MODELLING ISSUES IN REPETITIVE CONSTRUCTION AND AN APPROACH TO SCHEDULE UPDATING

by

RHYS E. WILLIAMS

B.A.Sc., University of British Columbia, 1983

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
Department of Civil Engineering

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
July 1985

© Rhys E. Williams, 1985
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Civil Engineering

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date: July 12, 1985
ABSTRACT

Planning and control of time and other resources are crucial to the construction of large projects. Yet, current computerized techniques are unable to model the work patterns by which construction personnel plan a project. Furthermore, these methods are not capable of reflecting the day to day changes which must be monitored to control the construction site.

The purpose of this thesis is to promote the useability of computerized planning and scheduling through the development of the heuristic manner by which construction personnel perceive the project. Site studies held in cooperation with Poole Construction Limited and Foundation Company of Canada were performed using a computer scheduling system at the University of British Columbia which contained a prototype model of repetitive work. It provided insight to the process of repetition and rhythm by which projects are planned and to the requirements of the updating process necessary to monitor, and hence control the project.

Two models evolved. The definition of the general repetitive structure was formulated to provide construction personnel with a tool by which to model the process of repetition. The definition of an updating process was formulated capable of monitoring daily progress on a construction site. Work performed with these models have shown them to be realistic in their approach to construction management.
ACKNOWLEDGEMENT

The research contained in this thesis was made possible through funding by the Natural Sciences and Engineering Research Council of Canada (NSERC).

I wish to express my gratitude to Dr. Alan D. Russell, my supervisor, for his guidance and advice throughout the study. His effort in reviewing the thesis and giving suggestions to improve its content are greatly appreciated.

I would like to thank Bob French, Project Engineer for Poole Construction Limited, Frank Smith and David Bellamy, Site Supervisor and Project Manager respectively for Foundation Company of Canada for their cooperation and time given throughout the research.

Finally, I wish to thank my family and friends for their encouragement and support.

I am especially grateful to Jennifer Anne Cook, my friend, for her time and effort given throughout the thesis.
TABLE OF CONTENTS

ABSTRACT ....................................................................................... ii
ACKNOWLEDGEMENT ..................................................................... iii
TABLE OF CONTENTS .................................................................... iv
LIST OF FIGURES .......................................................................... vi
LIST OF TABLES .............................................................................. vii

CHAPTER I - BACKGROUND

1.1 Introduction ................................................................. 1
1.2 Thesis Objectives ....................................................... 3
1.3 Methodology ............................................................... 4
1.4 Thesis Overview ......................................................... 5

CHAPTER II - THE REPETITIVE ACTIVITY

2.1 Objectives ................................................................. 6
2.2 Laramee's Model ......................................................... 9
2.3 Previous Studies ........................................................ 13
2.3.1 The Site Studies .................................................... 16
2.4 Observations ............................................................ 18
2.4.1 A) Flexibility .......................................................... 18
2.4.1.1 User Interface .................................................... 21
2.4.2 B) Logic Relationships ........................................... 22
2.4.2.1 System Loops ..................................................... 24
2.4.3 C) Continuity .......................................................... 26
2.4.4 D) Criticality .......................................................... 31
2.4.4.1 Losses Due to Linear Repetition ...................... 34
2.5 The General Repetitive Structure ................................. 37
2.5.1 A) Data Structure .................................................. 40
2.5.2 B) Flexibility .......................................................... 46
2.5.3 C) Continuity .......................................................... 46
2.6 The Repetitive Building Process ................................. 48
2.6.1 The Flowchart ......................................................... 49
2.7 Summary ................................................................. 54
## CHAPTER III - CONTINUOUS UPDATING

3.1 Introduction ................................................. 56
3.2 Criteria Used to Promote the Use of Updating .......... 57
3.3 Criterion 1: Integration into the System ............... 60
   3.3.1 The Data Base Centre .......................... 61
      3.3.1.1 The Log .................................... 62
   3.3.2 The Time Centre ................................... 62
   3.3.3 The Monetary Centre .............................. 64
   3.3.4 The Decision Making Process .................... 65
   3.3.5 Litigation Support ................................ 66
3.4 Criterion 2: The Updating Process ....................... 69
   3.4.1 Phase A ........................................... 69
   3.4.2 Phase B ........................................... 71
   3.4.3 Schedule Interaction ............................. 72
3.5 Criterion 3: Automation of the Updating Process ....... 74
3.6 The Continuous Updating Algorithm ..................... 76
   3.6.1 The Activity Window .............................. 77
   3.6.2 The Activity Durations ............................ 78
   3.6.3 The Activity Starts and Finishes ................. 79
   3.6.4 Substantial Completion vs Total Completion ....... 80
   3.6.5 Procurement ....................................... 81
   3.6.6 The Hierarchical Structure of Links .............. 82
   3.6.7 The Validity of Logic Links ....................... 84
   3.6.8 The Completed Activity and its Logic Links ....... 86
   3.6.9 The Scheduled Activity and its Logic Links ...... 86
   3.6.10 The Ongoing Activity and its Logic Links ....... 90
   3.6.11 Summary ......................................... 92
3.7 Updating the Repetitive Structure ....................... 93
   3.7.1 The Update Block .................................. 94
   3.7.2 Update Convention for the Repetitive Activity .... 96
3.8 The Computer Model ....................................... 98
3.9 Conclusion .............................................. 99

## CHAPTER IV - FURTHER ISSUES AND CONCLUSION

4.1 Introduction .............................................. 100
4.2 The Generality of Repetitive Work ..................... 100
4.3 Computer Implementation and the Construction Industry 102
   4.3.1 User Interface .................................... 104
4.4 Computers and Data Storage ............................. 105
4.5 Expert Systems .......................................... 107
4.6 Recommendations for Future Work ...................... 108
4.7 Conclusion .............................................. 110

BIBLIOGRAPHY ................................................. 112
LIST OF FIGURES

2.1 Laramee's Repetitive Structure ......................... 11
2.2 Repetitive Structure with Multiple Crews ................ 11
2.3 The Typical Link ............................................. 12
2.4 Non-Typical Links ........................................... 13
2.5 Finished Schedule ............................................ 14
2.6 Modelling Changes in Number of Crews by Varying Production Rate ................. 19
2.7 Examples of Graphic Output of the Repetitive Structure ....................... 22
2.8 Resequencing and Non-Typical Logic Constraints .................. 23
2.9 Resequencing and Typical Logic Constraints .................... 24
2.10 A Logic Loop Involving a Repetitive Activity ................. 25
2.11 Continuity of Internal Rhythm ................................ 28
2.12 Continuity of External Rhythm ................................ 29
2.13 Continuity Imposing Criticality ................................ 32
2.14 Continuity and Its Use of Float ................................ 33
2.15 Worst and Best Case Models of Linear Repetition ................. 35
2.16 Examples Based on the General Repetitive Activity Definition .................. 42
2.17 The Sequential File INPUT .................................. 43
2.18 Tradeoff Between Generality of Repetitive Activity Definition and Efficiency of Data Input .......... 45
2.19 The Use of Pools in the Simulation Process .................... 50
2.20 The Sequential File OUTPUT .................................. 51
2.21 Flowchart of the Repetitive Activity Building Process .................. 52
2.22 Finished Model of the Repetitive Activity ...................... 55

3.1 Simplified Construction Management Information System ....................... 60
3.2 The Updating Process Flowchart ................................ 70
3.3 The Activity Window and Its Logic Links ......................... 77
3.4 Three Types of Activities and Their Durations .................... 78
3.5 Start and Finish Definitions and Their Time Zones ............... 79
3.6 Example of Constraint Modeled as a Logic Link .................... 84
3.7 Forward and Backward Pass of a Scheduled Activity ............... 87
3.8 Invalidation of Successor Links with an Actual Start Date ............. 88
3.9 Invalidation of Successor Links with an Actual Finish Date ............ 89
3.10 Forward and Backward Pass of the Ongoing Activity ................ 91
3.11 The Repetitive Activity Update Flowchart ......................... 95
3.12 Example of an Updated Repetitive Activity ...................... 98
LIST OF TABLES

3.1 The Four Sets of Data Required for Updating .............. 82

3.2 The Hierarchical Structure of Links ....................... 83
CHAPTER I

BACKGROUND

1.1 Introduction

The key focus of this thesis is to promote the useability of computerized planning and scheduling in the construction industry. This is achieved by expanding scheduling theory to model the process of repetition and rhythm found on the construction site through the definition of a general repetitive activity structure. It is further achieved by defining a useable updating process that automatically updates the network based on inputs typically collected on daily progress reports.

Construction projects consist of hundreds, and often thousands of activities which must be coordinated and controlled to produce a successful project completion. Such large volumes of data cannot be maintained by the human mind. There is a strong need for computerization to perform calculations quickly and concisely. The advent of Program Evaluation Review Technique (PERT) and Critical Path Methods (CPM) in the late 1950's brought the problems of scheduling large projects to the surface. CPM provided an avenue by which activities could be scheduled based on a set of logic supplied by the user. The computer could then produce a schedule by which the project could be built. While significant detail was required of the
user, the calculations of the schedule could be performed swiftly.

Construction firms would compute the CPM schedule and begin the project only to find that work progress did not always follow the schedule. There was a need to adjust the schedule to reflect the project advancement and provide an updated schedule for the remaining activities. This often involved reviewing all logic of the network that was not realized on the project. This proved to be a very difficult and time consuming task for the user.

Further, for repetitive types of construction (eg. bridges, high-rises, etc.) the construction industry was also hampered by data which had to be entered on an activity by activity basis (low-level data). Sitework for such project types is often paced by grouping activities into production lines. The project is often perceived as large blocks of data and not on a single activity basis. It proved to be very difficult to enter enough data to represent such groupings on CPM or PERT.

Conventional scheduling of large project types has not been well received by the construction industry. Two important reasons are:

A) Schedules are not capable of updating to project conditions quickly enough to help in critical decision analysis.
B) Data grouping is difficult and time consuming to model on the activity by activity basis of current network models.

1.2 Thesis Objectives

The idea of higher-level inputs has been addressed by Laramee (Reference 16) who developed a definition of a repetitive activity which consisted of typical activities performed at numerous locations on site. While the algorithm did allow some advantages of continuity of work and rhythm, it was limited in its ability to model the construction site and did not allow for arbitrary sequence changes when updating.

The objectives of this thesis are twofold. First, to use Laramee's prototype model as a base by which to study the repetitive nature of construction through an in depth site study and develop the repetitive activity into a generalized higher level structure. Second, to further use the opportunity of the site study to define an updating process for the construction industry, including an algorithm for the automatic updating of the schedule thus eliminating the need for the user to constantly mould the network logic to fit actual progress.
1.3 Methodology

The thesis originated with an extensive search of literature. This was followed by a limited examination of a $30 million, 26 storey high-rise project. It became evident that the repetitive activity could be clearly defined. It was also clear that continuous updating of the schedule was essential if scheduling was to be a useful construction management tool.

The requirements of the updating process were defined. A prototype system containing the automatic updating ability was added to a scheduling package containing Laramee's repetitive definition. The system was tested extensively on a $5 million, 4 storey commercial building. The test project involved substantial interaction with the site superintendent and regular contact with the project manager. Suggestions from management were incorporated with test results to mould the system into its final state.

