PLANNING AND CONTROL OF HIGH-RISE BUILDING CONSTRUCTION

Arash Ranjbaran

A Thesis

in

The Department of Building, Civil and Environmental Engineering

Presented in the partial fulfillment of the requirements for the degree of
Master of Applied Science at Concordia University
Montréal, Québec, Canada

February 2007

Your file Votre référence
ISBN: 978-0-494-28891-7
Our file Notre référence
ISBN: 978-0-494-28891-7

ABSTRACT

Planning and Control of High-Rise Building Construction

Arash Ranjbaran

This research presents an integrated model for the planning, scheduling and control of high-rise building construction. The model has three main modules: planning, scheduling and progress reporting. The developed model is flexible, providing an open architecture that allows for user interaction. The proposed model is designed using object-oriented modeling and builds on earlier developments made by El-Rayes (1997) and Moselhi. It also captures the experience gained from an actual case study of a recently constructed institutional high-rise building. The case study assisted in the model developments in establishing the job logic vis-à-vis the relationships among the project activities and progress reporting, supporting exception reporting and the generation of different types of reports. The case is of a 17 floor institutional high-rise building constructed for Concordia University in downtown Montréal. The planning module of the developed model was designed using knowledge extracted from the literature and the experience gained from the case study. It has the ability to assign both the contractor's own work force and subcontractors simultaneously to reflect the status of current practice. The scheduling module uses resource driven scheduling techniques for repetitive construction and considers the impact of the learning curve and crew work continuity. An optimization algorithm designed using dynamic programming is embedded in the model to enable the optimization of the project schedule considering the following priorities: 1) cost; 2) time; and 3) their combined effect; in a manner similar to what is known as A + B bidding in highway construction. The model was coded using Microsoft Visual C++ Version 6.0 and Microsoft Visual C++ .NET. The tracking and control model uses the Earned Value technique and is capable of providing various efficient reports, to suite the needs of different levels of management. Numerical examples are presented to illustrate the essential features of the developed model.

ACKNOWLEDGEMENTS

I would like to express my deepest and most sincere appreciation to my thesis supervisor, Professor Osama Moselhi, for his guidance, technical assistance, inspiration, support, patience and encouragement. His valuable insights and commitment far exceeded the call of duty and his input to the thesis greatly helped to improve the contents of the thesis significantly. I will always owe the fulfillments of my professional and academic achievements to him.

I wish to express my thanks to the faculty and staff of the Department of Building, Civil and Environmental Engineering, who always were very hospital and supportive and made my experience at Concordia University very pleasant and unforgettable. I would like to specially thank Dr Ahmed Hassanein, who was much more than a mentor and acted the same way a caring elder brother would. I am very grateful of my colleagues in the construction management lab, who were always very supportive, and I will never forget their kindness.

I would also like to extend my gratitude to my mother, my late father and my brother who always encouraged me, believed in me and provided a great deal of psychological support. I would also like to thank the professionals working in the industry who took the time to share some of their experience with me. Finally yet importantly I would like to thank my physiotherapist, George Demirakos who helped me return to normal health conditions, without his help, I would not have been able to walk, sit or stand normally anymore.

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION

1.1. INTRODUCTION....1

1.2. DEFINITION....2

1.3. REPETITIVE CONSTRUCTION....4

1.4. CONSIDERATIONS IN THE PLANNING AND CONTROL OF HIGH-RISE CONSTRUCTION....4

1.5. RESEARCH OBJECTIVES....7

1.6. THESIS ORGANIZATION....8

CHAPTER TWO: LITERATURE REVIEW

2.1. INTRODUCTION....10

2.2. PLANNING AND SCHEDULING TECHNIQUES AND THEIR LIMITATIONS....13

2.2.1. Bar Charts....13

2.2.2. Network Scheduling Techniques....14

2.3. SCHEDULING TECHNIQUES FOR REPETITIVE CONSTRUCTION....18

2.3.1. The Line of Balance Scheduling Technique (LOB)....18

2.3.2. Linear Scheduling Methods (LSM)....20

2.4. OPTIMIZED SCHEDULING....21

2.4.1. Operations Research Models....21

2.4.2. Simulation Models....25

2.4.3. Artificial Intelligence (AI) Models....25

2.4.4. Resource Driven Scheduling and Optimization of Repetitive Activities.....26

2.5. PLANNING AND CONTROL OF HIGH-RISE BUILDINGS....31

2.6. THE LEARNING CURVE (PRODUCTIVITY IMPROVEMENT THROUGH REPETITION) 35

2.6.1. *The Straight-Line Model* *37*

2.6.2. *The Cubic Model* *38*

2.7. COMMERCIALLY AVAILABLE COMPUTER SOFTWARE 40

2.8. SUMMARY AND CONCLUSION 43

CHAPTER THREE: PROPOSED MODEL

3.1. INTRODUCTION 45

3.2. OBJECT-ORIENTED MODELING 45

3.3. ANALYSIS STAGE 49

3.3.1. *Current Practice* *51*

3.3.2. *The Industry's Opinions and Requirements* *59*

3.3.3. *Object Model* *61*

3.3.4. *Dynamic Model* *67*

3.4. DESIGN STAGE 68

3.4.1. *Assistance in Planning Module* *70*

3.4.2. *Input Module* *73*

3.4.3. *Scheduling Module* *75*

3.4.4. *Subcontractor Scheduling Algorithm Stage One* *82*

3.4.5. *Subcontractor Scheduling Algorithm Stage Two* *83*

3.4.6. *Integrated Time and Cost Control Module and Calculations* *86*

3.5. SUMMARY 95

CHAPTER FOUR: COMPUTER IMPLEMENTATION

4.1. INTRODUCTION 97

4.2. MODEL 98

4.3. GRAPHICAL USER INTERFACE (GUI) 99

4.3.1. Menus, Toolbar and Status bar 100

4.3.2. Dialog Boxes 103

4.4. INPUT AND OUTPUT 108

4.5. SYSTEM VALIDATION 109

4.5.1. Numerical Example I 110

4.5.2. Numerical Example II 113

4.5.3. Numerical Example III 118

4.5.4. Numerical Example IV 120

4.6. SUMMARY 125

CHAPTER FIVE: CASE STUDY

5.1. INTRODUCTION 126

5.2. DESCRIPTION OF THE CASE STUDY 126

5.3. SCHEDULE GENERATED USING HRPS 132

5.4. SUMMARY 135

CHAPTER SIX: CONCLUSIONS

6.1. SUMMARY AND CONCLUDING REMARKS 136

6.2. RESEARCH CONTRIBUTIONS 139

6.3. RECOMMENDATIONS FOR FUTURE RESEARCH 140

REFERENCES....142

APPENDIX I: HRPS RESULTS....148

APPENDIX II: CONSTRUCTION PROCESSES PICTURES (CONCORDIA UNIVERSITY'S EV BUILDING)....154

APPENDIX III: SAMPLES OF THE PRECEDENCE RELATIONSHIPS USED IN THE PLANNING MODULE....162

APPENDIX IV: HRPS GENERATED SSCHEDULE FOR CONCORDIA UNIVERSITY'S EV BUILDING....164

APPENDIX V: MICROSOFT PROJECT CONSTRUCTION SCHEDULE FOR CONCORDIA UNIVERSITY'S EV BUILDING 181

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

Figure 1.1: High-Rise Building Definition (Skyscrapers, 2005) 3

CHAPTER TWO: LITERATURE REVIEW

Figure 2.1: Stages used in the algorithm (El-Rayes and Moselhi, 1998) 28

Figure 2.2: Object Model for LSCHEDULER (El-Rayes, 2001) 29

Figure 2.3: Logarithmic Plot of Hypothetical Learning Curve 39

Figure 2.4: Screenshot of Transcon Xposition graphical display 42

Figure 2.5: Data input into the Transcon Xposition software 42

CHAPTER THREE: PROPOSED MODEL

Figure 3.1: Commonly used Work Breakdown Structure 51

Figure 3.2: Activities and Relationships of High-Rise Construction (El-Rayes, 1997) 63

Figure 3.3: Proposed Object Model 66

Figure 3.4: State Diagram 68

Figure 3.5: Proposed Model's Architecture 70

Figure 3.6: State Diagram of Repetitive Class 85

CHAPTER FOUR: COMPUTER IMPLEMENTATION

Figure 4.1: Input and output of developed software 98

Figure 4.2: Main Screen of "HRPS" 99

Figure 4.3: HRPS Main Menu Bar 100

Figure 4.4: Project Menu 100

Figure 4.5: Planning Menu 101

Figure 4.6: Activity Menu 102

Figure 4.7: Relationship Menu 102

Figure 4.8: Display Menu 103

Figure 4.9: Project Start Date 103

Figure 4.10: Weather and Learning Curve Options 103

Figure 4.11: Project Schedule and Cost Optimization Options 104

Figure 4.12: Data Date Entry for Project Control and Schedule Updating 104

Figure 4.13: Integrated Time and Cost Control Dialog 104

Figure 4.14: Standard Planning Template Data Entry 105

Figure 4.15: Selective Activities Planning Template Data Entry 105

Figure 4.16: Non-Repetitive Activity Data 106

Figure 4.17: Typical Repetitive Activity Data Entry 106

Figure 4.18: Non-Typical Repetitive Activity Data Entry 106

Figure 4.19: Number of Crew(s) Data Entry 107

Figure 4.20: Crew Data Entry 107

Figure 4.21: Subcontractor Data Entry 108

Figure 4.22: Relationship Definition Data Entry 108

CHAPTER FIVE: CASE STUDY

Figure 5.1: 9th Floor Plan 128

Figure 5.2: Building Section B-B' 129

Figure 5.3: Building Elevation from Ste-Catherine Street 130

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

Table 1-1: Canada's top five cities (Emporis, 2005) 2

Table 1-2: The 10 highest buildings in Montreal (Emporis, 2005) 2

CHAPTER THREE: PROPOSED MODEL

Table 3-1: Data Members of Project Class 75

Table 3-2: Main Member Functions of Project Class 75

Table 3-3: Data Members of Activity Class 77

Table 3-4: Main Member Functions of Activity Class 78

Table 3-5: Data Members of Subcontractor Class 80

Table 3-6: Main Member Functions of Subcontractor Class 80

Table 3-7: New Data Members of Repetitive Class used in Project Controls 94

Table 3-8: Control Module Functions 95

CHAPTER FOUR: COMPUTER IMPLEMENTATION

Table 4-1:Project Menu Functions 101

Table 4-2: Planning Menu Functions 101

Table 4-3: Activity Menu Functions 102

Table 4-4: Relationship Menu Functions 102

Table 4-5: Activity Durations for Numerical Example I 111

Table 4-6: Optimized Schedule Generated by Selinger (1980) 111

Table 4-7: Optimized Schedule Generated by Russell and Caselton (1988) 112

Table 4-8: Optimized Schedule Generated by HRPS 112

Table 4-9: Quantities of Work for Numerical Example II 114

Table 4-10: Crew Data for Example II 115

Table 4-11: Generated Schedule for Example II 116

Table 4-12: Optimized Schedule Generated for Example II 117

Table 4-13: Crew Data for Numerical Example III 118

Table 4-14: Subcontractor Data for Numerical Example III 119

Table 4-15: Generated Schedule for Numerical Example III 119

Table 4-16: Quantities of Work for Numerical Example IV 120

Table 4-17: Crew Data for Numerical Example IV 120

Table 4-18: Subcontractor Data for Numerical Example IV 120

Table 4-19: Generated Schedule for Example IV 121

Table 4-20: Cumulative Data on the 17th Day 123

Table 4-21: Earned Value Analysis at the Project Level 123

Table 4-22: Earned Value Analysis at the Activity Level 124

CHAPTER ONE

INTRODUCTION

1.1. Introduction

In modern times skyscrapers and high-rise buildings define the skylines of cities, which in turn, more often than not, represent the financial strength or growth of that area. The increase of the world's population has resulted in cities growing in altitude as a contrast to older times when they would occupy more space in area on the ground. This has resulted in an increase in population density, which so far has seemed to be a feasible solution to accommodating the demand for work, education and housing for such large numbers.

A city such as Montreal with a population of 1.8 million people has over 400 high-rise buildings. In Canada's top five cities there are over 3000 high-rise buildings (Emporis, 2005). This shows that high-rise buildings have become increasingly popular over the years. Asia has the most high-rise buildings, with over 29,000 buildings making up for 34.19% of the entire world's high-rise buildings followed by North America, with almost 25,000 buildings constituting for 28.83% of the world. (Emporis, 2005) Hong Kong City has 7,426 high-rise buildings, the largest number of high-rise buildings in a single city. Table 1-1 shows the population and the number of high-rise buildings in Canada's largest cities. Table 1-2 shows the tallest high-rise buildings in the city of Montreal.

Table 1-1: Canada's top five cities (Emporis, 2005)

City	Population	Buildings
Toronto	2,481,494	1,615
Vancouver	583,296	519
Montreal	1,812,723	410
Ottawa	774,072	278
Edmonton	716,450	226

Table 1-2: The 10 highest buildings in Montreal (Emporis, 2005)

Building Name	Height(m)	Floors	Year
Le 1000 rue de la Gauchetière	205	51	1992
Le 1250 Boulevard René-Lévesque	195	47	1992
Tour de la Bourse	190	47	1964
Place Ville-Marie	188	43	1962
La Tour CIBC	180	45	1962
Le 1501 McGill College	158	36	1992
Le Complexe Desjardins	152	40	1976
Maison des Coopérants	146	34	1987
Place Montréal Trust	134	30	1988
Édifice de la Banque Canadienne Nationale	133	32	1968

1.2. Definition

The Emporis Data Committee (EDC) defines high-rise buildings as follows:

A high-rise building is defined as a building 35 meters or greater in height, which is divided at regular intervals into occupiable levels. To be considered a high-rise building an edifice must be based on solid ground, and fabricated along its full height through deliberate processes (as opposed to naturally-occurring formations) (Emporis, 2005)

A high-rise building is distinguished from other tall man-made structures by the following:

1. It must be divided into multiple levels of at least 2 meters height;
2. If it has fewer than 12 such internal levels, then the highest undivided portion must not exceed 50% of the total height;
3. Indistinct divisions of levels such as stairways shall not be considered floors for purposes of eligibility in this definition.

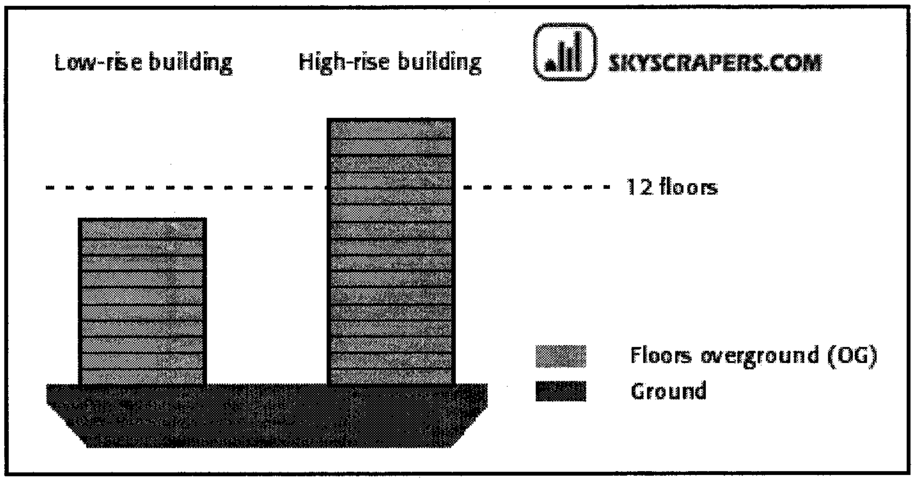


Figure 1.1: High-Rise Building Definition (Skyscrapers, 2005)

Any method of structural support which is consistent with this definition is allowable, whether masonry, concrete, or metal frame. In the few cases where such a building is not structurally self-supporting (e.g. resting on a slope or braced against a cliff), it may still be considered a high-rise building but is not eligible for any height records unless the record stipulates inclusions of this type (Emporis, 2005).

1.3. Repetitive Construction

High-rise construction falls under the category of repetitive construction. In repetitive construction, the same activities are repeated a number of times at different locations. Other examples of repetitive construction can be highways, multiple housing projects, dike construction and pipeline construction projects. The benefits of scheduling construction activities related to such projects using repetitive construction scheduling techniques are as follows:

1. Maintaining crew work continuity
2. Benefiting from the learning curve effect
3. Reducing the number of fires and hires
4. Minimizing crew and equipment idle time
5. They produce resource driven schedules

Traditional scheduling techniques have proven not to be capable of providing all these benefits simultaneously therefore their use has been ruled out. Further disadvantages of using these techniques will be fully discussed in Chapter Two.

1.4. Considerations in the Planning and Control of High-Rise Construction

Various challenges should be considered in the development of an effective model for the planning and control of high-rise building construction. These challenges have been identified as results of the construction process. During the construction phase of a project, job superintendents place resource utilization as a priority, thus creating resource driven schedules will help reflect the actual construction process better. High-rise construction consists of both repetitive and non-repetitive activities therefore in order to provide an effective schedule, the non-repetitive activities have to be scheduled using resource leveled traditional network techniques and the repetitive activities have to be scheduled using resource driven scheduling techniques associated with repetitive construction. Thus, another important consideration in high-rise construction is integrating traditional network scheduling techniques to the algorithms used to schedule repetitive construction in an effective and practical environment. It is important to note that effective scheduling is a result of good planning, which has been described as knowledge intensive, experience based and to the same degree is site dependent (Moselhi, 2004). Access to the job site is among such issues, since commercial and institutional high-rise buildings are built in crowded urban centers, where issues such as traffic control and limitations on work time can become governing constraints.

A large quantity of work within high-rise buildings is subcontracted, consequently the development of a resource driven algorithm capable of utilizing both direct labor and subcontractors will add to its practicality. It is very challenging to effectively incorporate subcontractors into a construction schedule due to the fact that in many occasions the project management team has a minimal amount of information on the performance of the subcontractors and consequently integrating their input in the development of practical schedules. In many instances the amount of information gets better as time passes and therefore schedule tools should allow for efficient updating.

Another inevitable challenge is calculating realistic activity durations. Every project is unique therefore estimating activity durations requires input from the field. However, using data from similar jobs, productivity factors and experience provide an acceptable basis for project scheduling. Due to the repetitive nature of high-rise construction, the effect of the learning curve cannot be neglected and therefore must be incorporated into estimating activity durations, using suitable models and factors.

The development of a tracking and control methodology tailored to suite the requirements and characteristics of high-rise construction must be counted among the most important challenges. Developing a tracking and control methodology for high-rise construction is complicated, mainly due to the simultaneous presence of subcontractors, the contractors own work force, the large number of activities involved in such projects and the concurrent presence of more than one crew for each activity. An efficient tracking and control methodology should be able to identify problems in a timely manner so that corrective action can be taken. It should also be able to assist in evaluating the performance of subcontractors and general contractors' own work force.

The generation of specialized reports, which can articulately address the various needs of different levels of management is also important. The level of details in the reports has to be tailored depending on whom they are being generated for. Every superintendent and subcontractor should be able to generate the schedule associated to the work that has been assigned to them. Progress reports showing the actual amount of work performed and comparing it with the project baseline is an essential part of project control. These reports have to be able to show the amount of progress done on each unit, by each subcontractor, the project as a whole and the amount of progress of a specific activity.

1.5. Research Objectives

The objective of this research is to study planning and scheduling of high-rise buildings and develop an integrated methodology that supports planning, scheduling and control of institutional and high-rise building projects. The methodology should also be able to address the concerns and considerations regularly faced in the industry and thus providing flexible modeling, which can be used by different levels of management. The methodology used to achieve this objective is outlined below.

1) To study the planning and scheduling process of an actual case of a high-rise building with a focus on understanding the dependency and degree of overlap among the different activities considered in the schedule. And to study related literature focusing on scheduling of repetitive activities.
2) Study the earlier work carried out at Concordia University with a focus on expanding the scheduling engine developed by El-Rayes (1997) for repetitive construction and tailoring it to meet the unique domain specific needs of institutional and high-rise building construction.

3) Study the learning curve impact on generated schedules.

4) Extract the knowledge from the case study and literature to support project planning.

5) Allow for schedule updating as the project proceeds.

6) Develop a resource driven algorithm for subcontractors involved in high-rise construction.

7) Formulate an effective algorithm capable of providing efficient tracking and control for high-rise construction, based on the earned value method. The algorithm should be flexible and allow the project to be analyzed from different aspects.

8) Provide valuable and flexible reports at different levels of management

9) Implement the proposed algorithms in a user-friendly prototype software system.

10) Provide database support to store historical data and assist the user in the planning stage.

1.6. Thesis Organization

Chapter 2 presents a review of previous research on resource driven scheduling, the existing scheduling techniques used for repetitive construction, planning and control of high-rise buildings, and the learning curve. It also presents the limitations associated with scheduling high-rise buildings using both deterministic and probabilistic network techniques.

Chapter 3 introduces the analysis and design stage of the proposed object-oriented model for planning, scheduling and controlling high-rise building construction. The design stage describes classes, scheduling calculations and consideration of the learning curve in the planning stage. It also includes a section that describes current industry practices on scheduling. A tracking and control algorithm is presented which describes how the developed model is capable of performing Earned Value calculations for repetitive construction.

Chapter 4 presents the implementation stage of the present object-oriented model. The chapter describes the development of a prototype software system, and outlines its main modules, and its input and output. The chapter also presents numerical examples to validate the developed model and its optimization, scheduling with contractors and control features.

Chapter 5 describes an actual case study, based on which the planning module was developed, and analyzes the project schedule generated using network scheduling techniques and compares it with the schedule generated using the developed model.

In Chapter 6, a summary of the results of this study are stated, the main research contributions are highlighted, and recommendations for future research are presented.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

The main purpose of Construction Management is to deliver a project on time, within a certain budget and in accordance to pre-defined quality standards. Time, cost and quality create a triangle, which is called the *fundamental triangle of project management*. The planning of a project is carried out in a manner to accommodate these criteria. That is why planning is the most knowledge intensive, ill-structured and challenging phase in the development of a project cycle (Moselhi, 2004). In planning, a project is broken down into smaller identifiable components, either activities or work packages, and then the interrelationships among them are established. Scheduling is the next step, where activity durations are predicted using prevailing productivity levels and available resources in conformity with the planning phase. (Moselhi, 2004)

The creation of a realistic schedule also serves purposes other than the one stated above, in fact its use is not just limited to the construction stage, and it is extended to the pre-construction and post-construction stages as well. The schedule provides the necessary insight for the project manager or his/her representative to identify the required resources and plan for their timely allocation ahead of time. Cash flows, the assignment of work crews, delivery of material and equipment allocation are such considerations. Schedules are also appropriate tools for project control. Popular project control tools, such as the

Earned Value Technique, compare the *as-planned schedule* with the *as-built schedule* and evaluate the performance of the project by doing so (Moselhi, 1993). With the obtained information, timely and necessary *corrective action* can be taken if there is any deviation from the planned performance. In the post construction stage, project schedules serve as a reference to facilitate construction claims and disputes. Similar to project control, the *as-built* and *as-planned* schedules are compared and the different types of delays (non-excusable, excusable non-compensable and excusable compensable) are identified, liquated and compensated (Alkass *et al*, 1996).

Different types of construction projects are planned and scheduled according to their characteristics, in order to achieve an optimum schedule in respect to *the fundamental triangle of construction management*. Since the planning and scheduling of high-rise buildings are discussed in this thesis, it will be looked into more deeply and a higher concentration of detail will be deployed. Among the available categories existing in construction, high-rise buildings fall into the category of repetitive construction projects.

Repetitive construction projects are made up of a number of similar or identical units (El-Rayes, 1997). Examples of repetitive construction could be high-rise buildings, housing projects, pipeline networks, highways, airport runways, railways, bridges, tunnels, sewer mains, mass transit systems, wind energy farms, water pipes and civil infrastructure.

Repetitive construction has a sub-category called linear construction projects. The difference between linear construction projects and repetitive construction projects lies in how units are defined. Linear construction projects are of interest to the planners and schedulers of repetitive construction projects because of their similarity. In repetitive construction projects, units have physical significance. Clear examples of such definition could be a typical floor in a high-rise building or a model house in a housing project. In linear construction projects the definition of a unit is a much more compelling task. A highway project can be a good example to show the complexity. A repetitive unit could either be a number of kilometers of the highway, or it could be defined by a portion of the length contracted to subcontractors. It could also perhaps be a combination of the two. As seen, a clear and visible definition cannot be stated; therefore another hurdle in linear scheduling is the actual definition of a repetitive unit (Vorster *et al.*, 1992; Hassanein, 2002).

In either repetitive or linear construction projects, construction activities can be either *repetitive* or *non-repetitive* (El-Rayes, 1997), for example the flooring activity that is repeated in each typical floor of a high rise building can be considered *repetitive*, while the pouring of the foundation which is only done once is considered *non-repetitive*. The *repetitive* activities themselves are divided into two categories, *typical* and *non-typical* activities. *Typical* activities are those who have the same duration in every repetitive unit whereas *non-typical* activities are those who may have different durations in every repetitive unit (Selinger, 1980;

Johnston, 1981; Chrzanowski and Johnston, 1986; Russell and Caselton, 1988; El-Rayes and Moselhi, 1998). The constant activity time assumed in *typical* activities comes from the assumption that the activity work quantities are equal, so with constant productivity rates, the activity durations will also be equal (Hassanein, 2002).

2.2. Planning and Scheduling Techniques and Their Limitations

There is no clear evidence as to how and when planning and scheduling for construction activities started. There are many great structures left behind from ancient times and it seems impossible for them to have been built randomly or by trial and error on site. What seems evident is that the job-logic has always been something clear in the mind of the constructors. For example, the builders through the ages have always known whether from experience or by pure logic, that in order to build a roof for shelter, they need beams, which cannot be built without the presence of columns, therefore before they could build the roof, they had to build the columns followed by the beams.

2.2.1. Bar Charts

Bar charts are still popular and are used in construction today. Its graphical nature makes it easily understood by all levels of management and supervision, thus becoming an effective mean of communication between engineers and foremen. It has also emerged as a tool to identify the required resources, and resource leveling and allocation is often done using Bar charts. The most notable deficiency of bar charts is the fact that, they are not able to illustrate the logical interrelationship between the activities. The critical path of activities, that affect the total project duration, can not be identified as a consequence of this disadvantage.

2.2.2. Network Scheduling Techniques

The limitations of Bar Charts called for the need to develop new techniques in scheduling. Network techniques are either deterministic or probabilistic.

Deterministic Network Scheduling Techniques

The Arrow Diagram Method (ADM) and the Precedence Diagram Method (PDM) are the two common deterministic network techniques available. These methods are also known as the Critical Path Method (CPM). In ADM, activities are represented by arrows that connect two nodes to each other while the nodes can be considered as benchmarks. In PDM, activities are the nodes themselves and the arrows represent the job logic. PDM has a number of advantages over ADM; there is no need for 'dummy activities' in PDM, and ADM is only able to consider a finish to start precedence relationship whereas PDM can consider various types of relationships such as finish to start, start to start, start to finish and finish to finish with lag and lead times.

Probabilistic Network Scheduling Techniques

As far as probabilistic network scheduling techniques are concerned, the 'Program Evaluation and Review Technique' (PERT) considers three different durations for each activity, the most optimistic, the most likely and the most pessimistic durations. This characteristic enables the scheduler to model the uncertainty associated with the duration of each activity. Other than the fact that PERT has the limitations of deterministic network methods, its use is limited due to the assumptions it is based on. An example of the assumptions could be the fact that it assumes the total project to have a normal statistical distribution, when all the individual activities have beta statistical distributions, the error in this assumption can be neglected if there are more than six activities (Moselhi, 2004).

Limitations of Network Scheduling Techniques

The shortcomings of network scheduling techniques in regard to repetitive construction are many. Network scheduling methods were originally developed for military purposes, where time is of essence and the financial and quality considerations are of the second degree of importance. From a practical point of view network techniques are poor models of the construction process because no attempt is made to include the priorities of the controller of the process, the site superintendent, who thinks in terms of work crews, not activities, in terms of idle resources, not float, and in terms of queuing crews instead of critical paths (Kavanagh, 1985).

Network techniques emphasize on minimizing the total project duration and thus make the fundamental, unrealistic assumption that resources are unlimited and centrally controlled. Contractors, however, are primarily interested in minimizing the resource input and maximizing resource utilization (Kavanagh, 1985). In fact there is evidence that contractors avoid using Gantt charts and network schedules in highly repetitive projects such as high-rise buildings since these projects are highly resource constrained in addition to being time constrained (Arditi and Albulak, 1986).

Network scheduling techniques do not take into account the learning curve effect on productivity. Therefore they forecast the entire project with constant productivity rates when they are variable in reality.

Another disadvantage of network scheduling techniques is that for complex projects, a network schedule tends to become extremely complex and detailed. It is observed that field personnel, who are not usually trained to understand the methodologies of network scheduling, get confused by the complex schedule (Chrzanowski and Johnston, 1986). For example, if a high rise building with 25 typical floors is to be considered and if the work of each floor can be broken down into 22 activities, the project network would consist of 550 activities, which complicates the understanding of the schedule and control process.

In addition to all of the mentioned shortcomings of network scheduling, there are hidden interrelationships between activities due to resource constraints which are not shown in the actual network. The methods used to calculate the float of activities can not depict this constraint, therefore there is actually a 'Phantom Float' which alters the network calculations and perhaps even the total project time (Kim and de la Garza, 2003).

The application of traditional scheduling techniques to repetitive construction adds to the shortcomings of these methods. Bar charts, CPM and PERT are incapable of maintaining crew work continuity even if resource allocation or the Resource-constrained Critical Path Method (RCPM) is utilized. Crew work continuity makes maximum use of the learning curve effect, maintains a constant workforce by reducing the number of hires and fires, minimizes the crew and equipment idle time, retains skilled labor and last but not least it has proven to be an effective resource utilization strategy for repetitive construction (Ashley, 1978; Birrell, 1980; Hassanein, 2002). Today attempts are being made to maintain crew work continuity by utilizing multi-skilled labor in network scheduling techniques as well. This, not only enhances productivity but it also increases job security and employment durations for laborers, which results in better morale for the workers (Burleson *et al*, 1998; Gomar *et al*, 2002).

Traditional scheduling methods also impose a high level of detail, which make masks the uncertainty and production decision realities that actually control the project. They also complicate the implementation of multiple-crew strategies. They can not provide data for the progress of individual crews alongside the progress of the project itself (Hassanein, 2002).

2.3. Scheduling Techniques for Repetitive Construction

Scheduling Techniques for Repetitive Construction are resource driven and tailored in a fashion to meet its requirements. These requirements are the precedence relationships (job logic), and the crew work continuity constraint. The crew availability constraint was later introduced. The scheduling techniques available in the literature are of two general categories. The first assumes that all activities are *typical* and the second also has the ability to consider *non-typical* activities. The methods in the first category are commonly known as the Line of Balance (LOB) scheduling techniques and the others are simply known as Linear Scheduling Methods (LSM).

2.3.1. The Line of Balance Scheduling Technique (LOB)

As mentioned above, due to the nature of repetitive construction, bar charts and network scheduling methods have proven to be inefficient means for the planning, scheduling and control of such construction projects. This reality led to the emergence of resource driven scheduling techniques, which were developed in order to fulfill the requirements of repetitive construction, namely the work continuity constraint. The US Navy developed the Line of Balance (LOB) technique in the early fifties (Harmelink, 2001). It was primarily a technique used in the manufacturing industry designed for planning, organizing, integrating, tracking and controlling production flow of repetitive units in an effort to meet customer delivery requirements (Moselhi, 2004). The LOB technique is based on the assumption that all the activities are typical. Many other techniques were also developed using the same assumption (Al Sarraj, 1990; Arditi and Albulak, 1986; Carr and Meyer, 1974; Lumsden 1968; NBA, 1966). The National Building Agency (NBA) published a report in 1966 where the LOB technique was presented in graphical form, where each activity is represented by two parallel lines. The number of housing units was plotted versus time and the *repetitive* activities were represented by bars (El-Rayes, 1997). Carr and Meyer (1974) described the LOB technique in its present form.

It was later concluded that the *non-repetitive* activities of a high-rise building are better scheduled using network techniques whereas the *repetitive* activities are best scheduled using the vertical production method (VPM), where the floor levels are drafted against time on a graph (O'Brien, 1975; Thabet and Beliveau, 1994). Al Sarraj (1990) formulated the LOB technique and developed its algorithms and put the technique on a mathematical basis. This enabled the development of computer-based applications of LOB. Other models have also been developed to enhance the functionality of LOB (Suh 1993; Suhail and Neale 1994; Harris, 1996).

The assumption that all the activities are *typical* has turned the LOB technique into an unpopular method for the planning and scheduling of repetitive construction. If the learning curve effect alone is considered, the activities within a construction project will not have the same duration, thus the assumption of the LOB method is not a valid one for practical use. Another limitation of the LOB technique is that it can be difficult to use on projects that require a large number of trades and operations. The problems arise from the difficulty of showing all the information on one chart, especially when monitoring progress (Halpin and Woodhead, 1976; Arditi and Albulak, 1986).

2.3.2. Linear Scheduling Methods (LSM)

In reality, repetitive construction projects consist of *non-typical* activities, because generally, activity durations can vary from one repetitive unit to another due to variations in the quantities of work encountered and/or crew productivity attained in performing the work in these units (Moselhi and El-Rayes, 1993(a)). Linear scheduling methods have the ability to consider *non-typical* activities in addition to *typical* activities. LSM, like LOB, can be displayed graphical where each activity is displayed as a single line, with either constant or changing slope, which depends on the productivity associated with the activity.

LSM was first presented in a simple two dimensional format, in which one axis displayed the project time-line and the other represented the repetitive units (Johnston, 1981). Simplicity was found to be an instrumental attribute in LSM since it has the ability to easily convey detailed information regarding the project, such as the actual work completed and resource allocation (Chrzanowski and Johnston, 1986). In order to enhance the practicality of LSM, the need for it to integrate both *repetitive* and *non-repetitive* activities was recognized and addressed (O'Brien, 1975; Laramee 1983). The need for LSM to have the capability to consider multiple predecessors and successors for each activity was also identified (Birrell, 1980; Cole, 1991) and addressed (Birrell 1980; Russell and McGowan 1993; Russell and Wong 1993; Suhail and Neale, 1994).

2.4. Optimized Scheduling

As mentioned before, in the practice of professional construction management, time, cost and quality are of essence (Barrie and Paulson, 1992). However in the literature, the optimization of construction projects with repetitive activities has been in regard to time and cost. Attempts made to optimize LSM using mathematically based models can be categorized as follows: 1) operations research models; 2) simulation models; and 3) artificial intelligence (AI) models (Hassanein, 2002).

2.4.1. Operations Research Models

In recent attempts made to optimize repetitive construction, operation research models have proven to be the most popular tool among the researchers. Operations research models can either be linear programming models or dynamic programming models.

Reda (1990) developed a linear programming formulation that minimizes project direct costs. This formulation was based on a number of assumptions: 1) there are no lags between activities; 2) the productivity rates are constant; and 3) the possibility of work interruptions are ignored.

Selinger (1980), however, was the first to develop a dynamic programming formulation solution to optimize linear schedules. The formulation managed to maintain crew work continuity however it did not consider cost. Handa and Barcia (1986) presented a model that could account for variable production rates and maintained the work continuity constraint but did not enforce it. The model however was incapable of considering multiple crews for activities.

Russell and Caselton (1988) built on the works of Selinger (1980) and Handa and Barcia (1986) and formalized a two-variable N-stage dynamic programming solution that can provide the minimum project duration. In order to minimize project duration, pre-defined interruption vectors for activities were defined, which contradicts the work-continuity constraint; however it obtains the objective of schedule optimization in respect to time. The model also carries out a sensitivity analysis to identify the near optimal solutions. The limitations of this model are 1) it does not consider cost, thus by minimizing the overall duration of the project it minimizes indirect costs however there is no guarantee that the overall costs will be minimized as well; and 2) it is incapable of considering multiple predecessors and/or successors.

Moselhi and El-Rayes (1993 (a) & (b)) proposed an object oriented optimization model using a two-variable N-stage dynamic programming formulation that considered overall project cost as a priority as well as the learning curve effect and the impact of weather on productivity. The model had the ability to account for all different types of precedence relationships: 1) Regular-Relations; 2) Repetitive-Relations; and 3) Hetero-Relations. The model utilized objects to represent different types of activities and relationships. The design of these objects incorporates and integrates the two scheduling techniques for *repetitive* and *non-repetitive* activities. The model's optimization procedure was executed in two stages, forward and backward paths, and enforced work continuity. The above mentioned model will be explained in more detail in section 2.4.4.

Eldin and Senouci (1994) also used a two-variable N-stage dynamic programming formulation to minimize total project cost using pre-defined multiple work interruptions. The model, however, could only consider one crew per activity. El-Rayes and Moselhi (1998) developed a flexible and comprehensive algorithm that could consider: 1) type of activity (typical or non-typical); 2) number of crews assigned to work simultaneously on a task; 3) interruption of crew work continuity; 4) crew availability periods on site; and 5) the order of executing repetitive units. The model has the ability to generate interruption vectors that would minimize total construction cost by itself, unlike the model presented by Eldin and Senouci (1994) where it was necessary to input pre-defined interruptions.

The model was later enhanced in order to have the capability of considering the combined effect of time and cost, which is also known as A+B bidding (Herbsman, 1995) and is used for highway construction. In highway construction contractors bid on the cost (A) and time (B), and the lowest combined bidder (A+B) is awarded the project (Herbsman, 1995). Scheduling and optimization models for linear projects are also applicable to repetitive projects. For linear projects crews are assigned to adjacent units rather than any available unit as in repetitive construction. That is why even though this method was tailored for highway construction; it is worth mentioning and analyzing for the purpose of high-rise buildings.

Moselhi and Hassanein (2003) developed an object oriented model that employs a two-state-variable, N-stage, dynamic programming formulation coupled with a set of heuristic rules. This model was tailored to meet the requirements of linear projects and specifically highway construction; therefore it had the ability to optimize either project duration, total cost or their combined effect (A+B bidding). The model supported multiple crews to work simultaneously on any activity while accounting for: 1) multiple successors and predecessors with specified lead and lag times; 2) the impact of transverse weather obstructions, such as rivers and creeks, on crew assignments and associated time and cost; 3) the effect of inclement weather and the learning curve on crew productivity; and 4) variations in quantities of work from one unit to another.

2.4.2. Simulation Models

Computer simulation models are utilized to estimate the consequence of varying some of the deterministic input elements in the construction process. Kavanagh (1985) presented SIREN (SImulation of REpetitive Networks), a repetitive construction model coded in the GPSS language. The model would first carry out a deterministic analysis and then it would use the Monte Carlo simulation. However, the randomness inherent in simulation models renders them inaccurate in modeling construction operations, which are not random in nature (Barrie and Paulson, 1992). The complexity of developing an accurate correlation matrix limits the use of simulation models in construction (Hassanein, 2002).

2.4.3. Artificial Intelligence (AI) Models

In today's world, Artificial Intelligence (AI) plays a great role. It's involvement in everyday life can not be denied. Construction, although a conservative industry by nature, has not hesitated to utilize techniques such as neural networks, genetic algorithms and knowledge-based expert systems. Neural networks mimic the biological neural structures of the central nervous system (Adeli and Karim, 1997). A good example for the application of neural networks to construction is the model developed by Moselhi *et al* (1993) for mark-up estimation. Adeli and Karim (1997) developed a mathematical formulation to minimize the direct costs of repetitive projects. A limitation of neural networks is that it operates as a black box, providing no justification to rationalize the provided solution (Hassanein, 2002).

Genetic algorithms are another AI technique that is being used widely within the construction industry. Genetic algorithms are, like neural networks, based on a function of human anatomy; they mimic gene multiplication. A "survival of the fittest" approach is employed to determine the optimum solution among a pool of possible alternatives (Hassanein, 2002). A good example for the application of genetic algorithms to construction is the model developed by Hegazy (1999) for the optimization of resource allocation and leveling using genetic algorithms. Hegazy and Wassef (2001) developed a spreadsheet application employing genetic algorithms and Microsoft Project ® to optimize the total cost of repetitive projects. The model could only consider typical activities and a maximum of three predecessor or successors (Hassanein, 2002).

Compared to the abilities of operations-research-based models, the currently available AI models are very limited. They have however, verified the applicability of AI techniques to the optimization problem and paved the way for future progress.

2.4.4. Resource Driven Scheduling and Optimization of Repetitive Activities

El-Rayes (2001) presented an object-oriented model for scheduling of repetitive projects, which he named LSCHEDULER, which incorporated the following procedures: 1) resource driven scheduling of *repetitive* activities (El-Rayes and Moselhi, 1998); 2) optimization of repetitive construction scheduling (Moselhi and

El-Rayes, 1993 (a) and (b)); and 3) integration of *repetitive* and *non-repetitive* activities. The scheduling algorithm in LSCHEDULER considers the following three constraints: 1) logical precedence relationships; 2) crew availability; and 3) crew work continuity.

The scheduling algorithm has the flexibility and practicality to be able to consider typical and non-typical repetitive activities, single and multiple crews, activity interruption for optimization purposes, crew availability on site, sequence of construction operations, weather impact and learning curve effect. It also integrates repetitive activities with non-repetitive activities and thus considers three different kinds of relationships: 1) regular; 2) repetitive; and 3) hetero. A regular relation is a relation between two non-repetitive activities. A repetitive relationship is a relationship between two repetitive activities. A hetero relationship is a relationship between a repetitive activity and a non-repetitive activity or two activities at different repetitive units.

The scheduling algorithm is applied in two stages to every single repetitive activity. The first stage is designed to comply with the logical precedence relationships and crew availability constraints, and the second achieves further compliance with the crew work continuity constraint. The second stage is mandatory since the results for scheduling generated in the first stage do not necessarily maintain work continuity constraint for all crews. The following figure demonstrates the two stages and shows the shift generated by the second stage of the scheduling algorithm to make the schedule comply with the work continuity constraint.

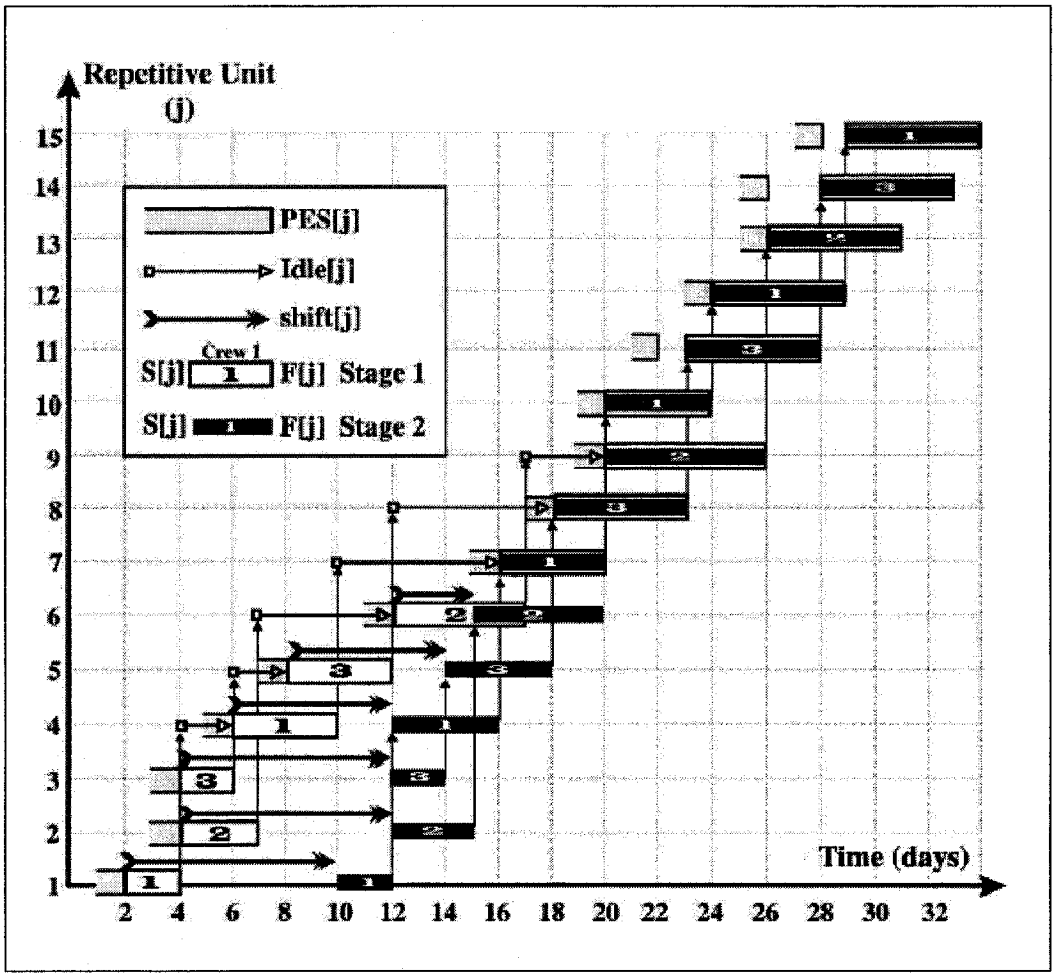


Figure 2.1: Stages used in the algorithm (El-Rayes and Moselhi, 1998)

The model was developed using Object Oriented Modeling and has the following ten classes: 1) Project; 2) Project-Data; 3) Date; 4) Activity; 5) Regular-Relation; 6) Repetitive Activity; 7) Non-Repetitive Activity; 8) Repetitive Relation; 9) Hetero Relation; and 10) Crew-Formation.

