEntity.WeldInspectionFinish = Nothing
                NewPieceEntity.PaintingFinish = Nothing
                NewPieceEntity.CuttingStart = Nothing
                NewPieceEntity.FittingStart = Nothing
                NewPieceEntity.FitInspectionStart = Nothing
                NewPieceEntity.WeldingStart = Nothing
                NewPieceEntity.WeldInspectionStart = Nothing
                NewPieceEntity.PaintingStart = Nothing

                Select Case RFIDItem.RFIDCount
                    Case 1
                        NewPieceEntity.CuttingFinish = RFIDItem.MinDate
                    Case 2
                        NewPieceEntity.CuttingFinish = RFIDItem.MinDate
                        NewPieceEntity.FittingFinish = RFIDItem.MaxDate
                        NewPieceEntity.FitInspectionFinish = RFIDItem.MaxDate
                    Case 3
                        NewPieceEntity.CuttingFinish = RFIDItem.MinDate
                        NewPieceEntity.FittingFinish = RFIDItem.MinDate
                        NewPieceEntity.FitInspectionFinish = RFIDItem.MinDate
                        NewPieceEntity.WeldingStart = RFIDItem.MaxDate
                    Case 4
                        NewPieceEntity.CuttingFinish = RFIDItem.MinDate
                        NewPieceEntity.FittingFinish = RFIDItem.MinDate
                        NewPieceEntity.FitInspectionFinish = RFIDItem.MinDate
                        NewPieceEntity.WeldingFinish = RFIDItem.MaxDate
                        NewPieceEntity.WeldInspectionFinish = RFIDItem.MaxDate
                    Case 5
                        NewPieceEntity.CuttingFinish = RFIDItem.MinDate
                        NewPieceEntity.FittingFinish = RFIDItem.MinDate
                        NewPieceEntity.FitInspectionFinish = RFIDItem.MinDate
                        NewPieceEntity.WeldingFinish = RFIDItem.MaxDate
                        NewPieceEntity.WeldInspectionFinish = RFIDItem.MaxDate
                        NewPieceEntity.PaintingStart = RFIDItem.MaxDate
                End Select
                NewPieceEntity.UpdateAttributeValues()
                NewPieceEntity.UnconditionalAttributeOwnershipDivestiture("CuttingManHour", "FittingManHour",
"WeldingManHour", "FitInspectionManHour", "WeldInspectionManHour", "PaintingManHour", _
```

```
                "CuttingFinish", "FittingFinish", "WeldingFinish", "FitInspectionFinish", "WeldInspectionFinish",
"PaintingFinish", _
                "CuttingStart", "FittingStart", "WeldingStart", "FitInspectionStart", "WeldInspectionStart", "PaintingStart")
            Else 'MaxNumberOfRFIDUpdate < =RFIDItem.RFIDCount
                ''Current piece progress
                PieceItem.CompletedPieces = 1 + PieceItem.CompletedPieces
            End If
        Next
    Next
End Sub

Private Sub MyPieceEntityFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyPieceEntityFactory.ReflectAttributeValues
    Dim CompletedPieceEntity As Steel_PieceEntity = MyPieceEntityFactory(e.theObject)
    'Check if the Piece Exists'''Sometimes the piece will be igonred if its weight is Zero
    If MyDataEnvironment.PieceList.Keys.Contains(CompletedPieceEntity.PieceID) Then
        Dim RelatedPiece As Piece = MyDataEnvironment.PieceList(CompletedPieceEntity.PieceID)
        Dim RelatedDivision As Division = MyDataEnvironment.DivisionList(RelatedPiece.DivisionID)
        'Calculate and send piece CPI at piece level
        ''Current piece progress
        RelatedPiece.CompletedPieces = 1 + RelatedPiece.CompletedPieces
        Dim PieceProgress As Double = RelatedPiece.CompletedPieces / RelatedPiece.Quantity
        ''Piece CPI
        Dim PieceSpentHour As Double = (CompletedPieceEntity.CuttingManHour + _
                        CompletedPieceEntity.FittingManHour + _
                        CompletedPieceEntity.WeldingManHour + _
                        CompletedPieceEntity.FitInspectionManHour + _
                        CompletedPieceEntity.WeldInspectionManHour + _
                        CompletedPieceEntity.PaintingManHour) ' In Second
        RelatedPiece.TotalSpentHours = PieceSpentHour + RelatedPiece.TotalSpentHours
        RelatedDivision.TotalSpentHours = PieceSpentHour + RelatedDivision.TotalSpentHours
        Dim PieceEarnedHour As Double = RelatedPiece.CompletedPieces * RelatedDivision.FabManHour _
                                    * RelatedPiece.Portion 'In second
        Dim PieceCPI = PieceEarnedHour / RelatedPiece.TotalSpentHours
        'Check the Division CPI and Both piece and division SPI
        ''Check the start date
        If RelatedDivision.Progress = 0 Then
            If Not (CompletedPieceEntity.CuttingStart = Nothing) Then
                RelatedDivision.Start = CompletedPieceEntity.CuttingStart
            Else
                RelatedDivision.Start = MyDataEnvironment.CurrentDate
            End If
        End If
        'Update No of Pieces
        NoPieceReceived = NoPieceReceived + 1
        RelatedDivision.CompletedPieces = RelatedDivision.CompletedPieces + 1
        'Update the Piece ListBox
        LblCompPiece.Text = NoPieceReceived.ToString
        ''Update the current earned progress at the division level
        RelatedDivision.Progress = Math.Min(1, RelatedDivision.Progress + RelatedPiece.Portion)
        '' Calculate Division CPI
        Dim DivisionCPI As Double = 1
        If RelatedDivision.TotalSpentHours > 0 Then
            Dim DivisionEarnedManhour As Double = RelatedDivision.Progress * RelatedDivision.FabManHour
            DivisionCPI = DivisionEarnedManhour / RelatedDivision.TotalSpentHours
        End If
        'Calculate Scheduled Progress
        Dim SchPassedDays As Double
        If RelatedDivision.Start < MyDataEnvironment.CurrentDate Then
            SchPassedDays = Math.Max(0, DateDiff(DateInterval.Day, RelatedDivision.Start,
MyDataEnvironment.CurrentDate))
        Else
            SchPassedDays = 0
        End If
        Dim SchTotalDays As Double = Math.Max(1, DateDiff(DateInterval.Day, RelatedDivision.Start,
RelatedDivision.Required))
        Dim SchProgress As Double = Math.Min(1, SchPassedDays / SchTotalDays)
        'Calculate SPI
        Dim DivisionPieceSPI As Double
```

```
            If SchProgress = 0 Then
                DivisionPieceSPI = 1
            Else
                DivisionPieceSPI = RelatedDivision.Progress / SchProgress
            End If
            'Reflec latest CPI and SPI to the related Division
            RelatedDivision.CPI = DivisionCPI
            RelatedDivision.SPI = DivisionPieceSPI
            'Update VPiece
            Dim MyVpiece As Steel_VPiece = MyVPieceFactory.RegisterObjectInstance()
            MyVpiece.DivisionID = RelatedDivision.DivisionID
            MyVpiece.PieceKey = RelatedPiece.PieceKey
            MyVpiece.CPI = PieceCPI
            MyVpiece.SPI = DivisionPieceSPI
            MyVpiece.Progress = PieceProgress
            MyVpiece.UpdateAttributeValues()
            ''Report the latest achieved progress
            Dim DivPreviousProgress As String
            If DivisionsProgressList.Keys.Contains(RelatedDivision.DivisionID) Then
                'Remove the previous progress and add the new one
                DivPreviousProgress = DivisionsProgressList(RelatedDivision.DivisionID)
                DivisionsProgressList.Remove(RelatedDivision.DivisionID)
                LstCompDivision.Items.Remove(DivPreviousProgress)
                DivisionsProgressList.Add(RelatedDivision.DivisionID, RelatedDivision.ToString)
                LstCompDivision.Items.Add(RelatedDivision.ToString)
            Else 'Just add the new progress
                DivisionsProgressList.Add(RelatedDivision.DivisionID, RelatedDivision.ToString)
                LstCompDivision.Items.Add(RelatedDivision.ToString)
            End If
            Me.Refresh()
            '' Add number of completed divisions if division completed
            If Math.Round(RelatedDivision.Progress, 3) >= 1 And Not
MyDataEnvironment.CompletedDivisions.Contains(RelatedDivision.DivisionID) Then
                MyDataEnvironment.CompletedDivisions.Add(RelatedDivision.DivisionID)
            End If
        End If 'Piece Exists
        'Delete the Piece from the RTI
        MyPieceEntityFactory.DeleteObjectInstance(CompletedPieceEntity)
    End Sub

    Private Sub MyCalendarFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyCalendarFactory.ReflectAttributeValues
        Dim DateHandle As Cosye.Hla.Rti.AttributeHandle = MyCalendarFactory.GetAttributeHandle("CurrentDate")
        Dim NewDay As Steel_Calendar = MyCalendarFactory(e.theObject)
        'Move the day ahead and check the scheduled pieces
        MyDataEnvironment.CurrentDate = NewDay.CurrentDate
        MyDataEnvironment.WorkingDays = NewDay.DayNo
        'Check if the day no has been changed/ New day has been started
        If e.theValues.Contains(DateHandle) And NewDay.CurrentDate <= MyDataEnvironment.FinishDate Then
            'Send the Scheduled pieces for this day if any existed
            SendEntities(MyDataEnvironment.CurrentDate, e.theTime + 1)
            'Calculate the Delay
            CalculateCurrentDelay()
            'Update the Delay
            MyShopProductivity.TotalDelay = MyDataEnvironment.MySchedule.TotalDelay
            MyShopProductivity.DelayRate = MyDataEnvironment.DelayRate
            MyShopProductivity.UpdateAttributeValues()
            'Set the Values on screen
            SetInterface()
        End If
    End Sub
    '' ''''''''''''''''''Calculate and set delays
    Private Sub CalculateCurrentDelay()
        Dim CurrentTotalDelay As New Delay
        'Calculate curretn total delay
        For Each DivisionItem As Division In MyDataEnvironment.DivisionList.Values
            If DivisionItem.Progress < 1 And DivisionItem.Required < MyDataEnvironment.CurrentDate And Not
DivisionItem.Required = Nothing Then
```

```
                DivisionItem.Delay = Math.Min(0, DateDiff(DateInterval.Day, MyDataEnvironment.CurrentDate,
DivisionItem.Required))
            End If
            CurrentTotalDelay.Delays = CurrentTotalDelay.Delays + DivisionItem.Delay
            CurrentTotalDelay.WeightedDelays = CurrentTotalDelay.WeightedDelays + DivisionItem.WeightedDelay
        Next
        'Assign the caculated delay
        MyDataEnvironment.SetDelay(CurrentTotalDelay)
    End Sub