The requirements of the repetitive activity evolved throughout the extensive testing of the system. Systematic updating of the project, made possible by the newly added updating process, provided insight into the effects of continuity, float and criticality of the repetitive structure. A general structure was then defined for the repetitive activity to be added to the updating definition and complete the model.

Of prime importance was a model which was simple to use and
of an acceptable format to the construction industry. This entailed capturing the notion of updating and repetition in the manner in which it is perceived by the construction personnel. Furthermore, it had to be presented in a way that personnel could quickly grasp and thus promote its use. The model aimed at minimal inputs by the user and promoted an efficient user interface.

1.4 Thesis Overview

Chapter II defines the concept of the repetitive activity and outlines the characteristics by which the construction industry perceives its role in providing continuity and rhythm to the project. The final definition is included at the end of the chapter.

Chapter III defines the construction industry's view of updating and suggests a process by which continuous monitoring of the schedule may be attained. The continuous updating algorithm is included at the end of the chapter along with its handling of the repetitive activity described in Chapter II.

Chapter IV discusses issues which arose from work in the areas of data storage, expert systems and its communication to the construction industry. The chapter closes with recommendations for future work and the conclusion to this thesis.
CHAPTER II

THE REPETITIVE ACTIVITY

2.1 Objectives

The purpose of this chapter is to develop a technique by which to capture the inherent process of linear repetition evident in high-rise and other forms of linear construction into a computerized scheduling algorithm. This process will enable concurrent manipulation of large blocks of data giving the user a powerful scheduling ability to model work patterns and the intrinsic rhythm which they provide to the project.

Scheduling a construction project requires the definition of each activity and a realization of how it fits into the overall process. This level of detail generates a substantial amount of data which must be compiled if an accurate model is to be attained. While the data cannot be significantly decreased without losing modelling capability, it is possible to redefine inputs so that much of the data input can be calculated automatically and in a way that reflects work continuity patterns desired by construction personnel.

The above premise is based on defining inherent structures in the construction process which enable data to be described at a higher level and then generate the lower level information;
much like the word "alphabet" which generates the sequence "ABCD...to Z", eliminating substantial data input, yet retaining all essential details of the data base. These structures are known as higher level information objects: high level data capable of breaking down to a full low level base. A paper by Anderson, Fjosne and Solberg (Reference 1) showed this methodology could substantially reduce the inputs required for the model. Thus the algorithm offers optimization of data input and the ability to model the inherent process by which large projects are controlled.

Literature (References 6,15,20 and 28) shows the area of linear repetition to be one of great potential to capture the heuristic nature of the construction process and benefit from its increased coordination, learning effects, and resource handling. Its natural grouping of low level data makes it a prime area for the adoption of higher level structures. The repetitive structure is typically found on large repetitive type projects such as high-rises, roadways, and pipelines, which are conducive to rhythmic progression of work packages along a series of locations.

O'Brien (Reference 19) recognized the hybrid of repetitive and non-repetitive activities present in the construction of high-rise buildings and classified them as two distinct modes of construction. Stradal and Cacha (Reference 29) defined three classes of repetitive work sequences which they felt to be noteworthy factors to consider during scheduling. These
sequences were based on the sense of rhythm which they brought to the project.

Peer (Reference 22) noted that manufacturers often used repetition as a tool to optimize the productivity of their machine shops. He likened repetitive work of high-rise construction to a manufacturing process, defining a system where single and repetitive activities could be amalgamated into one system to take advantage of higher productivity through continuity of work. He also stated the insufficient exploration of this structure in scheduling theory:

"The need for creating working continuity and balancing the whole process into an integrated production system is completely neglected.".

The strongest statement came from Birrell (Reference 3) who was keenly aware of heuristic structures of the construction industry and was not convinced that current scheduling abilities were sufficient:

"There have been various presentations of networks for various uses, and the construction industry has tried to use the CPM and PERT variations with unsatisfactory results mainly due to their incompatibility to the essence of the construction process."

He stressed the need to tap the experience of the management personnel the heuristic concept of construction scheduling:

"This heuristic concept of the construction process is a model that is simple, strong, realistic, optimally cheap to use, but unfortunately not written down. It is only realized by those people who are, or have been, involved in actual construction management. Senior construction practitioners who have put into
writing their thoughts on the matter reach conclusions similar to the above paraphrase and their prestige adds weight to the concept.”

He also surmised that a realistic model is based on organizing the project into smaller definable "work squads" which will pass through the locations in a set sequence; the necessary information to achieve this being a fundamental knowledge of construction technology and its industrial organization. Birell showed a strong leaning towards the repetitive process concept built upon the specific technological and organizational environment of each project.

The literature shows linear repetition to be an important yet not fully understood aspect of scheduling. The goal of this chapter is therefore to fully realize and develop the linear repetitive process by attaining higher level information objects which capture the essence of work continuity and rhythm found on the construction site.

2.2 Laramee's Model.

Laramee (Reference 16) developed a working model to show how the repetitive nature of high-rise construction could be modelled to take advantage of continuity of work and yet not lose sight of the individual activity definition. This was achieved by the definition of a repetitive structure—an activity which is repeated at several locations in space in a linear progression of time—modelled with high level data. This
data is then used to generate the low level data necessary to complete the data base.

The high level data required to generate the repetitive structure consists of:

a) activity name, code number and general factors  
b) starting location  
c) ending location  
d) number of crews  
e) production rates of each location  
f) start to start time  
g) finish to start time  

The data of a) represents the primary links to the user, the data base, and external constraints of the activity such as weather factors and learning curves. The data of b) to g) generates the activity shown in Figure 2.1. All locations between level 3 and level 6 are automatically included: a premise of Laramee's model. The production rate is 4 days/level. The finish to start (FS) link between the levels of a particular crew (eg. FS=1) models the time required for setup and material handling. If more than one crew is specified (eg. crews=2), a start to start (SS) link between crews is permitted to model sharing of resources between crews (eg. SS=2, Figure 2.2). These links (FS, SS) join levels within the repetitive structure and are referred to as its internal logic.
As the model consists of a hybrid schedule of repetitive and non-repetitive activities, two logic links are needed.

a) typical link
b) non-typical link

These links interrelate the high and low level structures and are referred to as the external logic.

The typical link is used to represent a relationship between the same levels of two repetitive activities (Figure 2.3). This link is commonplace during many phases of construction where one work package follows the next (e.g., build form, place rebar, pour concrete...). The non-typical link is used to represent all other relationships (Figure 2.4). This link is essential to amalgamating a hybrid system and ensuring a complete logic network.

The Forward and Backward Pass required to generate the schedule are performed on a time-space matrix. Blocks of data
Figure 2.4 Non-Typical Links

are moved in a certain direction (i.e. along the time axis) but maintain their allocation in space. During each pass, the repetitive activity is moved as a block of data to maintain its relative structure on the matrix. Thus, the process ensures coordination of all activities within the constraints of time and space and at the same time allows continuity of repetitive structures (Figure 2.5).

2.3 Previous Studies

Laramee's model surpassed that of Birrell by defining the basic requirements of a repetitive activity and implementing them into a working system. Early site studies showed the definition to be powerful in its ability to model in large blocks of data, but limited in its application.
When Professor A.D. Russell relocated to the University of British Columbia from Concordia University, two extensions were added to Laramee's model (Reference 26):

A) The ability to overlap activities using lags other than FS=0 (SS, SF, FS, FF) was added. This allowed a certain increased flexibility of basic logic links.

B) A staging effect was added to divide the repetitive activity into segments, where each segment could have its own production rate. This allowed increased flexibility of the work profile.

The author worked with this revised system and found the repetitive aspect could only model a select group of
construction work patterns. Furthermore, the model was fragile and could not take any modification during the course of a project. This was due to the basic definition of the data structure by which repetition was built. Much of the structure included logic links that were "implied" by its geometric shape. However, updating the schedule disturbed the shape and implied links were lost, eliminating the ability to define the repetitive activity as low level data, and hence no longer conforming to the definition of higher level information objects. Two points were clear:

A) The repetitive activity as defined by Laramee did not provide a generalized definition by which to capture recognized work patterns in construction.

B) The repetitive activity was not able to maintain its higher level structure when subjected to arbitrary changes to these work patterns.

The author generalized the repetitive activity structure by undertaking a specific site study using the revised Laramee model combined with the updating process described in Chapter III. The work involved close interaction with construction personnel to generate feedback from all hypotheses. The following sections discuss observations from the research.
2.3.1 The Site Studies

Studies used in the research include: construction of a section of the Advanced Light Rapid Transit (ALRT) project in Vancouver, British Columbia; the Olympic Oval for the 1988 Winter Olympic Games in Calgary, Alberta; and a 26 storey high-rise in downtown Vancouver.

In addition, a long term site study was required to allow a thorough trial of prototype repetitive structures. Most important was the need to find a site which could be modelled throughout all phases of the project so that all aspects of the activity could be explored. This included: early scheduling, fixing start dates, redefinition of activities and their relationships, ongoing activities, updating effects and final presentation of the as-built schedule. The desired study was a medium size (5-10 million dollars) project consisting of significant repetitive work to justify using the model, yet small enough to permit a high ongoing variance in construction methodologies, allowing continual redefinition of the repetitive procedures to test the model's flexibility.

The long term site study was performed on a $5 million commercial building located in the Kitsilano area of Vancouver. The project had 6 levels which could be modelled as a repetitive activity. The General Contractor, Foundation Company of Canada began excavation in January 1984 and planned to finish the project at the end of January 1985. The site study was
implemented in March 1984 and continued until construction completion in March 1985.

A schedule was developed and updated on a regular basis through coordination of the Site Superintendent, Frank Smith; an internationally experienced member of the company. Throughout the study, new techniques were tested to promote the useability of a computer generated scheduling system on a construction site.
2.4 Observations

The attributes of a repetitive activity can be grouped into four categories:

A) Flexibility: degree to which it can be modelled
B) Logic Relationships: its ties to other activities
C) Continuity: its contribution to project rhythm
D) Criticality: its effect on the use of float

A discussion of each category and its effect on the schedule is given below.

2.4.1 A) Flexibility

Laramee's definition permits the modelling of activities as repetitive if they are joined in sequential floors, portraying a regular rhythm on consecutive locations of a project.

Modelling of the ALRT project showed a further application of rhythm. Two difficulties were encountered using Laramee's original model. First, because of design considerations not all activities were present at each location. In particular, crossheads were repeated on a non-regular basis. This necessitated their treatment as non-repetitive low level activities.
Second, activities often required changes in the number of crews in order to meet client imposed milestones. As Laramee's model did not allow changes in the number of crews for the activity, it was modelled by changing the production rate. For example, if the number of crews was doubled for a particular stage of the activity, the production was doubled. However, this was found to give an inaccurate schedule at the low level data as the start and finish of each level were not correct (see Figure 2.6). Consequently, the project ended up being broken into five phases with work continuity being imposed through the use of milestones. The model needed to be extended to allow a change in the number of crews at each stage of the activity to provide an adequate breakdown of crew levelling. If the number of crews is increased, the new crew can be started as soon as the other resources (materials etc.) are made available. If the

![Figure 2.6 Modelling Changes in Number of Crews by Varying Production Rate](image-url)
number of crews is decreased, the first crew to finish its current level is dropped so as to expedite the crew levelling process.

