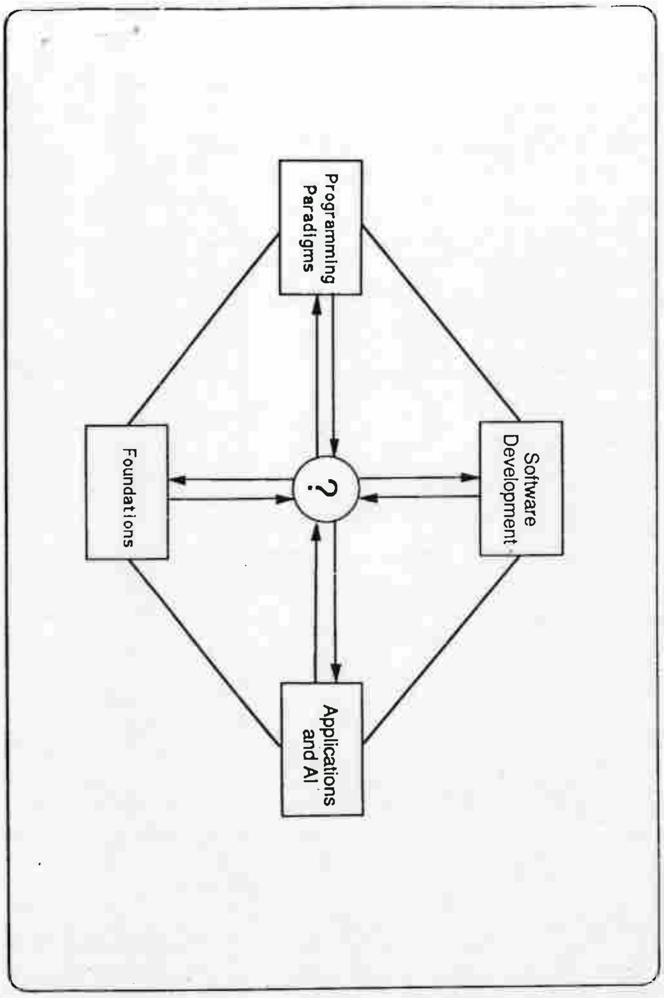
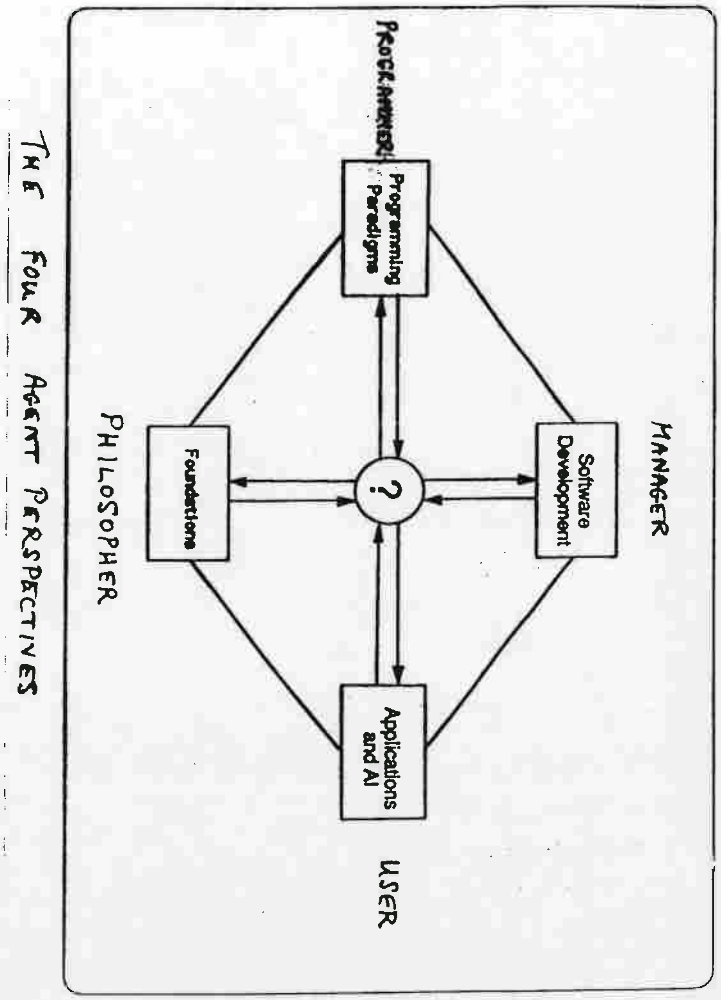




Issues for software research

Programming paradigms

What programming paradigms should we teach?
Are paradigms suited to particular applications?