    Private Sub MyCalendarFactory_InitializeInitialInstances(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyCalendarFactory.InitializeInitialInstances
        'Sets the Entered Paramers
        SetParameters()
        'Reads the start date and sets it for My Data Environment
        MyDataEnvironment.StartDate = (CmbStart.Value.Date)
        Dim DataImportDuration As Double = Val(TxtDuration.Text)
        MyDataEnvironment.FinishDate = DateAdd(DateInterval.Month, DataImportDuration,
MyDataEnvironment.StartDate)
        MyDataEnvironment.FinishDate = DateAdd(DateInterval.Day, (DataImportDuration Mod 1) * 30,
MyDataEnvironment.FinishDate)
        MyDataEnvironment.CurrentDate = MyDataEnvironment.StartDate
        'Set the entered date as the first day of Federation
        Dim StartDay As Steel_Calendar = MyCalendarFactory.RegisterObjectInstance()
        StartDay.StartDate = MyDataEnvironment.StartDate
        StartDay.CurrentDate = MyDataEnvironment.CurrentDate
        StartDay.SetOverTimeForDay = 0
        StartDay.UpdateAttributeValues()
        'Divest the ownership of the calendar attribute
        StartDay.UnconditionalAttributeOwnershipDivestiture("CurrentShiftType", "CurrentShiftHours", "CurrentDate",
"SetOverTimeForDay")
        ''Send in shop entities
        If Not InShopPieceSent Then
            SendInShopEntities(MyDataEnvironment.CurrentDate)
            InShopPieceSent = True
        End If
        ''Send initial piece entities
        SendEntities(MyDataEnvironment.CurrentDate, 0)
        ''Set Interface
        SetInterface()
    End Sub

    Private Sub fedAmb_TimeAdvanceGrant(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.TimeAdvanceGrantEventArgs) Handles fedAmb.TimeAdvanceGrant
        'Federation Termination Condition
        If e.theTime > 0 And MyDataEnvironment.CompletedDivisions.Count >= MyDataEnvironment.DivisionList.Count _
        And MyDataEnvironment.CurrentDate > MyDataEnvironment.FinishDate Then
            'Set the Simulation Finish Time
            LblSimFinish.Text = Date.Now.TimeOfDay.ToString
            LblSimDuration.Text = (-DateDiff(DateInterval.Second, Date.Now, SimulationStartTime) / 60).ToString
            MyDataEnvironment.MySchedule.SimDuration = (DateDiff(DateInterval.Second, Date.Now, SimulationStartTime)
/ 60)
            'Report total Results
            MyDataEnvironment.ReportSchedule()
            rtiAmb.ReadyToTerminate()
        Else
            rtiAmb.NextMessageRequest(e.theTime + 24 * 3600)
        End If
    End Sub

    Private Sub MyShopProductivityFactory_RegisterInitialInstances(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyShopProductivityFactory.RegisterInitialInstances
        MyShopProductivity = MyShopProductivityFactory.RegisterObjectInstance
        'Calculate initial Delay
        CalculateCurrentDelay()
        'Update the Delay
        MyShopProductivity.TotalDelay = MyDataEnvironment.MySchedule.TotalDelay
        MyShopProductivity.DelayRate = MyDataEnvironment.DelayRate
        MyShopProductivity.UpdateAttributeValues()
```

```
    End Sub

    ' Set Interface
    Private Sub SetInterface()
        LblCurDate.Text = MyDataEnvironment.CurrentDate.Date.ToString
        LblTotalDelay.Text = MyDataEnvironment.MySchedule.TotalDelay.ToString
        LblTotalWDelay.Text = MyDataEnvironment.MySchedule.TotalWeightedDelay.ToString
        LblDelayRate.Text = MyDataEnvironment.DelayRate.ToString
        LblWDelayRate.Text = MyDataEnvironment.WeightedDelayRate.ToString
        LblWorkingDay.Text = MyDataEnvironment.WorkingDays.ToString.ToString
    End Sub

End Class
```

Discrete Event Simulation Federate

```
Imports Cosye.Hla.Rti
Imports Simphony.Simulation

Public Class FrmFabDES
    Dim Shop As New ShopFloor
    Dim Stations As Dictionary(Of String, Station) = Shop.StationList
    Dim Midbuffers As Dictionary(Of String, MidBuffer) = Shop.MidBufferList
    Dim Movers As Dictionary(Of String, Mover) = Shop.MoverList
    Dim FabShopControl(7) As List(Of Control)
    Dim FabShopListBox As List(Of ListBox)
    Dim NoPieceCompleted As Integer = 0
    Dim ReScheduleNum As Integer = 0
    Dim TotalHybridInteractionsFromSD As Integer = 0
    Dim TotalHybridInteractionsFromDES As Integer = 0
    Dim MyCalendar As Steel_Calendar
    Dim MLC As Double = 99
    Dim NextUtilizationSet As Long = 0

    '' '''''''''''''''' '' Initialize the stations
    Private Sub SetInterface()
        Dim MyDataShop As New DataShop
        '' ''''''''''''''Initialize Main form control
        CmbCut.Text = MyDataShop.FillMainFormControl("CmbCut")
        CmbInspect.Text = MyDataShop.FillMainFormControl("CmbInspect")
        CmbPaint.Text = MyDataShop.FillMainFormControl("CmbPaint")
        CmbShop.Text = MyDataShop.FillMainFormControl("CmbShop")
        TxtMidBufferNum.Text = MyDataShop.FillMainFormControl("TxtMidBufferNum")
        TxtMoverNum.Text = MyDataShop.FillMainFormControl("TxtMoverNum")

        '' ''''''''''''''Initialize the Buttons
        'Form Cutting Collection
        FabShopControl(1) = New List(Of Control)
        FabShopControl(1).Add(Me.BtnCut1)
        FabShopControl(1).Add(Me.BtnCut2)
        FabShopControl(1).Add(Me.BtnCut3)
        FabShopControl(1).Add(Me.BtnCut4)
        FabShopControl(1).Add(Me.BtnCut5)
        FabShopControl(1).Add(Me.BtnCut6)
        FabShopControl(1).Add(Me.BtnCut7)
        FabShopControl(1).Add(Me.BtnCut8)
        SetVisibilityControl(Convert.ToInt32(CmbCut.Text), FabShopControl(1))
        'Form Fitting Collection
        FabShopControl(2) = New List(Of Control)
        FabShopControl(2).Add(Me.BtnFit1)
        FabShopControl(2).Add(Me.BtnFit2)
        FabShopControl(2).Add(Me.BtnFit3)
        FabShopControl(2).Add(Me.BtnFit4)
        FabShopControl(2).Add(Me.BtnFit5)
        FabShopControl(2).Add(Me.BtnFit6)
        FabShopControl(2).Add(Me.BtnFit7)
        FabShopControl(2).Add(Me.BtnFit8)
```

```
SetVisibilityControl(Convert.ToInt32(CmbShop.Text), FabShopControl(2))
'Form Welding Collection
FabShopControl(3) = New List(Of Control)
FabShopControl(3).Add(Me.BtnWeld1)
FabShopControl(3).Add(Me.BtnWeld2)
FabShopControl(3).Add(Me.BtnWeld3)
FabShopControl(3).Add(Me.BtnWeld4)
FabShopControl(3).Add(Me.BtnWeld5)
FabShopControl(3).Add(Me.BtnWeld6)
FabShopControl(3).Add(Me.BtnWeld7)
FabShopControl(3).Add(Me.BtnWeld8)
SetVisibilityControl(Convert.ToInt32(CmbShop.Text), FabShopControl(3))
'Form Inspection Collection
FabShopControl(4) = New List(Of Control)
FabShopControl(4).Add(Me.BtnInspect1)
FabShopControl(4).Add(Me.BtnInspect2)
FabShopControl(4).Add(Me.BtnInspect3)
FabShopControl(4).Add(Me.BtnInspect4)
FabShopControl(4).Add(Me.BtnInspect5)
FabShopControl(4).Add(Me.BtnInspect6)
FabShopControl(4).Add(Me.BtnInspect7)
FabShopControl(4).Add(Me.BtnInspect8)
SetVisibilityControl(Convert.ToInt32(CmbInspect.Text), FabShopControl(4))
'Form Painting Collection
FabShopControl(5) = New List(Of Control)
FabShopControl(5).Add(Me.BtnPaint1)
FabShopControl(5).Add(Me.BtnPaint2)
FabShopControl(5).Add(Me.BtnPaint3)
FabShopControl(5).Add(Me.BtnPaint4)
FabShopControl(5).Add(Me.BtnPaint5)
FabShopControl(5).Add(Me.BtnPaint6)
FabShopControl(5).Add(Me.BtnPaint7)
FabShopControl(5).Add(Me.BtnPaint8)
SetVisibilityControl(Convert.ToInt32(CmbPaint.Text), FabShopControl(5))
''Form Mover Collection
'FabShopControl(6) = New List(Of Control)
'FabShopControl(6).Add(Me.BtnMover50)
'FabShopControl(6).Add(Me.BtnMover51)
'FabShopControl(6).Add(Me.BtnMover52)
'FabShopControl(6).Add(Me.BtnMover53)
'FabShopControl(6).Add(Me.BtnMover54)
'FabShopControl(6).Add(Me.BtnMover55)
'FabShopControl(6).Add(Me.BtnMover56)
'FabShopControl(6).Add(Me.BtnMover57)
Dim MoverNum As Integer = Convert.ToInt32((Math.Min(Val(TxtMoverNum.Text), 8)))
'SetVisibilityControl(MoverNum, FabShopControl(6))
Dim MidBufNum As Integer = Convert.ToInt32(Math.Min(Val(TxtMidBufferNum.Text) + 1, 12))
'Form MidBuffer Lable Collection
FabShopControl(7) = New List(Of Control)
FabShopControl(7).Add(Me.LblMidBuf70)
FabShopControl(7).Add(Me.LblMidBuf71)
FabShopControl(7).Add(Me.LblMidBuf72)
FabShopControl(7).Add(Me.LblMidBuf73)
FabShopControl(7).Add(Me.LblMidBuf74)
FabShopControl(7).Add(Me.LblMidBuf75)
FabShopControl(7).Add(Me.LblMidBuf76)
FabShopControl(7).Add(Me.LblMidBuf77)
FabShopControl(7).Add(Me.LblMidBuf78)
FabShopControl(7).Add(Me.LblMidBuf79)
FabShopControl(7).Add(Me.LblMidBuf80)
FabShopControl(7).Add(Me.LblMidBuf81)
SetVisibilityControl(MidBufNum, FabShopControl(7))
'Form MidBuffer Collection
FabShopListBox = New List(Of ListBox)
FabShopListBox.Add(Me.LstMidbuf70)
FabShopListBox.Add(Me.LstMidbuf71)
FabShopListBox.Add(Me.LstMidbuf72)
FabShopListBox.Add(Me.LstMidbuf73)
FabShopListBox.Add(Me.LstMidbuf74)
```

```
    FabShopListBox.Add(Me.LstMidbuf75)
    FabShopListBox.Add(Me.LstMidbuf76)
    FabShopListBox.Add(Me.LstMidbuf77)
    FabShopListBox.Add(Me.LstMidbuf78)
    FabShopListBox.Add(Me.LstMidbuf79)
    FabShopListBox.Add(Me.LstMidbuf80)
    FabShopListBox.Add(Me.LstMidbuf81)
    SetVisibilityListBox(MidBufNum, FabShopListBox)
    ' """""""""""Initialize values of the Stations
    Dim StationItem As Station
    For i As Integer = 0 To 5
        For j As Integer = 1 To 8
            StationItem = New Station
            StationItem.ID = i * 10 + j
            Stations.Add(StationItem.ID.ToString, StationItem)
        Next
    Next
    ' """""""""""Initialize values of the MidBuffers
    Dim MidBufferItem As MidBuffer
    For i As Integer = 70 To 89
        MidBufferItem = New MidBuffer
        MidBufferItem.ID = i
        Midbuffers.Add(MidBufferItem.ID.ToString, MidBufferItem)
    Next
    ' """""""""""Initialize values of the Movers
    Dim MoverItem As Mover
    For i As Integer = 50 To 69
        MoverItem = New Mover
        MoverItem.ID = i
        Movers.Add(MoverItem.ID.ToString, MoverItem)
    Next
End Sub