The Foundation study was also found to consistently invoke non-regular repetitive work patterns. Furthermore, due to the relatively small size of the project, site personnel often took advantage of changing the sequence of the work patterns to alleviate problems such as storage of material, or delays on certain floors. At one point, the sequence was totally reversed and the finishing activities were rescheduled from the top floor downwards to allow for the late shop drawings on the lower levels.

These studies clearly show a need for a more general definition. In particular, work patterns consisting of non-regular repetition must be recognized. The ability to resequence the pattern must also be examined.

Recent steps to generalize the repetitive structure have been implemented in response to the needs of each study. These include the ability to model patterns consisting of semi-regular repetition (i.e. work X levels, then skip Y) and also a reverse ordering of the structure. These steps respond to the most common problems of the repetitive structure and serve to bridge the gap until a general structure can be implemented.
2.4.4.1 User Interface

Side effects of generalizing the structure are increased data in both the input and storage phase. At some point, the generality becomes self defeating. This theoretically occurs as the high level information object approaches equal data storage and user time as its low level counterparts. However, in reality it is user time that is considered the governing factor. Regardless of its minimized data storage, a complicated user interface will cause the user to shun the program.

A user interface must be designed to maintain a simple input process for the user. This suggests that input prompts be kept to Laramee's simplified definition unless the non-generalized version is required. Such methodology also ensures minimum data storage, expanding the data base for each activity as it requires a greater generality.

Construction personnel have shown a desire to use the repetitive structure. However, they find it difficult to relate to an input routine based on labels. It would considerably ease their use of the system if a simple graphic representation of the repetitive structure could be implemented to help affirm and reassure the user of such inputs. This step is currently being added; an example of its output is shown in Figure 2.7.
2.4.2 B) Logic Relationships

It is important to review the effects that reordering has upon non-typical logic constraints which still exist but may now play a different role. For example in the case of Figure 2.8, reversal of sequence will require the two non-typical logic links to be broken and relinked to the opposite end of the repetitive activity. All links are made with specific locations.
of the repetitive activity in order to retain the low level data.

Figure 2.8 Resequencing and Non-Typical Logic Constraints

Care must also be taken with typical logic links when a change in sequence is adopted for a group of repetitive activities. For example, Figure 2.9 shows three repetitive activities which have been scheduled. If the "Form" activity was to change its sequence, it would be advisable to reflect that change in sequence in the remaining two activities to ensure an optimization of continuity between activities to minimize float time.

Grouping of repetitive activities often occur on the site. Birell (Reference 3) referred to them as shadow activities: each activity is activated as soon as its predecessor is completed (eg. concrete is poured as soon as the last piece of rebar is in
When changing the sequence of a group of activities the typical links would automatically readdress themselves. However, non-typical links must be reviewed for possible redefinition of their roles. Tracking of such logic links could be greatly aided by a graphical display of chosen activities and their logic links.

2.4.2.1 System Loops

Laramee's definition of the repetitive structure lacked the facility to assign logic links to and from the same repetitive activity. When this situation occurred (Figure 2.10) the algorithm perceived it as a "loop" in the system.

The loop is caused by the adoption of sequence step ordering from CPM (Reference 12). A repetitive activity is given a single sequence step so that it may be moved as a block of data.
Figure 2.10 A Logic Loop Involving a Repetitive Activity

during the Forward and Backward Pass. Thus, when a logic check is run, it will not accept a repetitive activity which has both a successor and predecessor from the same activity. In reality, the repetitive activity spans many sequence steps and can often handle the above situation when it is checked at its lower level.

To circumvent this problem, one could negate the link which occurred at the later moment in time (i.e. the return link to a repetitive activity) allowing the Forward and Backward Passes to proceed. The link could then be reinstated as an independent logic test to ensure lower level data compatibility. If the single activity was found to be scheduled within the duration shown as C, no loop exists. However, if the single activity could not be scheduled within duration C (due to a duration greater than C or interference from links to other activities) a "continuity" loop exists.
The situation of a continuity loop involving a repetitive activity has been noted by Peer (Reference 23) whose answer is to break the continuity in the repetitive process. However, this situation need not be automatically handled by the algorithm. A continuity loop in the system can point out to the user a mistake in his logic or a missed point of reference. Given the structure of the loop, it could then be possible to redefine the problem area by one of the many tools available to the scheduler: such as splitting the single activity, or redefining its intent. If no redefinition is possible, the user can then make the decision to adjust continuity for that particular activity.

To promote such schedule manipulation it would be essential to provide the data in a graphical format so that their interaction could be easily seen. This would once again suggest a graphical display of activities and their logic links.

2.4.3 C) Continuity

The purpose of repetitive activities is to promote continuity of resources by providing a rhythm to the project. This continuity can be seen as a double hierarchy:

a) partial activity level
b) total activity level
The partial activity level is derived by ensuring continuity of each resource on that particular activity. This was provided for in Laramee's work by the levelling of shared resources, material handling, and crew availability (Section 2.2). The total activity level is achieved by the sum of the partial activity levels. This was provided for in Laramee's work by the overall production rate, given the resources available.

Laramee's repetitive activity was built on the partial activity level whereby the crew's rhythm was considered precedent (i.e., if specified 2 crews, each crew must work on every second designated location). The total rhythm thus evolved, taking on the shape generated by the partial rhythms of the structure. Given a large variance in the production rates of the locations of the repetitive activity, the precedent of partial rhythm often caused a slowing down of the total rhythm. This was caused when the algorithm demanded that crew A perform the next location, regardless that crew B was idle (Figure 2.11).

Consultation with Foundation personnel quickly revealed that the continuity of the total activity level was paramount to the rhythm of the entire project. Crews were perceived as a flexible resource, used to attain the required production rate of the activity as a whole. This suggested a major change to Laramee's philosophy of the repetitive structure. The repetitive activity needed to be redefined to allow the total activity level to take precedence. Using the crews as they became
available, time between locations could be minimized hence accelerating the cumulative production rate of the overall structure. This would also ensure an optimal distribution of the crews for maximum utility (Figure 2.12).

Continuity at the partial activity level is controlled by a lag factor for material handling and a lag factor for shared resources. The idea is to model the time taken to move all material and acquire all shared resources before the activity can start. It also implies that one batch of shared resources is shared by all crews. Situations exist where 2 or 3 batches of shared resources are allotted between 4 to 5 crews, or where each pair of crews has its own batch of shared resources. It is often found in the construction of the high-rise core, or the towers of a bridge. These are scheduled specifically to give a cycle which is usually based on the availability of formwork and
Figure 2.12 Continuity of External Rhythm

concrete. Such situations impose a more complicated definition of shared resources, yet one that is used extensively in linear projects. There is a need to allow for the definition of multiple batches of shared resources, using the usual premise of "first batch available to next waiting crew".

There is a third choice of continuity of repetitive structures, that of no continuity. This choice was overlooked by Laramee. The Foundation study, due to its small size, often failed to provide continuity of certain perceived repetitive activities. Once this fact is realized it is important to be able to relax continuity for that activity for it restricts the timing of the levels to be performed. This does not mean that the model has failed. Its initial modelling as a repetitive structure allowed much of the input data to be automatically calculated. The later perception of no continuity allowed the
levels of the structure to act as individual activities (i.e. the block of data was broken down to low level data).

An important effect of relaxing continuity of a repetitive structure is its ability to model a shadow activity (Section 2.4.2). To ensure a shadow activity is performed as soon as its predecessor allows, it cannot be restricted to the fixed shape of a repetitive activity (unless both the predecessor and its shadow activity can by perfectly matched as repetitive activities and that site implementation does not deviate from the model). However, a shadow activity cannot be executed unless its resources are available; a constraint that is ensured by the internal links generated for a repetitive activity. Therefore, by modelling the shadow activity as a repetitive structure and then relaxing the continuity restraint, the logic links imposed allow the shadow activity to bend to the shape of its predecessor, yet limit its flexibility to the availability of its resources.

The case can occur where the shadow activity is to begin as soon as its predecessor allows; resources are to be assigned to such an activity upon request to ensure its immediate progression. This situation is often seen when modelling "curing" times for newly poured sections of concrete. Thus, to allow its appropriate scheduling, the continuity of a repetitive activity is dropped along with its internal logic (page 10) to ensure the instant provision of materials and manpower. The external logic remains intact to ensure the correct placement of
the locations of the shadow activity with its predecessor and preserve its links to the network.

2.4.4 D) Criticality

By imposing a fixed structure for the repetitive activity, a new constraint is added to scheduling theory. Float is now defined by the activity's ability to be executed without causing delay in the early start of downstream activities (Free Float) or prolongation of the project (Total Float) given that continuity of repetitive structures is observed. In such a case, all locations (i.e. low level activities) within the repetitive structure (i.e. high level activity) are perceived as having identical float. A simple example demonstrates this (Figure 2.13). In this case, the Forward and Backward Pass consist of moving three activities (two non-repetitive and one repetitive). Since the repetitive activity is moved as a block to maintain continuity it can be seen that both passes generate identical time-space matrices and hence all activities are critical (Total Float = 0). A more detailed look at the schedule will show this to be true if continuity is observed. However, only location 1 and 2 of the repetitive activity must be performed in time in order to finish the project at time T. While continuity of levels 3, 4, 5 and 6 is desired to maintain a rhythm on the job, it cannot be said to be critical to the project duration. Obviously, there must be a distinction between continuity regarded as critical to the project duration and continuity for
Figure 2.13 Continuity Imposing Criticality

the sake of rhythmic project progression.

By allowing the repetitive structure to be backpassed as low level data (i.e. location by location), the non-critical portions can be allocated their appropriate float (Figure 2.14a). This could then be used to the user's advantage. Having identified the non-critical locations, it is possible to reduce their production rate, thus maintaining continuity and releasing a portion of the resources to assist in other areas (Figure 2.14b). An alternative would be to maintain the same production rate and invoke a discontinuity allowing all resources of the activity to be allocated to critical areas and later returned to the activity (Figure 2.14c).

The utilization of the float can be left to the user to allocate as each situation warrants. A graphic display of the
Figure 2.14 Continuity and Its Use of Float
repetitive activity and its float would aid greatly in the manipulation of both continuity and resources.

Field study enforces this definition of criticality as continuity is not always considered essential for activities off the critical path. This is due to possible resource reallocation of non-critical activities which could be utilized to shorten the project duration and thus by far overshadow the benefits of continuity of non-critical areas. Non-critical areas were also considered as a stockpile of work to be executed when the critical path was found to have an excessive resource profile (i.e. rainy days, material shortage, manpower congestion, etc.).

2.4.4.1 Losses Due to Linear Repetition

It can be seen that invoking continuity into a schedule should not be taken lightly. While it does allow benefits of rhythm and high level input, it can waste useful float if production rates are not carefully chosen. The worst case model is shown in Figure 2.15a. Continuity has delayed the "block" activity as the Early Start Date (ESD), given continuity, is 3 days later than the true ESD. The 3 days of float cannot be utilized by the scheduler since the Free Float and Total Float are calculated from the ESD given continuity. The best case model is shown in Figure 2.15b. Continuity does not cause any increase in project duration and is truly critical at all locations. In true life situations, the model will fall between
Figure 2.15 Worst and Best Case Models of Linear Repetition

the best and worst case limits. Where it is not possible to manipulate the production rate of the repetitive activity to take advantage of such float (due to high rental of equipment, limited crane time etc.) it is the delayed Early Start caused by continuity that must be carefully weighed against the gains in rhythm and high level input.

