Top-down vs. bottom-up approaches

Generally by the mid-1980s the top-down paradigm of symbolic AI was being questioned while distributed and bottom-up models of mind were gaining popularity. In computation two major fields developed, connectionism and evolutionary computing. Other bottom-up trends in AI have been, situated cognition (with varied threads including anthropology and robotics) and distributed AI. Shades of the rationalist-empiricist debate are seen here.

For robotics see Alternative Essences of Intelligence, The Cog Team, MIT, 1998. http://www.ai.mit.edu/people/brooks/papers/group-AAAI-98.pdf

The advantages and disadvantages of the t-d and b-u approaches in AI are complementary.

- Top-down (aka symbolic) approach
    - Hierarchically organised (top-down) architecture
    - All the necessary knowledge is pre-programmed, i.e. already present - in the knowledge base.
    - Analysis/ computation involves creating, manipulating and linking symbols (hence propositional and predicate- calculus approach).
    - ``Serial executive'' might be seen as the conscious rule-interpreter which acts on the parallel-processing unconscious intuitive processor.
    - Thus the program performs better at relatively high-level tasks such as language processing aka NLP - it is consistent with currently accepted theories of language aquisition which assume some high-level modularity. But how well are subtleties of language handled?
- Bottom-up approach - eg. neural networks
    - Models are built from simple components connected in a network.
    - Relatively simple abstract program consisting of learning cycles.
    - Program builds its own (distributed) ``knowledge base'' and ``commonsense assertions''.
    - Normally done with parallel processing, or more commonly with data structures simulating parallel processing, such as neural networks.

"... intelligence emerges from the interactions of large numbers of simple processing units" (Rumelhart et al., 1986 - PDP, vol. 1, p. ix)

- Closer relation to the known microstructure and functioning of the brain, thus might allow for more explicit modelling. See brain analogy later.
- Built-in learning mechanism, thus adaptivity and flexibility. See cognition analogy later.
- Better able to model lower-level human functions, such as image recognition and motor control - hence robotics, computer vision and speech recognition.

Each method fails where the other excels. In NLP the b-u approach would take too long to build up the rich knowledge base required for even simple language behaviour. In robotics the knowledge base is too dependent on the external environment for explicit pre-programing to be feasible - adaptivity of b-u is useful here.

Object-orientated approach might provide a compromise. Most neural networks today are programmed using object oriented software.

(see example in Ralph Morelli et. al. (Eds.) **Minds, Brains and Computers** (Chapter 1), Ablex, Norwood, NJ, 1992)