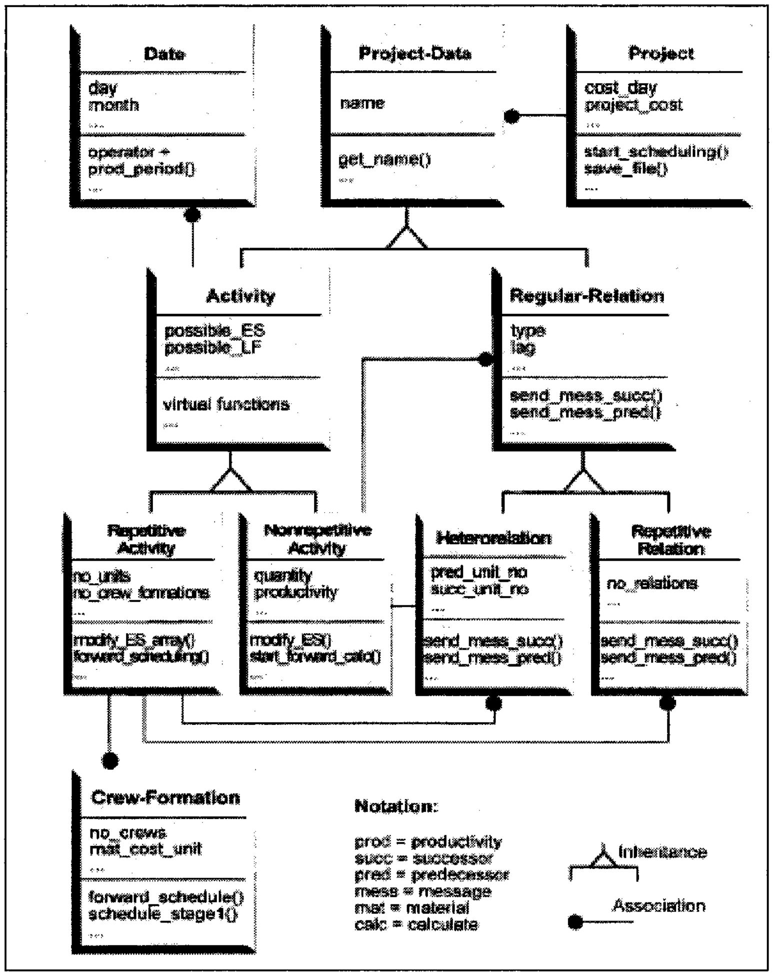


Figure 2.2: Object Model for LSCHEDULER (El-Rayes, 2001)

The Project class was designed to accommodate the unique characteristics of repetitive construction projects. It can perform a number of functions such as 1) determining the presence of an object in the project; 2) inserting a new object to the project; 3) sorting project activities; 4) initiating scheduling calculations; 5) displaying scheduling results; 6) saving project data to a binary file; and 7) opening a binary file and retrieving project data.

Project-Data class is the most generic class and includes only one string data member called name. Date class was designed to perform the following functions: 1) determine the weekday for a calendar date; 2) add the duration of an activity to a calendar date; 3) subtract the duration of an activity from a calendar date; and 4) calculate the productivity factor due to weather impact for a specific construction activity.

Activity class does not include any functions of its own; it does however include generic data, which applies to both repetitive and non-repetitive construction activities. The Repetitive Activity and Non-Repetitive Activity classes were designed to reflect the special characteristics of repetitive and non-repetitive construction activities respectively.

The Regular-Relation class represents the general precedence relationship between two non-repetitive activities: 1) finish to start; 2) start to start; 3) start to finish; and 4) finish to finish. The Repetitive-Relation class characterizes the precedence relationship between two repetitive activities. The Hetero Relation class was intended to represent the relationship between either a non-repetitive activity and a repetitive activity, or two particular units of two different repetitive activities. Finally, the Crew-Formation class was designed to reflect crew utilization data such as the number of available crews, daily output and cost while performing scheduling operations.

LSCHEDULER has an interruption algorithm and an optimization module, which can optimize 1) time; 2) cost; and 3) the combined effect of time and cost (also known as the A+B in highway projects but must not be mistaken for it because it does not include a daily road user cost). It is an object oriented model that employs a two-state-variable, N-stage, dynamic programming formulation coupled with a set of heuristic rules. The interruption algorithm automatically generates feasible interruption vectors for each crew formation during scheduling. The optimization module is also capable of incorporating cost, which can be valuable information for the team assessing the project.

2.5. Planning and Control of High-Rise Buildings

The available scheduling models for repetitive construction can not be utilized as effective tools in the planning and control of high-rise projects. For example, the application of the generic model developed by El-Rayes (2001) is limited due to its inability to: 1) consider the utilization of direct work force and/or subcontractors; 2) provide construction tracking and control abilities; and 3) generate effective reports (El-Rayes *et al*, 2002).

Moselhi and Nicholas (1990) presented an expert hybrid system for construction planning and scheduling. The model had some interesting features such as: 1) it determined the job-logic among the activities entered through an end-user interface; and 2) it modified the unimpacted durations of activities to a realistic duration. It considered the impact of reduced labor productivity due to weather.

However it was based on the assumption that all construction activities are equally affected by the same weather conditions. El-Rayes *et al* (2002) developed an object oriented model for the planning and control of housing construction, called 'Residential Planner' that could generate schedules at three levels: 1) entire project; 2) particular housing unit; and 3) an individual contractor. This model can be expanded for use in high-rise construction and a typical floor can be considered as a repetitive unit. Hassanein (2002) presented a model for the tracking and control of linear infrastructure projects. The model has a number of interesting features including: 1) incorporating both *repetitive* and *non-repetitive* activities; 2) accounting for resource availability; 3) enabling the definition of non-sequential activities; and 4) permitting the definition of multiple crews for each activity and monitoring the progress of the crews individually. This model can be modified and tailored to meet the requirements of high-rise construction as well.

Arditi *et al* (2002) developed a scheduling system for high rise building construction. In this system, two new concepts were introduced for the Line of Balance method: 1) the concept of 'flexible' unit networks; and 2) the concept of 'multi-level' LOB diagrams.

The concept of 'flexible' unit networks allows for activities to move forward or backwards in time within the network without additional cost, as long as it does not violate the existing precedence relationships or in other words the activity will have the freedom to move within its float. This is based on the following three assumptions: 1) there are no changes in the unit cost of the following items; labor, equipment, material; 2) the differential project overhead cost created by such a move should be very small in comparison to the total project cost, so that it can be neglected; and 3) the integrity of the precedence remains intact and reflects the actual relationships in the construction. The advantages of this new concept have been described as follows: 1) each unit can have a network schedule unique to itself; 2) a very large CPM for the entire project can be replaced with a series of flexible unit network schedules for each unit; 3) it can provide a simple logic for computer programming; and 4) it will have the ability to provide the associated Gantt charts for each unit.

The concept of 'multi-level' LOB diagrams allows for unit networks to be composed of three levels of activities: 1) main activities; 2) sub activities; and 3) sub-sub activities. The performance of dependent activities is restricted due to reasons such as the construction method used, technical considerations and safety risks (Arditi *et al*, 2002). The computerized high rise integrated scheduling system ('Chriss') within the model automatically generates the third level network of sub-sub activities.

Among the interesting capabilities of the developed system, one can point out the following: 1) the system can be used to simulate different situations. For example, the user can modify the unit network by changing the sequence of main activities, the sequence of main activities, the sequence of subactivities, or crew assignments of sub-subactivities. As a result the system can obtain several alternative schedules in little time; and 2) the user is asked by the system to define a precedence network when sequencing the main activities and the subactivities, the user must also assign certainty values to the precedence relationships that range from -100 to +100. The system then integrates the user's certainty values with the information stored in the database and generates a network. The user however has the option of not accepting the generated network and specifying the relations as he/she prefers.

The developed system has a number of limitations: 1) It makes the assumption that the quantities of work are all the same in all of the floors, or in order words it only considers typical repetitive activities and therefore is incapable of considering floors (units) with varying quantities of work; 2) it can only consider a single crew for each activity and is not capable of considering multiple crews for a single activity; 3) it can not integrate repetitive and non-repetitive activities together, therefore there is no such thing as a hetero relationship between activities; 4) there is no mention of cost; 5) the user will have to manually adjust the productivity rates, whereas in LSCHEDULER (El-Rayes, 2001) the effects of weather and the learning curve are done by the software itself; and 6) the developed system needs to modified slightly to be able to accept new compositions of activities.

2.6. The Learning Curve (Productivity improvement through repetition)

When performing repetitive construction activities, improvements are witnessed in productivity. The time and effort required to complete the repetitive tasks decrease as the number of repetitions increase. Thomas et al (1986) have identified eight reasons for this: 1) increased worker familiarization; 2) improved equipment and crew coordination; 3) improved job organization; 4) better engineering support; 5) better day-to-day management and supervision; 6) development of more efficient material supply systems; and 8) stabilized design leading to fewer modifications and rework.

Most researchers recognize two or three distinct phases in learning. The first takes place through an operational learning phase; workers develop familiarity with the task. The second entails learning the routine, which leads to better coordination learning the routine, which leads to more efficiency, and minor site and environmental improvements. A third phase can either happen simultaneously or succeed the first two phases if management and craftsmen make a continuous effort to improve by applying a combination of techniques.

Learning curve theory states that whenever the production quantity of a new or changed product doubles, the productivity will increase by a certain percentage of the previous unit or the cumulative average rate. This percentage establishes the slope of the learning curve and is called the learning rate. In other words the units form a geometric series with a common ratio of two. As an example, the productivity improvements or cost reductions between the fourth and the eighth, the twentieth and the fortieth and so on would express the relation. If the learning rate were to be 75 percent, it would take the eighth unit 0.75 times as long as the fourth, and the fortieth unit 0.75 times as long or cost 0.75 times less than the twentieth unit. Typical learning curves in construction fall in the 70 to 90 percent range (Oglesby, 1989). It is worthwhile to mention that although the first and second units have a ratio of two, calculations are rarely based on the first unit due to the "experience effect" which will be explained in more detail shortly.

Researchers have developed five basic mathematical models, which portray the variations in productivity as a function of the number of units produced. The various models are: 1) the straight-line power model; 2) the Stanford "B" model; 3) the cubic power model; 4) the piecewise (or stepwise) model; and 5) the exponential model. The fundamental problems associated with them are as follows: 1) determining the best predictive model; 2) understanding the factors affecting the rate of learning; 3) estimating the learning curve model parameters; and 4) quantifying the effect of delays upon performance (Thomas et al, 1986).

Due to its common use in the construction industry and this thesis, the straight-line and cubic models will be discussed in more detail;

2.6.1. The Straight-Line Model

The straight-line model makes the assumption that the learning rate is constant. It is called the straight line-model because when it is plotted on a log-log scale it forms a straight line (Wright, 1936).

$$Y_{cu} = AX^{-n} \qquad \text{(Eq. 2.1)}$$

Y_{cu}: cumulative average cost, man-hours of time per unit

A: cost, man-hours of time required for the first unit

X: unit or sequence number

n: slope of logarithmic curve

Eq. 2.1 can be written in logarithmic form as follows:

$$\log Y_{cu} = \log A - n \log X$$

The improvements in productivity can be calculated as follows:

$$S = 2^{-n} \qquad \text{(Eq 2.2)}$$

S: learning rate

2.6.2. The Cubic Model

Unlike the straight line model, the cubic model assumes the learning rate is variable due to the combined effects of previous experience and the leveling off of productivity as the activity approaches completion (Carlson, 1973).

$$\log\ Y = \log A - n_1 \log X + C(\log X)^2 + D(\log X)^3 \qquad \text{(Eq 2.3)}$$

Y = unit cost or cumulative average cost or man hours.

A = cost of first unit, a known value

n_1 = initial logarithmic slope at first unit, a known value

X = unit number

C = quadratic factor

D = cubic factor

C and D are unknown. Their values can be determined using a second data point through which Eq 2.3 passes must be known. Carlson (1973) defined this point as the standard quantity, X_{sp}. The logarithmic slope, n_{sp}, must be known at this point. Eq 2.3 can be expressed as follows for the standard quantity point:

$$\text{Log } Y_{sp} = \text{Log } A - n_1 (\log X_{sp}) + C(\log X_{sp})^2 + D(\log X_{sp})^3 \qquad \text{(Eq 2.4)}$$

X_{sp} = standard quantity, a standard number of quantity produced, a known value

Y_{sp} = unit cost or average cost per unit at the standard quantity point, a known value

n_{sp} = logarithmic slope at the standard quantity, a known value

The logarithmic slope, n_{sp}, expressed as the derivative of Eq 2. 4 as follows:

$$\frac{dy}{dxsp} = n_{sp} = +2C(\log X_{sp}) + 3D(\log X_{sp})^2 \qquad \text{(Eq 2.5)}$$

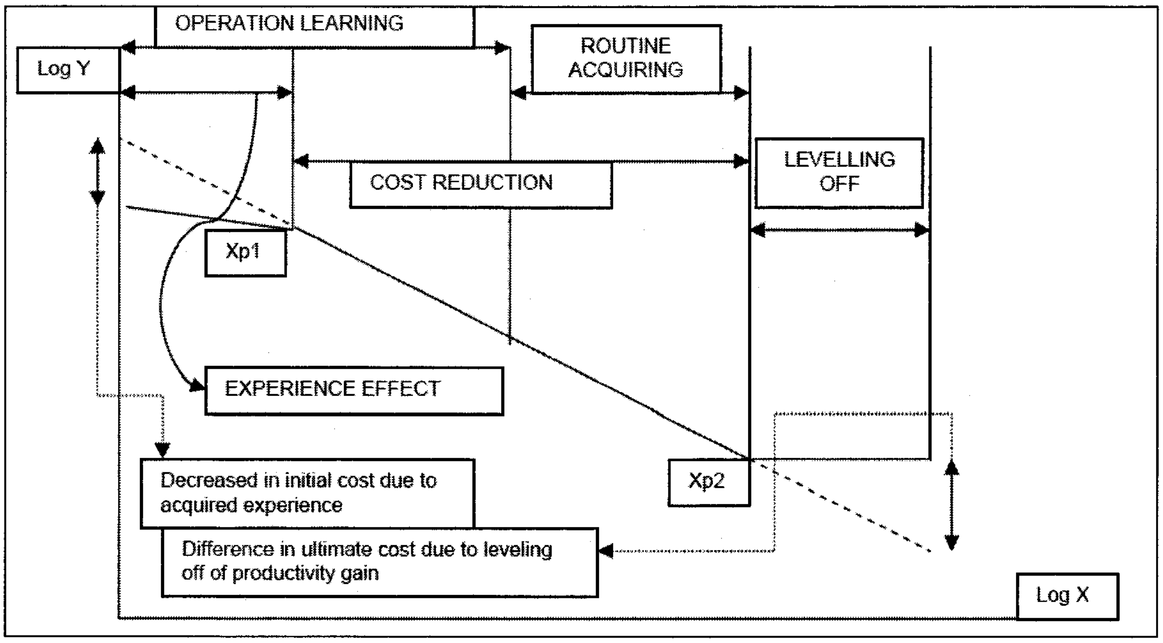


Figure 2.3: Logarithmic Plot of Hypothetical Learning Curve

Figure 2.3 shows a hypothetical learning curve which is divided into two phases; an initial operation-learning phase, during which labor productivity increases rapidly as workers acquire sufficient knowledge of the task to be performed, and a routine-acquiring phase, during which a more gradual improvement in labor productivity is attained through a growing familiarity with the job and through refinements in methods, organization, etc. Some models account for the effect of prior experience and a leveling off or stabilization of the process. The point X_{p1} denotes the end of the acquired experience phase, and X_{p2}, called the standard production point, marks the end of the learning curve effect. Beyond this point no further productivity improvements were realized (Thomas et al, 1986).

In construction the straight-line model, which assumes the learning rate is a constant value, is the most commonly used model. For forecasting purposes, using the cumulative data is preferred to using the unit data. The cumulative average curve has considerable power to smooth out the unit data. The unit curve or perhaps a moving average curve contains more appropriate data or controlling current operations (Ward and Thomas, 1984). Thomas et al, 1986 evaluated 65 construction activities and in their research, the cubic model consistently provided better results. The cubic model can be used to model the effects of prior acquired experience and the leveling off of man-hours at the end of the operation.

2.7. Commercially Available Computer Software

The most popular commercial computer software available for project scheduling are based on deterministic methods such as the critical path method. Primavera is currently the most powerful commercially available scheduling tool and is establishing a monopoly over the construction market as more and more owners demand their schedules from contractors to be done using this software and courts of law accept the analysis done by it for construction claims. Microsoft

Project is another very popular tool due to its integration powers with Microsoft Office and specifically Microsoft Excel, which is a very powerful computer spreadsheet. A powerful feature within these software is that they are able to display the schedule as bar charts from the schedule generated using network analysis techniques, therefore the bar charts take the precedence relationships and resource constraints into consideration. These software however share the shortcomings of network scheduling techniques and are unable to provide resource driven scheduling and consider the productivity changes among other things.

A recently available software named Transcon Xposition was created to reflect the graphical characteristics of linear scheduling and provides a graphical display of the linear schedule which can in turn clearly show crew allocation and crew work continuity. However the above mentioned software does not have a scheduling engine and is incapable of actually generating a schedule. It can only display the results generated by another software or a scheduler. The following two figures (Figures 2.4 and 2.5) show an example of this software using results generated from the developed software in this thesis.

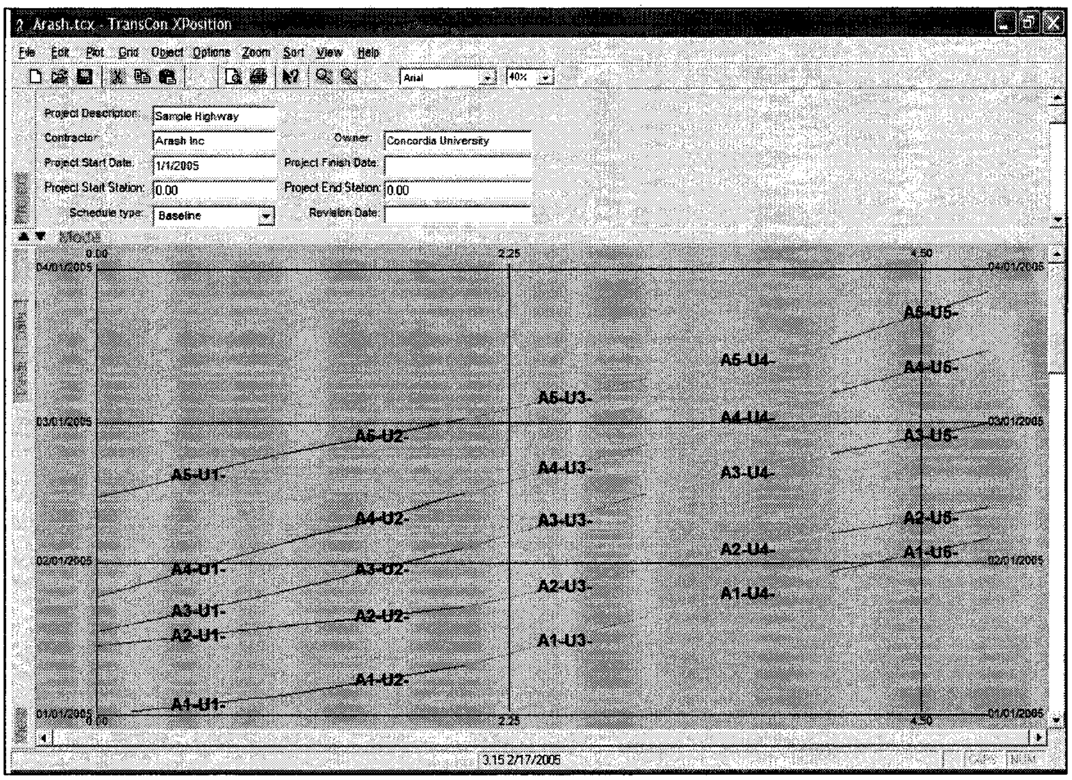
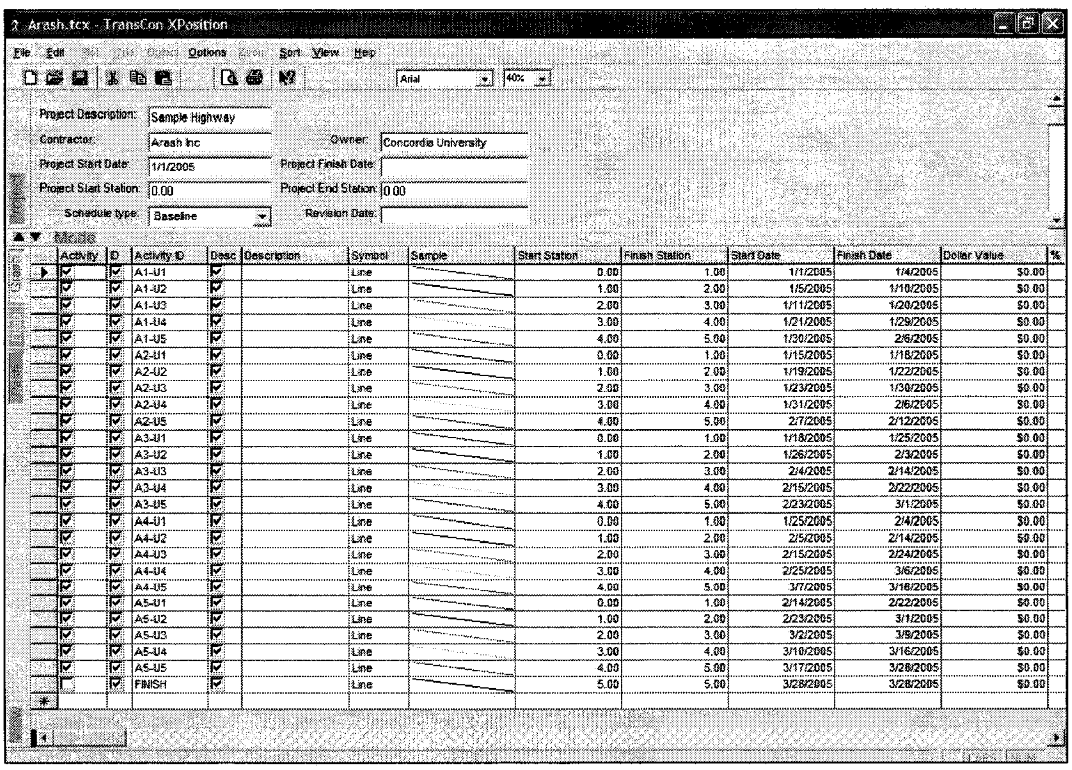


Figure 2.4: Screenshot of Transcon Xposition graphical display



Activity	ID	Activity ID	Desc	Description	Symbol	Sample	Start Station	Finish Station	Start Date	Finish Date	Dollar Value	%
✓	✓	A1-U1	✓		Line		0.00	1.00	1/1/2005	1/4/2005	$0.00	
✓	✓	A1-U2	✓		Line		1.00	2.00	1/5/2005	1/10/2005	$0.00	
✓	✓	A1-U3	✓		Line		2.00	3.00	1/11/2005	1/20/2005	$0.00	
✓	✓	A1-U4	✓		Line		3.00	4.00	1/21/2005	1/29/2005	$0.00	
✓	✓	A1-U5	✓		Line		4.00	5.00	1/30/2005	2/6/2005	$0.00	
✓	✓	A2-U1	✓		Line		0.00	1.00	1/15/2005	1/18/2005	$0.00	
✓	✓	A2-U2	✓		Line		1.00	2.00	1/19/2005	1/22/2005	$0.00	
✓	✓	A2-U3	✓		Line		2.00	3.00	1/23/2005	1/30/2005	$0.00	
✓	✓	A2-U4	✓		Line		3.00	4.00	1/31/2005	2/6/2005	$0.00	
✓	✓	A2-U5	✓		Line		4.00	5.00	2/7/2005	2/12/2005	$0.00	
✓	✓	A3-U1	✓		Line		0.00	1.00	1/18/2005	1/25/2005	$0.00	
✓	✓	A3-U2	✓		Line		1.00	2.00	1/26/2005	2/3/2005	$0.00	
✓	✓	A3-U3	✓		Line		2.00	3.00	2/4/2005	2/14/2005	$0.00	
✓	✓	A3-U4	✓		Line		3.00	4.00	2/15/2005	2/22/2005	$0.00	
✓	✓	A3-U5	✓		Line		4.00	5.00	2/23/2005	3/1/2005	$0.00	
✓	✓	A4-U1	✓		Line		0.00	1.00	1/25/2005	2/4/2005	$0.00	
✓	✓	A4-U2	✓		Line		1.00	2.00	2/5/2005	2/14/2005	$0.00	
✓	✓	A4-U3	✓		Line		2.00	3.00	2/15/2005	2/24/2005	$0.00	
✓	✓	A4-U4	✓		Line		3.00	4.00	2/25/2005	3/6/2005	$0.00	
✓	✓	A4-U5	✓		Line		4.00	5.00	3/7/2005	3/16/2005	$0.00	
✓	✓	A5-U1	✓		Line		0.00	1.00	2/14/2005	2/22/2005	$0.00	
✓	✓	A5-U2	✓		Line		1.00	2.00	2/23/2005	3/1/2005	$0.00	
✓	✓	A5-U3	✓		Line		2.00	3.00	3/2/2005	3/9/2005	$0.00	
✓	✓	A5-U4	✓		Line		3.00	4.00	3/10/2005	3/16/2005	$0.00	
✓	✓	A5-U5	✓		Line		4.00	5.00	3/17/2005	3/28/2005	$0.00	
	✓	FINISH	✓		Line		5.00	5.00	3/28/2005	3/28/2005	$0.00	

Figure 2.5: Data input into the Transcon Xposition software

As it can be seen in Figure 2.5, the data is inputted and a graphical display of the schedule is presented in Figure 2.4. Figure 2.4 shows the crew work continuity constraint and precedence relationships clearly.

Currently there is no commercially available software for the planning, scheduling and control of high-rise buildings. The commercially available computer scheduling software are not capable of reflecting the needs and characteristics of repetitive construction as they either suffer from the same shortcomings and limitations of the techniques used to build them or either they are a mere display tool. It can be concluded that it is necessary to create an algorithm capable of considering the unique characteristics of repetitive construction and applying it to a user friendly interactive environment.

2.8. Summary and Conclusion

This chapter presented a review of recent literature on scheduling of construction projects with *repetitive* activities. Traditional scheduling techniques and their shortcomings in respect to repetitive construction were also discussed. The emergence of linear scheduling methods, the pros and cons of each of the developed techniques, their considerations and limitations were also reviewed. The appropriate tools available for the planning and control of high-rise buildings were identified.

The literature reveals that in order to develop an effective tool for the optimized planning, scheduling and control of high-rise buildings, consideration should be given to scheduling: 1) the work-continuity constraint; 2) crew availability; 3) precedence relationships with lag and lead times (FS, FF, SS, SF); 4) integration of *repetitive* and *non-repetitive* activities; 5) multiple predecessors and/or successors; 6) the ability to generate work interruptions for optimization purposes (total project duration, cost and their combined effect); 7) consideration of direct work force and/or subcontractors utilization; 8) typical and non-typical activities; 9) weather impact; 10) learning curve effect; 11) provide construction tracking and control capabilities; and 12) the ability to generate effective reports.

From the literature it is also understood that determining the job logic among activities can be established based on captured knowledge and experience in that domain. This can be carried out through field studies, analyses of actual project cases and survey quantities of contractors specialized in high-rise construction.

CHAPTER THREE

PROPOSED MODEL

3.1. Introduction

This chapter presents the development of the proposed Object-Oriented model for planning and scheduling of high-rise buildings. The model is flexible and considers a number of factors that are commonly encountered in the industry. In addition to an extensive literature review, a series of interviews were carried out with industry representatives and specialists in order to identify the desirable main features of efficient planning and scheduling model. The model is developed using the Object-oriented modeling concept, which consists of three fundamental stages: analysis, design and implementation (Rumbaugh *et al.* 1991). This chapter presents the analysis and design stages of the proposed model.

3.2. Object-Oriented Modeling

Object-Oriented Design attempts to satisfy the needs of the end user via real world modeling capabilities (Khoshafian and Abnous, 1995). Object orientation is the natural way human beings interpret the world. Human beings are gifted with the ability of *abstraction* which allows them to see things as *objects* rather than thinking of the particles that they are composed of. As an example, humans think of beaches as simply beaches rather than millions of sand particles. Objects have *attributes (states)* which can describe them, like shape, color, size, etc.

They exhibit *behaviors (operations)* that define their operation, for example, a glass holds water, a lamp provides light, etc. Objects can be differentiated from each other by their attributes and behavior. Different objects that have similar attributes and demonstrate similar behavior constitute a *class*, for example a skateboard, a tricycle, a bicycle, a motorcycle, a car and a truck have much in common and can be categorized in one class. The objects in the above mentioned example all have the same general operation which is transportation; therefore they can all be placed under the "vehicle" class.

Object-Oriented-Design (OOD) benefits from class relationships and utilizes *inheritance* relationships (multiple or single), where newly developed classes in addition to absorbing the attributes and operations of an existing class, have their own unique characteristics. In the previous example the newly created motor-powered vehicle class, inherits all the attributes and operations of the vehicle class in addition to adding the new motor-powered related attributes and operations. OOD *encapsulates* data (attributes) and functions (operations) into packages called objects (Deitel and Deitel, 2003). Objects posses the *information hiding* characteristic, which prevents other objects of knowing how they function internally. However they can communicate with each other across well defined interfaces, the same way people use elevators, without fully being aware of their internal operation.

As mentioned before Object-Oriented-Design consists of three stages: analysis, design and implementation. In the analysis stage the problem has to be clearly defined. A well defined problem statement should provide the necessary system requirements. The requirements will serve as a guide to the solution of the actual problem. The output of the analysis stage must have the ability to indicate what the system is intended to execute.

The next step is to identify the required classes which would enable the system to function as desired. This is done by locating all the nouns in the problem statement. The nouns which do not perform important duties are omitted. The nouns are further filtered by using the concept of inheritance, where objects with similar attributes are placed into a class which can represent them. After the classes are identified, *class diagrams* assist in modeling the classes and their relationships that belong to the system.

The Unified Modeling Language (UML) requires each rectangle that represents a class, to be divided into three parts. The top part should contain the name of the class, the middle should comprise of the class's attributes and the bottom part should demonstrate the class's operations (Deitel and Deitel, 2003). Classes can relate with each other using *associations*. In class diagrams, the solid lines that connect classes together represent the associations between the classes.

Associations are read: "one to one", "one to two", "one to many", etc which indicate the number of objects in each class that participate in the relation with the other class. The participating classes in each association can also have a *role*. Roles assist in providing a clear definition for the relationships between the classes.

Classes can be decomposed into attributes and functions which, in turn establish the identity of the class, and are means of distinguishing the classes from one another. The attributes are identified by examining the problem statement and looking for descriptive words and phrases. An attribute is created for each descriptive word or phrase and assigned to a class. If there is any data that a class must need, that is also created as an attribute.

Object-Oriented-Design permits for objects to have states. An object's *state* describes its momentary status at any point in time. The *initial state* of an object is the default state of an object after it has been tailored to suite the requirements of the system at first. Objects states are prone to *transition* when they are invoked by receiving messages from another object. *Statechart diagrams* provide a graphical representation of how and under what circumstances an object can change its state. In other words, the flow of messages, changes in states and state transitions of an object can be portrayed with a state diagram. Each state of an object is graphically shown as a rounded rectangle that contains the state's name. These messages are also called *events*. Arrows represent *transitions* between states.

There is a variation to the Statechart diagram and it is called the *activity diagram*. The focus of an activity diagram lies on the actions that an object performs during its lifetime. These diagrams model the *internal work flow* of an object (Deitel and Deitel, 2003).

One object can interact with another object by invoking the other objects' *operations*. An *operation* by definition is a service that a class grants to *clients* (users) of that class. In general, objects do not carry out the operations spontaneously. A *receiving object* (also known as a server object) receives a message (is invoked) by a *sending object* (often called a client object).

Similar to attributes the operations of an object can be identified within the problem statement. The difference however is that the verbs and verb phrases, rather than the nouns, within the problem statement are examined which in turn helps determine the operations of the classes.

3.3. Analysis Stage

The primary step towards developing an Object-Oriented model is the *Analysis Stage*. The purpose of this stage is to analyze a real world problem and consequently state the problem clearly, develop the model that represents the problem being considered, identify its objects, functions, and relationships using abstraction, inheritance and object identity (Khoshafian and Abnous, 1995). The most prominent fruit of this stage is developing the object-model, which as stated before represents the static nature of the model and a dynamic model, which outlines the sequence of operations performed at different stages (Rumbaugh et al, 1991).

The fundamental and most crucial step in the analysis stage is to clearly understand the nature of the existing problem and its characteristics, and to provide a clear and crisp definition for the problem at hand. An intensive and comprehensive literature review was carried out to clearly comprehend the essential characteristics of planning, scheduling and control of high-rise building construction and identify the shortcomings of the methods described in the literature.

In addition to the literature review, a series of interviews were carried out with experienced professionals in the industry in order to understand the current practices in the industry and to be able to identify the areas where progress can be made. The members of the project management team and design consultants of Concordia University's new Engineering and Visual Arts Building were also interviewed to be able to grasp and reflect the industry's opinions and requirements in the proposed model. The author's personal work experience was also helpful in the development of the proposed model, by contributing to the scheduling and control modules.

3.3.1. Current Practice

In order to be able to develop a realistic schedule and fully take advantage of constructability, a representative of a contracting company was interviewed and the following is a brief illustration of how planning and scheduling is done in the industry.

Work Breakdown Structure

The first step taken in the industry towards planning and scheduling is to break down the project into identifiable smaller components. Different companies have different approaches towards their Work Breakdown Structure (WBS) but an example of a commonly used procedure is depicted in the following diagram (Peter Kiewit Sons' Inc, 2006):

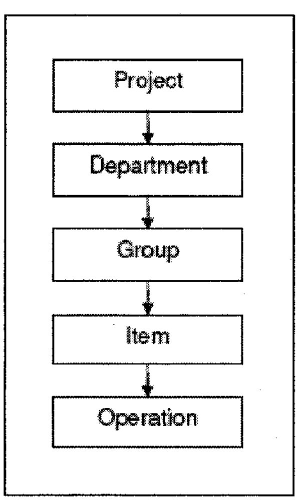


Figure 3.1: Commonly used Work Breakdown Structure

The project is divided into different departments, which depends on the nature of the job and its unique requirements, for example foundations and earthwork can be considered as a department. The departments can be further divided into groups, for example piling could represent a group. An item looks into groups with more detail, for example steel piles and concrete piles. And finally an operation is the smallest component in a Work Breakdown Structure, to continue the example, unloading, driving and cutting off the pile fall into this category. Items and operations form a cost code which will be used in the accounting system and project controls.

For high-rise buildings the following guidelines have been set: 1) the minimum item cost should be the maximum of either $10,000 or 0.1 % of the bid; 2) the minimum operation cost should be the greater of either $1,000 or 0.01 % of the bid; and 3) the minimum duration of an operation should not be less than one week.

Planning

Professionals in the industry describe planning as a team effort. They believe that those who implement the plan should be the ones developing it provided they have a good knowledge and understanding of the operations that will be performed. The industry assigns the following people to directly participate in the planning stage: 1) Project Sponsor; 2) Project Manager; 3) Job Superintendent; 4) Project Engineer; 5) Schedule Engineer; and 6) Field Superintendents.

Participation by key subcontractors is vital to developing a workable plan, since as will be pointed out in the following chapter, most of the work on high-rise building construction is done by subcontractors. The representative described the necessity of their participation using the following words: "No one understands their operation better than they do, so we should not leave them out of our planning." Another benefit of including the subcontractors is that it brings them and the contractors together in a non-adversary environment to discuss the project, and among other things, problems are detected early and solutions determined well in advance.

The industry professionals also point out the importance of initially understanding the scope of work and project specifications. Once the project is clear, the following steps are taken: 1) the project is organized and divided into manageable components (WBS); 2) the core (driving) work is identified; 3) a construction strategy for completing the driving work is developed; and 4) a constructions strategy is developed for completing the supporting work. These steps will be described in more detail in the following section.

Scheduling

Once a plan is complete, it has to be communicated to the different levels of supervision and execution within the project. This is done by incorporating the plan into a schedule, in other words, scheduling is a means of communicating the project strategy and determining the amount of time required to carry it out. A schedule must be made to make optimal use of time and to ensure the activities are completed on time so that no sacrifices are made in the quality of the planning in order to meet deadlines. The development of an effective and realistic schedule requires the following steps: 1) the construction activities and tasks must be sequenced in the most logical and efficient order; 2) a duration must be assigned to each and every single activity and task within the schedule; and 3) adjustments must be made as deemed necessary to ensure the schedule matches the planned strategy and time frame required to complete the job.

Scheduling is described as a seven step process by professionals working in the industry. The first step is to understand the scope of work and specifications. This stage belongs to the planning process but can not be separated from the scheduling process. It has been described as the easiest part of scheduling however it must never be overlooked since it may have serious consequences such as heading in the wrong direction from the start. The following questions help facilitate this stage:

1. How much time is available to complete the work?
2. Is it necessary to break down the project into phases?
3. Are there any milestones which should be set or met? When and what are they?
4. What are the penalties or liquidated damages for not meeting the milestones?

5. What are the restrictions on the project?
    - Seasonal access restrictions
    - Seasonal work restrictions
    - Work hour restrictions
    - Holiday restrictions
    - Noise restrictions
    - Restricted access to certain areas
    - Other contractors or activities that may occupy the work zone
    - Utility relocations
    - Restrictions on equipment type
    - Prescribed methods of work (method specifications)
6. What is the basis of payment?
7. What are the permitting requirements?
8. What are the scheduling requirements?

The next step is planning the work. This step also belongs to the planning stage but is integral to the scheduling process. As mentioned before the emphasis here is based on the core work, since the rest of the project revolves around the core work. Focusing on the core work will help put the scheduling effort where it is needed the most.

The industry likes to keep the work simple, they believe complicating the construction strategy leads to the confusion of lower levels of management, who are responsible for the execution of the operations. Their construction strategy is developed based on how they will build the job, while remaining true to their slogan: "Don't schedule first and then plan to meet your schedule later." They rightfully believe that a good schedule is the end result of a good plan, which includes factors such as safety, quality, cost and time. Contractors like to think in terms of resources such as manpower, equipment and materials. They ask themselves the following questions before developing a plan:

1. What hazards are associated with this operation?
2. What level of quality is required?
3. What is the budget?
4. How much time is there to complete the project?

A good plan should include the following: 1) a description of the work; 2) crew sizes and trades; 3) tools and supplies; 4) permanent materials; 5) equipment; 6) engineering support; 7) work to be done by others; 8) staging area for receivables; and 9) access to the work.

Access to the work is often overlooked in planning, whereas if access is addressed in the plan, the work will be both safer and more productive and it will also improve the appearance of the work site. These are very important factors in institutional and commercial high-rise building construction since they are often constructed in busy urban centers where heavy traffic and work hour restrictions exist.

The third step is listing the work which needs to be performed. After the plan is formulated, a list of activities and tasks must be made, which are needed to be performed. The list includes the quantity, duration and productivity associated with each task or activity. It is important to be thorough while making such a list because, this list will form the basis for the schedule later, and leaving activities and tasks off such a list could affect production while the work is underway.

The fourth step is sequencing the work. In this stage many considerations should be taken into account, among the most important one can name precedence relations and resource constraints. It is best to sequence the work in the most efficient manner possible to optimize resource allocation.

The fifth step is scheduling the activities. It is important to remember to note that people and equipment should be scheduled to perform a task, and the wrong practice of scheduling activities and then looking for people to perform it should be avoided. Considering real productivity rates based on experience or similar jobs done previously helps in predicting a realistic time for carrying out the job. Resource driven scheduling is very popular with contractors because they can assign an activity to a certain person, crew or subcontractor and later monitor their progress and hold them responsible for falling behind schedule or going over the assigned budget. Contractors believe that once an activity is directly assigned to somebody, it will help ensure that the activity is completed within budget and in accordance to the pre-defined budget.

The sixth step is monitoring the progress. The schedule is a representative of the goals set for completing the work. Each activity on the schedule represents a goal, which must be met. It should be clear and understood what every item on the schedule represents, when that work should be completed and who is responsible for completing it. An effective schedule must help control the work, since it is made for that purpose.

The last step is updating the schedule. Experience has proven that something will always change, and all schedules need to be updated. When things change, it is wrong to try to impose the old schedule, the schedule should be updated. A good schedule must have the capability of being updated and showing the impact of change. A schedule should be able to identify problems so that *preventive action* or *corrective action* can be taken in a timely fashion. Among reasons that cause change, the following can be named: 1) poor execution of the plan; 2) equipment failures; 3) weather impacts; 4) changed site conditions; 5) procurement delays; and 6) change orders.

As it can be seen, the industry is interested in maximizing the use of its resources, developing flexible plans and construction strategies capable of being updated on a regular basis, and considering contractual and work site conditions and clauses.

3.3.2. The Industry's Opinions and Requirements

The project management team and design consultants for Concordia University's new Engineering and Visual Arts Building were interviewed on a number of occasions and they made note of the following:

1. Project management teams most frequently use network scheduling techniques for high-rise buildings. As indicated in the literature review, these techniques have many shortcomings and are considered neither practical nor effective due to the reasons mentioned in the previous chapter.
2. The contract for high-rise buildings are typically awarded to general contractors, which in turn subcontract most of the work to professional subcontractors which are highly specialized and efficient in their domain. Both general contractors and subcontractors are mainly concerned with crew work continuity, which can be reflected very well using scheduling techniques for repetitive construction.
3. As mentioned above, most of the work is subcontracted by the general contractor and therefore the project management team is left with very little information to provide an efficient and realistic schedule. Therefore, it is necessary to develop a model capable of scheduling the work with

minimal available information, and the ability of producing a better schedule as more data becomes available.

4. Current tracking and control techniques look at the project as one large entity and are incapable of providing specifics. It is sometimes necessary to know the production of a single crew, a floor in the high-rise building or the progress of a subcontractor and the current techniques are not able to provide such information for the project management team to be able to implement corrective action or preventive action in a timely manner.
5. The efficiency of scheduling, tracking and control of high-rise buildings can be improved by producing effective and useful reports. These reports should include both workdays and calendar days.
6. Three basic needs must be satisfied before any work activity can begin; access, logic and resources. Access not only includes physical access for the crew to the work area but also time-based access. Logic is the same as precedence relationships, the predecessors have to be completed before a successor can begin or their lag and lead requirements must be satisfied depending on the relationship (FS, FF, SS and SF). Resources must also be carefully taken into account. Their availability and procurement time must be included into a schedule.
7. Weather deserves special mention by itself because unlike Force Majeure or outside-influenced restrictions, it is inevitable.

The above findings are combined with conclusions drawn from the literature review and are incorporated into the development of the object model during the analysis stage, as described in the following section.

3.3.3. Object Model

The objective of an object model is to incorporate concepts from the real world that are essential and important, into the application. The object model should represent the static structure of the model objects. It represents the objects, their data, functions and relationships to other objects. For example, a model for the planning, scheduling and tracking and control of high-rise building construction should identify and consider objects that are instrumental to the planning, scheduling and control process; such as the number of floors, activities and precedence relationships.

The object model is represented graphically with a diagram to show the model classes, their hierarchy and association with other classes. As mentioned earlier Unified Modeling Language (UML) requires each rectangle that represents a class, to be divided into three parts. The top part should contain the name of the class, the middle should comprise of the class's attributes and the bottom part should demonstrate the class's operations.

The classes in an object model can be arranged in a hierarchy through *inheritance*, which is one of the main concepts of object-oriented modeling. A *super-class* compromises of data and functions that are common to more than one class. *Sub-classes* branch out of the *super-class,* which inherit all the data and functions of the *super-class*. This means that the highest levels of the hierarchy are the most general, while each lower level is more specialized than the one before it and once a set of characteristics are defined for a certain *super-class*, all the *sub-classes* which fall under that super-class inherit those characteristics.

A similar approach to El-Rayes (1997), Ramanathan (2000) and Hassanein (2002) is utilized to: 1) consider the different types of activities and relationships commonly used in high-rise building construction; and 2) to identify the necessary classes. The object model proposed by El-Rayes (1997) can be seen in the previous chapter, Section 2.4.4, Figure 2.2. El-Rayes (1997) considered two different types of activities: 1) repetitive; and 2) non-repetitive which can both be placed under the super-class, activity. Three different precedence relationship types were considered: 1) Regular; 2) Repetitive; and 3) Hetero, which itself can be of two different types: 1) either used to link a repetitive activity to a non-repetitive activity or vice-versa; or 2) represent a relation between two specific units of two different repetitive activities. Both the Repetitive relation and the Hetero Relation can be placed under a more general Regular Relation class.

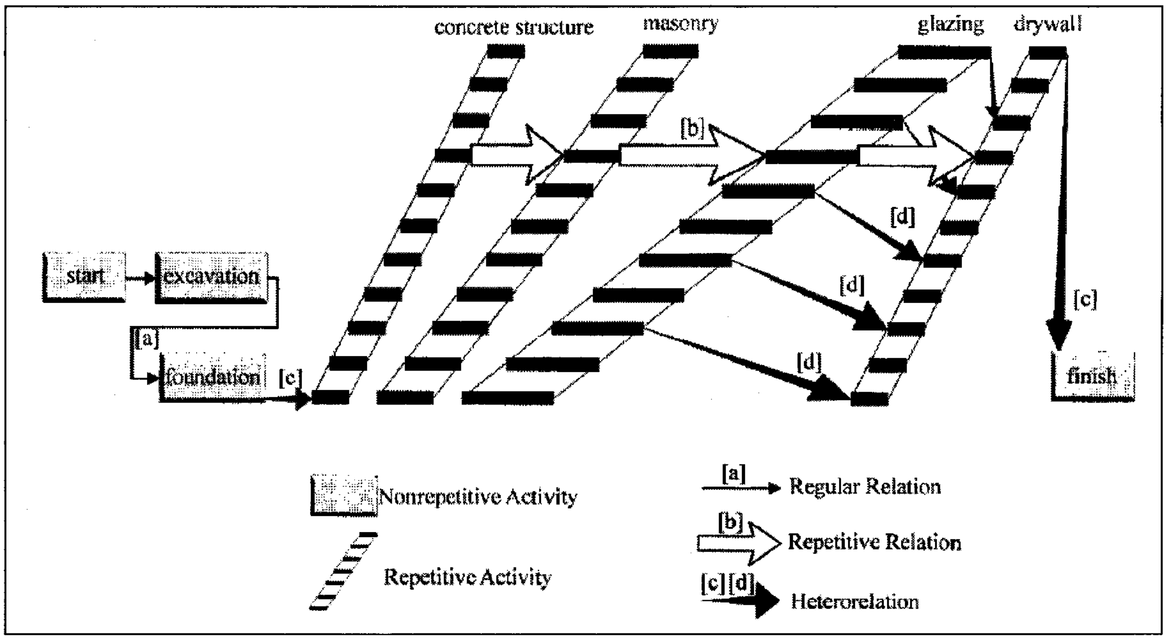


Figure 3.2: Activities and Relationships of High-Rise Construction (El-Rayes 1997)

A total of eleven classes are incorporated in the present model for the planning, scheduling and tracking and control of high-rise building construction. The proposed object model takes full advantage of the characteristics of object-oriented modeling such as inheritance and object hiding in order to perform the required functions.

