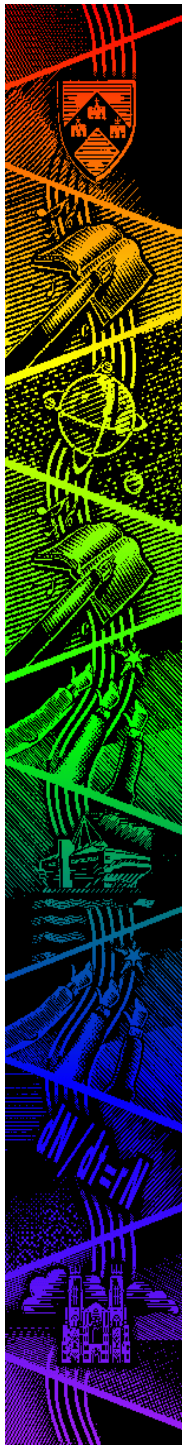
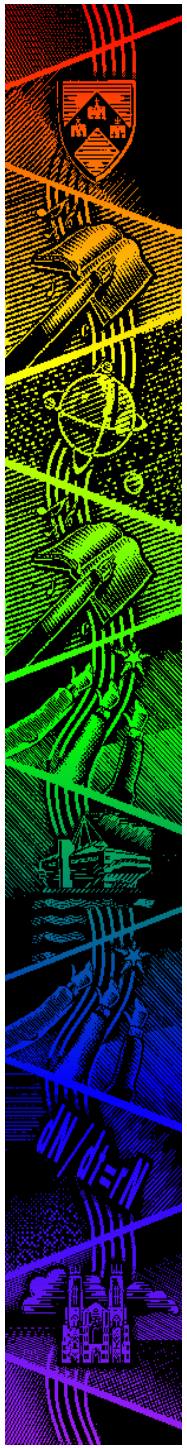


Semi-formal Methodologies

The first generation methodology

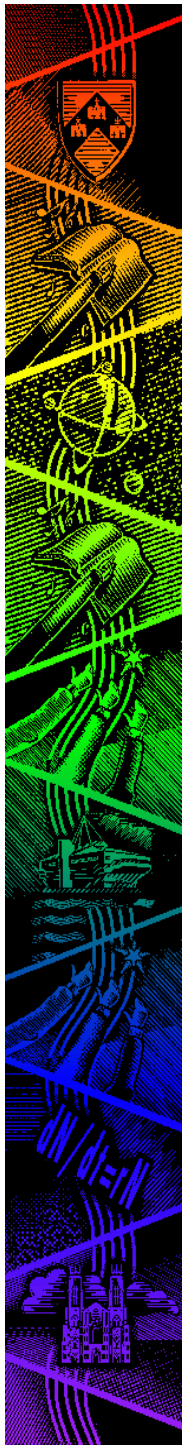
THE UNIVERSITY of York

Chris Kimble
February 2008



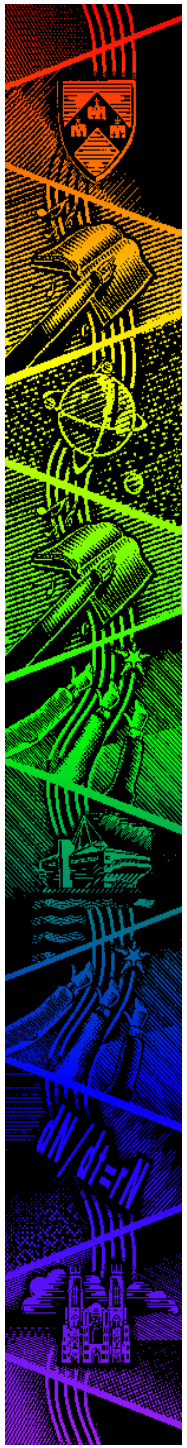
Overview

- A review of the theory
- Three types of semi-formal method
  - Structured Systems Analysis and Design Method (SSADM)
  - Structured analysis, design and implementation of information systems (STRADIS)
  - Jackson Systems Development (JSD)
- Strengths and weaknesses
- What happens in practice



A review of the theory

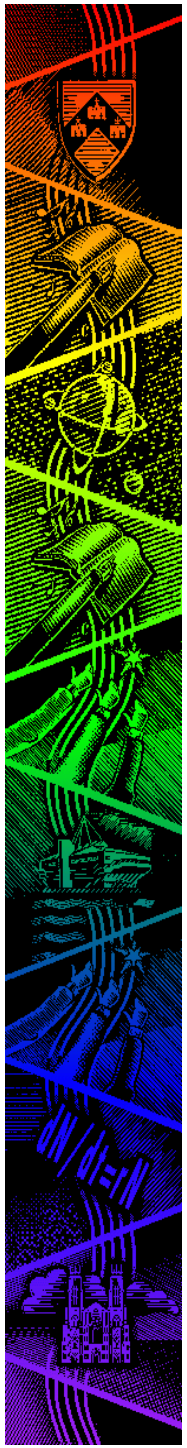
- What are the key features of semi-formal methods?
  - Have strong links to program design methods
  - Assumes the software description is complete, but not closed
  - Software designs that are logically correct do not always reflect the properties the same software design has in reality
  - The focus is on the logical flow of control in the program, which frees the programmer from having to be concerned with the physical details of the implementation
  - Allowing a split between logical and physical designs = anti-realist ontology and rationalist epistemology



