Chapter 8

Conclusion

"Ask her to wait a moment - I am almost done." — Carl Friedrich Gauss (1777-1855), while working, when informed that his wife was dying.

This chapter briefly summarises this work in Section 8.1 and provides some ideas for future research in Section 8.2.

8.1 Summary

First, this dissertation examined the computational intelligence field, distinguishing between types of problems and the algorithms that can be used to solve them. The complexities introduced by hybrid algorithms were also explored as well as commonalities such as stopping criteria, measurements and the representation of problems.

Design patterns capture the experiential knowledge of expert designers as reusable patterns. Software based on these patterns benefits from more flexible designs that are more able to support new features, often by merely composing classes in different ways. Further, open source software was explored as a mechanism to facilitate collaboration and improved peer review.

CILib demonstrates how design patterns can be applied to provide a flexible computational intelligence framework. Scientific experimentation is facilitated by this flexibility and a simulator governed by an XML configuration file, which enables any algorithm, in any configuration, to be executed on any suitable problem while measuring any number of properties. The improved peer review of open source software and the liberal

use of unit testing should result in CILib becoming a very reliable platform. The fact that CILib is open source software also provides effective incentives for collaboration, including reputation rewards and sharing of development resources.

CiClops was introduced as a platform that primarily addresses the scalability limitations of CILib, a task greatly facilitated by the services provided by J2EE containers. Further, the benefits of a shared simulation repository and the need for statistical analysis tools that provide decision support for their proper use were discussed.

Thus, the combination of CILib and CiClops adequately addresses most of the problems set out in Section 1.2:

- **Duplication of effort:** CILib being open source means that any collaborator is made aware of what others are doing, through a common code base. Further, CiClops provides a common repository of past simulation data so that expensive simulations do no need to be executed more than once.

- **Failure to take latest developments into account:** Once again, the shared open source code base means that once a new idea is implemented, it is immediately available to everyone. That is, any specialist implementing a specific feature immediately makes the platform more general.

- **Insufficient testing on problems:** CiClops enables new experiments to be configured with ease, reducing the effort required to set up more tests. Further, past simulation data can be reused in comparisons and simulations can be executed rapidly on a parallel cluster of workstations.

- **Poor parameter choices:** Good parameter choices for algorithms can be communicated as default values in CILib. Further, CiClop's simulation repository improves awareness of better parameter choices.

- **Conflicting results:** Unit testing, a clean pattern based design and the open source peer review should all contribute to error free software.

- **Invalid statistical inference:** This issue is addressed as an item of future work in Section 8.2

Finally, a number of business models were suggested for exploiting the software for financial gain.

8.2 Future work

The following potential avenues of research have been inspired by this work:

- **The role of open source in collaborative research:** Open source clearly has benefits for collaborative software development. Its role should be studied further, to identify and quantify critical success factors when using open source as a means to facilitate collaborative research, so that these factors may be applied to other projects. If and when CILib becomes successful as a collaborative tool beyond the borders of the CIRG@UP, it can be analysed as a case study to achieve this goal.

- **PSO Taxonomy and characterisation of optimisation problems:** A solid foundation for performing empirical studies is provided by the combination of CILib and CiClops. In this light, the original goal of creating a PSO taxonomy and empirically testing PSOs should be revisited. Further, a method of characterising optimisation problems should be investigated to determine the type of problems for which a particular optimisation algorithm is best suited.

- **MathML for benchmark functions:** Benchmark functions in CILib are implemented using a separate class for each function, resulting in a very large number of classes and no way to define new functions without resorting to writing code. MathML, an XML grammar for defining mathematical expressions, [9] should be investigated as an alternative. A primary concern will be the efficiency of this approach, since benchmark functions are typically executed in tight loops. One possibility worth investigating is compiling MathML function descriptions directly into Java byte code at run time so that they become the equivalent of classes.

- **Statistical analysis tools:** Tools for hypothesis testing need to be implemented in CiClops in consultation with a domain expert on statistics. There is a fair amount of disagreement within the research community regarding the appropriateness of parametric tests when their assumptions are not satisfied [124]. The robustness of parametric tests when their assumptions are not met needs to be properly investigated. Further, alternatives such as Monte Carlo simulations, non-parametric tests, robust procedures, data transformations and re-sampling techniques should also be investigated [124].

- **Aspect Oriented Programming:** The attribute oriented functionality provided by XDoclet (refer to Section 5.4) is a subset of the broader Aspect Oriented Programming (AOP) paradigm [68, 32]. AOP groups together related pieces of code, or aspects, which can be applied across multiple classes by means of source code annotations. For example, the persistence logic provided by a J2EE container is an aspect which can be applied to entity beans by means of XDoclet tags. AOP should be investigated as a means to further improve the design of CILib and CiClops.

- **Mining simulation data:** CiClops has the potential to generate large volumes of simulation data. Data mining [119, 50] techniques should be investigated to determine trends in simulation data. In cases where the underlying data mining algorithms are based on CI techniques, as many are, an interesting question of whether CI techniques be applied recursively to make sense of CI simulation results can be answered.

- **Improved testing and development methodologies:** Unit testing and traditional development methodologies break down for experimental research code. Agile methodologies, such as extreme programming [11], should be studied as inspiration for composing new development methodologies. Further, robust testing mechanisms for stochastic processes should be investigated.