'Show selected number of FabShop Control
Private Sub SetVisibilityControl(ByVal VisNo As Integer, ByRef ShopControl As List(Of Control))
    Dim ShopCntrl As New Control
    For i As Integer = 0 To VisNo - 1
        ShopCntrl = ShopControl.Item(i)
        ShopCntrl.Visible = True
    Next
    For i As Integer = VisNo To ShopControl.Count - 1
        ShopCntrl = ShopControl.Item(i)
        ShopCntrl.Visible = False
    Next
End Sub

'Show selected number of FabShop ListBox
Private Sub SetVisibilityListBox(ByVal VisNo As Integer, ByRef ShopControl As List(Of ListBox))
    Dim ShopCntrl As New Control
    For i As Integer = 0 To VisNo - 1
        ShopCntrl = ShopControl.Item(i)
        ShopCntrl.Visible = True
    Next
    For i As Integer = VisNo To ShopControl.Count - 1
        ShopCntrl = ShopControl.Item(i)
        ShopCntrl.Visible = False
    Next
End Sub

'Set text on the  FabShop Controls
Private Sub SetControlsText()
    Dim ShopCntrl As Control
    Dim PieceID As String = ""
    'Set the Station related controls
    Dim StationItem As Station
    Dim StationID As Integer
    For i As Integer = 0 To 4
        For j As Integer = 0 To 7
            'Check if station ID exists
```

```
        StationID = i * 10 + j + 1
        If Stations.Keys.Contains(StationID.ToString) Then
            'Read the control
            ShopCntrl = New Control
            ShopCntrl = FabShopControl(i + 1)(j)
            'Read the station
            StationItem = New Station
            StationItem = Stations(StationID.ToString)
            PieceID = Shop.PieceIDofHandle(StationItem.CurrentPieceHandle)
            'Check if any Piece is on the station
            If PieceID.Length > 0 Then
                ShopCntrl.Text = Microsoft.VisualBasic.Right(PieceID, 7)
                ShopCntrl.BackColor = Color.LightPink
            ElseIf StationItem.State = ToolState.Suspend Then 'Suspended
                ShopCntrl.Text = ""
                ShopCntrl.BackColor = Color.LightBlue
            Else 'Idle
                ShopCntrl.Text = ""
                ShopCntrl.BackColor = Color.LightGreen
            End If
        End If
    Next
Next
'Set the Controls related to the Mover
Dim MoverItem As Mover
Dim MoverID As Integer
Dim BusyMover As Integer = 0
Dim TotalMover As Integer = CType(TxtMoverNum.Text, Integer)
For j As Integer = 0 To TotalMover - 1
    'Check if station ID exists
    MoverID = 50 + j
    If Movers.Keys.Contains(MoverID.ToString) Then
        'Read the control
        'Read the station
        MoverItem = New Mover
        MoverItem = Movers(MoverID.ToString)
        PieceID = Shop.PieceIDofHandle(MoverItem.PieceOnMover.PieceHandle)
        'Check if any Piece is on the Mover
        If PieceID.Length > 0 Then
            BusyMover = BusyMover + 1
        Else 'No Piece
        End If
    End If
Next
LblCraneBusy.Text = BusyMover.ToString
LblCraneIdle.Text = (TotalMover - BusyMover).ToString
'Set the Controls related to the Midbuffer
Dim MidBufferItem As MidBuffer
Dim MidBufferID As Integer
Dim MidBufferList As ListBox
Dim Count As Integer
For j As Integer = 0 To 11
    'Check if station ID exists
    MidBufferID = 70 + j
    If Midbuffers.Keys.Contains(MidBufferID.ToString) Then
        'Read the control
        MidBufferList = New ListBox
        MidBufferList = FabShopListBox(j)
        'Read the midbuffer
        MidBufferItem = New MidBuffer
        MidBufferItem = Midbuffers(MidBufferID.ToString)
        'Read count from mid buffer
        For Each Count In MidBufferList.Items
        Next
        If MidBufferItem.PieceHandleList.Count > 0 Then
            If Not Count = MidBufferItem.PieceHandleList.Count Then
                MidBufferList.BackColor = Color.LightPink
                MidBufferList.Items.Clear()
                MidBufferList.Items.Add(MidBufferItem.PieceHandleList.Count)
```

```
            End If
        Else
            MidBufferList.Items.Clear()
            MidBufferList.BackColor = Color.LightGreen
        End If
    End If
Next
'Set the Hour Label
LblHour.Text = (Convert.ToInt64(Shop.CurTime / 36) / 100).ToString
'Set Piece Completed Lable
LblPieceCompleted.Text = NoPieceCompleted.ToString
'Referesh
Me.Refresh()
End Sub

'Set the controls Unable
Private Sub SetTheControlsUnable()
    'Text boxes
    TxtMidBufferNum.Enabled = False
    TxtMoverNum.Enabled = False
    'Combo boxes
    CmbCut.Enabled = False
    CmbInspect.Enabled = False
    CmbPaint.Enabled = False
    CmbShop.Enabled = False
End Sub

'Set the final entered Values for the shop Object
Private Sub SetFinalValuesForTheShopObject()
    'Set Stations
    Dim StationItemID As Integer
    Dim FabShopControlItm As Control
    Dim i As Integer
    For i = 0 To 4
        For j As Integer = 0 To 7
            FabShopControlItm = New Control
            'Retrive the related button to the station
            FabShopControlItm = FabShopControl(i + 1).Item(j)
            If FabShopControlItm.Visible = False Then
                StationItemID = i * 10 + j + 1
                Stations.Remove(StationItemID.ToString)
            End If
        Next
    Next
    'Set Movers
    For i = (50 + Convert.ToInt32((TxtMoverNum.Text))) To 69
        Movers.Remove(i.ToString)
    Next
    'Set MidBuffers
    For i = (71 + Convert.ToInt32((TxtMidBufferNum.Text))) To 89
        Midbuffers.Remove(i.ToString)
    Next
End Sub
Private Sub DES_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Set interface
    SetInterface()
    Dim MyDataShop As New DataShop
    'Fill Stations
    'Dim StationForm As New StationInfo
    MyDataShop.FillStationList(Stations)
    'Fill MidBuffers
    'Dim MidBufForm As New MidBufferInfo
    MyDataShop.FillMidBuferList(Midbuffers)
    'Fill Mover
    'Dim MoverForm As New MoverInfo
    MyDataShop.FillmovererList(Movers)
End Sub
Private Sub CmbCut_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CmbCut.TextChanged
```

```
        'Check if all controls have been loaded
        If (Not FabShopControl(1) Is Nothing) Then
            If FabShopControl(1).Count = 8 Then
                SetVisibilityControl(Convert.ToInt32(CmbCut.Text), FabShopControl(1))
            End If
        End If
    End Sub

    Private Sub CmbShop_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CmbShop.TextChanged
        'Check if all controls have been loaded
        If (Not FabShopControl(2) Is Nothing) And (Not FabShopControl(3) Is Nothing) Then
            If FabShopControl(2).Count = 8 And FabShopControl(3).Count = 8 Then
                SetVisibilityControl(Convert.ToInt32(CmbShop.Text), FabShopControl(2))
                SetVisibilityControl(Convert.ToInt32(CmbShop.Text), FabShopControl(3))
            End If
        End If
    End Sub

    Private Sub CmbInspect_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles CmbInspect.SelectedIndexChanged
        'Check if all controls have been loaded
        If (Not FabShopControl(4) Is Nothing) Then
            If FabShopControl(4).Count = 8 Then
                SetVisibilityControl(Convert.ToInt32(CmbInspect.Text), FabShopControl(4))
            End If
        End If
    End Sub

    Private Sub CmbPaint_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles CmbPaint.SelectedIndexChanged
        'Check if all controls have been loaded
        If (Not FabShopControl(5) Is Nothing) Then
            If FabShopControl(5).Count = 8 Then
                SetVisibilityControl(Convert.ToInt32(CmbPaint.Text), FabShopControl(5))
            End If
        End If
    End Sub

    Private Sub TxtMoverNum_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TxtMoverNum.TextChanged
    End Sub

    Private Sub TxtMidBufferNum_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TxtMidBufferNum.TextChanged
        'Check if all controls have been loaded
        If (Not FabShopListBox Is Nothing) Then
            If FabShopListBox.Count = 12 Then
                Dim MidBufNum As Integer = Math.Min(Convert.ToInt32(TxtMidBufferNum.Text) + 1, 5)
                SetVisibilityListBox(MidBufNum, FabShopListBox)
                SetVisibilityControl(MidBufNum, FabShopControl(7))
            End If
        End If
    End Sub

    'Load the station form
    Private Sub BtnStationInfo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnStationInfo.Click
        Dim StationForm As New StationInfo
        StationForm.LinkedStations = Stations
        StationForm.FillListsCombos(Convert.ToInt32(CmbCut.Text), Convert.ToInt32(CmbShop.Text),
Convert.ToInt32(CmbInspect.Text), Convert.ToInt32(CmbPaint.Text))
        If Shop.CurTime > 0 Then
            StationForm.ShouldSaveTheChanges = False
        End If
        StationForm.Initialize()
        StationForm.ShowDialog()
    End Sub
```

```
    'Load the Midbuffer form
    Private Sub BtnMidBuf_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnMidBuf.Click
        Dim MidBufForm As New MidBufferInfo
        MidBufForm.LinkedMidbuffers = Midbuffers
        Dim MidBufNum As Integer = Math.Min(Convert.ToInt32(TxtMidBufferNum.Text) + 1, 12)
        SetVisibilityListBox(MidBufNum, FabShopListBox)
        SetVisibilityControl(MidBufNum, FabShopControl(7))
        MidBufForm.FillListsCombos(MidBufNum - 1, Convert.ToInt32(CmbCut.Text), Convert.ToInt32(CmbShop.Text),
Convert.ToInt32(CmbInspect.Text), Convert.ToInt32(CmbPaint.Text))
        If Shop.CurTime > 0 Then
            MidBufForm.ShouldSaveTheChanges = False
        End If
        MidBufForm.Initialize()
        MidBufForm.ShowDialog()
    End Sub

    'Load the Mover form
    Private Sub BtnMover_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnMover.Click
        Dim MoverForm As New MoverInfo
        MoverForm.LinkedMovers = Movers
        MoverForm.LinkedMidbuffers = Midbuffers
        MoverForm.LinkedStations = Stations
        Dim MoverNum As Integer = Math.Min(Convert.ToInt32(TxtMoverNum.Text), 20)
        MoverForm.FillListsCombos(MoverNum)
        If Shop.CurTime > 0 Then
            MoverForm.ShouldSaveTheChanges = False
        End If
        MoverForm.Initialize()
        MoverForm.ShowDialog()
    End Sub

