Scientific Theory in Informatics

Case Study

Skeleton Report and Guidelines

1977
UNIVERSITY
OF SKÖVDE
School of Informatics

Table of Contents

Preface

This document provides some advice on how to conduct a case study for Scientific Theory in Informatics. It also provides guidelines for structure and content, including recommendations on the length of the report and titles for section headings. It provides guidance on how to focus your report to addresses the assessment criteria and demonstrate that you have achieved the required learning outcomes. Perhaps most importantly, it first explains what the goal of the case study is and how to get started working on it.

Goals of the Case Study

Let's look closely at the brief for the case study.

> "The goal of the case study is to assess how well two or more candidate modelling methodologies, core theories, or fundamental techniques might contribute to the effective and efficient delivery of an information system. Specifically, given a selected application domain, the goal is perform a critical appraisal and comparative analysis of two or more *competing models* drawn from drawn from the material covered in class and deployed in either the computational, cognitive, or socio-technical aspect of that application area."

The phrase "competing models" needs some explanation.

First, by *model* we mean a modelling methodology, a theory, or a fundamental technique, i.e. any of the topics covered in the course, or related topics belonging to one of the three perspectives on Informatics (i.e. computational, cognitive, socio-technical). Since most of the topics we covered are theories, we will refer to theories throughout the following.

Second, if two theories are *competing* it means they are focussed on achieving the same goal but with different approaches. For example, the different organizational theories grouped under Morgan's eight metaphors are alternative models that, effectively, compete with one another and all have relative strengths and weaknesses. Another example: the cognitivist approach to cognitive science competes with the enactive or embodied approaches. Again, both have a different way of modelling cognition and each is based on a different set of assumptions (or properties, to use the terminology of McClelland (2009); see Modelling Methodology lecture). Yet another example: a deterministic finite automaton and a push-down automaton are different competing but compatible models of computation and each has certain limitations on the kind of problem it can address. The case study investigates the relative strengths and weaknesses of competing approaches in the context of a specific application scenario, i.e. it conducts a grounded comparative analysis in order to identify the most appropriate theory *for that scenario*.

Conducting the Case Study

Unlike the assignment exercise, the case study does not have to address all aspects of informatics and all of the theories and techniques covered in the course. Thus, there are four choices to be made early in the case study:

1. The sub-discipline(s) of informatics to be addressed (i.e. computational, cognitive, socio-technical);
2. The candidate theories (your choice: derived from, or related to the course material);
3. The application scenario (either your own choice in connection with your own research, or one provided for the purpose by the course director, augmented as necessary to provide a good basis for candidate theories selected);
4. The phase(s) of the development lifecycle to be targeted.

Making these choices is part of the exercise. However, you can seek advice when making them. You should refer to the course director for general advice and the lecturer or lecturers who delivered the classes on your chosen theories for specialist advice. These topic experts may also suggest some supplementary reading material.

The following summarizes the main steps in the case-study process.

1. Choose the sub-discipline(s) of informatics to be studied (computational, cognitive, or socio-technical).
2. Choose the candidate theories and techniques.
3. Identify the phase or phases of the development lifecycle that these theories will inform (many theories are applicable to the problem modelling phase).
4. Expand on the application scenario and specify the information system to be delivered.
5. Summarize these decisions and seek general advice from the a relevant course teacher.
6. Revise your plans, if necessary.
7. Seek specialist advice from the topic experts (i.e. the lecturers who delivered the classes from which you selected your candidate theories); request supplementary reading.
8. Revise your plans, if necessary.
9. Carry out the case study:
    - Survey possible approaches, noting as many as possible alternatives.
    - Select two or more approaches and justify the selection.
    - Conduct a comparative analysis of strengths and weaknesses of each approach and critical appraisal from the perspective of competing theories.
    - Explain how each approach could be used; use examples drawn from your application scenario.
    - Create guidelines for the deployment of each approach, making reference to strengths and weaknesses.
    - Identify your conclusions regarding the relative merits of each approach and the benefits of using theory to guide practice.

   You should continue to consult with the topic experts as required throughout this process.

   Do not leave it to the end to begin writing your report: start writing from the outset.
10. Repeatedly read and revise your report.
11. Submit your report for assessment.

Addressing the Learning Outcomes

As stated in the course plan and course handbook, upon successfully completing the course, a student will have achieved the following learning outcomes and will be able to:

1. Explain the scientific method of problem abstraction, hypothesis formation and test, experimentation, and analysis;
2. Explain the distinction between modelling methodology and instances of specific theories and models;
3. Explain the difference between descriptive and normative models;
4. Apply a working knowledge of a representative sample of core theories and fundamental techniques in informatics, in general, and in computational, cognitive, and socio-technical systems, in particular;
5. Compare and contrast competing theories and techniques in the context of typical computer-based information systems.

Table 1 in the course handbook notes that the case study is used to assess the degree to which all five outcomes have been achieved. Therefore, it is important that your report addresses these outcomes and demonstrates that you are able to do what the outcomes say you should be able to do. In the skeleton that follows, you will find some suggestions on how to do this effectively.

Please remember that these are just suggestions. You are free to adopt or ignore them as you see fit.

The remainder of this document is presented in the form of a skeleton case study report. Don't treat is as a strict template; rather consider it as a set of helpful guidelines.

1.0 Focus of the case study

Suggested length: 1-2 pages.