The hierarchy of class is shown in Figure 3.3. This model has two major changes when compared to the model proposed by El-Rayes (1997); the modifications have been made to suit the requirements of the domain specific field of high-rise building construction. Unlike the model proposed by Ramanathan (2000) the planning module and tracking and control module are designed as separate sets of functions rather than new classes for better software engineering practices, the developed prototype software takes up less memory and works faster with the proposed structure. The changes to the model proposed by El-Rayes (1997) are shown in black in Figure 3.3.

The proposed model does not have the following constraints: 1) the number of predecessor and successor activities; 2) the number of construction crews which can be assigned to carry out the activity; and 3) the number of availability periods for each of the crews. An additional Subcontractor class is added in order to be able to incorporate subcontractors into the schedule. The one to many association between the Activity class and the newly developed Subcontractor class, enables the end user to assign both repetitive and non-repetitive activities to subcontractors. The model proposed by El-Rayes (1997) did not have such an ability and the model proposed by Ramanathan (2000) could only assign subcontractors to repetitive activities.

The other major modification was creating a one to many association between the Activity class and the Crew Formation class, which unlike the two previous models has the ability of assigning crews and resources to non-repetitive activities as well as repetitive activities. Crew formations are defined for the activity object, enabling both the repetitive and non-repetitive subclasses to inherit all of its functions and data members. Consequently, non-repetitive activities will be fully integrated into the scheduling process due to the fact that they have a crew assigned to them for their execution. This added ability enables the end user not only to benefit from network scheduling techniques for non-repetitive activities but also blocks the resource from being used in both non-repetitive and repetitive activities as long as it is in use and the activity is not completed.

It is worthwhile to note that only a single construction crew can be assigned to a non-repetitive activity, because the activity will only be carried out once therefore there is no need to assign multiple crews to it. Similar to the model proposed by Hassanein (2002) the proposed model imposes no restrictions on: 1) the number of predecessors and successors for the activities; 2) the number of crews that can assigned to a single activity; 3) the number of availability periods for each crew. It has also the ability to consider multiple subcontractors for a single activity although owners and project management teams are in most cases reluctant to subcontract a single activity to more than one subcontractor.

The proposed object model benefits from the concepts offered by object oriented modeling such as inheritance and data encapsulation. As can be seen in Figure 3.3, *Project-Data* is the most generic class and contains general data such as name, the number of floors and measurement system (metric or imperial). All of this data is inherited by all lower levels of hierarchy or sub-classes. The classes *Activity* and *Regular Relationship* are situated at the second level of the hierarchy, which means they are derived from the super-class *Project-Data*. At the third level of the hierarchy, the classes *Repetitive Activity*, *Non-Repetitive*

Activity, *Repetitive Relation* and *Hetero Relation* can be seen. *Repetitive Activity* and *Non-Repetitive Activity* are sub-classes of the super-class *Activity* and *Repetitive Relation* and *Hetero Relation* are sub-classes of the *Regular Relation* super-class.

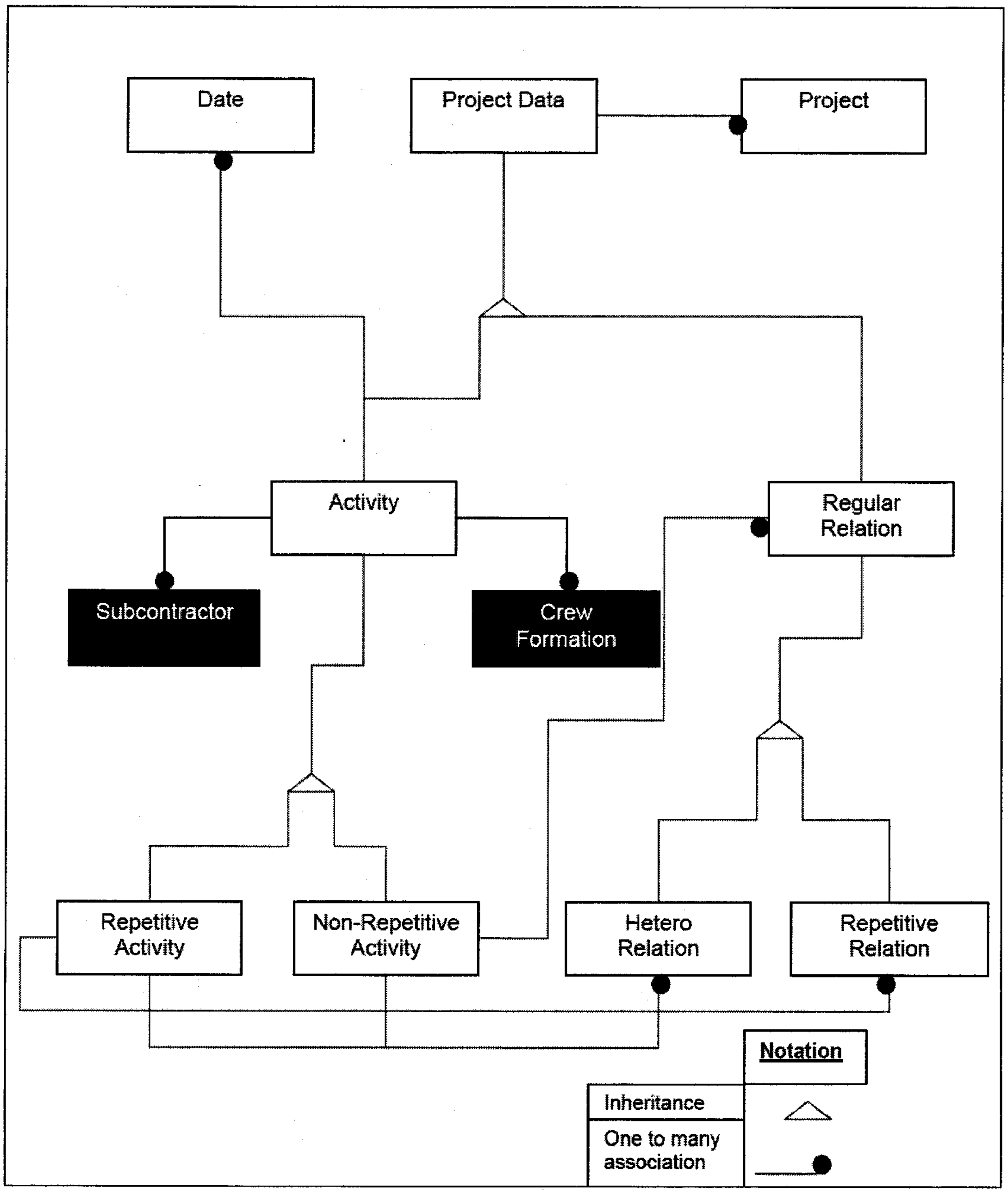


Figure 3.3: Proposed Object Model

Other than the classes in the hierarchy, four other classes are included into the present object model and have associations with the hierarchy classes. These four classes are: 1) *Project*; 2) *Data*; 3) *Crew Formation*; and 4) *Subcontractor*. The associations between these classes and the classes in the present object are not of inheritance but of a different kind. For example, each *Activity* object can have one or more associated *Crew-Formation* objects. As shown in Figure 3.3, such an association is represented by a line connecting the two classes, which has a darkened circle at its ends indicating one or more associations. *Project* class has designed functions at the project level such as adding a new activity or relation and initiating scheduling calculations. *Date* class is designed to convert workdays to calendar dates and develop calendar date schedules. *Crew-Formation* class is designed to take the different options available for labor and equipment resources into account. *Subcontractor* class is designed to consider the utilization of subcontractors.

3.3.4. Dynamic Model

The object model described in the previous section describes the static structure of an object-oriented model. It shows the structure of objects and their relationships to one another at a single moment in time, however it does not show how the attributes of the objects change over time (Rumbaugh et al 1991). A dynamic model however has the objective of describing the sequences of operations that occur to the objects. The main concepts that constitute a dynamic model are the following: 1) states; and 2) messages or events. A *state* of an object represents the values of its attributes at any given point in time. The values however do not remain constant. As time passes, objects send *messages* to one another which results in changes in the values of the attributes, and consequently a change in their *state.* A *message* acts like a stimulus from one object to another, which may lead to: 1) changing the *state* of the receiving object; 2) returning a message to the original sender; and 3) sending a message to a third object. A *state diagram* represents the pattern of *messages*, *states* and *state transitions* of an object. Dynamic models typically include a number of state diagrams, each of which represents a specific class. Section 3.2 explains how the graphical representation for state diagrams work.

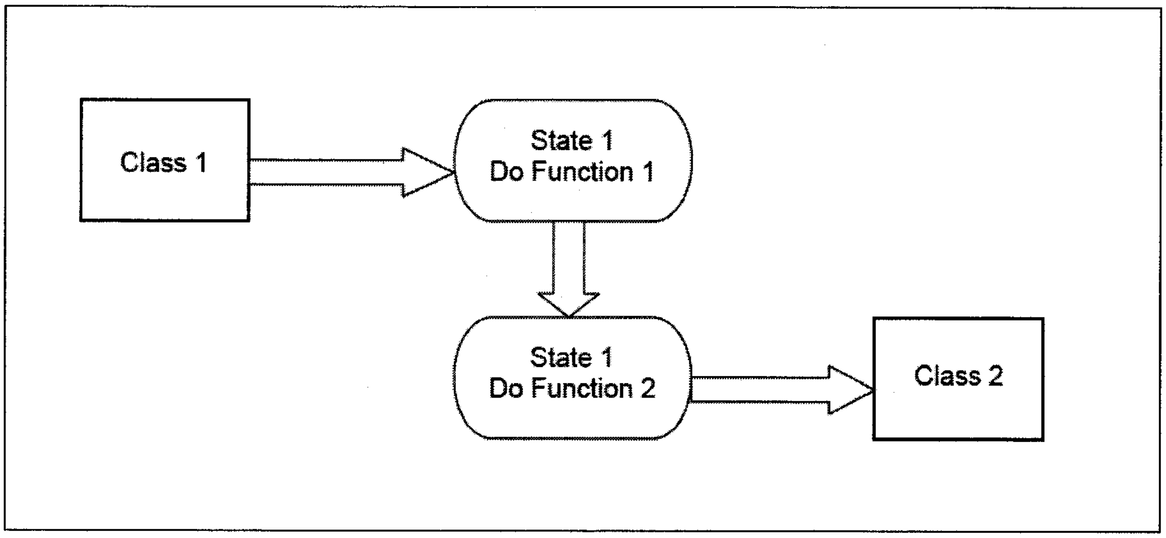


Figure 3.4: State Diagram

3.4. Design Stage

The Design Stage is the second stage in developing an Object-Oriented model and is based on the analysis stage. The proposed model's architecture comprises of six major components namely the Assistance in Planning, Input, Scheduling, Database, Control and Reports modules, which can be seen in Figure 3.5.

Each module involves one or more classes. These classes contain data and member functions designed to carry out a series of operations such as providing assistance in the planning stage, scheduling calculations, Earned Value analysis and generating reports. The following section presents a brief explanation of each class, its data members and main member functions, and provides important state diagrams with explanations.

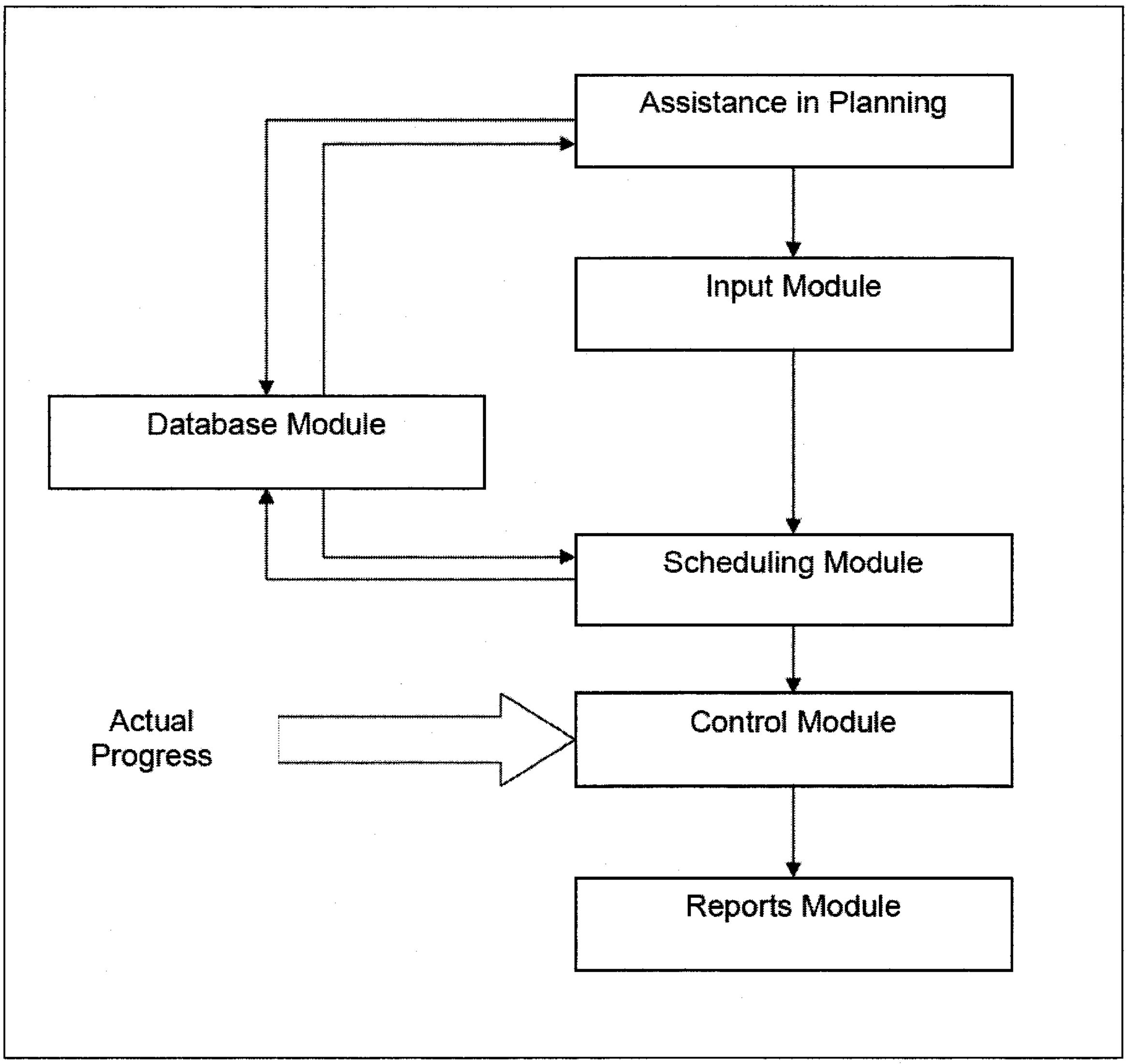


Figure 3.5: Proposed Model's Architecture

3.4.1. Assistance in Planning Module

The Assistance in Planning Module is designed to provide inexperienced planners with assistance in the planning stage. A template representing the most common job logic was developed making use of the knowledge captured during five interviews with industry representatives and an intensive study of two detailed construction schedules, including that of Concordia University's new

Engineering and Visual Arts building. . This information is supported by a database, which includes popular crew assignments, their cost and productivity for each of these activities.

The job logic itself contains both hard and soft logic. The term “hard logic” applies to a precedence relationship between construction activities which can not be changed, a good example of hard logic could be the relationship between columns and beams, the construction of beams can not start until the columns are finished. “Soft logic” on the other hand applies to a precedence relationship between construction activities which can be changed, for example; logically it is possible to paint the floors first and then the ceilings and vice versa. Tamimi and Diekmann (1998) have described soft or discretionary logic as follows: When the activities in an established network can be arranged in a variety of logical sequences. The approach taken to over come this in the planning stage is left to the optimization module, a 2 variable N stage dynamic programming method used to find the optimum time, cost or combination of the two.

The user can initially create a schedule based on the most common relationships but when it comes to optimizing the schedule, the soft logic will play its role. Based on the required optimization, the system will run all possible scenarios and choose the optimum one and output the results and the changes in the relationships.

The effect of the learning curve has also been incorporated in the assistance in planning module. The straight line model is the best model to predict productivity rates in the construction industry (Thomas et al, 1986) and has been chosen to predict the effect of the learning curve on productivity. The activities involved in high-rise construction have been divided into two groups: 1) labor intensive; and 2) equipment intensive. The labor intensive activities uses labor as its driving factor for completion whereas equipment intensive activities depend on the performance of equipment. An example for a labor intensive activity is painting where the completion of work depends on the performance of labor and an example for an equipment intensive activity could be excavation, where mechanical and hydraulic excavators are utilized.

In Chapter Two it was mentioned that typical learning curves in construction fall between the 70 to 90 percent range. The 70% to 85% learning curve range is associated with labor intensive activities and the 85% to 90% is considered for equipment intensive activities. These values are stored in the database module and are incorporated into the standard template and selective activities template of the assistance in planning module. With the new approach to considering the learning curve, the end user will have a more realistic reflection of its effect rather than using a constant 70% learning curve rate, which could lead to unrealistic durations for the activities before the leveling off or plateau of productivity.

3.4.2. Input Module

The input module is designed to get the data from the user in order to be able to perform scheduling calculations and perform tracking and control using integrated time and cost analysis techniques. Through this module the user will be able to select the activities he/she wants from the assistance in planning module, or input them in manually. The user will also be able to define various relationship types between the different chosen/input activities. The quantities for different units of the activities will be input through this module and the various crew combinations, their productivity and cost, capable of carrying out these activities will also be input. Project progress for the purpose of updating the schedule and performing integrated time and cost control of the project will also be input through this module. This module is responsible for accepting and storing the user input. It consists of two modules namely *Project-Data* and *Project*. *Project-Data* is the most generic class in the hierarchy of the object model. The *Project* class has a one to many relationship with *Project-Data* and is similar to that developed by El-Rayes (1997) and is described in Chapter Two.

Project class has been modified and tailored to facilitate the unique characteristics of the planning, scheduling and control process of high-rise building construction. New data functions have been added, and it is now capable of storing the date input by the user into a database for possible future use, and to update the current database. As shown in Figure 3.3, an object within the *Project class* has one-to-many relations with objects in *Activity* and *Date* classes. This means that multiple activities and dates can be associated with a single project.

The proposed design does not have any limitations on the number of activities that can be associated with a project. Its flexible design allows for adding, editing and removing activities. *Project class* considers the unique characteristics of high-rise building construction in its design, such as repetitive and non-repetitive activities, different types of relationships (regular, repetitive and hetero) and work performed by either subcontractors or the general contractors' own work force. This class is also capable of providing searches for the other classes since all the data is input and stored into it. *Project* class is designed such as to perform the following functions: 1) accepting and adding new data to the project; 2) sorting and arranging construction activities; 3) sorting and arranging relationships among construction activities; 4) saving and retrieving data; 5) initiating scheduling calculations; and 6) checking to see whether the precedence relations are free of logical errors, and that only one activity exists with no predecessors (start) and only one activity with no successors (finish). The main data members and member functions of *Project* class are listed in Tables 3.1 and 3.2 respectively.

Table 3-1: Data Members of Project Class

Data	Data Type	Description
measureUnit	String	The unit of measurement to be used in the project (SI system, Imperial system or user specified).
avgIndirectCost	Float	Average indirect cost of the project per day
numNonRepet	Integer	Number of non-repetitive activities in the project.
projectTotalCost	Float	Total project cost in $
projectStart	A pointer	A pointer to a date object specifying the user specified project start date
executionOrder	Integer	User specified integer value representing the execution order of the units.

Table 3-2: Main Member Functions of Project Class

Function	Description
getNumUnits()	Determines the number of floors in the high-rise project
getExOrder()	Obtains the order of execution for the housing units in the project
startScheduling()	Initiates scheduling calculations
displayMessages()	Displays error and other messages
getSubData()	Accepts subcontractor data such as name, expected duration, $ and availability period

3.4.3. Scheduling Module

The scheduling module is based on the module developed by El-Rayes (1997). The reader is referred to El-Rayes (1997), Moselhi and El-Rayes (1998(a) and 1998(b)), El-Rayes and Moselhi (1998), and El-Rayes (2001). The model developed by El-Rayes however did not incorporate work performed by subcontractors into the scheduling stage and it could not assign subcontractors to non-repetitive activities. Enhancements were made to the model to accommodate these considerations.

After the data is input, the project and activity data are passed on to the scheduling module. The scheduling module is designed to perform scheduling calculations for both non-repetitive and repetitive activities. The scheduling module performs its calculations while considering a number of practical factors commonly encountered in high-rise building construction such as the type of work force utilized (general contractors own work force or subcontractors), precedence relationships, execution order, crew availability at site, multiple crew utilization, the effect of weather and the learning curve.

The scheduling module consists of eight classes. The data members and functions were designed carefully to carry out specific and specialized functions in the scheduling process. These classes are derived based on the model developed by El-Rayes (1997) and were earlier described in Chapter Two. These classes are modified and expanded to meet the domain specific needs and requirements of high-rise building construction. Among the significant changes, one can point out that the scheduling module is now capable of assigning crews and resources to non-repetitive activities. The scheduling module can also assign both repetitive and non-repetitive activities to be carried out by subcontractors. By doing so, both the scheduling and control process will become more realistic and closer to the practices currently used in the industry. The following section will present the major modifications made in this module.

The *Activity* class is derived from the *Project-Data* class and is at the second level of the hierarchy (see Figure 3.3). The activity class is designed to reflect the characteristics of both repetitive and non-repetitive activities and consider factors affecting the scheduling of repetitive activities. This class is customized to suite the specific requirements of high-rise building construction, such as utilizing either the general contractor's own work force and/or one or more subcontractors for each of the activities.

The module considers both the repetitive and non-repetitive activity types. The Activity class was designed as a super class for repetitive and non-repetitive activity objects, enabling the use of polymorphism. The repetitive and non-repetitive activities inherit all the attributes of the Activity class, which basically serves as a base that defines their common attributes and behavior. Tables 3-3 and 3-4 respectively display the newly developed data and member functions.

Table 3-3: Data Members of Activity Class

actPerformBy	Integer	indicates who the activity is performed by, own work force or subcontractors
subNum	Integer	indicates which subcontractor is performing the work
dataBase	Integer	determines whether the user has opted for database support for crew support

Table 3-4: Main Member Functions of Activity Class

Function	Description
findActPerform	determines who the activity is carried out by input by user
findActExecOrder	determines the execution order input by user

Work on activities is performed by crew formations, which can be composed of both labor and equipment. If an activity is to be performed by the General Contractors own work force, a minimum of one crew formation has to be defined for it. A crew formation defines crews that can work simultaneously on any repetitive activity. It also stores the predicted productivity of the crew, the cost of material and labor and their availability periods on site. The model proposed by El-Rayes (1997) does not include a crew formation for non-repetitive activities. The proposed model can only assign one crew to each unit of a repetitive unit and only one crew to every non-repetitive activity.

Every activity stores a number of precedence relation objects that define the activity's relations with its predecessors and successors. An activity can be performed either by using the general contractors own work force or by subcontracting it to a highly specialized contractor. If the activity is subcontracted, one or more subcontractor objects can be defined for it. The proposed model uses an approach similar to Hassanein (2002) where crew formations are stored as dynamic lists rather than the arrays used by El-Rayes (1997), which enables addition and removal of crew formation objects. Using arrays limits the number of crew formations that can be defined for an activity, due to its rigid structure, and therefore no crew formations could be added after the activity was defined. Similar to crew formations, precedence relations are also stored in dynamic lists in the new model and thus eliminates the previous limitation of five predecessors and successors, which does not address the complexity of the relations in high-rise construction. Other benefits of utilizing dynamic lists is the implementation of better software engineering practices, using less memory and shorter processing times.

Typically, high-rise construction is awarded to a single general contractor, which in turn breaks down the scope of work and subcontracts most of it to specialized contractors and supervises them (Barrie and Paulson, 1992). To better reflect the reality of such construction, an object was developed to model subcontractors. In essence, a subcontractor is capable of carrying out a contract using either a lump sum or unit price contract. The same flexibility is given for time, the subcontractor can commit to a time per unit or a total time to complete the execution of all the units. Unlike the general contractors own work force, subcontractor crews are assumed to be available upon request. Tables 3-5 and 3-6 show the main data members and functions of the subcontractor class respectively.

Table 3-5: Data Members of Subcontractor Class

Data	Data Type	Description
subName	String Integer	Name of subcontractor
contractType	Integer	Type of contract awarded to the subcontractor. (Unit price or lump sum)
expDur	Integer	expected duration for the subcontracted activity in one unit or all units
subCost	Float	Cost of the subcontracted activity in one unit or all units

Table 3-6: Main Member Functions of Subcontractor Class

Function	Description
budgetCostSubUnit	Calculates the total cost for a subcontractor with a unit price contract
budgetTimeSubUnit	Calculates the total time for a subcontractor with a unit price contract

The three types of relation objects from El-Rayes (1997) are utilized to model precedence relations. Regular Relation represents the relation between two non-repetitive activities and is a super class to the other two objects; repetitive and hetero relation. Repetitive Relation models the precedence relation between two repetitive activities, while Hetero Relation characterizes the relation between: 1) a particular repetitive activity in a unit and non-repetitive activity; and 2) two different repetitive activity units at different locations. These objects were not altered in this study, other than enhancing encapsulation and polymorphism. The reader is referred to El-Rayes (1997) for an in-depth review of the attributes and member functions of relation objects.

The date object ensures accurate date representation. Project and Activity objects use date objects to store start and finish dates of the project and its activities. The date objects helps determine the start and finish dates of the project and its activities once the durations and project start date are given to it. Date objects also aid in determining the expected weather during the execution of an activity and therefore enabling the model to quantify the effect of weather on productivity during that period.

The scheduling module activates different functions while performing calculations depending on the type of activity (repetitive or non-repetitive) and whether it is performed by the general contractor's own work force or by subcontractors. The non-repetitive activities are scheduled using network techniques and are later resource leveled depending on the availability of crews and other resources. The repetitive activities are scheduled using the resource driven algorithm developed by El-Rayes (1997).

The subcontractors can have different options, unit price, lump sum, duration based on unit productivity or total duration for all the units. The model converts lump sum price into unit price by diving the total duration by the total estimated quantity of the work and then multiplying the calculated unit price by the quantity in each unit. It uses a similar approach if the subcontractor has a total anticipated duration for all the units. It divides the total time by the quantity of work and multiplies the calculated number by the quantity in each unit to estimate the duration required to complete each unit. Unlike the model proposed by Hassanein (2002), it does not assume the subcontractors resources are unlimited and allows the user to consider multiple constraints for the subcontractors resources and availability periods. The scheduling algorithm for repetitive activities performed by subcontractors is done in two stages.

3.4.4. Subcontractor Scheduling Algorithm Stage One

Similar to the algorithm proposed by El-Rayes (1997) for repetitive activities, the first stage of the algorithm calculates the earliest possible start for every activity satisfying logical precedence relationships, the same way network-scheduling techniques calculate early start and early finish dates for each activity. The model checks to see if all the predecessors for every activity are completed and after the latest predecessor for the activity is completed, the repetitive activity can start the next day.

The duration is calculated by dividing the quantity of work by the productivity; both of these are known and are input during the planning stage. Since moving within the units of a high-rise building does not require significant mobilization and demobilization time and costs, they are discarded. In this stage, the model also checks if the calculated dates comply with the subcontractor availability periods and only assigns the number of resources described as a limitation within those periods.

The reason for multiple constraints can be the possible addition of more resources within certain critical periods when the subcontractor is willing to accommodate the needs of the project management team. Another example can be that the subcontractor will finish another job and has the option of assigning the resources present on that project to the current one. As it can be seen, the multiple constraint option adds flexibility to the model and can better reflect the reality of the work done on the field.

3.4.5. Subcontractor Scheduling Algorithm Stage Two

As stated earlier satisfying logical precedence relations is not enough to guarantee work continuity. Figure 2.1 shows an example of the idle times, which might be a result of strictly using logical precedence relations to determine early start and finish dates for the activities. The model adds all the idle times for each repetitive activity and shifts the early start of the first unit of that activity by the sum of all the idle times for all the units. Consequently the work continuity constraint will be enforced, and there will be no more idle time for the subcontractors resources.

The optimization module, which is the same as proposed by El-Rayes (1997), a two variable n-stage dynamic programming model, might require some interruptions in work continuity to optimize project cost, project time or their combined effect. If there are any interruptions for the subcontractors work, the subcontractors indirect costs for that period of time is considered as a direct cost for the project in the overall cost.

Figure 3.6 represents the functions performed while scheduling repetitive activities through a state diagram. The state diagram depicts the changes in state of the class, as well as the messages it sends and receives. It shows the different classes involved in the two different stages of the scheduling process.

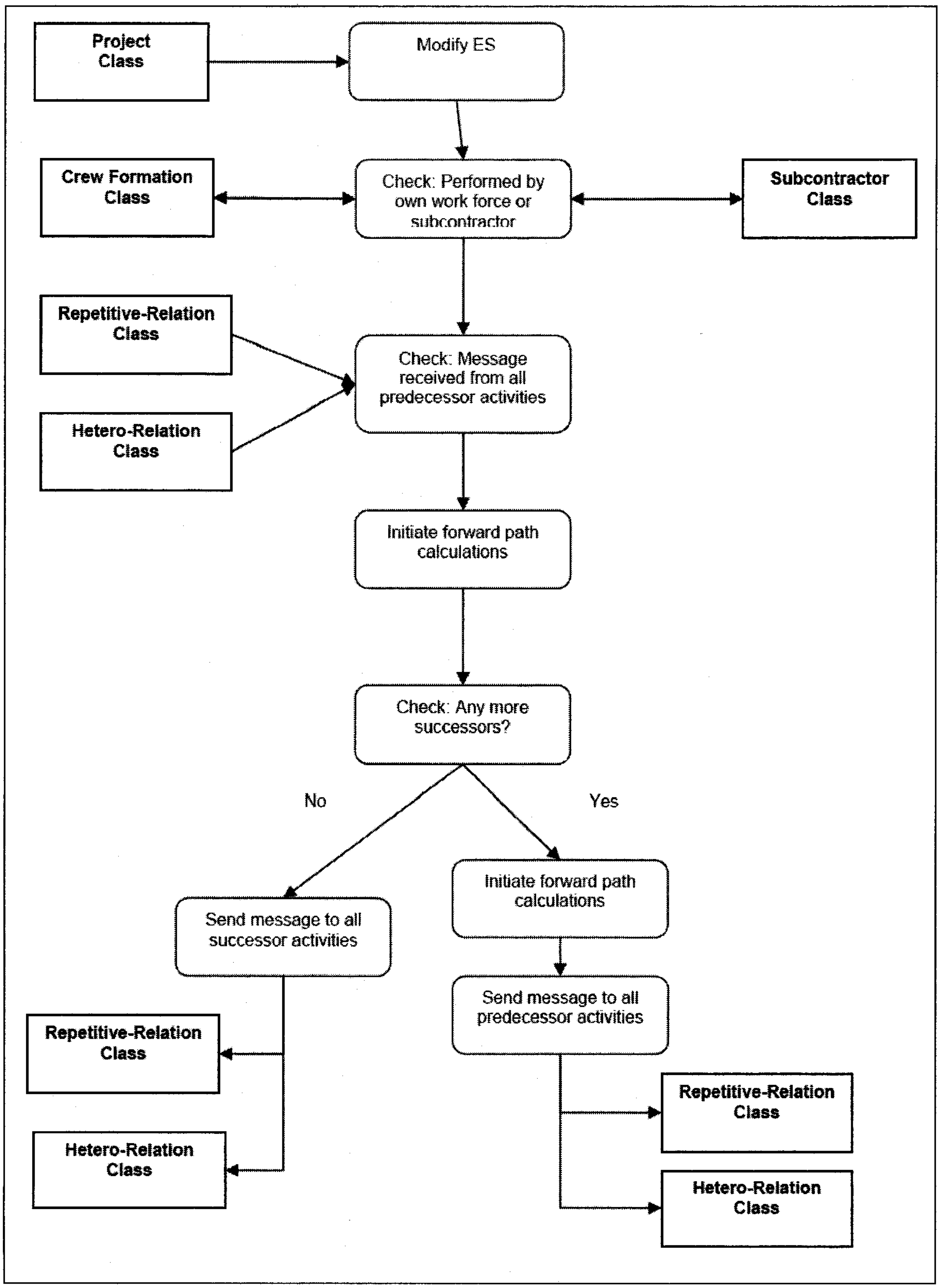


Figure 3.6: State Diagram of Repetitive Class

3.4.6. Integrated Time and Cost Control Module and Calculations

The integrated time and cost control module was built with the intent to control the progress of an on-going project on various levels. The following levels were chosen based on the literature and industry requirements: 1) The project as a whole; 2) At the subcontractor level; 3) At the general contractor's own work force level; 4) At the repetitive unit level (in most cases, on a floor by floor basis); and 5) At the activity level which will also include performance evaluation of the crews assigned to carry out the repetitive activity. This detailed tracking and control monitoring system will enable the project team to not only identify the problem(s) and deficiency(ies) well in advance, but also identify the responsible party(ies), provided that the project is updated on a consistent basis and the reporting time is chosen appropriately. An acceptable progress report time is on a weekly basis. Labor Distribution Reports (LDRs) are used to asses the direct costs and percent complete is used to asses how well the project is doing on schedule. Percent complete itself can be measured by numerous different methods, such as using the man-hours, using templates and using physical progress.

The tracking and control methodology used is based on the Earned Value. The Department of Energy in the United States of America provides the following description for Earned Value Management: Earned Value Management (EVM) is a systematic approach to the integration and measurement of cost, schedule and technical (scope) accomplishments on a project or task (DOE, 2006). By using the Earned Value methodology, the project should be able to provide objective answers to the following questions: 1) Where has the project been; 2) Where is the project now; and 3) What is the forecast for the project or where is the project going? Successful implementation of the Earned Value method requires the following input:

- Work Breakdown Structure (WBS)
- Organizational Breakdown Structure (OBS)
- Project Schedule
- Time-Phased Baseline Budget
- Cost/Resource Control Plan
- Change Control Plan

The Earned Value Method uses the following terminology:

Budgeted Cost of Work Scheduled (BCWS): this curve represents the project baseline.

Actual Cost of Work Performed (ACWP): this curve shows the actual cost.

Budgeted Cost of Work Performed (BCWP): this curve represents the earned value.

Cost Variance (CV) = ACWP – BCWP

Schedule Variance (SV) = BCWP – BCWS

Total (Accounting) Variance = ACWP – BCWS

Cost Performance Index (CPI) = BCWP / ACWP

Schedule Performance Index (SPI) = BCWP / BCWS

In order to implement integrated time and cost control on high-rise building construction the following steps have to be taken:

1) The first step is to generate the project baseline, also known as the Budgeted Cost of Work Scheduled (BCWS). In order to do so, the cost per day of every single activity in every unit has to be calculated. The cost of each activity consists of the following: 1) Labor Cost; 2) Equipment Cost; and 3) Material Cost. The scheduling module calculates these costs. They are either input by the user or extracted from the database as part of crew data. The cost per day of an activity is calculated using the below formula:

Average cost per day of an activity in a unit = $\sum$ (labor, equipment & material cost) / duration of activity

The average cost per day of every activity in every unit has to be calculated in order to generate the baseline curve. Once all the average costs per day of the activities have been calculated, their sum on every day which will be added to the accumulative total of the project direct costs and consequently the project baseline will be generated.

2) The second step required to perform project control calculations is ask the user to input a data date. The control module is meaningless unless there is a data date that will be used as a reference point in carrying out the calculations. Data date input should be the first question the system asks the user and the initial requirement for activating the control module.

3) Once the data date is defined, the model will identify all the activities that satisfy the following condition:

ES =< Data Date

(The ES is calculated by the scheduling engine)

The user must answer the following question for all the activities identified by the model and input the required data:

Is the activity complete?

If yes, the following data for each activity must be input: 1) AS (Actual Start); 2) AF (Actual Finish); 3) ACWP (Actual Cost of Work Performed); and 4) BCWP (Budgeted Cost of Work Performed).

If no, the following data for each activity must be input: 1) AS (Actual Start); 2) RD (Remaining Duration) OR % Complete (Percent Complete); 3) ACWP (Actual Cost of Work Performed); and 4) BCWP (Budgeted Cost of Work Performed).

If there are any activities behind schedule, meaning they have been scheduled to start before the data date but have not, their default values for both ACWP and BCWP will be zero. These values will depict savings in budget but will also demonstrate the slippage in schedule.

After all the activities which satisfy the ES =< Data Date condition have been identified and addressed, the model must also ask the user to all the activities which have started ahead of schedule and before or on the data date. The same question will then be asked and the corresponding data will be input.

4) All the required information is now input into the system. The following variables can now be calculated for every activity in each unit using the formulas mentioned above: 1) Cost Variance (CV); 2) Schedule Variance (SV); 3) Accounting Variance (AV); 4) Cost Performance Index (CPI); and 5) Schedule Performance Index (SPI).

For the project as a whole the sum of all the variables must be used as follows:

Cost Variance (CV) = $\sum$ACWP – $\sum$BCWP

Schedule Variance (SV) = $\sum$BCWP – $\sum$BCWS

Total (Accounting) Variance = $\sum$ACWP – $\sum$BCWS

Cost Performance Index (CPI) = $\sum$BCWP / $\sum$ACWP

Schedule Performance Index (SPI) = $\sum$BCWP / $\sum$BCWS

5) The model will isolate the activities based on the user's request into the following levels: 1) At the subcontractor level; 2) At the general contractor's own work force level; 4) At the repetitive unit level (in most cases, on a floor by floor basis); and 5) At the activity level, which will also include the assigned crews.

6) The next step will be to identify the activities with delay using the following formulas:

a) If the activity is completed:

$$\text{if} \begin{cases} \text{BCWP} > \text{BCWS} \\ \text{OR} \\ \text{AS} > \text{ES} \\ \text{OR} \\ \text{Duration} < (\text{AF} - \text{AS} + 1) \end{cases}$$

The model will then check the input database for the name of the subcontractor, or number of the crew performing the activity and include it in the report.

The percentage of schedule over-run will be calculated as follows:

Schedule Over-Run Percentage = 100 X (BCWP-BCWS)/BCWS

b) If the activity is not completed:

$$\text{if} \begin{cases} AS > ES \\ \text{OR} \\ \text{Current BCWS} < BCWP \end{cases}$$

The Current BCWS can be calculated as follows:

$$\text{Current BCWS} = (\underbrace{\text{cost per day of activity in specific unit}}_{\text{Previously calculated}}) \text{ X CCD}$$

CCD = current completed duration = (data date – actual start + 1)

The model will then check the input database for the name of the subcontractor, or number of the crew performing the activity and include it in the report.

The percentage of schedule over-run will be calculated as follows:

Schedule Over-Run Percentage = 100 X (BCWP-BCWS)/BCWS

7) The final step in the tracking and control will be the identification of activities with cost over-runs. This can be done by examining the following single condition:

ACWP > BCWP

The model will then check the input database for the name of the subcontractor, or number of the crew performing the activity and include it in the report.

The percentage of schedule over-run will be calculated as follows:

Cost Over-Run Percentage = 100 X (ACWP-BCWP)/BCWP

Table 3-7: New Data Members of Repetitive Class used in Project Controls

Data	Data Type	Description
avgQuantity	float	the average quantity in each unit
avgDuration	float	the average duration in each unit
avgCrewOutput	float	the average crew output in each unit
EarlyStart_DataData	bool	compares data date with the early start of activities
no_units	integer	the number of units
*completed	bool	a pointer to an array which determines whether the activity is completed
*startedAhead	bool	a pointer to an array which determines whether the activity has started ahead of schedule
*actualStart	integer	a pointer to an array where the actual start of the activity input by the user
*actualFinish	integer	a pointer to an array where the actual finish of the activity input by the user
*remainingDuration	integer	a pointer to an array where the remaining duration of the activity input by the user
*ACWP	float	a pointer to an array where the ACWP of the activity input by the user
*BCWP	float	a pointer to an array where the BCWP of the activity input by the user
*BCWS	float	a pointer to an array where the BCWS calculated by the model
*costPerDay	float	a pointer to an array where the cost per day of the activity calculated by the model
*scheduleVariance	float	a pointer to an array where the schedule variance calculated by the model
*costVariance	float	a pointer to an array where the cost variance calculated by the model
*accountingVariance	float	a pointer to an array where the accounting variance calculated by the model
*CPI	float	a pointer to an array where the Cost Performance Index calculated by the model
*SPI	float	a pointer to an array where the Schedule Performance Index calculated by the model

Table 3-8: Control Module Functions

Function	Description
display_control()	displays cost control results of repetitive activities
display_controlCostOverrun()	displays repetitive activities with cost over-run
display_controlDelayed()	displays repetitive activities with delayed
display_controlIncomplete()	displays over-run of incomplete repetitive activities
get_quantity()	returns the sum of unit's quantity for a repetitive activity
getAvgCrewOutput()	returns the average crew output of a repetitive activity
getAvgDuration()	returns the average duration of a repetitive activity
getAvgQuantity()	returns the average quantity of a repetitive activity
getCostPerDay()	returns the cost per day of a specific unit for a repetitive activity
setAvgCrewOutput()	sets the average crew output of a repetitive activity
setAvgDuration()	sets the average duration of a repetitive activity
setAvgQuantity()	sets the average quantity of a repetitive activity
setCostPerDay()	sets the cost per day of a repetitive activity

3.5. Summary

The proposed object model for the planning, scheduling and tracking and control of high-rise building construction was presented. This chapter presented the analysis and design stages of object oriented modeling. Interviews with professionals currently working in the industry were conducted to better understand industry practices. The conclusions from the interviews and the findings from the literature review were used to develop an object model. The module classes comprising the model were presented. Their main features were reviewed and described. The developed algorithms for the planning and tracking and control modules were presented and the enhancements for the scheduling module developed by El-Rayes (1997) were also discussed.

CHAPTER FOUR

COMPUTER IMPLEMENTATION

4.1. Introduction

This chapter presents the implementation stage of the proposed, planning, scheduling, and control model for high-rise building construction. The implementation stage follows the design stage of object-oriented modeling and is the final stage of the process (Rambaugh et al, 1991). The model is named High-Rise Planner Scheduler (HRPS) and is a window based application initially coded using Microsoft Visual C++ 6.0 and its object-oriented programming tools and later evolved into using Microsoft Visual C++ .NET. The prepared software is capable of running on Microsoft Windows XP, 2000 and ME through a user-friendly interface, which facilitates data input and retrieval. The user interface of HRPS incorporates menus, toolbars, dialog windows and multiple document interfaces windows. Figure 4.1 displays the input and the output of the software.

The software is also capable of producing reports at different levels of detail to suite the requirements of users. The generated schedule, control reports and crew assignments can be produced using both workdays and calendar dates. Microsoft Access was utilized as the database management system (DBMS). The developed software is capable of allowing for changes in both the activities and crew formations.

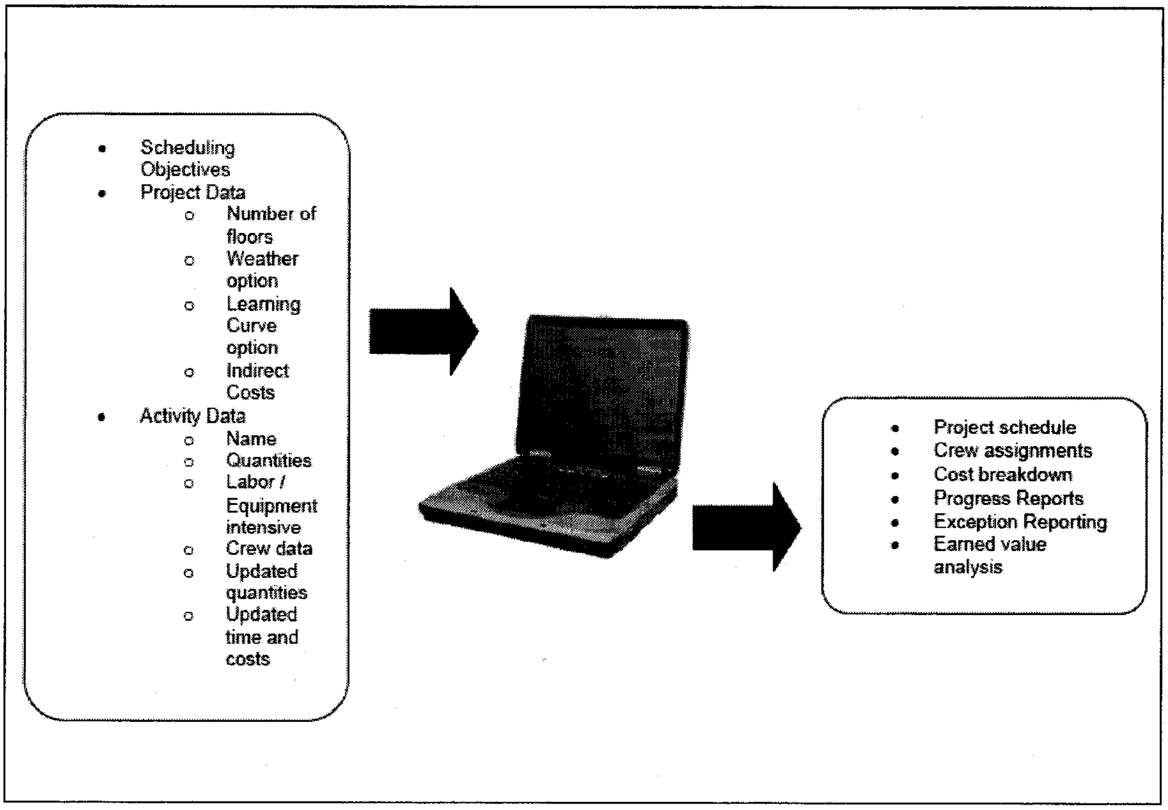


Figure 4.1: Input and output of developed software

4.2. Model

The proposed planning, scheduling, and planning and control model consists of 11 classes. These classes have been coded using the C++ programming language using two types of files, header (.h) and code (.cpp). Small programs can typically be written into one .cpp file but as the program gets larger, it is a common computer engineering practice to use .h files. Header (.h) files are text files where prototype function declarations are made. Code (.cpp) files contain class definitions. The implementation of the proposed model required the use of both header and code files.

4.3. Graphical User Interface (GUI)

High-Rise Planner Scheduler's graphical user interface was designed using Microsoft Visual C++ Version 6.0 and Microsoft MFC library in a manner to facilitate interactive project and activity data entry and minimize repetitive data input. The main screen is shown in Figure 4.2. As a precaution, menu and toolbar items are only enabled when and if it is possible.