When this trade off of float versus repetition was put forward to site personnel, they showed a strong leaning towards the case of repetition, especially if the activity involved a subcontractor for it provides the General Contractor with a
useful method of minimizing the problems caused by congestion of workers on the site by delaying the subcontractors start until his work progression is ensured. Once the subcontractor is called to work, he can proceed at a steady pace and thus avoid the problems of recall and material storage. This is an important characteristic of high-rise construction as material storage space and work space is often limited. Personnel however, did stress that repetition could be abandoned if its float characteristics were found to have a significant detrimental effect on the critical path.
2.5 The General Repetitive Structure

The aim of this section is to provide a general definition for the repetitive activity based on the recent site studies and the work done by Laramee and its further extensions. This definition takes into account the problems discussed in the above sections and provides a model by which future research in this area can be referred.

First it is important to present a definition of certain terms pertinent to the repetitive structure.

**Repetition:** a process which is repeated at two or more points in time.

**Continuity:** an uninterrupted connection in time, space or development.

**Repetitive Activity:** an activity or process that is recognized by construction personnel to display repetition and enhance continuity of resources through rhythmic progression.

Laramee's definition of the repetitive activity stated that all locations in a given range should be sequential, thus requiring only the first and last locations to be specified (Section 2.2). A comparison of the first and last locations also provided the "direction" the activity was to follow. This was not realized by Laramee but was initiated during the later site studies (Section 2.4.1).
Laramee's definition was not found to be flexible enough to model the majority of location patterns evident in repetitive activities. The extensions of "skipping" certain locations on a regular pattern and reverse ordering of the structure were added to the definition (Section 2.4.1).

The resource profile of Laramee's definition was found to be inadequate. A staging effect was added to divide the activity into segments, where each segment could have its own production rate (Section 2.3).

The author recommends that the following attributes be added to the extended Laramee's model to achieve a general model for the repetitive structure. Location patterns modelled by the repetitive activity must include non-regular cases (Section 2.4.1) as the physical outlay of a project does not always lend itself to a regular pattern. The resource profile must be extended to allow a crew profile to be determined for each stage of the activity (Section 2.4.1). Multiple batches of shared resources must be allowed as they play an important role in the rhythm of repetitive tasks (Section 2.4.3).

It is further recommended that a visual display of the repetitive activity be accessible so that the effect of all attributes on the activity's shape can be quickly demonstrated to the user in a comprehensive manner and that applicable adjustments may be made.
The site studies show that the repetitive structure should be based on a distinct set of variables that have strong roots in the construction industry. The attributes required of a repetitive activity are summarized below and expanded upon in the following sections.

A) Provide a data structure that:
   1) utilizes high level information objects to minimize user input and data storage.
   2) allows manipulation of work patterns as large blocks of data.

B) Provide flexibility that:
   1) enables modelling of the majority of work patterns found in construction.
   2) allows changes in the activity's shape to adapt to the changing construction site.

C) Provide continuity:
   1) at the partial activity level to promote continuity of resources.
   2) at the total activity level to obtain the benefits of rhythm.
2.5.1. A) Data Structure

While the inputs required to generate the sequential file do not constitute the major areas of saving in data input and storage (Reference 16), they are the major inputs visible to the user. As such, they form the essence of the user interface and must promote simplicity of use while allowing a general definition of the repetitive activity.

The inputs required by the user to promote the general model for the repetitive structure are:

a) activity name, code number, other general factors
b) sequence (eg. 1 to 6 exclude 3)
c) stages (eg. 1-4, 5-6)
d) production rates per stage (eg. 4,2)
e) number of crews per stage (eg. 2,2)
f) number of batches of shared resources (eg. B=1)
g) time required for shared resources (eg. SS=2)
h) time required for material handling (eg. FS=0)

The locations are entered by a simple set of commands designed to minimize user input and data storage.

A to B, exclude C,D,E,F....

plus

S to T, exclude U,V,W,X....
This is followed by the stages which essentially divides the repetitive activity into segments. A production rate and crew profile are then added for each stage and is assumed to be the same for all sequential locations of that stage. The number of batches of shared resources is given and is assumed to be constant for the duration of the activity. The time that a crew requires the shared resources is denoted, along with the time required for the crew to move other resources between its locations. Examples of repetitive activities generated from these inputs are shown in Figure 2.16.

The data base used will depend on the computing environment in which the algorithm is housed, its primary functions and its integration to other areas of project management. However, all data bases must be able to produce a definition of the repetitive activity with low level data. This provides a reference point from which all bases can be compared. The author assumes the input can produce a sequential file (Figure 2.17) consisting of the locations, a list of stages, the number of crews working and their production rate for each stage. This is linked to the memory containing the number of batches of shared resources used and the time requirements for shared resources and material handling.

The above definition enables most work patterns to be modelled using very little user input. It is not considered worthwhile generalizing the definition further and increasing complexity of user input. Yet, it is not advisable to deter a
Figure 2.16 Examples Based on the General Repetitive Activity Definition
user entering a work pattern if it is considered favourable to be modelled. The model described allows the data to be entered on a location by location basis if desired. While this eliminates some saving in storage of this particular repetitive activity, advantage can still be taken in the gains in continuity, manipulation of large blocks of data, and the modelling of shadow activities.

The graph shown in Figure 2.18a illustrates the trade off between generality of the repetitive definition and its corresponding input. The data has been developed based on the number of characters that must be input by the user to model each level of generality of the repetitive model. This measurement was used as it is an absolute value and is not
effected by the data base structure in which it is housed. The stages have been set at five locations each. The regular repetitive activity denotes all sequential locations are included, as per Laramee's model. The non-regular repetitive activity is based on 20% of locations excluded. The minimal repetitive activity is based on 60% of locations excluded. The low level activity is shown for comparison and is based on the CPM minimum of one predecessor and one successor per location.

The characters counted correspond only to those input requirements shown above which must be entered by the user and cannot be predefined by the user interface. Note, this does not include the activity description which is only entered once for the repetitive activity yet must be entered at every location with low level data. The characters counted also do not include any links made between the repetitive activity (or the low level data used for comparison) and any other parts of the network.

The graph shown in Figure 2.18b highlights the increasing efficiency of the regular, non-regular and minimal repetitive patterns with the increasing size of the activity, settling down to .9, 1.1 and 1.5 characters per location respectively. The low level data requires a consistent 4.0 characters per location to model a repetitive activity and does not have the advantage of imposing continuity through the manipulation of large blocks of data.
Figure 2.18 Tradeoff Between Generality of Repetitive Activity Definition and Efficiency of Data Input
2.5.2. B) Flexibility

Changes in sequential order of the locations are allowed by reentering the pattern of the activity over the old data. The stages and other data will remain tied to each location and thus maintain data integrity with the reordering process. However, this data can also be reentered if required. Changes due to updating will be discussed in Chapter III.

2.5.3 C) Continuity

Continuity at the partial level must provide a smooth flow of each resource along its locations as it progresses. This can be defined as three constraints:

a) material handling between floors of a particular crew
b) sharing of resources between multiple crews
c) allocation of crews to particular levels

The first two constraints can be dealt with by using the FS and SS links as defined by Laramee (Section 2.2). The third constraint must be handled by ensuring that each crew is reallocated as soon as it completes a location.

By controlling all three constraints, continuity at the total activity level ensures that the activity as a whole
progresses in the most expedient manner possible by starting each succeeding location as soon as all resources are available, thus minimizing any period of non-productivity.
2.6 The Repetitive Building Process

The site studies have shown that building a repetitive activity requires a knowledge of available resources (crews, material, shared resources) and their placement in time and space. Each location to be processed must know:

1) Which crew will be available first,
2) how long it will take the crew to transport their materials and
3) when will shared resources be available from other crews.

Although each location must satisfy the three criteria listed above, the problem can be simplified to "keeping the crews productive". Shared resources and time for material handling are resources which dictate if a crew is able to work. Thus, given an available crew, it is possible to find when it may start its next location by the amount of time it takes to acquire a batch of shared resources from another crew, and move its other materials from the last location completed to the next location to be performed.

The definition of the process is analogous to a simulation model (Reference 14). The only points in time that must be examined are when a crew completes a location. It is then assigned to its next location as quickly as possible. Thus the process can be modelled as simulation using next-event
incrementing based on the crew's finish date for each location.

The simulation process calls for multiple crews and multiple batches of shared resources. Pools must be set up to keep track of these variables. Pool A (Figure 2.19) denotes a table containing each crew, the day it is available (Sa), and weather it is currently employed by the repetitive activity. Pool B (Figure 2.19) denotes a table containing each crew, the day that shared resources will be available (Sb), and the crews which are currently using the shared resources.

2.6.1 The Flowchart.

The sequential file INPUT (Figure 2.17) is used to read the activity data for the simulation process discussed in the following section. The sequential file OUTPUT (Figure 2.20) houses the finished repetitive activity; it contains each location processed, the crew number and its production rate, its Early Start and Early Finish Date given continuity, and a predecessor list of SS links. A column is not required for the FS links as they exist between all locations of each crew and can therefore be derived from the crew number column. However, SS links join locations that use the same batch of shared resources and are not confined to specific crews.

The flowchart for the building of the repetitive activity is shown in Figure 2.21. The algorithm starts at Day 1 (Si=1)
and fills in Pools A and B (all \( S_a \) and \( S_b = 1 \)) to show all crews are currently available. The \( X \) batches of shared resources are given to the first \( X \) crews by affixing a flag (denoted by an asterisk) to show their current location (i.e. in case of a tie, give to the lowest crew number). These defaults will be changed in the case of update data to reflect the current status of the activity. The update phase will be discussed in Chapter III.

The INPUT file is used to determine the next location to be modelled, the number of crews working at this stage and their production rate. The location and production rate are placed in
file OUTPUT to denote the correct modelling order. Pool A is checked to ensure the correct number of crews is working in this stage, each being denoted by an asterisk. In the case of adding a crew, the lowest crew number not working is initiated. In the case of losing a crew, the first crew to finish its current location is dropped.

The next available crew is found by searching Pool A for the earliest Sa which displays an asterisk. This is based on the crew's finish date of the last location modelled plus time required to move materials to the new location (Sa=Fi+FS+1). Note, one unit of time is added due to the definition of the start and finish dates (see Laramee, Reference 16). The crew's number is then placed in file OUTPUT. The first available batch of shared resources is found by searching Pool B for the earliest Sb which displays an asterisk. This is based on the crew's start date of the last location modelled plus time.
Figure 2.21 Flowchart of the Repetitive Activity Building Process
required to use shared resources \((S_b = S_i + S_S)\). The asterisk is then removed from this crew and placed with the crew at the current location. The location from which the batch of shared resources was procured is also placed in the predecessor column of the file OUTPUT to denote the SS link between the locations.

The later of \(S_a\) and \(S_b\) denotes the time at which the current level may be modelled using the first available crew, given that the materials are already there to be used and that shared resources are available from the last crew who used them. This start \((S_i)\) is then placed in file OUTPUT. The finish date of the level is found by adding its production rate to its start \((F_i = S_i + P_i)\) and is also placed in file OUTPUT. Pool A is then updated to show when the current crew will be available next. Pool B is updated to show when the current level will be finished with the shared resources.

