

Domain Analysis: From Tarpit Extraction to Object Mania?

It's Not My Fault

James M. Neighbors
Bayfront Technologies, Inc.
neighbors@netcom.com



Things I Didn't Do

- 5,000lb DA methodology
"I'm from the government/big business and I'm here to help government/big business."
It doesn't have to be that complex.
- OO mania
"I'm from a publisher/OO CASE tool maker and I'm here to seminar/consult."
OOP is a good encapsulation technique that *might* reach into architectural design phase of system development. Don't get crazy. It doesn't solve your analysis and design problems.
- More OO mania
"You do focus primarily on the user's objects don't you?"
- OOPSLA DA Workshop
OOA and OOD is still as good as it was when I saw it in the 1970s. Bit stronger data model.



Things I Did Do

- Tried to shift focus from KB implementing technologies to user problem domain.
*"Analyze the objects **and** operations of the problem domain."*
- Tried to shift NATO software components from being stuck in the code level of abstraction.
"Reuse analysis and design, not code."
- Advocate languages over libraries for reuse.
"FORTRAN is a domain language." -SQL, VHDL, HTML, etc.
For ease of use and composition reasons.
- Advocate refinement through levels of abstraction.
"A domain object or operation is refined by implementing it in a collection of other domains."
Provides variation in implementation and refinement goals (e.g., interpreters simulations, and diagrams).
- Advocate domain specific optimizations.
"Code-level optimizers don't know anything."



Things I Will Do

1. You release a collection of objects or set of functions and types.
I'll do type dependencies among the collection and form it minimally into a protocol of usage. The grammar of the protocol serves as the domain language. Now go to step 2.
2. You release a "framework" or "architecture" with some implementing components.
I'll recast the components in existing domains and form the parametric framework into a domain language implemented by the recast components. Now go to step 3.
3. You release a domain-specific language (e.g., SDL, SQL, VHDL).
I'll generate a parser for it and bind the semantics to existing domains. Gain the benefit of analysis tools (e.g., data flow analysis). Gain the benefit of refinement variation (e.g., code, simulation, and diagrams)

I think it's a **great** idea for you to do all of the above.