Rationalism and Anti-Realism

- *Rationalist* arguments deal principally with epistemology claiming that reason is source of all knowledge and that everything that can be known, must be intelligible and rationally explicable

- *Anti-Realist* arguments deal principally with ontology claiming that the perception of reality is so bound to the mind that observes it, that it is impossible to conceive of the 'true' nature of objects

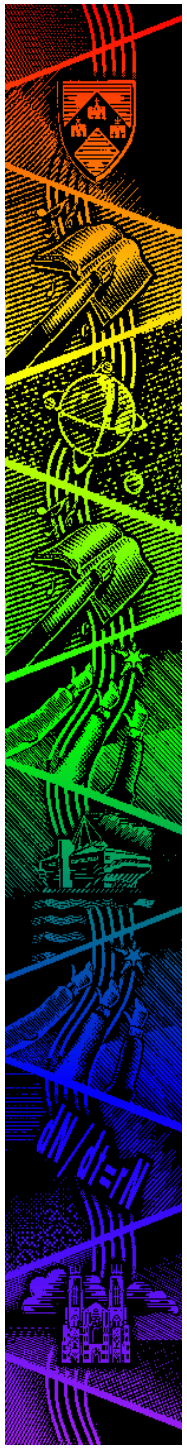


Semi-formal methods

History:

- Most semi formal methods originated in or around the 1970's

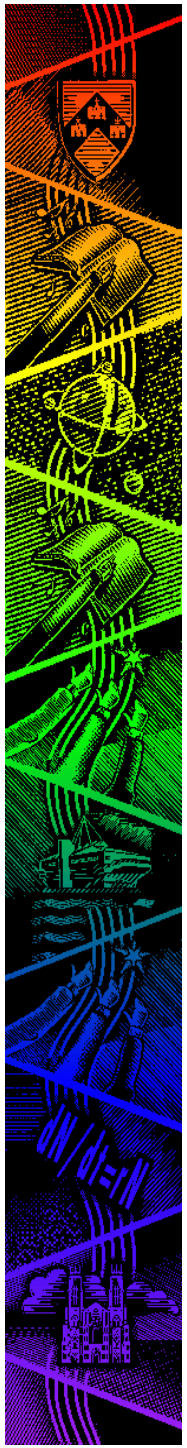
- The problem was seen as one of taking a manual system and 'computerising' it.



Types of semi-formal method

- There are many examples of such methods - most of which have been rarely if ever used.

- We will consider the three most influential:
  - Structured Systems Analysis and Design Method (SSADM)
  - Structured analysis, design and implementation of information systems (STRADIS)
  - Jackson Systems Development (JSD)

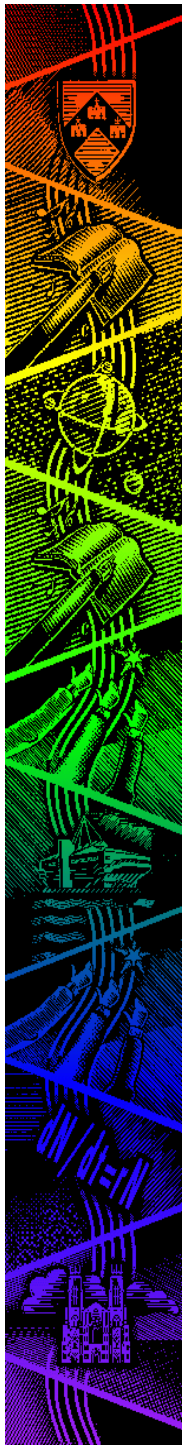


Structured Systems Analysis and Design Method (SSADM)

- SSADM is a methodology developed for the Central Computing and Telecommunications Agency (CCTA) which is responsible for computer training and procurement for the UK Civil Service

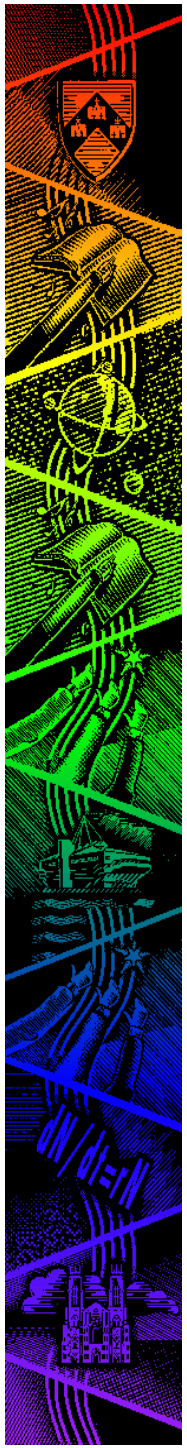
- It is sometimes described as a data-driven methodology because of its emphasis on data modelling and the construction of databases

