Software Development as System Development

A. Software development and system development have a different character
    software development involves interaction with text and formal languages
    systems development is concerned with physical instruments, actors and devices
Both activities present in the development of e.g. reactive systems

How do we find a unifying perspective within which to view them?

B. Tend to think of the historical progression as
    computer -> software -> system
but in fact have always had two strands of computer-related activity:
    reliable stand-alone generalised calculator non-interactive batch I-O symbolic processor
    one-off embedded reactive device interacting concurrently stimulus-response
e.g. von Neumann and Turing were concerned with both ...

C. Divergence between the cultures of software / systems:
    abstract mathematical as in theoretical computer science — closed world
    situated physically-based as in computer systems engineering — open development
arises from presumption of reliability and adoption of generic conventions for calculators
In modern trends in architecture and application presumption of reliability not enough:
    no general purpose parallel programming language
    concurrency, interrupts, communication issues

D. Motivation to regard treat software development in the system context (Winograd, Jackson)
System components characterised by stimulus-response behaviour (Deutsch)
Reliability and genericity of computing device allows s/w act as artefact representing s-r pattern
    correspondence between text of program and stimulus-response pattern of computer
Works well for classical computational problems, ineffective for general interaction
        e.g. user-computer

E. Empirical Modelling method concerned with developing computer-based techniques for constructing cognitive artefacts to represent the stimulus-response pattern of a device directly: may be non-linguistic
Conceiving models with which we can interact (e.g. state-charts) to represent metaphorically the interactions between devices that cannot be experienced (e.g. involving radio / electrical currents etc)
Generalisation of software development in so far as software is artefact (cf software as means to end)

Sources:
Empirical Modelling for Requirements, Empirical Modelling Principles for Cognitive Artefacts
Winograd, Harel, von Neumann