CHAPTER 1

1. INTRODUCTION

1.1 INTRODUCTION

This research is concerned with the study and analysis of multi-agent coordination for project scheduling and proposes a model for resource allocation based on priority rules in order to minimise the project duration by overlapping the activities of the project. Traditional planning deals with the problem of finding activities to satisfy a given goal and scheduling solves the problem of allocating limited resources to activities and to limited time. In particular, the planning problem is solved first, which generates a set of activities, and the scheduling problem is solved next, which allocates the resources and time to the activities. A project should be managed through the integration of activities that can easily share the information. The integration can be achieved by relating the activities of the project and hence improved coordination and flow of project and resource information between activities become essential to minimise the project makespan.

In order to reduce the project duration, to utilise the resources in an optimum way and to meet the changes, project planning and scheduling need to be integrated with Concurrent Engineering (CE) principles which in turn yield Integrated Project Management (IPM). CE can be used to improve the timing of access to resources which is particularly critical to a project because it can directly affect the project duration. Coordination is a crucial problem in CE and coordination is then defined as the process of managing dependencies between activities. Project Evaluation and Review Technique (PERT) and Resource-Constrained Project Scheduling Problem
(RCPSP) are the methods used for planning and control of a project. But, complex project that has overlapped and branch-merge activity (task) pattern needs flexible and robust project management. Multi-Agent System (MAS) is a technique to support concurrent engineering in project management and there arises a need for resource allocation among competitive multiple agents. Therefore, an integrated planning and scheduling algorithm has been developed and employed for better resource allocation and task scheduling to minimise the project makespan.

This chapter presents an introduction to project management and concurrent engineering and lists the various factors of activity dependencies and precedence relations in concurrent engineering. It also analysis the problem of resource allocation for project scheduling and it gives an introduction to agent-based planning and scheduling. The problem of constraint on resources that enables the execution of tasks in project scheduling is elaborated. Then, resource and precedence constraints and the need of resource management in multi-agent system have been introduced. Objectives are formed after the detailed discussion.

1.2 PROJECT MANAGEMENT

Advanced representation and solution techniques in project planning and scheduling received significant attention during the past decades from the part of research communities and the industry. Project Management (PM) has been developed from different fields of application including construction, engineering and software development. Traditionally, product development process consists of a set of activities to be performed sequentially. Quick product development and reducing time to market are the major challenges for all manufacturing industries in competitive market.
Project management is defined as managing and directing time, material, personnel and costs to complete a project in an orderly manner and to meet the established objectives of time, costs and technical and / or service results (Spinner 1997). Project management activities can be adopted with flexibility and hence necessity arises for resource allocation and coordination in project management. Traditional project management approaches emphasise the importance of scheduling, while careful allocation of resources across projects is critical and has a direct influence on the project quality and time management (Shen et al. 2005).

The project management paradigm has been shifting from a restrictive (traditional) management approach towards a more collaborative approach (Cleetus et al. 1996; Evaristo and Fenema. 1999; Maurer 1996; Romano et al. 2000). In the more restrictive project management approach, the focus was on “management” or “control” which implied a top-down view of how projects are conducted and controlled (Cleetus et al. 1996; Augustine et al. 2005). The communication, coordination and tracking of project activities become key issues for project success. Researchers in the PM field have indicated that PM systems should support basic project management functions and additional functions such as collaboration in project management (Clarke 1999); information sharing and file management (Weiser and Morrison 1998) and project process management (Turner. 2000). Chen et al. (2006) provided a collaborative PM approach to manage projects and proposed a framework for the collaborative PM software development. A problem in project management involves the allocation of limited resources to the individual tasks comprising the project. In complex project management, it is desirable to make the best use of available resources to improve the efficiency of project execution.
1.2.1 Planning

Planning is an action selection phase where possible actions to be performed on the projects are selected and partially ordered to reach the desired goal(s). The resulting plan (also called project) consists of activities that must be carried out to achieve a predefined goal. Planning generally refers to the process of deciding what to do, that is to say, the process of transforming strategic objective into executable activity networks and deals with goals, states and actions. Due to the complexity and the variety of planning problems, most researchers consider static problems in which activities are known in advance and constraints are fixed. The process of developing the project plan and the control is analysed and the impact of combining planning and team building on project’s success has been investigated by Michael Thomas et al. (2008). The planning functions in make-to-order manufacturing environments are generally described by the three-level hierarchy presented. The levels of decision making are called strategic, tactical and operational (Kovács 2005). Planning on the operational level unfolds the first segments of the plan into detailed resource assignments and operation sequences. Based on the information, the required resources and the number of activities are determined.

