

EXAMPLES OF REQUIREMENTS DEFINITION

Note: These correspond to previous students' projects and are provided for illustration and criticism. It is not assumed that these requirements are all adequate.

Project:

A Tool for Automatic Generation and Validation of PERT Diagrams with Expression Notation

Aim:

The basic target aim in the dissertation project is to produce a software tool that is capable for the solving of difficulties in both the construction and validation of PERT diagrams, in particular the automatic generation of A-o-A PERT diagrams, the notation expression that represents it, and the creation of PERT diagrams and validation of. In simple terms, would be a program written that is distributable to both businesses and the public, and is primarily capable of automatic generation A-o-A PERT diagrams (with the use of a notation expression) as well as the ability to create and validate them. A possible extension in the scope to the dissertation project is to enable the program to carry the equivalent ability for activity-on-arrow (A-o-N) PERT diagrams.

Functional Requirements:

- Automatic A-o-A PERT diagram generation – The program written should carry the ability to automatically generate A-o-A PERT diagrams from an activity-and-precedence table, whose data content are manual inputs by the user. EXTENSION: Program should also be able to auto-generate A-o-N PERT diagrams from an AAP (Activity and Precedence) table.
- Notation Expression Generation – The activity-and-precedence table should first be translated into a notation expression format before it is directly made into a PERT diagram. The notation expression should be capable of being transformed into a PERT diagram or back into an activity-and-precedence table.
- Critical Path Calculation – Upon the auto-generation of the relevant PERT diagram, the critical path of the program should also be calculated, and displayed with an option to the user's preference.
- Avoidance of Overlapping – The PERT diagrams produced should not overlap at all unless they are not planar. The generation process of the PERT diagram should implement this; diagrams should be "made not to contain overlaps" not "made then overlaps removed". An algorithm should be used for the diagram generation process.
- Creation of A-o-A PERT diagrams – The program should allow users to create A-o-A PERT diagrams in order for the validation of the diagrams.
- Validation of A-o-A PERT diagrams – The program should be able to validate whether a created diagram is in a correct form. Should the diagram be in an incorrect state, the program should produce a list of items showing the details about where and what type of errors were found. EXTENSION: Program should also be able to validate A-o-N PERT Diagrams.
- Custom Drawing PERT diagrams – The program should allow users to be able to directly draw PERT diagrams to their preference.

Non-functional requirements:

- The program should be user friendly and aesthetically pleasing, appropriately designed for standard business and home software.
- Program instructions should easy to understand, abstract but clear.
- A 'help' section should be available, to assist users when usage difficulties are faced using the program. Additional assistance should be displayed where necessary.
- Avoid program confusion.

Project:

A Decision Support System for Office Space Management

Aim:

In any sizeable organisation, managing the massive amount resources can become an expensive and a time consuming task. One of these many resources is the building(s) which must be adequate to both house and allow efficient operation of all entities, an entity being a member of staff or a facility. Allocating enough building space for every required entity in an organisation can become problematic with the many restrictions and limited space. Many institutes have planning and layout guidelines which must be followed, plus each room may also have special requirements. Being able to plan a suitable layout can take considerable time and any future changes will consume yet more time and money. Making an optimised floor plan that wastes the minimal amount of space, can accommodate future changes in both staff and facilities and meets all restrictions is the problem.