- The current version of SSADM is version 4+



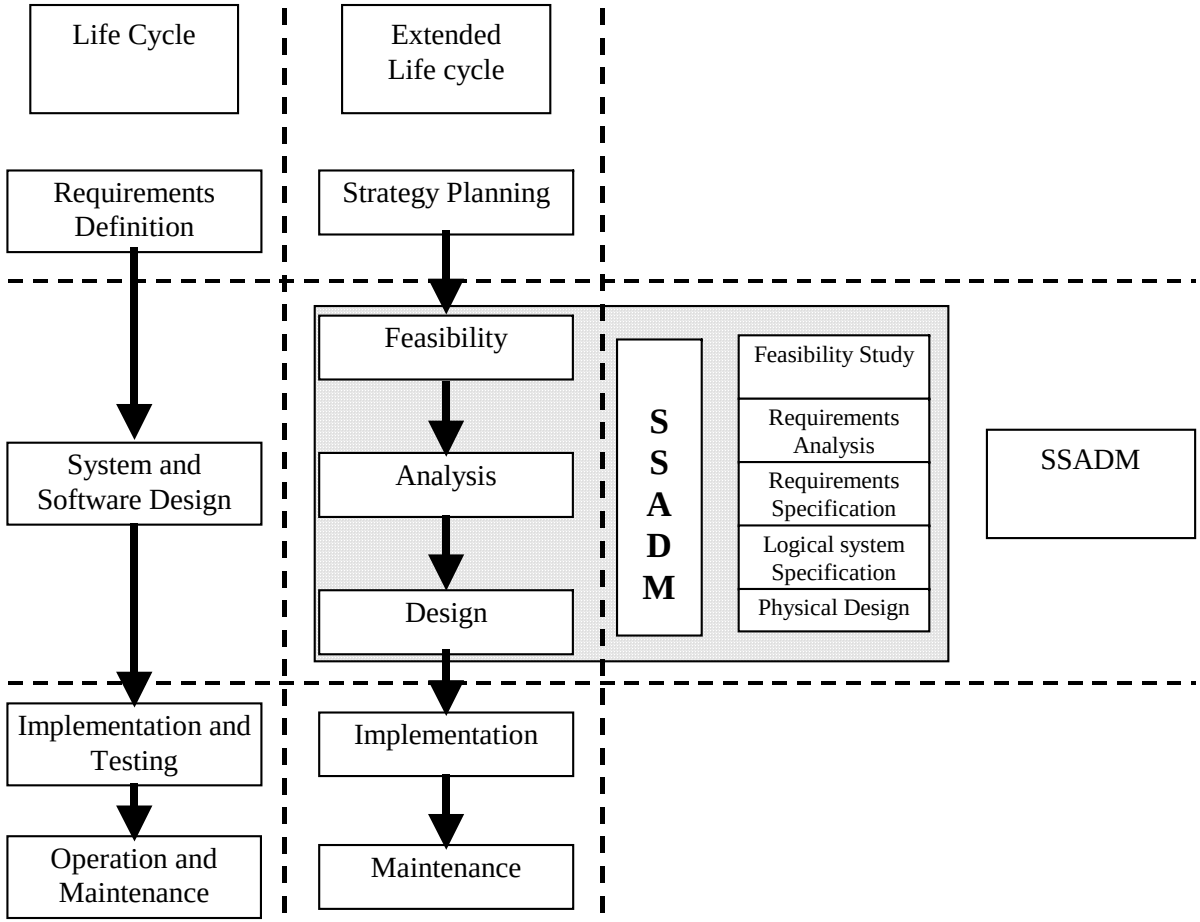
Structured Systems Analysis and Design Method (SSADM)

- SSADM is a systematic approach to the design and implementation of large-scale IS projects.
- It has seven stages within a five ‘module’ framework. Each has its own set of plans, timescales, controls and monitoring procedures and links to the next module.
- The activities of each stage are precisely defined and have associated end products (known as deliverables), which facilitate the management of such projects.

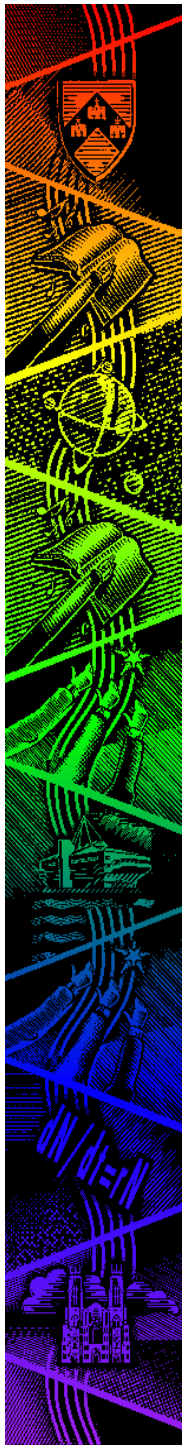


Seeing SSADM in Context

– Where does SSADM fit into the big picture?



