Some comparative studies

Logo as a naïve way to specify geometry procedurally
cf. Donald as naïve definitive approach to geometry

Two agendas:
- paradigms for programming [procedural/declarative]
- educational technology and computer programming [Papert and constructionism]

… can contrast initially using logoparserRoe2002.

Procedural programming style …

procedural actions having side-effects
"recipes to create state"
In logoparserRoe2002, have a script to record state

current state as an accumulation of side-effects
- issues with undo
- can change present, but harder to fix the past

educational value: "teaching computational thinking"

Declarative programming style …

express the current state of affairs
aspire to frame all state-change statically
can cope with rationalised change

tension between declarative and procedural
- easy to specify where things currently are
- not so easy to specify how to navigate to them

computational thinking: how to make recipe for state

Character of the states described …

Procedural programming style …

- state as a trace, but no significant observables
  "where we turned the corner in the room"
- no latent behaviours / dependencies to distinguish one feature (point/line) from another

Declarative programming style …

- referential transparency
  precludes change that has not been rationalised
  no counterpart of a genuine observable

More sophisticated forms of programming

… introduce declarative elements to procedural languages through object-like abstractions

… introduce procedural elements to declarative languages through non-logical techniques (such as lazy evaluation, cuts etc)

… both place constraints upon the meaning of change
cf. post-theory rather than pre-theory

Definitive programming
vs. programming with dependency

Consider 'adding dependency' to a sophisticated programming environment …

… case study
due to Chris Roe
based on Imagine Logo

Imagine Logo (Kalas et al, PL for educational use)
logo extension / object-orientation / event-based

Matters arising …

? Implications of 'adding dependency' to a sophisticated programming environment …

more powerful programming environment	✓
conceptual coherence	?

? … relation to:
generalised spreadsheet environments
event-driven programming techniques