Once the project is defined, one of the main tasks in project management is the scheduling of this project that is the temporal arrangement of the activities. Thus, after the planning phase, a scheduling phase is needed where enough resources are assigned to ensure the successful execution of the project. Scheduling is performed on a detailed problem representation, for individual operations, with respect to fixed capacities.

1.2.2 Scheduling

A schedule is a plan of procedure for a project and scheduling is the action of making a schedule. Scheduling is the allocation of resources to activities over time.
and effective scheduling can improve project performance and the utilisation of resources. The reasons for lengthy project duration and the causes for extension of project completion time in case of international development projects have been identified by Kamrul A and Indira Gunawan (2010). Deterministic project scheduling models assume that the processing time of a task is fixed and known in advance of scheduling. Scheduling and sequencing is concerned with the optimum allocation of resources over time. Scheduling deals with defining which activities are to be performed at a particular time and sequencing concerns the ordering in which the activities have to be performed.

Makespan minimisation is the most researched and widely applied objective in the project scheduling domain. The makespan is defined as the time span between the start and the end of the project. Since the start of the project is usually assumed to be at \( t = 0 \), minimising the makespan reduces to minimising the maximum of the finish times of all activities and it is a regular performance measure. Scheduling is responsible for making detailed, executable schedules that achieve the goals set by project plans. Therefore, scheduling has to assign resources to tasks as well as to determine their order of execution. Beyond satisfying precedence and resource constraints, the solution should approach optimality with respect to some optimisation criterion. Close-to-optimal solution of scheduling problems requires expressive and flexible models and efficient, customised solution methods.

The scheduling process decides when and how, i.e. which resources to use to execute various activities and over what time frames. Moreover, a planning decision—an introduction of an activity is tightly coupled with a scheduling decision—an allocation of the activities to time and resources. Hence, planning and scheduling need to be integrated which gives the way to introduce concurrent engineering principles.
CE is a philosophy that focuses on integrating and mapping processes to shorten the development cycle of the product (Skalak et al. 1997).

1.3 CONCURRENT ENGINEERING

Concurrent engineering is an approach for improving the development of new products. It is extensively used and many benefits have been reported (Miranda and Morries 2002). It has been noticed that many of the advantages attributed to CE might also be obtained through a well-implemented project management approach (Cleland and Ireland 2000). Recent trends in scheduling attempt to fill the gap between scheduling in theory and scheduling in practice, with the aim to give answer (respond) to market demand with more efficient method to solve complex scheduling problems. The importance of good scheduling strategies in project management leads to the need of developing efficient methods to solve complex scheduling problems.

CE is a widely accepted philosophy to shorten the product development life cycle and CE is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to make the developers to consider all elements of the product life cycle from conceptual through to disposal, including quality, cost, schedule, and user requirements. According to the report on concurrent engineering implementation, it has significantly improved the time to market, quality, costs, etc., in the manufacturing industry. The capture, change and transfer of evolving resource information become the main requirements for enabling concurrent engineering practices in project management. Devi Tirupathi and Rajat Roy (1997) have identified the need for an integrated information model for effective project management. A
model of activities and activity networks had been proposed and developed by them to configure and manage the project information under concurrent engineering. The capabilities of Dependency Structure Matrix (DSM) based methods are used to manage projects for reduced schedule risk and shorter cycle time (Browning, 1998).

Loch and Terwiesch (1998) presented an analytical model of concurrent engineering which combines the decisions of overlapping and dependency between tasks with the goal of minimising time-to-market. Ji-Hong and Chang (2002) proposed and developed an approach to provide an optimisation scheduling algorithm for concurrent activities with a priority function and provided the necessary basis for scheduling activities in CE. The activities in an engineering product development life cycle are usually implemented sequentially and independently. Traditional project management tools provide excellent recording, reporting and scheduling functions, such as Gantt charts, CPM (Critical Path Method), and PERT (Program Evaluation and Review Technique), but they fail to provide higher-order capabilities. Even with the advancement of research on project management, some of the important practical issues have not been taken into account or satisfactorily resolved, such as the resource allocation, the dependency among resources and activities.