Functional Requirements:

- 1) Creation and editing of 2D floor plans for one building floor using straight lines only
- 2) Creation and editing of entities
- 3) Creation and editing of constraints
- 4) A user interface encompassing the floor plan, entities, and constraints
- 5) A print option or an exportable format suitable for printing which can produce a document suitable for use as a floor layout diagram
- 6) The ability to manually edit entities assigned rooms
- 7) The ability to manually assign room names
- 8) The ability to save and load Space Allocation data
- 9) Automatically edit rooms and assign entities based upon the constraints, entities, and the floor plan

Non-Functional Requirements:

- 1) Runs on Microsoft Windows XP
- 2) Allow quick, precise, understandable, and easily accessible editing of in-program data
- 3) Provide an accessible and usable User Interface to someone who has basic computer knowledge
- 4) Has a data representation that allows AI algorithms to propose an optimised floor plan
- 5) To provide extensive documentation for both a user and a systems developer
- 6) Have a clean object oriented design allowing good maintainability

Project:

Building Space Management – Developing a system to aid in its optimisation, with the assistance of Haringey Council

Aim:

The dissertation seeks to address ways of maximising the efficiency in the use of space within local authority buildings. Local authorities own a varied portfolio of buildings and the audit commission have identified assets that are estimated to be “valued at around £78 billion...costing £5 billion a year to run and manage...including over 21,000 schools, 3,800 libraries, 1,800 leisure facilities and 2,200 community centres” (Audit Commission, 2000). Consequently any failure to manage this space effectively will result in considerable amount of money being wasted which could have been channelled into front-line services.

Outline Requirements Specification:

Note: Requirements specification is not yet fully developed.

System Overview

- To provide a Decision Support Tool for the management and allocation of office space within a portfolio of buildings.

Intended Users

- To be operated by up to four non-concurrent users

Computer Experience

- Competent day-to-day users of mainstream business software (e.g. Microsoft Office). Level 1

Main Problems to be addressed

- Formalisation of data handling.
- Transferring of knowledge and information from the human element into computer systems as far as practical.
- Providing closed looped system.

Level of Security

- Basic level password protection
- Conform to Data Protection Act

Hardware Constraints

- Capable of operation on both thick and thin client systems.

Software Constraints

- Run on a Microsoft Windows XP operating system
- Must conform to Haringey Councils software policy (TBC)

Number of Records

- Currently 150 buildings
- Capacity to expand to up to 500 buildings

Accessibility

- Accessible from remote location
- No concurrency required

Technical Support Required

- Capable of being operated and managed to on-site IT support and technicians 11

History

- Require Integration of data in existing systems and process: Excel and Hard-Copy forms

Business Processes

- See Flowchart appendix 2 to be re-engineered to flowchart in appendix 3.

Functional Requirements

- Store Space and Building Information as a “model”
- Capable of carrying out “what if” analysis.
- Ability to produce space distribution output
- Generation of Key Performance Indicators

Data to be Held

- Building Information in terms of space, services and location.

Project:

An Online Implementation of Diplomacy – the Game

Aim:

Diplomacy is a classic game of political intrigue and military power. Typically, there are between 2 and 7 players and the objective is to achieve dominance of Europe. The rules of the game are easy but playing diplomacy requires careful analysis, and involves negotiation between the players. The goal of this project is to develop a computer version of the diplomacy game that is interactive, visually attractive and incorporates an effective tool for the negotiation between players. The project also involves the implementation of known diplomacy tactics so that the computer can act as one of the players in the game. Given its nature, it is required that the game is implemented in such a way that each player can access the program from different locations.

Functional Requirements:

- Host multiple games simultaneously
- Users able to playing more than one game at a time
- Individual profiles for each user
- Track the history of a game (such as orders and messages)
- Handle the creation, continuation and ending of a game
- Implement the 2000 rule book rules
- Handle paradoxes (both solvable and unsolvable) gracefully
- Validate an order without knowledge of other orders
- Allow controlled communication (negotiation) between players

Non-functional requirements:

- Accessible to users on the internet with just a web browser
- Platform and browser independent client
- Server works with Apache Tomcat and MySQL Database
- Flexibility in both the rules and map, e.g. ability to implement more than one map, and for the adjudicator to use more than one set of rules
- Easy to use interface for a Diplomacy novice, one that is useable and visually attractive
- Designed to replace the old mailing system of game play
- Have an open format for data storage and interaction, allows easy extension by third parties