Chris Kimble
February 2008



The Modules & Stages of SSADM

Module One = Feasibility study

- Stage 1 = Feasibility

Module Two = Requirements analysis

- Stage 2 = Investigation of current environment
- Stage 3 = Business system options

Module Three = Requirements specification

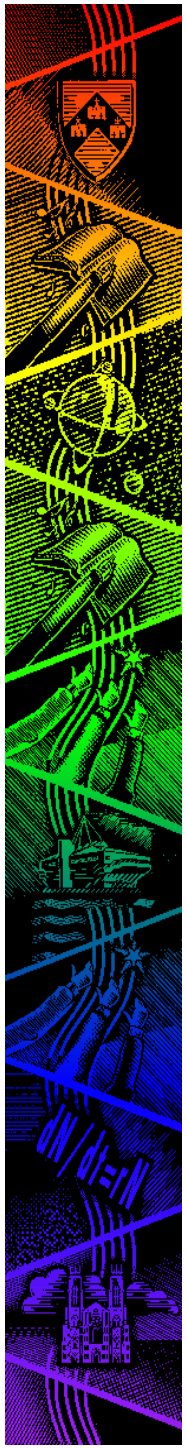
- Stage 4 = Definition of requirements

Module Four = Logical systems specification

- Stage 5 = Technical system options
- Stage 6 = Logical design

Module Five = Physical design

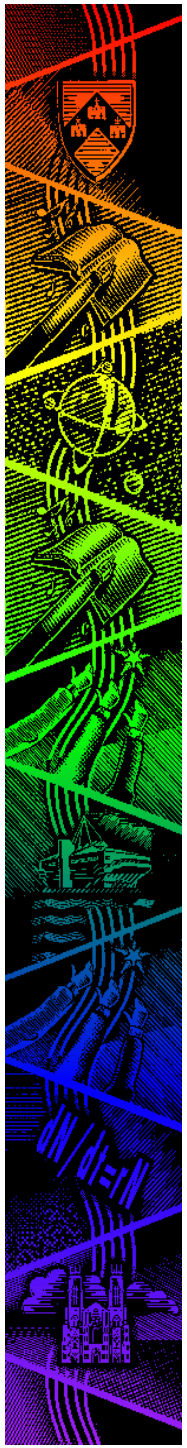
- Stage 7 = Physical design



Module One

- The first module in SSADM is a single stage feasibility study. The objective is to make sure the project is viable and to identify problems or obstacles to a successful project.

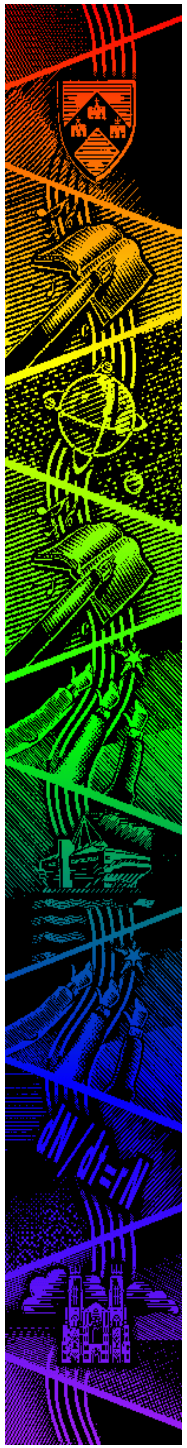
- This phase has four steps:
  - Prepare for the study, which assesses the scope of the project
  - Define the problem, which compares the requirements with the current position
  - Select feasibility option, which considers alternatives and selects one
  - Assemble feasibility report, which acts as a trigger for the next stage



Module Two

- The next module is the requirement analysis module, which consists of two stages which are carried out sequentially:
  1. investigation of current environment
  2. business systems options.

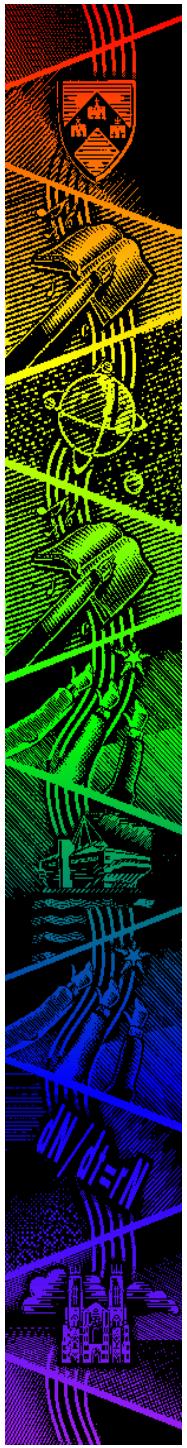
- The objective is to gain a full understanding of the requirements and establish a programme for the rest of the project.



Module Two

- Stage 1 - model current environment
  - The work in module 1 is repeated but in greater detail.
  - Data Modelling

    Data models are used to look at the data in the current system, *data-flow diagrams* are used to analyse the *physical and logical data flow* in the current system and *data dictionaries*, which represent the detailed structure of the information in the current system are produced.
  - These are carefully analysed and a requirements catalogue produced which lists of all the desirable features a new system as objectively as possible.



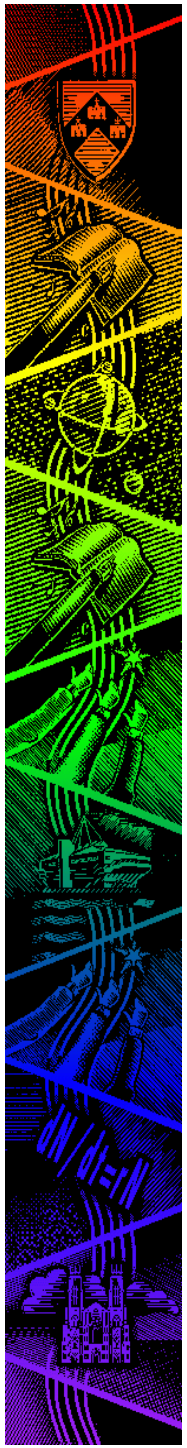
Module Two

- Stage 2 - business systems options

  The objective of this stage is to *scope* the project; it is at this stage that the *functionality* of the new system is determined and agreed.