    Private Sub MyCalendarFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyCalendarFactory.ReflectAttributeValues
        Dim ShiftTypeHanle As Cosye.Hla.Rti.AttributeHandle =
MyCalendarFactory.GetAttributeHandle("CurrentShiftType")
        'Check if the Current ShiftType has been changed
        If e.theValues.Contains(ShiftTypeHanle) Then
            MyCalendar = MyCalendarFactory(e.theObject)
            Shop.CurDate = MyCalendar.CurrentDate
            If Shop.CurrentShiftType = ShiftType.Close And e.theTime > 0 Then 'Set the total close time
                Shop.TotalCloseTime = Shop.TotalCloseTime + e.theTime - Shop.LastCloseTimeStarted
                Shop.LastCloseTimeStarted = 0
            End If
            If MyCalendar.CurrentShiftType = ShiftType.Close Then 'Set the start of close time
                If e.theTime < Double.MaxValue Then
                    Shop.LastCloseTimeStarted = e.theTime
                    Shop.CloseTimeWillBeFinished = e.theTime + MyCalendar.CurrentShiftHours * 3600
                Else
                    Shop.LastCloseTimeStarted = 0
                    Shop.CloseTimeWillBeFinished = MyCalendar.CurrentShiftHours * 3600
                End If
            End If
            Shop.CurrentShiftType = MyCalendar.CurrentShiftType
            LblCurDate.Text = MyCalendar.CurrentDate.ToString
            Try
                LblWorkDay.Text = MyCalendar.DayNo.ToString
            Catch ex As Exception
                'DayNo Has not been published yet
            End Try
        End If
    End Sub
    'Register Station Productivities
    Private Sub RegisterStationsProductivities()
        Dim MyStationProductivity As Steel_StationProductivity
        For Each stationItem As Station In Stations.Values
            MyStationProductivity = MyStationProductivityFactory.RegisterObjectInstance
            MyStationProductivity.ID = stationItem.ID
            MyStationProductivity.MaxOperator = stationItem.MaxReqOperators
```

```
            MyStationProductivity.MinOperator = stationItem.MinReqOperators
            MyStationProductivity.CurOperator = stationItem.AssignedOperatorNo
            MyStationProductivity.SFunction = stationItem.SFunction
            MyStationProductivity.UpdateAttributeValues()
            TotalHybridInteractionsFromDES = TotalHybridInteractionsFromDES + 1
            TxtHybridFromDES.Text = TotalHybridInteractionsFromDES.ToString
            MyStationProductivity.UnconditionalAttributeOwnershipDivestiture("CurOperator")
        Next
    End Sub

    Dim PreviouShiftOfUpdateStationState As ShiftType = ShiftType.Close
    'Update Station State
    Public Sub UpdateStationsStates(ByVal MyShift As ShiftType)
        Dim StationItem As Station
        If MyShift = ShiftType.Close Then
            If PreviouShiftOfUpdateStationState <> ShiftType.Close Then 'Report all stations as idle
                PreviouShiftOfUpdateStationState = ShiftType.Close
                For Each MyStationProductivity As Steel_StationProductivity In MyStationProductivityFactory
                    StationItem = New Station
                    StationItem = Stations(MyStationProductivity.ID.ToString)
                    If StationItem.State <> ToolState.Idle Then
                        MyStationProductivity.State = ToolState.Idle
                        MyStationProductivity.UpdateAttributeValues()
                        StationItem.ReportedState = ToolState.Idle
                        TotalHybridInteractionsFromDES = TotalHybridInteractionsFromDES + 1
                        TxtHybridFromDES.Text = TotalHybridInteractionsFromDES.ToString
                    End If
                Next
            End If
        ElseIf PreviouShiftOfUpdateStationState <> ShiftType.Close Then 'Report busy station by end of close shift
            PreviouShiftOfUpdateStationState = ShiftType.DayShift
            For Each MyStationProductivity As Steel_StationProductivity In MyStationProductivityFactory
                StationItem = New Station
                StationItem = Stations(MyStationProductivity.ID.ToString)
                If StationItem.State = ToolState.Busy Then
                    MyStationProductivity.State = ToolState.Busy
                    MyStationProductivity.UpdateAttributeValues()
                    StationItem.ReportedState = ToolState.Busy
                    TotalHybridInteractionsFromDES = TotalHybridInteractionsFromDES + 1
                    TxtHybridFromDES.Text = TotalHybridInteractionsFromDES.ToString
                End If
            Next
        Else
            For Each MyStationProductivity As Steel_StationProductivity In MyStationProductivityFactory
                StationItem = New Station
                StationItem = Stations(MyStationProductivity.ID.ToString)
                If StationItem.State <> StationItem.ReportedState Then
                    MyStationProductivity.State = StationItem.State
                    MyStationProductivity.UpdateAttributeValues()
                    StationItem.ReportedState = StationItem.State
                    TotalHybridInteractionsFromDES = TotalHybridInteractionsFromDES + 1
                    TxtHybridFromDES.Text = TotalHybridInteractionsFromDES.ToString
                End If
            Next
        End If
    End Sub

    Dim ItIsFirstReceivedPiece As Boolean = True
    Private Sub MyPieceEntityFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyPieceEntityFactory.ReflectAttributeValues
        Dim NewPieceEntity As New PieceEntityHandle
        NewPieceEntity.Piece = MyPieceEntityFactory(e.theObject)
        NewPieceEntity.PieceHandle = e.theObject
        NewPieceEntity.Piece.AttributeOwnershipAcquisition("CuttingStart", "CuttingManHour", "CuttingFinish",
"FittingStart", "FittingManHour", "FittingFinish", _
         "WeldingStart", "WeldingManHour", "WeldingFinish", "FitInspectionStart", "FitInspectionManHour",
"FitInspectionFinish", "WeldInspectionStart", "WeldInspectionManHour", "WeldInspectionFinish", "PaintingStart",
"PaintingManHour", "PaintingFinish")
        ''should be done only on firt time received updated
```

```
        If ItIsFirstReceivedPiece Then
            ItIsFirstReceivedPiece = False
            'Set the controls unEnable
            SetTheControlsUnable()
            'Set the final entered Values for the shop Object
            SetFinalValuesForTheShopObject()
            'Register and update the StationProductivity objects
            RegisterStationsProductivities()
        End If
        If NewPieceEntity.Piece.CuttingFinish = Nothing Then 'Piece is not in shop
            'Handle New arrived piece
            Shop.HandleNewPieceArrived(NewPieceEntity)
        Else 'Piece is in shop (Initial Condition of the piece)
            Dim LocationID As Integer
            Dim LocationType As AssociatedTool
            Dim NumOperationDone As Integer = 0
            Shop.HandlePieceInShopArrived(NewPieceEntity, LocationID, LocationType, NumOperationDone)
            If NumOperationDone = 1 Then
                LstCutFinish.Items.Add(NewPieceEntity.Piece.PieceID)
            ElseIf NumOperationDone = 2 Or NumOperationDone = 3 Then
                LstFitFinish.Items.Add(NewPieceEntity.Piece.PieceID)
            ElseIf NumOperationDone = 4 Or NumOperationDone = 5 Then
                LstWeldFinish.Items.Add(NewPieceEntity.Piece.PieceID)
            End If
        End If

        'Update the controls
        '' '''''''''''''''''''''''''''''''''''''''''''''''''
    End Sub

    Dim NoEventScheduledTimeStep As Integer = 1
    Private Sub fedAmb_TimeAdvanceGrant(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.TimeAdvanceGrantEventArgs) Handles fedAmb.TimeAdvanceGrant
        'Check if shop is close or open
        If Shop.CurrentShiftType = ShiftType.Close Then
            rtiAmb.NextMessageRequest(Shop.CloseTimeWillBeFinished)
        Else
            'Set the current time of the shop
            Shop.CurTime = e.theTime - Shop.TotalCloseTime
            'Process any internal events that should occur at the current time.
            Shop.MyEngine.Simulate(Shop.CurTime)
            If Shop.MyEngine.ScheduledEventCount = 0 Then 'Create a fake event
                'Schedule an Empty Event
                Dim EmptyEventParameters As New EventParameters
                Shop.MyEngine.ScheduleEvent(EmptyEventParameters, Shop.EmptyEvent, NoEventScheduledTimeStep)
                'Set the time advancement step
                NoEventScheduledTimeStep = 1
                Dim InitialMidBuffer As MidBuffer = Shop.MidBufferList("70")
                If InitialMidBuffer.PieceHandleList.Count = 0 Then
                    NoEventScheduledTimeStep = 1000
                End If
            End If
            ' Update the controls on the form
            SetControlsText()
            'Update all completed pieces
            If Not Shop.CompletedPieceList Is Nothing Then
                Dim CompletedPiece As PieceEntityHandle
                Dim CompletedPieceEntity As Steel_PieceEntity
                While Shop.CompletedPieceList.Count > 0
                    NoPieceCompleted = NoPieceCompleted + 1
                    CompletedPiece = New PieceEntityHandle
                    CompletedPiece = Shop.CompletedPieceList(0)
                    Shop.CompletedPieceList.RemoveAt(0)
                    'Updated the attributes
                    CompletedPieceEntity = MyPieceEntityFactory(CompletedPiece.PieceHandle)
                    CompletedPieceEntity.CuttingStart = CompletedPiece.Piece.CuttingStart
                    CompletedPieceEntity.CuttingManHour = CompletedPiece.Piece.CuttingManHour
                    CompletedPieceEntity.CuttingFinish = CompletedPiece.Piece.CuttingFinish
                    CompletedPieceEntity.FittingStart = CompletedPiece.Piece.FittingStart
```

```
            CompletedPieceEntity.FittingManHour = CompletedPiece.Piece.FittingManHour
            CompletedPieceEntity.FittingFinish = CompletedPiece.Piece.FittingFinish
            CompletedPieceEntity.WeldingStart = CompletedPiece.Piece.WeldingStart
            CompletedPieceEntity.WeldingManHour = CompletedPiece.Piece.WeldingManHour
            CompletedPieceEntity.WeldingFinish = CompletedPiece.Piece.WeldingFinish
            CompletedPieceEntity.FitInspectionStart = CompletedPiece.Piece.FitInspectionStart
            CompletedPieceEntity.FitInspectionManHour = CompletedPiece.Piece.FitInspectionManHour
            CompletedPieceEntity.FitInspectionFinish = CompletedPiece.Piece.FitInspectionFinish
            CompletedPieceEntity.WeldInspectionStart = CompletedPiece.Piece.WeldInspectionStart
            CompletedPieceEntity.WeldInspectionManHour = CompletedPiece.Piece.WeldInspectionManHour
            CompletedPieceEntity.WeldInspectionFinish = CompletedPiece.Piece.WeldInspectionFinish
            Try
                CompletedPieceEntity.PaintingStart = CompletedPiece.Piece.PaintingStart
                CompletedPieceEntity.PaintingManHour = CompletedPiece.Piece.PaintingManHour
                CompletedPieceEntity.PaintingFinish = CompletedPiece.Piece.PaintingFinish
            Catch ex As Exception
                'No Painting is required
            End Try
            CompletedPieceEntity.UpdateAttributeValues()
        End While
    End If
    'Request Next Message consider the Engine Accuracy
    Dim NextMessageTime As Double
    If (Shop.MyEngine.TimeNext - Shop.CurTime) >= fedAmb.Lookahead Then
        NextMessageTime = Shop.MyEngine.TimeNext + Shop.TotalCloseTime
    Else
        NextMessageTime = (Shop.CurTime + fedAmb.Lookahead + Shop.TotalCloseTime)
    End If
    rtiAmb.NextMessageRequest(NextMessageTime)
End If
'Update the station state if it has changed
UpdateStationsStates(Shop.CurrentShiftType)
End Sub