This process is repeated for each floor of the repetitive activity to be modelled. The finished model is contained in file OUTPUT and is ready to be scheduled.

An example of the finished model is shown below in Figure 2.22 along with its INPUT and OUTPUT files. Each crew displays an FS link between its locations to ensure time for material handling. SS links are displayed between the appropriate crews to ensure the availability of shared resources. Note the general structure of the activity can be easily checked by comparing the INPUT file with the visual display providing feedback in a form
that can be easily absorbed; locations modelled, stages and production rates are self evident. Closer scrutiny, such as tracking of crews and batches of shared resources is possible by viewing the OUTPUT file.

2.7 Summary

The repetitive activity has been developed based on the needs of the construction industry. The algorithm describes a model by which the user may utilize the process of repetition and rhythm on the construction site. Use of construction terminology and graphic feedback promotes its acceptance as a useful management tool. The foregoing model is not yet integrated into the system at the University of British Columbia. However, field studies have shown the need for the management support it provides.
Figure 2.22 The Finished Model of the Repetitive Activity
3.1 Introduction

The purpose of this chapter is to focus on the problems of updating a computerized schedule to effectively monitor the project. The methods by which the construction industry monitors the construction process are examined and related into a computerized schedule algorithm capable of handling day to day changes on site.

A computer scheduling algorithm which can produce a clear and precise logic network from a simple set of commands would be expected to thrive in the construction industry. Yet it has not happened. Users have thrown away their schedule in frustration early in the project. Computer scheduling has been touted as a useful means of getting to know the characteristics of the job, but not useful for control (Davis, Reference 7).

Lack of control has been attributed to lack of model flexibility. Control can only be implemented when the schedule reflects the day to day status of the project so that changes and their effects can be evaluated. The unresponsive nature of existing models has forced users to reduce updating periods to unacceptable limits; once per month is typical. This precludes
any efficient control of the project since it reflects a net effect of all the changes thus obscuring detection of responsibility to cost and performance (Fondahl, Reference 11).

While there is a lack of literature on updating, Fondahl and Davis reflect the mood of the construction industry towards the useability of monitoring projects with a computerized scheduling algorithm. Existing models are unwieldy in their ability to change with the ever moving construction site. Thus, they fail to supply the user with up-to-date schedule changes and cannot reflect to a Construction Management Information System (CMIS) schedule effects on other aspects of the project.

This chapter defines an updating procedure which can respond to the users needs and maximize the potential of a CMIS. It also proposes an algorithm which provides a continuous updating facility based on typical schedule progress data. The result is a scheduling package which is able to provide the user with an ongoing picture of the schedule to help in the everyday decision analysis and control of the project.

3.2 Criteria Used to Promote the Use of Updating

Certain criteria of the updating system were evolved from a preliminary study accomplished with the help of PCL (Poole Construction Limited). Meetings with the Project Manager and several site visits provided sufficient background to generate
the basis of the updating process as applied in construction.

A useable updating procedure must respond to the needs of the construction industry. It must reflect ongoing schedule progress and provide useful data feedback in a format that construction personnel are familiar with. As such it should utilize the data and the process by which the industry effects control of the project.

A useable updating procedure must also require minimal time commitment by the user who does not wish to become a data input clerk. Inputs must be limited to that data which cannot be collected by any other means. Automation of the system must therefore be maximized.

A computerized updating procedure must utilize the capabilities of a CMIS. Its ability to cross reference data can be used to integrate data from all areas of the project effected by changes in the schedule.

In summary, the updating process must meet three specific criteria if it is to achieve its maximum potential and ensure its use by the construction industry. First, it must reflect the updating process currently used so as to respond to the heuristic nature by which the industry applies it. Second, it must be streamlined to ensure the user is only required to perform functions which cannot be handled by the computer so as to maximize time for review and control. Third, it must exist
within a CMIS to utilize the support capability of the computer system.

A prototype continuous updating system was designed based on the above three criteria. Input data was modelled from daily progress reports. Output data was presented in a manner familiar to site personnel. A method of providing automatic network manipulation for the updating process was added to minimize the user time commitment to those situations warranting control. Monitoring the schedule was thus essentially automatized. The updating system was integrated into the scheduling package containing the extended version of Laramee's repetitive activity definition.

An extensive testing of the updating system was performed with the Foundation Company of Canada (Section 2.3.1). Observations from the study further reinforced the criteria from which it was based and presented a more complete definition of the updating requirements of the construction industry. These observations are presented below in the format of the three criteria in which they were studied.
3.3 Criterion 1: Integration Into the System

A CMIS allows information transfer from all aspects of the project. This aids in decision making and coordinates changes in both cost and time. Parameters of such a system were set out by Paulson (Reference 21) from which A.D. Russell (Reference 27) later developed the design of a prototype system. A simplified system is shown below to highlight the flow of data and its applications (Figure 3.1).

![Diagram of Simplified Construction Management Information System](image)

Figure 3.1 Simplified Construction Management Information System

The update process must be sufficient to ensure that the progress of the project is reflected throughout the CMIS. The
major centres to satisfy are the Data Base Centre, the Time Centre, and the Monetary Centre. These centres must be linked to provide data accessibility and support to the decision making process. Lastly, the system must be able to make a legal stand given that it can be challenged in court.

While all of the above points are expanded upon below, the thesis is mainly concerned with the problems of updating with respect to time. The other areas of a CMIS are included to provide the reader with a complete scope of the update process.

3.3.1 The Data Base Centre

The data base is the storage area of the CMIS for all information pertaining to projects ongoing and completed. Ideally, the data is filed using a standardized code structure adopted by all departments and projects. This ensures continuity of the knowledge base over time, even in the event of a change in personnel. Standardization of the code structure also permits a learning process between projects. Information can be accessed to any aspect of the project in which data was collected and stored. However, information cannot be utilized without caution; it must be used within the context in which it was recorded. As such, all information should be accompanied by a log.
3.3.1.1 The Log

It is essential that a log be added to the data base to clarify the situation in which data is recorded. This precaution is due to the proliferation of variables that can effect a project. For instance: a high price for rebar may be due to a supplier monopoly; likewise, low productivity in formwork may be due to unexpected cold weather. Thus, a log of special circumstances must be recorded if the data is to be useful for future reference.

In summary, to satisfy the link to the data base, the updating process must work off a standard set of company codes and use a log to identify the specific project environment at the time the data was collected.

3.3.2 The Time Centre

The time centre models the project in terms of activities and their location in time. The update process must provide sufficient data to portray the daily progress of these activities. This includes actual start and finish dates and "% complete" if the activity is ongoing. The "% complete" refers to the physical condition of the activity (this will be explained in the next section on Cost Centre). Therefore, to describe the % complete in time, which is needed to calculate how much longer the activity will take, a mechanism for estimating the
"remaining duration" must also be given for ongoing activities. The status of procurement activities must also be supplied for they have a direct effect on the schedule of downstream activities (Procurement, while not a major aspect of the thesis, will be expanded upon in section 3.6.2).

Essential to the correct documentation of a project is the daily site report. This can be largely performed by the computer through the use of an automatic log which dates all update entries. The user can then complete the log by recording any change orders incurred and the current project environment (weather, equipment availability etc). Such documentation allows the user to separate the cause and effect of certain activity changes if required for purposes of billing, reports to management or litigation.

In summary, to satisfy the link to the time centre, a daily (or any other period deemed useful) progress report must be issued, tied to an automatic log or diary. This progress data must be sufficient to ensure updating of the schedule, it must include:

a) actual starts and finishes
b) % complete and remaining duration
c) procurement status
d) change orders and
e) prevailing project conditions
3.3.3 The Monetary Centre

Integration of the monetary and time centres has long been advocated (References 18 and 21) as it promotes parallel processing of the schedule and cash flow. The monetary centre houses two philosophies (pricing and costing) which are viewed as very separate entities by the scheduler.

The price centre is tied directly to the schedule for it depicts the price that must be paid by the owner for each activity performed by the contractor. Pricing strategies vary with the type of contract (cost-plus, lump-sum, unit-price, etc.) and are established by the contractor in the bid to the owner (Reference 2) such as tying all costs to specific milestones or to quantity measurements. However, detailed discussion of pricing strategies is beyond the scope of this thesis.

The cost centre depicts the cost to the contractor of completing the project. Such costs are often tied to the schedule on an activity basis in the form of manpower and materials to provide the contractor with a cash outflow profile of the project. This allows the contractor to perform time-cost tradeoffs during the analysis of the various scenarios by which he is able to construct the project. It also provides a mechanism by which to determine the impact of change orders and claims.
The link between the time centre and the monetary centre is usually made by the "% complete" data entered during the update phase. This depicts the % **physically** complete of the activity as it is the preferred measure of payment used by the construction industry.

In summary, to satisfy the link to the monetary centre, cash values must be tied to the schedule at the activity level and progress measurement of the project must include a physical measurement of each activity.

### 3.3.4 The Decision Making Process

A well structured CMIS promotes information flow to aid the user in his decisions. This is commonly referred to as a Decision Support System (Reference 4). In updating, decision analysis comprises of the "trial and error method"; alternative solutions are defined and analysed to compare their outcome. This heuristic approach is a trademark of the construction industry. Its eclectic nature makes it very difficult to identify a set of consistent variables by which to perform an optimization routine, especially if alternatives can be generated quickly by the system. To this end the updating process must allow a "what if" structure to promote ideas of the user.

The "what if" structure must be supported by information
from all aspects of the project so that alternatives can be compared. This involves feedback from all three centres of the CMIS. Typical reports involving the use of time are bar charts, linear planning charts, resource plots, network logic and productivity curves. It is important to stress that no data can be looked at in a vacuum from the rest (Reference 11) for a decision usually has a ripple effect throughout the CMIS. It is thus expected that all significant details of the decision be recorded in the log for future reference.

In summary, to satisfy the heuristic approach to decision analysis, the updating process must provide an avenue to promote a "what if" approach in solving the schedule problems. This avenue must provide access to all aspects of the CMIS to draw on any information deemed relevant. Such "what if" statements, should be documented in the log for future reference.

3.3.5 Litigation Support

It has long been realized by the construction industry that accurate systematic records are essential to present a successful case for litigation. Care must be taken to ensure the CMIS is structured in a manner acceptable in court.

Computerized Critical Path Methods are widely accepted by the courts as a useful and reliable tool in litigation. However, network analysis will not be accepted as evidence if it is not
explicitly tied to project records (Reference 30).

Care must also be taken in the preparation of computerized printout as evidence. It is not clear whether section 30 of the Canada Evidence Act (Reference 5) regards computer printout as a business document, even though case histories have shown a general acceptance of such forms (Reference 24). This problem was recently addressed by the Federal/Provincial Task Force who prepared a report on Uniform Rules of Evidence (Reference 25).

"After considering the various alternatives, the majority of the Task Force recommends that the Uniform Evidence Act follow an intermediate course (of the statutory provisions), namely, that in addition to other requirements for business documents there be only three conditions of admissibility of computer evidence:

1. Proof that the data upon which the printout is based is of a type regularly supplied to the computer during the regular activities of the organization from which the printout comes.

2. Proof that the entries into the data base from which the printout originates were made in the regular course of business; and

3. Proof that the computer program used in producing the printout reliably and accurately processes the data in the data base."