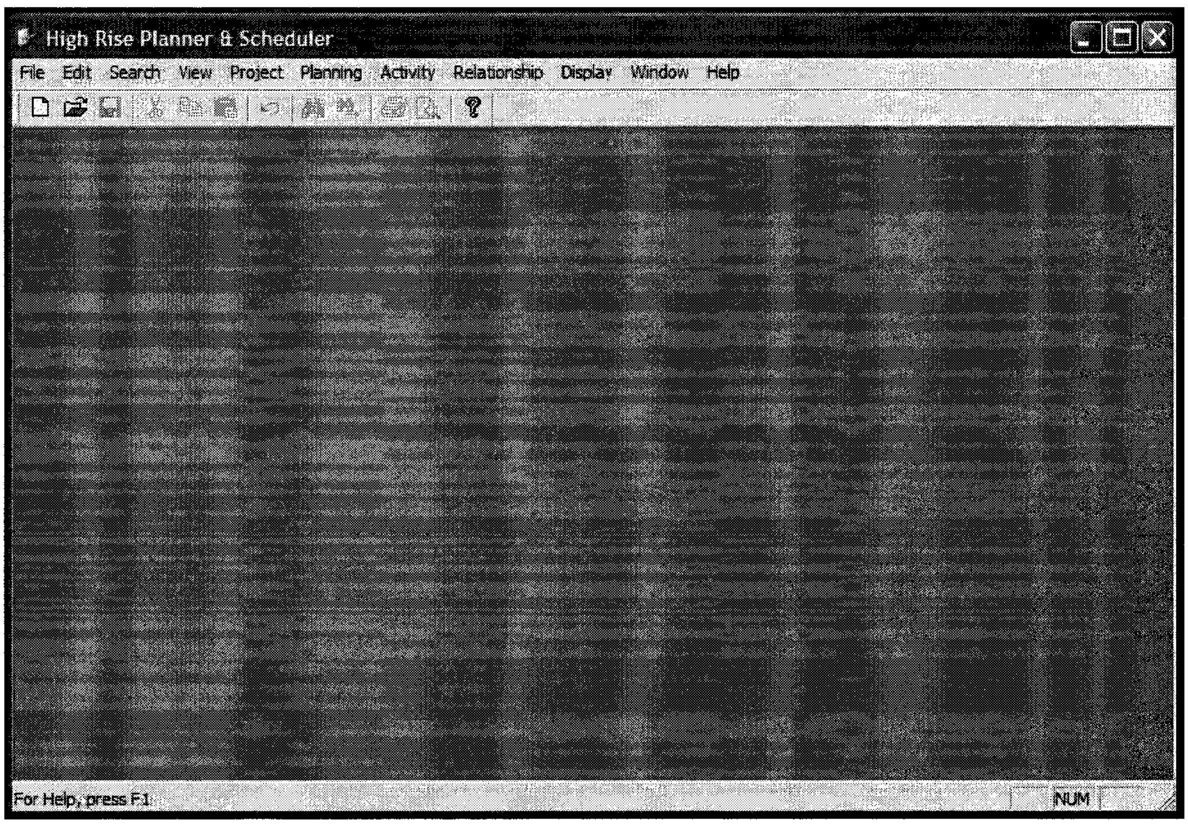


Figure 4.2: Main Screen of "HRPS"

4.3.1. Menus, Toolbar and Status bar

HRPS has a menu at the top of the screen from which the end user can select the specific function necessary, as shown in Figure 4.3. The menu bar consists of eleven menus: File, Edit, Search, View, Project, Planning, Activity, Relationship, Display, Window and Help. The File, Search, Edit, Window and Help menu perform standard Windows functions, which can be found in most applications. The rest of the menu items perform specific functions as shown in Figures 4.3 to 4.8. Tables 4-1 to 4-4 summarize these menu items, their associated functions and dialog boxes.

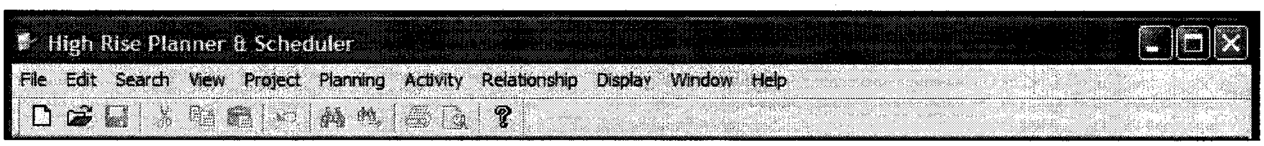


Figure 4.3: HRPS Main Menu Bar

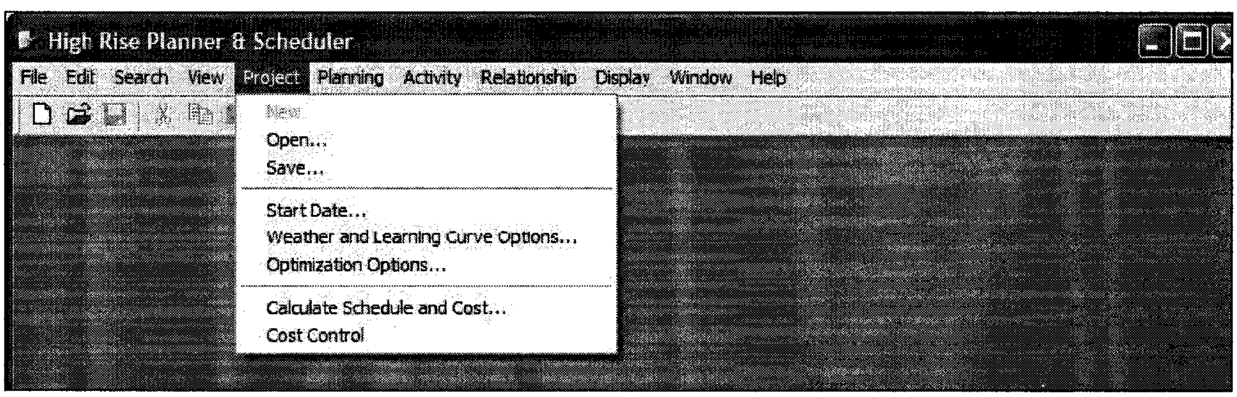


Figure 4.4: Project Menu

Table 4-1:Project Menu Functions

Menu Item	Associated Dialog Box	Function
Start Date	4.9	Accepts the project start date
Weather and Learning Curve Options	4.10	Specifies whether to accept the impact of weather and/or the learning curve in scheduling calculations
Optimization Options	4.11	Specifies whether to accept schedule, cost or their combined effect's optimization
Calculate Schedule and Cost	-	Starts scheduling and cost calculations
Cost Control	4.12	Initiates the project progress monitoring process

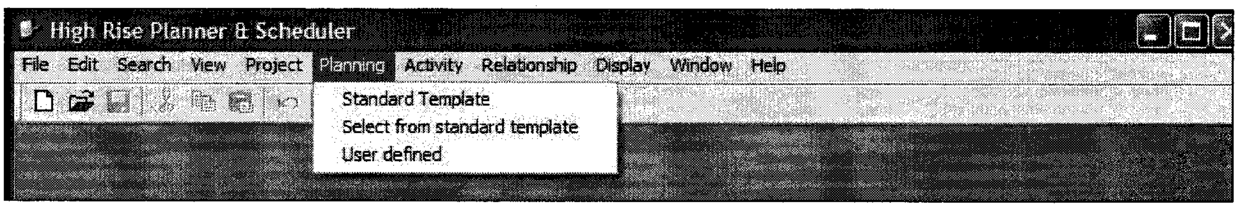


Figure 4.5: Planning Menu

Table 4-2: Planning Menu Functions

Menu Item	Associated Dialog Box	Function
Standard template	4.14	Specifies whether the user wishes to use the standard template prepared from the case study and the literature
Select from standard template	4.15	Specifies whether the user wishes to use some but not all of the activities within the standard template.
User defined	4.16 - 4.22	The user will have to manually input the activities, their type, the relationships among them.

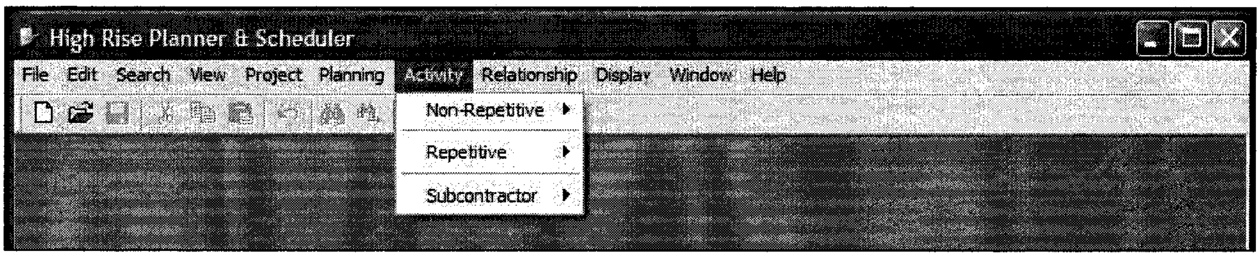


Figure 4.6: Activity Menu

Table 4-3: Activity Menu Functions

Menu Item	Associated Dialog Box	Function
Non-Repetitive	4.16	Data regarding non-repetitive activities is entered here
Repetitive	4.17 - 4.18	Data regarding repetitive activities is entered here
Subcontractor	4.19	Data regarding subcontractors is entered here

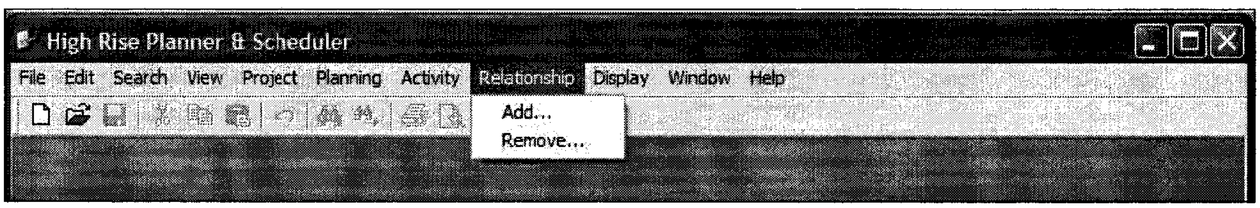


Figure 4.7: Relationship Menu

Table 4-4: Relationship Menu Functions

Menu Item	Associated Dialog Box	Function
Add	4.22	Defines relationships between activities
Remove	-	Removes relationships between activities

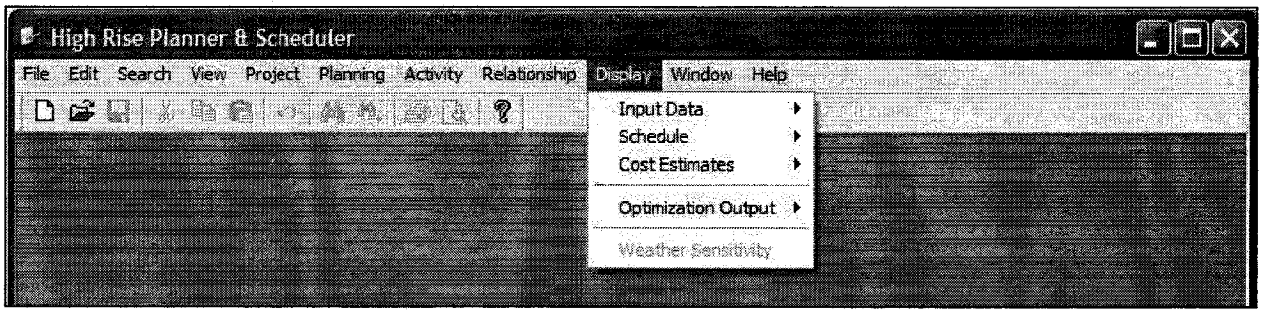


Figure 4.8: Display Menu

4.3.2. Dialog Boxes

HRPS uses a number of dialog boxes to facilitate user-input. Several of the significant dialog boxes are shown in Figures 4.9 to 4.22.

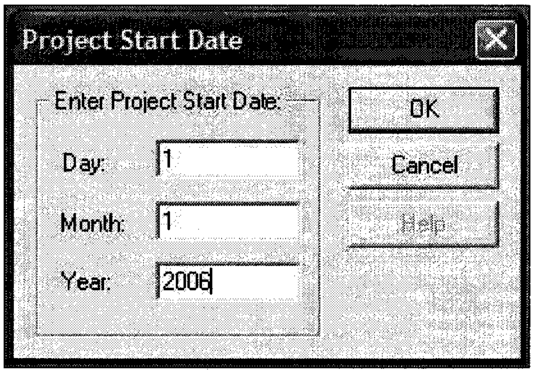


Figure 4.9: Project Start Date

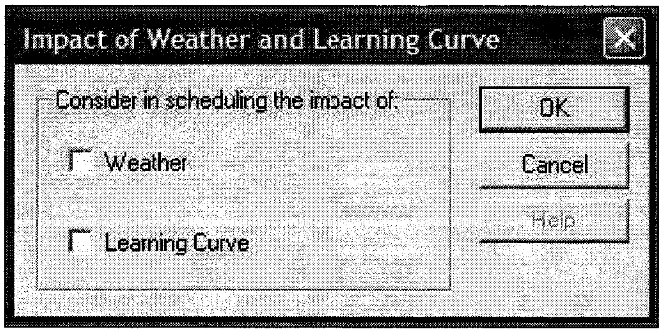


Figure 4.10: Weather and Learning Curve Options

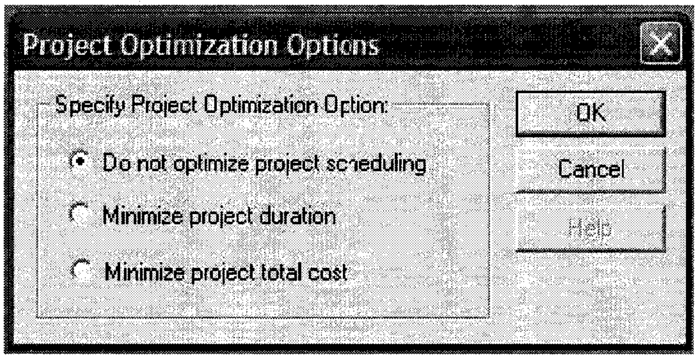


Figure 4.11: Project Schedule and Cost Optimization Options

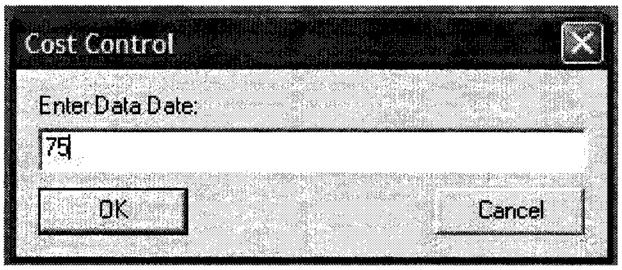


Figure 4.12: Data Date Entry for Project Control and Schedule Updating

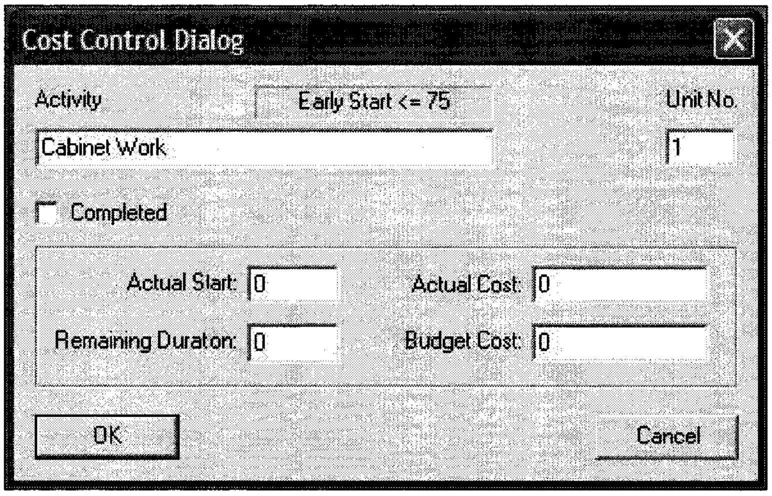


Figure 4.13: Integrated Time and Cost Control Dialog

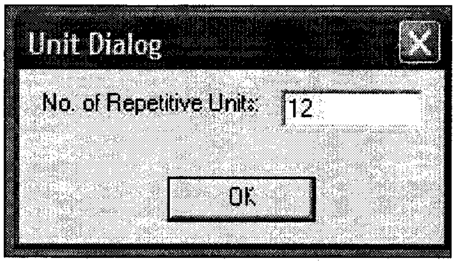


Figure 4.14: Standard Planning Template Data Entry

Figure 4.15: Selective Activities Planning Template Data Entry

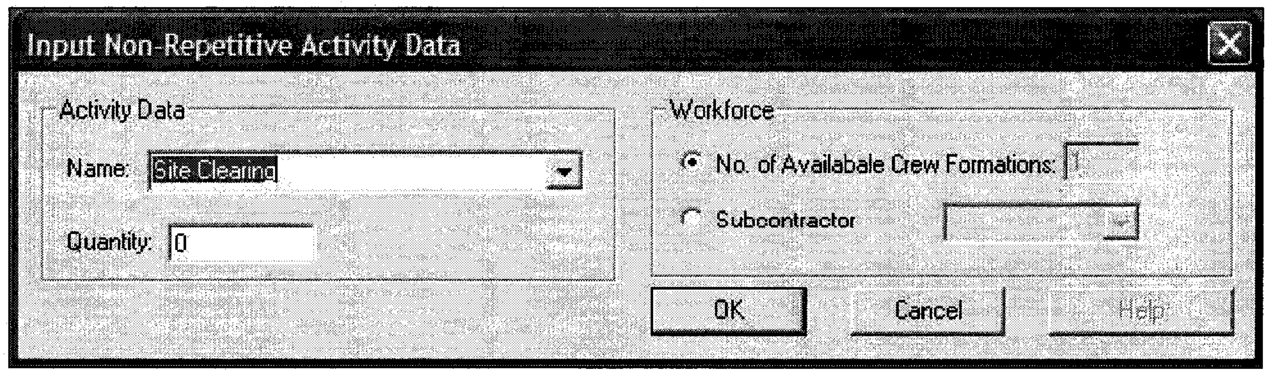


Figure 4.16: Non-Repetitive Activity Data

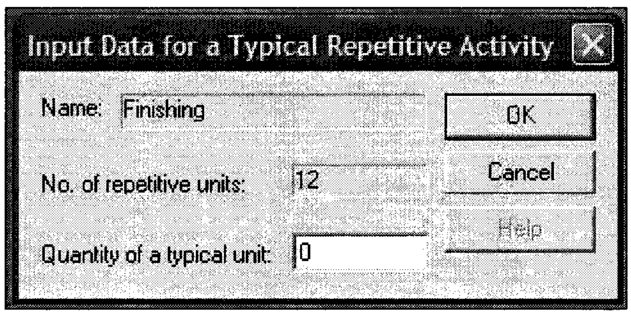


Figure 4.17: Typical Repetitive Activity Data Entry

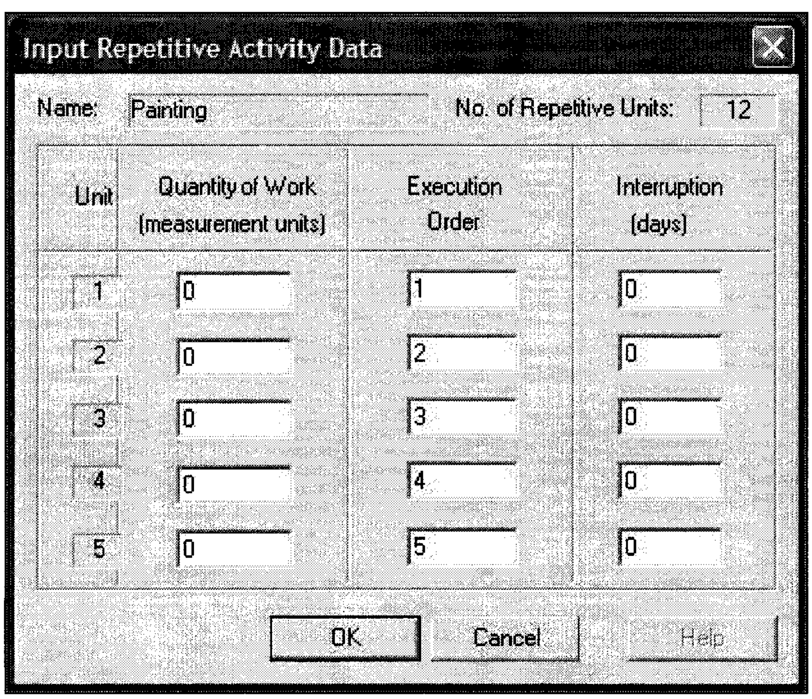


Figure 4.18: Non-Typical Repetitive Activity Data Entry

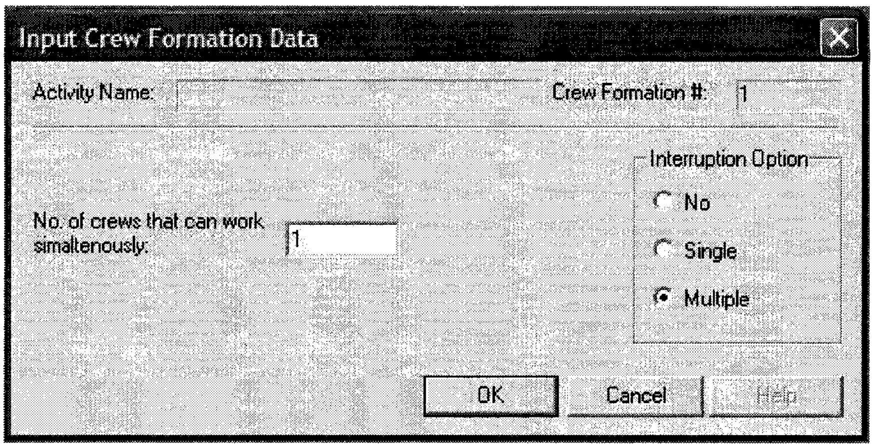


Figure 4.19: Number of Crew(s) Data Entry

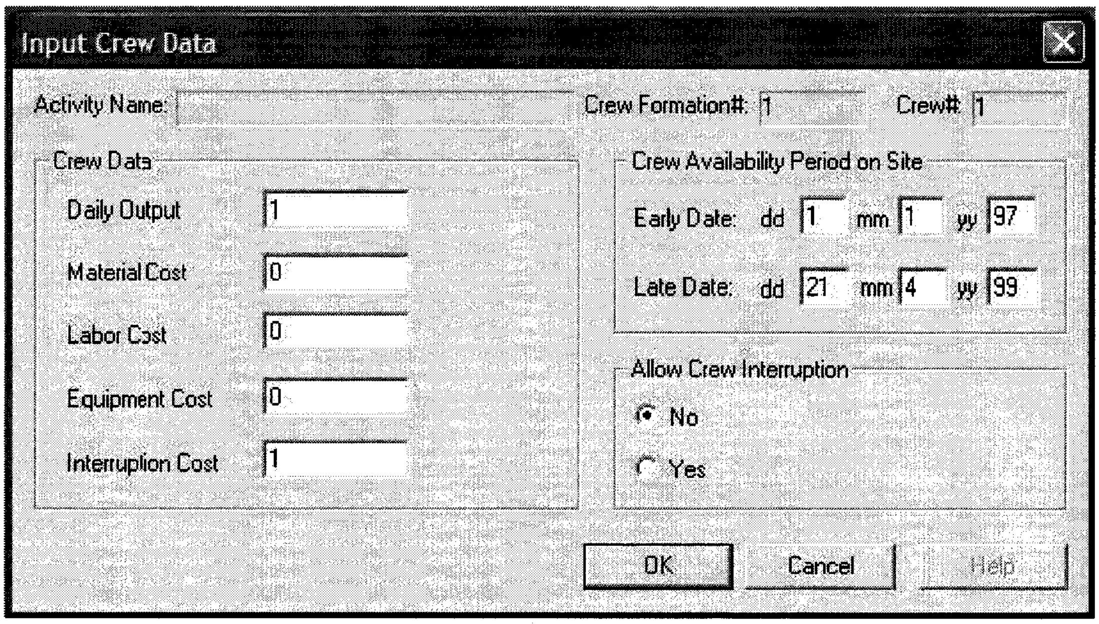


Figure 4.20: Crew Data Entry

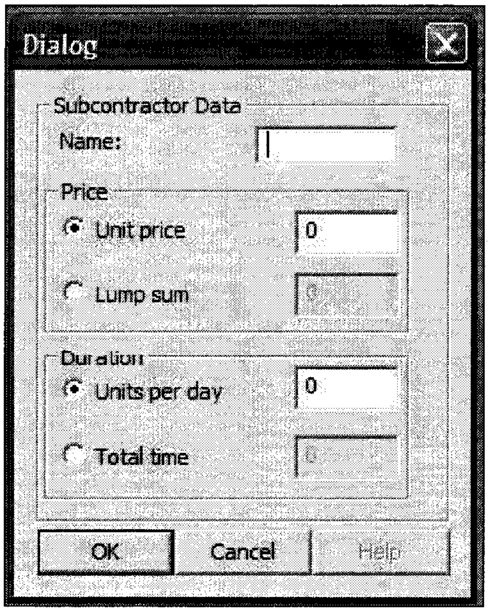


Figure 4.21: Subcontractor Data Entry

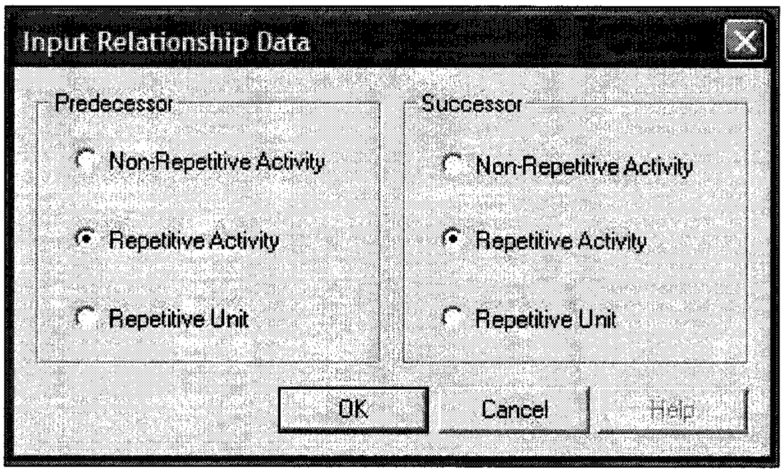


Figure 4.22: Relationship Definition Data Entry

4.4. Input and Output

HRPS has three main functions: 1) providing assistance in planning; 2) scheduling; and 3) integrated time and cost control of high-rise building construction. In order for HRPS to provide useful output, specific and correct input has to be fed into it. For HRPS to be able to provide assistance in planning, the end user either has the option of using the standard template derived from the case study and the literature, or using some of the activities in the template or the user has the option of introducing his/her activities and defining the relationships among them. To perform scheduling, HRPS requires data input regarding the number of repetitive units, the quantities of work, crew productivity, precedence relationships, subcontractor information, the non-repetitive activities and the costs associated with performing the work, on either a unit rate basis or a lump sum price. The integrated time and cost control module requires the user to report project progress accurately in terms of actual start, actual finish, percent complete or remaining duration, Actual Cost of Work Performed (ACWP) and Budgeted Cost of Work Performed (BCWP). HRPS will then be able to provide a schedule and perform Earned Value analysis at different levels to assist the project management team in monitoring the project progress efficiently. Figure 4.1 displays the input and output of HRPS.

4.5. System Validation

In order to validate the system four numerical examples are presented. The first example was initially analyzed by Selinger (1980) and later by Russell and Caselton (1988) and subsequently by Moselhi and El-Rayes (1998). This example is used to compare the results of the optimization module with the above-mentioned previously developed models. The second example is a project with five repetitive activities and is used to validate the optimization module originally introduced by El-Rayes and Moselhi (1998).

The third example expands on the second example and illustrates some of the system's capabilities beyond what El-Rayes and Moselhi (1998) had built; taking the role of subcontractors into account in the same project and generates a schedule based on the input given from the subcontractors. The model developed by El-Rayes and Moselhi (1998) was not capable of considering the role of subcontractors, and only utilized the prime contractor's own work force. It used crew productivity, crew availability and the cost of resources to schedule the activities and calculate the cost.

The developed system is capable of doing the same while considering the role of subcontractors and any combination of time and cost in the following contract types: 1) lump sum; 2) unit price; 3) total time; and 4) unit time. Numerical Example IV demonstrates the tracking and control abilities of the developed system and generates reports based on actual progress reports. The tables presented here are the results generated from the model and are put into Microsoft Excel for better presentation. The actual output of the software can be seen in Appendix I.

4.5.1. Numerical Example I

In order to validate the model a concrete bridge with repetitive activities is analyzed. This example project is also used to demonstrate the use of the present optimization model and illustrate its capabilities. The project has four repetitive units which includes the following repetitive activities: excavation, foundations, columns, beams, and slabs. Each repetitive activity is carried out by a single crew, which sequentially progresses from the first to the last unit. The relationships among succeeding activities is finish to start with no lag time. Table 4-5 shows the corresponding activity for each crew in every unit.

Table 4-5: Activity Durations for Numerical Example I

Act.	Excavation	Foundation			Columns			Beams				Slabs	
Unit/Crew	1	1	2	3	1	2	3	1	2	3	4	1	2
1	12.5	11.5	14.4	19.2	18.1	15.1	12.9	8.6	10.0	12.0	15.0	-	-
2	15.6	12.0	15.0	20.0	15.0	12.5	10.7	9.3	10.8	13.0	16.2	15.8	17.8
3	10.8	10.5	13.1	17.5	22.5	18.7	16.1	10.2	11.9	14.2	17.8	13.0	14.7
4	16.7	10.0	12.5	16.7	17.5	14.6	12.5	8.0	9.4	11.2	14.1	16.7	18.7

Table 4-6 displays the results of Selinger (1980), Table 4-7 the results generated by Russell and Caselton (1988) and Table 4-8 by HRPS. The early start, early finish dates and the number of interruptions are shown in these tables.

Table 4-6: Optimized Schedule Generated by Selinger (1980)

Unit	Excavation			Foundation			Columns			Beams			Slabs		
	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter
1	0.0	12.5	-	13.7	28.1	-	32.6	45.5	-	47.3	59.3	-	-	-	-
2	12.5	28.1	-	28.1	43.1	-	45.5	56.2	-	59.3	72.3	-	72.3	88.1	-
3	28.1	39.0	-	43.1	56.2	-	56.2	72.3	-	72.3	86.6	-	88.1	101.2	-
4	39.0	55.6	-	56.2	68.7	-	72.3	84.8	-	86.6	97.9		101.2	117.9	-
Selected Crew	1			2			3			3			1		

Table 4-7: Optimized Schedule Generated by Russell and Caselton (1988)

Unit	Excavation			Foundation			Columns			Beams			Slabs		
	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter
1	0.0	12.5	-	17.6	29.1	-	29.1	42.1	-	43.0	51.6	-	-	-	-
2	12.5	28.1	-	29.1	41.1	-	42.1	52.8	-	55.6	64.9	4.0	64.9	80.7	-
3	28.1	38.9	-	45.1	55.6	4.0	52.8	68.9	-	68.9	79.0	4.0	80.7	93.7	-
4	38.9	55.6	-	55.6	65.6	-	68.9	81.4	-	83.0	91.0	4.0	93.7	110.4	-
Selected Crew	1			1			3			1			1		

Table 4-8: Optimized Schedule Generated by HRPS

Unit	Excavation			Foundation			Columns			Beams			Slabs		
	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter	ES	EF	Inter
1	0.0	12.5	-	15.6	27.1	-	27.2	40.1	-	43.0	51.6	-	-	-	-
2	12.5	28.1	-	28.1	40.1	1.0	40.1	50.8	-	51.6	60.9	-	61.3	77.1	-
3	28.1	38.9	-	40.1	50.6	-	50.8	66.9	-	66.9	77.1	6.0	77.1	90.1	-
4	38.9	55.6	-	55.6	65.6	5.0	66.9	79.4	-	80.1	88.1	3.0	90.1	106.8	-
Selected Crew	1			1			3			1			1		

As it can been seen from Tables 4-6, 4-7 and 4-8, the optimized schedule generated by Selinger (1980) has a total project duration of 117.9 and no interruption days. The schedule optimization by Russell and Caselton (1988) resulted in a total project duration of 110.4 days with 16 interruption days. The results generated by HRPS match the results of Moselhi and El-Rayes (2001) with a project duration of 106.8 days which includes 15 interruption days.

4.5.2. Numerical Example II

A numerical example of a highway is analyzed to demonstrate the optimization features of the developed system. Highway projects are of repetitive nature and since this example examines the optimization feature of the system which is related to the scheduling module and there is no planning involved. It is used to test the functionality and validate this feature of the system; making use of the results reported by El-Rayes and Moselhi (1998).

Five serial activities are considered in the construction of 15 kilometers of a highway. The project has been divided into 15 units, each is a kilometer long. The activities, in order of precedence are: 1) cut and chip trees; 2) grub and remove stumps; 3) earthmoving; 4) base; and 5) paving. The activities all have finish to start relationships with no lag. The quantities associated with the activities are shown in Table 4-9. Crew data is summarized and displayed in Table 4-10.

Table 4-9: Quantities of Work for Numerical Example II

Cut and chip trees			Grub & remove stumps			Earthmoving		
Unit	Quantity (m2)	Order	Unit	Quantity (m2)	Order	Unit	Quantity (m3)	Order
1	12000	1	1	12000	1	1	6000	4
2	12000	2	2	12000	2	2	6000	3
3	18000	3	3	18000	3	3	6000	2
4	12000	4	4	12000	4	4	7000	1
5	18000	5	5	18000	5	5	8600	5
6	30000	6	6	30000	6	6	7000	6
7	36000	7	7	36000	7	7	6500	7
8	30000	8	8	30000	8	8	6000	8
9	24000	9	9	24000	9	9	6000	9
10	24000	10	10	24000	10	10	6000	10
11	18000	11	11	18000	11	11	6000	11
12	12000	12	12	12000	12	12	6000	12
13	12000	13	13	12000	13	13	6000	13
14	12000	14	14	12000	14	14	6000	14
15	12000	15	15	12000	15	15	6000	15

Base			Pavement		
Unit	Quantity (m2)	Order	Unit	Quantity (m2)	Order
1	32000	1	1	32000	1
2	32000	2	2	32000	2
3	32000	3	3	32000	3
4	32000	4	4	32000	4
5	32000	5	5	32000	5
6	32000	6	6	32000	6
7	32000	7	7	32000	7
8	32000	8	8	32000	8
9	32000	9	9	32000	9
10	32000	10	10	32000	10
11	32000	11	11	32000	11
12	32000	12	12	32000	12
13	32000	13	13	32000	13
14	32000	14	14	32000	14
15	32000	15	15	32000	15

Table 4-10: Crew Data for Example II

Activity	Crew No.	Daily Output (units per day)	Earliest available day	Latest available day
Cut and chip trees	1	3000	0	-
	2	3000	0	-
	3	3000	0	18
	4	3000	24	40
Grub and remove stumps	1	4000	0	-
	2	4000	0	-
Earthmoving	1	1200	10	70
	2	800	10	70
Base	1	3200	26	-
	2	3200	26	-
	3	3200	26	-
	4	3200	26	-
Paving	1	4000	0	-
	2	4000	0	-
	3	4000	0	-

The generated schedule is displayed in Table 4-11

Table 4-11: Generated Schedule for Example II

Cut and chip trees				Grub & remove stumps				Earthmoving			
Unit	ES	EF	Crew	Unit	ES	EF	Crew	Unit	ES	EF	Crew
1	0	4	1	1	8	11	1	1	24	32	2
2	0	4	2	2	9	12	2	2	21	26	1
3	0	6	3	3	11	16	1	3	16	24	2
4	4	8	1	4	12	15	2	4	15	21	1
5	4	10	2	5	15	20	2	5	26	34	1
6	6	16	3	6	16	24	1	6	32	41	2
7	8	20	1	7	20	29	2	7	34	40	1
8	10	20	2	8	24	32	1	8	40	45	1
9	20	28	1	9	29	35	2	9	41	49	2
10	20	28	2	10	32	38	1	10	45	50	1
11	24	30	4	11	35	40	2	11	49	57	2
12	28	32	1	12	38	41	1	12	50	55	1
13	28	32	2	13	40	43	2	13	55	60	1
14	30	34	4	14	41	44	1	14	57	65	2
15	32	36	1	15	43	46	2	15	60	65	1

Base				Paving			
Unit	ES	EF	Crew	Unit	ES	EF	Crew
1	32	42	1	1	42	50	1
2	35	45	2	2	45	53	2
3	37	47	3	3	47	55	3
4	35	45	4	4	50	58	1
5	42	52	1	5	53	61	2
6	45	55	2	6	55	63	3
7	47	57	3	7	58	66	1
8	45	55	4	8	61	69	2
9	52	62	1	9	63	71	3
10	55	65	2	10	66	74	1
11	57	67	3	11	69	77	2
12	55	65	4	12	71	79	3
13	62	72	1	13	74	82	1
14	65	75	2	14	77	85	2
15	65	75	4	15	79	87	3

The project was then optimized with time as a priority and the results are displayed in Table 4-12.

Table 4-12: Optimized Schedule Generated for Example II

Cut and chip trees				Grub & remove stumps				Earthmoving			
Unit	ES	EF	Crew	Unit	ES	EF	Crew	Unit	ES	EF	Crew
1	0	4	1	1	4	7	1	1	20	28	2
2	0	4	2	2	5	8	2	2	17	22	1
3	0	6	3	3	7	12	1	3	12	20	2
4	4	8	1	4	8	11	2	4	11	17	1
5	4	10	2	5	11	16	2	5	22	30	1
6	6	16	3	6	16	24	1	6	28	37	2
7	8	20	1	7	20	29	2	7	30	36	1
8	10	20	2	8	24	32	1	8	36	41	1
9	20	28	1	9	29	35	2	9	37	45	2
10	20	28	2	10	32	38	1	10	41	46	1
11	24	30	4	11	35	40	2	11	45	53	2
12	28	32	1	12	38	41	1	12	46	51	1
13	28	32	2	13	40	43	2	13	51	56	1
14	30	34	4	14	41	44	1	14	53	61	2
15	32	36	1	15	43	46	2	15	56	61	1

Base				Paving			
Unit	ES	EF	Crew	Unit	ES	EF	Crew
1	28	38	1	1	38	46	1
2	31	41	2	2	41	49	2
3	33	43	3	3	43	51	3
4	31	41	4	4	46	54	1
5	38	48	1	5	49	57	2
6	41	51	2	6	51	59	3
7	43	53	3	7	54	62	1
8	41	51	4	8	57	65	2
9	48	58	1	9	59	67	3
10	51	61	2	10	62	70	1
11	53	63	3	11	65	73	2
12	51	61	4	12	67	75	3
13	58	68	1	13	70	78	1
14	61	71	2	14	73	81	2
15	61	71	4	15	75	83	3

The results displayed in Table 4-8 closely match the results generated by El-Rayes and Moselhi (1998).

4.5.3. Numerical Example III

In order to demonstrate the capabilities of the model beyond those of the model developed by El-Rayes and Moselhi (1998), another example is considered. This example builds on the second example and introduces changes so as to generate a suitable scenario. Two subcontractors are considered to perform two of the activities of Example I. In this example the first subcontractor, SUB A, shall cut and chip the trees with a daily production of 12,000 units per day. The second subcontractor, SUB B, will perform the Base activity. SUB B claims it needs 30 days to perform the base activity in all the 15 units of the project. Neither SUB A nor SUB B has provided information on their crew formations or their construction methods. The generated schedule reflects the role of the subcontractors and the general contractor's own work force based on the input data. Tables 4-13 and 4-14 show the crew data and the data provided by the subcontractors, respectively.

Table 4-13: Crew Data for Numerical Example III

Activity	Crew No.	Daily Output (units per day)
Grub and remove stumps	1	4000
	2	4000
Earthmoving	1	1200
	2	800
Paving	1	4000
	2	4000
	3	4000

Table 4-14: Subcontractor Data for Numerical Example III

Activity	Units per day	Total days
Cut and chip trees SUB A	12000	-
Base SUB B	-	30

Table 4-15 displays the generated schedule.

Table 4-15: Generated Schedule for Numerical Example III

Cut and chip trees				Grub & remove stumps				Earthmoving			
Unit	ES	EF	Sub	Unit	ES	EF	Crew	Unit	ES	EF	Crew
1	0	1	A	1	1	4	1	1	4	9	1
2	1	2	A	2	2	5	2	2	5.25	12.75	2
3	2	3.5	A	3	4	8.5	1	3	9	14	1
4	3.5	4.5	A	4	5	8	2	4	12.75	21.5	2
5	4.5	6	A	5	8	12.5	2	5	14	21.17	1
6	6	8.5	A	6	8.5	16	1	6	21.17	27	1
7	8.5	11.5	A	7	12.5	21.5	2	7	21.5	29.63	2
8	11.5	14	A	8	16	23.5	1	8	27	32	1
9	14	16	A	9	21.5	27.5	2	9	29.63	37.13	2
10	16	18	A	10	23.5	29.5	1	10	32	37	1
11	18	19.5	A	11	27.5	32	2	11	37	42	1
12	19.5	20.5	A	12	29.5	32.5	1	12	37.13	44.63	2
13	20.5	21.5	A	13	32	35	2	13	42	47	1
14	21.5	22.5	A	14	32.5	35.5	1	14	44.63	52.13	2
15	22.5	23.5	A	15	35	38	2	15	47	52	1

Base				Paving			
Unit	ES	EF	Sub	Unit	ES	EF	Crew
1	26.13	28.13	B	1	28.13	36.13	1
2	28.13	30.13	B	2	30.13	38.13	2
3	30.13	32.13	B	3	32.13	40.13	3
4	32.13	34.13	B	4	36.13	44.13	1
5	34.13	36.13	B	5	38.13	46.13	2
6	36.13	38.13	B	6	40.13	48.13	3
7	38.13	40.13	B	7	44.13	52.13	1
8	40.13	42.13	B	8	46.13	54.13	2
9	42.13	44.13	B	9	48.13	56.13	3
10	44.13	46.13	B	10	52.13	60.13	1
11	46.13	48.13	B	11	54.13	62.13	2
12	48.13	50.13	B	12	56.13	64.13	3
13	50.13	52.13	B	13	60.13	68.13	1
14	52.13	54.13	B	14	62.13	70.13	2
15	54.13	56.13	B	15	64.13	72.13	3

4.5.4. Numerical Example IV

This example displays the tracking and control capabilities of the developed model. A project with five repetitive units is first scheduled and later progress data is input into the system and control reports are generated. Table 4-16 shows the quantities associated with the activities. Table 4-17 displays the crew data for the activities and Table 4-18 displays the information for the subcontractor present on the project, which has a lump sum contract.

Table 4-16: Quantities of Work for Numerical Example IV

Act A		Act B		Act C		Act D		Act E	
Unit	Quantity	Unit	Quantity	Unit	Quantity	Unit	Quantity	Unit	Quantity
1	20000	1	15000	1	6000	1	24000	1	14000
2	25000	2	20000	2	9000	2	16000	2	21000
3	30000	3	15000	3	12000	3	28000	3	42000
4	25000	4	30000	4	18000	4	12000	4	35000
5	20000	5	25000	5	27000	5	32000	5	14000

Table 4-17: Crew Data for Numerical Example IV

Activity	Crew No.	Daily Output (units per day)	Material Cost per unit ($)	Labor Cost per day ($)	Equipment Cost per day ($)
Act A	1	5000	5	500	1500
	2	5000	5	500	1500
Act B	1	5000	10	1000	500
	2	5000	10	1000	500
Act C	1	3000	50	1500	1000
Act E	1	7000	10	500	2000

Table 4-18: Subcontractor Data for Numerical Example IV

Activity	Time	Cost ($)
Act D	28 days	5600000

The schedule generated from the above information is shown in Table 4-19.

Table 4-19: Generated Schedule for Example IV

Act A				Act B				Act C			
Unit	ES	EF	Crew	Unit	ES	EF	Crew	Unit	ES	EF	Crew
1	1	4	1	1	9	11	1	1	12	13	1
2	1	5	2	2	7	10	2	2	14	16	1
3	5	10	1	3	12	14	1	3	17	20	1
4	6	10	2	4	11	16	2	4	21	26	1
5	11	14	1	5	15	19	1	5	27	35	1

Act D				Act E			
Unit	ES	EF	SUB	Unit	ES	EF	Crew
1	16	21	D	1	28	29	1
2	22	25	D	2	30	32	1
3	26	32	D	3	33	38	1
4	33	35	D	4	39	43	1
5	36	43	D	5	44	45	1

The project progress was monitored daily and at the end of the 17^{th} day the progress gathered from the field is shown in Table 4-16. The Actual Cost of Work Performed (ACWP) and Budgeted Cost of Work Performed (BCWP) or the Earned Value is also included in this report, therefore it is possible to perform integrated time and cost control.

Some of the activities were incomplete at the data date, the end of the 17^{th} day, therefore in order to estimate the remaining duration, the reported manhours were subtracted from the estimated manhours in the baseline schedule. Other methods which could be used are: 1) templates; and 2) physical progress measurements. In some cases the required manhours may be complete but the actual progress is not, concrete curing is a good example. In other cases the physical progress may not reflect the actual percentage of completion, since many manhours may be required to do the finishing work.

The data gathered in this example enables the model to perform integrated time and cost control. Time and cost only are not enough to measure the progress of a project. In some cases the project may be behind budget and ahead of schedule, or behind schedule but is more cost effective; in cases like these cost control or schedule control alone can not reflect the true reality of the project. It can be concluded that integrated time and cost control is an effective form of reporting which assists in evaluating project progress. The model will perform the Earned Value analysis and determine the performance of the project based on the input data which can be seen below in Table 4-20.

Table 4-20: Cumulative Data on the 17th Day

Activity	AS	AF	Remaining Duration	ACWP	BCWP	BCWS
ACT A1	1	4	-	108000	108000	108000
ACT A2	1	5	-	135000	135000	135000
ACT A3	5	9	-	157500	162000	162000
ACT A4	6	10	-	135000	135000	135000
ACT A5	10	12	-	104500	108000	108000
ACT B1	9	12	-	156000	155000	154500
ACT B2	7	11	-	207500	206500	206000
ACT B3	13	15	-	154500	154500	154500
ACT B4	12	16	-	307500	310000	309000
ACT B5	16	-	3	103000	103000	154500
ACT C1	13	14	-	305000	305000	305000
ACT C2	15	16	-	455000	460000	457500
ACT C3	17	-	3	152500	152500	152500
ACT C4	-	-	6	-	-	915000
ACT C5	-	-	9	-	-	1372500
ACT D1	15	-	3	600000	600000	400000
ACT D2	-	-	4	-	-	600000
ACT D3	-	-	7	-	-	1400000
ACT D4	-	-	3	-	-	600000
ACT D5	-	-	8	-	-	160000
ACT E1	-	-	2	-	-	145000
ACT E2	-	-	3	-	-	217500
ACT E3	-	-	6	-	-	435000
ACT E4	-	-	5	-	-	362500
ACT E5	-	-	2	-	-	145000

Two different types of reports were generated for this example. The first looks at the project as a whole, and the second looks at individual activities. Tables 4-21 and 4-22 display these results, respectively.

