

An Empirical Modelling perspective on computing

Theory of Algorithms: Computer Science is dominated by the idea that all interaction with computers has to be conceptualised in terms of the Turing machine model. Abstract "computational thinking" and combinatorial analysis play a most significant role in studying algorithms. Other aspects of research on algorithmic theory cannot be framed purely in such abstract mathematical terms. These include: the role that data structures play in human perception of computational states and processes; the qualities of dynamic algorithms, which maintain properties of a system as its inputs are dynamically modified; and the pragmatic machine-dependent considerations relating to algorithmic space-time complexity. *Empirical Modelling puts its primary emphasis on modelling the states of live experience. It can model the interactive environments and automatic agency that support algorithmic activity, and express data structures in such terms. Algorithms themselves, and associated logical abstractions such as invariants, relate to specific patterns of interaction amongst agents within an environment. Empirical Modelling is particularly relevant to issues such as comprehension and visualisation that defy mathematical formalisation.*

Computer Programming: Much of the traditional computer science curriculum revolves around programming. Historically, much attention has been given to different programming paradigms, and to programming language concepts and constructs that can assist in making programs more intelligible and effective. *Empirical Modelling originated in research aimed at relating procedural and declarative paradigms. It has led us to conclude that an effective integration of procedural and declarative ingredients is possible, but only within the scope of a modelling activity that is broader than programming. State-transitions within an Empirical Modelling framework are conceived in a manner that is well-matched to both human and machine agency and interpretation, and also gives direct support for parallelism.*

Software Engineering: Software engineering can be seen as placing programming activity within a broader context. The subject draws on a wide range of sources, both pragmatic and formal, encompassing (e.g.) design methodologies, requirements analysis techniques such as Jackson's Problem Frames, concepts such as object-orientation and design patterns, and the representational apparatus associated with the Unified Modelling Language. *Empirical Modelling proposes a methodology for modelling with dependency that potentially offers computer support for the sense-making activities that precede software specification. It aspires to creating interactive environments that model a domain in such a way that software optimised to specific functions can be generated semi-automatically. It also offers a framework for concurrent and collaborative design activities.*

Human Computer Interaction: The study of HCI within computer science has been centred on useability issues, and on the design and impact of interfaces for visualisation and manipulation. Topical concerns in HCI are promoting ease-of-use in attaining user goals, developing appropriate affordances for interaction, and enhancing user experience. *Empirical Modelling is concerned with constructing interactive artefacts to embody patterns of observables, dependencies and agency conceived and encountered by the modeller. The primary role for such an artefact is as a "construal" that mediates the modeller's understanding. Potential 'uses' for such a construal are associated with patterns of interaction that can be ritualised and potentially automated, but it is not in general appropriate to conceive Empirical Modelling as guided by any specific functional objectives.*

Semantics: Much effort has been devoted in theoretical computer science to devising mathematical models of the behaviours of programs and of programming language constructs. As Brian Cantwell-Smith has observed, formal semantics of this nature does not address the semantic relation between an executing program and its 'real-world' interpretation, which plays a far more prominent and significant role in the application. *Empirical Modelling is driven by the latter semantic relation. This possibility is enabled because dependency serves to simplify the description of state transitions and establishes a much more intimate correspondence between agent actions in the domain and atomic state transitions in the computer model. Where formal semantics supplies complex mathematical models of computational mechanism, Empirical Modelling aspires to malleable experientially mediated models of meaning. The semantic models supported by Empirical Modelling can be of many different varieties, and encompass models of the breadth and richness associated with a "constructivist" outlook.*

Artificial Intelligence: The traditional emphasis in AI has been upon using computers to emulate human expertise. Some of the deepest and most controversial issues in AI concern: what view to take of the distinction between human and machine agency; the relationship between logicist and non-logicist perspectives and between computational and non-computational models of mind; the frame problem and the symbol grounding problem. *Empirical Modelling offers practical principles and tools for model-building in which the primary semantic relationships are experientially mediated. It provides a basis for non-logicist accounts of intelligence grounded in the perception of conjunctive relationships. The classification of experience within the Empirical Modelling framework is determined pragmatically, as are issues such as the attribution of agency and intelligence.*

Systems and Applications: A modern computer science curriculum contains many topics that broadly relate the core themes above to systems and applications. These include: computer systems and architecture; database systems; computer graphics; visualisation; agent-based systems; ubiquitous computing etc. *Empirical Modelling can be viewed as a very general form of concurrent systems modelling that can be applied both prescriptively and descriptively in developing virtual machines and in exploring external phenomena. Its fundamental concepts of observables, dependency and agency appear to have wide applicability and generality.*

Philosophy of Computing: Orthodox computer science has to a great extent been oriented towards what Winograd and Flores identify as a 'rationalistic tradition'. Modern developments in computing applications and technology make it increasingly difficult to sustain this orientation, and motivate a wider philosophical awareness than has often been observed in science and engineering. *Empirical Modelling demands an alternative foundation for which a most apt candidate appears to be William James's philosophic attitude of Radical Empiricism. Particular philosophical questions raised by considering it as an alternative framework for thinking through computing include: How much do we need to know about domains before we can compose an algorithm? What knowledge and assumptions are needed? What philosophical presuppositions concerning agency does programming require? How do representations in the mind relate to representations in the computer? What is the relationship between an abstract computational process and its physical embodiment? How can we most effectively blend human and computer activities, as is required for non-routine design, creativity and learning? How do we account for the negotiation and construction of meaning most prominently associated with computing in the arts and humanities?*