

KITE

by Garth Wolfendale
Enterprise Architect
MCS, PacWest, USA

RESOLUTION OF PROGRAMMING AND DATA DISTINCTION

We currently separate programmatic/configuration/application data from coding. For example, XML is the de facto standard for defining and implementing data and we choose our coding languages as C, C++, C#, Java etc. Prolog, a declarative, clausal language can do both. In fact There is a simple 1-1 conversion from XML to Prolog. e.g.

```
<x>
  <x1> .....</x1>
  <x2>
    <x3> ..... </x3>
  </x2>
</x>
```

Readily translates to :-

```
x:-
  x1,
  x2.
x1:-
  .....

x2:-
  x3.

X3:-
  .....
```

However the latter is active and not only defines the data but how it can be processed .. also providing for backtracking and multiple entry points or 'goals'. Also there are many efficient implementations of Prolog based on interpreters and compilers. In this way there is no real distinction between data and programs .. they are one entity .. The data not only defines itself, but also provides the various ways of processing it in the same 'package'. This model is compatible with any application

platform (Java, .NET etc.) and Prolog itself is compatible with traditional Functional, OO and/or SOA models implementing its predicates.

Just a Thought.