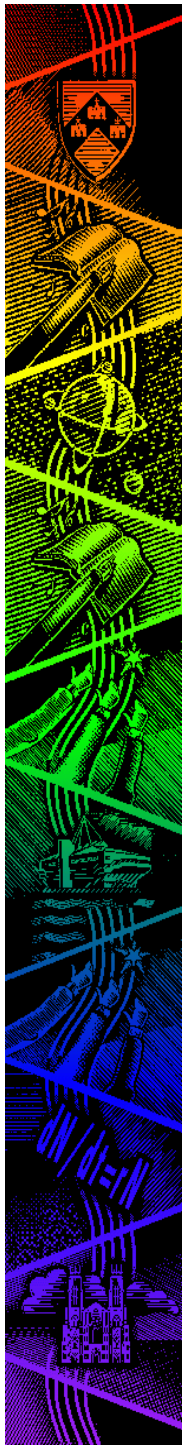
  – Different options or scenarios that meet some or all of the requirements from the previous stage are considered.

  – These options are costed (e.g. for development time, risk, etc) and reviewed to select one that best meets the project board's requirements.



Module Three

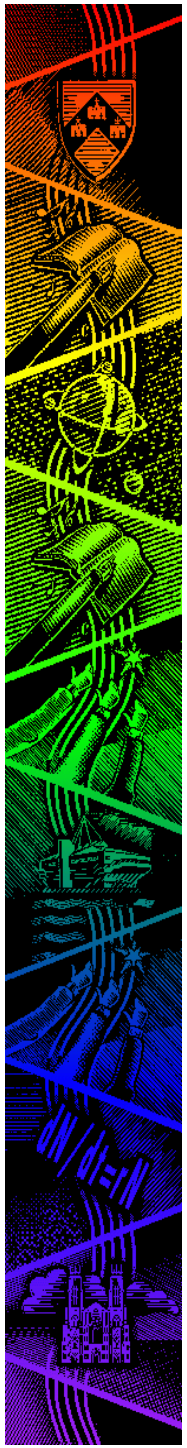
- The next module is the requirements specification module where the option from the previous stage is developed further to bring it closer to the final system.
  - *Data modelling* and *analysis* are replaced by *specification* and *logical design*.
  - The requirements catalogue and the data flow models are updated and the logical entity model extended and normalised (to third normal form). Entity life histories are also created.
  - Requirements that may have been missed in the previous module are added and requirements that are now irrelevant are removed.



Module Four

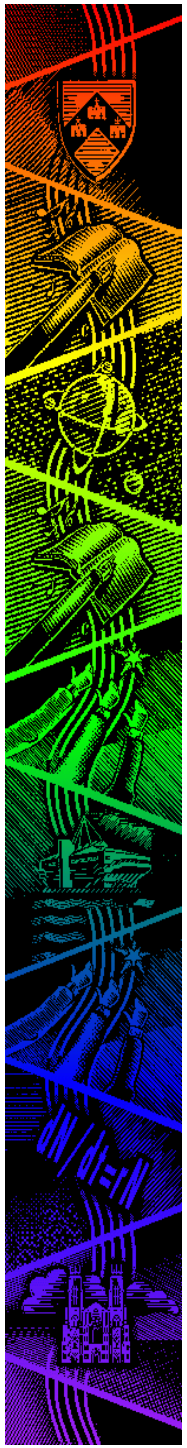
- The next module is the logical system specification. This module consists of two stages which are carried out *in parallel*:
  5. technical systems options
  6. logical design

- The objective is to ensure a smooth transition from requirements to a working system by considering at the physical and logical designs *at the same time*.



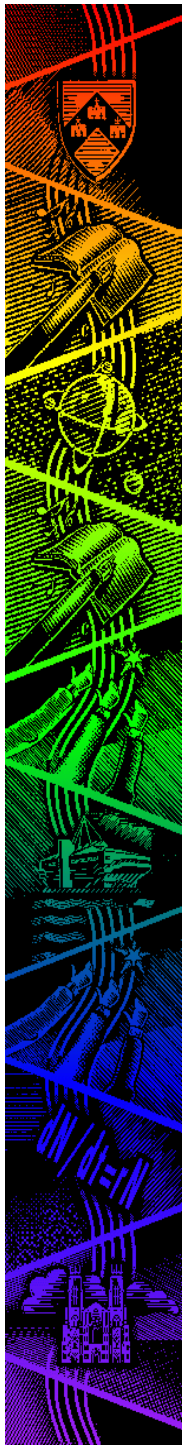
Module Four

- Stage 1 - Technical systems options
  - Technical systems options look at implementation specific details.
  - The objective is to identify *constraints*, for example, the hardware platform, time and cost, performance, security and service level requirements may be regarded as ‘given’ for any particular system.