Software development

How should we develop large software systems?
What specification methods should we teach?

Applications and AI

How do we relate programs to applications?
How do we interpret program behaviour?

Foundations

What is an adequate theory of computation?
What is a principled account of programming?

Sources / influences

John Backus (Turing lecture 1979):

- no satisfactory general-purpose programming paradigm: major problem in reconciling virtues of a declarative style with need for history-sensitivity in computation

William Kent (Data + Reality)

- no satisfactory data model that corresponds well to everyday concepts of identity and change & so captures the needs of data base users

Brian Cantwell-Smith (2 Lessons of Logic)

- Logicism is an inappropriate framework for AI: no satisfactory foundational theory to account for "commonsense reasoning": what can learn from logic, what necessarily have to *reject*

David Harel (Biting the Silver Bullet)

- essential role for animation of requirements in developing reactive system software: the future is visual: s/w development thru' close interaction requirements <--> testing via rapid prototyping

Preamble to "A World of Pure Experience"

It is difficult not to notice a curious unrest in the philosophic atmosphere of the time, a loosening of the old landmarks, a softening of oppositions, a mutual borrowing from one another on the part of systems anciently closed, and an interest in new suggestion, however vague, as if the one thing sure were the inadequacy of the extant school-solutions. The dissatisfaction with these seems due for the most part to a feeling that they are too abstract and academic. Life is confused and superabundant, and what the younger generation appears to crave is more of the temperament of life in its philosophy, even though it were at some cost of logical rigor and of formal purity.

James : ERE p39

Two approaches to programming

Pragmatic

Build complex systems without framework
analyse retrospectively

=> *what is the semantics of OOP?*

Formal

Prescribe prog systems via mathematical model
try to devise techniques for application

=> *what are the limitations of declarative programming?*

A1

8

Problems of representing state central to parallelism

Reasons

- if agents are to interact they need knowledge of state
  cf special case of representing user-computer interaction
- strong parallel (!) between problems of state and parallelism

Two approaches to state / parallelisation in programming

keep implicit	*circumscribe the application*
make explicit	*expose the machine*

Archetypal problem in programming

Functional programming hides state .....
..... *how to represent interaction effectively?*

Object-oriented programming opens up state ....
..... *how to keep conceptual control?*

9

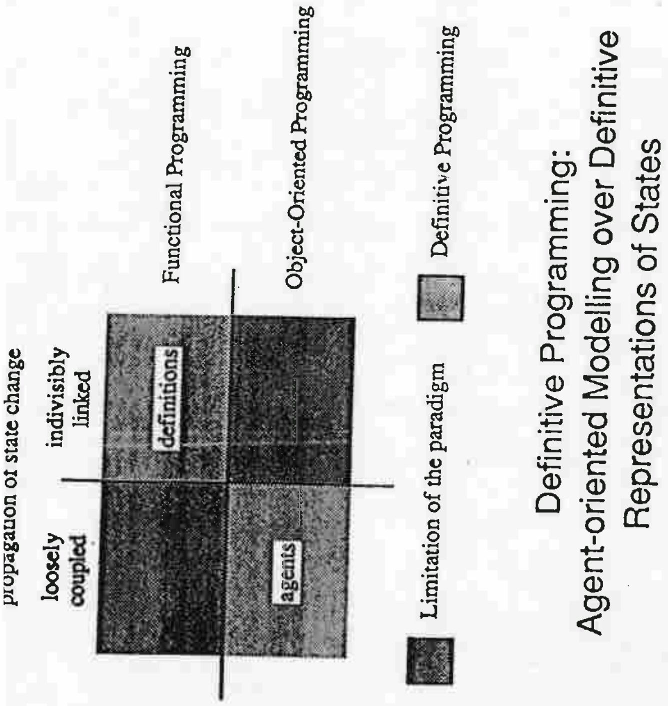
Baldwin

Why we can't program multiprocessors the way we're trying to do it now

" .... two deep flaws of existing languages:
1) reliance on side-effects,
2) use of iteration or recursion to express data parallelism ..."