Table 4-21: Earned Value Analysis at the Project Level

ACWP	3081000
BCWP	3094500
BCWS	2941500
SV	153000
CV	-13500
AV	139500
CPI	1.0044
SPI	1.052

As it can be seen, the project has a negative Cost Variance (CV), which indicates that the project doing well financially. The positive Schedule Variance (SV) shows that the project is ahead of schedule. This could also be concluded from the fact that Cost Performance Index and Schedule Performance Index, are each, greater than one.

The report generated for the individual activities is shown in Table 4-22.

Table 4-22: Earned Value Analysis at the Activity Level

Activity	Unit	ACWP	BCWP	BCWS	SV	CV	AV	CPI	SPI
Act A	1	108000	108000	108000	0	0	0	1	1
Act A	2	135000	135000	135000	0	0	0	1	1
Act A	3	157500	162000	162000	0	-4500	-4500	1.02857	1
Act A	4	135000	135000	135000	0	0	0	1	1
Act A	5	104500	108000	108000	0	-3500	-3500	1.03349	1
Act B	1	156000	155000	154500	500	1000	1500	0.99359	1.00324
Act B	2	207500	206500	206000	500	1000	1500	0.995181	1.00243
Act B	3	154500	154500	154500	0	0	0	1	1
Act B	4	307500	310000	309000	1000	-2500	-1500	1.00813	1.00324
Act B	5	103000	103000	154500	-51500	0	-51500	1	0.666667
Act C	1	305000	305000	305000	0	0	0	1	1
Act C	2	455000	460000	457500	2500	-5000	-2500	1.01099	1.00546
Act C	3	152500	152500	152500	0	0	0	1	1
Act C	4	0	0	0	0	0	0	0	0
Act C	5	0	0	0	0	0	0	0	0
Act D	1	600000	600000	400000	200000	0	200000	1	1.5
Act D	2	0	0	0	0	0	0	0	0
Act D	3	0	0	0	0	0	0	0	0
Act D	4	0	0	0	0	0	0	0	0
Act D	5	0	0	0	0	0	0	0	0
Act E	1	0	0	0	0	0	0	0	0
Act E	2	0	0	0	0	0	0	0	0
Act E	3	0	0	0	0	0	0	0	0
Act E	4	0	0	0	0	0	0	0	0
Act E	5	0	0	0	0	0	0	0	0

4.6. Summary

This chapter presented the computer implementation stage of the developed system. Object-oriented design is employed and the system operates in Microsoft Windows XP, 2000 and ME. All of the three modules were coded using Microsoft Visual C++ 6 and Microsoft Visual C++ .NET. The user interface incorporates menus, toolbars, dialog windows and multiple document interfaces windows. Three numerical examples were presented to display the different capabilities of the system.

CHAPTER FIVE
CASE STUDY

5.1. Introduction

This chapter presents the study of the schedule generated for the construction of a high-rise institutional building located in downtown Montreal. The project is an academic building for the Faculty of Engineering and Visual Arts of Concordia University. The case study was used primarily to assist in planning and test and validate the system and its functions.

5.2. Description of the Case Study

The case study is Concordia University's newly constructed Engineering and Visual Arts Building, referred to later as the EV building, located in downtown Montreal. There are many interesting aspects worth mentioning about this building. The building was financed through bonds issued by the university. Beyond financing and other unique characteristics of this building, the description included here focuses primarily on aspects that relate to planning, scheduling and control.

The building consists of two towers, one twelve floors, for Visual Arts, and the other seventeen, for Engineering and Computer Science. The two towers of the complex are completely integrated at every floor level through links and common corridors. The two towers are of concrete construction and rest on a 3-level base.

The concrete structure utilizes flat slab construction with a slab thickness of 229 mm (9 inches). The typical grid dimensions are 9 m X 9 m. Each floor is 4.1 m in height, from slab to slab. Mechanical rooms built of structural steel are at the top of each tower. The building envelope is primarily pre-glazed curtain wall, zinc cladding and aluminum panels. The interior features all-glass guide rails, five circular staircases, each linking three floors.

The 75,000 m^2 complex integrates all research and teaching activities under a single roof. The 17 floor building houses the offices of the Dean and the entire teaching staff. It houses over 300 specialized labs, conference and meeting rooms, as well as student areas. Among the state-of-the-art research facilities installed are engineering equipment, wet and heavy laboratories, a clean room and computer rooms are notable. Figures 5.1, 5.2 and 5.3 show a typical floor plan, a section and an elevation of the building respectively.

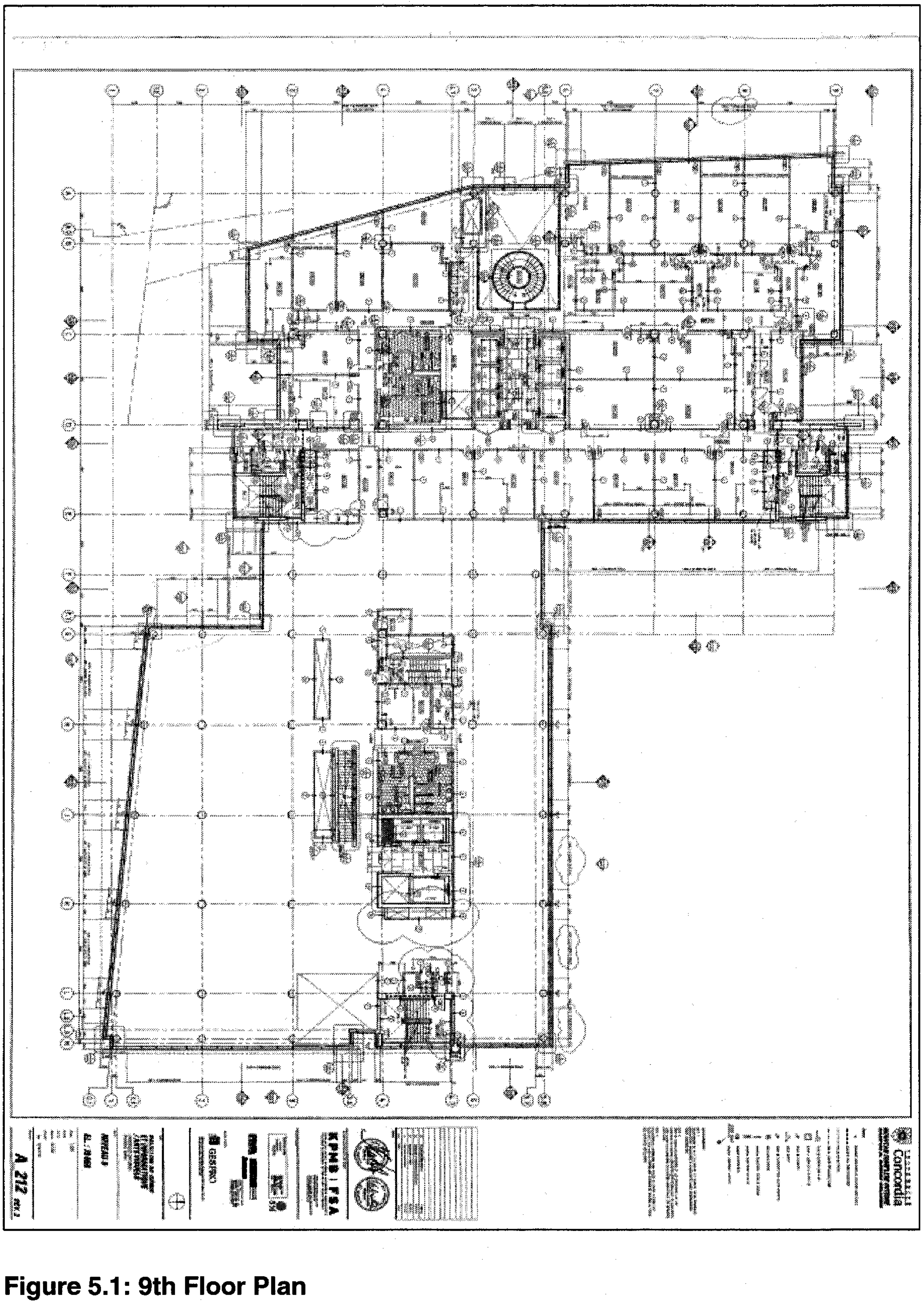


Figure 5.1: 9th Floor Plan

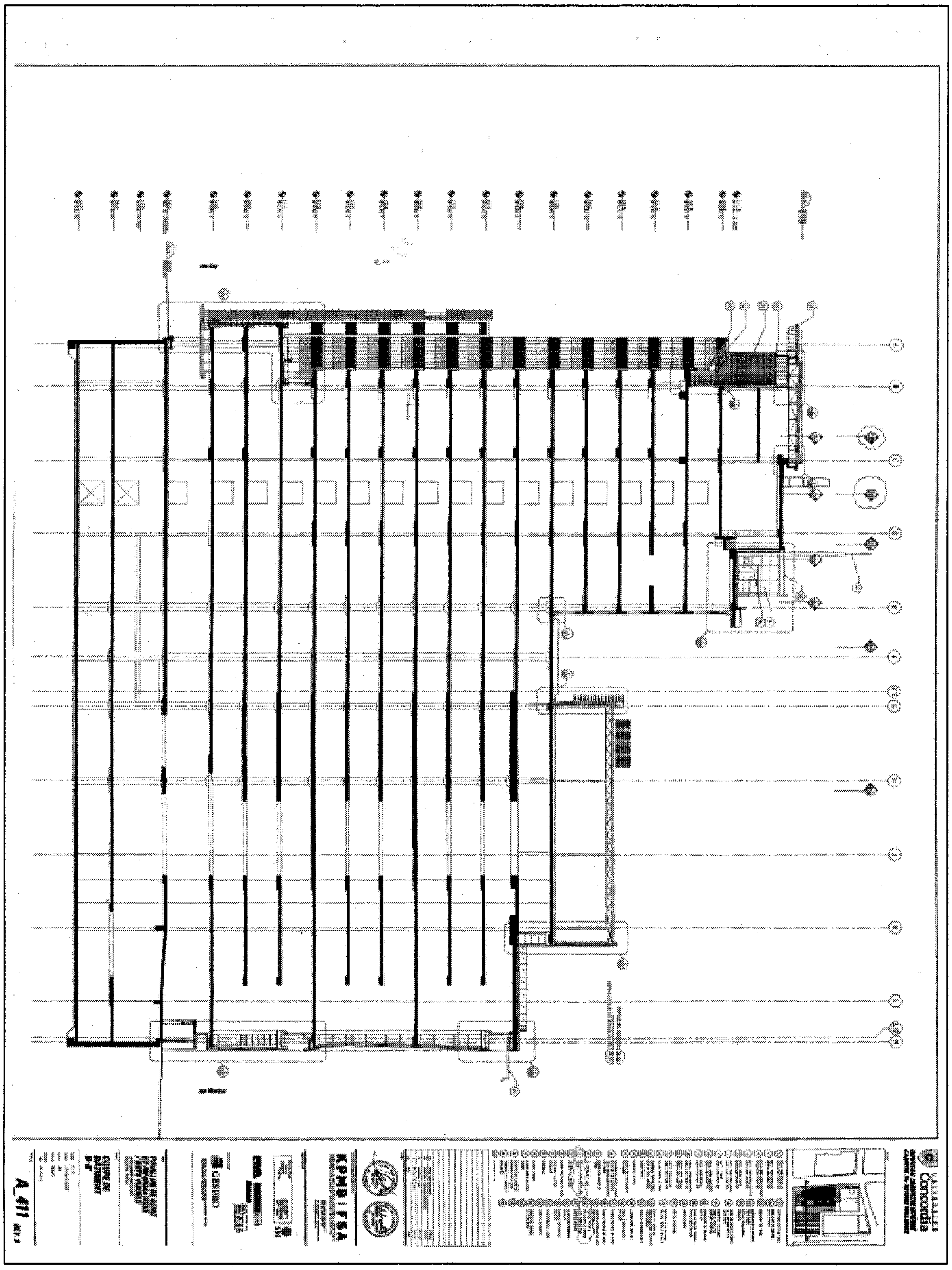


Figure 5.2: Building Section B-B'

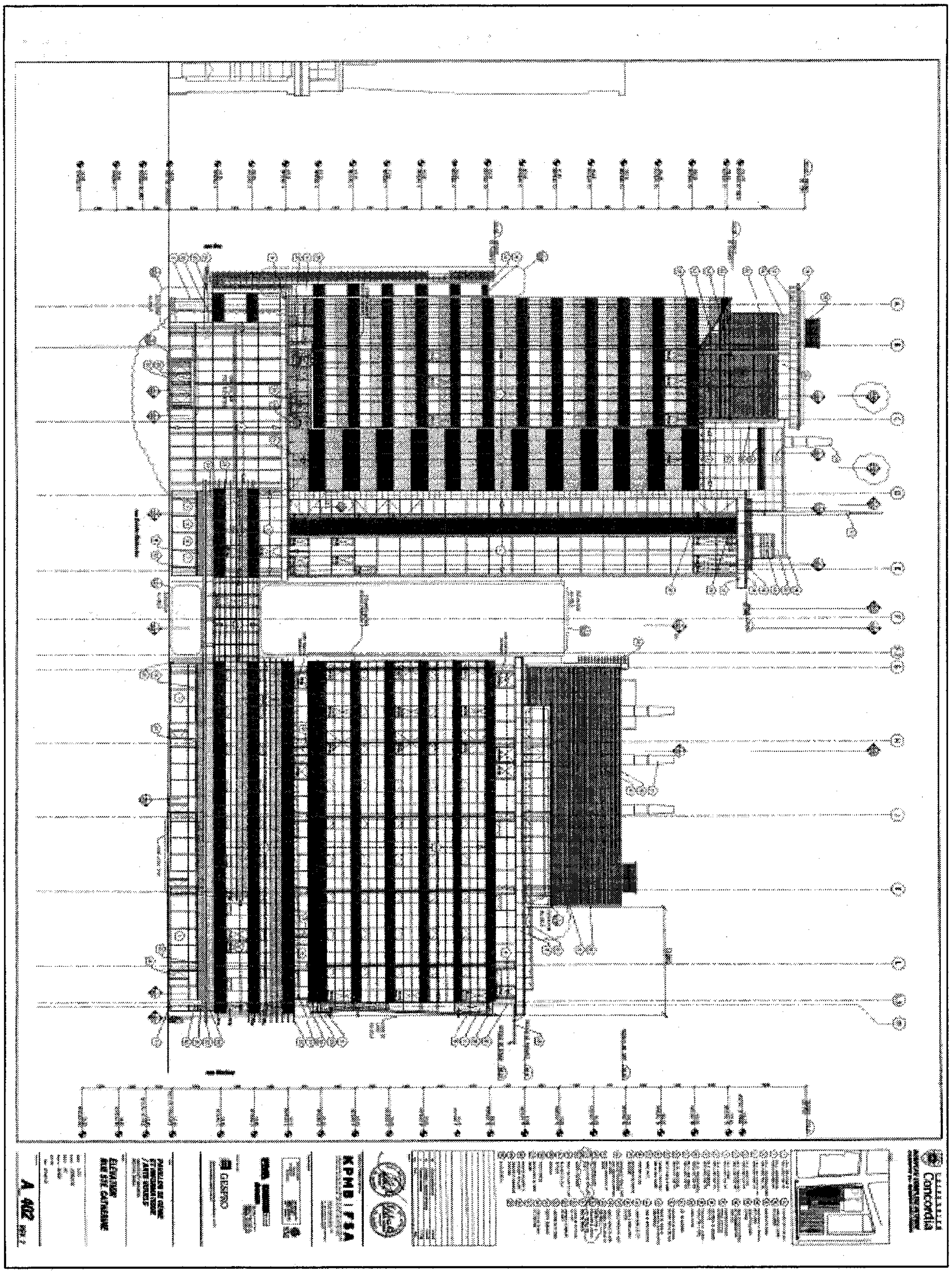


Figure 5.3: Building Elevation from Ste-Catherine Street

The project was delivered using phased construction, through three main commitment packages: 1) excavation and shoring; 2) concrete structure (concrete framing from the foundation up to the 17th floor); and 3) the balance of the complex which included HVAC and plumbing, electro-mechanical, escalators and elevators, building envelope and interior finishing.

During construction the project team was confronted with major challenges, among them: 1) access; and 2) rock excavation. Building two high-rise towers in a vibrant urban center such as downtown Montreal required a great deal of planning. Many permits had to be obtained and safety played a significantly more important role than it usually does, due to the daily pedestrian population and traffic. The foundation required rock excavation, shoring, and soldier piles, which had to be done with extra care as not to disrupt the adjacent buildings and the underground metro facilities. The soldier piles were used for the protection of excavation and were primarily tube piles with the exception of the side on Guy Street, where H cross section piles were used. Some of the excavated material contained contaminants and had to be transported to an off-site location. A number of construction progress photographs are included in Appendix II.

The concrete framing of the building is a good example of how an activity can be repetitive and non-repetitive at the same time. For the non-repetitive part of the concrete structure, traditional forms were used, but for the repetitive section, flying forms were used. The difference in construction methods leads to differences in productivity, thus requiring the activity to be split into two sections; 1) repetitive; and 2) non-repetitive.

Twenty-four major repetitive activities were identified. The inter-relationships among them were studied and analyzed. The project team used Microsoft Project to create the construction schedule. Microsoft Project utilizes deterministic network CPM-scheduling. The deficiencies for using the critical path method for high-rise construction have been stated in Chapter Two. The result of using such an approach leads to having a very large number of activities in the schedule, which makes the schedule very difficult to manage and maintain. The project management firm went through the same planning process as indicated in Chapter Three. The Microsoft Project schedule is included in the Appendix III.

5.3. Schedule Generated Using HRPS

The schedule of the case study was re-generated using the developed system, using the same quantities and crew formations. Since the actual schedule is available in bar chart format, the results were put into Primavera Project Planner 3.1 in order to compare the results using bar charts. The activities have been further sorted by the construction crews performing them to clearly display the crew work continuity constraint. The schedule can be seen in Appendix II.

In order to be able to compare the two schedules, the same start date of February 23, 2003 was selected. The schedule generated by HPRS ends on

March 17, 2005 whereas the schedule generated by Microsoft Project ends on April 26, 2005. The schedule generated by HRPS includes holidays and weather days. The same calendars were used for both schedules, work was to be done five days a week for eight hours a day.

After the completion of the initial non-repetitive activities such as excavation and foundation construction, the repetitive activities begin, concrete structure construction is the first of such nature. The durations for this activity in both schedules correspond, since it is the predecessor for all other activities. As mentioned before the same crews and production rates were used for both schedules.

The schedule generated by HRPS ensures crew work continuity therefore it can been seen that a number of activities in the schedule generated by HRPS start significantly later than they do in the schedule created using deterministic scheduling techniques in Microsoft Project. As mentioned in Chapter Three, the scheduling engine, first ensures that the logical relationships between the activities are satisfied and then goes on to enforce the crew work continuity constraint. The start of some activities may be shifted to a later date as a consequence of this two stage scheduling algorithm. The Flooring activity begins on February 6, 2004 in the original schedule whereas it begins only on April 2, 2004. They finish at approximately the same time, October 18, 2004 for the original and October 5, 2004 for the HRPS generated schedule. This indicates the presence of crew idle time for the Flooring activity in the Microsoft Project schedule.

There are some activities, which due to differences in construction strategy may not correspond to one another in the schedules. The Doors and Hardware activity is a good example; it starts on June 1, 2004 and ends on October 1, 2004 in the original schedule whereas it starts on March 19, 2004 and ends on August 23, 2004 in the schedule generated by HRPS. It is worth noting that in this case the overall duration for the activity in both the schedules remain the same.

HRPS has generated a schedule for Concordia University's Engineering and Visual Arts building that retains the job logic and is resource driven. A comparison of the two schedules shows that the schedule generated by HRPS is based on realistic production rates, number of crews and is sensitive towards weather. The benefits of enforcing the work continuity constraint have been stated in Chapter Two. The schedule in bar chart format displays how HRPS is capable of generating a schedule that is capable of doing enforcing crew work continuity.

5.4. Summary

This chapter introduced Concordia University's Engineering and Visual Arts building, which was used as a case study for this research. The case study was used to obtain the job logic and study the hard and soft logic which exists in high-rise building construction. The case study was also used to test the developed model. HRPS was used to generate a schedule using the same information and input that the original schedule was based on. The schedule generated by HRPS was input into Primavera Project Planner 3.1 in order to compare the original schedule and the schedule generated by HRPS using bar charts. The schedule generated by HRPS was then compared with the schedule used to construct the high-rise building.

CHAPTER SIX

CONCLUSIONS

6.1. Summary and Concluding Remarks

An object oriented model which provides: 1) assistance in planning based on information gathered from the literature, industry and a solid case study; 2) resource driven scheduling, based on the techniques of repetitive construction; and 3) tracking and control abilities, based on the Earned Value method has been developed. Practical factors that affects such construction is considered, consequently the developed model can be applied to real-life projects and is capable of providing savings in time and resources. The model is developed using Object Oriented Design in three main stages: 1) analysis; 2) design; and 3) implementation to facilitate the planning, scheduling and control of high-rise building construction. The model also incorporates a number of planning, scheduling and control algorithms for the planning stage, including subcontractors in the scheduling stage and providing detailed Earn Value analysis to track and control the project supporting exception reporting and the generation of different types of reports.

The assistance in planning module is designed using knowledge extracted from the literature, industry experience and a solid case study. The assistance in planning module gives the user the option to use one of the following three options:

1. Using a typical template, which comprises of the most commonly used activities and relationships in such construction;
2. Choosing a number of activities from the typical template, the relationships will automatically be adjusted with the deletion of one or more activities;
3. Manually entering the activities and define their relationships one by one.

It also incorporates the straight line learning curve model in the planning stage, where productivities are adjusted for activities depending on their nature, depending on whether they are labor driven or equipment driven. It can also convert work days to calendar days and calculate the effect of weather on the activities which can get affected.

The scheduling algorithm for subcontractors is based on the method used by El-Rayes (1997) and complies with three primary constraints: 1) multiple precedence relations; 2) resource availability periods; and 3) crew work continuity. The scheduling is done in two stages, the first stage complies with the precedence relationships and checks for the availability of the subcontractor where the second checks and adjusts for crew work continuity. A number of other practical factors were taken into consideration while designing the scheduling algorithm for subcontractors to better reflect the reality of how construction is done in the industry and how subcontractors are managed.

The tracking and control module is based on the Earned Value method. The integrated effect of time and cost are considered by the algorithm and the module has the ability to provide exception reporting and can generate different types of reports. The module can track and control the project based on repetitive unit (each floor), repetitive activity, subcontractor, individual crew, and the entire project. As mentioned above it uses Earned Value principals to calculate schedule and cost variances and is able to forecast the cost of the project upon completion based on current information. The tracking and control module also uses the cubic learning curve model to forecast productivities for activities and updates the schedules according to the newly calculated values.

The model is implemented in a computer system operating in a Windows environment, providing a user friendly interface with menus, toolbars, status bars and dialog windows. All of the modules are implemented in HRPS (High-Rise Planner Scheduler). The model is coded using Microsoft Visual C++ Version 6 and Microsoft Visual C++ .NET. and employs Microsoft Foundation Classes (MFC). The model is capable of generating the schedule, tracking and control reports in different formats at varying levels of detail to suit all tiers of management. A case study of Concordia University's new Engineering and Visual Arts building built in downtown Montreal was also presented.

6.2. Research Contributions

The primary contribution of this research is the development of a domain specific model, capable of providing assistance in planning, resource driven scheduling and tracking and control for high-rise building construction. The development of this model incorporates the following contributions:

1. Identifying the activities required for high-rise building construction and their interrelationships by means of studying the literature, interviewing industry professionals and performing an in-depth analysis of a solid case study.
2. The development of an assistance in planning module, which helps novice planners plan the work related to high-rise building construction.
3. The incorporation of the straight-line learning curve model into the planning stage.
4. The incorporation of the cubic learning curve model into the schedule updating module, and forecasting section of the tracking and control module.
5. Creating an algorithm which integrates subcontractors with the general contractors own work force, using the techniques of repetitive construction.
6. The formulation of an integrated time and cost tracking and control algorithm using the Earned Value technique.

7. The implementation of the proposed algorithms in a user-friendly prototype software system, which can be operated in a Windows environment.
8. The schedule for Concordia University's new EV building was mapped using the developed prototype software.

6.3. Recommendations for Future Research

This study presents a model designed capable of planning, resource driven scheduling and controlling high-rise building construction. The model is flexible and was designed with input from both academia and the industry. However in order to enhance its capabilities and potential applications, the following recommendations for future research can be made:

1. The optimization module is designed using dynamic programming and can be enhanced to more efficient methods such as genetic algorithms.
2. The developed model can only be used by one user at a time. In order to add to the practicality of the model, it can be modified to be stored on a server on the internet and can accept changes from multiple users during the schedule updating process.
3. For the structure, three dimensional scanning methods or time-lapsed photography can be used to measure the progress of work, thus providing automatic input into the tracking and control module.
4. The developed scheduling technique uses deterministic methods, but it can be expanded to produce probabilistic schedules.

5. The ability to create construction work crews can be added, where the user can choose the desired composition and the productivity rates and costs are estimated using help from the database module. If the estimated results are not acceptable, the user must be able to modify the results.
6. Expanding the scheduling algorithm to optimize resource sharing among projects (Hassanein, 2002). This feature would prove particularly useful to general contractors who focus their resources in central urban locations.

REFERENCES

Adeli, H. and Karim, A., 1997, *"Scheduling/Cost Optimization and Neural Dynamics for Construction"*, Journal of Construction Engineering and Management, ASCE, Vol. 123, No. 4, pp. 450 – 458.

Alkass, S. T., Mazerolle, M. and Harris, F., 1996, *"Construction Delay Analysis Techniques"*, Construction Management and Economics, E & FN Spon, Vol. 14, pp. 375 – 394.

Al Sarraj, Z. M., 1990, *"Formal Development of Line-of-Balance Technique"*, Journal of Construction Engineering and Management, ASCE, Vol. 116 No. 4, pp. 689 – 704.

Arditi, D. and Albulak, Z., 1986, *"Line-of-Balance Scheduling in Pavement Construction"*, Journal of Construction Engineering and Management, ASCE, Vol. 112, No. 3, pp. 411 – 424.

Arditi, D., Sikangwan, P. and Tokdemir, O. B., 2002, *"Scheduling System for High Rise Building Construction"*, Construction Management and Economics, E & FN Spon, Vol. 20, pp. 353 – 346.

Ashley, D. B., 1978, *"Simulation of Repetitive Unit Construction"*, Journal of Construction Engineering and Management, ASCE, Vol. 106 No. CO2, pp. 185 – 194.

Barrie, D. S. and Paulson, B. C., 1992, *"Professional Construction Management"*, Third Edition, McGraw Hill Inc., New York, New York, USA.

Birrell, G., 1980, *"Construction Planning – Beyond the Critical Path"*, Journal of Construction Engineering and Management, ASCE, Vol. 106 No. CO3, pp. 389 – 407.

Burleson, R. C., Haas, C. T., Tucker, R. L. and Stanley, A., 1998, *"Multiskilled Labor Utilization Strategies in Construction"*, Journal of Construction Engineering and Management, ASCE, Vol. 124 No. 6, pp. 480 – 489.

Carlson, J.G.H., 1973, *"Cubic Learning Curves: Precision Tool for Labor Estimating"*, Manufacturing Engineering and Management, Nov 1973, pp. 22-25.

Carr, M. I. and Meyer, W. L., 1974, *"Planning Construction of Repetitive Building Units"*, Journal of the Construction Division, ASCE, Nov. 1974, pp. 22 – 25.

Chrzanowski, E. and Johnston, D., 1986, *"Application of Linear Scheduling"*, Journal of Construction Engineering and Management, ASCE, Vol. 112, No. 4, pp. 476 – 491.

Cole, L., 1991, *"Construction Scheduling: Principles, Practices, and Six Case Studies"*, Journal of Construction Engineering and Management, ASCE, Vol. 117, No. 4, pp. 579 – 588.

Deitel, H. M., Deitel, P. J., 2003, *"C++ How to Program"*, Fourth Edition, Prentice Hall, New Jersey, U.S.A.

Department of Energy, 2006, United States of America Department of Energy Website, Office of Engineering and Construction Management, Information Center, (http://oecm.energy.gov/Default.aspx?tabid=143).

Eldin, N. and Senouci, A., 1994, *"Scheduling and Control of Linear Projects"*, Canadian Journal of Civil Engineering, CSCE, Vol. 21, pp. 219 – 230.

El-Rayes, K., 1997, *"Optimized Scheduling for Repetitive Construction Projects"*, PhD thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Quebec, Canada.

El-Rayes, K., 2001, *"Object-Oriented Model for Repetitive Construction Scheduling"*, Journal of Construction Engineering and Management, ASCE, Vol. 127, No. 3, pp. 199 – 205.

El-Rayes, K. and Moselhi, O., 1998, *"Resource-Driven Scheduling of Repetitive Activities"*, Construction Management and Economics, E & FN Spon, Vol. 16, pp. 433 – 446.

El-Rayes, K., Ramanathan, R. and Moselhi, O., 2002, *"An Object-oriented model for planning and control of housing construction"*, Construction Management and Economics, E & FN Spon, Vol. 20, pp. 201 – 210.

Emporis Buildings, 2005, Emporis Buildings Website. High-Rise Statistics, (http://www.emporis.com/en/bu/sk/st/).

Gomar, J. E., Haas, C. T. and Morton, D. P., 2002, *"Assignment and Allocation Optimization of Partially Multiskilled Workforce"*, Journal of Construction Engineering and Management, ASCE, Vol. 128 No. 2, pp. 103 – 109.

Halpin, D. W. and Woodhead, R. W., 1976, *"Design of Construction Process Operations"*, Wiley, New York, New York, U.S.A.

Handa, V. and Barcia, R., 1986, "*Scheduling Linear Projects Using Optimal Control Theory*", Journal of Construction Engineering and Management, ASCE, Vol. 112, No. 3, pp. 387 – 393.

Harmelink, D. J., 2001, "*Linear Scheduling Model: Float Characteristics*", Journal of Construction Engineering and Management, ASCE, Vol. 127 No. 4, pp. 255 – 260.

Harris, R. B., 1996, "*Scheduling Projects with Repeating Activities*", UMCEE Report No. 96-26, Civil and Environmental Engineering Department, University of Michigan, Ann Arbor, Michigan, U.S.A.

Hassanein, A., 2002, "*Planning and Scheduling Highway Construction Using GIS and Dynamic Programming*", PhD thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Quebec, Canada.

Herbsman, Z. J., 1995, "*A+B Bidding Method – Hidden Success Story for Highway Construction*", Journal of Construction Engineering and Management, ASCE, Vol. 121, No. 4, pp. 430 – 437.

Hegazy, T., 1999, "*Optimization of Resource Allocation Using Genetic Algorithms*", Journal of Construction Engineering and Management, ASCE, Vol. 125, No. 3, pp. 167 – 175.

Hegazy, T. and Wassef, N., 2001, "*Cost Optimization with Repetitive Non-serial Activities*", Journal of Construction Engineering and Management, ASCE, Vol. 127, No. 3, pp. 183 – 191.

Johnston, D., 1981, "*Linear Scheduling Method for Highway Construction*", Journal of the Construction Division, ASCE, Vol. 107, No. CO2, pp. 247 – 261.

Kavanagh, D. P., 1985, "*SIREN: A Repetitive Construction Simulation Model*", Journal of Construction Engineering and Management, ASCE, Vol. 111, No. 3, pp. 308 – 323.

Khoshafian, S., and Abnous, R., 1995, "*Object Orientation: Concepts, Analysis and Design, Languages*", Database, Graphical User Interface, Standards, Wiley, New York, New York, U.S.A.

Kim, K. and de la Garza, J. M., 2003, "*Phantom Float*", Journal of Construction Engineering and Management, ASCE, Vol. 129, No. 5, pp. 507 – 517.

Laramee, J. B., 1983, "*A Planning and Scheduling System for High-Rise Building Construction*", M. Eng. Thesis, Center for Building Studies, Concordia University, Montreal, Quebec, Canada.

Lumsden, P. W., 1968, *"The Line of Balance Method"*, Pergamon Press, Elmsford, New York, New York, U.S.A.

Moselhi, O., 1993, *"Applied Earned Value for Project Control"*, CIB W-65, Trinidad, W.I., September 1993, pp. 144 – 152.

Moselhi, O., 2004, *"Construction Planning and Control"*, Graduate Course # BLDG 6801, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Quebec, Canada, January – April 2004.

Moselhi, O. and El-Rayes, K., 1993(a), *"Scheduling of Repetitive Projects with Cost Optimization"*, Journal of Construction Engineering and Management, ASCE, Vol. 118, No. 4, pp. 681 – 697.

Moselhi, O. and El-Rayes, K., 1993(b), *"Least Cost Scheduling for Repetitive Projects"*, Canadian Journal of Civil Engineering, CSCE, Vol. 20, pp. 834 – 843.

Moselhi, O. and Hassanein, A., 2003, *"Optimized Scheduling of Linear Projects"*, Journal of Construction Engineering and Management, ASCE, Vol. 129, No. 6, pp. 664 – 673.

Moselhi, O., Hegazy, T. and Fazio, P., 1993, *"DBID – Analogy – Based DSS for Bidding in Construction"*, Journal of Construction Engineering and Management, ASCE, Vol. 119, No. 3, pp. 466 – 479.

Moselhi, O. and Nicholas, M. J., 1990, *"Hybrid Expert System for Construction Planning and Scheduling"*, Journal of Construction Engineering and Management, ASCE, Vol. 116, No. 2, pp. 221 – 238

O'Brien, J. J., 1975, *"VPM Scheduling for High-Rise Buildings"*, Journal of the Construction Division, ASCE, Vol. 101, No. 4, pp. 895 – 905.

Oglesby, C., 1989, Productivity Improvement in Construction, McGraw Hill Inc, N.Y., U.S.A.

Peter Kiewit Sons' Inc., 1997, *"Engineering Manual"*, Volume Three, Peter Kiewit Sons' Inc, Omaha, Nebraska, U.S.A.

Ramanathan, R., 2000, *"Computerized Scheduling and Control of Residential Housing Projects"*, MASC thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Quebec, Canada.

Reda, R. M., 1990, *"RPM: Repetitive Project Modeling"*, Journal of Construction Engineering and Management, ASCE, Vol. 116, No. 2, pp. 316 – 330.

Rumbaugh, J., Blaha, M. Premerlani, W., Eddy, F. and Lorensen, W., 1991, *"Object-Oriented Modeling and Design"*, Englewood Cliffs, Prentice Hall, New Jersey. U.S.A.

Russell, A. D. and Caselton, W. F., 1988, *"Extensions to Linear Scheduling Optimization"*, Journal of Construction Engineering and Management, ASCE, Vol. 114, No. 1, pp. 36 – 52.

Russell A. D. and McGowan, N., 1993, *"Linear Scheduling: A Practical Implementation"*, Computing in Civil Engineering, ASCE, Anaheim, California, USA, June 7 - 9, pp. 279 – 286.

Russell A. D. and Wong, W. C. M., 1993, *"New Generation of Planning Structures"*, Journal of Construction Engineering and Management, ASCE, Vol. 119, No. 2, pp. 196 – 214.

Selinger, S., 1980, *"Construction Planning for Linear Projects"*, Journal of the Construction Division, ASCE, Vol. 106, No. CO2, pp. 195 – 205.

Suh, K., 1993, *"RUSS: A Scheduling System for Repetitive-Unit Construction Using Line-of-Balance Technology"*, PhD thesis, Illinois Institute of Technology, Chicago, Illinois, U.S.A.

Suhail S. A. and Neale, R. H., 1994, *"CPM and Line of Balance"*, Journal of Construction Engineering and Management, ASCE, Vol. 120, No. 3, pp. 667 – 684.

Tamimi, S. and Diekmann, J., 1988, *"Soft Logic in Network Analysis"*, Journal of Computing in Civil Engineering, ASCE, Vol. 2, No. 3, pp. 289 – 300

Thabet, W. Y. and Beliveau, Y. L.,1994, *"HVLS: Horizontal and Vertical Logic Scheduling for Multistory Projects"*, Journal of Construction Engineering and Management, ASCE, Vol. 120, No. 4, pp. 875 – 892.

Thomas, H. R., Mathews, C. T. and Ward, G. T., 1986, *"Learning Curve Models of Construction Productivity"*, Journal of Construction Engineering and Management, ASCE, Vol. 112, No. 2, pp. 245 – 258

Vorster, M. C., Beliveau, V. J. and Bafna, T., 1992, *"Linear Scheduling and Visualization"*, Transportation Research Record, Vol. 1351, pp. 32-39.

Ward, J. G. and Thomas, H. R. Jr., 1984, *"A Validation of Learning Curve Models Available to the Construction Industry"*, Construction Management Research Series Report No.20, The Pennsylvania State University, Department of Civil Engineering, University Park, Pennsylvania, Aug. 1984, pp 174.

Wright, J. R., 1936, *"Factors Affecting the Cost of Airplanes"*, Journal of Aeronautical Sciences, Feb 1936, pp.124 – 125.

APPENDIX I

HRPS OUTPUT

Activity	Unit	Crew #	Duration	ES	EF
Cut & chip trees	1	SUBA	1	0	1
Cut & chip trees	2	SUBA	1	1	2
Cut & chip trees	3	SUBA	2	2	3.5
Cut & chip trees	4	SUBA	1	3.5	4.5
Cut & chip trees	5	SUBA	2	4.5	6
Cut & chip trees	6	SUBA	3	6	8.5
Cut & chip trees	7	SUBA	3	8.5	11.5
Cut & chip trees	8	SUBA	3	11.5	14
Cut & chip trees	9	SUBA	2	14	16
Cut & chip trees	10	SUBA	2	16	18
Cut & chip trees	11	SUBA	2	18	19.5
Cut & chip trees	12	SUBA	1	19.5	20.5
Cut & chip trees	13	SUBA	1	20.5	21.5
Cut & chip trees	14	SUBA	1	21.5	22.5
Cut & chip trees	15	SUBA	1	22.5	23.5
Grub & remove chips	1	1	3	1	4
Grub & remove chips	2	2	3	2	5
Grub & remove chips	3	1	5	4	8.5
Grub & remove chips	4	2	3	5	8
Grub & remove chips	5	2	5	8	12.5
Grub & remove chips	6	1	8	8.5	16
Grub & remove chips	7	2	9	12.5	21.5
Grub & remove chips	8	1	8	16	23.5
Grub & remove chips	9	2	6	21.5	27.5
Grub & remove chips	10	1	6	23.5	29.5
Grub & remove chips	11	2	5	27.5	32
Grub & remove chips	12	1	3	29.5	32.5
Grub & remove chips	13	2	3	32	35
Grub & remove chips	14	1	3	32.5	35.5
Grub & remove chips	15	2	3	35	38
Earthmoving	1	1	5	4	9
Earthmoving	2	2	8	5.25	12.75
Earthmoving	3	1	5	9	14
Earthmoving	4	2	9	12.75	21.5
Earthmoving	5	1	8	14	21.17
Earthmoving	6	1	6	21.17	27
Earthmoving	7	2	9	21.5	29.63
Earthmoving	8	1	5	27	32
Earthmoving	9	2	8	29.63	37.13
Earthmoving	10	1	5	32	37
Earthmoving	11	1	8	37	42
Earthmoving	12	2	5	37.13	44.63
Earthmoving	13	1	5	42	47
Earthmoving	14	2	8	44.63	52.13
Earthmoving	15	1	5	47	52

Figure I - 1: Numerical Example III Schedule Page 1

Activity	Unit	Crew #	Duration	ES	EF
Base	1	SUBB	2	26.13	28.13
Base	2	SUBB	2	28.13	30.13
Base	3	SUBB	2	30.13	32.13
Base	4	SUBB	2	32.13	34.13
Base	5	SUBB	2	34.13	36.13
Base	6	SUBB	2	36.13	38.13
Base	7	SUBB	2	38.13	40.13
Base	8	SUBB	2	40.13	42.13
Base	9	SUBB	2	42.13	44.13
Base	10	SUBB	2	44.13	46.13
Base	11	SUBB	2	46.13	48.13
Base	12	SUBB	2	48.13	50.13
Base	13	SUBB	2	50.13	52.13
Base	14	SUBB	2	52.13	54.13
Base	15	SUBB	2	54.13	56.13
Paving	1	1	8	28.13	36.13
Paving	2	2	8	30.13	38.13
Paving	3	3	8	32.13	40.13
Paving	4	1	8	36.13	44.13
Paving	5	2	8	38.13	46.13
Paving	6	3	8	40.13	48.13
Paving	7	1	8	44.13	52.13
Paving	8	2	8	46.13	54.13
Paving	9	3	8	48.13	56.13
Paving	10	1	8	52.13	60.13
Paving	11	2	8	54.13	62.13
Paving	12	3	8	56.13	64.13
Paving	13	1	8	60.13	68.13
Paving	14	2	8	62.13	70.13
Paving	15	3	8	64.13	72.13

Figure I - 2: Numerical Example III Schedule Page 2

Activity	Unit	Crew #	Duration	ES	EF
Act A	1	1	4	1	4
Act A	2	2	5	1	5
Act A	3	1	6	5	10
Act A	4	2	5	6	10
Act A	5	1	4	11	14
Act B	1	1	3	9	11
Act B	2	2	4	7	10
Act B	3	1	3	12	14
Act B	4	2	6	11	16
Act B	5	1	5	15	19
Act C	1	1	2	12	13
Act C	2	1	3	14	16
Act C	3	1	4	17	20
Act C	4	1	6	21	26
Act C	5	1	9	27	35
Act D	1	SUBD	6	16	21
Act D	2	SUBD	4	22	25
Act D	3	SUBD	7	26	32
Act D	4	SUBD	3	33	35
Act D	5	SUBD	8	36	43
Act E	1	1	2	28	29
Act E	2	1	3	30	32
Act E	3	1	6	33	38
Act E	4	1	5	39	43
Act E	5	1	2	44	45

Figure I - 3: Numerical Example IV Schedule

Activity	Unit	ACWP	BCWP	BCWS	SV	CV	AV	CPI	SPI
Act A	1	108000	108000	108000	0	0	0	1	1
Act A	2	135000	135000	135000	0	0	0	1	1
Act A	3	157500	162000	162000	0	-4500	-4500	1.02857	1
Act A	4	135000	135000	135000	0	0	0	1	1
Act A	5	104500	108000	108000	0	-3500	-3500	1.03349	1
Act B	1	156000	155000	154500	500	1000	1500	0.99359	1.00324
Act B	2	207500	206500	206000	500	1000	1500	0.995181	1.00243
Act B	3	154500	154500	154500	0	0	0	1	1
Act B	4	307500	310000	309000	1000	-2500	-1500	1.00813	1.00324
Act B	5	103000	103000	154500	-51500	0	-51500	1	0.666667
Act C	1	305000	305000	305000	0	0	0	1	1
Act C	2	455000	460000	457500	2500	-5000	-2500	1.01099	1.00546
Act C	3	152500	152500	152500	0	0	0	1	1
Act C	4	0	0	0	0	0	0	0	0
Act C	5	0	0	0	0	0	0	0	0
Act D	1	600000	600000	400000	200000	0	200000	1	1.5
Act D	2	0	0	0	0	0	0	0	0
Act D	3	0	0	0	0	0	0	0	0
Act D	4	0	0	0	0	0	0	0	0
Act D	5	0	0	0	0	0	0	0	0
Act E	1	0	0	0	0	0	0	0	0
Act E	2	0	0	0	0	0	0	0	0
Act E	3	0	0	0	0	0	0	0	0
Act E	4	0	0	0	0	0	0	0	0
Act E	5	0	0	0	0	0	0	0	0

Figure I - 4: Numerical Example IV Activity Control Results

Entire Project

==============	==============
sumACWP	3081000
sumBCWP	3094500
sumBCWS	2941500
SV	153000
CV	-13500
AV	139500
CPI	1.0044
SPI	1.052

Figure I - 5: Numerical Example IV Project Control Results

APPENDIX II

CONSTRUCTION PROCESSES PICTURES

(CONCORDIA UNIVERSITY EV BUILDING)