These conditions state that the CMIS must be reliable and accurate. The inputs used for evidence must parallel those which are used regularly in the course of the project.

In summary, to satisfy litigation support, the updating
process must acquire all data to support its claims during its regular progress measurement. This includes documentation of change orders. When revising the schedule (due to significant variance of progress from the schedule), the reason for changes must be documented so that user reasoning can be later acknowledged. Information used to effect a change must be accompanied by its log so as to read data in a reliable and accurate fashion.
3.4 Criterion 2: The Updating Process

The construction industry views the updating process as having two distinct phases:

a) A continuous updating phase
b) A review and control phase

The phases and their links to the CMIS are shown in Figure 3.2.

3.4.1 Phase A

Progress measurement is input on a daily basis to ensure close monitoring and documentation of the project. An update analysis, prompted by the user, is usually performed on a periodic basis when the net effect of progress of the schedule is considered worthy of noting. However, an analysis can be prompted at any time the user considers it beneficial. This allows new data to be analysed quickly if it is believed to disturb the latest schedule, yet permits unimpeded tracking of the project when progress conforms to the schedule.

The outcome of the analysis is provided in terms of the updated schedule, progress reports and any other data the user may wish to use. From these reports the user can decide if the schedule is under control. If the answer is yes, the cycle is repeated. If the answer is no, the user must enter the review
Figure 3.2 The Updating Process Flowchart
and control stage.

3.4.2 Phase B

At some point in time, it is imminent that the schedule, which is based on a static model, will no longer reflect the dynamics of the project. This is often due to an increasing variance between progress reports and scheduled activities. It can also be due to constraints which did not exist or were not realized at the time the schedule was implemented, such as change orders or new estimates of activity durations.

At this point, it is time to review, effect control and release a revised schedule which, given the new constraints of the problem, once again projects an acceptable outcome. A heuristic analysis or multiple scenario approach is adopted in order to find the most appropriate revisions. The analysis draws on all capabilities of the CMIS to trade off variables of money and time to achieve a satisfactory outcome. Once the preferred schedule has been decided and reasoning for the choice documented, a revised schedule is released.
3.4.3 Schedule Interaction

It is important to note the role of each phase and its effect on the schedule. The role of Phase A is primarily the collation of update material. It is considered part of the daily site routine and can be performed by most site personnel. Such work requires no direct interaction with the schedule. Phase B performs the role of data analysis. This can require extensive interaction with the schedule to respond to the update conditions. Such decision making is usually handled by people authorized to allocate company resources in response to schedule updates.

In summary, the updating flowchart must provide the segregation of data collation and data analysis. This effectively limits direct schedule interaction to authorized personnel and permits data collection to proceed unhindered until a review and subsequent control of the project is required.

Therefore:

a) It must provide a two tier system by which a schedule can be continuously updated and, when necessary, revised.

b) Both phases must provide full potential for input and documentation.
c) Both phases must allow full access to the CMIS for project status reports and, in the case of phase B, for alternative analysis of time and money.

d) Phase B must provide a heuristic decision making capability by providing a "loop" in the flowchart to review alternative procedures.
3.5 Criterion 3: Automation of the Updating Process

The continuous updating phase requires an algorithm that can update a schedule and feed all aspects of the CMIS as each period of progress measurement is input. This must be done with minimum work by the user as the function is merely to track progress against the schedule and portray the timing of the remaining activities. Since the logic for the schedule has already been defined by the user, the algorithm has a data base from which to draw all necessary data to perform this task automatically.

The continuous updating algorithm has not been feasible in practice (Reference 11) due to:

a) The high cost of maintaining computer programs.
b) Problems of rearranging network logic associated with such changes.

Computer technology has effectively eliminated the problems of computer cost. However, the manipulation of logic links is still left to the user who must review the network each time progress data varies from the sequence of the schedule, thus maintaining a high personnel cost. The author has developed a continuous updating algorithm which detects sequence changes and adjusts the remaining activities accordingly (This algorithm is defined in the next section). This effectively eliminates any user input beyond the input of progress data, except when corrective action
must be initiated.

In summary, the continuous updating phase must adopt an algorithm which can support a CMIS and update the schedule for each period of progress measurement, using no time commitment by the user other than that of daily progress reports.
3.6 The Continuous Updating Algorithm

The aim of this section is to describe the continuous updating algorithm that was used in the prototype system to further study the updating process. It was found to respond well to changes in the schedule and supplied an ongoing picture of work progress. The study showed the algorithm to be both robust and realistic in its approach to monitoring and controlling the project.

Continuous updating requires a mechanism which can span the past and future to create a single network. It must be able to detect sequence variations between the program data (past tense) and the schedule (future tense) and make appropriate network adjustments to maintain the network integrity. These adjustments are usually made manually by the user each time the schedule is updated. This essentially is the "bridge" which spans the tenses and allows the schedule to respond to the progress data. The algorithm proposes a definition of the principles by which the user adjusts the network. By adopting these principles it is possible for the algorithm to make the appropriate adjustments and hence perform the update automatically.

Before looking at the network adjustments required for updating it is first necessary to review scheduling theory and introduce some new definitions that updating imposes.
3.6.1 The Activity Window

Logic links used by scheduling theory define the boundaries of time that each activity can be executed whilst maintaining the project duration. In essence, logic creates each activity's window, bound on the left by its predecessors and bound in the right by its successors as shown in Figure 3.3. Its predecessors determine how early the activity can start without breaking the predecessor logic. This is the operation known as the Forward Pass. Its successors determine how late the activity can finish without breaking the successor logic. This operation is known as the Backward Pass.
3.6.2 The Activity Durations

It is no longer possible to classify all activities under "scheduled"; denoting that they are to happen in the future tense. It is now possible to have three classifications of activities (Figure 3.4):

a) The Completed Activity
b) The Ongoing Activity
c) The Scheduled Activity

![Diagram of three types of activities and their durations]

Figure 3.4 Three Types of Activities and Their Durations

The Completed Activity, as the name suggests, denotes that its duration was started and finished within the Past Time Zone. Thus, it has an Actual Duration (the span of time in which the activity was actually performed). The Ongoing Activity has a duration which started in the Past Time Zone and extends into
the Future Time Zone. The amount of time remaining to finish the activity is called the Remaining Duration. The Total Duration of the whole activity over both time zones is called the Projected Actual Duration. The Scheduled Activity is totally within the Future Time Zone, therefore it is already defined by scheduling theory. The duration has both a scheduled start and a scheduled finish, as such it is called the Scheduled Duration.

3.6.3 The Activity Starts and Finishes

The starts and finishes of activities must also be redefined when updating (see Figure 3.5).

Figure 3.5 Start and Finish Definitions and Their Time Zones

For a completed activity, the actual start and finish are known; they are events which have taken place, and are called
the Actual Start Date (ASD) and the Actual Finish Date (AFD) of the activity. For an Ongoing Activity, its duration has also started, therefore it too will have an Actual Start Date. However, the Ongoing Activity has not yet been completed; it is still unknown where exactly it will end in the Future Time Zone. This is the typical situation in scheduling theory and thus it will be given an Early Finish Date (EFD) and a Late Finish Date (LFD) based on its remaining duration. The Scheduled Activity which has not yet begun will not have a fixed beginning or end, therefore scheduling theory rules it will have an Early Start Date (ESD), a Late Start Date (LSD) an Early Finish Date and a Late Finish Date.

3.6.4. Substantial Completion Verses Total Completion

Completing an activity is not always a simple case. An activity may be 95% complete but require extended time to perform the last 5% due to a few missing items (eg. a plug for the sink). Clearly this activity is not totally complete, yet neither is it likely to hold up any activity downstream. Therefore, for scheduling purposes the activity has been Substantially Completed and can be given an Actual Finish Date. To denote that the activity is not yet Totally Complete, a "% complete" (ie. 95%) is added to the update data base for use in the monetary centre as billing data or in the data base centre for reports.
3.6.5 Procurement

As procurement is essential to the progress of the activities it supplies, it seems reasonable to make a direct link between procurement and the activities in the schedule. Thus any delay or advance in procurement would be reflected in the schedule. However, the construction industry is not unanimous in this decision.

While procurement is definitely connected to the schedule, construction personnel do not think it is direct. This is based on the view that procurement is external to the project until its delivery on site. It is also characterized by long lead times to allow for manufacturing and transportation making it difficult to monitor closely. The process is often undertaken by administrative personnel from head office and hence is not under the control of site personnel.

Based on the view of the construction industry, procurement has been indirectly linked to the schedule by calculating the lead time needed from the date it is required on site. By linking it in this manner, a change in the delivery date of materials does not automatically effect the schedule. However, if the user wishes to let the procurement effect be seen, a schedule start date can be imposed directly to the activity affected.
3.6.6 The Hierarchical Structure of Links

We now have four possible sets of data that must be kept for an Updating Scheduling Algorithm. These sets of data, shown in Table 3.1, must be prioritized if one is to expect the network to bridge the gap between time zones. The following system is suggested.

<table>
<thead>
<tr>
<th>ESD</th>
<th>EFD</th>
<th>LSD</th>
<th>LFD</th>
<th>SSD</th>
<th>SFD</th>
<th>ASD</th>
<th>AFD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Traditional Scheduling

Defines window for each activity.

Defaults to ESD unless specific date is entered.

Records activity progression.

Table 3.1 The Four Sets of Data Required For Updating

The ESD, EFD, LSD, LFD are the basis of the network derived by Scheduling Theory to depict the windows in the Future Time Zone for each activity. However, it does not invoke a fixed start or finish for any activity. The Scheduled Start and its related Scheduled Finish do invoke a fixed quantity which can more clearly define the placing of that activity in time. As such, the SSD and SFD must overrule the Early counterparts for they reduce the uncertainty of the schedule (In the construction
industry, if no Scheduled Start Date is imposed, it is commonly assumed to be the same as the ESD). In turn, the Actual Start and Finish Dates must overrule the Scheduled counterparts for they are the actual quantity that occurred. This system is summarized below in Table 3.2 as the Hierarchical Structure of Links.

<table>
<thead>
<tr>
<th></th>
<th>START</th>
<th>FINISH</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOST CERTAIN</td>
<td>ASD</td>
<td>AFD</td>
</tr>
<tr>
<td></td>
<td>SSD</td>
<td>SFD</td>
</tr>
<tr>
<td>LEAST CERTAIN</td>
<td>ESD, LSD</td>
<td>EFD, LFD</td>
</tr>
</tbody>
</table>

Table 3.2 The Hierarchical Structure of Links
3.6.7. The Validity of Logic Links

With the new definitions of updating now in place, it is time to look at the network adjustments that are required to perform automatic updating.

Schedules are held together by logic links which the scheduler believes depicts the physical constraints of the project. For example; allowing the concrete to set for three days before stripping the forms would be outlined as shown in Figure 3.6 with the constraint being modelled as an FS link of three days. What would happen to the schedule if the forms were stripped only two days after the concrete was poured due to a newly adopted curing technique? This, given the network logic is not feasible, yet it has actually occurred and is therefore a justifiable statement to make when updating. It is therefore necessary to examine logic links during the updating phase and determine if they are still valid.

![Figure 3.6 Example of Constraint Modelled as a Logic Link](image)
Logic links that do not conform to the progress data are considered non-valid. These links are by-passed by the Forward and Backward Pass but are not deleted. They are kept to ensure that at least one successor and one predecessor exist for every activity to provide a complete logic network. They are also useful as a flag to show deviations from the schedule in claim situations and management reports.