1.1 Informatics sub-disciplines

The first section of the case study report should identify the sub-disciplines, or perspectives, of informatics that you will focus on. This could be the computational, the cognitive, or the socio-technical aspects of informatics, or perhaps a combination of them. Typically, if you intend to conduct a comparative analysis of competing theories, you will have to focus on just one sub-discipline since these theories will usually model one aspect of an information system.

1.2 Candidate theories and techniques

Here you need to identify the theories and techniques you intend to target in the case study. These should be competing theories (e.g. the cognitivist and the enactive approaches to modelling cognition or building intelligent systems).

1.3 Phases of the development lifecycle

A particular theory or technique normally targets one phase of the development lifecycle. Many of the topics covered in this course, e.g. probability theory, automata theory, game theory, organizational theory, provide the basis for the computational model that allows the essential problem to be solved and thus they target the problem modelling phase. Other topics, e.g. algorithmic strategies and system & software quality, target later phases in the development lifecycle, such as design, implementation, and testing. In this subsection, you should set out exactly what aspects of the development lifecycle you are focussing on and explain why the theories you have chosen are relevant.

2.0 Specification of the application scenario

Suggested length: 3-4 pages.

This is where you expand on the chosen application in order to showcase how theory can guide practice in the design and implementation of a computer-based information system. The focus in this section is both on the problem and the solution. Thus it targets the first four phases of the development lifecycle:

1. Problem identification
2. Requirements elicitation
3. Problem modelling
4. System analysis and specification

The key goal in this section is to summarize an application scenario that provides the requirements that the information system will satisfy, to identify the essential computational problem, and then to provide an outline specification of the information system that will deliver that solution.

All this allows you to then explain the relevance of the theories or techniques you have chosen to address in your case study.

If the theories and techniques you have selected target the problem modelling phase, you should explain how they facilitate a solution to that computational problem, i.e. how they can provide some part of the computational model. If they target a different phase in the development lifecycle, e.g. design, then you should explain the particular aspects of the application that will benefit from a solid theoretical foundation, such as the need for a certain type of algorithm in order to render a solution tractable.

In this section, you may take the opportunity to demonstrate that you can explain the scientific method of problem abstraction, hypothesis formation and test, experimentation, and analysis (Learning Outcome 1). One way to do this is to interpret the system development lifecycle as an example of the scientific method; you can do this by explaining which phase of the lifecycle corresponds to a stage in the scientific method. Another good way to do it is to explain the process of computational modelling.

Doing this will also allow you to address Learning Outcome 2 and explain the distinction between modelling methodology (the process by which you map from a *problem* encapsulated in the requirements to a proposed *solution*) and instances of specific theories and models (i.e. the result of the modelling process: the theory or technique you choose as the basis for your model) … think of Richard Feynman from the Modelling Methodology class and you'll get the idea.

In this section you may also explain the difference between descriptive and normative models (Learning Outcome 3). This can be done by considering the computational model that forms the basis for your information system: does it assume that the models works in a particular way (e.g. decisions are based on knowledge and rules) or does it focus on just being faithful to observed behaviour? (e.g. the recommendations an online shopping application made are consistent with the preferences of a sample group of people).

3.0 Survey and selection of theories and techniques

Suggested length: 1-2 pages.

While you have already identified the theories or techniques you intend studying, here you need to demonstrate that you are aware of the possible alternatives. Ask the advice of the topic experts if you unsure what these might be.

4.0 Comparative analysis

Suggested length: 4-6 pages.

For each theory or technique you select, you should discuss its strengths and weaknesses in the context of the chosen application scenario and the information system to be designed and implemented. Remember that you are trying either to establish which is the best (i.e. the most appropriate) theory under the circumstances, i.e. in the context of this specific application scenario.

This section directly addresses Learning Outcome 4 (the ability to apply a working knowledge of a representative sample of core theories and fundamental techniques in informatics, in general, and in computational, cognitive, and socio-technical systems, in particular) and Learning Outcome 5 (the ability to compare and contrast competing theories in the context of typical computer-based information systems). You don't have to do anything other than address these issues but it may help to keep these learning outcomes in mind as you analyse the selected theories.

5.0 Application of selected approaches

Suggested length: 1-2 pages.

The aim of this section is to convince a reader that you understand how to use the theories in the context of a real-world application. The focus here is on pragmatic issues: how would you envisage using the theory or technique in the relevant phase of the development lifecycle. For example, how would you actually use software quality techniques to improve the dependability of the information system?

Again, this section provide you with an opportunity to achieve Learning Outcome 4, i.e. the ability to apply a working knowledge of a representative sample of core theories and fundamental techniques in informatics, in general, and in computational, cognitive, and socio-technical systems, in particular;

6.0 Guidelines for deployment

Suggested length: 1-2 pages.

This section provides an opportunity for your to demonstrate your mastery of the topics, as well as your insight, by suggesting some guidelines for other people who might be interested in using this particular theory or technique.   It would be natural to base this on your analysis of the strengths and weaknesses of the approach, possibly augmented by considering the opportunities that exist for extending or improving it, as well as the risks (threats) posed by inappropriate deployment.

7.0 Conclusions

Suggested length: 1-2 pages.

Finally, this is where you identify your conclusions regarding the relative merits of each of the selected theories or techniques. You can also reflect on the advantages and disadvantages (if there are any) of using theoretical considerations to guide practice in the realization of information systems.