- *Generalisations*. Researchers generalise and specify limits on the generalisations they identify. Generalisation allows research to be interpreted and applied to a wide variety of situations. However, researchers must know the limitations of these generalisations. Generalisations stem from your own wisdom and evolve from your deductive reasoning, which leads you to develop ideas about things you have not encountered before, with certain caveats.

Failure to apply these characteristics perpetuates the status quo – everything remains unchallenged and stays the same. Without an open mind to things, without a critical eye and without an ability to generalise your understanding to different things, you will not make a contribution to knowledge. This is, after all, the main aim of your research.

1.5 What are projects?

1.5.1 *Introduction*

Although you should now understand what is meant by research and the research process, it is still necessary to identify what is meant by projects, computing projects in particular, and how research fits within this context. This section begins by discussing what is meant by projects in a more general sense first.

A project has been defined as 'something which has a beginning and an end' (Barnes 1989, cited by Turner 1993: 4). Unfortunately, this rather broad definition does not encapsulate the underlying purpose of projects, which is to bring about some form of beneficial change. This change takes you from a current existing situation to a desired situation sometime in the future. This can be represented by the *Meliorist model* shown in Figure 1.3. In this figure a project is represented by a set of actions that you perform. A project thus enables you to move from one situation to another. Your movement towards the desired situation might stem from a dissatisfaction with your current situation, a lure towards a situation which appears more satisfactory, or some combination of the two.

The desirable situation in this case represents some form of contribution to knowledge – perhaps representing the development of a new tool, technique, discovery and so on. The term 'contribution' in this context necessarily implies the uniqueness of the project and the novelty of its outcomes.

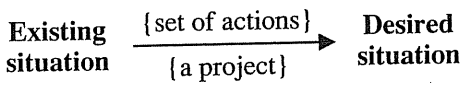

Figure 1.3 *The Meliorist model*

While project managers are concerned with other aspects of projects, such as their complexity, constraints, organisational aspects and so on, as an individual you will only be concerned with the change that your project brings about – that is, the contribution that it will make. This simplistic interpretation of projects will do for now.

So far projects have been identified as having a beginning and an end: bringing about a beneficial change by making some kind of contribution. Looking more specifically at computing projects in particular, you need to see what kind of contributions these projects can make.

Computing projects come in all different shapes and sizes as the field they are drawn from is immense. However, these days it is more widely recognised, within academic institutions, that computing projects need to do more than, for example, develop a piece of software. The project that you pursue must involve an element of research, it must justify its context, and evaluate and discuss its results. Merely developing a tool or algorithm with no evaluation or contextualisation may well be acceptable in industry, where commercial solutions are required. However, within the academic world, this is not the case and, depending on the nature of your project, it will have to contain an element of research to a greater or lesser extent.

The computing project that you embark upon gives *you* an opportunity to make your *own* contribution. There is little point in doing a project that merely regurgitates the work of others. Your own thoughts, ideas and developments *are* important, and these are the things that people reading your report are interested in. It is through your project that you will develop, not only your own skills, but also the ideas and work of others. The level of contribution made by undergraduate and postgraduate projects is looked at in more detail in Section 4.1.

The following section introduces the different kinds of project that you are likely to encounter within the field of computing. In each of these cases it has been identified how these projects make some kind of academic contribution. They do not merely follow a simplistic project process to develop a product at the end of the day.

1.5.2 Computing project types

As a guideline, projects in computing tend to fall into one of the following five categories:

- *Research-based project*: 'many good dissertations do no more than review systematically, and impose some structure on, a field of interest' (Sharp and Howard 1996: 25). A research-based project involves a thorough investigation of a particular area, improving your understanding of that area, identifying strengths and weaknesses within the field and acknowledging areas suitable for further development and

investigation. This kind of project will involve some form of literature search and review and would be suitable for undergraduate or taught Masters courses.

- *Development project*: this category includes the development not only of software and hardware systems, but also of process models, methods and algorithms. It may well require you to include evaluation, requirements documentation, designs, analyses and fully documented test results, along with user manuals or guides.

  Depending on the nature of your course the focus for a development project may vary. For example, for software engineering courses, more emphasis may be placed on the development and evaluation of a piece of software following particular process models that generate interim evaluatory documentation. Information systems courses may require you to focus more on the development of broader systems using 4GLs, CASE tools and/or database systems. In this case human–computer interaction (HCI), customer issues and requirements capture problems may be more your focus.

  Whichever kind of development project you tackle it is unlikely that the development of a product would be acceptable on its own. You would normally be expected to include a critical evaluation of the product as well as the development process used. Critical evaluation emphasises the distinction between the academic quality of your work and technical ability alone.
- *Evaluation project*: this category encompasses all projects that involve some form of evaluation as their main focus. A project of this nature might involve an evaluation of several approaches to a particular problem, an evaluation of two or more programming languages (applied in different contexts or to different problems), an evaluation of an implementation process within a particular industry, an evaluation of different user interfaces, an evaluation of a particular concept, and so on. Projects in this category may well include case studies as a vehicle for evaluating the issue under consideration.
- *Industry-based project*: this is simply an industry-based project that involves solving a problem within either an organisation or another university department. Care must be taken with these kinds of projects to ensure that they are not 'hijacked' by the sponsor. In other words, your project must not be forced in a direction that the company wishes it to go which is not necessarily suitable for your academic work or your course. You will probably find that an action research method is employed in this kind of project.
- *Problem solving*: this can involve the development of a new technique to solve a problem or might involve improving the efficiency of existing approaches. It might also involve the application of an existing problem solving technique to a new area. In these cases, some form of evaluation

would be expected; for example, did your new approach work well or did you discover reasons why it was unsuitable for problems of this nature?

These categories are not mutually exclusive and you may find that your project draws on approaches that are identified in more than one of them. In addition, the nature of your project will have an effect on the methods you will use to tackle it. The research methods that you might employ within your project were discussed earlier.

1.5.3 *Programming in computing projects*

It is not necessarily the case that, because you are on a computing course of one kind or another, you will automatically be expected to write a program. Computing is a broad field and encompasses many topics, including information systems, software engineering, knowledge engineering, HCI, data communications, networks, computer systems architecture and so on. Not all of these fields involve programming and to write a program for the sake of it is clearly ill advised.

Sometimes programming is the main emphasis of your project; for example, if you are on a software engineering course. At other times you may need to write a program as a 'vehicle' for testing and demonstrating one thing or another; for example, to test out some ideas, demonstrate a technique or algorithm, or evaluate some HCI concepts.

Whatever the case, as a computing student you will naturally be expected to produce code that is of an acceptable *quality*. Although you may not be expected to produce a fully documented piece of software with test plans, designs, evaluation and so on, any code that you do produce should be satisfactory for your aims. Your supervisor should be able to advise you on the breadth and depth of any software that you produce as part of your project, so make sure that you liaise with him or her closely.

1.6 Summary

- Research is defined as 'a considered activity which aims to make an original contribution to knowledge'.
- The research process can be either *sequential*, *generalised*, *circulatory* or *evolutionary*.
- Research can be classified according to its *field*, *approach* and *nature*. Approaches to research include *case studies*, *experiments*, *surveys* and *action research*.
- Computing projects tend to fall into one of the following five