Interference in parallel programming
procedural:
"has this variable currently an appropriate value?"
functional:
"is this variable currently defined?"

Ultimate goal for a parallel programming language:
support a clear statement of the data dependencies"



Implications and Future Development

.... for Practical Programming:

Agent-oriented modelling / definitive representations of state
"paradigm for exploratory programming"
Applies in context where have to 'design the machine'.

Possible directions of future development

1) marriage with OOP

Open-ended modelling --> circumscription / encapsulation
Creates *objects* in the underlying algebra
for modelling at a higher level of abstraction

2) application to modelling for target architectures

Modelling techniques so far directed at describing computational devices as they occur in e.g. reactive systems
Can use same principles to target existing architectures
e.g. to address portability of software issues
to obtain parallel implementations

3) adapt for data-intensive applications

Much development of spreadsheet principles already concerned with data-base and interface issues (RL/1 from CWI). Need to develop relationship to such existing practical / theoretical research (e.g. functional dependencies in DBs).

01

PROGRAMMING

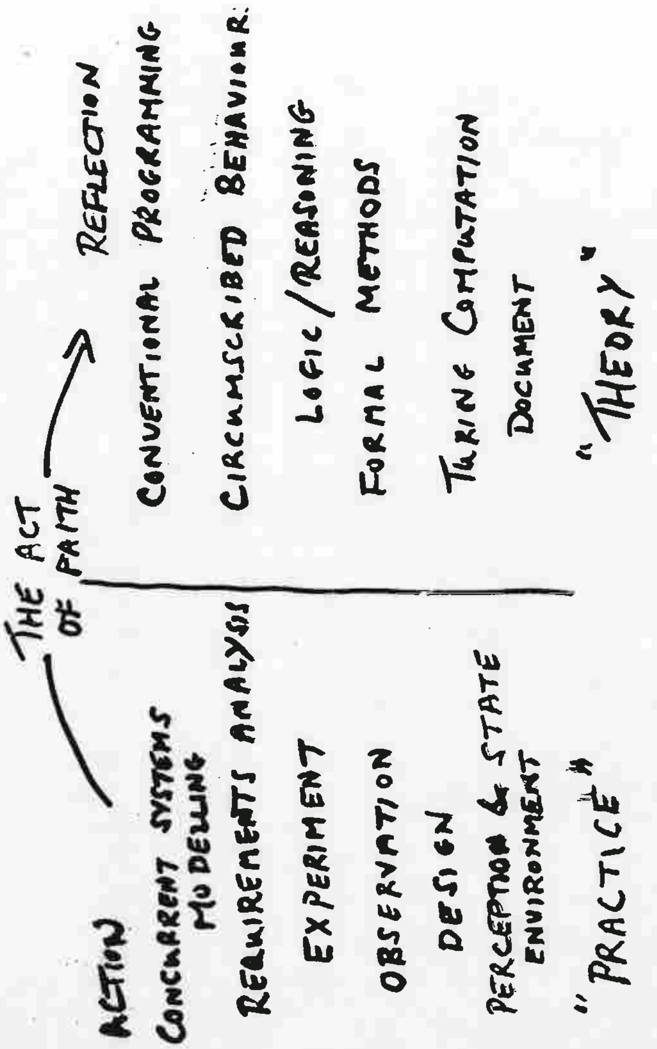
1. TRANSLATION TECHNIQUES
   TRIDENT ETC.
2. DEPENDENCY AT M/C LEVEL
   SCICS HEAPSORT
3. PORTABILITY ISSUES
   MODELLING ARCHITECTURES
   E.G. SAND.

SOFTWARE DEVELOPMENT

?

SYSTEMS ?

