



Designing Support for Troubleshooting Complex Network Problems

Barbara Mirel
University of Michigan



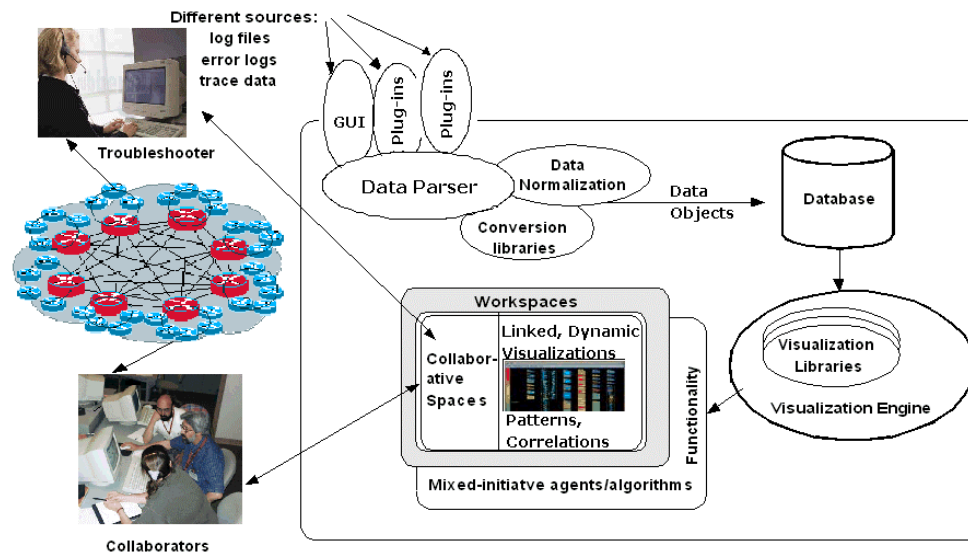
What's a complex problem?


Scenario: Videoconference disruptions

- Reality check (measures \neq consequences)
- Insufficient data to easily “locate – diagnose – fix”
- Best-effort for insolvable “butterfly effect” problem: judge trade-offs under uncertainty
- Communication across segments/organizations

What's our focus for support?

- **Integrated experiences/integrated support**
 - Standalone problem solving environment and workspace
 - Situations outside the reach of “self-healing” systems or diagnostics → augment human intelligence





1st step, user modeling: Just the right level of analysis

Designers can readily identify:

- **Low level** <*Design focus on discrete data*>

Run tests/gather data for segment-by-segment analysis

- **High level** <*Design focus on aggregates, info displays, best practices knowledge bases*>

Integrate data to recursively “locate – diagnose – fix”

Many views: functional, topology, symptom-syndrome

Balance intentions and network constraints



What have we found, cont'd?

Harder to identify:

- **Mid-level <Design focus on integrated moves & strategies in unfamiliar territory – augment human intelligence>**

“Reality checks” e.g. when measurements don’t jive with effects or intuitions can’t be trusted → Communication

Decisions about trade-offs

Accessible data /Available data

Recognition/discovery of non-obvious patterns

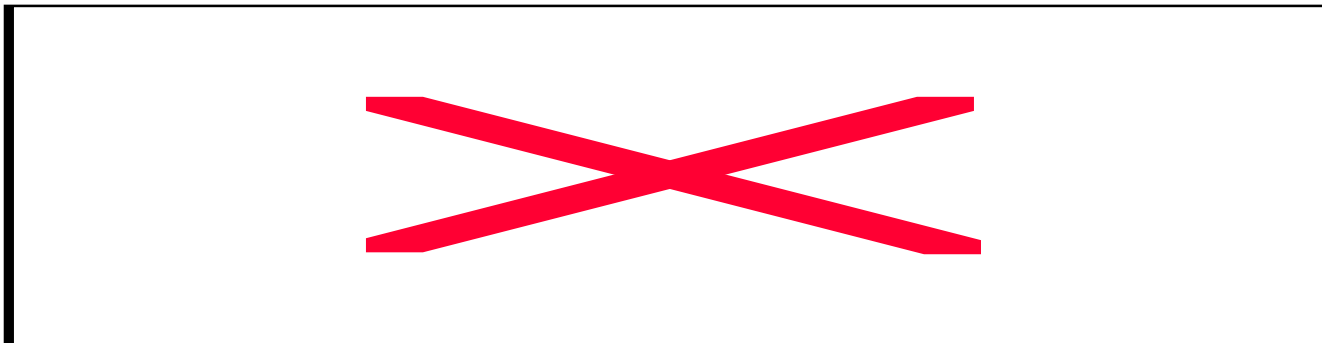
What's next?

- **Continued learning from support engineers**

Mid-level “pragmatic” challenges in problem solving

Turn models → open source framework/workspace

- **Debriefings, prerequisites, and politics**



Data, Knowledge bases, Databases, Tools, Coordination and agreed-upon roles and responsibilities, Shared meanings and criteria



NANOG, February 2004