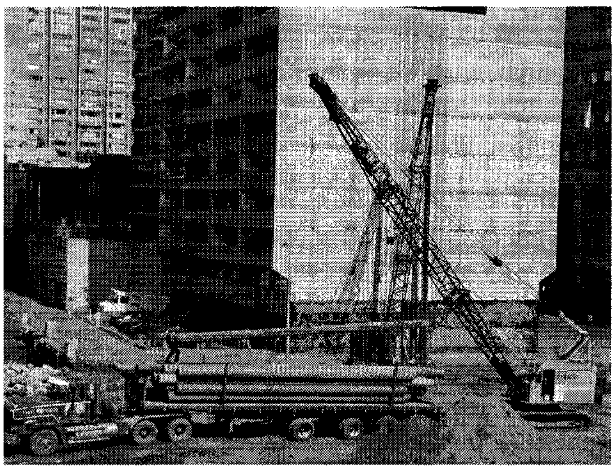


Figure II - 1: Construction Processes 1

Figure II - 2: Construction Processes 2

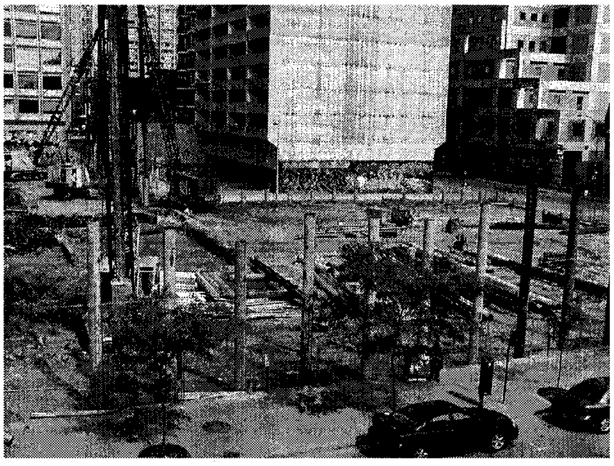


Figure II - 3: Construction Processes 3

Figure II - 4: Construction Processes 4

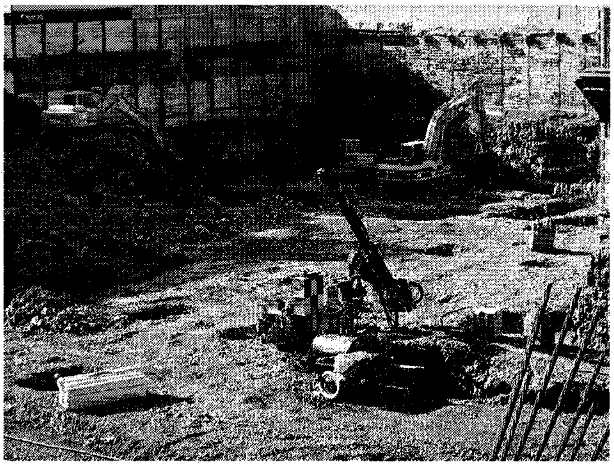


Figure II - 5: Construction Processes 5

Figure II - 6: Construction Processes 6

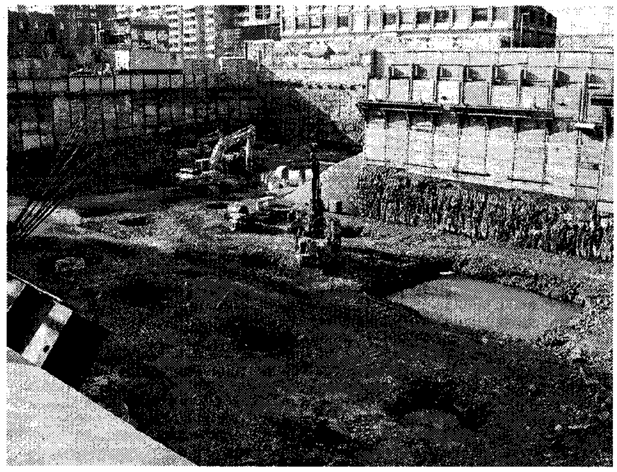


Figure II - 7: Construction Processes 7

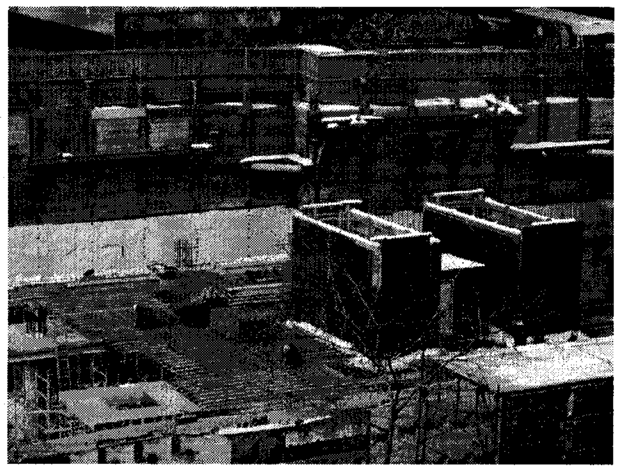


Figure II - 8: Construction Processes 8

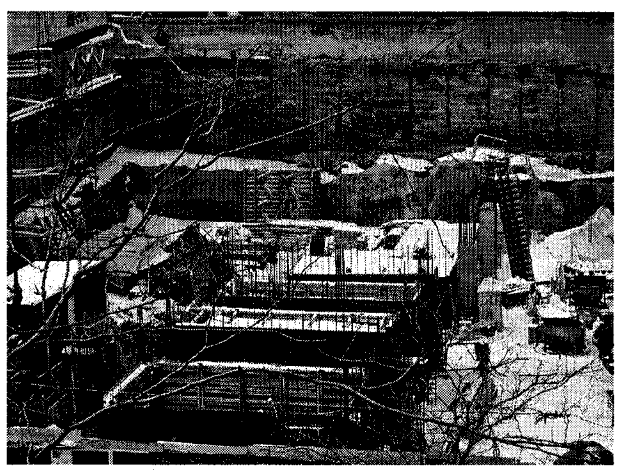


Figure II - 9: Construction Processes 9

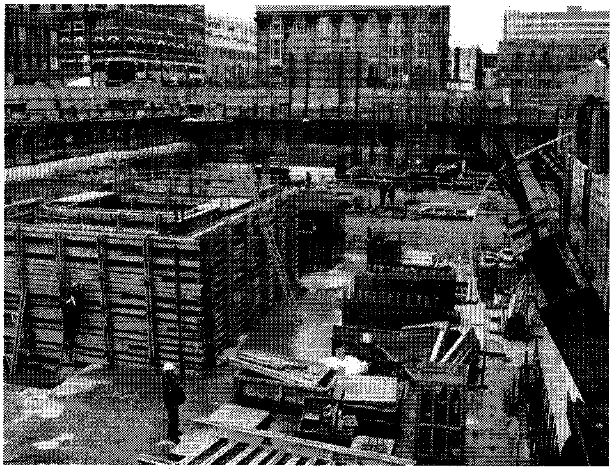


Figure II - 10: Construction Processes 10

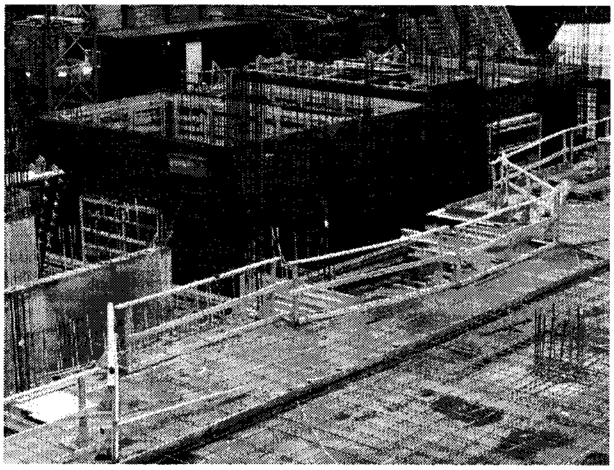


Figure II - 11: Construction Processes 11

Figure II - 12: Construction Processes 12

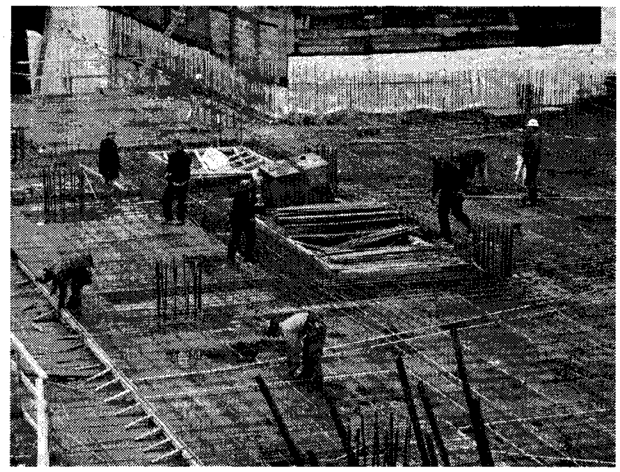


Figure II - 13: Construction Processes 13

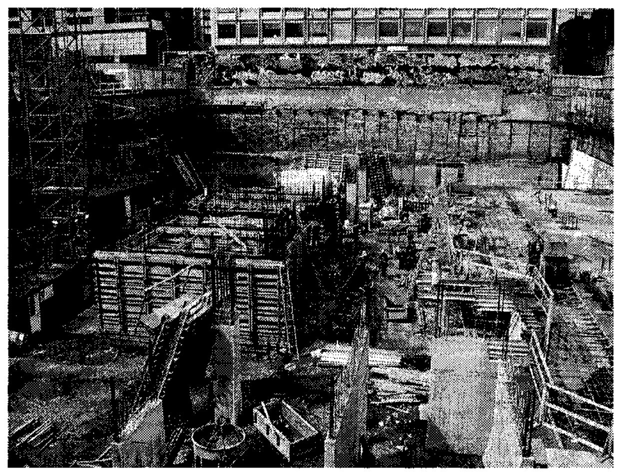


Figure II - 14: Construction Processes 14

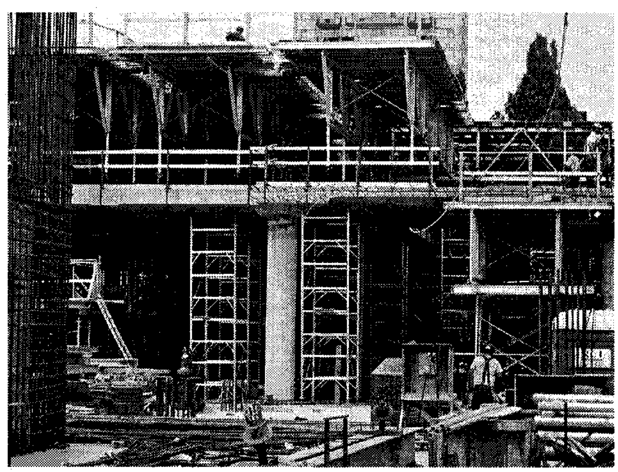


Figure II - 15: Construction Processes 15

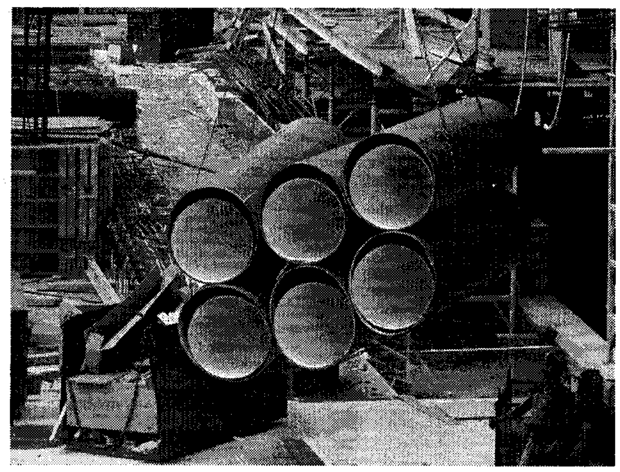


Figure II - 16: Construction Processes 16

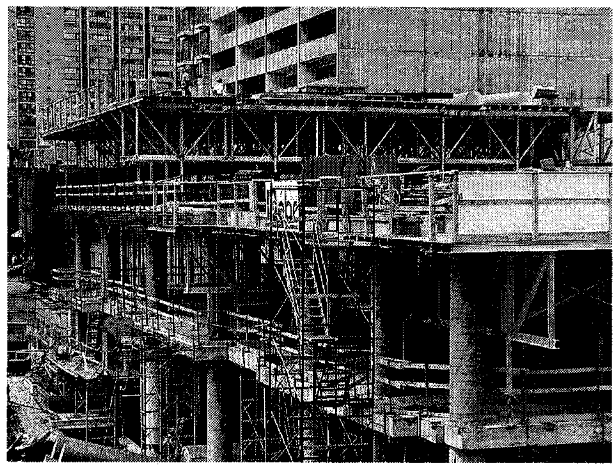


Figure II - 17: Construction Processes 17

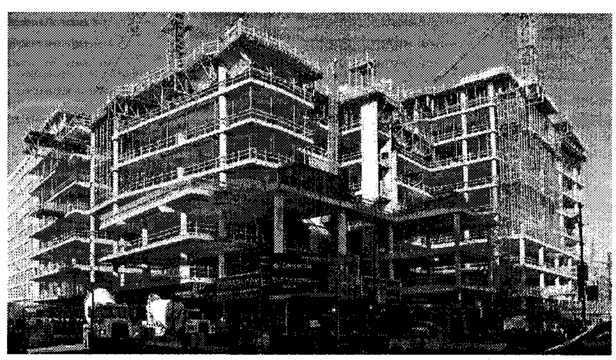


Figure II - 18: Construction Processes 18

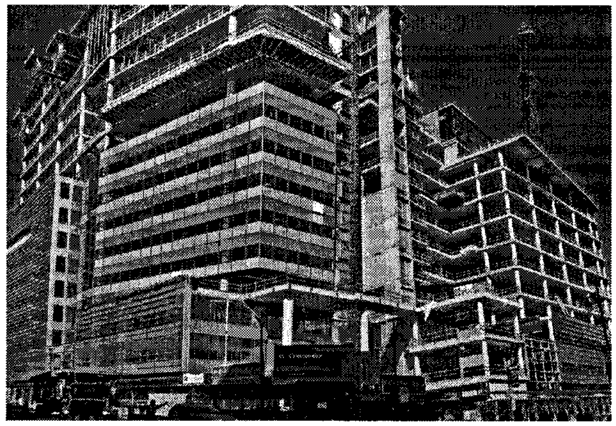


Figure II - 19: Construction Processes 19

Figure II - 20: Construction Processes 20

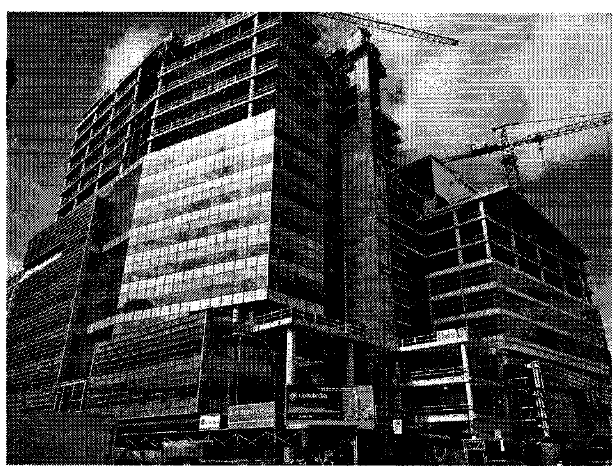


Figure II - 21: Construction Processes 21

APPENDIX III

SAMPLES OF THE PRECEDENCE RELATIONSHIPS USED IN THE PLANNING MODULE

Table III - 1: Examples of Repetitive Relations

Predecessor	Successor	Relation Type	Lag	
Ceilings	Ventilation Finish	Finish to Finish	0	
Concrete Structure	Ventilation Rough	Finish to Start	0	
Doors and Hardware	Floors	Start to Start	10	
Gypsum Boards(Walls)	Painting	Start to Start	0.75	of predecessor duration
Interior Glazing	Painting	Finish to Finish	0	
Painting	Cabinet Work	Finish to Start	0	
Painting	Ceilings	Start to Start	10	
Painting	Wooden Boards(Ceil)	Finish to Start	0	
Plumbing Rough	Plumbing Finish	Finish to Start	0	
Ventilation Rough	Plumbing Rough	Start to Start	5	
Ventilation Rough	Steel Structure	Start to Start	0	

Table III - 2: Examples of Hetero Relations

Predecessor	unit	Successor	unit	Relation Type	Lag
Concrete Structure	3	Curtain Wall	1	Finish to Start	0
Concrete Structure	10	Elevator Doors	1	Finish to Start	0
Concrete Structure	4	Steel Structure	1	Finish to Start	0

