A Modest Proposal
For Preventing Middleware Research From Being a Burden to Industry, and For Making It Beneficial to The Public

john (jonathan) swift wilkes
Middleware panel 2006
Problem: Middleware is under threat

- industry wants “relevance”
- the academic community wants “rigor”
- it’s getting too easy for others to join in

- root cause:
  - Middleware has become too simplistic
What to do about it?

• **complexity is good for Middleware**

  • employment for you + your students
  • endless chances to innovate
  • keeps others at bay
    – less risk of “we did that a decade ago”
What to do about it?
Resist misguided calls for “simplicity”

• Most problems between approaches result from **over-simplification**
  – your approach doesn’t handle this corner case
  – 100% solutions are essential
  – every deployment has unique requirements
What to do about it?
Focus on mechanisms, not goals

yes: simple-looking **interfaces**
  – can readily be measured
  (e.g., lines of code, number of methods)

no: clear **goals and metrics**
  – this requires careful thought
  – eliminates many avenues for complexity
What to do about it?
ruthlessly eliminate alternatives

Example: (a)synchronous invocations:

- it would be a disaster if people realized just how easy both techniques were to understand
- insist on just one choice
  - lots of openings for baroque workarounds
What to do about it?
Premature standardization is great!

• fossilizes poor ways of achieving end goals
  – more opportunities for exotic workarounds

• lack of consensus ➔ include all alternatives

• doesn’t require proof of usefulness
What to do about it?

Emphasize the irrelevance of customer needs

- today’s customers are stuck on yesterday’s problems
  ➔ their guidance can safely be ignored
What to do about it?
Pooh-pooh “interoperability”

• it’s just an excuse for somebody to impose their answer on you

• it requires thorough understanding of an alternative point of view
What to do about it? Obfuscatory representations

• example: why say

  {diskDrive:u
   {serialNumber "1234-5678"}
  }

• when you could have said:

  <sst:object type="diskDrive"
  name="u"> <sst:object type="serialNumber">
  <cbt:string>1234-5678</cbt:string> </sst:object> </sst:object>

only wimps can’t think in raw XML!
What to do about it?

Obfuscatory representations

which better hides the intent from outsiders?

not XML

XML
help save Middleware!

join the new ACM Campaign for Actively Complex Middleware (CACM)