5. Special Purpose Languages

- The languages we have looked at so far, as well as the paradigms on which they are based, are intended as general purpose languages, i.e. we expect that every programming task can be done using the languages.

- There are languages that were developed with a special purpose in mind (also called domain specific languages).

- Some of these languages later were extended to become general purpose languages and usually this was done using one of the paradigms we looked at.

Special Purpose Languages (cont.)

- Usually, special purpose languages require more intelligence in compiler/interpreter and run-time system and one reason why some of these languages never were successful among the target audience were bad compilers/interpreters and run-time systems that essentially threw away any advantages of the languages.

- In this sense, Haskell or PROLOG come near to special purpose languages, although they were Turing-complete from the beginning.

- But special purposes definitely will also in future drive development of programming languages.
Examples

- Text processing/display:
  - TeX, LaTeX
  - HTML
- Data base creation/interaction
  - SQL
- Education/teaching
  - Pascal, Modula
- Symbolic Math
  - Mathematica
  - Matlab

Examples (cont.)

- Statistics
  - R
  - SPSS
- Hardware description languages
  - Verilog
  - VHDL
- Computer games
  - Maya
  - Unreal Engine
- Spreadsheets
  - Excel

Examples (cont.)

- Languages for multi-agent systems
  - Jade
  - netlogo
- Languages for embedded systems
  - Usually company-specific (and kept as private as possible)
- Systems for programming by demonstration/examples
  see the following
Special purpose language vs library

- Library “extends” already existing language and therefore requires being able to program in that language.
- Special purpose language can be easier to program in.
- Special purpose language might be too specific for a particular user application.
- Library is easier changed than language.
- Special purpose language only defines what “intelligence” need to be realized and different implementations are possible.
- Library is its own implementation.

Security: special purpose language run-time system code can be hidden. Library usually has to be open.