APPENDIX IV

HRPS GENERATED SCHEDULE FOR CONCORDIA UNIVERSITY'S EV BUILDING

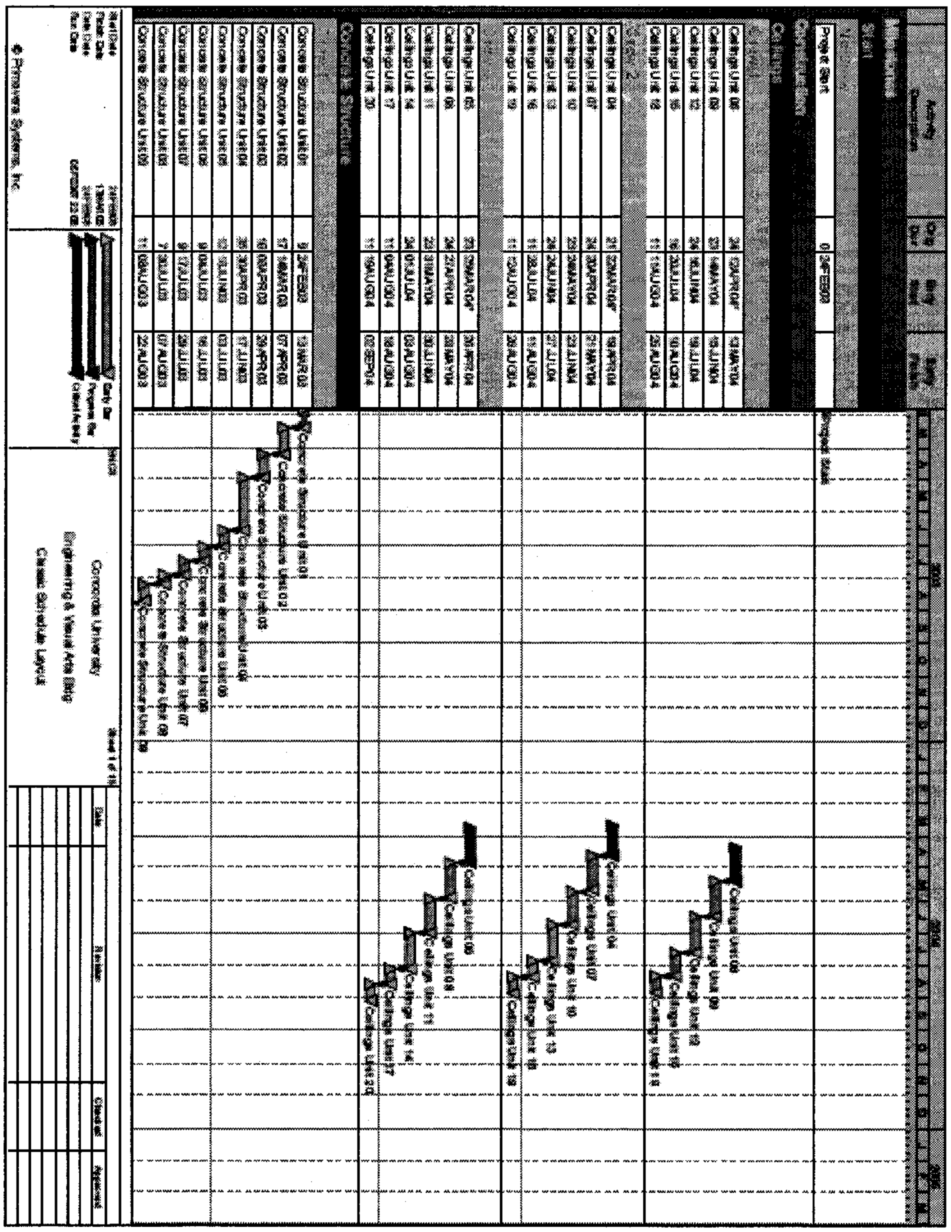


Figure IV - 1: HRPS Generated Schedule in Bar Chart Format Page 1

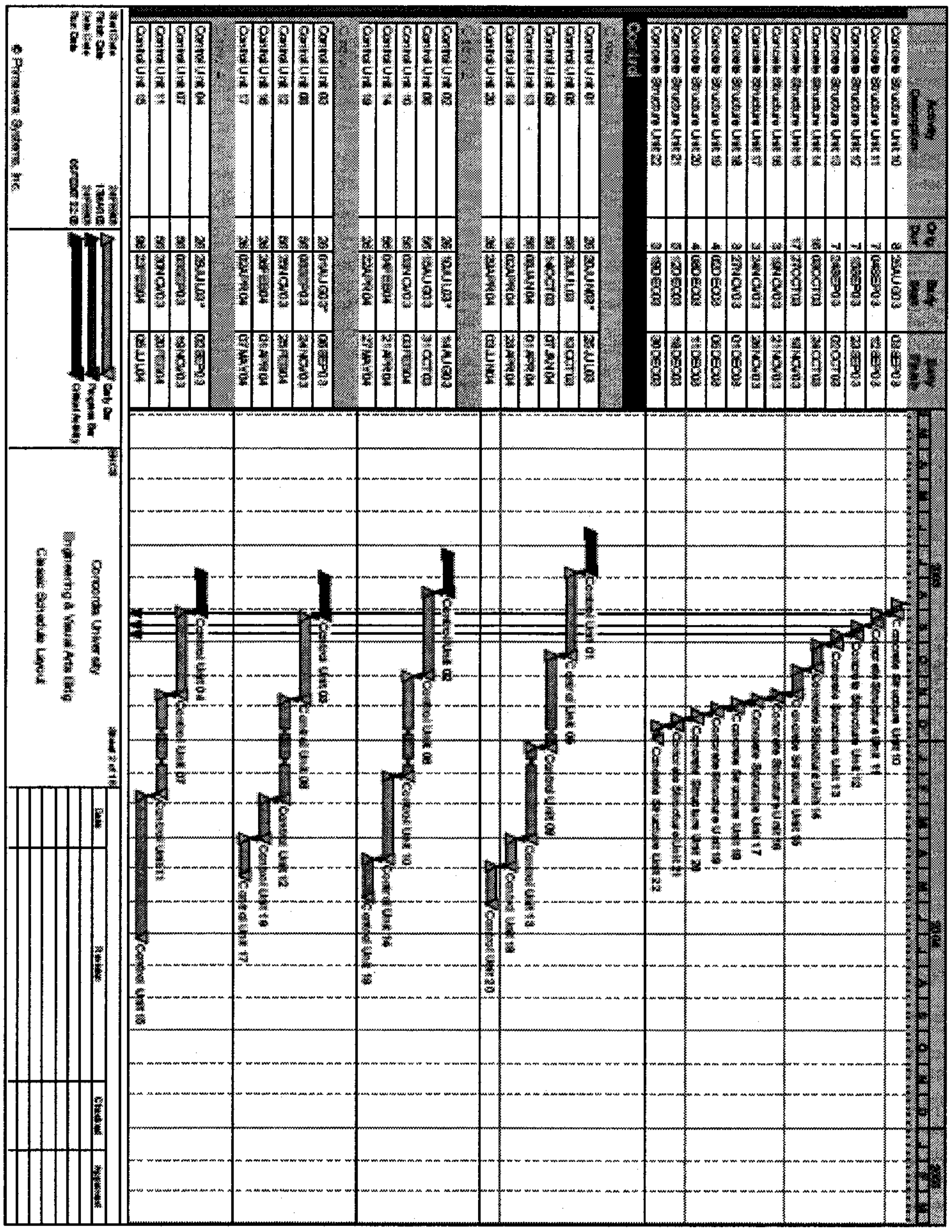


Figure IV - 2: HRPS Generated Schedule in Bar Chart Format Page 2

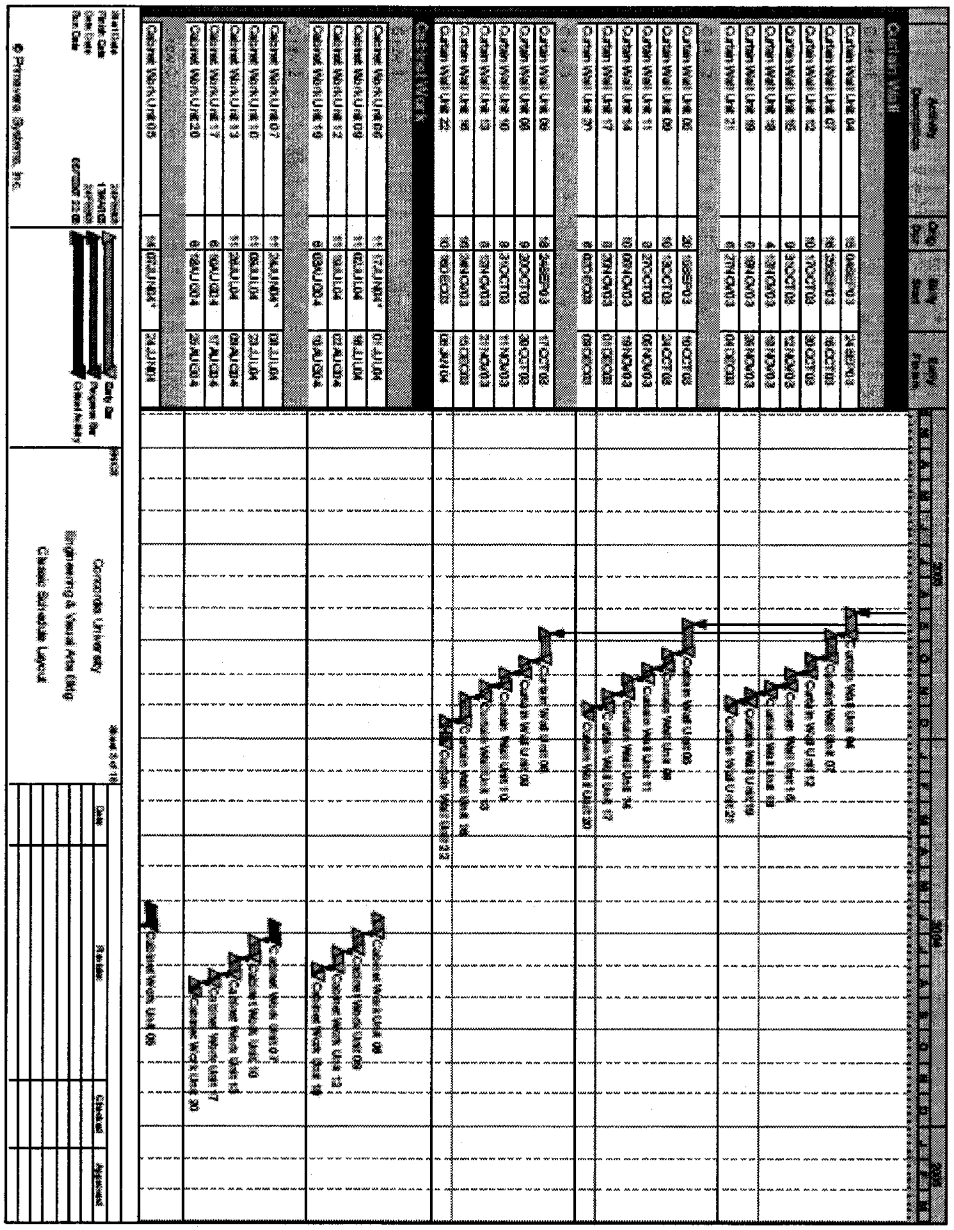


Figure IV - 3: HRPS Generated Schedule in Bar Chart Format Page 3

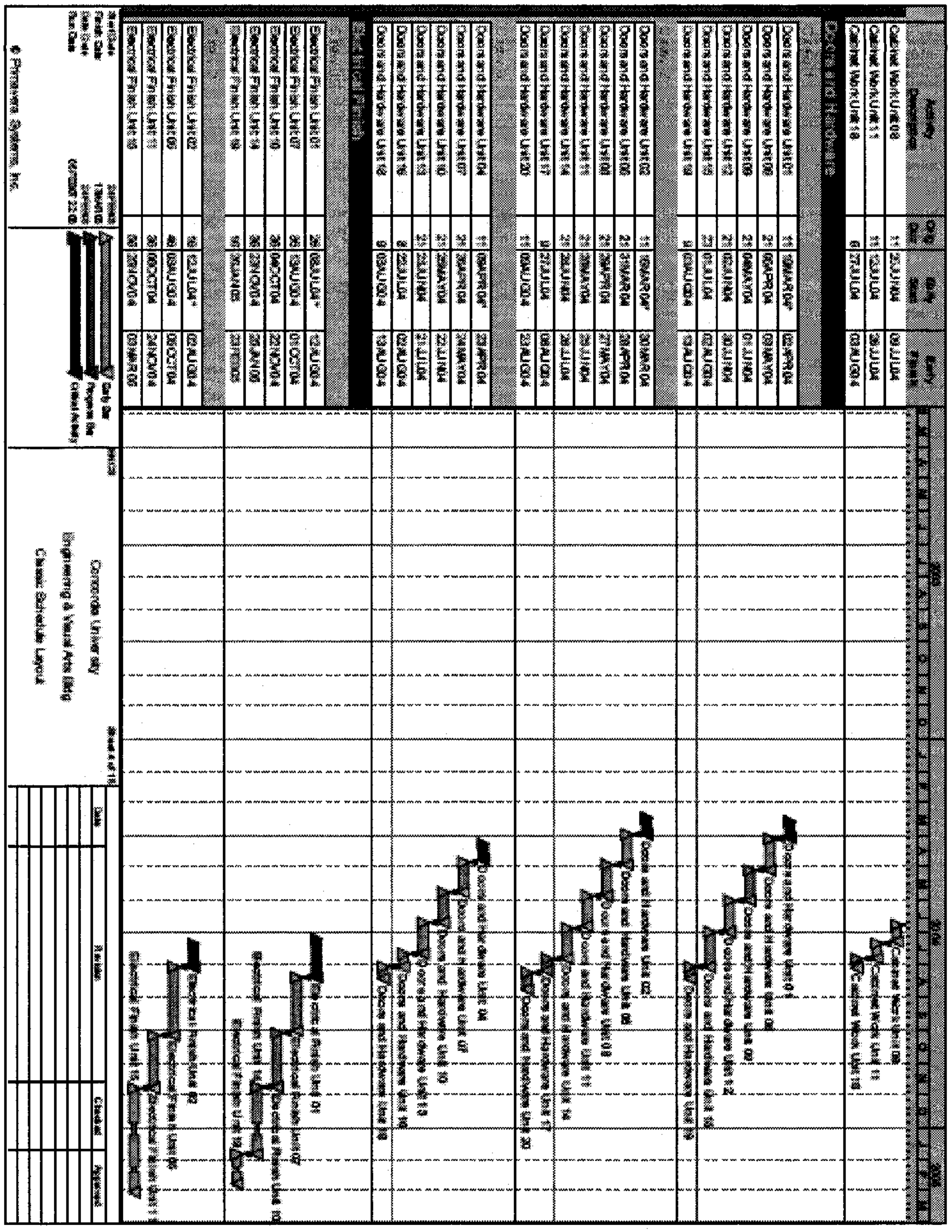


Figure IV - 4: HRPS Generated Schedule in Bar Chart Format Page 4

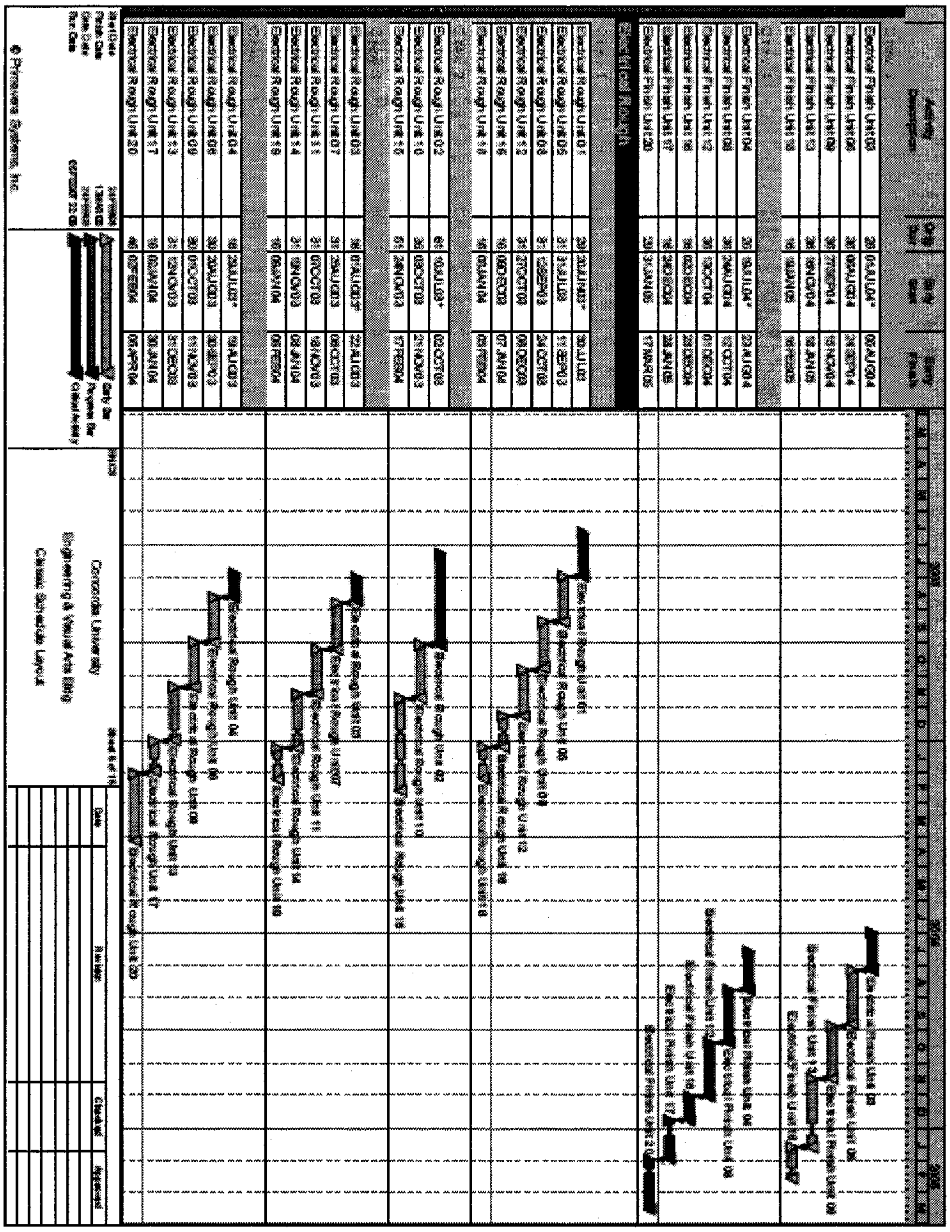


Figure IV - 5: HRPS Generated Schedule in Bar Chart Format Page 5

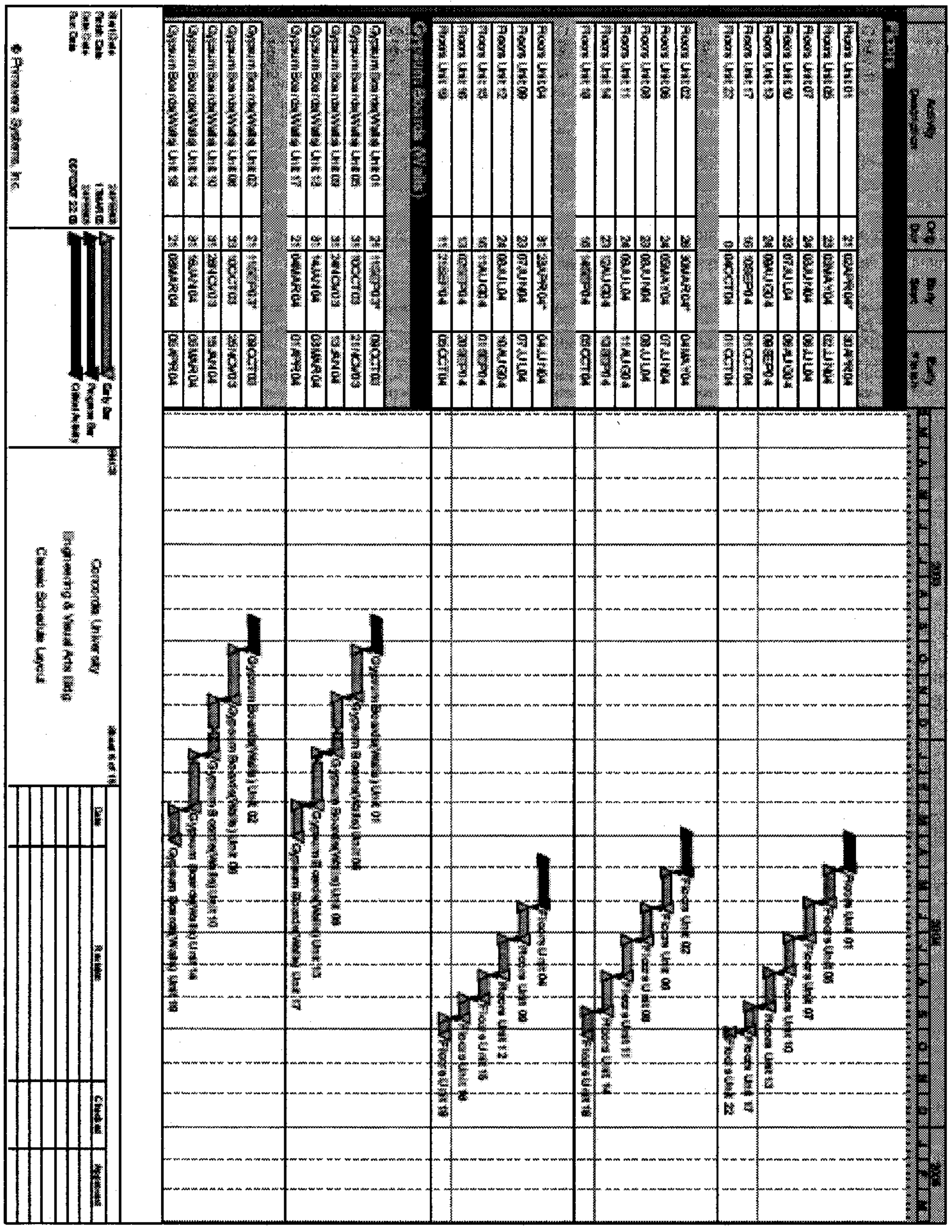


Figure IV - 6: HRPS Generated Schedule in Bar Chart Format Page 6

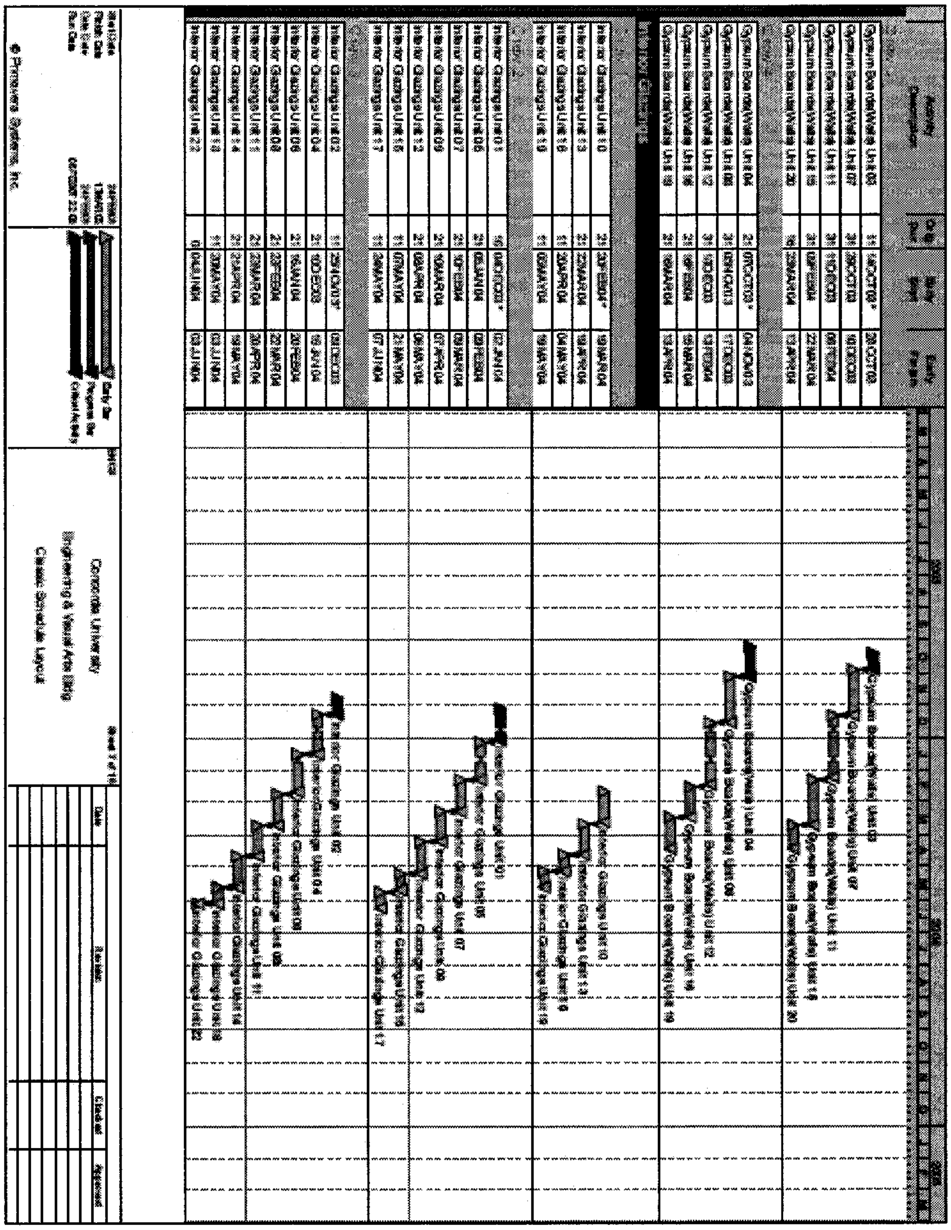


Figure IV - 7: HRPS Generated Schedule in Bar Chart Format Page 7

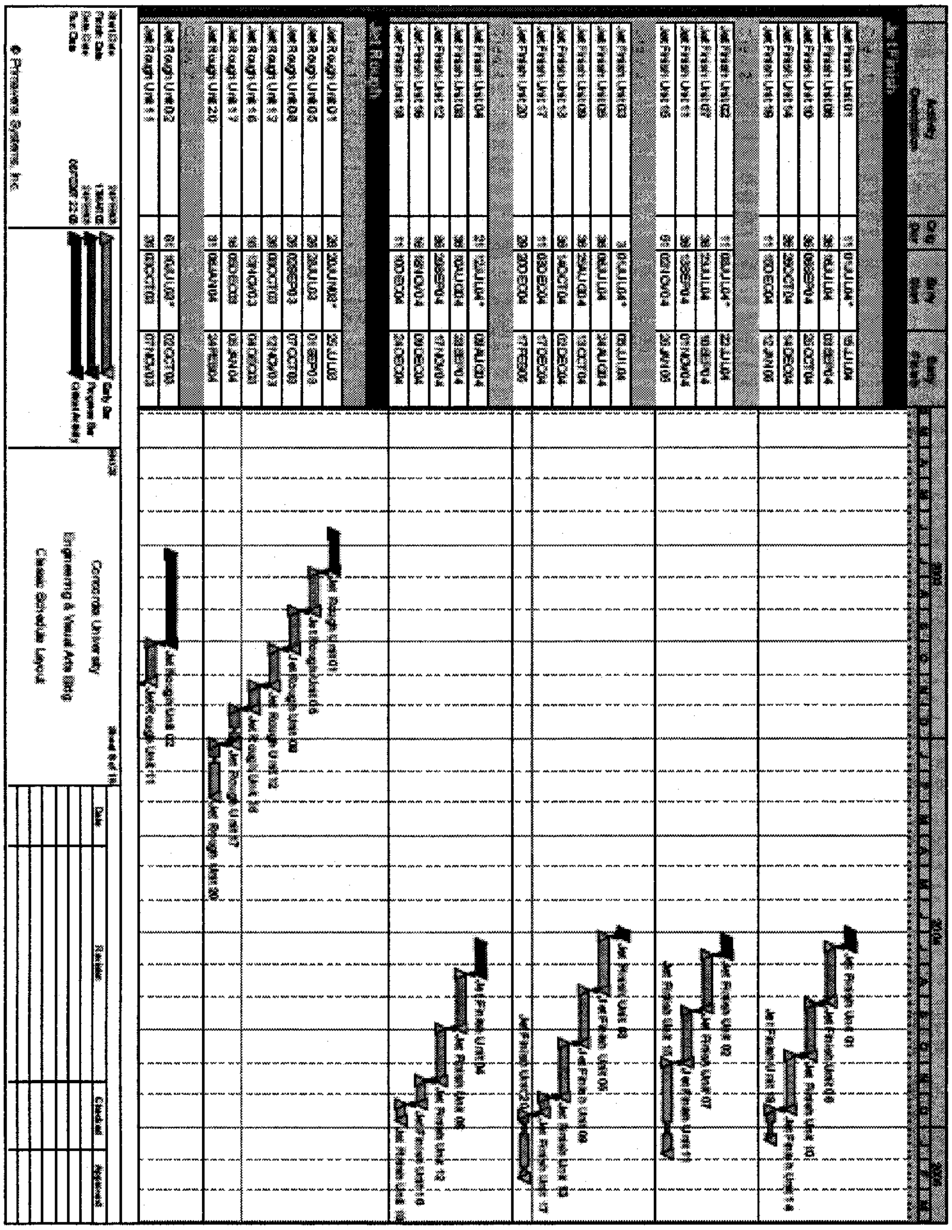


Figure IV - 8: HRPS Generated Schedule in Bar Chart Format Page 8

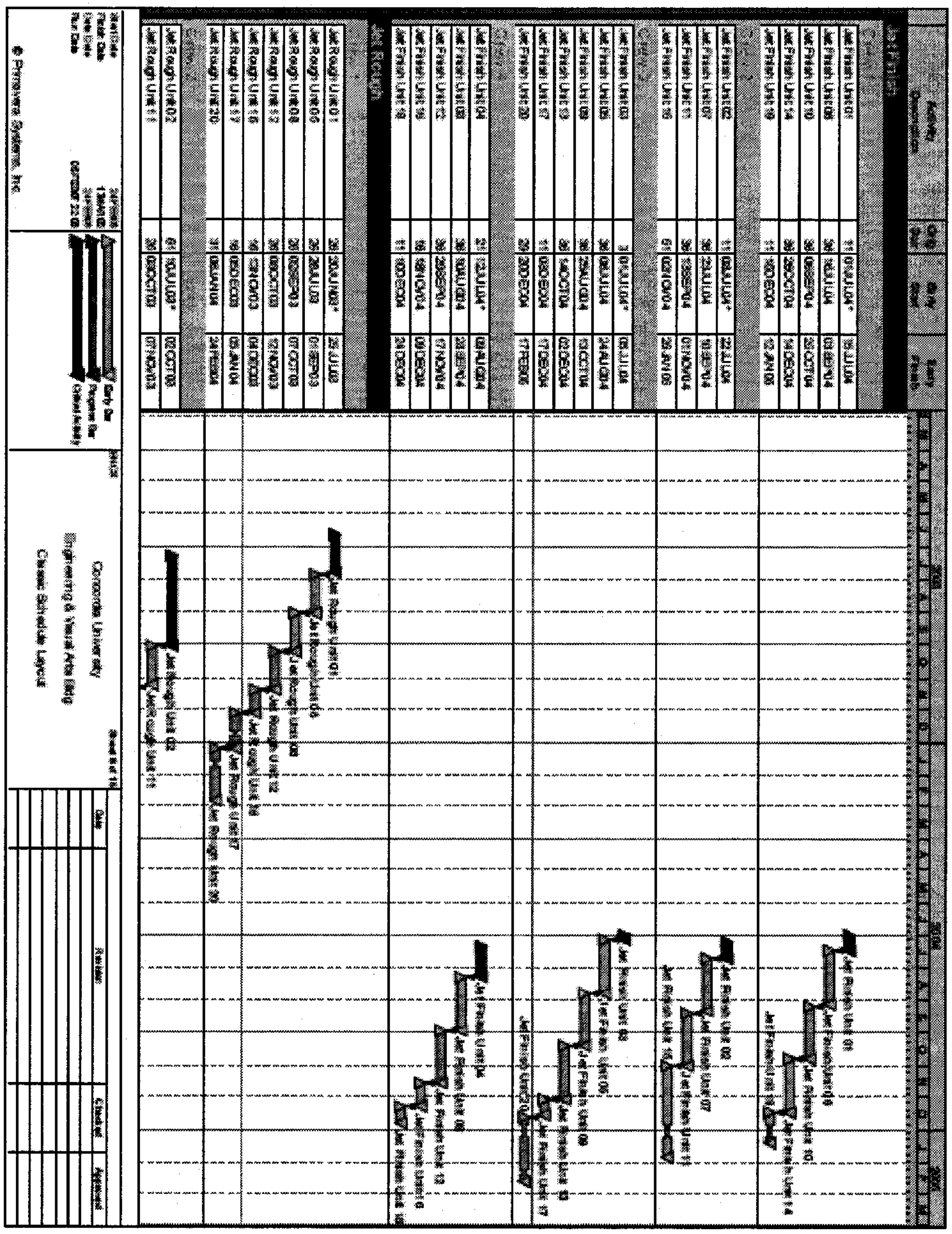


Figure IV - 9: HRPS Generated Schedule in Bar Chart Format Page 9

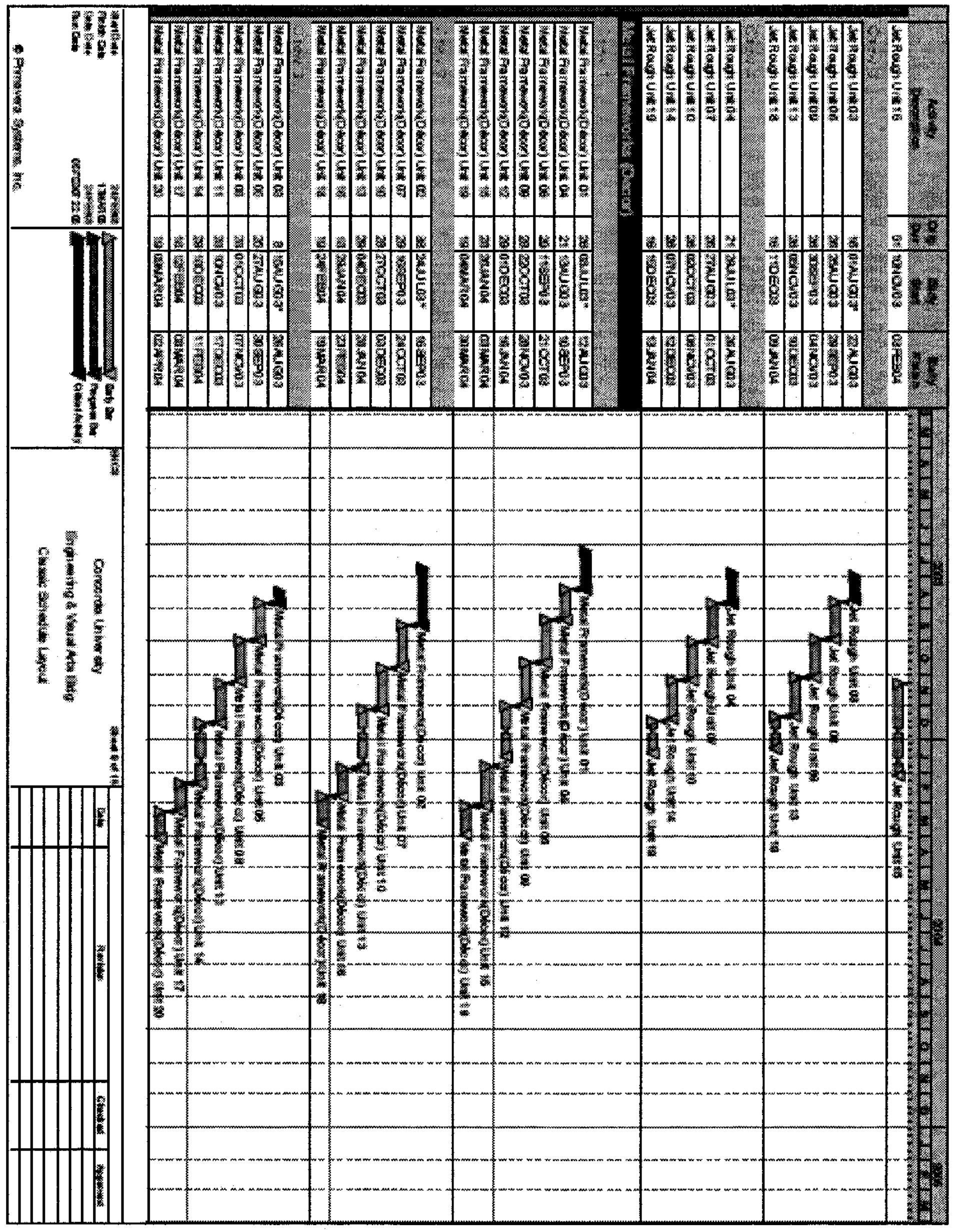


Figure IV - 10: HRPS Generated Schedule in Bar Chart Format Page 10

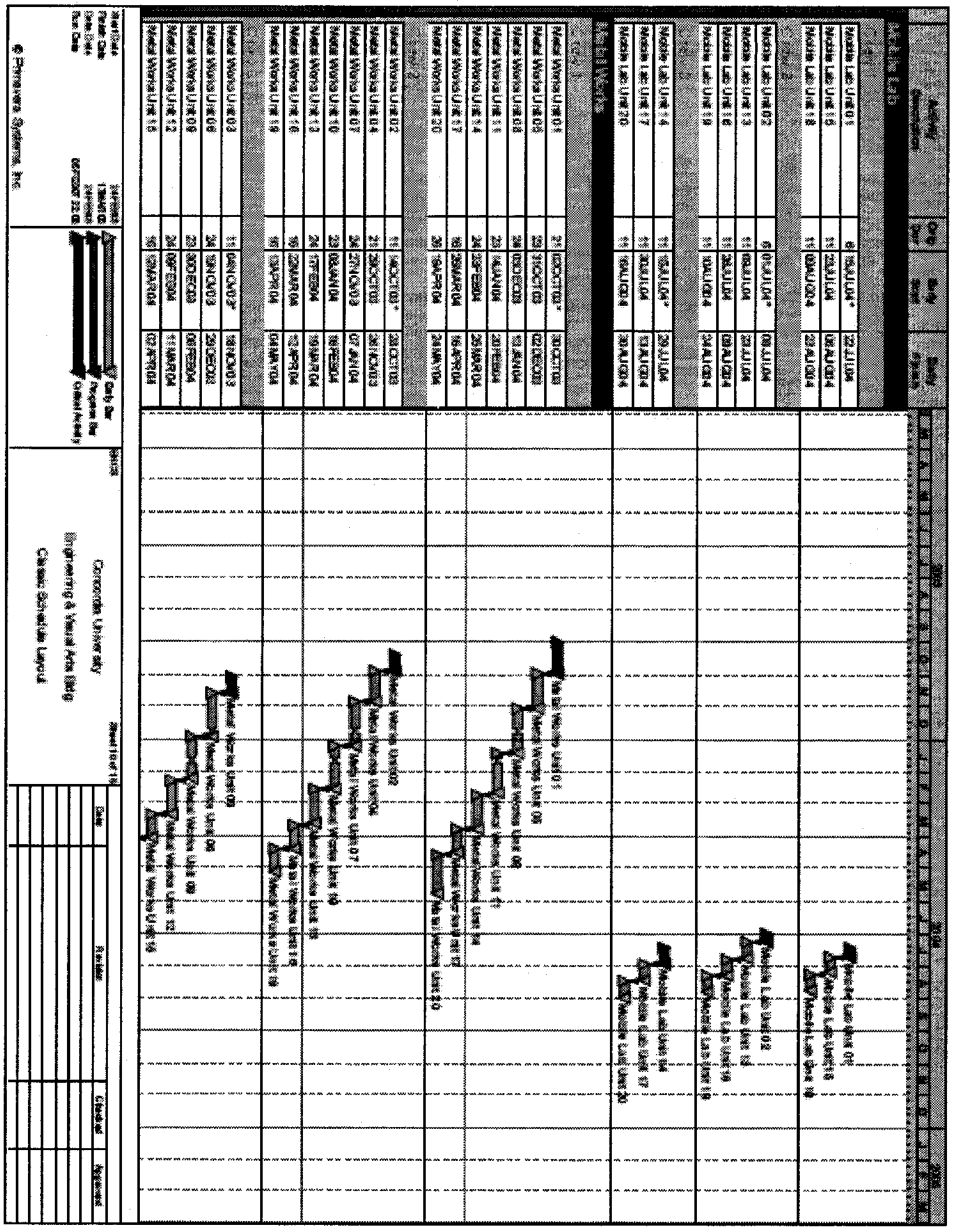


Figure IV - 11: HRPS Generated Schedule in Bar Chart Format Page 11

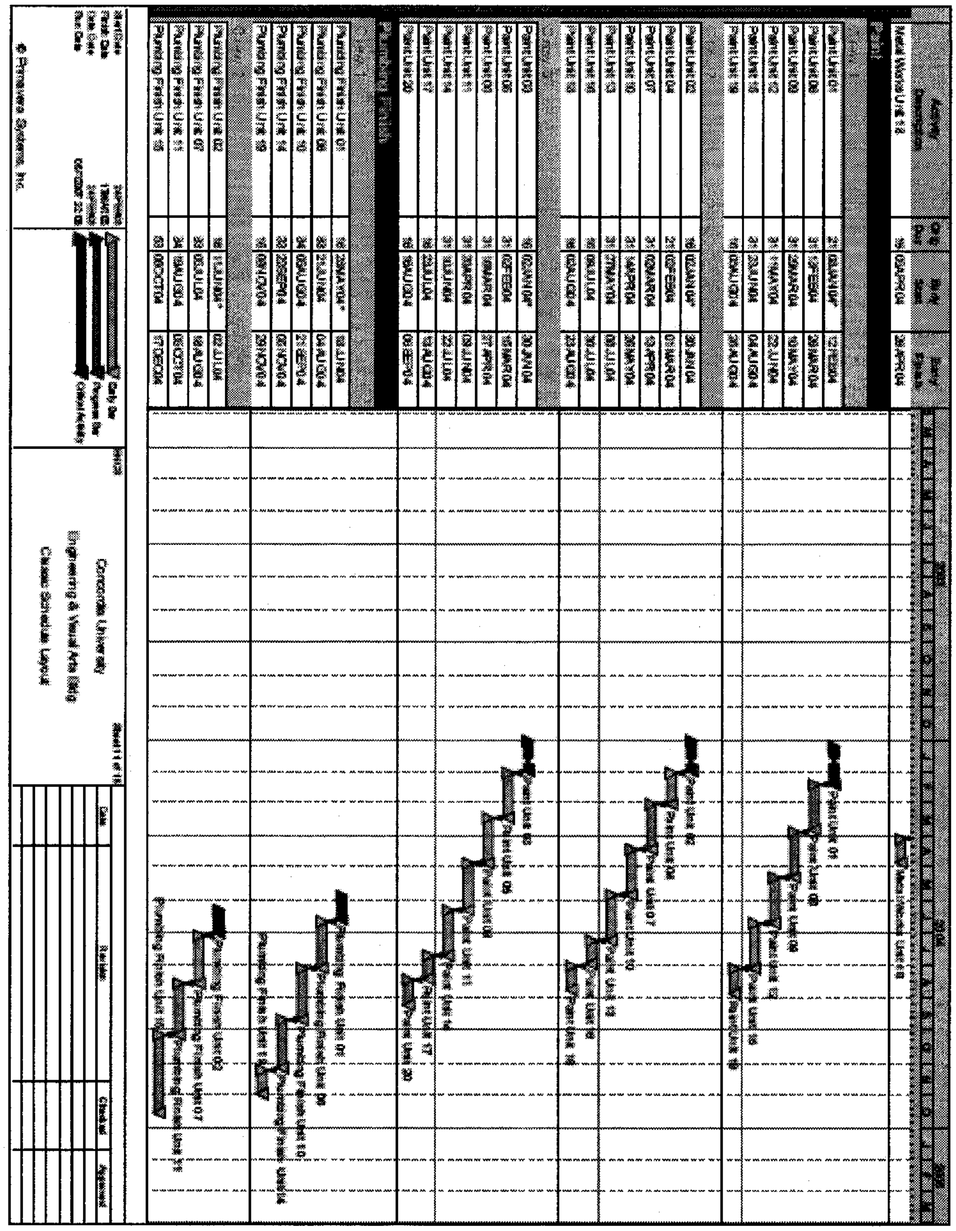


Figure IV - 12: HRPS Generated Schedule in Bar Chart Format Page 12

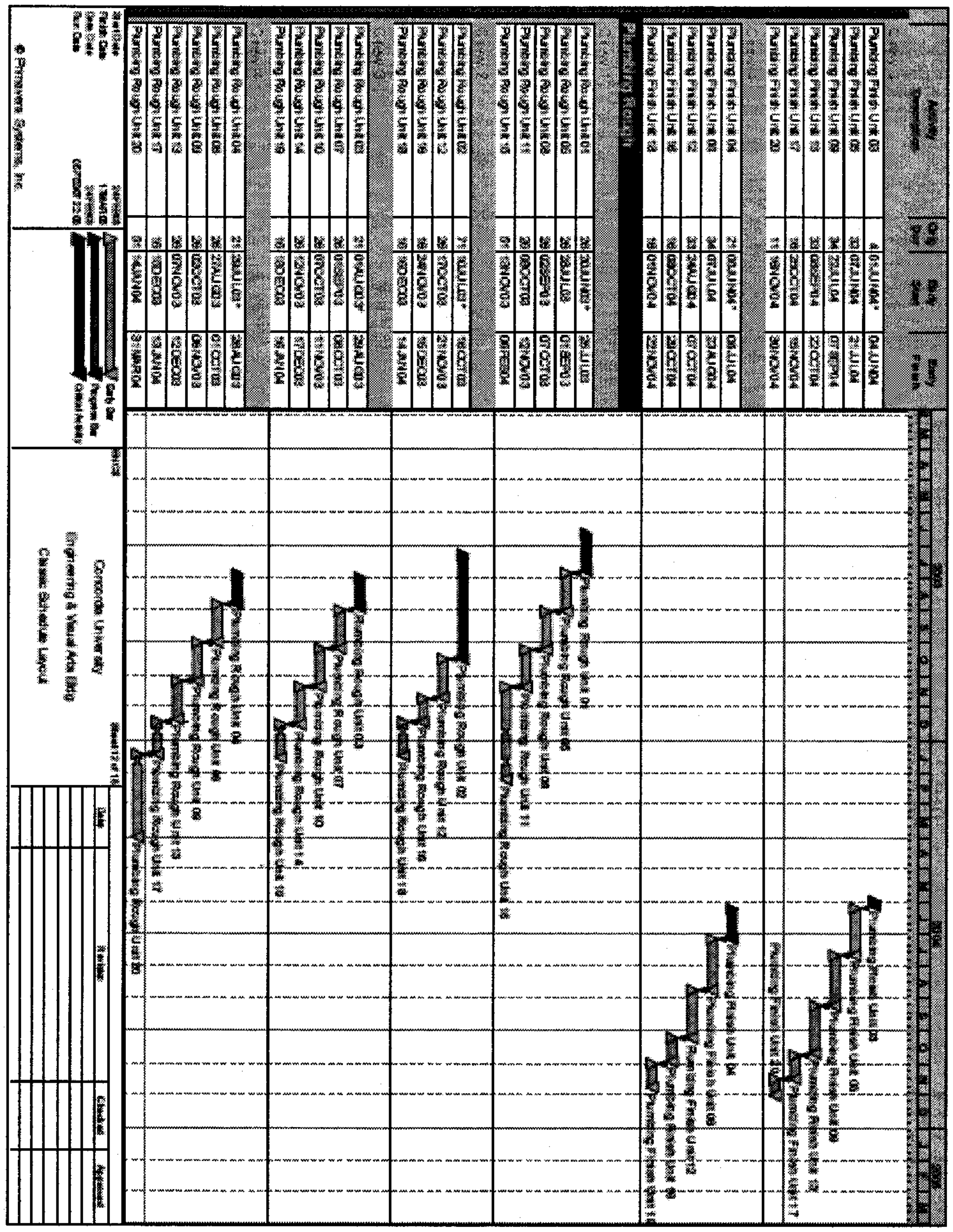


Figure IV - 13: HRPS Generated Schedule in Bar Chart Format Page 13

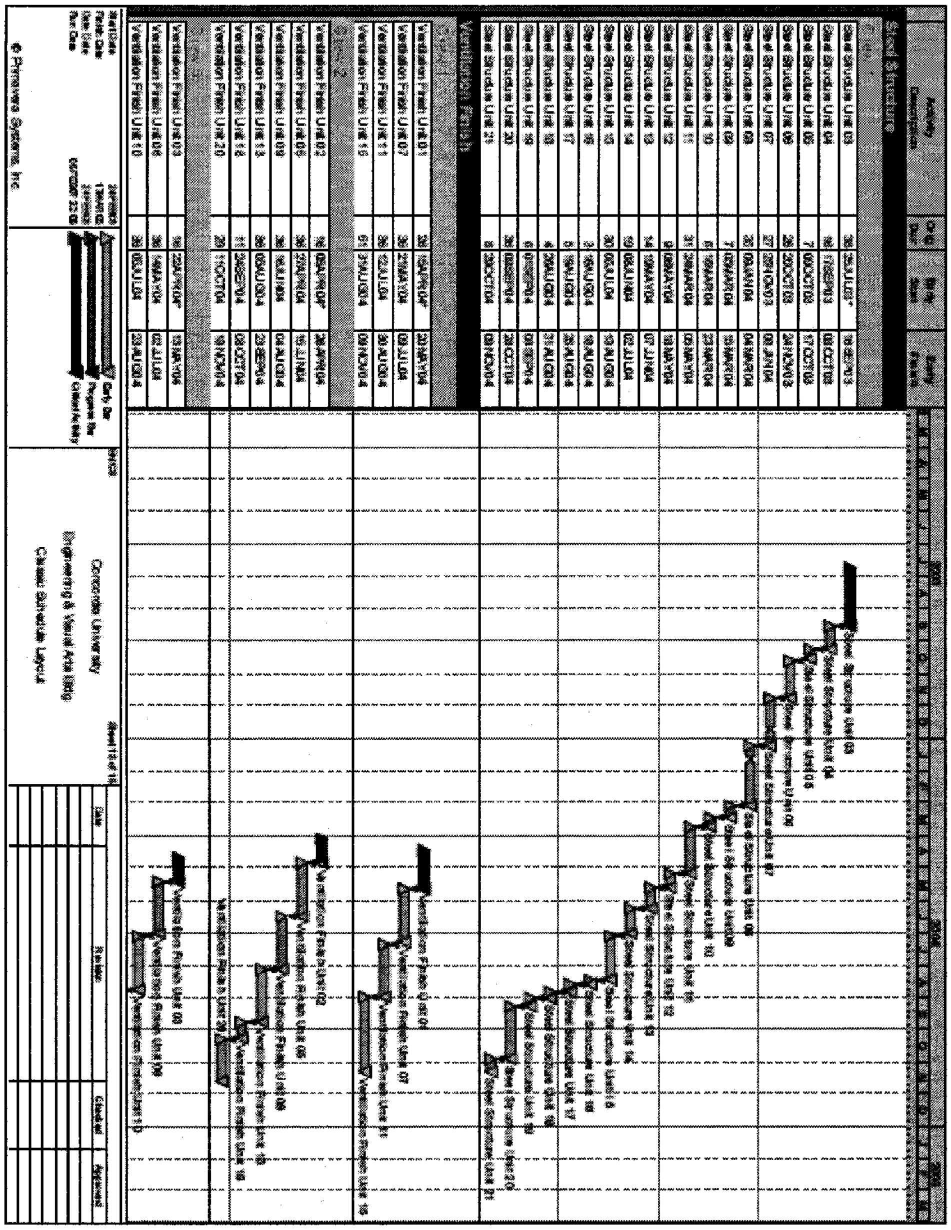


Figure IV - 14: HRPS Generated Schedule in Bar Chart Format Page 14

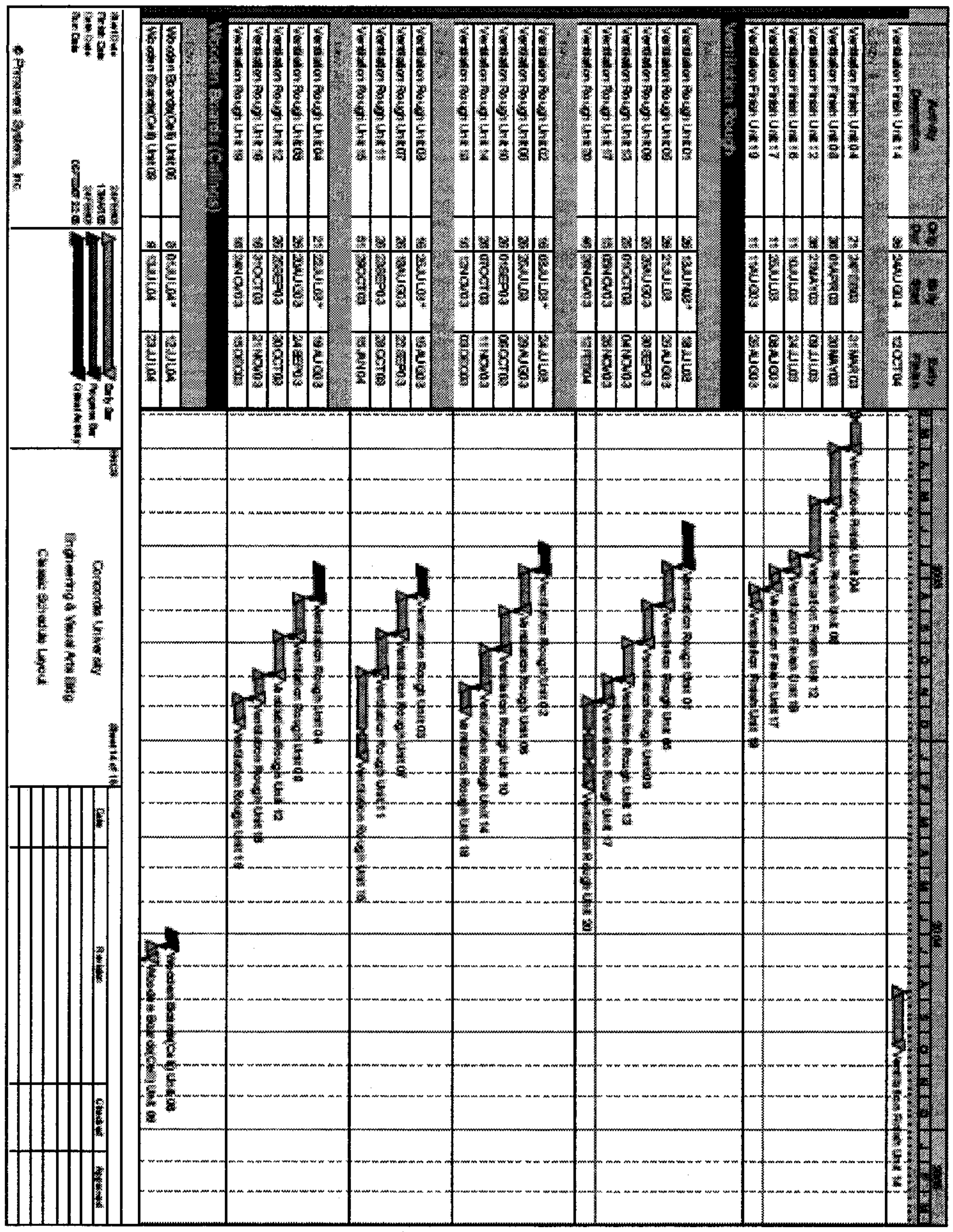


Figure IV - 15: HRPS Generated Schedule in Bar Chart Format Page 15

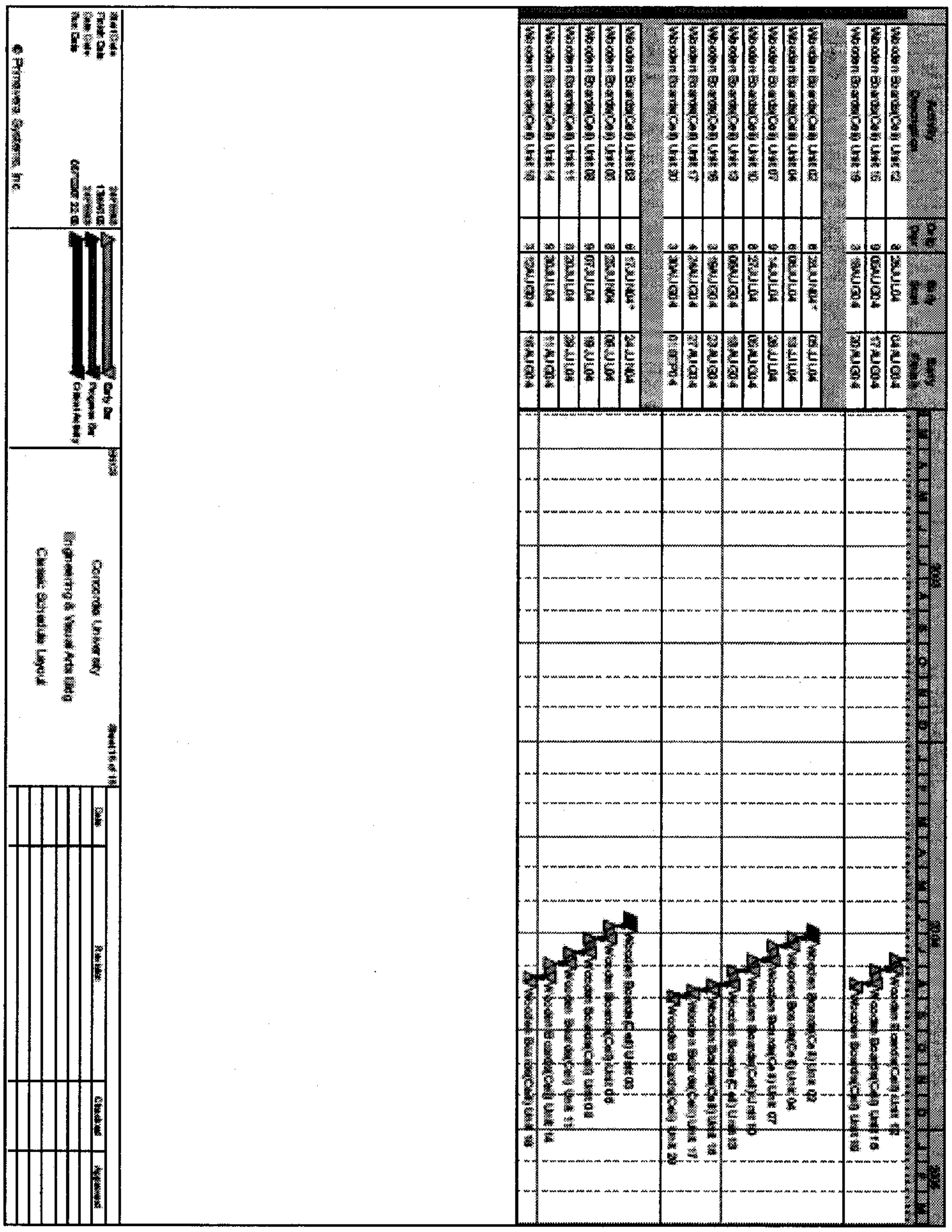


Figure IV - 16: HRPS Generated Schedule in Bar Chart Format Page 16

APPENDIX IV

MICROSOFT PROJECT CONSTRUCTION SCHEDULE

FOR CONCORDIA UNIVERSITY'S EV BUILDING