1.3.1 Activity dependencies

The project scheduling problems are made of activities, resources, precedence relations and performance measures and a project consists of a number of activities also known as jobs, operations and tasks. In order to complete the project successfully, each activity has to be processed which determines the duration of the activity (time taken to complete the activity). Consider two tasks, labeled \( T_1 \) and \( T_2 \).
Figure 1.1 shows directed graphs (digraphs) of two possible ways in which the two tasks can be related. If task T2 simply requires a resource from task T1, then the two tasks are dependent and the tasks are executed one by one. On the other hand, the two would be entirely independent if tasks T1 and T2 could be performed simultaneously with no interaction. Coordinating either the dependent (series) tasks or the independent (parallel) tasks with limited resources is quite complex. Certainly, with no limitation on resources, the parallel tasks can be completed more quickly.

An overlap pattern represents the case when activity T2 can begin from the upstream activity T1 and proceed with the activity. The degree of overlapping is the way to represent how early activity T2 can start before the completion of activity T1. Obviously, a no-overlap pattern has zero degree of overlapping. There are several different types of collaboration among multiple activities when solving and making decisions: feedback, interaction, cycle, communication and branching and merging.

In project scheduling, constraints and coordination is a two-stage operation where upstream activities provide resources of the project to downstream activities. The activities in projects where operations in the two-stages (overlapping and coupled) are called concurrent activities and concurrency involves the use of resource
and precedence information for coordination between upstream and downstream activities (Krishnan and Ulrich 2001). This mode of concurrency does not distinguish between the degree of concurrency, whether all or part of the operations are concurrent, but rather concentrates on the interdependence of upstream and downstream activities. Victoria and Barrie (2005) examined whether cooperative planning and uncertainty affect the magnitude of rework in concurrent engineering project with upstream and downstream operations and found that project delay is primarily influenced by the magnitude of downstream rework and downstream delay. Therefore, the aspects of concurrent activities that affect project completion are: resource-constraints, precedence-constraints and coordination among interdependent activities.

1.3.2 Precedence relations

A project is specified as a finite set of activities that require the resources for processing and a project is a complex network of many relationships among activities which are called as patterns. In general, the relationship between two activities can be independent or dependent and there are four basic types of dependency between activities such as Finish-Start, Finish-Finish, Start-Finish and Start-Start. With a Start-Start (SS) dependency, the predecessor must start before the successor can start. In this type of dependency, the successor and predecessor usually overlap.

In branching and merging, a branching represents the behaviour which one activity is separated into one or more according to logical and chronological constraints. On the other hand, merging represents the behaviour when the results of several activities are combined to form one set of outputs. So, branching always pairs
with merging. Branching (or merging) patterns can be divided into AND, Selective OR (SOR), or exclusive OR (XOR) types based on how many activities are selected in a particular pattern (Jun et al. 2005). When all parallel activities are selected, it can be called as AND routing type. When only one alternative activity is selected, the pattern can be called as XOR routing type. When multiple (but not all) activities are selected, it is of SOR type.

1.3.3 Resource management

The current challenges in resources management are less characterized by solving well-defined problems but rather by building flexible and sustainable resource management regimes. The responsibility of the resource management is to provide information about the types and availability of resources and to track their usage. Resource management and scheduling is the way of distributing resources among tasks (Kolisch and Padman 1997). This maximises throughout and efficiently utilises the available resources. A resource may be either continuous or discrete and may be treated as either divisible or indivisible. A sharable resource can be allocated to a number of different tasks at the same time. A resource may be consumable in the sense that the task holding the resource may use up the resource during execution. Therefore, the resource management plays an important role during the project execution to minimise the project duration which is desirable.

Three important and intricately interconnected tasks in the resource management are: modelling the resources, identifying and retrieving the resources and scheduling the resources (Ghosh et al. 2004). Usually, the modelling task takes place before simulation, potentially in parallel with the process definition work. Modelling or specification of resources is dependent on the domain for which a process is being
defined. Identification of resource can be achieved based on the status of resources such as beginning, intermediate and end. Retrieving the resource allocates the requested resource to the task and then, the resources are scheduled to the task based on various policies. If a resource is scheduled, it can be made available again for other tasks to use by releasing it. A resource is released on the completion of an activity for which it is scheduled. A resource is also released if the duration for which it is scheduled expires and it is not retrieved for use at this time. Hence, the resource management problem has two main components: ① to allocate the right resources at the right time, and ② to distribute the resources to the right tasks.