It is not possible for the computer to take the user's new data and transform it into new logic links (except for the situation of ongoing activities which will be discussed later in the chapter). However, by maintaining the integrity of the network this function is not essential to performing the continuous update phase. New logic links, if any, are left to the user to insert when the authorized personnel deems a review of the network is in order.

The validity of each linkage type can be characterized for all three types of activities. Each activity window is described below with its valid links. Non-valid links are also shown: depicted by a slash through the date (e.g. ASD = a non-valid ASD link).
3.6.8. The Completed Activity and its Logic Links

The successor and predecessor links define the window in which an activity might occur. Since the activity has already occurred, its window is defined by the activity's Actual Start and Actual Finish Dates. Thus, the validity of links to and from the activity do not affect its placement in time since no Forward and Backward Pass are performed on a Completed Activity.

3.6.9 The Scheduled Activity and its Logic Links

The validity of each link for the Scheduled Activity is presented in Figure 3.7 for both the Forward and Backward Pass.

The Forward Pass of a Scheduled Activity is very simple. The predecessor links must always be joined with the latest and most up to date information as this will ensure the most probable schedule is followed. This relationship is found by using the Hierarchical Structure of Links to provide the earliest start based on the known constraints, including those that have become apparent during the update phase. For example; the early start may suggest that the scheduled activity start on the first day of the working week. However, a delay in materials might have invoked a Scheduled Start Date two days later (note, SSD has been input directly by the user in view of late procurement). Given this set of circumstances, it would be irrational to schedule that activity to start on the first day.
Figure 3.7 Forward and Backward Pass of a Scheduled Activity

when new information physically defaulted its start by two days. Likewise, an Actual Start Date overrules any other Start Date due to it being the true start of the activity.

The Backward Pass of a Scheduled Activity demands a little more insight into the effect of successor activities. As in the Forward Pass it is important to make the successor connection with the latest and therefore most certain data. This would suggest the Hierarchical Structure of Links with two exceptions.

1) If the successor, for whatever reason, was found to have actually been started, the logic link is no longer valid. This can be easily demonstrated in the example shown in Figure 3.8. Whereas it is good practice to paint the walls of a room before
placing the carpet so that there is no chance of spilling paint, it may be possible that the painter is unavailable. In this case it would be advisable to lay the carpet first and to protect it when the painter returns to the job. Therefore, successor links with an Actual Start are no longer valid.

2) The other exception involves making a logic link invalid if the successor has already finished. A similar example (see Figure 3.9) demonstrates this point. Practice would suggest that the painter finish painting walls of each room before the light switch plates are all affixed to avoid paint on the switch plates. However, if the painter is unavailable, the light switch plates could be finished first. This will of course make the painters job a little more difficult as he must now paint
carefully around the plates, but it does allow the project to stay on schedule. In this case, successor links with an Actual Finish are no longer valid.

If the Scheduled Activity had only one successor, and this link was declared non-valid, it would float to the end of the project. At this point the user must decide if the float is realistic. If it is not, the user can either leave it with the knowledge that the activity is soon to be performed and hence the float will disappear, or the schedule can be reviewed and the activity tied to a new predecessor. Experience on site has shown that the float is usually neglected unless the activity is being closely monitored.
3.6.10 The Ongoing Activity and its Logic Links

The Ongoing Activity presents a special case as it spans the past and future. Since the activity has been started, it follows that its remaining duration be allowed to continue its progress. The completed portion of the Ongoing Activity does not need to be scheduled as it has already been fixed in time (Section 3.6.8). However, the Remaining Duration must undergo a Forward and Backward Pass in order to calculate its float.

Given the above argument, it is reasonable to assume that the Remaining Duration may not have to be performed immediately (i.e. user may wish to use up some of it's float, or it may have a logic link which will delay its earliest finish). This is not considered a discontinuity in the activity, it is merely a period of zero productivity. This argument is strengthened by the definition of Actual Duration in Section 3.6.2 (Actual Duration = Actual Finish - Actual Start) which lends no credence to its productivity rate with time.

A general rule of thumb to remember for the Hierarchical Structure of Links of Ongoing Activities is that it is based on the Scheduled Activity. This can clearly be seen by a direct comparison of diagrams for the Forward and Backward Pass of the Remaining Duration portion of the Ongoing Activity (Figure 3.10) with that of Figure 3.7.

The Ongoing Activity does have certain characteristics of
its own regarding the validity of logic links. During the Forward Pass, the Remaining Duration has no predecessor links tied to its start (Predecessor links from other activities tied to the Actual Start of the Ongoing Activity do not affect its placement). The possibility can therefore arise of a Remaining Duration which has no predecessors at all (The Ongoing Activity may not have required any SF or FF links). This would allow the Remaining Duration to have an Early Start Date at the beginning of the project. This is prevented by giving the activity a new predecessor stating that the Remaining Duration can start no earlier than the next working day on the job (i.e. TNOW).

The Backward Pass faces a similar situation. The Remaining Duration does not have any successor links tied to the its
start. As such, the only successor links are attached to its Finish (FS and FF links). As in the case of the Scheduled Activity, links involving an ASD or AFD of the successor are non-valid (see Figures 3.8 and 3.9). Given that the Remaining Duration has all of its successors declared non-valid, it would float to the end of the project. Once again it is up to the user to decide what must be done if this condition arises.

3.6.11 Summary

By adopting the Hierarchical Structure of Links it is possible to promote an automatic updating procedure based on the most up-to-date schedule information. The continuous updating algorithm does not profess to optimize the network. Its purpose is to maintain network integrity so that the effect of update data can be automatically displayed, thus performing Phase A of the updating process defined in Section 3.4.1. If the effect of the update is not satisfactory to the user, the schedule can be reviewed and the appropriate changes made, allowing the user to maintain direct control of the schedule as defined in Phase B of the updating process, Section 3.4.2.
3.7 Updating The Repetitive Structure

The continuous updating algorithm provides a robust updating process for networks based on low level data. The repetitive structure is essentially compatible with the algorithm as it is also built from low level information (premise of high level input). Some clarification however, must be given to update data which does not fully agree with the repetitive activity definition. This section discusses the flexibility of the update process required to model the repetitive structure and provides a procedure to meet its needs.

If a crew is presently working on two floors, the model must recognize the crew is balancing its time between multiple ongoing locations. This was often found to happen on site; a crew would be in the process of finishing one location and starting another. Also, the model must adjust to update data which does not follow its designated floor sequence. This situation occurred frequently due to lack of space or drying times requiring the subcontractor to work around certain locations.

The above cases are addressed in the repetitive building process by means of the update block of the repetitive activity flowchart (Figure 2.21). The process is as follows.

Given the sequential input file (Figure 2.17), all completed locations are removed since they do not have to be
scheduled. Ongoing locations are then processed through the update block. The scheduled locations are further added to the structure as described in Section 2.5.4. The update block of the flowchart has been expanded in Figure 3.11 and its procedure outlined below in the following section.

3.7.1 The Update Block

Each ongoing location must be checked to see how much longer (if at all) it requires the shared resources. This is achieved by finding its Actual Start Date and then adding the time it requires the shared resources minus the time it has already used them \( S_b = ASD + SS - (TNOW - ASD) \). If the value of \( S_b \) is less than \( TNOW \), the crew no longer requires the shared resources and \( S_b \) is given the value of \( TNOW \) and placed in Pool B to show that the resources are available immediately. If the value of \( S_b \) is greater than \( TNOW \), the crew still requires the shared resources and its value is placed directly into Pool B to denote the day the shared resources will be available next. The remaining duration is added, along with any FS link, to the update date to establish when the crew of the ongoing location will be available \( S_a = TNOW + REMDUR + FS \). The \( S_a \) value is then placed in Pool A. The remaining duration of the ongoing activity must be recorded in file SEQ if it is to be included in the schedule. As such, the ongoing location is given a start date equal to the update date \( S_i = TNOW \). The finish of the remaining duration is also placed in file SEQ and is calculated
Calculate when crew is finished with shared resources.
\[ S_b = ASD + [SS - (T_{NOW} - ASD)] \]

Calculate when crew is available next.
\[ S_a = T_{NOW} + REMDUR + FS \]

Proceed to model remaining part of ongoing location, starting at \( T_{NOW} \).
\[ S_i = T_{NOW} \]

Calculate finish by applying the Remaining Duration.
\[ F_i = T_{NOW} + REMDUR \]

Repeat for each ongoing location.
\( S = S + 1 \)
\( S > C \) ?

Give \( X \) batches of Shared Resources to the \( X \) latest crews to use them.
Give * to \( X \) latest \( S_b \)’s

Denote available crews by taking the \( N \) latest locations to be started.
Give * to \( N \) latest ASD’s

To Repetitive Activity Flowchart.

Figure 3.11 The Repetitive Activity Update Flowchart
by adding the remaining duration to the update date \( (F_i = T_{NOW} + \text{REMDUR}) \). This process is repeated for all ongoing locations.

Once all of the ongoing locations have been calculated, it is time to designate the positions of all resources (i.e. crews and shared resources). This can be difficult if there are crews with multiple ongoing locations since the positioning of crews and resources is not clear from the update data given. This data can be asked of the user but would require a substantial increase to the update data inputs monitored in Section 2.5.1. The algorithm therefore sets a convention which allows the process to be performed automatically.

3.7.2 Update Convention for the Repetitive Activity

As all ongoing locations are allowed to continue their progress (Section 3.6.10), the placement of resources does not affect their timing. Also, any duration allowance for handling multiple locations per crew will be reflected in the "remaining duration" input by the user. Therefore, the importance of locating the resources of a repetitive activity lies in the availability of crews and shared resources for the locations yet to be performed. Based on this premise, \( X \) batches of shared resources are located on the \( X \) ongoing locations which finish last. Likewise, the \( Y \) crews are located on the \( Y \) ongoing locations which were started last.
As shared resources must be used by all locations, the above assumption will produce the correct placement, as the latest ongoing locations obviously require the shared resources last (provided they are performed to completion once shared resources are available). The crew assumption also reflects correct resource placement for most situations (i.e. a crew proceeds to its next location even though the current location is not quite finished; a crew skips one location because of technical problems and proceeds to the next). However, it is possible to produce misleading results if a crew simultaneously starts two new locations. The author believes that such a situation does not reflect the continuity by which the activity is modelled (crews progressing a location at a time) and would require an activity review by the user. Note, the crew assumption does allow this situation to be processed and hence maintain a schedule, albeit based on continuity which might not exist.

An example of an updated repetitive activity is shown in Figure 3.12. Note that all links (FS, SS) are included, along with the new links of ongoing locations to TNOW.
LEVEL 1  COMPLETED.
LEVEL 2  REMAINING DURATION = 1 DAY.
LEVEL 4  REMAINING DURATION = 2 DAYS.
LEVEL 5  NOT STARTED.
LEVEL 6  NOT STARTED.

Figure 3.12 Example of an Updated Repetitive Activity

3.8 The Computer Model

The prototype system is implemented on an IBM Personal Computer (PC) at The University of British Columbia. It includes integration of the continuous updating algorithm with the extended definition of Laramee's repetitive activity structure as described in Chapter II. Work is currently underway to incorporate all aspects of this chapter. This entails adopting the author's definition of the repetitive activity process to
take full advantage of the resource levelling flexibility as described in Section 3.7.