Private Sub MyStationProductivityFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyStationProductivityFactory.ReflectAttributeValues
    Dim MyStationProd As Steel_StationProductivity = MyStationProductivityFactory(e.theObject)
    Dim MyProductivityHandle As AttributeHandle = MyStationProductivityFactory.GetAttributeHandle("Productivity")
    Dim MyCurOperatorHandle As AttributeHandle = MyStationProductivityFactory.GetAttributeHandle("CurOperator")
    Dim MyStation As Station = Stations(MyStationProd.ID.ToString)
    Dim EventKey As String = MyStation.ID.ToString & "_" & MyStation.CurrentPieceHandle.ToString
    'Count the interactions
    TotalHybridInteractionsFromSD = TotalHybridInteractionsFromSD + 1
    TxtHybridFromSD.Text = TotalHybridInteractionsFromSD.ToString
    'Remove Extra Engine Supplementaries
    Shop.RemoveExtraEngineSupplementaries()
    'Check if Productivity has been updated
    If e.theValues.Contains(MyProductivityHandle) Then
        'Check if rescheduling is required
        If Shop.EngineEntities.Keys.Contains(EventKey) Then
            Dim ScheduledTime As Double = Shop.EngineTimes(EventKey)
            If ScheduledTime > Shop.CurTime Then
                'Count num of reschedules
                ReScheduleNum = ReScheduleNum + 1
                TxtReschedules.Text = ReScheduleNum.ToString
                'Calculate New Time
                Dim NewTime As Double = (ScheduledTime - Shop.CurTime) * _
                        MyStation.Productivity / MyStationProd.Productivity
                'Retrieve the Event information
                Dim ChangedEventParameters As New EventParameters
                ChangedEventParameters = Shop.EngineEntities(EventKey)
                'Cancel Currently Scheduled Information
                Shop.MyEngine.CancelEvent(ChangedEventParameters)
                Shop.EngineEntities.Remove(EventKey)
                Shop.EngineTimes.Remove(EventKey)
                'Schedule New Event
                Shop.MyEngine.ScheduleEvent(ChangedEventParameters, Shop.StationSrviceFinished, NewTime)
                Shop.EngineTimes.Add(EventKey, NewTime + Shop.CurTime)
                Shop.EngineEntities.Add(EventKey, ChangedEventParameters)
```

```
            End If
        End If
        'Set the new Productivity
        MyStation.Productivity = MyStationProd.Productivity
    End If
    'Check if Current Operator has been updated
    If e.theValues.Contains(MyCurOperatorHandle) Then
        If Shop.EngineEntities.Keys.Contains(EventKey) Then
            Dim ScheduledTime As Double = Shop.EngineTimes(EventKey)
            If ScheduledTime > Shop.CurTime Then
                'Retrieve the Event information
                Dim ChangedEventParameters As New EventParameters
                ChangedEventParameters = Shop.EngineEntities(EventKey)
                'Sample a duration for new and old operator #
                Dim NewOprNo As Integer = MyStationProd.CurOperator
                Dim OldOprNo As Integer = MyStation.AssignedOperatorNo
                Dim NewOprDistribution As Distribution = MyStation.DurationStructure(NewOprNo.ToString)
                Dim OldOprDistribution As Distribution = MyStation.DurationStructure(OldOprNo.ToString)
                Dim NewOprDuration As Double = NewOprDistribution.DurationFactor
                Dim OldOprDuration As Double = OldOprDistribution.DurationFactor
                'Count num of reschedules
                ReScheduleNum = ReScheduleNum + 1
                TxtReschedules.Text = ReScheduleNum.ToString
                'Calculate New Time
                Dim NewTime As Double = (ScheduledTime - Shop.CurTime) * NewOprDuration / OldOprDuration
                'Cancel Currently Scheduled Information
                Shop.MyEngine.CancelEvent(ChangedEventParameters)
                Shop.EngineEntities.Remove(EventKey)
                Shop.EngineTimes.Remove(EventKey)
                'Schedule New Event
                Shop.MyEngine.ScheduleEvent(ChangedEventParameters, Shop.StationSrviceFinished, NewTime)
                Shop.EngineTimes.Add(EventKey, NewTime + Shop.CurTime)
                Shop.EngineEntities.Add(EventKey, ChangedEventParameters)
            End If
        End If
        'Set the Current Operator #
        MyStation.AssignedOperatorNo = MyStationProd.CurOperator
    End If
End Sub

Private Sub fedAmb_EndExecution(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
fedAmb.EndExecution
    'Define the Data shop and its related methods
    Dim MyDataShop As New DataShop
    'Update the statios tables mainly for the utilization purposes
    MyDataShop.UpdateStationTables(Stations, "StationsReport")
    'Update Mainform Controls
    MyDataShop.UpdateMainFormTables("CmbCut", CmbCut.Text)
    MyDataShop.UpdateMainFormTables("CmbInspect", CmbInspect.Text)
    MyDataShop.UpdateMainFormTables("CmbPaint", CmbPaint.Text)
    MyDataShop.UpdateMainFormTables("CmbShop", CmbShop.Text)
    MyDataShop.UpdateMainFormTables("TxtMidBufferNum", TxtMidBufferNum.Text)
    MyDataShop.UpdateMainFormTables("TxtMoverNum", TxtMoverNum.Text)
    'Update MLC Table
    MyDataShop.ReportMLCResult(MLC, MyCalendar.StartDate, MyCalendar.CurrentDate, NoPieceCompleted,
SimulationStartTime, TotalHybridInteractionsFromSD, TotalHybridInteractionsFromDES, ReScheduleNum)
End Sub

Private Sub fedAmb_BeginExecution(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
fedAmb.BeginExecution
    SimulationStartTime = DateTime.Now
End Sub

Private Sub BtnSaveLayout_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnSaveLayout.Click
    ''Update the set value in data base
    Dim MyDataShop As New DataShop
    MyDataShop.UpdateMainFormTables("CmbShop", CmbShop.Text)
    MyDataShop.UpdateMainFormTables("CmbCut", CmbCut.Text)
```

```
        MyDataShop.UpdateMainFormTables("CmbInspect", CmbInspect.Text)
        MyDataShop.UpdateMainFormTables("CmbPaint", CmbPaint.Text)
        MyDataShop.UpdateMainFormTables("TxtMoverNum", TxtMoverNum.Text)
        MyDataShop.UpdateMainFormTables("TxtMidBufferNum", TxtMidBufferNum.Text)
    End Sub

    Private Sub MyShopProductivityFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyShopProductivityFactory.ReflectAttributeValues
        Dim MyShopProductivity As Steel_ShopProductivity = MyShopProductivityFactory(e.theObject)
        MLC = MyShopProductivity.MLC
    End Sub

    Dim SimulationStartTime As New DateTime

    Private Sub fedAmb_AnnounceSynchronizationPoint(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.AnnounceSynchronizationPointEventArgs) Handles fedAmb.AnnounceSynchronizationPoint
        'Initialize the simulation engine
        Shop.MyEngine.InitializeEngine()
        Shop.MyEngine.InitializeScenario()
    End Sub

End Class
```

Visualization Federate

```
Imports Cosye.Hla.Rti

Public Class VisualizationFederate
    Dim MyTekla As New Tekla
    Dim MyVisualData As New DataVisual
    Dim MyCalendar As Steel_Calendar
    Dim NewDivisionRequest As Boolean = False
    Dim RequestedDivision As New DivisionItem
    Dim PieceProgressList As New Dictionary(Of String, VisualPiece)
    Dim PauseTheFederation As Boolean = False
    Dim ShowTekla As Boolean = False
    Dim ActiveShowTeklaMode As Boolean = False

    Private Sub MyVPieceFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyVPieceFactory.ReflectAttributeValues
        Dim NewVPiece As Steel_VPiece
        Dim NewVisualPiece As New VisualPiece
        NewVPiece = MyVPieceFactory(e.theObject)
        NewVisualPiece.PieceKey = NewVPiece.PieceKey
        NewVisualPiece.DivisionID = NewVPiece.DivisionID
        NewVisualPiece.CPI = NewVPiece.CPI
        NewVisualPiece.SPI = NewVPiece.SPI
        NewVisualPiece.Progress = NewVPiece.Progress
        'Update the model color if current show tekla mode is true
        If ActiveShowTeklaMode Then
            MyTekla.UpdateColors(NewVisualPiece)
        End If
        'Update the data base and return true if related division is new
        If MyVisualData.UpdateCompletedPiecesTable(NewVisualPiece) Then
            Dim MyDivisionItem As New DivisionItem
            MyDivisionItem.DivisionID = NewVPiece.DivisionID
            MyVisualData.ReadDivisionFile(MyDivisionItem.DivisionID, MyDivisionItem.DivisionFile)
            LstDivision.Items.Add(MyDivisionItem)
        End If
        'Updating the piece progress list
        If MyTekla.CurrentDivision = NewVisualPiece.DivisionID Then
            If PieceProgressList.Keys.Contains(NewVisualPiece.PieceKey) Then
                Dim OldVisualPiece As VisualPiece = PieceProgressList(NewVisualPiece.PieceKey)
                LstPieceProgress.Items.Remove(OldVisualPiece.ToString)
                LstPieceProgress.Items.Add(NewVisualPiece.ToString)
                PieceProgressList.Remove(NewVisualPiece.PieceKey)
                PieceProgressList.Add(NewVisualPiece.PieceKey, NewVisualPiece)
```

```
            Else
                LstPieceProgress.Items.Add(NewVisualPiece.ToString)
                PieceProgressList.Add(NewVisualPiece.PieceKey, NewVisualPiece)
            End If
        End If
        'Go to the last item in the list
        If LstPieceProgress.Items.Count > 0 Then
            LstPieceProgress.SetSelected((LstPieceProgress.Items.Count - 1), True)
        End If

        Me.Refresh()
    End Sub

    Private Sub BtnChangeDivision_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnChangeDivision.Click
        RequestedDivision = CType(LstDivision.SelectedItem, DivisionItem)
        'Change the current division
        If Not RequestedDivision.DivisionID = MyTekla.CurrentDivision _
        Or (MyTekla.RelatedTeklaFileIsOpen And Not ShowTekla And ActiveShowTeklaMode) Then
            LblRequestedDivision.Text = RequestedDivision.ToString
            MyTekla.RelatedTeklaFileIsOpen = False
            NewDivisionRequest = True
        ElseIf (Not MyTekla.RelatedTeklaFileIsOpen And ShowTekla) Or _
        (Not ActiveShowTeklaMode = ShowTekla) Then
            LblRequestedDivision.Text = RequestedDivision.ToString
            NewDivisionRequest = True
        Else
            RequestedDivision = New DivisionItem
        End If
    End Sub
    Private Sub MyCalendarFactory_ReflectAttributeValues(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.ReflectAttributeValuesEventArgs) Handles MyCalendarFactory.ReflectAttributeValues
        MyCalendar = MyCalendarFactory(e.theObject)
        'Check if DayNo has been Updated to commit the changes
        If MyTekla.CurrentDivision.Length > 0 Then
            Dim DayNoHandle As AttributeHandle = MyCalendarFactory.GetAttributeHandle("DayNo")
            If e.theValues.Contains(DayNoHandle) Then
                MyTekla.CommitChanges()
            End If
        End If
        LblDate.Text = MyCalendar.CurrentDate.ToString
        Me.Refresh()
    End Sub

    Private Sub fedAmb_BeginExecution(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
fedAmb.BeginExecution
        MyVisualData.InitializeTables()
        If RdBShowTekla.Checked Then
            MyTekla.OpenApplicaion()
        End If
    End Sub

    Private Sub fedAmb_EndExecution(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
fedAmb.EndExecution
        MyTekla.CloseApplicaion()
    End Sub

