

David Dodds

david_dodds_2001@yahoo.com

Open-Meta Computing Inc



David Dodds

Second Order Metaprogramming



David Dodds

TOPICS:

Second Order Metaprogramming,
Ontologies

First Order Metaprogramming

- Scheduler programs , dispatchers, compilers
 - Planning, SHRDLU (Winograd), blocks world (problem), Microplanner, Strips, Abstrips
 - Learning, Genetic Algorithms, Artificial Neural Networks
 - Aspect Oriented Programming (AOP)
 - reflection [Java Reflection, Smalltalk, etc]
 - Situation Awareness (military, business)
-
-

Second Order Metaprogramming

“Essentially, metaprogramming is an operation in which a central control system controls hundreds of thousands of programs operating in parallel simultaneously.”

METAPROGRAMMING = making Models,
using Symbols, Analogizing, making
Metaphors, inventing and using
Mathematics, Lilly.

Second Order Metaprogramming 2

- Society of Mind, Marvin Minsky
- Biology of Cognition / Autopoietic Theory
Maturana and Varela
- Learning, "Thinking", planning de Bono
- Second Order Cybernetics Bateson



Second Order Metaprogramming 3

- I have an integrating memory of things as I live my life. holistic, ganzfeld
 - Reminding, Roger Schank (Scripts, Frames, Plans)
 - On Intelligence, memory-prediction model, Jeff Hawkins
 - Meme, Schemata , Sowa / Peirce
Conceptual Structures. Firstness,
Secondness, Thirdness.
-
-

Second Order Metaprogramming Tools

- Maturana Varela Autopoietic Theory
Implementation in machine
 - Analogy Metaphor (event horizon, gravity well, black hole) [Bateson cybernetics, Lakoff, Gentner, Hofstadter], game physics
 - Blackboard Technology
 - Posted Agenda
-
-

Second Order Metaprogramming Tools 2

- Agent-based Aspect Oriented Programming
 - Agent-based reflection [Java Reflection, Smalltalk, etc]
 - Context, Situation Awareness, situated processing.
 - OPRS, reactive, deliberative
 - Agent-based OntoLT-like ontology generating system (from reflection results)
-
-

Second Order Metaprogramming questions

- In the past more often than not programs that modified other programs produced an oops.
 - In biological systems a genetic change, more often than not, reduces adaptability or survival of the organism.
 - Entropy always takes its toll. The only counter-action to loss is negative entropy, negentropy.
 - It is believed that intelligence, learning, thinking are, or at least can be, negentropy.
-
-

Second Order

Metaprogramming questions 2

- A successful system probably doesn't have to be conscious-ants, bees, roaches.
 - Adaptability is probably the longevity factor not intelligence per se.
 - Getting away from the programming 'man at the center' godhead fixation.
 - How does the mp system know what to do? When to do it?
-
-

Second Order

Metaprogramming questions 3

- How do you know what to do and when to do it?
 - Survival, Persistence, Adaptation, Growth, Discovery, Efficiency
 - Context, Situation Awareness, situated processing.
 - What is needed is at least the discovery of the mental equivalent of the DNA "magical" helix.
-
-

OpenCYC and SUMO

- DAML is used in OpenCYC & SUMO to express taxonomic interrelationships, amongst the general physical, cultural and social knowledge coded there. Terms like `#$PurposefulAction` and `#$performedBy` are related to other CYC concepts represented, in such a way that a reasoner can perceive “connections” not directly stated in input.
-
-

Lesley Evans slide 1

Requirements for 2nd Order Metaprogramming Language

- What are metaprogramming languages?
 - How do these differ from meta-modeling and ontologies?
 - What components are necessary to support 2nd order metaprogramming language?
-
-

Lesley Evans slide 2

Requirements for 2nd order metaprogramming language

- Broker exchanges or results of evaluation between ontologies
->
 - Defined syntax ->
 - Domain-specific ->
 - The language must have its own meta-model to which it can map disparate ontologies
 - Should be higher-level and 'human' readable and writable
 - To avoid needing to be all things to all data
-
-

Lesley Evans slide 3

Requirements for 2nd order metaprogramming language

- Require sizable memory cache->
 - CPU cycles ->
 - Distribution ->
 - Assumptions
 - Memory-intensive processes required to overlap ontologies
 - Require cycles sufficient for numerous iterations of meta-programs
 - Current technology allows for distributed processing to handle the need for memory, CPU
-
-