3.9 Conclusion

A definitive updating process was evolved through the analysis of two site studies. The goal was to give the user an ongoing picture of the schedule, backed with sufficient information to perform the everyday decisions inherent on a construction site. The process utilized the structure of a CMIS to integrate updating into the company strategy. A continuous updating algorithm was designed to remove unnecessary user involvement and ensure continuous monitoring of the project. During the many continuous updating phases performed it proved to be realistic and robust. The algorithm thus provided a responsive updating process which enhances the heuristic nature by which the construction industry applies decision analysis and control.
CHAPTER IV

FURTHER ISSUES AND CONCLUSION

4.1 Introduction

The purpose of this chapter is to summarize the types of continuity requirements encountered in practice which can be supported by a definition of the repetitive activity and therefore which should be allowed for in the design of a system. Problems of integrating a computerized system into the industry are also discussed. The uses of the system in respect to updating and data storage are examined with a look at possible extensions into the field of "expert" or "smart" systems. Recommendations for future work are then given. The chapter ends with the conclusion to the thesis.

4.2 The Generality of Repetitive Work

Throughout the thesis, the repetitive activity was described in terms of a high-rise project. Such construction lends itself to continuity as each floor can only be built if the floor below is completed. The process starts with the superstructure and then leads naturally to the building services and interior finishing. The repetitive activity was found to be present at successive locations and progress up the structure.
Thus the activity was typically described as regular and omnidirectional. Such a description defines many linear projects such as roadways and pipelines.

The ALRT project was mentioned in Chapter II. It was noted for its non-sequential location pattern which responded to the different design sections of the monorail columns. Thus, the repetitive activity did not follow the linear progression of the monorail but skipped along its length at particular locations. This type of repetitive nature was classified as non-regular yet omnidirectional. Such activities are often seen on linear projects which display two or more characteristics that dictate the construction methodology of the locations. For example, a highway that has varying cut and fill sections, or a transmission line whose tower designs change due to the load characteristics it supports.

Certain projects which display linear work are often constructed from both ends simultaneously and progress to the middle where the job is completed. The purpose is to speed up the project by increasing the manpower yet avoiding the problems of congestion. This method is often used for bridge spans particularly of the suspension design where the large cantilevered sections cause great stresses during the construction phase. It is also found on tunnelling projects where site congestion is a significant constraint. Such projects are found to exhibit both regular and non-regular repetition. Each end usually has its own resources and thus is likely
modelled as an independent repetitive activity.

Certain projects show a repetitive nature even though the locations are not set out in a linear pattern. For example an apartment building composed of one and two bedroom units. There are obviously many repetitive activities to be performed per unit. Therefore, the modelling of the project could be planned as linear by defining locations in the order in which they are to be completed. As the order of the project is not based on its physical layout, such cases often show a greater variety of repetitive activities to optimize shared resources. Other similar projects encountered during this thesis include subdivisions, commercial buildings and sport stadiums.

Repetitive projects are to be found in many aspects of construction. Continuity and rhythm are often significant factors by which a project is planned. It is important to allow the scheduler to model these qualities if computerized scheduling is to be a useful management tool.

4.3 Computer Implementation and the Construction Industry

Changes, while prevalent on the site, are not easy to initiate in the management structure. Most management comprises of personnel who have worked their way up through the ranks. Such people have a keen understanding of the industry, yet often lack any theory by which to manage it.
This profile of construction management was made clear during the initiation of the Critical Path Method in the 1960's. Its complexity and numerous calculations hindered its acceptance. While it has been slowly adopted, few construction personnel understand the theory by which it works, thus leaving it open for abuse. Understanding is only shown at its lowest level of outputs, such as the bar chart and critical path.

Both the updating and repetitive activity definitions described in this thesis, were built with this in mind. The inputs were kept as simple as possible and all assumptions minimized to prevent abuse due to a misunderstanding of the theory.

The repetitive activity requires inputs which the user intuitively considers when formulating the job: production rate, locations and resources. The algorithm performs a matching of inputs to ensure all crews are consistently working when resources are available. The user could perform this function manually if time permitted.

The updating algorithm requires no more inputs than those already used by construction personnel on their daily progress reports. The algorithm bypasses links which were found not to exist and adds new links to ongoing activities to maintain automatic network integrity. All links chosen are based on the most recent data available.
The author contributes the quick acceptance of the above definitions to their simple, yet realistic nature. Construction personnel felt comfortable conversing in the format of the algorithms. The straightforward principles of the process promoted considerable faith in the system.

4.3.1 User Interface

Actual field trials gave the author an insight to initiating computers in construction. A most important premise is that of keeping the user-computer interaction simple. This is partially controlled by the simplicity of the inputs required of the user, but also important is its presentation on the screen. This is known as the user interface.

Important to a successful user interface are concise commands supported where possible by graphics. The language of conversation should be natural to the user. It should include concepts which are already known or can be easily learnt; acronyms should be established terms of the construction industry.

User interface design principles have been outlined by Foley and VanDam (Reference 10), some of which can be applied to the construction industry. These include providing feedback to user inputs (echoing) to promote confidence in the system and
 provision for a correction mechanism to prevent inhibition of the system's use. Consistency in the display of information is very important to gain a feeling of familiarity from the user as is the minimization of user recall needed from screen to screen. An important factor noted by the author and mentioned by Foley and VanDam is the need for "human-factors fine-tuning" or field trials of the system.

4.4 Computers and Data Storage

An issue of updating which has not been addressed is that of the storage requirements of old schedules. Once a revised schedule is accepted the old schedule is no longer used. However, it does contain data still of use to the contractor. It can be used in litigation, management reports and as future reference on new projects.

In litigation it is often necessary to provide the court with a historical account of the project. Lee (Reference 17) states the importance of both the "As-planned" and "As-built" schedules for time impact analysis to settle claims. Such forensic scheduling utilizing network methodology is well accepted in litigation, particularly when the schedules are prepared so that a direct comparison can be made between them. Fisk (Reference 9) outlines the considerable importance of project documentation and emphasizes the need for timely entries into diaries and logs so that information loss is minimized.
Well documented projects greatly strengthen a case in court.

Management reports are often extracted from the schedule to show particular aspects of the project. These often denote the problem areas or deliver a concise overview of the project. Such reports are also useful for support in negotiations and the arbitration processes that can arise. They cover a wide range of collected data and can vary extensively in their detail of time analyses.

Historical data bases are useful for estimating future jobs. While such data is traditionally gained from an estimator's experience, maintaining it on a data base allows data retention when construction personnel change and provides a quick and complete reference tool.

Data storage requirements depend on the demands of the system. It is expected that most contractors would wish to retain enough data for litigation and minimal reporting. This would require the storage of all documentation and all revised schedules. Periodic updated schedules should also be stored to show overall site progress and to highlight specific areas of interest. Any further data base for purposes of estimating or detailed reporting will depend on management and their desired knowledge base.
4.5 Expert Systems

A promising emerging hybrid of management systems is the expert system. It essentially performs all functions of the CMIS, plus has the ability to analyze and offer suggestions on problem areas which arise based on knowledge gained from construction experts (Reference 13). Such a system could establish itself in the construction industry where it could perhaps be used to balance off the lack of theory of construction personnel who use CPM base systems.

Fenves (Reference 8) recognized the use of expert systems for monitoring the construction process. Situations which it could help include high consumption of float, possible procurement delays, and update logic which does not conform to scheduled logic. Such situations are easy to monitor in the theoretical sense but can elude the user. It could thus provide foresight of potential problems and point out areas where the model does not fit well with the actual project data.

The updating algorithm described in this thesis could be said to fit some of the characteristics of an expert system as it dictates the validity of logic links based on a set of rules developed from the construction industry. It thus provides the user with an update status and points out areas which must be reviewed in order to give an an acceptable outcome. Such inference allows the user to benefit from updating procedures used by experienced personnel and invokes organization and
formulation of the thought process used.

An expert system demands criteria by which it can assess the current situation. Apart from the theoretical points outlined above, it could be provided with user specified criteria. This would include a given variance in production rates of activities, specified allowable slippage of a milestone, forecasting of activity completion based on earlier projects or extrapolated values.

The expert system offers many opportunities to the construction industry. Its assessments could offer a robust analysis and be fine-tuned by the individual user. Construction personnel could benefit from expert analysis of the many heuristic situations that arise. It is a tool which would force researchers to take a closer look at how a construction project is actually run and greatly aid in making computers more compatible with the construction industry.

4.6 Recommendations for Future Work

The general definition of the repetitive activity is not yet implemented into the prototype system at the University of British Columbia. There is a need to test this definition to ensure it can model the majority of work patterns found in construction. Its integration with the continuous updating process must also be completed. This should not prove to be
difficult as the updating has already been shown to work well with the extended version of Laramée's repetitive structure.

The above work is expected to be implemented into the system in the near future. Although this will complete the focus of thesis, the author suggests some new areas of interest by which the useability of computerized scheduling may be enhanced.

A most important area yet to be captured in construction management is that of user interface. Of particular importance is the exploitation of graphics to portray inputs and the project as a whole. This is a difficult task for time centres tend to be very large and difficult to display on a screen. Repetitive activities can benefit greatly by a graphic display of the structure before it is incorporated into the network. The user can thus quickly follow each crew's movement along the project and pick out any problems which may be apparent. Graphic displays of sections of the network would also help distinguish relationships between specific groups of activities. This area requires considerable work to capture the format by which the construction industry may best gain. It is expected that some problems of graphic presentation may be solved with the future advancements in computer graphics.

A data storage system to save all progress data could be developed and stored in a standard base for the construction industry to take full advantage of the update algorithm. Computer technology is quickly allowing large blocks of data to
be stored at minimal cost. Such knowledge will provide the experience of finished projects to be applied to future work through well supported costing and scheduling data and through the enhancement of the decision making process.

Developing an expert system based on prevalent problem areas in construction is an area that needs immediate attention. Its potential to focus the user's concentration and confidence on the computer system supports its development. Work must be focused on the problem areas of scheduling. Experts must be found to apply their knowledge to creating problem solving techniques. Such methodology is difficult to generate with current computer hardware. Future realization of parallel processing will greatly aid in its implementation.

4.7 Conclusion

The goal of this thesis was to promote the useability of computerized scheduling systems in the construction industry. This goal has been met through the development of two algorithms.

The definition of the general repetitive structure provides construction personnel with a tool by which to model the process of repetition found in many construction sites. This allows the user to schedule the work flow patterns by which the rhythm of a project is controlled.
The definition of the continuous updating algorithm provides the user with a schedule capable of monitoring the daily progress of the construction site. It demands no more inputs than required in a typical daily progress report. The algorithm thus allows the user to respond to everyday changes in the construction site and implement effective control.

Both algorithms have been based on a simple heuristic approach to promote their integration into the construction industry. Care was taken to maintain a user friendly system through extensive field observations and testing to understand the basics by which projects are formulated and controlled.

The site studies show the algorithms to be both realistic and robust. They have been found to fit well with construction management philosophy. The thesis thus creates two useable tools which advance the useability of computerized scheduling in the construction industry.
BIBLIOGRAPHY


5. Canada Evidence Act, Revised Statute, Chapter 307, Section 1.