    Private Sub fedAmb_TimeAdvanceGrant(ByVal sender As System.Object, ByVal e As
Cosye.Hla.Rti.TimeAdvanceGrantEventArgs) Handles fedAmb.TimeAdvanceGrant
        If NewDivisionRequest Then
            LblRelatedTeklaFileOpen.ForeColor = Color.Red
            LblRelatedTeklaFileOpen.Text = "No 3D Model Linked!"
            ActiveShowTeklaMode = False
            If MyTekla.SetNewDivision(RequestedDivision.DivisionID, PieceProgressList, ShowTekla) Then ' returns true if
the file successfully gets open
                LblRelatedTeklaFileOpen.ForeColor = Color.Green
                LblRelatedTeklaFileOpen.Text = "See the Linked 3D Model!"
                ActiveShowTeklaMode = True
            End If
```

```
            LblCurrentDivision.Text = RequestedDivision.ToString
            LblRequestedDivision.Text = ""
            ' Set the Piece Progress List Box
            LstPieceProgress.Items.Clear()
            For Each VisualPieceItem As VisualPiece In PieceProgressList.Values
                LstPieceProgress.Items.Add(VisualPieceItem.ToString)
            Next
            If LstPieceProgress.Items.Count > 0 Then
                LstPieceProgress.SetSelected((LstPieceProgress.Items.Count - 1), True)
            End If
            MyTekla.CurrentDivision = RequestedDivision.DivisionID
            NewDivisionRequest = False
        ElseIf MyTekla.RelatedTeklaFileIsOpen Then
            MyTekla.CommitChanges()
        End If
        'Check if user wants to pause the Simulation
        Dim Result As DialogResult
        While PauseTheFederation
            Result = MessageBox.Show("Do you want to continue?", "Continue", MessageBoxButtons.YesNo)
            If Result = Windows.Forms.DialogResult.Yes Then
                PauseTheFederation = False
                LblFedPause.Text = ""
            End If
        End While
        Dim NextDay As Double = (e.theTime + 60 * 60)
        rtiAmb.TimeAdvanceRequest(NextDay)
    End Sub

    Private Sub BtnPauseFederation_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
BtnPauseFederation.Click
        PauseTheFederation = True
        LblFedPause.Text = "Federation Pause Request"
    End Sub

    Private Sub RdBDontShowTekla_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
RdBDontShowTekla.Click
        RdBShowTekla.Checked = False
        ShowTekla = False
        MyTekla.RelatedTeklaFileIsOpen = False
    End Sub

    Private Sub RdBShowTekla_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles RdBShowTekla.CheckedChanged
        RdBDontShowTekla.Checked = False
        ShowTekla = True
    End Sub
End Class
```

Data tables

The federation contains a very large MS Access database. Actual data, as planned data and simulation results are stored in this database. Federates connect to this database for getting required data and/or updating the database. The database is a relational database containing several tables. Samples of these tables are presented below.

Piece Table

Table C-1. Sample data in the piece table.

div_id	piece_id	quantity	weight	fab_mhrs	fabdwg_no	assembly_pos
13028	386536	1	635	15.41	24	16R24
13028	386524	1	2170	15.41	12	16R12
13028	386588	1	109	15.41	80	16R80
13028	386577	5	876	15.41	68	16R68
13028	386574	2	361	15.41	65	16R65
9772	160533	4	3	20.00	8600	1A38600
9772	160553	1	0	20.00	1	1A31
9772	160568	1	0	20.00	16	1A316
9772	161920	9	6	20.00	8600	1A38600
11808	222038	7	1	60.00	8600	71A8600
11808	222038	7	1	96.25	8600	71A8600
12855	227213	4	48	38.81	90003	3B90003
12855	227213	4	48	60.00	90003	3B90003
12855	227214	1	154	38.81	9008	3B9008
12855	227214	1	154	60.00	9008	3B9008
12857	227245	8	96	38.99	90004	4A90004
12857	227245	8	96	60.00	90004	4A90004
12857	227246	1	154	38.99	9009	4A9009
12857	227246	1	154	60.00	9009	4A9009
11057	227323	1	74	60.00	9039	59B9039
11808	230173	1	65	60.00	5001	71A5001
11808	230173	1	65	96.25	5001	71A5001
11808	230179	1	55	96.25	5000	71A5000
11898	230697	1	33	39.03	1	13C1
11898	230697	1	33	60.00	1	13C1

where:

div_id: is the id of the division each piece belongs to.

piece_id: is the piece id.

quantity: is the number of identical pieces in a division, with the same piece id.

weight: is the weight of the steel piece

fab_mhrs: is the estimated man-hours required for fabricating the steel piece.

fabdwg_no: is the identification of the drawing contains the details and specifications required for fabricating the steel piece.

assembly_pos: is an id for each piece of steel. While div_id and piece_id are automatically generated by the system, assembly_pos contains the division number and piece number defined on the drawings that come to the shop with each steel piece. This id is basically used for interacting with the visualization federate.

Division Table

Table C-2. Samples of data available in the Division table.

div_id	fab_start_date	required_date	Weight_sum	requires_painting	NoOfPieces
		≈			
16527	07-Jan-09	13-Jan-09	66636	FALSE	8754
16521	07-Jan-09	09-Jan-09	27524	FALSE	3062
16652	07-Jan-09	11-Feb-09	19003	FALSE	6513
16519	08-Jan-09	14-Nov-08	37701	FALSE	5942
13769	08-Jan-09	08-Aug-08	52098	FALSE	13368
17710	08-Jan-09	28-Jan-09	5203	TRUE	182
16518	12-Jan-09	08-Jan-09	24270	FALSE	5019
14395	12-Jan-09	30-Jan-09	28645	FALSE	8312
17653	12-Jan-09	26-Jan-09	15878	FALSE	978
17017	12-Jan-09	26-Jan-09	15111	TRUE	3354
17016	12-Jan-09	26-Jan-09	42775	FALSE	5388
16687	12-Jan-09	26-Jan-09	68661	FALSE	17162
16340	12-Jan-09	09-Feb-09	4285	FALSE	552
14394	12-Jan-09	30-Jan-09	28645	FALSE	8312
13974	12-Jan-09	18-Jan-09	45166	FALSE	16500

where:

div_id: is the id of the division.

fab_start_date: is the scheduled fabrication start date for each division.

required_date: is the due date upon which all the pieces within the division is required to be fabricated.

weight_sum: is the weight of the division.

requires_painting: is an index that determines whether the division requires painting (True) or not (False).

NoOfPieces: refers to the number of pieces forming each division.

Report Table

A sample of report data-tables used for collecting simulation results is shown in Table C-3.

Table C-3. A sample table for reporting the simulation results.

fab _date	Run1 (kg)	Run2 (kg)	Run3 (kg)	Run4 (kg)	Run5 (kg)
13-Feb-08	452868.2722	407263.38	426350.28	493042.94	441232.83
14-Feb-08	358167.8717	378898.49	417673.78	372725.79	342310.5
19-Feb-08	94265.40958	93021.315	86806.462	99915.868	76495.709
20-Feb-08	307022.0553	296593.33	308622.6	268483.74	300559.24
21-Feb-08	174446.4453	201239.69	210191.52	170005.87	215339.93
26-Feb-08	228676.4834	221224.68	196512.43	238860.1	228138.53
27-Feb-08	171080.7074	149729.18	175290.51	148882.02	184190.99
28-Feb-08	356925.8706	366648.77	394247.71	383440.44	353450.63
03-Mar-08	263046.2486	278576.01	219269.5	254284.23	265294.31
05-Mar-08	199377.6952	191608.14	260830.22	191692.21	187579.31
06-Mar-08	160132.7422	165124.22	158975.1	168962.29	166318.1
10-Mar-08	128013.5606	94845.501	102179.26	102055.54	132259.42
11-Mar-08	241946.1894	271649.52	203300.85	260962.96	241002.22
12-Mar-08	131323.0756	115624.6	153837.44	99981.823	113696.31
13-Mar-08	414324.3278	423004.01	439699.12	396385.98	427582.16
18-Mar-08	112776.0074	125485.18	116826.01	121026.37	115042.74
19-Mar-08	294730.4045	308842.88	285434.41	301909.04	277327.23
20-Mar-08	310168.0754	298959.19	337017.09	305629.8	338629.74
24-Mar-08	218383.1833	192320.17	142308.98	183103.95	211942.14
25-Mar-08	131061.9298	129180.66	94867.012	97983.055	133286.64
27-Mar-08	486305.9301	476283.31	417134.71	450573.41	484685.98
# of pieces	38388	38268	38698	38545	38380
Complexity	2.034	2.034	2.018	2.030	2.034

where:

fab _date: is the date upon which steel pieces fabricated according to the simulation model.

# of pieces: is the total number of steel pieces fabricated within the specified time interval.

Complexity: is the average complexity of the piece fabricated.

Run X: is the calculated weight of steel pieces in kilogram in the X^{th} run.

References

Azimi, R., & Alvanchi, A. (2009). "Development of the industrial steel construction federation in COSYE Environment." Proceedings of the 16th Annual Canadian Construction Research Forum, Banff, Canada, 173-185.

Appendix D: Intelligent adjuster

This appendix presents the steps taken for developing the intelligent adjuster discussed in Chapter 1 and Chapter 3.

Automatic capture of complexity factors

Since piece complexity was one of the influencing factors affecting the activity durations and this information was not available, an application in VB.NET (Figure D-1) was developed to automatically capture complexity factors and weights of steel pieces from 3D models of steel projects. The process involves opening the 3D model of the desired project in Tekla and then running the application. Upon clicking the "Run" button, the application connects to the model, calculates the complexity factor and complexity intensity for each piece of steel, derives the weight of the piece and stores these in an MS Excel spreadsheet.

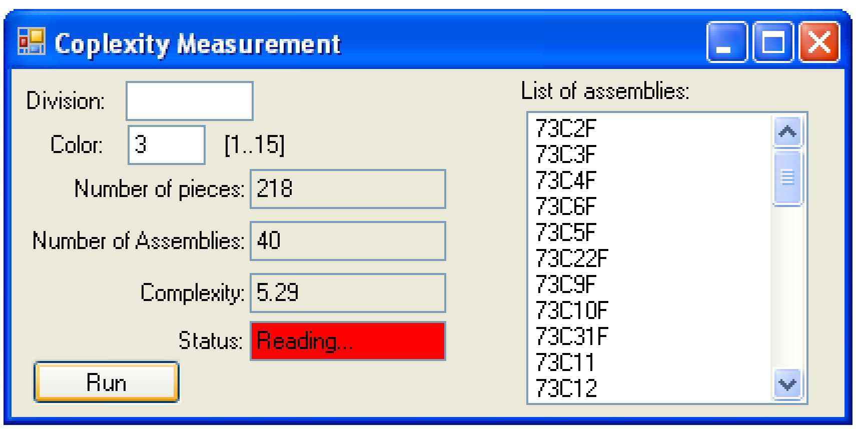


Figure D-1. Application for capturing piece complexity automatically.

Samples of the data captured automatically from a 3D model are shown in Table D-1. Details regarding how the complexity is calculated have already been provided in Chapter 3.

Table D-1. Sample of data captured automatically from 3D models by the application developed.

Assembly	Complexity	Complexity intensity	Weight(Kg)
50A1	12	4	582
50A10	15	4	534
50A100	1	1	6
50A101	3	1	674
50A102	4	2	680
50A103	3	1	104
50A104	4	2	31
50A105	1	1	27
50A106	1	1	14
50A107	1	1	27
50A108	3	1	29
50A109	1	1	14
50A11	3	1	234
50A110	3	1	29
50A111	5	2	1
50A112	1	1	55
50A113	2	1	132
50A114	8	3	54
50A115	1	1	6
50A116	1	1	0
50A12	3	1	196
50A13	17	5	634
50A14	17	5	631
50A15	19	5	293
50A16	9	3	280
50A17	3	1	98
50A18	7	2	137
50A19	6	2	139
50A2	17	5	555

Relationship between influencing factors and productivity rate

The complexity information for the steel pieces was captured in the previous phase. In this phase, historical data enhanced with piece complexity were used to determine the relationship between weight, shift rank, intensity of complexity (i.e., influencing factors) and the productivity rate and activity duration (outputs). A sample of available historical data in terms of fitting operation is represented in Table D-2.

Table D-2. Samples of historical data regarding fitting operation.

PieceID	Weight(kg)	Shift (Day1-Night 2)	Rank	Duration(Min.)	Complex. intensity	production rate (ton/mnhr)
73	168	1	3	22	3	0.46
74	168	1	3	28	3	0.36
75	168	2	2	30	3	0.34
76	174	1	1	65	2	0.16
77	213	1	2	30	2	0.43
78	216	1	3	28	2	0.46
79	216	2	2	20	2	0.65
80	218	1	3	25	2	0.52
81	221	2	3	60	3	0.22
82	221	2	3	60	3	0.22
83	260	1	2	70	3	0.22
84	277	1	1	25	3	0.66
85	288	1	3	65	4	0.27
86	305	1	2	30	2	0.61
87	314	1	1	75	3	0.25
88	329	1	1	45	4	0.44
89	334	1	1	25	3	0.80
90	346	1	1	27	2	0.77

Using *Neuroshell2*, an artificial neural network (ANN) based on the historical data was trained and used for predicting the productivity rate (and consequently activity duration) for the steel pieces being fitted. Relative weights of different influencing factors in forecasting fitting durations calculated by *Neuroshell2* are shown in Figure D-2.

Activity durations can also be forecasted at the division level. Let us assume that three divisions are planned to be fabricated within a certain period of time as shown in Table D-3 . Let us assume that one division (i.e., DivID=53A) is being fitted during day shifts, one division(i.e., DivID=50A) is being fitted during night shifts and one division(i.e., DivID=51A) is being fitted during both day and night shifts. For the 51A the shift assigned is the average between 1 and 2 i.e., 1.5.

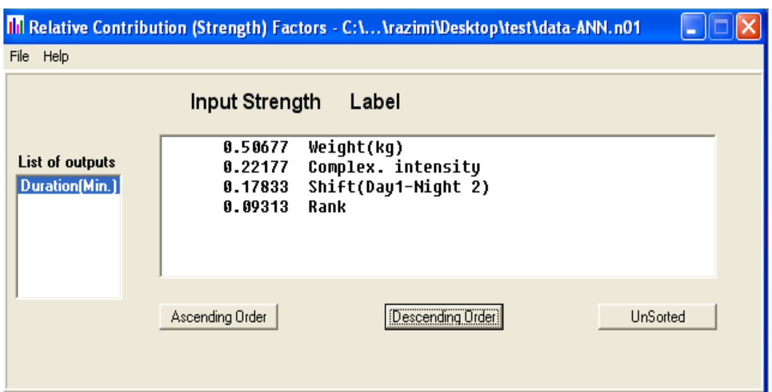


Figure D-2. Relative weights of different influencing factors in forecasting fitting durations.

Table D-3. Sample data used to clarify the forecasting process.

Div.ID.	Weight(kg)	Shift (Day1-Night 2)	Rank	Complex. intensity	productivity rate(ton/mnhr) calculated by ANN
50A	9000	2	1.4	2	0.75
51A	8500	1.5	1.9	3	0.39
53A	6250	1	2.2	3	0.77

Rank of the skilled workers is calculated as follows:

Rank = $\frac{\sum n_i \times R_i}{n_i}$

where:

R_i: is the rank of the workers which can be 1, 2 or 3

N_i: is the number of available workers with the rank R_i

for example if 3 workers with rank 1 and 2 workers with tank 2 are working on division 50A, the average rank is 1.4. (i.e., $\frac{3\times1+2\times2}{5}$).

The same approach was considered for training an ANN regarding the welding operation. Samples of historical data used for training the ANN for forecasting welding durations are presented in Table D-4.

Table D-4. Samples of historical data regarding welding operation.

Weight(kg)	Shift (Day 1- Night 2)	Rank	Complexity intensity	Duration(Min.)
69	2	3	2	65
79	2	2	2	35
81	1	1	2	40
86	2	1	2	30
122	1	2	2	55
91	2	1	2	45
214	2	2	2	70
131	1	2	2	50
139	2	1	2	35
934	1	2	2	88
85	2	1	2	66
99	2	1	2	37
137	1	1	2	48
484	1	2	2	68
512	2	2	2	40
54	2	3	3	119
111	1	1	3	90

Code for automated piece complexity capture