- HAREL STATECHARTS
  SIT : BUILD + LINK
  STATECHARTS ⊂ LSD
- EMPIRICAL APPROACH
- CONCURRENT ENGINEERING
- COMPOSITIONAL ASPECTS.



6

Programming

Writing a program for a computational device =>

- representing its repertoire of <u>indivisible</u> actions (machine code)
- representing other state changing activities in terms of these (as in compiling)

Indivisibility may be matter of protocols, as in e.g. "waiting for spreadsheet to update before interpreting"

Interpretation of the program is with reference to an independent state-transition model:

Correspondence:

state changes in the computation device <-> state changes in reference model

Correspondence based upon

protocols for **observation**

conventions for **interpretation**

subject to an appropriate experimental framework.

7

Thesis:

Sequential / batch programming / IO abstractions mean

can ***ignore*** *the protocols for observation and conventions for interpretation-* ***abstract*** *away through implicit conventions*

BUT must restore them if to make sense of computation in general e.g. reactive systems.

Current mathematical models for computation are only suited to semantics of computational devices whose properties are presumed **static**.

Key problem in reactive systems / AI: can't **preconceive** what can be used for computation. E.g. don't know in advance that a biro is a pea-shooter.

APPLICATIONS 4

Motivation for linking 1st & 2nd factors (FORM / CONTENT)

Need to know how to:

- write programs that are easy to interpret*
- write interactive programs to adapt to user
- integrate requirements analysis and spec
- model CAD, where user introduces knowledge incrementally
- program a robot to make correspondence: between **internal model** & **sensory input**

*AND DOES THIS <u>MEAN</u> ANYTHING ?

AI / APPLICATIONS

- "COMMONSENSE CONCURRENCY"
- PRAGMATIC STANCE
- IDENTIFY SIMULTANEOUSLY
  DOMAIN OF VALIDITY
  +
  WHAT IS OBSERVED / BELIEVED
- SOCIETY OF MIND
  BEETLES

FORM CONTENT

Conventions to link 1st and 2nd factor aspects:

descriptive identifiers
lazy evaluation
data structures to reflect the **application** objects
etc

BUT
This is inadequate ... need new **principles** to deal with 1st and 2nd factor **interaction**

3

3 **tenets** of classical logic to be **reconstructed**

CONTEXT DEPENDENCE
use can be ignored. A sentence must represent its whole content explicitly.

INTERACTION OF FIRST/SECOND FACTORS
locally first & second factors treated independently, ultimately globally related.

CF Defn of **formal**
"From step to step, in a formal proof, the
first-factor inference procedure
can not depend on or affect
second-factor semantic interpretation"

MORE DISCRIMINATE MODELLING
language and modelling are treated as
distinct types of representation:
linguistic reference relation non-transitive,
but
modelling is transitive and "free":
can use a model of X in place of X.

"promiscuous modelling"

Agent-oriented analysis

> ... the healthy thing for philosophy is to leave off grubbing underground for what effects effectuation, or what makes actions act, and to try to solve the concrete questions of
> ***where effectuation in this world is located*** [identification of agency],
> ***of which things are the true causal agents there*** [attribution to agents],
> and
> ***of what the more remote effects consist*** [interpretation in state-based terms].
>
> If we could know what causation really and transcendentally is in itself, the only *use* of the knowledge would be to help is to recognize an actual cause when we had one, adn so to track the future course of operations more intelligently out. The mere abstract inquiry into causation's hidden nature is not more sublime than any other inquiry equally abstract.

James: ERE p185-6.

cf. I am perfectly willing to admit any number of noumenal beings or events into philosophy if only their pragmatic value can be shown. (James ERE p242)

DB

Activity from Agency

The *real* facts of activity: three principal types

- vehicle of real activity is a consciousness of wider time-span than ours
- "ideas" struggling with one another are the agents, and prevalence of one set of them is the action [cf. Society of Mind]
- nerve-cells are the agents, & the resultant motor discharges are the acts achieved

How to arbitrate between alternative interpretations?