The performance of project management systems varies widely depending upon the availability of needed resources and precedence relations. Thus the precise specification of resources needed by, and available to, a system is an important basis for reasoning about and optimising system behaviour. Previous resource models used in project management and control flow have lacked the rigor to support reasoning and optimisation. Researchers demonstrated the need for the resources in coordinating and optimising the execution of project activities. Clearly, the abundance of resources can enable execution of tasks in parallel, thereby speeding up accomplishment of larger goals. This same phenomenon is also clearly observable in broader class of coordination. The lack of resources causes contention and the need arises for some tasks to wait for others to complete. Often, potential delays can be avoided or reduced by using resource analysis to identify ways in which tasks can be made to execute in parallel that avoid resource contention. With this understanding, it is possible to identify in which situations only a particular resource will suffice and in which situations any of a class of resources may be acceptable for the task. This
identification of critical resource needs can also facilitate the identification of likely bottlenecks in the execution of a task.

1.3.4 Resource-constrained project scheduling

The process of a complex project is complicated due to constraints like geometric structure constraints, design or manufacturing specifications, information constraints, resources constraints and so on. These constraints are imposed on the design tasks and make them inter-related. Because of the complicated dependencies, it is very difficult to carry out these tasks concurrently. A task can be started either when the precedence constraints are satisfied or all the constraints including precedence and resource imposed on the task are satisfied. Resource constraints are one of the most important factors which must be considered in project scheduling. Various resource allocation techniques have been developed to satisfy resource constraints and optimal use of limited resources.

Basic approaches to address the constrained resource allocation problem are optimising methods and heuristic methods. Optimising methods i.e. linear programming, find the best allocation of resources to tasks but are limited in the size of problems whereas heuristic methods are realistic approaches that may identify feasible solutions to the problem and they essentially use simple priority rules such as shortest task first and longest task first to allocate resources.

The resource-constrained project scheduling problem involves the scheduling of project network activities subject to precedence and resource constraints under the objective of minimising the project makespan. A branch-and-bound procedure (Demeulemeester and Herroelen 1992; Brucker 2000) had been developed for
optimally solving the basic RCPSP, where the precedence relations are of the finish-start type with zero time lag, and the limit on the availability of renewable resources remains constant over time. Chen (2004) developed a column generation based branch and bound method for finding optimal solutions for NP-hard scheduling problems and the developed algorithms are capable of solving the problems with a medium size. An Enumerative Branch-And-Cut procedure (EBAC) has been presented by Germou and Jonathan (2005) for minimizing the total project duration of a construction project under multiple resource constraints based on an enumeration tree. Instead of searching for better schedule alternatives as many exact algorithms do, EBAC takes the feasible schedule alternatives (i.e., satisfying both the sequence and resource constraints).

### 1.3.5 Resource constraints

While planning is concerned with synthesizing actions which achieve an objective, scheduling deals with the allocation of actions over time under resource and complex time constraints. Actions in scheduling are usually referred to as activities or tasks. The activities have start and end times; these time points are bound by constraints asserting requirements such as “activity A much finish atleast d time units before activity B begins”, or “activity A must finish no later than d time units after B ends”. Such constraints are known as, respectively, minimum and maximum time lags (or generalised precedence relations). Activities require resources, and the problem of allocating activities in time involves taking into account the precedence constraints as well as the limited capacities of the resources. In addition, the optimisation problem in scheduling can have a variety of objective functions, such as minimising the latest end time of the activities (makespan) or minimising the start time of all activities (tardiness).
Activities release and require resources. If an activity requires a resource then there must exist another activity that releases this resource and vice versa. It is possible to have several requires and several releases of the resources and hence a many-to-many relationship exists between the activities. The above release/require relation is modelled as a resource dependency because it describes the dependencies between their resources (Figure 1.2). The resource dependency specifies which activities are connected and what is the time-lag between the activities.

Resources flow (arrows) between the activities (rectangles) and the resource flow defines resource constraints. CE need to be introduced within the resource constraints of the project and a phased implementation plan that matched the resource constraints has been recommended to improve new product development process (Estoritio 2002). Resource constraint is one of the major constraints that directly impacts on the task execution and resource constraint that causes the resource dependencies among the task executions is considered.
A precedence constraint states that the connected activities have to be executed in the given order, in distinct time units. Connectivity and cyclicity are the two criteria that have been proposed to distinguish between different cases of precedence and resource relations.