```
Imports System
Imports System.Collections
Imports System.Collections.Generic
Imports System.Text
Imports Excel
Imports Tekla.Structures
Imports Md = Tekla.Structures.Model
Imports System.IO
Public Class Complexity
    Dim APP As New Excel.Application
    Dim worksheet As Excel.Worksheet
    Dim workbook As Excel.Workbook
    Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
        Dim M As New Md.Model()
        Dim a As Integer
        Dim p As Integer = 2
        Dim AsmWeight As Double
        Dim AsEnum As Md.ModelObjectEnumerator =
M.GetModelObjectSelector().GetAllObjects()
        Dim piececounter As Integer = 1
        Dim assemblycounter As Integer = 1
        While AsEnum.MoveNext()
            StatusTextBox.Text = "Reading..."
            StatusTextBox.BackColor = Color.Red
            Dim assembly As Md.Assembly = TryCast(AsEnum.Current,
Md.Assembly)
            If assembly IsNot Nothing Then
                ' do something with assembly
                ' for exam: write assembly's number
                Dim pos As String = ""
                assembly.GetReportProperty("ASSEMBLY_POS", pos)
                Dim AssSuffix As String = ""
assembly.GetReportProperty("USERDEFINED.ASSEMBLY_SUFFIX",
AssSuffix)
                If InStr(pos, "(?)") <> 0 Then
                    pos = Strip(pos, "(?)")
                End If
                pos = pos & AssSuffix
                If InStr(pos, DivTextBox.Text) <> 0 Then
                   If Not MyListBox.Items.Contains(pos) Then
                        MyListBox.Items.Add(pos)

                        AssemblyTextBox.Text =
assemblycounter.ToString
                        assemblycounter = assemblycounter + 1
                        If assemblycounter > 0 Then
                            CompxTextBox.Text =
Math.Round(piececounter / assemblycounter, 2)
                        End If
                        Dim PrtList As ArrayList =
assembly.GetSecondaries()
                        PrtList.Add(assembly.GetMainPart())
                        a = 0
```

```
                        For Each part As Md.Part In PrtList
                            '     ' do something with part
                            '     ' for exam: change their colors
                            part.Class = ColorTextBox.Text
                            '1 : Grey - 2: Red - 3: Green - 4:
Dark blue - 5: Cyan - 6: Yellow - 7: Dark magenta - 8: Dark grey
- 9: Dark red
                            ' - 10: Dark green - 11:  Dark cyan -
12: Magenta - 13: Orange - 14: Blue
                            part.Modify()
                            '*** Total number of components in
the division
                            PartsTextBox.Text = piececounter
                            piececounter = piececounter + 1
                            '*** count number of components in
each assembly
                            a = a + 1
                        Next

assembly.GetReportProperty("ASSEMBLY_WEIGHT_GROSS", AsmWeight)
                        updateExcSheet(pos, a, p,
Math.Round(AsmWeight, 0))
                        AsmWeight = 0
                        p = p + 1
                        PrtList.Clear()
                        M.CommitChanges()
System.Windows.Forms.Application.DoEvents()
                    End If
                End If
            End If
        End While
        StatusTextBox.Text = "Reading is Done!"
        StatusTextBox.BackColor = Color.Green
    End Sub
    Public Function Strip(ByVal psLine As String, ByVal
psRemoveStr As String) _
   As String
        Dim iLoc As Integer
        iLoc = InStr(psLine, psRemoveStr)
        Do While iLoc > 0
            psLine = Microsoft.VisualBasic.Left(psLine, iLoc - 1)
& _
                  Mid(psLine, iLoc + Len(psRemoveStr))
            iLoc = InStr(psLine, psRemoveStr)
        Loop
        Strip = psLine
    End Function
    Sub updateExcSheet(ByVal StAssembly As String, ByVal
Complexity As Integer, ByVal RowID As Integer, ByVal asweight As
Double)
        Dim pathstr As String = Directory.GetCurrentDirectory()
        pathstr = pathstr.Substring(0, pathstr.Length - 29)
        workbook = APP.Workbooks.Open(pathstr &
"CaptureComplexity.xls")
        worksheet = workbook.Sheets.Item(1)
        'writing data
```

```
        worksheet.Cells(RowID, 1).Value = StAssembly
        worksheet.Cells(RowID, 2).Value = Complexity
        worksheet.Cells(RowID, 3).Value =
calComplexIntens(Complexity)
        worksheet.Cells(RowID, 4).Value = asweight
        workbook.Close(SaveChanges:=True)
        APP.Quit()
    End Sub

    Function calComplexIntens(ByVal complexity As Double) As
Integer
        Dim intensComplexity As Integer
        If complexity < 4 Then
            intensComplexity = 1
        ElseIf complexity < 8 Then
            intensComplexity = 2
        ElseIf complexity < 12 Then
            intensComplexity = 3
        ElseIf complexity < 16 Then
            intensComplexity = 4
        Else
            intensComplexity = 5
        End If
        Return intensComplexity
    End Function
    ' Intelligent generation of the welding duration
    Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        Dim pathstr As String = Directory.GetCurrentDirectory()
        pathstr = pathstr.Substring(0, pathstr.Length - 29)
        MessageBox.Show(pathstr & "CaptureComplexity.xls")
    End Sub
End Class
```

Code for intelligent generation of fitting duration

```
    Function calcpro(ByVal inarray As Double()) As Double
        Dim outarray() As Double = {0, 0}
        Dim netsum As Double
        Static feature2(14) As Double
        ' inarray(1) is Weight_(kg)
        ' inarray(2) is Shift(Day1-Night_2)
        ' inarray(3) is Rank
        ' inarray(4) is Complex._intensity
        ' outarray(1) is Duration(Min.)
        If (inarray(1) < 5) Then inarray(1) = 5
        If (inarray(1) > 2915) Then inarray(1) = 2915
        inarray(1) = 2 * (inarray(1) - 5) / 2910 - 1
        If (inarray(2) < 1) Then inarray(2) = 1
        If (inarray(2) > 2) Then inarray(2) = 2
        inarray(2) = 2 * (inarray(2) - 1) - 1
        If (inarray(3) < 1) Then inarray(3) = 1
        If (inarray(3) > 3) Then inarray(3) = 3
```