Module Four

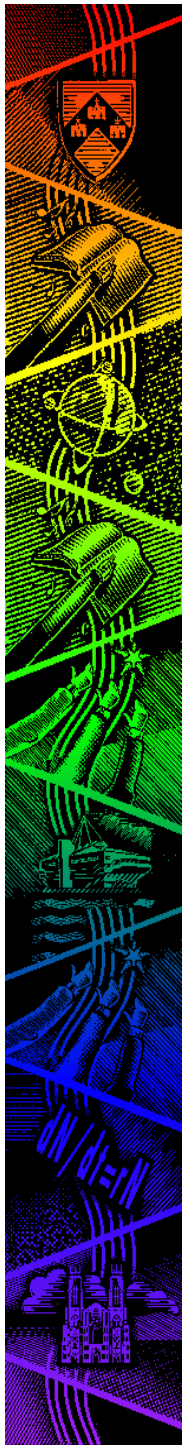
- Stage 2 - Logical design
  - This is a logical statement of what the system is required to do.
  - In this stage the detail of how the system will *control operations* will be specified together with the rules for *validating data*.
  - Dialogue and menu structures and designs for particular users or user roles are defined here and users may become directly involved with the development of prototypes.



Module Five

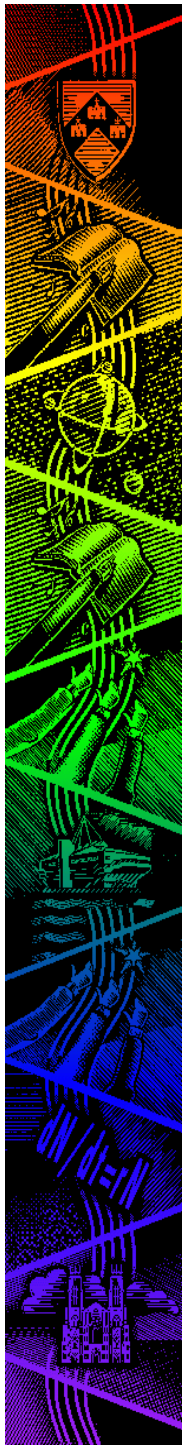
- The final stage is where the *logical data model* is *converted* into an appropriate design and *mapped* onto a *specific physical platform*; it is at this point the SSADM stops and programming and testing start.

- From here, it should be possible to develop all of the programs that are necessary to give the software its required functionality.



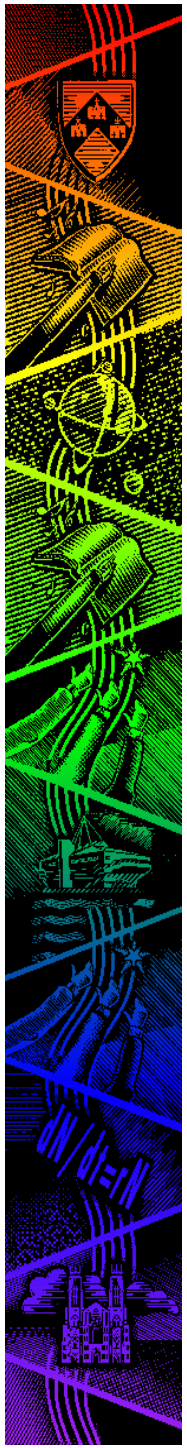
STRADIS

- STRuctured Analysis, Design and Implementation of information Systems (STRADIS)
  - Developed by Gane and Sarson in 1979 and built on the work of Stevens, Myers and Constantine (1974)
  - Based on functional decomposition and the use of the data flow diagram
  - It is concerned with the selection and organisation of program modules and interfaces.
  - Used when there is a backlog of systems waiting to be developed or insufficient resources available to develop all of the potential systems, e.g. by McDonnell Douglas Corporation.



STRADIS - Initial study

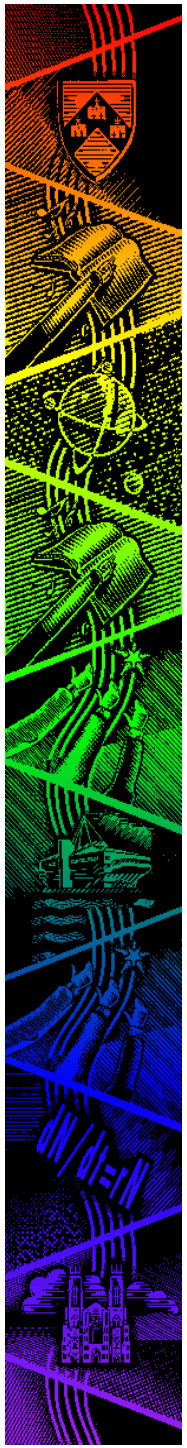
- The main objective is to ensure that the most important aspects of a system are developed. The key criteria in this process are the monetary costs and benefits of each proposal.
  - Data is collected from managers and users in the relevant areas and a high level DFD constructed together with an estimate of the times and costs of proceeding to a detailed investigation.
  - Similar to the traditional feasibility study but is not as resource intensive and does not include alternative approaches.



STRADIS - Detailed study

- If the initial study is approved, further potential users / stakeholders of the system are identified:
  1. Senior managers with profit responsibilities.
  2. Middle managers of the departments affected.
  3. End users who will work directly with the system.