A precedence relation is called, isolating iff only tasks belonging to the same sub-set of tasks $T_u$ are related, i.e. for all tasks $T_i \in T_u$, $T_j \in T_u$, $T_i < T_j$ implies $u = u'$ and it is called connecting iff at least two tasks from different subsets of task $T_u$ are related, i.e. there exist task $T_i \in T_u$, $T_j \in T_u$, $T_i < T_j$ such that $u \neq u'$. A precedence relation is called, acyclic iff it forms no cycle, i.e. there exists no subset of tasks $\{T_1, \ldots, T_n\} \subseteq T$, where $T_1 < T_2 < T_3, \ldots, T_{n-1} < T_n$ and $T_1 \neq T_n$ and it is called cyclic iff it forms at least one cycle, i.e. there exists a subset of tasks $\{T_1, \ldots, T_n\} \subseteq T$, where $T_1 < T_2 < T_3, \ldots, T_{n-1} < T_n$ and $T_1 = T_n$.

Formally, resource-constrained project scheduling is the following.

Given: A set of tasks $T$, a set of resources $R$, a capacity function $C: R \to N$, a duration function $D: T \to N$, a utilization function $U: T \times R \to N$.

Find: An assignment of start times $S: T \to N$ satisfying the following.

1. Precedence Constraints: If $T_1$ precedes $T_2$ in the partial order $P$, then $S(T_1) + D(T_1) \leq S(T_2)$.

2. Resource Constraints: For any time $t$, let running $(t) = \{T | S(T) \leq t < S(T) + D(T)\}$ Then for all times $t$, and all $r \in R$, $\sum_{T \in \text{running}(t)} U(T, r) \leq C(r)$.
1.4 MULTI-AGENT SYSTEM

Coordination is a central issue in multi-agent system and mathematical and operational research techniques are being used to realise coordination. Coordination of different activities should be considered in decision making to complete a project quickly and the activities have to be synchronized and allowed to perform concurrently. Concurrent activities introduce several issues about how to maintain scheduling across the agent set for effective performance. Decentralised multi-agent system comprises agents who act autonomously based on local information and achieving coordination in such system is nontrivial, but it is essential in most applications.

An agent is a piece of autonomous software created by and acting on behalf of a user (or some other agent). It is set up to achieve a modest goal, with the characteristics of autonomy, interaction, reactivity to environment, as well as pro-activeness. Agents are autonomous software systems that are intended to describe the behaviour. An enormous advantage of agent-based modelling is the ability to assess the plausibility of the behaviour of agents, the ways in which the agents interact and the consequences of that behaviour and interaction. Effective coordination is critical to the success of project since the distributed activities are highly interdependent due to sharing of resource and input-output relationships.

The fundamental model of project is a set of activities or tasks which are to be performed in an overlapping fashion in order to achieve the goal. Each activity can be viewed as an agent (autonomous program) and the process can be carried out by using multi-agent system. The multi-agent system is a coordinated execution of tasks to
carry out a process and it is successful only if the agents in the system are ready to cooperate. Yuhong et al. (2000) used multi-agent system as a technique to support project management in distributed environment and presented methods to schedule activities and resolve resource conflict by message exchanging and negotiation among agents. A multi-agent system approach is much better suited for autonomic computing (Tesauro et al. 2004). Jennings (2004) advocates an agent-based approach based on decomposing problem in terms of decentralised, autonomous agents that can engage in flexible, high level interactions.

1.5 MOTIVATION

When estimating the project completion time in project management, previous research often focused on one factor and assumed the other factors causing little effects on the overall project duration. A common realistic constraint in project scheduling which requires the coordination of multiple agents is the limitation of resources for execution. In this case, concurrency is affected by limited resources and so, there is a need for modifying the strategies of the planning / coordination problem. Specifically, the part of this research work tackles planning and scheduling problem in two contexts, namely when planning is carried out centrally and scheduling is done in the context of distributed multi-agent coordination. Therefore, an integrated planning and scheduling architecture is presented and employed to explore the multi-agent coordination problems in function of their resource-related characteristics.

One of the major goals is to shorten the life cycle by increasing more execution overlaps of the tasks involved in the project. Aiming at this goal, this work focuses on the question “When does the task require the resource from its upstream
tasks and when the required resource will be available?" The resource constraint-driven execution semantics is proposed to handle this question. This constraint-driven semantics loose the time strictness of the resource constraints that imposed on the tasks, which enable the task and it can be started earlier as soon as the required resource is available rather than after the upstream tasks are completely finished by keeping the precedence relations.