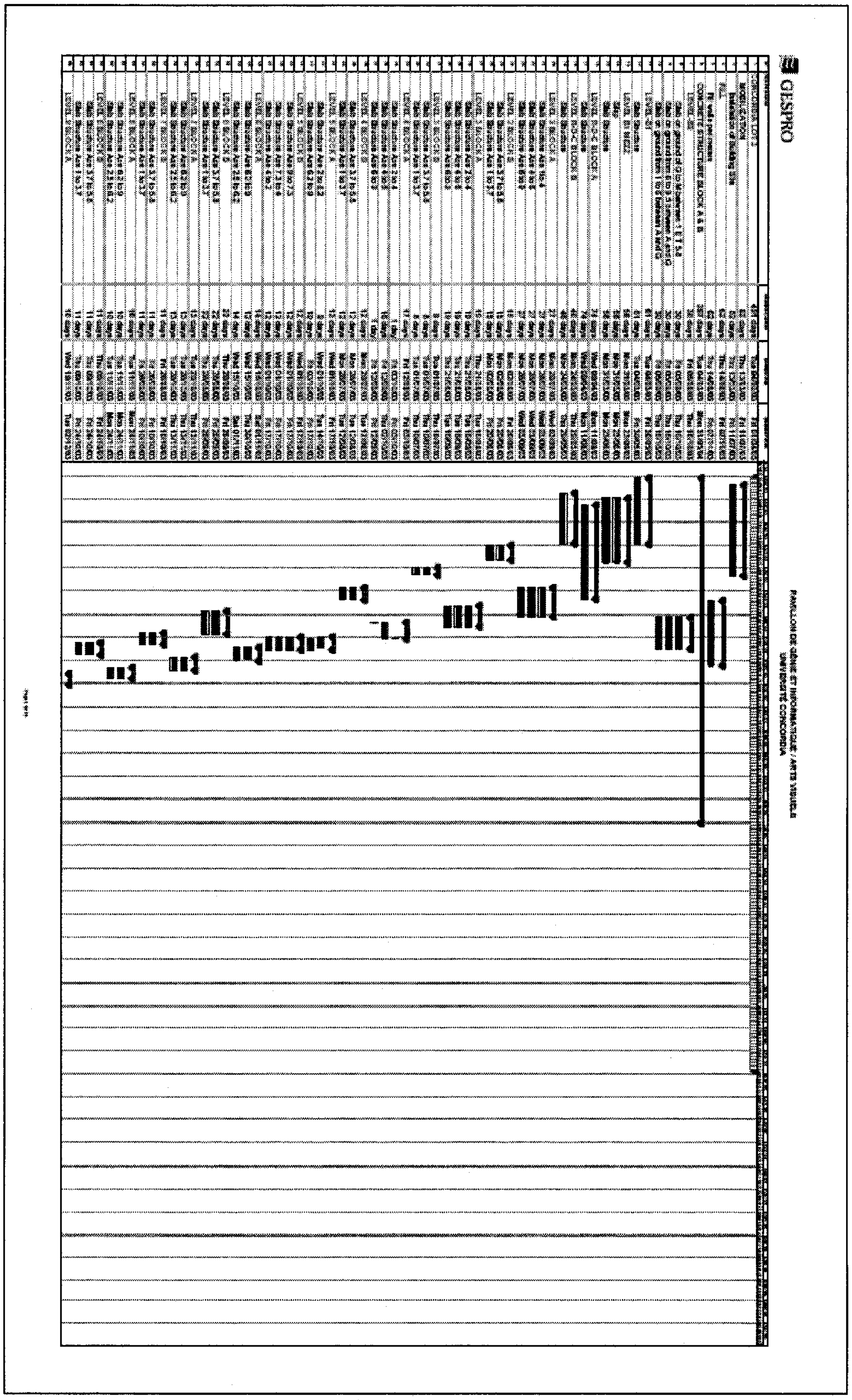


Figure V - 1: Microsoft Project Schedule for the EV Building Page 1

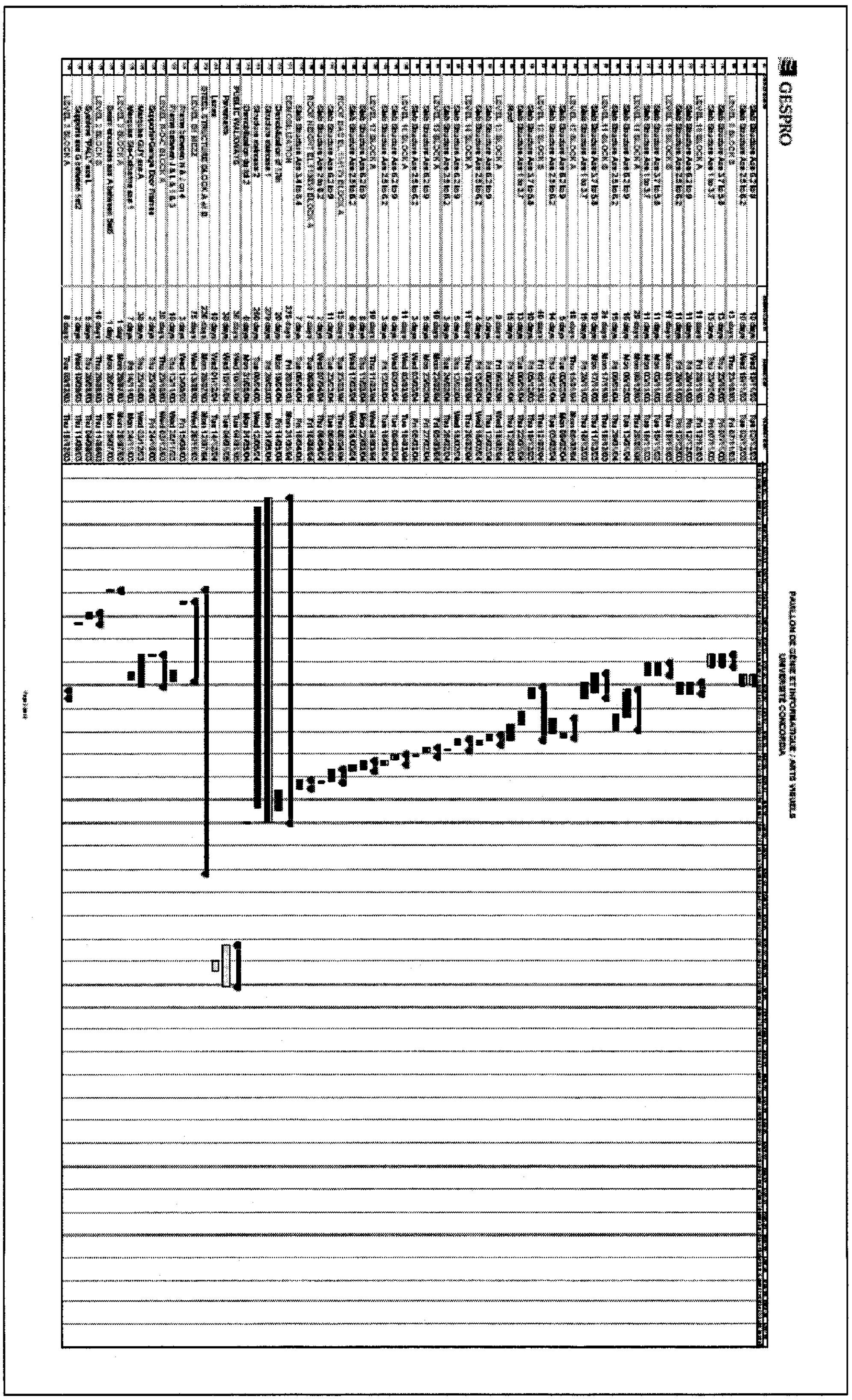


Figure V - 2: Microsoft Project Schedule for the EV Building Page 2

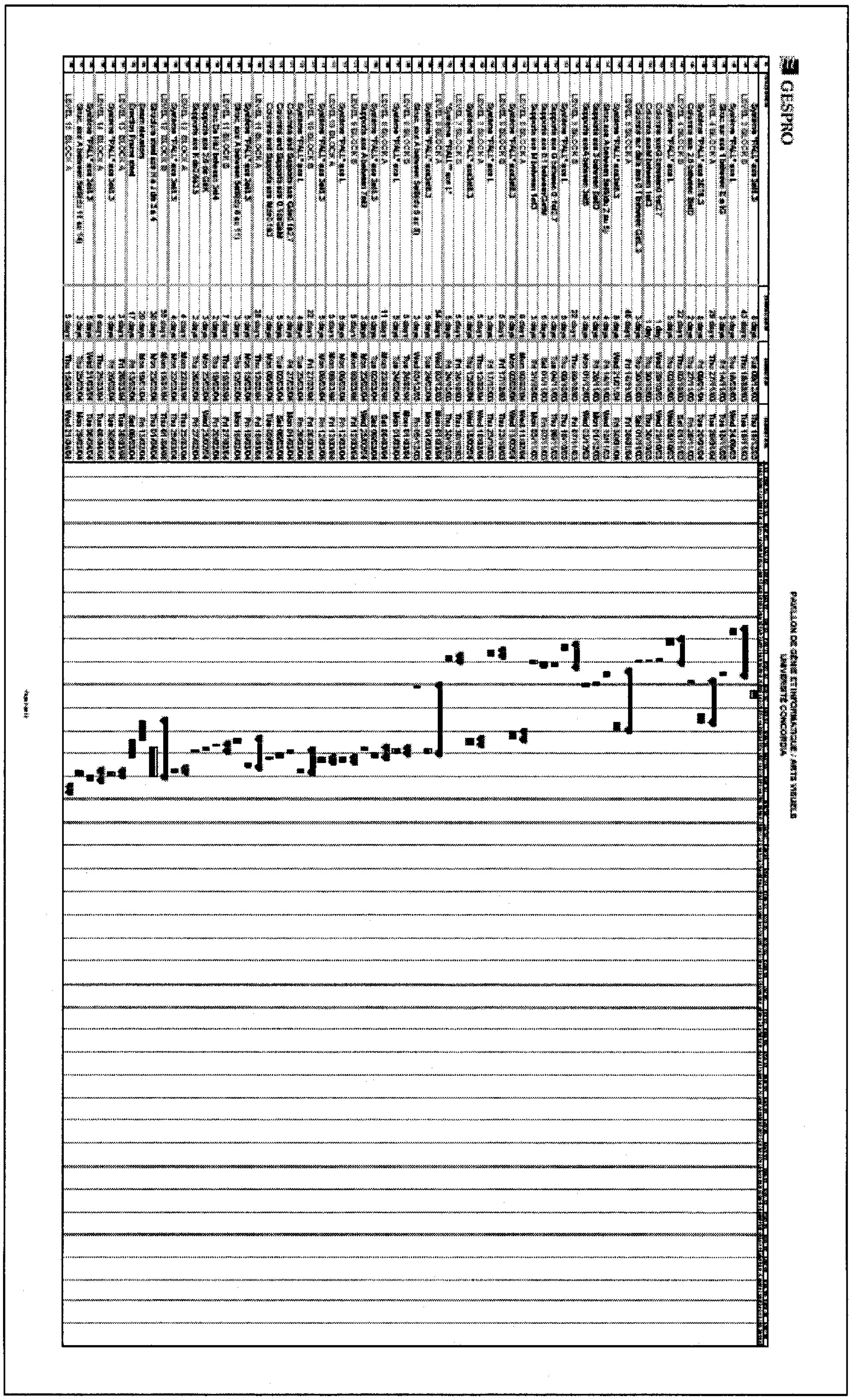


Figure V - 3: Microsoft Project Schedule for the EV Building Page 3

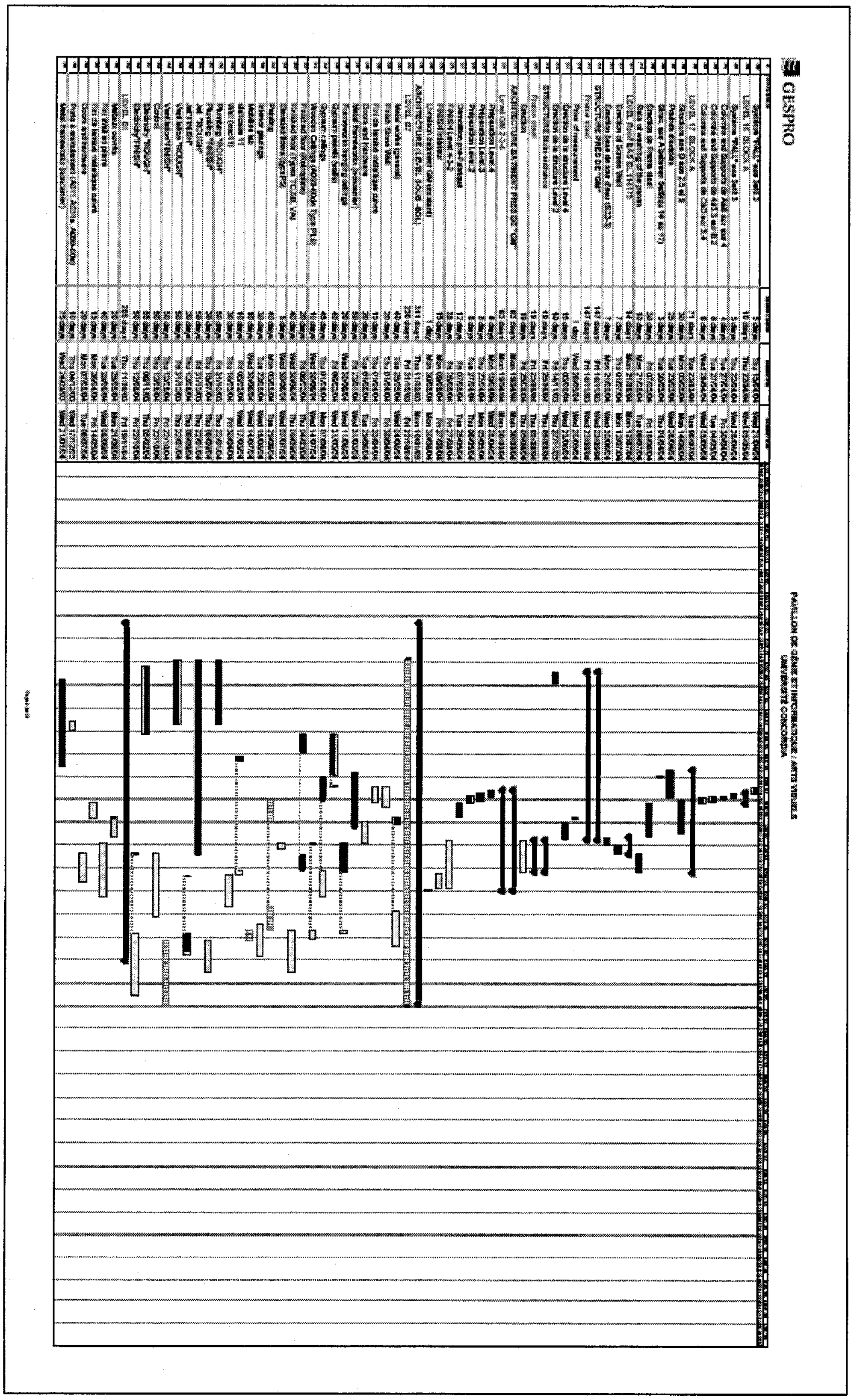


Figure V - 4: Microsoft Project Schedule for the EV Building Page 4

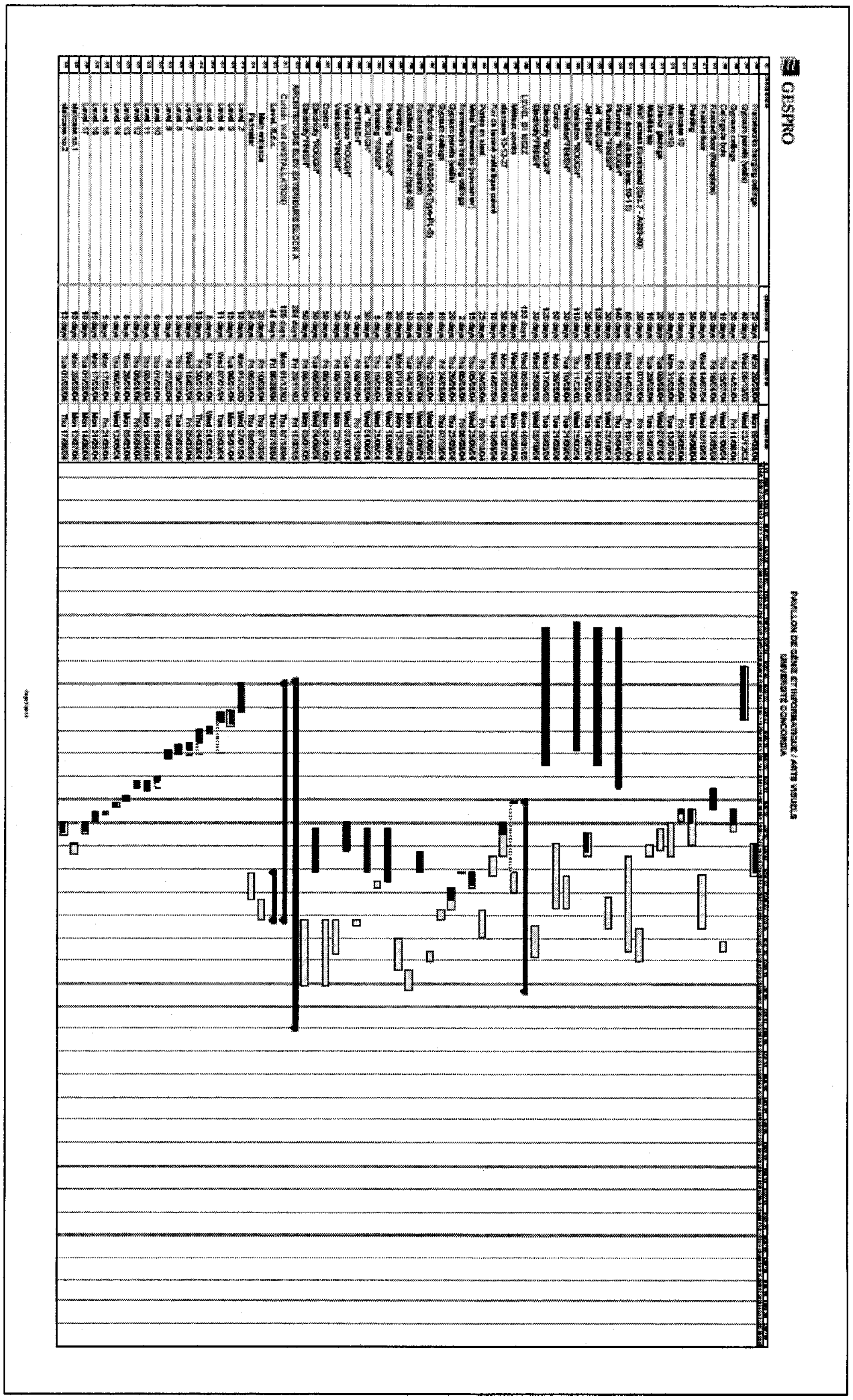


Figure V - 5: Microsoft Project Schedule for the EV Building Page 5

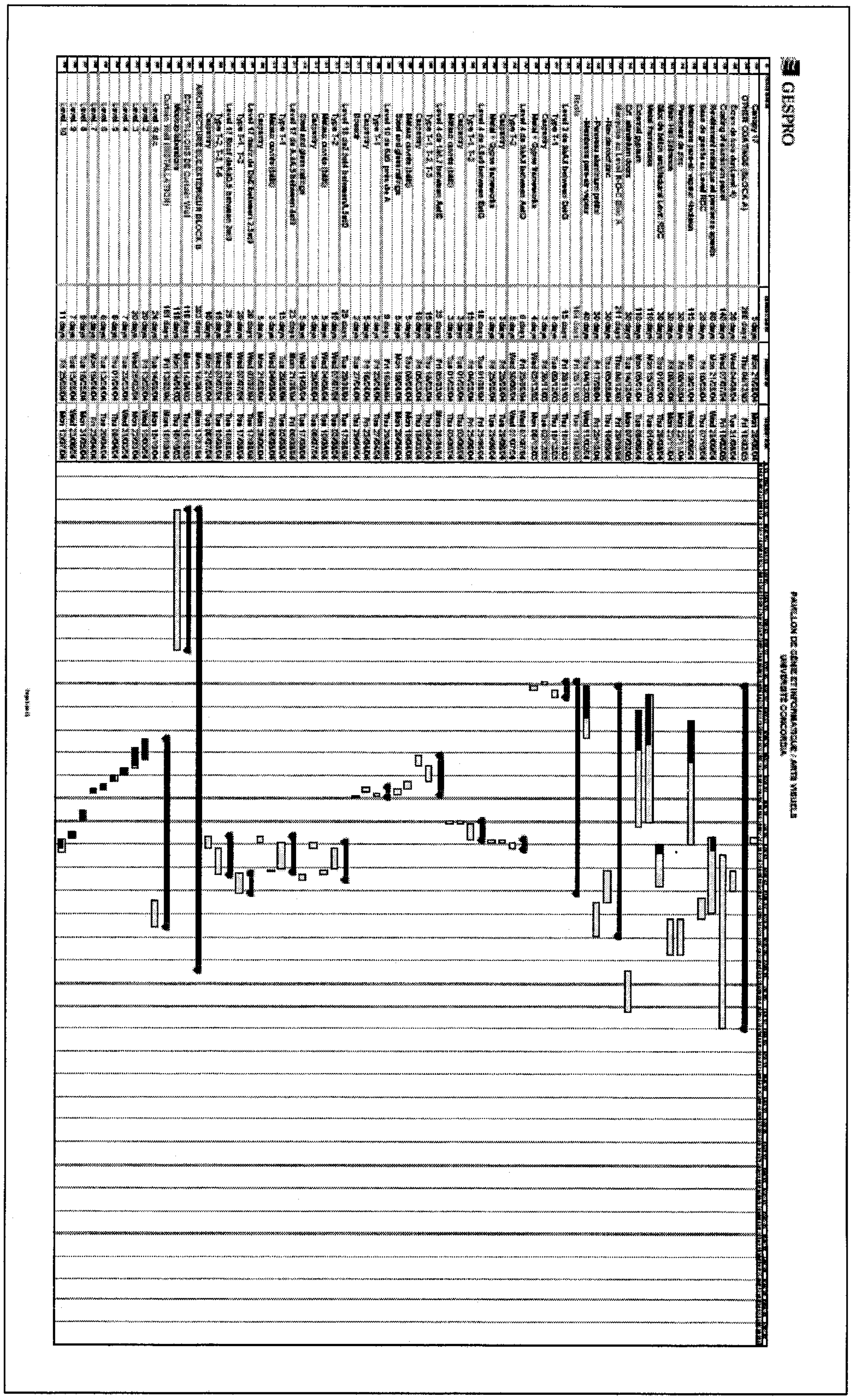


Figure V - 6: Microsoft Project Schedule for the EV Building Page 6

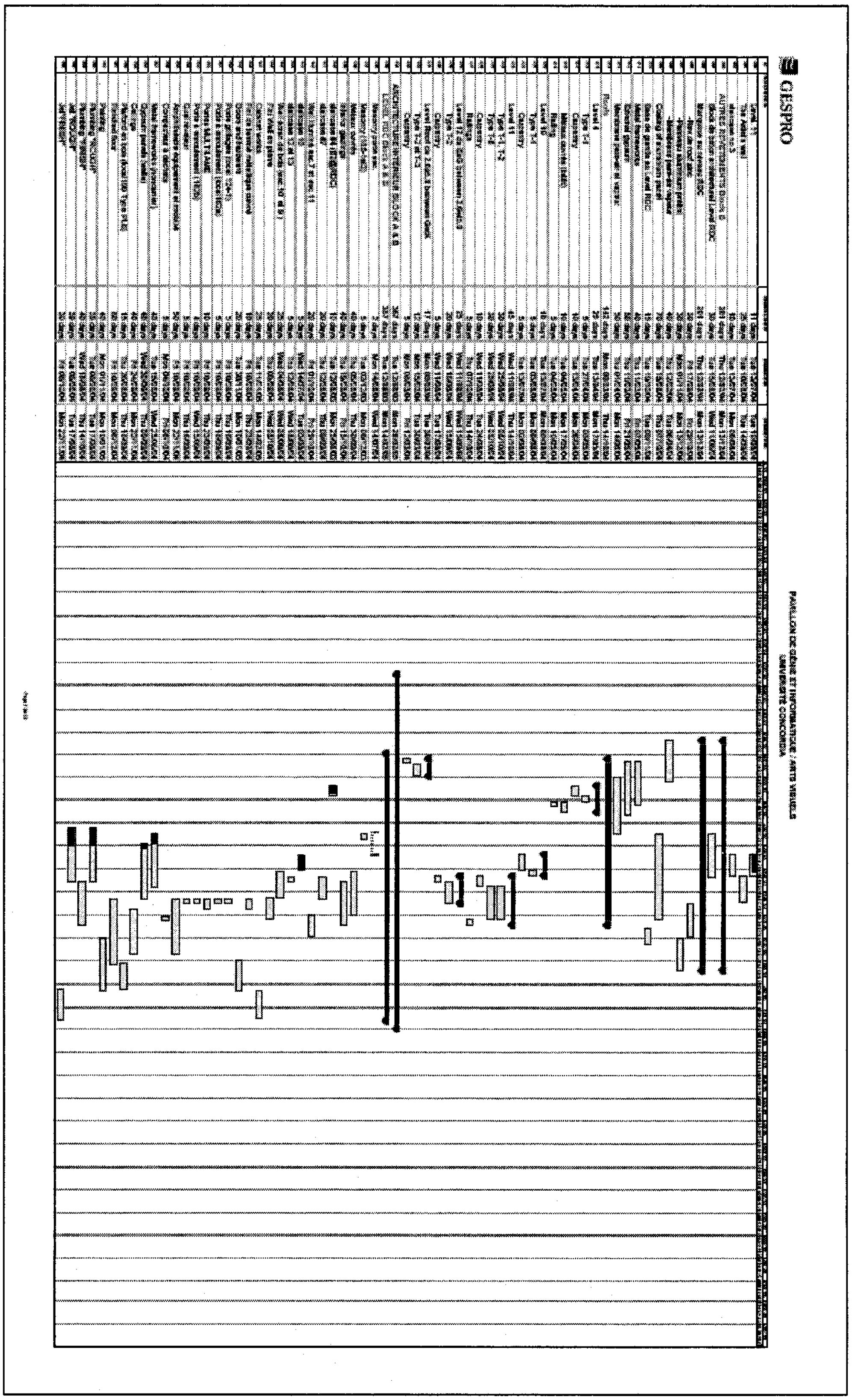


Figure V - 7: Microsoft Project Schedule for the EV Building Page 7

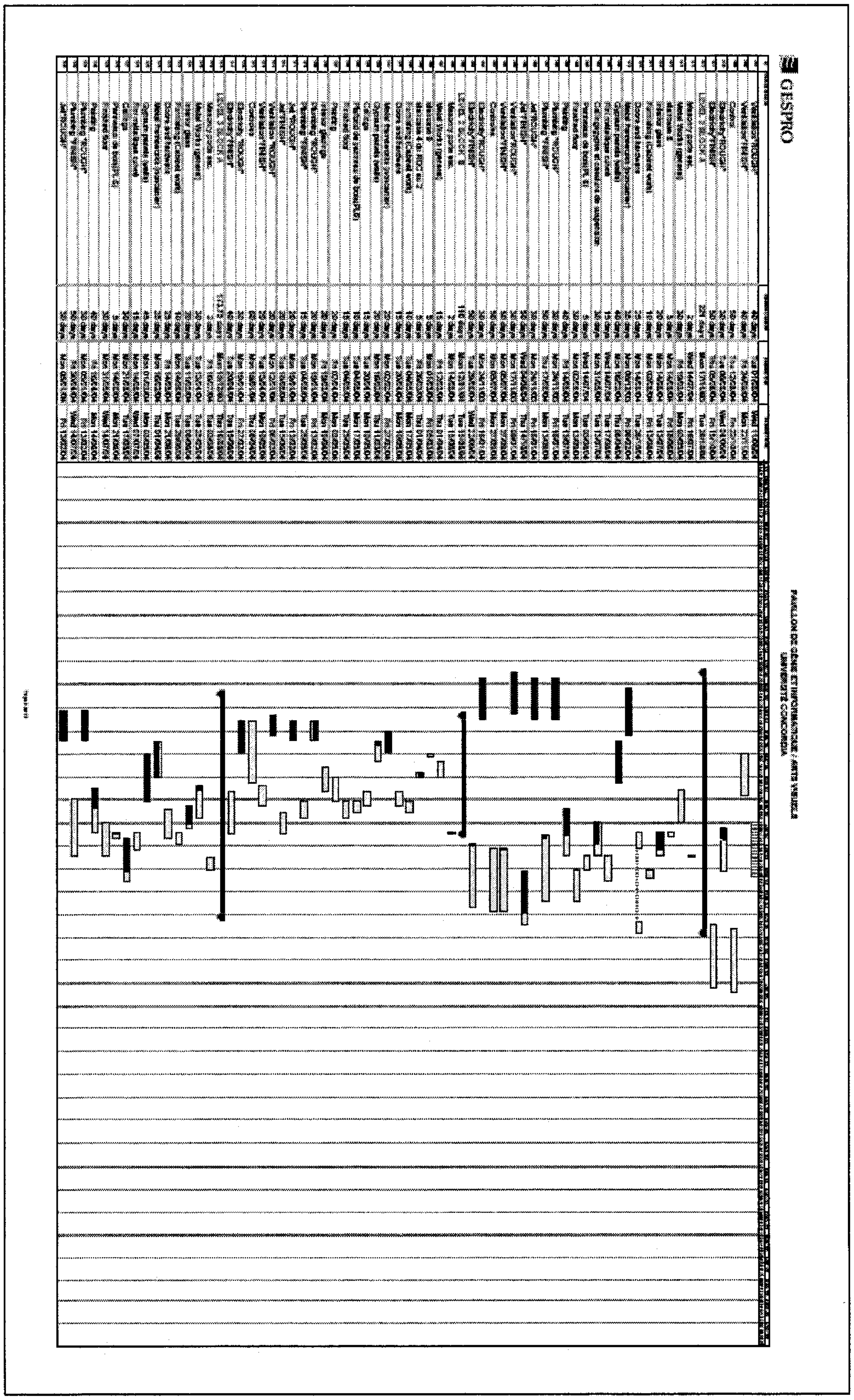


Figure V - 8: Microsoft Project Schedule for the EV Building Page 8

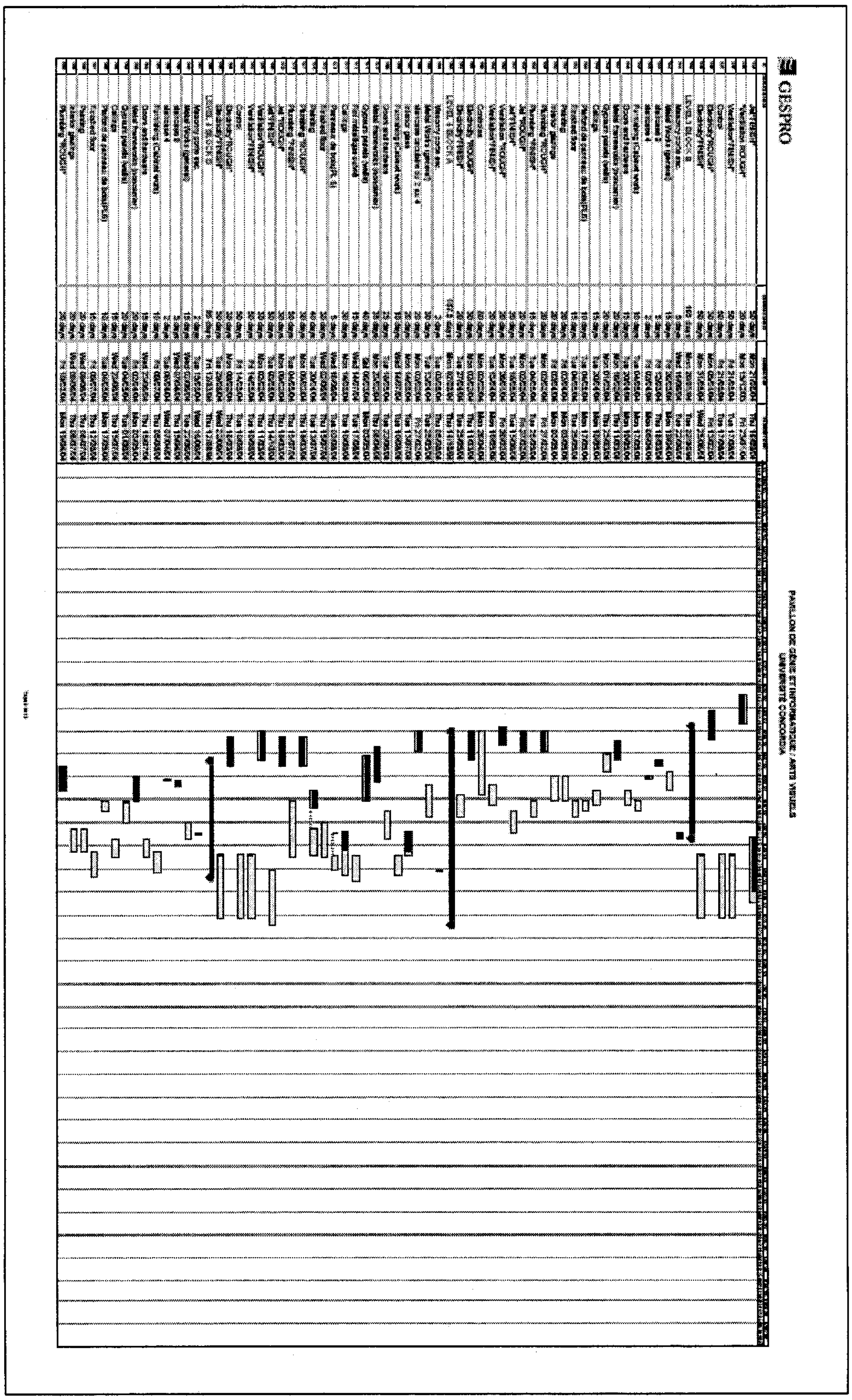


Figure V - 9: Microsoft Project Schedule for the EV Building Page 9

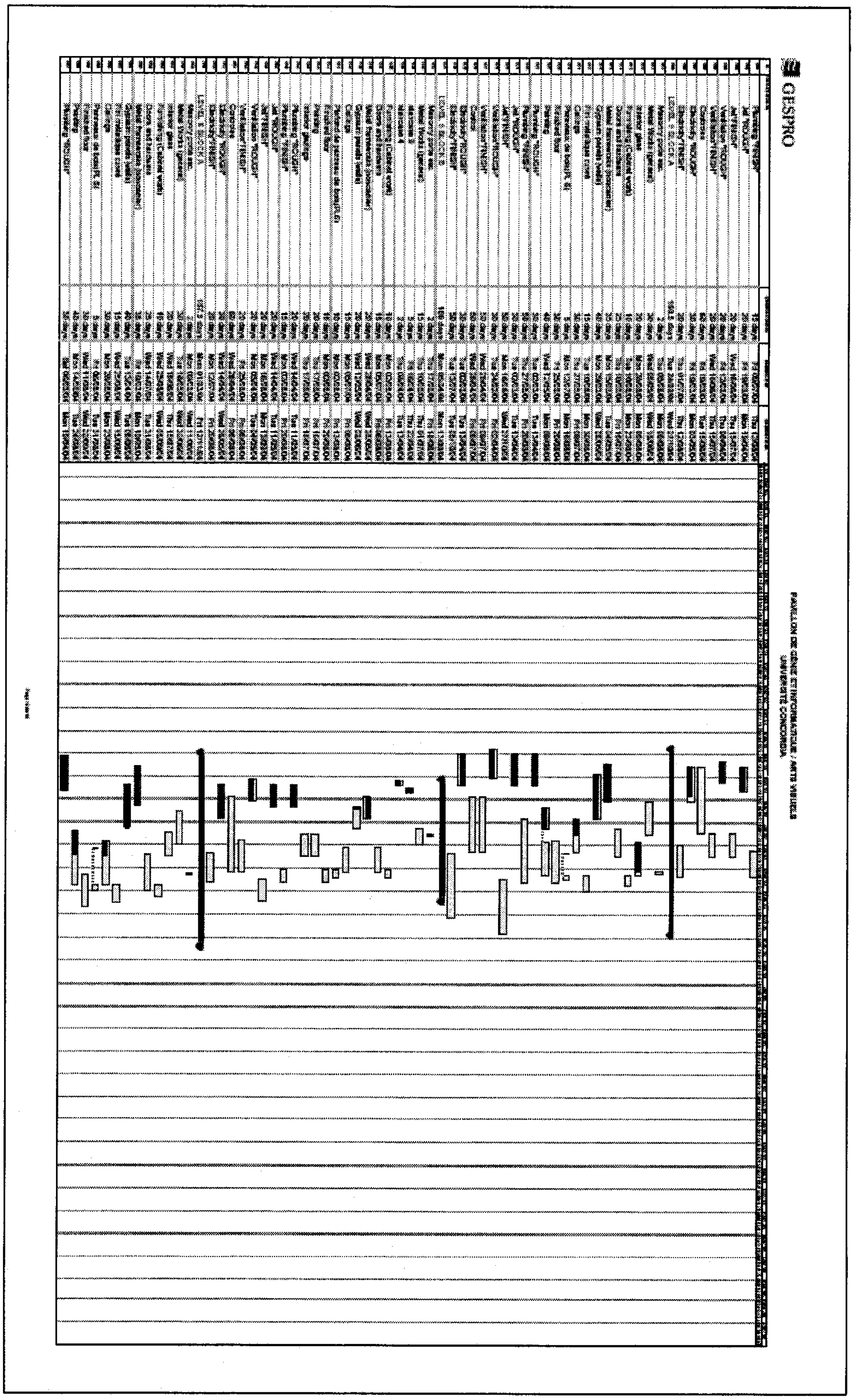


Figure V - 10: Microsoft Project Schedule for the EV Building Page 10

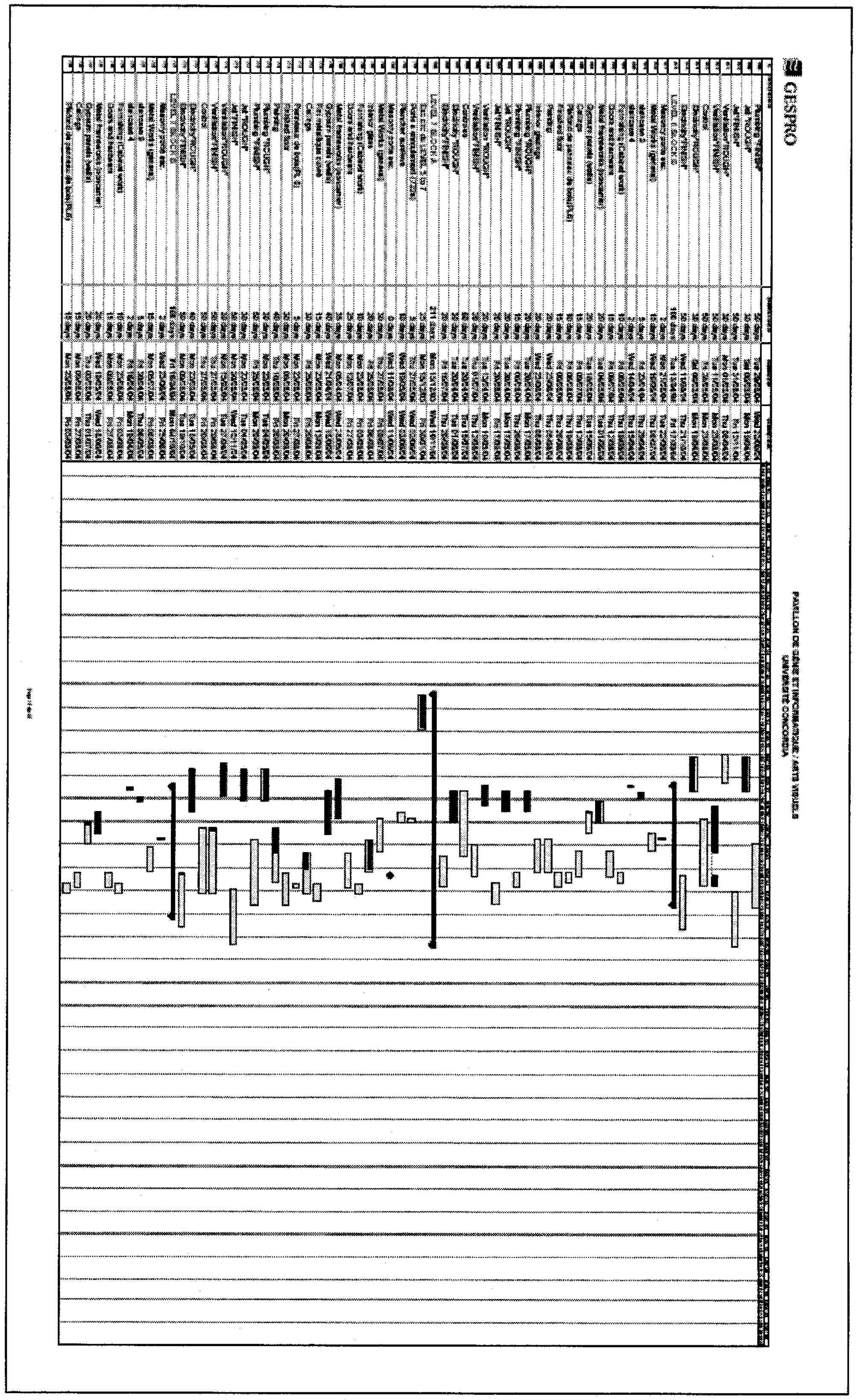


Figure V - 11: Microsoft Project Schedule for the EV Building Page 11

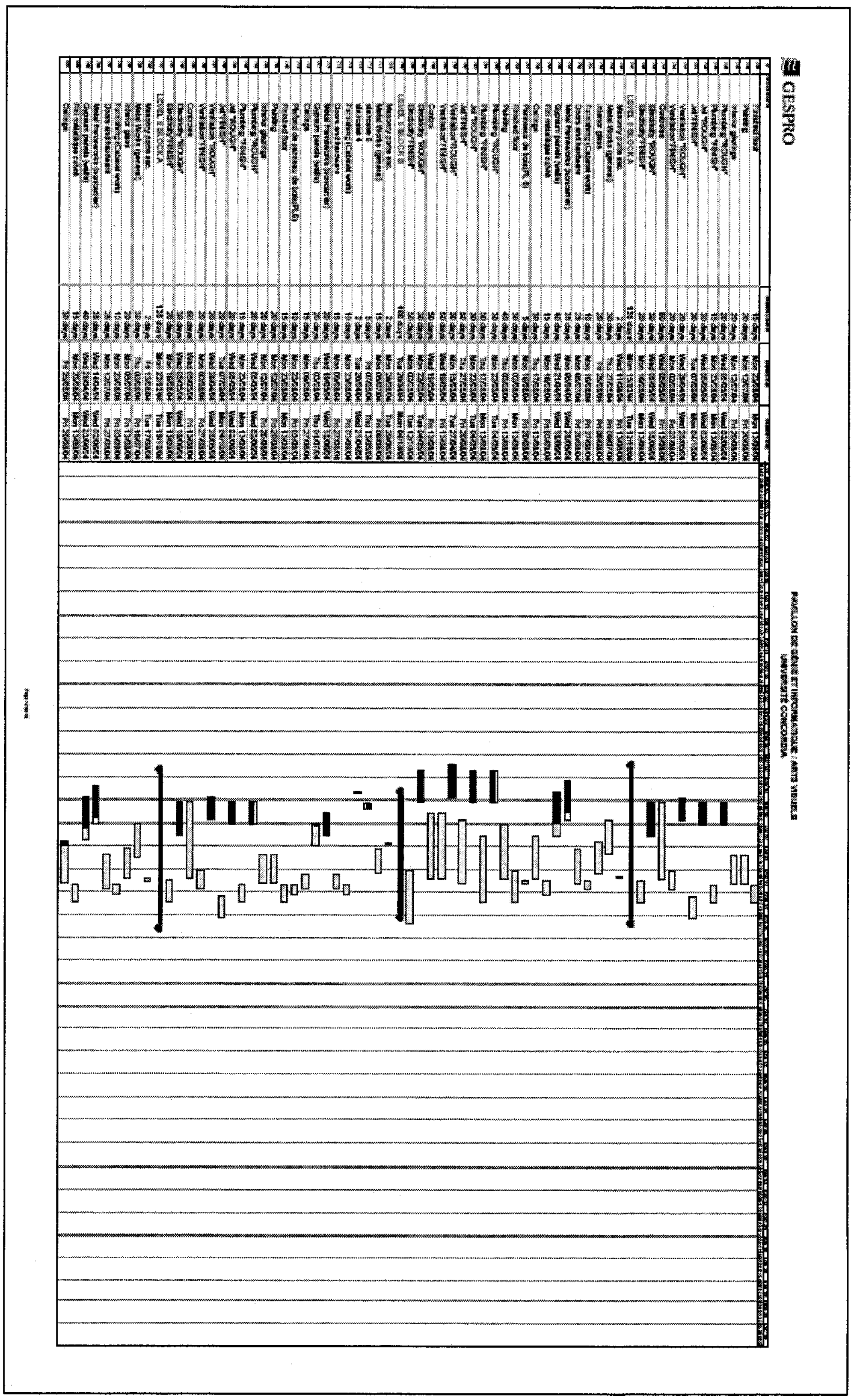


Figure V - 12: Microsoft Project Schedule for the EV Building Page 12

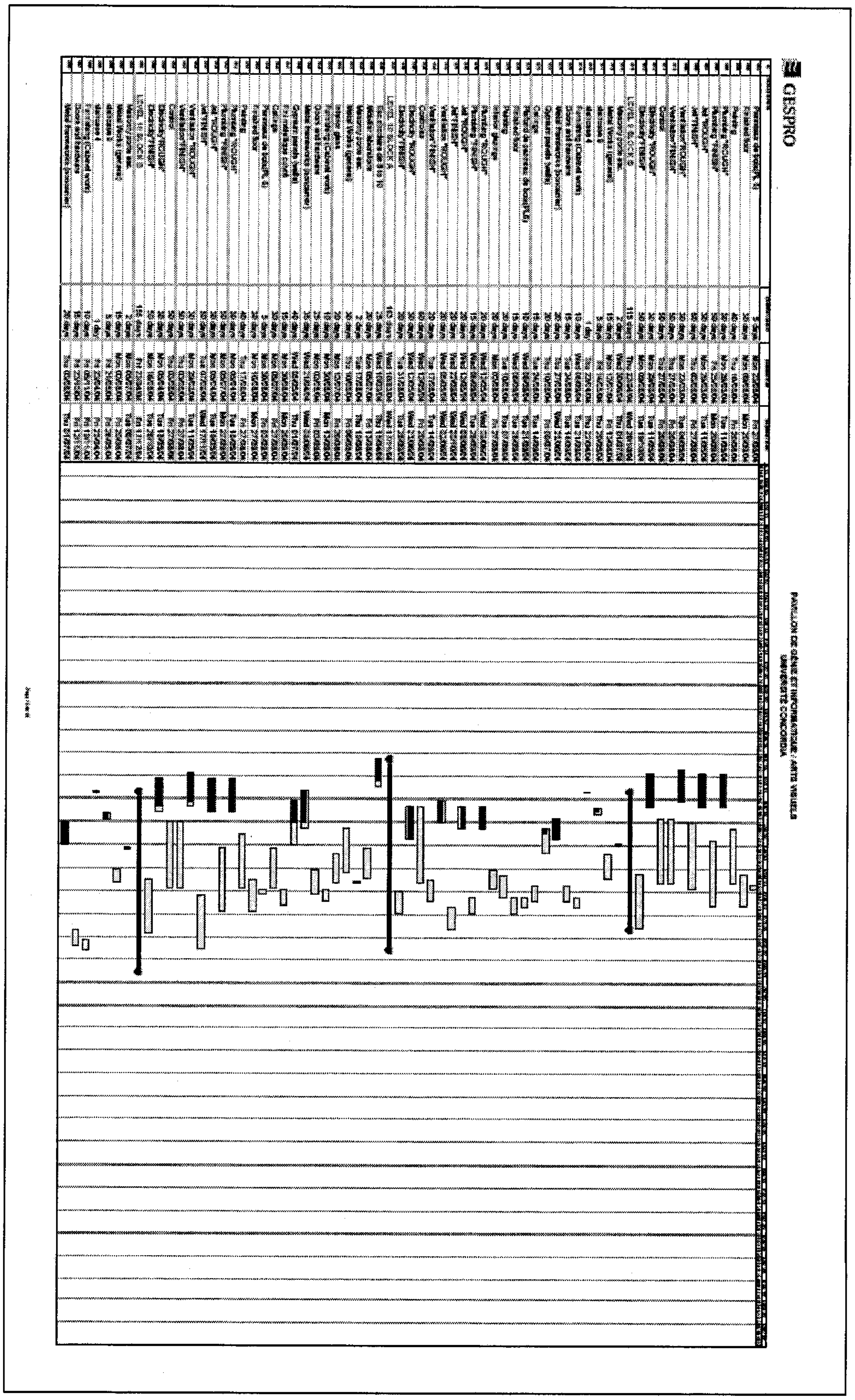


Figure V - 13: Microsoft Project Schedule for the EV Building Page 13

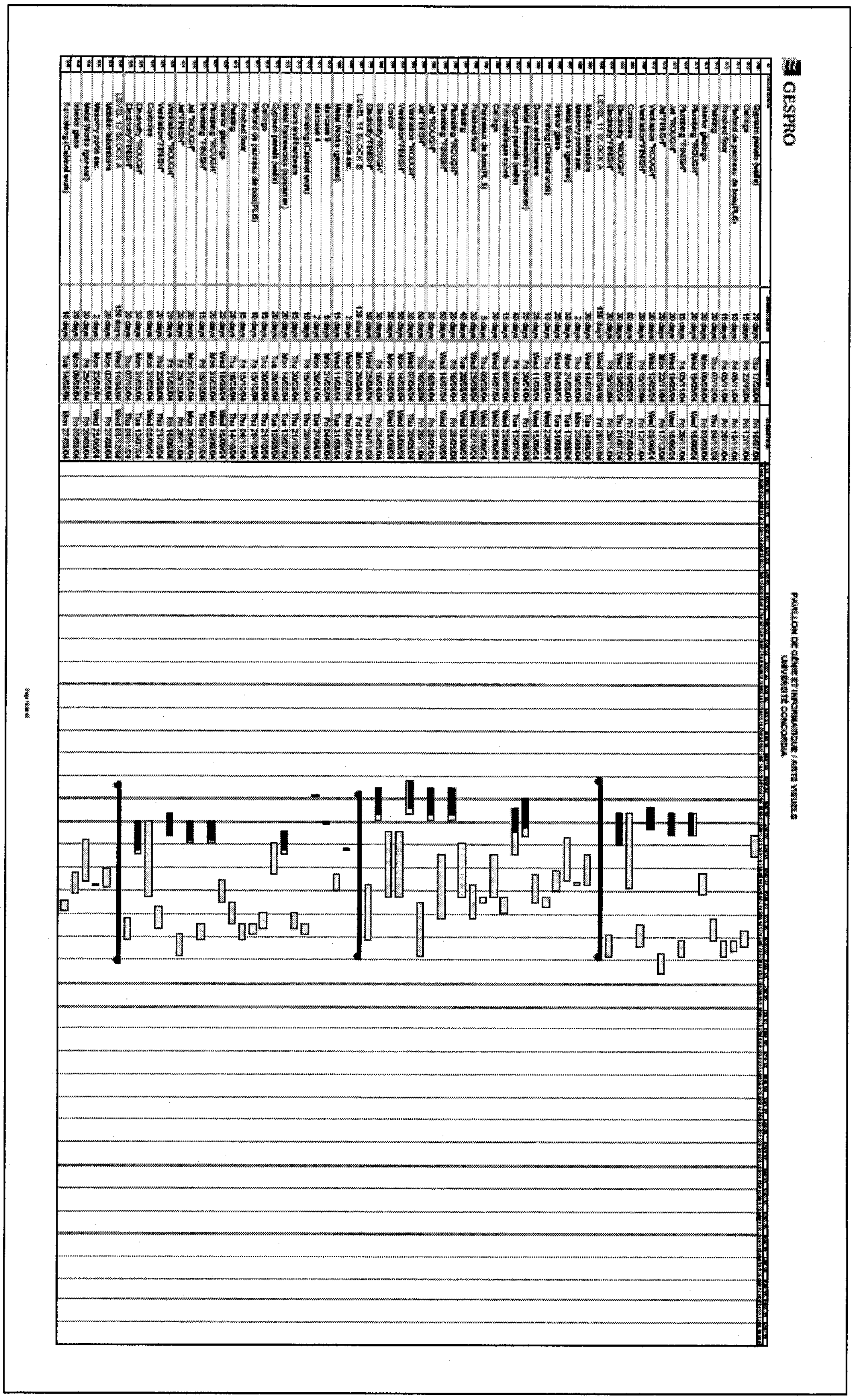


Figure V - 14: Microsoft Project Schedule for the EV Building Page 14

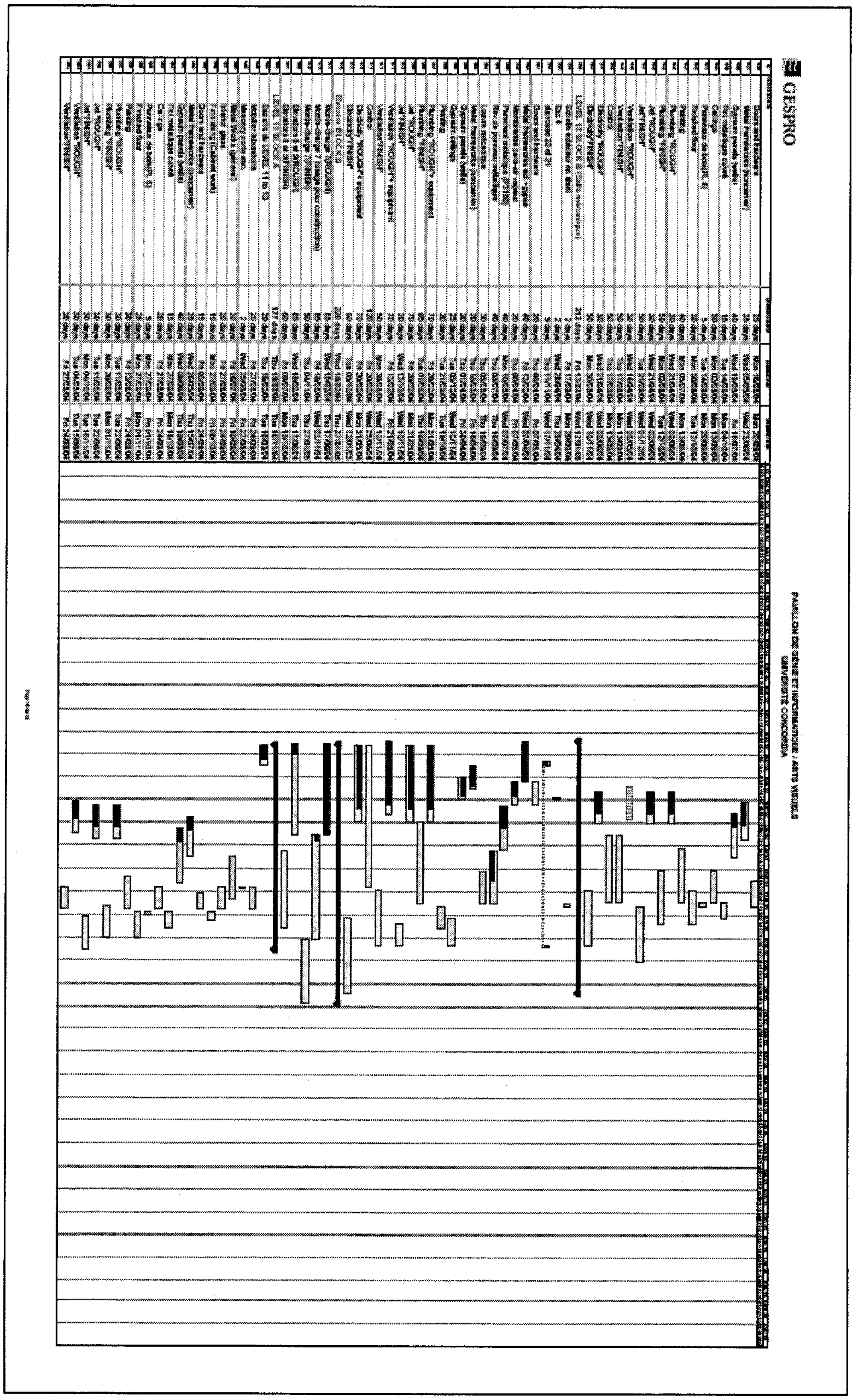


Figure V - 15: Microsoft Project Schedule for the EV Building Page 15

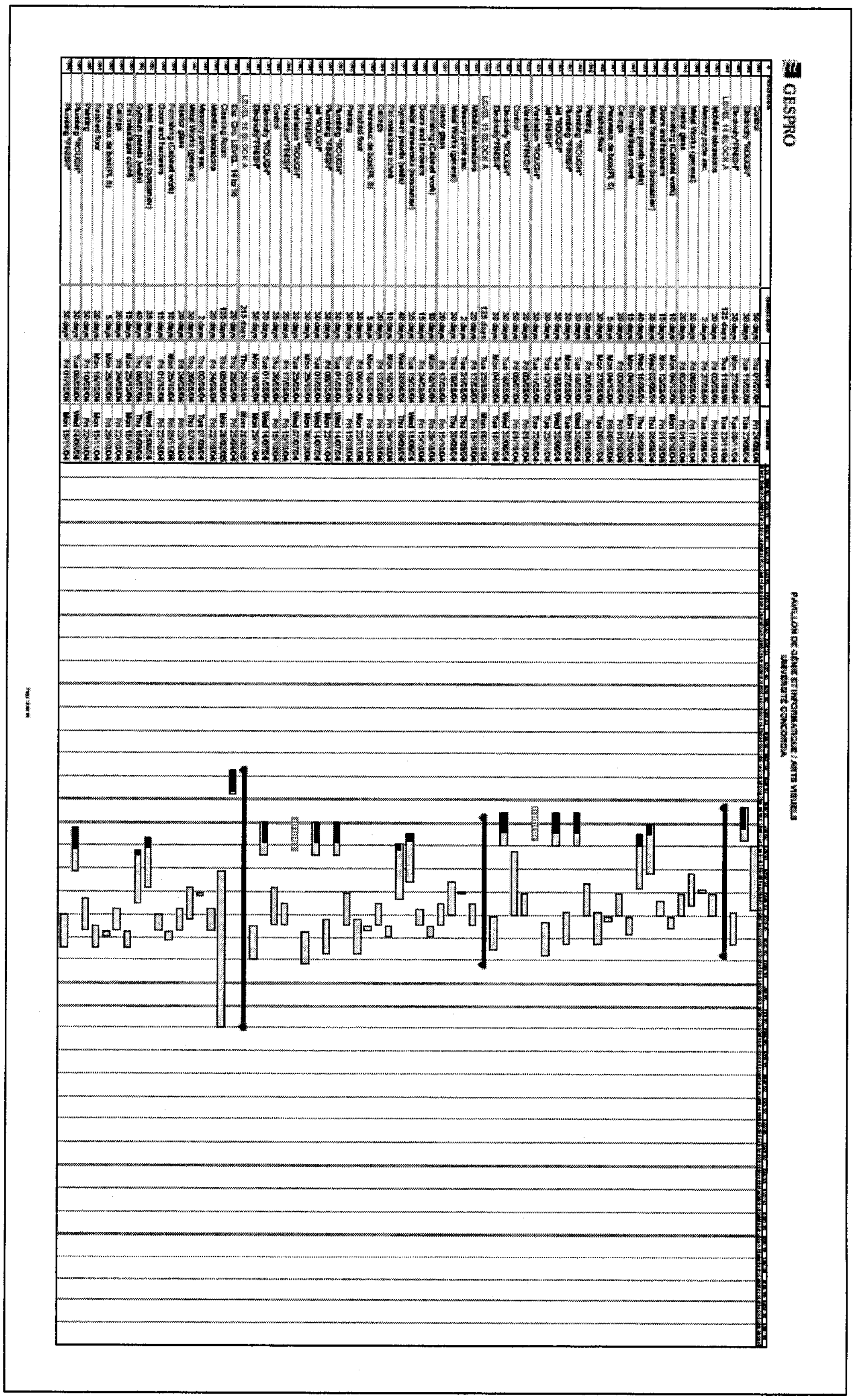


Figure V - 16: Microsoft Project Schedule for the EV Building Page 16

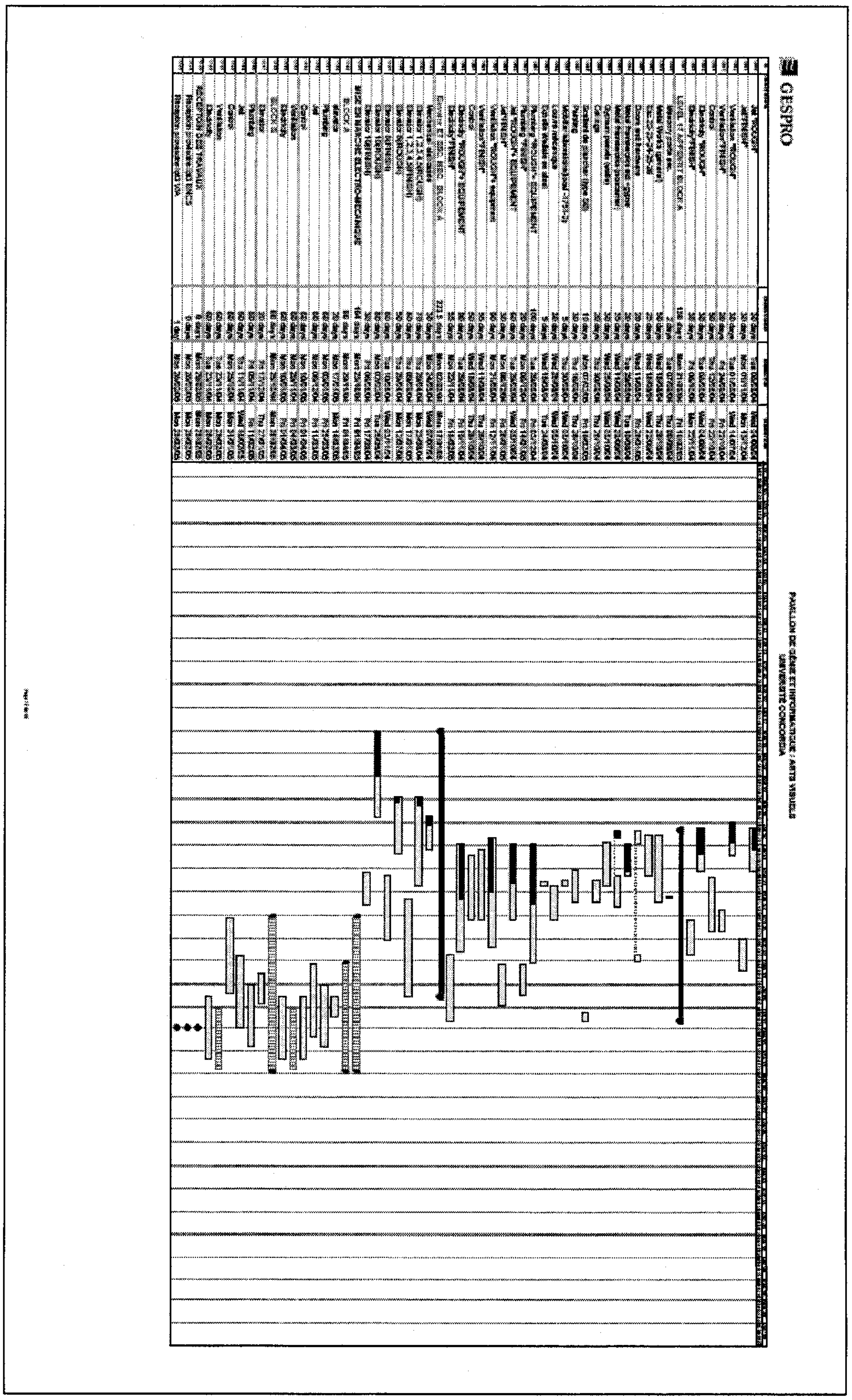


Figure V - 17: Microsoft Project Schedule for the EV Building Page 17