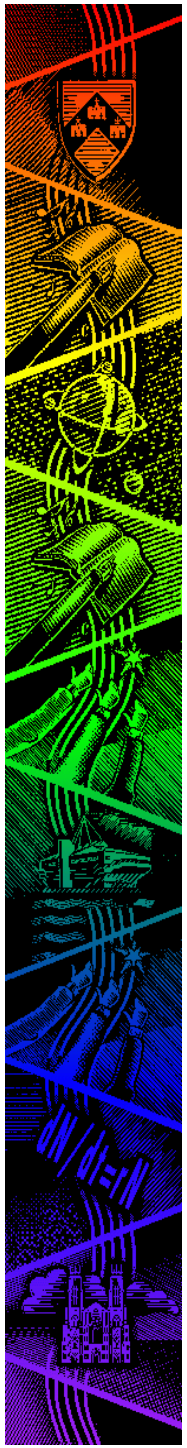
  – The analyst prepares a draft logical DFD of the new system which are ‘walked through’ with users to check their validity.
  – The detailed study is presented to management and a decision is made to either stop or proceed to the next phase.



STRADIS - Alternative solutions

- Analysts and designers now work together to produce three alternative designs together with estimates of costs, benefits, time-scales, hardware and software
  1. A low-budget, quick implementation which may not meet all the objectives
  2. A mid-budget, medium-term version, which achieves a majority of the objectives
  3. A higher budget, more ambitious, version achieving all the objectives

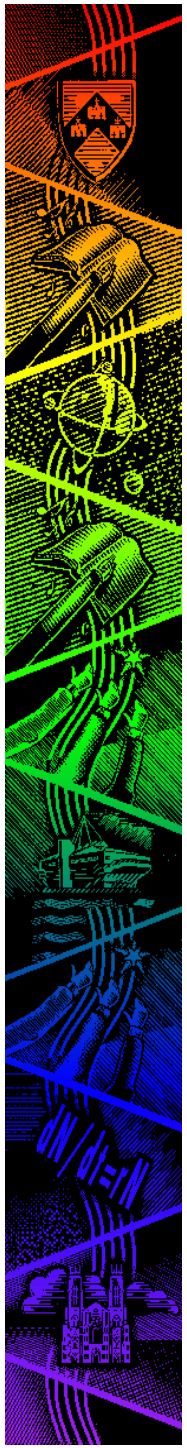
  – A logical DFD is produced that shows that all of the key system objectives are met.



STRADIS - Physical design

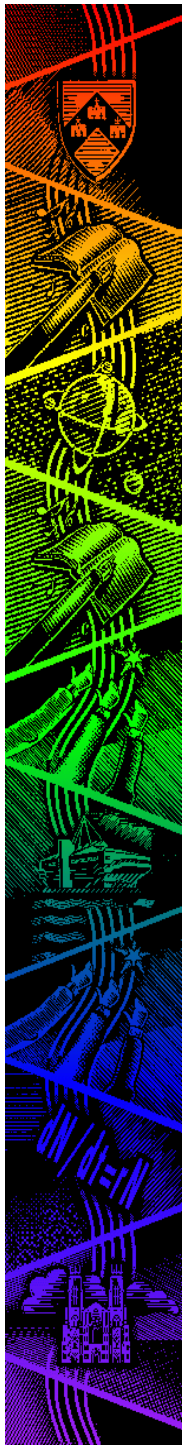
- The designer seeks to identify either of two alternative structures for the system:
  1. All transactions follow very similar processing paths - this is termed a ‘transform-centred’ system.
  2. Transactions require different processing - this is termed a ‘transaction-centred’ system.

  – DFDs are expanded and enhanced and the content of the data dictionary is completed. Low level detail is validated and agreed with the users.
  – The physical files and/or database are designed based on the data specified at the logical level and a modular hierarchy of functions is derived from the DFD.



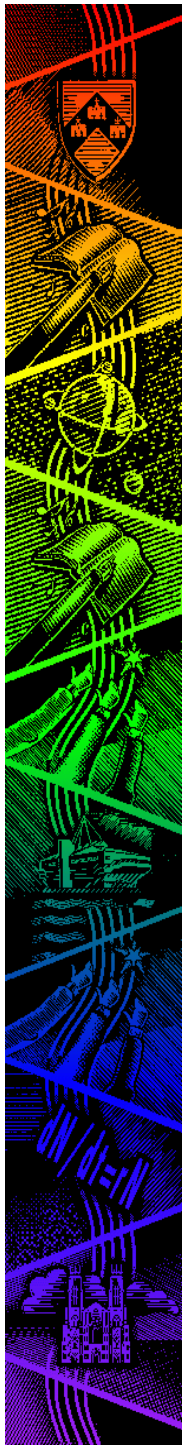
Jackson Systems Development (JSD)

- Jackson Systems Development (JSD) is an extension of program design methodology, Jackson Structured Programming (JSP).
- The main goal of JSD is to produce maintainable software, and its emphasis is on developing software.
- It does not deal with project selection, cost justification, requirements analysis, project management, user interface, procedure design or user participation.