What is the real activity?
= What will be the actual results?

humanly and dramatically, we like to believe that activities of both wider and of narrower span are at work ...

no philosophic knowledge of the general nature and constitution of tendencies, or of the relation of larger to smaller ones, can help us to predict which of all the various competing tendencies that interest us in this universe are likeliest to prevail

James ERE: P175, 178-80

Bootstrapping techniques in empirical modelling

Building upon experiments

Hooke's Law experiment relies on arithmetic must be able to calculate in order to model the experimental measurements

underlying algebra is to
the experimenter as interpreter
as

measuring instrument is to
the experimenter as observer

Binding form to content

how is the association established and elaborated?

1. privately: by ostensive means

identify through action / demonstration
this thing / that thing

2. identification by utterance or icon

association via repeated use in context

3. publically: by common consensus

as gauged by intelligent response and action

Bootstrapping techniques in empirical modelling

Extending the semantic range of expt'l methods

experiment has three aspects [Gooding]

acquire conviction
that something specific reliably happens

acquire conviction
that something happens in far greater generality than can be experimentally tested

acquire conviction
that experimental content can be communicated and is shared

Conclusion

"formal": can be defined
wrt observation of physical system
(cf Smith: formality
... reduces to physical realisability)

Expect formal foundations in future to be based on multi-agent models of physical systems

0-agent systems = trad formal math models

0-agent = "all behaviour preconceived"

current theorists indulge in post-conceiving
anyone can pre-conceive after the fact

function abstraction in observation is historically primary linked to variables in the pre-19th century sense of fluents

key to multi-agent modelling:
principled use of function abstraction
function for IO relation if not **indivisible** abuse

Graham Bird, Philosophy, Manchester (1986)

* We are left ... with a puzzle about the role or sense of 'pure experience'. It is evidently of great importance in James's account, and yet also totally inarticulate. ... [cf.] Wittgenstein 'a nothing would do as well as something about which nothing can be said'. For James's pure experience has to be such that nothing can be said about it, if it is to fulfil the role for which it is cast. ... James refers to the 'speechlessness' of sensations and criticises Green for not simply accepting this. ....

.... Without some ability to characterise the experiences we have no means of determining their identity, and even no clear means of assessing James's central claim that we are presented with conjunctive relations in experience as well as atomic sensations. Despite these problems James does offer characterisations of such conjunctive relations and the pure experiences which involves them. But in order to characterise them in ordinary terms James plainly needs some device like the 'bracketing' of phenomenology, or a non-committal set of phenomenological descriptions, which could give sense to the idea of a neutral terminology.

James' Perspective on Radical Empiricism

These are the main features of a philosophy of pure experience. It has innumerable other aspects and arouses innumerable questions, but the points I have touched on seem enough to make an entering wedge. In my own mind such a philosophy harmonizes best with [many other philosophical outlooks*]. I can not, however, be sure that all these doctrines are its necessary and indispensable allies. It presents so many points of difference, both from the common sense and from the idealism that have made our philosophic language, that it is almost as difficult to state as it is to think it out clearly, and if it is ever to grow into a respectable system, it will have to be built up by the contributions of many co-operating minds.

James : ERE p90-91

* Other philosophical doctrines listed are
- radical pluralism
- novelty and indeterminism
- moralism and theism
- humanism

mu

DS

Potential MSc projects involving parallelism

1. Railway Signalling (MSJ)
2. Extending the ADM
3. Parallel implementation of EDEN
4. Parallel implementation of ADM
5. Comparative study of concurrent systems modelling techniques e.g. SDL vs Agent-oriented modelling, Statecharts vs LSD
6. Extension of LSD design: parametrisation, roles and subagents, agent references
7. Code generation for parallel m/c from definitive specification
8. Definitive implementation of EDEN
9. From requirements to formal specification of reactive systems
10. Formal association of requirements and testing
11. Scientific visualisation with parallelism (cf GRN)
12. A definitive notation to specify synchronisation of events
13. Communications network simulation (BT)
14. Development of environment for modelling reactive systems
15. Comprehensive formal analysis of a small LSD specification
16. Elaboration of LSD concepts in respect of intention, obligation of agents agents, roles, objects, processes etc
17. A comprehensive cricket simulation

Proposals for joint projects considered, subject to GRM approval