```
inarray(3) = 2 * (inarray(3) - 1) / 2 - 1
If (inarray(4) < 1) Then inarray(4) = 1
If (inarray(4) > 5) Then inarray(4) = 5
inarray(4) = 2 * (inarray(4) - 1) / 4 - 1
netsum = -0.1011257
netsum = netsum + inarray(1) * -0.3540539
netsum = netsum + inarray(2) * -0.08796645
netsum = netsum + inarray(3) * 0.03752822
netsum = netsum + inarray(4) * -0.6470942
feature2(1) = 1 / (1 + Math.Exp(-netsum))
netsum = 1.054638
netsum = netsum + inarray(1) * -1.039173
netsum = netsum + inarray(2) * -0.05273196
netsum = netsum + inarray(3) * 1.097538
netsum = netsum + inarray(4) * -0.3326041
feature2(2) = 1 / (1 + Math.Exp(-netsum))
netsum = -3.696589
netsum = netsum + inarray(1) * -3.937232
netsum = netsum + inarray(2) * -0.7574236
netsum = netsum + inarray(3) * 0.1611435
netsum = netsum + inarray(4) * 2.268472
feature2(3) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.462279
netsum = netsum + inarray(1) * -0.3399979
netsum = netsum + inarray(2) * 0.1835723
netsum = netsum + inarray(3) * -0.4094017
netsum = netsum + inarray(4) * -0.3940756
feature2(4) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.4472319
netsum = netsum + inarray(1) * -0.6039337
netsum = netsum + inarray(2) * -0.1805705
netsum = netsum + inarray(3) * -0.2867619
netsum = netsum + inarray(4) * -0.1345896
feature2(5) = 1 / (1 + Math.Exp(-netsum))
netsum = 2.032277
netsum = netsum + inarray(1) * 0.9672274
netsum = netsum + inarray(2) * -1.092819
netsum = netsum + inarray(3) * -1.098984
netsum = netsum + inarray(4) * -2.51388
feature2(6) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.2557442
netsum = netsum + inarray(1) * 0.2479528
netsum = netsum + inarray(2) * -0.2199142
netsum = netsum + inarray(3) * 0.1545784
netsum = netsum + inarray(4) * 0.7925887
feature2(7) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.5466508
netsum = netsum + inarray(1) * -0.3094164
netsum = netsum + inarray(2) * 0.110145
netsum = netsum + inarray(3) * -0.3382246
netsum = netsum + inarray(4) * -0.3353107
feature2(8) = 1 / (1 + Math.Exp(-netsum))
netsum = -1.056066
netsum = netsum + inarray(1) * -5.710096
netsum = netsum + inarray(2) * 0.8229778
netsum = netsum + inarray(3) * -0.3562932
```

```
    netsum = netsum + inarray(4) * -2.726255
    feature2(9) = 1 / (1 + Math.Exp(-netsum))
    netsum = -0.3148523
    netsum = netsum + inarray(1) * -4.706576
    netsum = netsum + inarray(2) * 2.080217
    netsum = netsum + inarray(3) * -0.2481121
    netsum = netsum + inarray(4) * -0.02105724
    feature2(10) = 1 / (1 + Math.Exp(-netsum))
    netsum = -2.194835
    netsum = netsum + inarray(1) * -3.345181
    netsum = netsum + inarray(2) * 2.714793
    netsum = netsum + inarray(3) * -0.433072
    netsum = netsum + inarray(4) * -2.513535
    feature2(11) = 1 / (1 + Math.Exp(-netsum))
    netsum = 0.577292
    netsum = netsum + inarray(1) * -0.3218301
    netsum = netsum + inarray(2) * -0.390413
    netsum = netsum + inarray(3) * 0.08293828
    netsum = netsum + inarray(4) * 0.05790515
    feature2(12) = 1 / (1 + Math.Exp(-netsum))
    netsum = 0.4619731
    netsum = netsum + inarray(1) * -0.1844887
    netsum = netsum + inarray(2) * 0.2073608
    netsum = netsum + inarray(3) * -0.0308763
    netsum = netsum + inarray(4) * -0.1202346
    feature2(13) = 1 / (1 + Math.Exp(-netsum))
    netsum = 0.2350025
    netsum = netsum + inarray(1) * -0.1005819
    netsum = netsum + inarray(2) * 0.1224153
    netsum = netsum + inarray(3) * 0.006307739
    netsum = netsum + inarray(4) * 0.1628662
    feature2(14) = 1 / (1 + Math.Exp(-netsum))
    netsum = 0.8551772
    netsum = netsum + feature2(1) * 0.5718973
    netsum = netsum + feature2(2) * -0.8949994
    netsum = netsum + feature2(3) * -2.801017
    netsum = netsum + feature2(4) * 0.6159478
    netsum = netsum + feature2(5) * 0.3036578
    netsum = netsum + feature2(6) * -2.192927
    netsum = netsum + feature2(7) * -0.2948131
    netsum = netsum + feature2(8) * 0.5521762
    netsum = netsum + feature2(9) * -3.586964
    netsum = netsum + feature2(10) * 5.255617
    netsum = netsum + feature2(11) * -3.092133
    netsum = netsum + feature2(12) * -0.001254916
    netsum = netsum + feature2(13) * 0.3857389
    netsum = netsum + feature2(14) * 0.1707626
    outarray(1) = 1 / (1 + Math.Exp(-netsum))
    outarray(1) = 235 * (outarray(1) - 0.1) / 0.8 + 5
    If (outarray(1) < 5) Then outarray(1) = 5
    If (outarray(1) > 240) Then outarray(1) = 240
    '*************************

    Return outarray(1)
End Function
```

Code for intelligent generation of welding duration

```
Function calcWelDur(ByVal inarray As Double()) As Double
    Dim outarray() As Double = {0, 0}
    Dim netsum As Double
    Static feature2(14) As Double
    ' inarray(1) is Weight_(kg)
    ' inarray(2) is Shift(Day1-Night_2)
    ' inarray(3) is Rank
    ' inarray(4) is Complex._intensity
    ' outarray(1) is Duration(Min.)
    If (inarray(1) < 5) Then inarray(1) = 5
    If (inarray(1) > 2915) Then inarray(1) = 2915
    inarray(1) = 2 * (inarray(1) - 5) / 2910 - 1
    If (inarray(2) < 1) Then inarray(2) = 1
    If (inarray(2) > 2) Then inarray(2) = 2
    inarray(2) = 2 * (inarray(2) - 1) - 1
    If (inarray(3) < 1) Then inarray(3) = 1
    If (inarray(3) > 3) Then inarray(3) = 3
    inarray(3) = 2 * (inarray(3) - 1) / 2 - 1

    If (inarray(4) < 1) Then inarray(4) = 1
    If (inarray(4) > 5) Then inarray(4) = 5
    inarray(4) = 2 * (inarray(4) - 1) / 4 - 1
    netsum = -0.1011257
    netsum = netsum + inarray(1) * -0.3540539
    netsum = netsum + inarray(2) * -0.08796645
    netsum = netsum + inarray(3) * 0.03752822
    netsum = netsum + inarray(4) * -0.6470942
    feature2(1) = 1 / (1 + Math.Exp(-netsum))
    netsum = 1.054638
    netsum = netsum + inarray(1) * -1.039173
    netsum = netsum + inarray(2) * -0.05273196
    netsum = netsum + inarray(3) * 1.097538
    netsum = netsum + inarray(4) * -0.3326041
    feature2(2) = 1 / (1 + Math.Exp(-netsum))
    netsum = -3.696589
    netsum = netsum + inarray(1) * -3.937232
    netsum = netsum + inarray(2) * -0.7574236
    netsum = netsum + inarray(3) * 0.1611435
    netsum = netsum + inarray(4) * 2.268472
    feature2(3) = 1 / (1 + Math.Exp(-netsum))
    netsum = 0.462279
    netsum = netsum + inarray(1) * -0.3399979
    netsum = netsum + inarray(2) * 0.1835723
    netsum = netsum + inarray(3) * -0.4094017
    netsum = netsum + inarray(4) * -0.3940756
    feature2(4) = 1 / (1 + Math.Exp(-netsum))
    netsum = 0.4472319
    netsum = netsum + inarray(1) * -0.6039337
    netsum = netsum + inarray(2) * -0.1805705
    netsum = netsum + inarray(3) * -0.2867619
    netsum = netsum + inarray(4) * -0.1345896
    feature2(5) = 1 / (1 + Math.Exp(-netsum))
    netsum = 2.032277
    netsum = netsum + inarray(1) * 0.9672274
```

```
netsum = netsum + inarray(2) * -1.092819
netsum = netsum + inarray(3) * -1.098984
netsum = netsum + inarray(4) * -2.51388
feature2(6) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.2557442
netsum = netsum + inarray(1) * 0.2479528
netsum = netsum + inarray(2) * -0.2199142
netsum = netsum + inarray(3) * 0.1545784
netsum = netsum + inarray(4) * 0.7925887
feature2(7) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.5466508
netsum = netsum + inarray(1) * -0.3094164
netsum = netsum + inarray(2) * 0.110145
netsum = netsum + inarray(3) * -0.3382246
netsum = netsum + inarray(4) * -0.3353107
feature2(8) = 1 / (1 + Math.Exp(-netsum))
netsum = -1.056066
netsum = netsum + inarray(1) * -5.710096
netsum = netsum + inarray(2) * 0.8229778
netsum = netsum + inarray(3) * -0.3562932
netsum = netsum + inarray(4) * -2.726255
feature2(9) = 1 / (1 + Math.Exp(-netsum))
netsum = -0.3148523
netsum = netsum + inarray(1) * -4.706576
netsum = netsum + inarray(2) * 2.080217
netsum = netsum + inarray(3) * -0.2481121
netsum = netsum + inarray(4) * -0.02105724
feature2(10) = 1 / (1 + Math.Exp(-netsum))
netsum = -2.194835
netsum = netsum + inarray(1) * -3.345181
netsum = netsum + inarray(2) * 2.714793
netsum = netsum + inarray(3) * -0.433072
netsum = netsum + inarray(4) * -2.513535
feature2(11) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.577292
netsum = netsum + inarray(1) * -0.3218301
netsum = netsum + inarray(2) * -0.390413
netsum = netsum + inarray(3) * 0.08293828
netsum = netsum + inarray(4) * 0.05790515
feature2(12) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.4619731
netsum = netsum + inarray(1) * -0.1844887
netsum = netsum + inarray(2) * 0.2073608
netsum = netsum + inarray(3) * -0.0308763
netsum = netsum + inarray(4) * -0.1202346
feature2(13) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.2350025
netsum = netsum + inarray(1) * -0.1005819
netsum = netsum + inarray(2) * 0.1224153
netsum = netsum + inarray(3) * 0.006307739
netsum = netsum + inarray(4) * 0.1628662
feature2(14) = 1 / (1 + Math.Exp(-netsum))
netsum = 0.8551772
netsum = netsum + feature2(1) * 0.5718973
netsum = netsum + feature2(2) * -0.8949994
netsum = netsum + feature2(3) * -2.801017
```

```
    netsum = netsum + feature2(4) * 0.6159478
    netsum = netsum + feature2(5) * 0.3036578
    netsum = netsum + feature2(6) * -2.192927
    netsum = netsum + feature2(7) * -0.2948131
    netsum = netsum + feature2(8) * 0.5521762
    netsum = netsum + feature2(9) * -3.586964
    netsum = netsum + feature2(10) * 5.255617
    netsum = netsum + feature2(11) * -3.092133
    netsum = netsum + feature2(12) * -0.001254916
    netsum = netsum + feature2(13) * 0.3857389
    netsum = netsum + feature2(14) * 0.1707626
    outarray(1) = 1 / (1 + Math.Exp(-netsum))
    outarray(1) = 235 * (outarray(1) - 0.1) / 0.8 + 5
    If (outarray(1) < 5) Then outarray(1) = 5
    If (outarray(1) > 240) Then outarray(1) = 240
    outarray(1) = outarray(1) / (0.6 + 0.17 * Rnd())
    Return outarray(1)
End Function
```