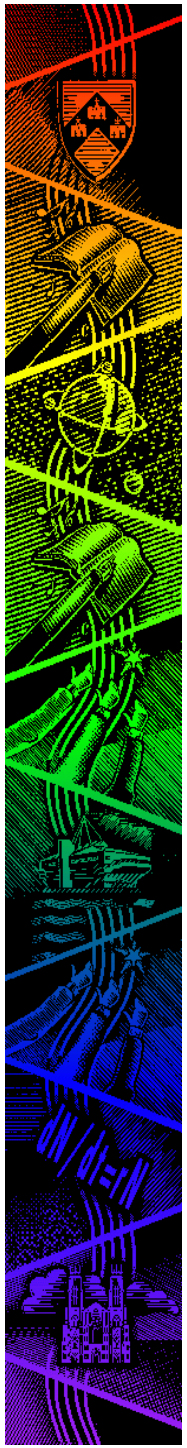
Jackson Systems Development (JSD)

- There are three major phases in JSD: the modelling phase, the network phase and the implementation phase.
  - The modelling phase is concerned with events and entities which are identified so that entity structures and entity life cycles can be formed.
  - The network phase is concerned with the inputs and outputs which are added to the model so that the input and output subsystems can be analysed.
  - The implementation phase is concerned with detailed design and coding.



Jackson Systems Development (JSD)

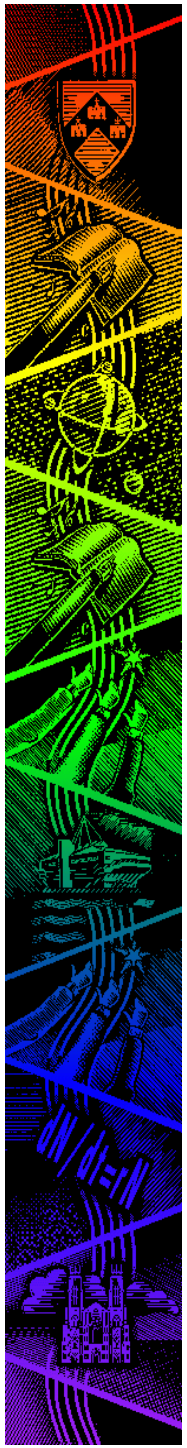
- Modelling phase
  1. Entity action step
  2. Entity structure step.
- Network phase
  3. Initial model step
  4. Function step
  5. System timing step.
- Implementation phase
  6. Physical system specification step.



Strengths of these approaches

In theory these systems:

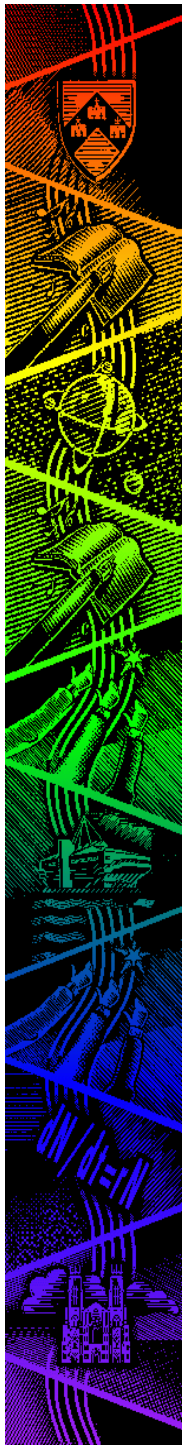
- Provide a systematic method of development so that progress can be effectively monitored
- Facilitate the quick and accurate development of code and data structures
- Produce a system which is well documented and easy to maintain
- Produce a system within an appropriate time limit and at an acceptable cost



Weaknesses of these approaches

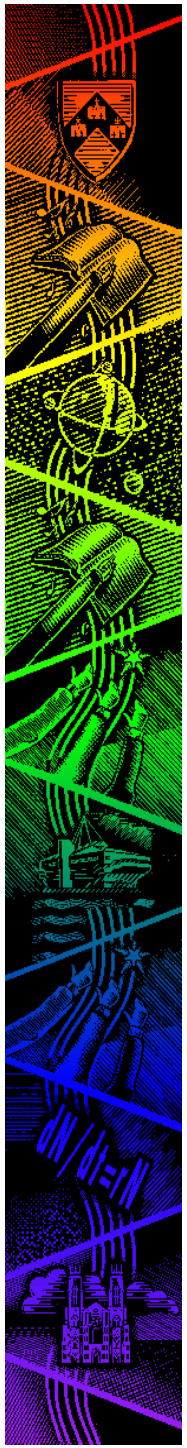
In practice they:

- They are too large and cumbersome and add to complexity rather than simplifying it.
- They are too prescriptive and become an excuse for not actually thinking about the problem.
- They assume that requirements can be found and then effectively “frozen”.
- They over-value the knowledge of the analyst and under-value the experience of the user.



Some ‘Modifications’

- The modifications to these methodologies have tended to be in the feasibility and design phases.
  - New approaches to fact finding attempt to identify problems before they become too expensive to correct.
  - Greater involvement of the user in the design to overcome some of the ‘user’ problems.



A practical example

- Middleton, P. (1999). Managing information system development in bureaucracies. Information and Software Technology, 41, 473 - 482.