For example, when the parallel tasks $T_2$ and $T_3$ requires a resource $r$ which is to be released by the task $T_1$, and then $T_2$ and $T_3$ cannot be started before task $T_1$ is finished because of precedence relation. The resource $r$ is an indivisible resource which is to be allocated for only one task at a time. Although the precedence constraints allow, tasks $T_2$ and $T_3$ cannot be executed concurrently because of this resource constraint. Therefore tasks $T_2$ and $T_3$ are forced to execute sequentially one by one (either $T_2$ followed by $T_3$ or $T_3$ followed by $T_2$). However, if the dependency is analysed, the following two questions arise:

- When does task $T_2$ release resource $r$?
- When does the task $T_3$ need this resource $r$?

If the resource $r$ is released by task $T_2$ not in the end or required by task $T_3$ not in the beginning, why not it can be overlapped between parts of the execution of these two tasks? Therefore, thoroughly analysing the resource constraints based on time, certain overlapping of tasks executions can be obtained. The work develops model addressing the two questions of ① how to overlap activities depending on the project characteristics, and ② how to coordinate the concurrent activities.
1.6 OBJECTIVES

Crucial for the multi-agent coordination in project scheduling under concurrent engineering environment is the availability of an effective algorithm for resource allocation which would reduce the project makespan. The effective resource allocation process depends on task interdependencies, resource interdependencies and constraints. The main objectives of this research have been to:

1. Study and analyse activity dependency, resource allocation and project scheduling in a multi-agent project management environment.

2. Design and develop an algorithm to identify the interdependencies among tasks, overlapping among activities due to resource constraints in a multi-agent project coordination.

3. Design and develop an algorithm to reschedule the activities to achieve both optimum resource utilisation and project time.

4. Test the validity of the algorithms by developing a prototype to demonstrate the primary features of the project scheduling especially resource allocation in a multi-agent coordination system.

1.7 THESIS ORGANISATION

This thesis describes the multi-agent project coordination for the project scheduling through resource allocation based on priority rules by overlapping the activities, design of an algorithm and development of a prototype and the tests that have been conducted using case studies to demonstrate the concepts designed and developed.
1.7.1 Review of existing project scheduling methods

Chapters 2, 3 and 4 provide a basic knowledge for understanding resource-constrained project scheduling and multi-agent system and the necessity of resource allocation for multi-agent coordination in project scheduling.

**Chapter 2** presents an introduction to project management and concurrent engineering. It analyses the problem of resource allocation for project scheduling and gives an introduction to various scheduling strategies. This chapter elaborates various project management tools such as PERT, Gantt, GERT and Petri nets. Then, the above project management tools are compared based on the execution of tasks in project scheduling.

**Chapter 3** reviews in detail about the existing scheduling procedures that have been developed to facilitate the design and implementation of the resource allocation for project scheduling.

**Chapter 4** describes the mechanisms used for allocating resources among agents in Multi-agent System. The chapter focuses on the techniques used for allocating resources among cooperative agents. This chapter also presents the ways of decomposing the tasks into sub-tasks along with agent architecture and introduces the uses of game theory for resource allocation in competitive multi-agent system.

1.7.2 Analysis and modelling

**Chapter 5** deals with the modelling of resource-allocation used during task execution. Two models - one captures the resource-dependent tasks where the resource to a task is defined as the value of time period of the resource requirement and resource release;
and the other model captures the resource-dependent tasks and allocates the resources to tasks based on resource allocation game.

Chapter 6 introduces a procedure for project scheduling that has been developed and then describes the development of algorithms such as task dependency identification, resource dependency identification, task overlapping, resource allocation and project makespan computation.

1.7.3 Prototype design and development

Chapter 7 presents the design and development of the prototype for project scheduling and describes the basic mechanisms employed by the scheduling component. This chapter discusses the functions that are employed for task overlapping and resource allocation based on priority rules. It also extends the approach developed previously and demonstrates the use of resource allocation game.

Chapter 8 presents experimental results. The performance of the scheduling algorithm has been examined and the results are compared with the traditional project scheduling tool such as Critical Path Method. A detailed case study has been presented and the results are discussed.

Chapter 9 summarises the research, describes the achievements and limitations, and presents recommendations for future research.