It's the protocol, …

Henrik Frystyk Nielsen

henrikn@microsoft.com

Web as Computational Eco-system

- Resources are globally available with vastly more resources available on the periphery than at the core of the network.
- Resources are heterogeneous and controlled by multiple trust domains across organizational boundaries.
- Resources present themselves as services to be discovered and exploited in a loosely coupled manner rather than as bodies of code to be downloaded.
- Extensibility happens without central authority or coordination.

In this world…

- Applications become composition of concurrent services rather than monolithic entities
- Synchronization happens through communication.
- Trend is for software as well as hardware.

The problem is that…

- Current programming models have…
- Poor support for concurrency and asynchronicity
  - Using threads or hand-coded state machines is just plain hard
  - Add loose coupling and this becomes even harder
- Testing decentralized/distributed apps is equally hard
  - Programs are debugged by testing them a lot. Such testing is both expensive and incomplete in its elimination of bugs.
- Different models at different layers in the stack
  - Bridging happens though lots of glue code between different models (objects, protocols, C-style APIs…)
- Gap between specification and implementation
  - How to know whether exposed behavior is intended or buggy?
  - How to know the semantics of an interaction?
  - How to know the behavior of a service?

The Task Ahead

- Need a programming model that can harness the computational power of the Web by distributing work over the available massively parallel, asynchronous loosely coupled resources in an integral manner.